A Branch-and-cut Procedure for the Udine Course Timetabling Problem

Edmund K. Burke[1], Jakub Mareček[1,2,*], Andrew J. Parkes[1], and Hana Rudová[2]

[1]Automated Scheduling, Optimisation and Planning Group
School of Computer Science, The University of Nottingham
Wollaton Road, Nottingham NG8 1BB, UK
[2]Faculty of Informatics, Masaryk University
Botanická 68a, Brno 602 00, The Czech Republic
[*]Contact author: jxm@cs.nott.ac.uk

August 20, 2008

Integer Programming: A Quick Guide I

1. Come up with an encoding of a solution
2. Come up with variables suitable for that encoding
3. Come up with a set of linear equations in those variables, satisfied iff the variables encode a feasible solution

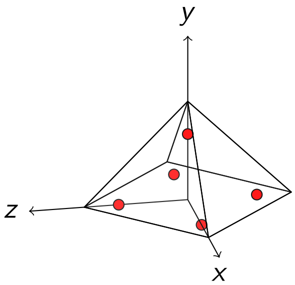

Figure: A poor relaxation.

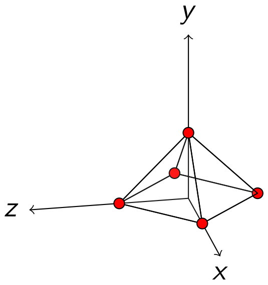

Figure: The many facets of a convex hull of integer points.

Integer Programming: A Quick Guide II

Internally, the solvers use "branch and bound", a form of divide and conquer tree search:

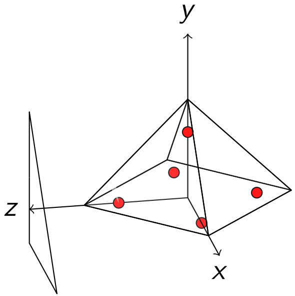

Figure: Either branch on a variable by adding $x \geq 2$ ...

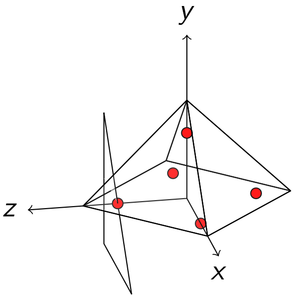

Figure: ... or by adding $x \leq 1$.

Integer Programming: A Quick Guide III

4. Optionally, come up with some more linear equations ("cuts") that have to be satisfied for each feasible solution
5. Optionally, show that those inequalities are facets of the polytope (i.e. full-dimensional)

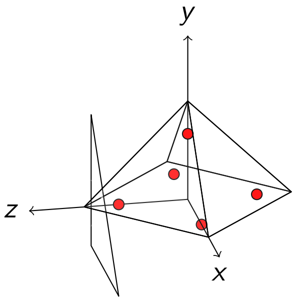

Figure: Pure cutting plane methods can go for ever ...

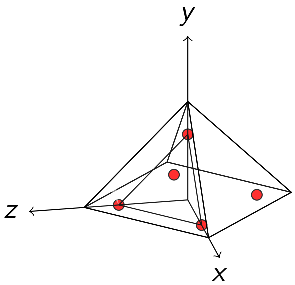

Figure: ... even if your cuts are facet-defining. Why?

Integer Programming: A Quick Guide IV

6. Optionally, decide what variables could be added ("priced-in") only later on
7. Optionally, come up with a (heuristic) decomposition

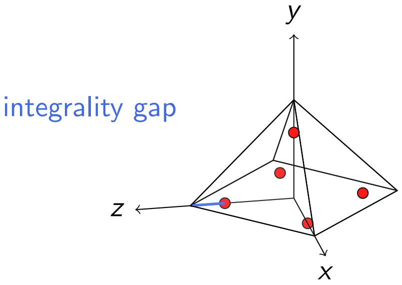

Figure: "Integrality gap" is the distance of lower and upper bounds.

