



## Flow Formulations for Curriculum-based Course Timetabling

**Bagger, Niels-Christian Fink; Kristiansen, Simon; Sørensen, Matias; Stidsen, Thomas Jacob Riis**

*Published in:*  
Annals of Operations Research

*Link to article, DOI:*  
[10.1007/s10479-018-3096-4](https://doi.org/10.1007/s10479-018-3096-4)

*Publication date:*  
2019

*Document Version*  
Peer reviewed version

[Link back to DTU Orbit](#)

*Citation (APA):*  
Bagger, N-C. F., Kristiansen, S., Sørensen, M., & Stidsen, T. J. R. (2019). Flow Formulations for Curriculum-based Course Timetabling. *Annals of Operations Research*, 280(1-2), 121-150. <https://doi.org/10.1007/s10479-018-3096-4>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

## Flow Formulations for Curriculum-based Course Timetabling

Niels-Christian F. Bagger · Simon  
Kristiansen · Matias Sørensen · Thomas R.  
Stidsen

Received: date / Accepted: date

**Abstract** In this paper we present two mixed-integer programming formulations for the Curriculum based Course Timetabling Problem (CTT). We show that the formulations contain underlying network structures by dividing the CTT into two separate models and then connect the two models using flow formulation techniques. The first mixed-integer programming formulation is based on an underlying minimum cost flow problem, which decreases the number of integer variables significantly and improves the performance compared to an intuitive mixed-integer programming formulation. The second formulation is based on a multi-commodity flow problem which in general is  $NP$ -hard, however, we prove that it suffices to solve the linear programming relaxation of the model. The formulations show competitiveness with other approaches based on mixed-integer programming from the literature and improve the currently best known lower bound on one data instance in the benchmark data set from the second international timetabling competition. Regarding upper bounds, the formulation based on the minimum cost flow prob-

---

The authors would like to thank Professor Stephan Røpke, Department of Management Engineering, Technical University of Denmark, and Professor Carsten Thomassen, Department of Applied Mathematics and Computer Science, Technical University of Denmark for fruitful discussions on the graphs and proofs. Furthermore, the authors would like to thank the anonymous reviewers for their valuable feedback to improve the manuscript. Niels-Christian Fink Bagger's industrial PhD project was funded by Innovation Fund Denmark (IFD). IFD has supported the work solely financially and has not participated in any research related activities.

---

N.C. F. Bagger, Corresponding author  
ORCID: [0000-0003-4665-6761](https://orcid.org/0000-0003-4665-6761)  
E-mail: [nbag@dtu.dk](mailto:nbag@dtu.dk), [bagger.nc@gmail.com](mailto:bagger.nc@gmail.com)

N.C. F. Bagger · M. Sørensen  
MaCom A/S, Vesterbrogade 48, 1., DK-1620 København V, Denmark

N.C. F. Bagger · M. Sørensen · T. R. Stidsen  
mORetime research group, Management Science, Department of Management Engineering,  
Technical University of Denmark, Produktionstorvet, Building 426B, DK-2800 Kgs. Lyngby,  
Denmark, <http://www.moretime.man.dtu.dk/>

S. Kristiansen  
RHA Software Group, Frederikkevej 2B, DK-2900 Hellerup Denmark

lem performs better on average than other mixed integer programming approaches for the CTT.

**Keywords** University Course Timetabling · Integer Programming · Minimum Cost Flow · Multi-Commodity Flow

## 1 Description and Literature

Each semester, universities face the problem of generating high-quality course timetables. A timetable determines when and where a course should take place. In this work we consider the Curriculum based Course Timetabling (CCT) Problem in which weekly lectures for multiple courses have to be scheduled and assigned to rooms. A week is divided into days (usually five or six) and each day is divided into time slots. We refer to a day and time slot combination as a period. The problem was introduced in track 3 of the second international timetabling competition (ITC2007) as described by [Di Gaspero et al \(2007\)](#), [McCollum et al \(2010\)](#) and [Bonutti et al \(2012\)](#).