Udine Course Timetabling I

- Udine Course Timetabling is Track 3 of the International Timetabling Competition 2007 [Gaspero and Schaerf, 2006]
- An extension of a pre-colouring with a bounded number of uses of each colour
- There is an edge between two vertices of the conflict graph if the corresponding events cannot take place at the same time
- Events are partitioned into disjoint subsets, called courses; events of a course take place at different times, and are freely interchangeable
- A small number of distinct, possibly overlapping, sets of courses, representing enrolments prescribed to various groups of students, are referred to as curricula
- The goal is to find a feasible bounded colouring, minimising the number of violations of four soft constraints

Udine Course Timetabling II

Input to Udine Course Timetabling:

- C, U, T, R, D, P are sets of courses, curricula, teachers, rooms, days, and periods, respectively
- $\tilde{U}^u$ is the non-empty set of courses in curriculum u
- N is a subset of C $\times$ P, giving forbidden course-period combinations
- E^c is the number of events course c has in a week
- S^c is the number of students enrolled in course c
- M^c is the prescribed minimum number of distinct week-days of instruction for course c
- $\tilde{T}^t$ is the subset of courses C taught by teacher t
- A^r is the capacity of room r

Udine Course Timetabling III

Further notation:

- $\tilde{D}^d$ is the subtuple (ordered subset) of P corresponding to periods in day d
- W is a vector of non-negative weights $(\mathrm{W}^{\mathrm{RCap}}, \mathrm{W}^{\mathrm{TSpr}}, \mathrm{W}^{\mathrm{TCom}}, \mathrm{W}^{\mathrm{RStb}})$ for the four soft constraints described below.

Udine at Nottingham I

- Dec 2006: An integer programming (IP) formulation of the original problem
- Feb 2007: A debugged IP formulation
- May 2007: A heuristic decomposition
  Poster at MIP: "Uses and Abuses of MIP in Timetabling"
- Jun 2007: Further formulations of graph colouring
  Talk at EURO: "Variability of IP Models of Course Timetabling"
- Aug 2007: Further formulations of soft constraints
  Paper at GOR: "Penalising Patterns in Timetables"

Udine at Nottingham II

- Oct 2007: Optimal transformation of graph colouring to multi-colouring and empirical tests
  Under revision in Annals of OR: "On a Clique-based IP Formulation of Vertex Colouring"
- Jan 2008: An improved version of the heuristic decomposition
  Under revision in Computers and OR: "Decomposition, Reformulation, and Diving in Timetabling"
- Feb 2008: A branch-and-cut using yet another formulation of soft constraints
  PATAT: "A Branch and Cut Procedure for Udine Course Timetabling"
- Jul 2008: A revised version of the heuristic decomposition

The Formulation

- Encoding
- Variables
- Hard constraints (using core variables)
- Soft constraints (using additional variables)
- The objective function

The Variables

Core variables:

- $x_{p,r,c} = 1$, iff course c is taught in room r at period p

Additional variables:

- $v_{d,c} = 1$, iff there is any event of course c held on day d
- $y_{r,c} = 1$, iff room r is used by course c
- m_c is the number of days course c is short of the recommended days of instruction
- $z_{u,d} \geq 1$, iff at least one pattern-penalising constraint is violated in the timetable for curriculum u and day d given by the solution. Higher values of $z_{u,d}$ are enforced only dynamically.

The Hard Constraints I

A given number of events, E^c, should be taught for each course c:

$$\forall c \in \mathrm{C} \qquad \sum_{p \in \mathrm{P}} \sum_{r \in \mathrm{R}} x_{p,r,c} = \mathrm{E}^c \tag{1}$$

At most one event can be taught in a single room at any period:

$$\forall p \in \mathrm{P}\, \forall r \in \mathrm{R} \qquad \sum_{c \in \mathrm{C}} x_{p,r,c} \leq 1 \tag{2}$$

At most one event of a single course can be taught at any period:

$$\forall p \in \mathrm{P}\, \forall c \in \mathrm{C} \qquad \sum_{r \in \mathrm{R}} x_{p,r,c} \leq 1 \tag{3}$$

The Hard Constraints II

At most one event taught by a single teacher can be taught at any period:

$$\forall p \in \mathrm{P}\, \forall t \in \mathrm{T} \qquad \sum_{r \in \mathrm{R}} \sum_{c \in \tilde{T}^t} x_{p,r,c} \leq 1 \tag{4}$$

At most one event within a single curriculum can be taught at any period:

$$\forall p \in \mathrm{P}\, \forall u \in \mathrm{U} \qquad \sum_{r \in \mathrm{R}} \sum_{c \in \tilde{U}^u} x_{p,r,c} \leq 1 \tag{5}$$

Some events cannot take place at some periods:

$$\forall \langle c, p \rangle \in \mathrm{N} \qquad \sum_{r \in \mathrm{R}} x_{p,r,c} = 0 \tag{6}$$

The Soft Constraints I

Summarise the course-day assignment into $v_{d,c}$:

$$\forall c \in \mathrm{C}\, \forall d \in \mathrm{D}\, \forall p \in \tilde{D}^d \qquad \sum_{r \in \mathrm{R}} x_{p,r,c} \leq v_{d,c} \tag{7}$$

$$\forall c \in \mathrm{C}\, \forall d \in \mathrm{D} \qquad \sum_{r \in \mathrm{R}} \sum_{p \in \tilde{D}^d} x_{p,r,c} \geq v_{d,c} \tag{8}$$

Compare this number with the prescribed number M^c and set $'m_c$ to the difference, if course c lacks some:

$$\forall c \in \mathrm{C} \qquad \sum_{d \in \mathrm{D}} v_{d,c} \geq \mathrm{M}^c - m_c \tag{9}$$

Summarise the course-room assignment into y:

$$\forall p \in \mathrm{P}\, \forall r \in \mathrm{R}\, \forall c \in \mathrm{C} \qquad x_{p,r,c} \leq y_{r,c} \tag{10}$$

The Soft Constraints II

Penalise patterns in daily timetables of individual curricula. First check isolated events in the first and the last period of the day, e.g.:

$$\forall u \in \mathrm{U}, d \in \mathrm{D}, \forall \langle p_1, p_2, p_3, p_4 \rangle \in \tilde{D}^d$$

$$\sum_{c \in \tilde{U}^u} \sum_{r \in \mathrm{R}} (x_{p_1,r,c} - x_{p_2,r,c}) \leq z_{u,d} \tag{11}$$

$$\sum_{c \in \tilde{U}^u} \sum_{r \in \mathrm{R}} (x_{p_4,r,c} - x_{p_3,r,c}) \leq z_{u,d} \tag{12}$$

The Soft Constraints III

Then look for triples of consecutive periods with only the middle period occupied by an event, e.g.:

$$\forall u \in \mathrm{U}, d \in \mathrm{D}, \forall \langle p_1, p_2, p_3, p_4 \rangle \in \tilde{D}^d$$

$$\sum_{c \in \tilde{U}^u} \sum_{r \in \mathrm{R}} (x_{p_2,r,c} - x_{p_1,r,c} - x_{p_3,r,c}) \leq z_{u,d} \tag{13}$$

$$\sum_{c \in \tilde{U}^u} \sum_{r \in \mathrm{R}} (x_{p_3,r,c} - x_{p_2,r,c} - x_{p_4,r,c}) \leq z_{u,d} \tag{14}$$

The Objective: Minimise a Weighted Sum

- Students left without a seat

$$\mathrm{W}^{\mathrm{RCap}} \sum_{r \in \mathrm{R}} \sum_{p \in \mathrm{P}} \sum_{\substack{c \in \mathrm{C} \\ \mathrm{S}^c > \mathrm{A}^r}} x_{p,r,c} \left(\mathrm{S}^c - \mathrm{A}^r \right)$$

- Violations of compactness

$$\mathrm{W}^{\mathrm{TCom}} \sum_{u \in \mathrm{U}} \sum_{d \in \mathrm{D}} z_{u,d}$$