The basic entities of the problem are the *courses* to schedule, and the *periods* and *rooms* that are available. Therefore, when formulating a Mixed Integer Program (MIP), it seems natural to define binary variables with three indices corresponding to a course, period and room where the binary variable would then take value one if the course is scheduled in the specified room at the period. Formulating the MIP of the problem this way is the most commonly used, see [Lach and Lübbecke \(2008\)](#). We show that such a formulation will create an unnecessarily large amount of binary variables and instead we formulate two MIPs containing a significantly smaller number of integer variables. The goal of the formulations is to decrease the number of integer variables, and we show that we can define some of the variables as continuous variables instead of integer variables. Throughout this paper, we go through numerous formulations and notations, and this, have provided an overview of all the notations we use in [Appendix A](#).

Besides the courses, periods and rooms the problem also contains *lecturers* and *curricula*, hence the name *Curriculum-based Course Timetabling*. Each course is taught by a lecturer, and a curriculum is a set of courses which may be followed by the same students. The resulting timetable must fulfill some specific hard constraints:

**Availability (A):** For each course a subset of the periods (maybe all the periods) are denoted as the *available* periods. It is not allowed to schedule a lecture in a period that is not available for the corresponding course.

**Lectures (L):** All lectures of all courses must be scheduled, and lectures must be scheduled in different periods.

**Conflicts (C):** If two courses are taught by the same lecturer or the courses are part of the same curriculum, the courses cannot be taught in the same period.

**Room Occupancy (RO):** A room can at most be occupied by one course in any period.

Besides the hard constraints, the problem also contains the following soft constraints:

**Room Capacity (RC):** If a lecture is scheduled in a room where the capacity is smaller than the number of students attending the course, then each student above the capacity is counted as a violation.

**Isolated Lectures (IL):** It is desired to schedule lectures from the same curriculum in adjacent periods. Two periods are considered to be adjacent if they belong to the same day and are in consecutive time slots. If a lecture from a curriculum is scheduled in a period and no lecture from the same curriculum is scheduled in an adjacent period, the lecture is denoted as being *isolated*. Every time there is an isolated lecture this counts as one violation.

**Minimum Working Days (MWD):** For each course, it is desired to spread the lectures across a given number of days. If the number of days that a course is scheduled is below this number, then the violation is the difference between the requested and the actual number of days that has been scheduled.

**Room Stability (RStab):** Each course should not be assigned to too many different rooms during the week. If a course is scheduled in at least two distinct rooms during the week, then the violation is the total number of distinct rooms assigned to the course minus one.

The objective of the CTT is to find a solution which fulfills all the hard constraints and minimizes a weighted sum of the violations of the soft constraints using non-negative weights.

Many researchers have considered the CTT since the ITC2007 competition, and we refer to [Bettinelli et al \(2015\)](#) for an excellent overview. Since we are formulating a MIP model of the problem, we mainly focus on other MIP based approaches in the literature in the remainder of this paper.

[Burke et al \(2008, 2010\)](#) introduced a MIP formulation named the *monolithic* formulation, based on the intuition of the three-index binary variables (for each course, period and room a binary variable is defined). The monolithic formulation is an exact model in the sense that it can be solved by a generic MIP solver to obtain the optimal solution, assuming that enough computational resources are available for the MIP solver. Unfortunately, many instances of the CTT cannot be solved within a reasonable time using a MIP solver for the monolithic formulation. Therefore [Burke et al \(2010\)](#) propose methods to derive lower and upper bounds based on the monolithic formulation. They obtain one lower bound by ignoring the soft constraints **RC** and **RStab**, which gives the possibility to ignore the assigning of rooms in the formulation. They add a constraint to ensure that no more lectures are scheduled in any period than the number of rooms that are available. They note that the constraint is equivalent to aggregating all the rooms into a single room with a capacity equal to the largest room and the number of lectures that can be scheduled in this room in a period is equal to the total number of rooms. They use this observation to generate another lower bound by aggregating the rooms into *multi-rooms*. A multi-room is a set of rooms where the capacity is equal to the capacity of the largest room in the set and the number of lectures that can be scheduled in the multi-room in a period is equal to the number of rooms in the set. In the tests, they show that the latter method provides stronger bounds on average. After they obtain a solution from one of the latter methods a *diving* heuristic is applied by taking the monolithic formulation and fixing the periods based on the solution from the lower bounding method to obtain a full solution. In a more recent paper, [Burke et al \(2012\)](#) give an exact branch-and-cut algorithm

which they also base on the monolithic formulation, but some of the objective costs are left out and instead added as cuts during the solution process. Furthermore, some valid inequalities are presented, which the MIP solver can take advantage of during the search. Their computational results show that the cutting plane approach leads to a better performance.