- Missing days of instruction

$$\mathrm{W}^{\mathrm{TSpr}} \sum_{c \in \mathrm{C}} m_c$$

- Changes of course-room allocations

$$\mathrm{W}^{\mathrm{RStb}} \sum_{c \in \mathrm{C}} \left(\left(\sum_{r \in \mathrm{R}} y_{r,c} \right) - 1 \right)$$

The Cuts

Necessary ("Type 1"):

- Cuts from "event/free-period" patterns

Optional ("Type 2"):

- Cuts from implied bounds
- Cuts from "day of instruction/day off" patterns
- "Lifted clique cuts" from graph colouring

Future work:

- Further cuts from graph colouring
- Cuts from bipartite weighted matching

Cuts from Event/Free-Period Patterns

- Constraints (11–14) don't force $z_{u,d} > 1$
- This can be done by enumeration of daily event/free-period patterns [Burke et al., 2008a]
- Patterns in daily timetables of individual curricula can also be thought of as $A = \langle a_1, \ldots, a_{|\mathrm{D}|} \rangle$
- a_i is equal to one if there is an event in the period i, -1 otherwise
- Penalisation with penalty p and $m = \sum_{a \in A, a=1}$:

$$\forall u \in \mathrm{U} \;\; \forall d \in \mathrm{D} \;\; \forall \langle p_1, p_2, \ldots, p_n \rangle \in \tilde{D}^d$$
$$p \left(-m + \sum_{i=1}^{n} \left(a_i \sum_{c \in \tilde{U}^u} \sum_{r \in \mathrm{R}} w_{c,p_i} \right) \right) \leq z_{u,d} \quad (15)$$

Cuts from Implied Bounds I

- There have to be at least E^c events of course c in the weekly timetable.
- Constraints (7–8) set $v_{d,c}$ to one, if and only if there is an event of course c in the timetable for day d.
- Logically, it follows that for each course c with $\mathrm{E}^c > 0$, at least one of $v_{d,c}$ has to be set to one:

$$\forall c \in \mathrm{C} \qquad \sum_{d \in \mathrm{D}} v_{d,c} \geq 1 \tag{16}$$

$$\forall c \in \mathrm{C} \qquad \sum_{r \in \mathrm{D}} v_{d,c} \leq \mathrm{E}^c \tag{17}$$

Cuts from Implied Bounds II

- Similarly, events of each course c take place at least in one room, but in fewer rooms than E^c:

$$\forall c \in \mathrm{C} \qquad \sum_{r \in \mathrm{R}} y_{r,c} \geq 1 \qquad (18)$$

$$\forall c \in \mathrm{C} \qquad \sum_{r \in \mathrm{R}} y_{r,c} \leq \mathrm{E}^c \qquad (19)$$

- There has to be at least one day of instruction, if all events have to be timetabled:

$$\forall c \in \mathrm{C} \qquad m_c \leq \mathrm{M}^c - 1 \qquad (20)$$

Cuts from Implied Bounds III

- We can also add a lower bound for course-room assignment $y_{r,c}$:

$$\forall r \in \mathrm{R}\, \forall c \in \mathrm{C} \qquad \sum_{p \in \mathrm{P}} x_{p,r,c} \geq y_{r,c} \tag{21}$$

- Or a stronger version thereof:

$$\forall r \in \mathrm{R}\, \forall c \in \mathrm{C} \qquad \sum_{p \in \mathrm{P}} x_{p,r,c} \leq \mathrm{E}^{c} y_{r,c} \tag{22}$$

Cuts from Day of Instruction/Day off Patterns I

- For all courses c where m_c is zero, the number of events taking place on a single day cannot be higher than one plus the number of events not necessary to maitain the spread of events throughout the week $(\mathrm{E}^c - \mathrm{M}^c)$
- Hence:

$$\forall c \in \mathrm{C}\, \forall d \in \mathrm{D} \qquad \sum_{p \in \tilde{D}^d} \sum_{r \in \mathrm{R}} x_{p,r,c} \leq 1 + \mathrm{E}^c - \mathrm{M}^c + m_c \tag{23}$$

Cuts from Day of Instruction/Day off Patterns II

- This constraint then can be naturally extended to cover two days:

$$\forall c \in \mathrm{C}\, \forall d_1 \in \mathrm{D}\, \forall d_2 \in \{d \in \mathrm{D} \mid d > d_1\}$$
$$\sum_{p \in \tilde{D}^{d_1} \cup \tilde{D}^{d_2}} \sum_{r \in \mathrm{R}} x_{p,r,c} \leq 2 + \mathrm{E}^c - \mathrm{M}^c + m_c \quad (24)$$

- In general, it is possible to come up with constraints linking an arbitrary number n of days:

$$\forall c \in \mathrm{C}\, \forall d_1 \in \mathrm{D}\, \forall d_2 \in \{d \in \mathrm{D} \mid d > d_1\} \ldots \forall d_n \in \{d \in \mathrm{D} \mid d > d$$
$$\sum_{p \in \bigcup_{i=1}^{n} \tilde{D}^{d_i}} \sum_{r \in \mathrm{R}} x_{p,r,c} \leq n + \mathrm{E}^c - \mathrm{M}^c + m_c \quad (25)$$

Cuts from Graph Colouring

- The usual linear programming relaxations is weak [Caprara, 1998]
- Known classes of cuts [Coll et al., 2002,?, Campêlo et al., 2003, Méndez-Díaz and Zabala, 2008]
- Clique cuts, the most potent class, correspond to:

$$\forall q \in \mathrm{Q}\, \forall p \in \mathrm{P} \qquad \sum_{r \in \mathrm{R}} \sum_{c \in q} x_{p,r,c} \leq 1, \tag{26}$$

where Q is a collection of complete subgraphs in the course-based conflict graph.

Future work: More Cuts from Graph Colouring

Coll et al. [2002], Campêlo et al. [2003] and Méndez-Díaz and Zabala [2008] also provide proofs of validity and full-dimensionality of several further classes of cuts:

- cuts from odd holes
- cuts from odd antiholes
- cuts from paths
- symmetry-breaking block colour inequalities
- neighbourhood inequalities replacing edgewise inequalities
- multicolour inequalities strengthening edgewise inequalities.

Their performance in timetabling context remains unknown.

Future work: Cuts from Bipartite Weighted Matching

Balas and Saltzman [1991], Gwan and Qi [1992] and Balas and Qi [1993] have proposed a number of strong valid inequalities for the three-index assignment problem, including:

- cuts from combs
- cuts from bulls.

This seems to be Gerald Lach's domain. ;-)

Computational Experience I

- Implemented in C++ with ILOG CPLEX 10 [ILOG, 2006]
- Cuts from implied bounds used statically
- Trivial separation of cuts from days of instruction / day off patterns
- Trivial separation of cuts from event / free period patterns
- Trivial separation of cuts from cliques
- Tuning of the choice of LP solver, based on a much smaller LP instance

Computational Experience II

- Times on a Intel Pentium 4 (3.20 GHz, 2 GB RAM) normalised using the benchmark (780 s)
- By using the tuning and a reduced formulation, the maximum root LP solve time decreases from 6489.69s (CPLEX 11!) to 197 s
- Standard CPLEX 10 using the basic formulation takes 48 hours to compute the LB of 33 for comp05, after tuning the solver for "best bound" and using cuts from implied bounds
- When we apply cuts from patterns, it is possible to obtain the lower bound of 183 within 30 minutes
- Comparison with Lach and Lübbecke [2008a] and Müller [2008]

Computational Experience III

	Monolithic 30m	This + IB 30m	Surface + IB 30m	This + IB + C 30m	This + IB + P 30m	Lach + CPLEX11 root	Lach + CPLEX11 58.5m	Müeller's UB 10×12.6m
comp01	-11	4	5	4	5	4	4	5
comp02	-59	0	0	0	6	0	8	51
comp03	-53	0	2	0	43	1	23	84
comp04	-56	0	0	0	2	12	27	37
comp05	-29	17	33	26	183	93	101	330
comp06	-71	6	6	6	6	7	7	48
comp07	-98	0	0	0	0	0	0	20