Lach and Lübbecke (2008, 2012) propose a method that divides the CTT into two stages, where they formulate each stage as a MIP model. In the first stage, they schedule the courses into periods, which only requires a binary variable for each course and period. They ensure that all the hard constraints are satisfied in the first stage problem. However only the soft constraints **MWD**, **IL** and **RC** can be taken into account. They include the **RC** constraints in the first stage by adding a variable for each course, period and distinct room capacity. In the second stage, they assign the courses to the rooms that the lectures should take place in, taking the **RStab** constraint into consideration. The solution from the first stage is used to fix the courses at the determined periods and the selected room capacities in the second stage.

Hao and Benlic (2011) propose a divide-and-conquer approach based on the first stage formulation of Lach and Lübbecke (2012), focusing on finding lower bounds. The method they present divides the MIP model into smaller parts by relaxing or removing some of the constraints such that they can decompose the model into a set of subproblems. They then obtain lower bounds for each subproblem, and the sum of all these lower bounds is then a lower bound of the original problem. The approach provides excellent results.

Cacchiani et al (2013) presented different formulations containing exponentially many variables. The approach that obtained the best computational results consists of two sets of the main binary decision variables. In each of the sets, a binary variable represents a schedule for an entire week. One of the sets of binary variables takes care of the soft constraints **RC** and **RStab** and the other set considers the soft constraints **MWD** and **IL**. The sum of the lower bounds of each of the set is a lower bound of the original problem. The method shows impressive results and obtains good lower bounds.

In the MIP-based methods found in the literature, it seems that decomposing the problem into smaller parts provides the best results. The downside by many of the methods is that they only provide lower bounds or that they come at the cost of sacrificing the guarantee of optimality, i.e., even if we solve the models to optimality, it does not guarantee that the solution is globally optimal. Based on these observations, the goal of this paper is to exploit the knowledge that decreasing the number of main decision variables provides better performance. Furthermore, we want to maintain the guarantee of optimality, and therefore we look for exact MIP formulations.

We base our work on the work (Bagger et al 2015) presented at the biennial Multidisciplinary International Scheduling Conference: Theory & Applications (MISTA) 2015 (<http://www.schedulingconference.org/>). There are some slight differences in this article compared to the extended abstract submitted to MISTA. In the abstract, we only discussed one flow formulation, and in this article we present two flow formulations. Furthermore, there were some missing details in the proof that we supplied in the appendix (Bagger et al 2015, Appendix A Proof of Proposition 2). Therefore the proposition and proof have been changed

to overcome this and the resulting mathematical model remains the same. Lastly the results have been updated with our newest test runs and with more data sets.

We organize the paper as follows: In Section 2 an intuitive MIP model is presented, based on the main binary variables having three indices. We then show how some of the integer requirements on the variables can be relaxed. In Section 3 we present the two formulations based on underlying network flow problems; we base the first on a *Minimum Cost Flow* problem and the second on a *Multi-Commodity Flow Problem*. We present and discuss the computational results in Section 4 and lastly some perspectives on the flow formulations are considered in Section 5.

Throughout the article, we assume that the reader is familiar with the maximum flow problem, the minimum cost flow problem, and the multi-commodity flow problem. We refer to Kleinberg and Tardos (2005) and Ahuja et al (1993) for details on these problems. Furthermore we define the following for notation purposes:

$$(x)^+ := \max\{x, 0\}$$

We use the notation  $\mathbb{Z}^+$  for the set of non-negative integers.

## 2 Three-Index Mixed Integer Programming Formulation

In this section, we present an intuitive MIP formulation of the CTT using three-index binary decision variables. We refer to this formulation throughout the article as the three-index formulation.