Computational Experience IV

	Monolithic 30m	This + IB 30m	Surface + IB 30m	This + IB + C 30m	This + IB + P 30m	Lach + CPLEX11 root	Lach + CPLEX11 58.5m	Müeller's UB 10 × 12.6m
comp08	-56	0	1	0	2	2	34	41
comp09	-57	0	6	0	0	19	40	109
comp10	-86	0	0	0	0	2	4	16
comp11	-17	0	0	0	0	0	0	0
comp12	-58	7	12	4	5	31	32	333
comp13	-56	0	4	0	0	20	37	66
comp14	-62	0	23	0	0	40	41	59

Conclusions

- Branch and cut is good for producing lower bounds
- Reformulation tricks are fun to play
- Optima can be expensive

- There's plenty of "future work" outlined in the paper
- We are happy to share source code and experience

- `http://cs.nott.ac.uk/~jxm`

Acknowledgments

The authors are grateful to Andrea Schaerf and Luca Di Gaspero, who kindly maintain the Udine Course Timetabling Problem, including all data sets. Andrew Parkes has been supported by the UK Engineering and Physical Sciences Research Council under grant GR/T26115/01. Hana Rudová has been supported by project MSM0021622419 of Ministerstvo školství, mládeže a tělovýchovy and by the Grant Agency of the Czech Republic under project 201/07/0205.

References I

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: Theory, algorithms, and applications*. Prentice-Hall, 1993.

Salem Mohammed Al-Yakoob and Hanif D. Sherali. A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations. *European J. Oper. Res.*, page In press., 2007.

Pasquale Avella and Igor Vasil'ev. A computational study of a cutting plane algorithm for university course timetabling. *J. Scheduling*, 8(6): 497–514, 2005.

Egon Balas and Li Qun Qi. Linear-time separation algorithms for the three-index assignment polytope. *Discrete Appl. Math.*, 43(1):1–12, 1993.

Egon Balas and Matthew J. Saltzman. An algorithm for the three-index assignment problem. *Oper. Res.*, 39(1):150–161, 1991.

References II

Victor A. Bardadym. Computer-aided school and university timetabling: The new wave. In Edmund K. Burke and Peter Ross, editors, *Practice and Theory of Automated Timetabling*, volume LNCS 1153 of *Lecture Notes in Computer Science*, pages 22–45, Berlin, 1996. Springer.

Camille B. Beyrouthy, Edmund K. Burke, Dario Landa Silva, Barry McCollum, Peter McMullan, and Andrew J. Parkes. Towards improving the utilisation of university teaching space. *J. Op. Res. Soc.*, to appear, 2007. Also appeared as NOTTCS-TR-2006-5.

Edmund K. Burke and Patrick De Causmaecker, editors. *Practice and Theory of Automated Timetabling*, volume LNCS 2740 of *Lecture Notes in Computer Science*, Berlin, 2003. Springer.

Edmund K. Burke and Sanja Petrovic. Recent research directions in automated timetabling. *European J. Oper. Res.*, 140(2):266–280, 2002.

References III

Edmund K. Burke, Kirk Jackson, Jeffrey H. Kingston, and Rupert F. Weare. Automated university timetabling: The state of the art. *Comput. J.*, 40(9):565–571, 1997.

Edmund K. Burke, Dominique de Werra, and Jeffrey H. Kingston. Applications to timetabling. In Jonathan L. Gross and Jay Yellen, editors, *Handbook of Graph Theory*, pages 445–474. CRC, London, UK, 2004.

Edmund K. Burke, Jakub Mareček, Andrew J. Parkes, and Hana Rudová. On a clique-based integer programming formulation of vertex colouring with applications in course timetabling. Technical Report NOTTCS-TR-2007-10, The University of Nottingham, Nottingham, 2007. Also available at http://arxiv.org/abs/0710.3603.

Edmund K. Burke, Jakub Mareček, Andrew J. Parkes, and Hana Rudová. Penalising patterns in timetables: Novel integer programming formulations. In Stefan Nickel and Jörg Kalcsics, editors, *Operations Research Proceedings 2007*, Berlin, 2008a. Springer.

References IV

Edmund K. Burke, Jakub Mareček, Andrew J. Parkes, and Hana Rudová. Decomposition, reformulation, and diving in timetabling. Technical Report NOTTCS-TR-2008-02, The University of Nottingham, Nottingham, 2008b.

Manoel Campêlo, Ricardo C. Corrêa, and Yuri Frota. Cliques, holes and the vertex coloring polytope. *Inform. Process. Lett.*, 89(4):159–164, 2003.

Alberto Caprara. Properties of some ILP formulations of a class of partitioning problems. *Discrete Appl. Math.*, 87(1-3):11–23, 1998.

Michael W. Carter. A Lagrangian relaxation approach to the classroom assignment problem. *INFORMS J. Comput.*, 27:230–245, 1989.

Michael W. Carter and Gilbert Laporte. Recent developments in practical course timetabling. In Edmund K. Burke and Michael W. Carter, editors, *Practice and Theory of Automated Timetabling*, volume LNCS 1408 of *Lecture Notes in Computer Science*, pages 3–19, Berlin, 1997. Springer.

References V

Fabio De Cesco, Luca Di Gaspero, and Andrea Schaerf. Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, and results. In *Practice and Theory of Automated Timetabling*, to appear, 2008.

Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Math.*, 4:305–337, 1973.

Pablo Coll, Javier Marenco, Isabel Méndez-Díaz, and Paula Zabala. Facets of the graph coloring polytope. *Ann. Oper. Res.*, 116:79–90, 2002.

Artur Czumaj and Andrzej Lingas. Finding a heaviest triangle is not harder than matrix multiplication. In *SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 986–994, Philadelphia, PA, 2007. SIAM.

Sophia Daskalaki, Theodore Birbas, and Efthymios Housos. An integer programming formulation for a case study in university timetabling. *European J. Oper. Res.*, 153:117–135, 2004.

References VI

Sophia Daskalaki, Theodore Birbas, and Efthymios Housos. Efficient solutions for a university timetabling problem through integer programming. *European J. Oper. Res.*, 160:106–120, 2005.

Maria Dimopoulou and Panagiotis Miliotis. An automated university course timetabling system developed in a distributed environment: A case study. *European J. Oper. Res.*, 153:136–147, 2004.

Luca Di Gaspero and Andrea Schaerf. Multi neighborhood local search with application to the course timetabling problem. In Burke and Causmaecker [2003], pages 262–275.

Luca Di Gaspero and Andrea Schaerf. Neighborhood portfolio approach for local search applied to timetabling problems. *J. Math. Model. Algorithms*, 5(1):65–89, 2006.

Luca Di Gaspero, Barry McCollum, , and Andrea Schaerf. The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (Track 3). Technical Report 2007/08/01, University of Udine DIEGM, Udine, Italy, 2007.

References VII

Geena Gwan and Li Qun Qi. On facets of the three-index assignment polytope. *Australas. J. Combin.*, 6:67–87, 1992.

ILOG. *ILOG CPLEX Advanced Programming Techniques*. ILOG S. A., Incline Village, NV, 2006.

Gerald Lach and Marco E. Lübbecke. Curriculum based course timetabling: Optimal solutions to the udine benchmark instances. In *Practice and Theory of Automated Timetabling*, to appear, 2008a.

Gerald Lach and Marco E. Lübbecke. Optimal university course timetables and the partial transversal polytope. In Catherine C. McGeoch, editor, *Experimental algorithms*, volume LNCS 5038 of *Lecture Notes in Computer Science*, pages 235–248, Berlin, 2008b. Springer.