Let  $\mathcal{C}$  be the set of courses,  $\mathcal{P}$  be the set of periods and  $\mathcal{R}$  be the set of rooms. Furthermore, there are days  $\mathcal{D}$ , curricula  $\mathcal{Q}$ , lecturers  $\mathcal{L}$ , the periods  $\mathcal{P}_d \subseteq \mathcal{P}$  that belongs to day  $d \in \mathcal{D}$ , the courses  $\mathcal{C}_q \subseteq \mathcal{C}$  which are part of curriculum  $q \in \mathcal{Q}$  and the courses  $\mathcal{C}_l \subseteq \mathcal{C}$  which are all being taught by lecturer  $l \in \mathcal{L}$ . For each period  $p \in \mathcal{P}$  we define the set  $\Phi_p$ . The set  $\Phi_p$  contains the periods that are adjacent to  $p$ , i.e. the periods occurring on the same day which is in the period directly before or after the time slot of  $p$ .

Let  $L_c$  be the number of lectures to be scheduled for course  $c \in \mathcal{C}$ ,  $C_r$  be the capacity of room  $r \in \mathcal{R}$ ,  $S_c$  be the number of students attending course  $c \in \mathcal{C}$  and let  $F_{c,p}$  be one if it is allowed to schedule a lecture from course  $c \in \mathcal{C}$  in period  $p \in \mathcal{P}$  and zero otherwise. Lastly,  $M_c$  is the minimum number of days that it is preferred to schedule lectures for course  $c \in \mathcal{C}$  in.

Let  $x_{c,p,r}$  be a binary variable deciding whether to schedule a lecture from course  $c \in \mathcal{C}$  in period  $p \in \mathcal{P}$  and room  $r \in \mathcal{R}$  or not.  $t_{c,d}$  is a binary variable taking value one if course  $c \in \mathcal{C}$  has at least one lecture at day  $d \in \mathcal{D}$ , and zero otherwise.  $w_c$  is an integer variable denoting the number of days below the given minimum that course  $c \in \mathcal{C}$  has lectures.  $z_{c,r}$  is a binary variable taking value one if course  $c \in \mathcal{C}$  is scheduled in room  $r \in \mathcal{R}$  at least once during the week, and zero otherwise.  $s_{q,p}$  is a binary variable taking value one if curriculum  $q \in \mathcal{Q}$  has an isolated lecture in period  $p \in \mathcal{P}$ . Let  $W^{\mathbf{RC}}$ ,  $W^{\mathbf{IL}}$ ,  $W^{\mathbf{MWD}}$  and  $W^{\mathbf{RStab}}$  be the non-negative weights of the constraints **RC**, **IL**, **MWD** and **RStab** respectively. The three-index formulation is given in the following model:

$$\begin{aligned} \min W^{\mathbf{RC}} & \sum_{c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot x_{c,p,r} \\ & + W^{\mathbf{IL}} \sum_{q \in \mathcal{Q}, p \in \mathcal{P}} s_{q,p} + W^{\mathbf{MWD}} \sum_{c \in \mathcal{C}} w_c \\ & + W^{\mathbf{RStab}} \sum_{c \in \mathcal{C}} \left( \sum_{r \in \mathcal{R}} z_{c,r} - 1 \right) \end{aligned} \quad (1)$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}, r \in \mathcal{R}} x_{c,p,r} = L_c \quad \forall c \in \mathcal{C} \quad (2)$$

$$\sum_{r \in \mathcal{R}} x_{c,p,r} \leq F_{c,p} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (3)$$

$$\sum_{c \in \mathcal{C}} x_{c,p,r} \leq 1 \quad \forall r \in \mathcal{R}, p \in \mathcal{P} \quad (4)$$

$$\sum_{c \in \mathcal{C}_i, r \in \mathcal{R}} x_{c,p,r} \leq 1 \quad \forall l \in \mathcal{L}, p \in \mathcal{P} \quad (5)$$

$$\sum_{c \in \mathcal{C}_q, r \in \mathcal{R}} x_{c,p,r} \leq 1 \quad \forall q \in \mathcal{Q}, p \in \mathcal{P} \quad (6)$$

$$\sum_{p \in \mathcal{P}} x_{c,p,r} \leq L_c \cdot z_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (7)$$

$$\sum_{p \in \mathcal{P}} x_{c,p,r} \geq z_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (8)$$

$$\sum_{r \in \mathcal{R}} z_{c,r} \geq 1 \quad \forall c \in \mathcal{C} \quad (9)$$

$$\sum_{p \in \mathcal{P}_d, r \in \mathcal{R}} x_{c,p,r} \geq t_{c,d} \quad \forall c \in \mathcal{C}, d \in \mathcal{D} \quad (10)$$

$$\sum_{d \in \mathcal{D}} t_{c,d} + w_c \geq M_c \quad \forall c \in \mathcal{C} \quad (11)$$

$$\sum_{c \in \mathcal{C}_q, r \in \mathcal{R}} \left( x_{c,p,r} - \sum_{p' \in \Phi_p} x_{c,p',r} \right) \leq s_{q,p} \quad \forall q \in \mathcal{Q}, p \in \mathcal{P} \quad (12)$$

$$x_{c,p,r} \in \mathbb{B} \quad \forall c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R} \quad (13)$$

$$z_{c,r} \in \mathbb{B} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (14)$$

$$t_{c,d} \in \mathbb{B} \quad \forall c \in \mathcal{C}, d \in \mathcal{D} \quad (15)$$

$$w_c \in \mathbb{Z}^+ \quad \forall c \in \mathcal{C} \quad (16)$$

$$s_{q,p} \in \mathbb{B} \quad \forall q \in \mathcal{Q}, p \in \mathcal{P} \quad (17)$$

The constraints (2) ensure that all lectures are scheduled. The courses are only allowed to be scheduled in available periods, which is ensured by the constraints (3) that also ensure that all lectures are scheduled in different periods. Constraints (4) make sure that no more than one course is scheduled in a room in every period. Constraints (5) and (6) ensure that courses that share a lecturer or are in the same

curriculum are not scheduled in the same period. The constraints (7) ensure that lectures are only scheduled in rooms that have been *opened*, whereby *opened* we mean a room  $r \in \mathcal{R}$  where  $z_{c,r} = 1$  for course  $c \in \mathcal{C}$ . Constraints (8) and (9) ensure that at least one lecture is scheduled in the *open* rooms and that at least one room is put to use by each course. Constraints (10) puts an upper bound on the  $t_{c,d}$  variable for each course  $c \in \mathcal{C}$  and day  $d \in \mathcal{D}$  such that it can take a positive value only if at least one lecture for  $c$  is scheduled at day  $d$ . Constraints (11) calculate the violation of the **MWD** constraint and the constraints (12) calculate for which periods the curricula have isolated lectures.

The variables  $t_{c,d}$ ,  $w_c$  and  $s_{q,p}$  can be relaxed to continuous variables, so we replace the variable domains; (15), (16) and (17) by the following:

$$0 \leq t_{c,d} \leq 1 \quad \forall c \in \mathcal{C}, d \in \mathcal{D} \quad (18)$$

$$w_c \geq 0 \quad \forall c \in \mathcal{C} \quad (19)$$

$$0 \leq s_{q,p} \leq 1 \quad \forall q \in \mathcal{Q}, p \in \mathcal{P} \quad (20)$$

Burke et al (2012) note that it is never possible that the minimum working days constraint is violated by more than  $M_c - 1$  since every course must be scheduled in at least one day. Therefore this can be used as an upper bound for the variable  $w_c$  which can help the MIP solver. This bound can be strengthened as all lectures must be scheduled for each course  $c \in \mathcal{C}$ . Let  $\mathcal{T}$  be the maximum number of periods for any day, i.e.,  $\mathcal{T} = \max_{d \in \mathcal{D}} \{|\mathcal{P}_d|\}$ . Then, at most  $\mathcal{T}$  lectures can be scheduled at any day, and therefore the number of days that the course is scheduled in must be at least  $\left\lceil \frac{L_c}{|\mathcal{T}|} \right\rceil$ , so we can add the following bounds:

$$0 \leq w_{c,d} \leq M_c - \left\lceil \frac{L_c}{|\mathcal{T}|} \right\rceil \quad \forall c \in \mathcal{C} \quad (21)$$

The constraints (9) also comes from Burke et al (2012) where they note that since all lectures must be scheduled then surely every course must occupy at least one room.