N. L. Lawrie. An integer linear programming model of a school timetabling problem. *Comput. J.*, 12:307–316, 1969.

References VIII

Barry McCollum. A perspective on bridging the gap between theory and practice in university timetabling. In Edmund K. Burke and Hana Rudová, editors, *Practice and Theory of Automated Timetabling*, volume LNCS 3867 of *Lecture Notes in Computer Science*, pages 3–23, Berlin, 2007. Springer.

Isabel Méndez-Díaz and Paula Zabala. A cutting plane algorithm for graph coloring. *Discrete Appl. Math.*, 156(2), 2008.

S. A. Mirhassani. A computational approach to enhancing course timetabling with integer programming. *Appl. Math. Comput.*, 175: 814–822, 2006.

Tomáš Müller. ITC-2007 solver description: A hybrid approach. In *Practice and Theory of Automated Timetabling*, to appear, 2008.

References IX

Keith Murray, Tomáś Müller, and Hana Rudová. Modeling and solution of a complex university course timetabling problem. In Edmund K. Burke and Hana Rudová, editors, *Practice and Theory of Automated Timetabling*, volume LNCS 3867 of *Lecture Notes in Computer Science*, pages 193–213, Berlin, 2007. Springer.

Sanja Petrovic and Edmund K. Burke. University timetabling. In Joseph Leung, editor, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, pages 1001–1023. CRC Press, Boca Raton, FL, 2004. ISBN 1584883979.

Andrea Qualizza and Paolo Serafini. A column generation scheme for faculty timetabling. In Edmund K. Burke and Michael A. Trick, editors, *Practice and Theory of Automated Timetabling*, volume LNCS 3616 of *Lecture Notes in Computer Science*, pages 161–173, Berlin, 2004. Springer.

Hana Rudová and Keith Murray. University course timetabling with soft constraints. In Burke and Causmaecker [2003], pages 310–328.

References X

Andrea Schaerf. A survey of automated timetabling. *Artificial Intelligence Rev.*, 13(2):87–127, 1999.

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

Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theor. Comput. Sci.*, 363(1):28–42, 2006.

Arabinda Tripathy. School timetabling – A case in large binary integer linear programming. *Management Sci.*, 30:1473–1489, 1984.

Milan Vlach. Branch and bound method for the three-index assignment problem. *Ekonom.-Mat. Obzor*, 3:181–191, 1967.

Dominic J. A. Welsh, and Martin B. Powel. An upper bound for the chromatic number of a graph and its application to timetabling problems. *Computer J.*, 10(1):85–86, 1967.

References XI

Paula Zabala and Isabel Méndez-Díaz. A branch-and-cut algorithm for graph coloring. *Discrete Appl. Math.*, 154(5):826–847, 2006.

David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Comput.*, 3(6):103–128, 2007.

	Monolithic formulation		Proposed formulation		
	LP dim.	Barrier	LP dim.	D. Simplex	Barrie
01	6484 × 5760	18.91 s	6516 × 5500	27.70 s	3.58
02	30034 × 27693	87.58 s	30128 × 26703	1185.02 s	54.90
03	26862 × 25489	80.91 s	26941 × 24563	674.44 s	49.97
04	33608 × 31265	63.83 s	33698 × 30525	1191.92 s	41.00
05	16189 × 14859	77.45 s	16259 × 12129	149.74 s	84.64
06	44050 × 41120	186.71 s	44168 × 40113	1798.53 s	73.17
07	60611 × 57012	510.05 s	60745 × 55895	1798.75 s	192.35
08	35642 × 33180	71.06 s	35735 × 32397	1175.29 s	43.25
09	32308 × 29979	88.31 s	32391 × 29024	1085.17 s	48.23
10	45870 × 43257	246.26 s	45996 × 42279	1798.97 s	105.03
11	8702 × 7126	9.31 s	8733 × 6672	26.03 s	7.69
12	27552 × 25007	295.08 s	27652 × 22117	669.02 s	134.76
13	35597 × 33200	70.13 s	35691 × 32353	1176.16 s	37.34
14	33288 × 30872	83.65 s	33384 × 30057	987.88 s	55.86