If  $F_{c,p} = 0$  for some course  $c \in \mathcal{C}$  and period  $p \in \mathcal{P}$  then we do not add the variables  $\{x_{c,p,r}\}_{r \in \mathcal{R}}$  to the model. Therefore the constraints (3) are redundant since every course is taught by exactly one lecturer and constraints (5) ensure that each lecturer has at most one lecture scheduled in any period. Furthermore the constraints (5) and (6) are replaced by clique inequalities. We do this by creating a *course conflict graph* where each node corresponds to a course. An edge is connecting two courses if they are in the same curriculum or taught by the same lecturer. We then enumerate all the maximal cliques by running the Bron-Kerbosch algorithm (Bron and Kerbosch 1973). Let  $\Gamma$  be the set of cliques and let  $\mathcal{C}_\gamma$  be the set of courses in the clique  $\gamma \in \Gamma$ . We replace all the constraints (5) and (6) by adding the following constraints for each clique  $\gamma \in \Gamma$  and period  $p \in \mathcal{P}$ :

$$\sum_{c \in \mathcal{C}_\gamma, r \in \mathcal{R}} x_{c,p,r} \leq 1 \quad \forall \gamma \in \Gamma, p \in \mathcal{P} \quad (22)$$

### 3 Network Flow Formulations

Our inspiration for the new MIP formulations comes from the two-stage decomposition described by [Lach and Lübbecke \(2008, 2012\)](#). In their work, they solve the problem by splitting the MIP model into two distinct models and then solve these models in sequence. In this paper, we also consider the CTT as two independent models (which are both derived from the three-index model), but instead of solving the two models independently, we re-combine the models into one model using flow formulation techniques. Thereby we obtain new exact formulations with different properties than the original three-index formulation. In [Section 3.1](#) we present a formulation for the CTT based on minimum cost flow. In [Section 3.2](#) we present a formulation for the CTT based on multi-commodity flow.

#### 3.1 Minimum Cost Flow

In this section, we present a formulation based on the minimum cost flow problem. In [Section 3.1.1](#) and [Section 3.1.2](#) we present two different MIP models for handling different aspects of the problem; the course-to-period assignment and the course-to-room assignment, respectively. Both of these models are derived from model (1) – (17). In [Section 3.1.3](#) we present how these two models are combined into a single model using minimum flow techniques. The resulting model of this combination has fewer integer variables than the original formulation in model (1) – (17), and is exact concerning the original formulation of the CTT.

##### 3.1.1 Course-to-Period Assignment

In this section we consider the CTT problem from the period aspect only, i.e., we ignore the existence of rooms. The goal of the problem is to assign courses to periods using the same criteria as in the three-index formulation. We give the MIP formulation of the course period assignment subproblem in the following model:

$$\min W^{\text{IL}} \sum_{q \in \mathcal{Q}, p \in \mathcal{P}} s_{q,p} + W^{\text{MWD}} \sum_{c \in \mathcal{C}} w_c \quad (23)$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}} x_{c,p} = L_c \quad \forall c \in \mathcal{C} \quad (24)$$

$$x_{c,p} \leq F_{c,p} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (25)$$

$$\sum_{c \in \mathcal{C}_\gamma} x_{c,p} \leq 1 \quad \forall \gamma \in \Gamma, p \in \mathcal{P} \quad (26)$$

$$\sum_{p \in \mathcal{P}_d} x_{c,p} \geq t_{c,d} \quad \forall c \in \mathcal{C}, d \in \mathcal{D} \quad (27)$$

$$\sum_{d \in \mathcal{D}} t_{c,d} + w_c \geq M_c \quad \forall c \in \mathcal{C} \quad (28)$$

$$\sum_{c \in \mathcal{C}_q} \left( x_{c,p} - \sum_{p' \in \Phi_p} x_{c,p'} \right) \leq s_{q,p} \quad \forall q \in \mathcal{Q}, p \in \mathcal{P} \quad (29)$$

$$x_{c,p} \in \mathbb{B} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (30)$$

$$0 \leq t_{c,d} \leq 1 \quad \forall c \in \mathcal{C}, d \in \mathcal{D} \quad (31)$$

$$0 \leq w_c \leq M_c - \left\lceil \frac{L_c}{|\mathcal{T}|} \right\rceil \quad \forall c \in \mathcal{C} \quad (32)$$

$$0 \leq s_{q,p} \leq 1 \quad \forall q \in \mathcal{Q}, p \in \mathcal{P} \quad (33)$$

Note that model (23) – (33) is equivalent to model (1) – (17), except that the rooms are ignored. Therefore we do not describe (23) – (33) in details.

### 3.1.2 Course-to-Room Assignment

In this section we consider the room assignment sub-problem, ignoring the existence of periods. The goal is to assign courses to rooms and the criteria given in the three-index model. We give the formulation for the room assignment in the following:

$$\min \quad W^{\text{RStab}} \sum_{c \in \mathcal{C}} \left( \sum_{r \in \mathcal{R}} z_{c,r} - 1 \right) \quad (34)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}} z_{c,r} \geq 1 \quad \forall c \in \mathcal{C} \quad (35)$$

$$z_{c,r} \in \mathbb{B} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (36)$$

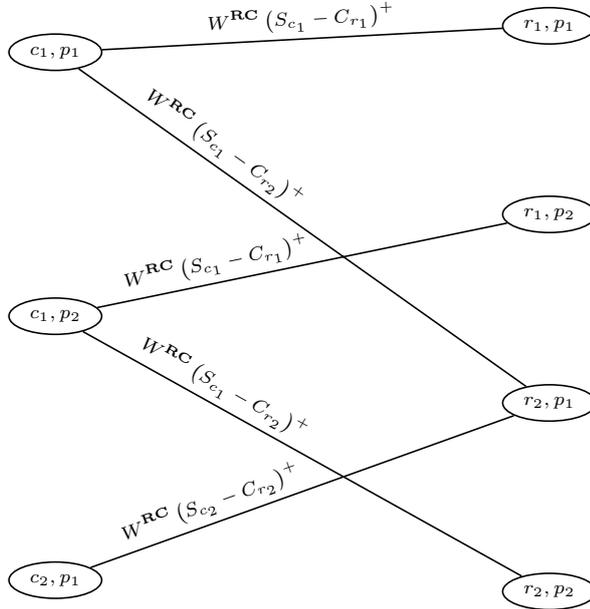
Note that in model (34) – (36) only the room stability objective is considered and not the room capacity. The reason is due to that the binary variable  $z_{c,r}$  only identifies whether or not room  $r \in \mathcal{R}$  is used by course  $c \in \mathcal{C}$  and not how many times the course is occupying the room. The room capacity will be taken care of in the later step.

### 3.1.3 Connecting the Models using the Minimum Cost Flow Problem

In this section we show how the minimum cost flow problem connects model (23) – (33) and model (34) – (36). Thereby we aim at obtaining a new exact formulation for the CTT with fewer integer variables.

If a solution  $\bar{x}$  to model (23) – (33) and a solution  $\bar{z}$  to model (34) – (36) is given then a new problem emerges; is the combined solution feasible, i.e., is there a feasible mapping from the assigned periods in  $\bar{x}$  to the assigned rooms in  $\bar{z}$  such that no room is occupied by two courses in the same period, and if so, what is the minimum room capacity penalty to any feasible mapping? A way to check this is to make a bipartite graph and solve a minimum cost maximum matching problem. For every course  $c \in \mathcal{C}$  create a node on the left-hand side of the bipartite graph for each period  $p \in \mathcal{P}$  that the course has been assigned to, i.e., if  $\bar{x}_{c,p} = 1$ . For every combination of a room and a period  $(r,p)$  create a node on the right-hand side of the graph. For every pair of course period nodes  $(c,p)$  and every pair of room-period node  $(r,p)$  put an edge between the nodes  $(c,p)$  and  $(r,p)$  if the course has been assigned to that room, i.e., if  $\bar{z}_{c,r} = 1$ , and set the weight of the edge to  $W^{\text{RC}}(S_c - C_r)^+$ . As an example of the matching problem, consider two courses,  $c_1$  and  $c_2$ , two periods,  $p_1$  and  $p_2$  and two rooms,  $r_1$  and  $r_2$ . Let course  $c_1$  be teaching

two lectures assigned in periods  $p_1$  and  $p_2$  and let course  $c_2$  be teaching one lecture assigned in period  $p_1$ . Furthermore let  $c_1$  be assigned to rooms  $r_1$  and  $r_2$  and let  $c_2$  be assigned to room  $r_2$ . The corresponding bipartite graph is illustrated in Fig. 1.



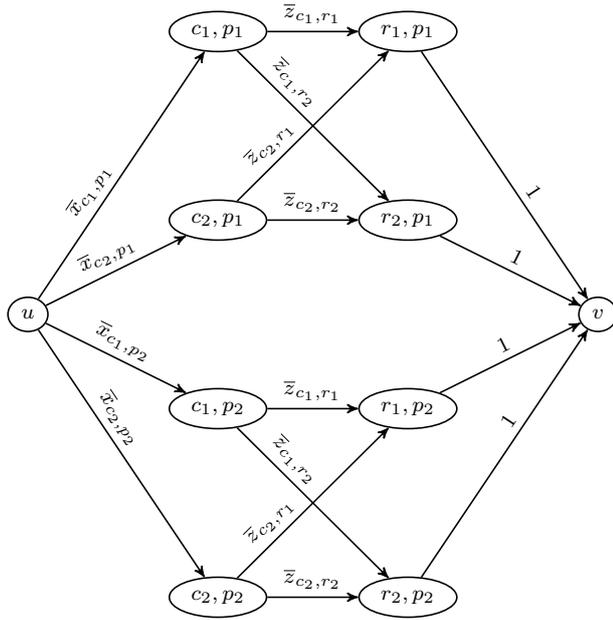
**Fig. 1** Example of the bipartite graph of an instance with courses,  $c_1$  and  $c_2$ , periods,  $p_1$  and  $p_2$  and rooms  $r_1$  and  $r_2$ . Course  $c_1$  has been assigned to periods  $p_1$  and  $p_2$  and rooms,  $r_1$  and  $r_2$ . Course  $c_2$  has been assigned to period  $p_1$  and room  $r_2$ . The labels in the nodes indicate the corresponding pair that the node belongs to. The labels above the edges are the corresponding weights.

If the solution  $(\bar{x}, \bar{z})$  is feasible, then a maximum matching must match all the left-hand side nodes into a node on the right-hand side, i.e., the value of the maximum matching must be  $\sum_c L_c$  and the cost of the matching corresponds to the total room capacity violation.

A way to solve the minimum cost maximum matching problem is to solve a flow problem on the graph with a source node ( $u$ ) which is connected to all the left-hand side nodes and a sink node ( $v$ ) which is connected to all the right-hand side nodes as described by [Kleinberg and Tardos \(2005\)](#) in Chapter 7 and [Ahuja et al \(1993\)](#) in Section 12.3 and 12.4. The weights on the new edges are all zero and the capacity of all edges are 1. Another way is to create a new graph representation  $\mathcal{G}_{\text{mcf}}$  and solve a flow problem on that graph. For every possible course-period combination create a node  $(c, p)$  and for every possible room-period combination create a node  $(r, p)$ . For every course  $c \in C$ , period  $p \in P$  and room  $r \in R$  there is an arc from the node  $(c, p)$  to the node  $(r, p)$ . Furthermore there is a source node ( $u$ ) with an arc to every node  $(c, p)$  and a sink node ( $v$ ) with an arc from every node  $(r, p)$ . One unit of flow in this graph corresponds to a course assignment, i.e., sending one unit of flow from ( $u$ ) to  $(c, p)$  for some  $c \in C$  and  $p \in P$  corresponds to assigning course  $c$  to period  $p$  and sending a unit of flow

through the arc  $(c, p) \rightarrow (r, p)$  for some  $c \in C$ ,  $p \in P$  and  $r \in R$  corresponds to assigning course  $c$  to room  $r$  in period  $p$ . The capacities are set for a given solution pair of model (23) – (33) and model (34) – (36).

For every course  $c \in C$  and period  $p \in P$  the capacity of the arc  $(u) \rightarrow (c, p)$  is  $\bar{x}_{c,p}$ . This is to depict that we can only send flow from the source  $(u)$  to a node  $(c, p)$  if the course  $c \in C$  is assigned to period  $p \in P$ . For every course  $c \in C$ , period  $p \in P$  and room  $r \in R$  the capacity on the arc  $(c, p) \rightarrow (r, p)$  is  $\bar{z}_{c,r}$  to denote that we can only send flow from a course-period pair to a room-period pair if the course is assigned to the corresponding room. The capacity of the arcs  $(r, p) \rightarrow (v)$  is set to one as a room can at most be occupied by one course in any period. An example of the graph is given in Fig. 2.



**Fig. 2** Illustration of the minimum cost flow graph of an instance with two courses, two rooms and two period. The labels above the arcs are the corresponding capacities. The weight of the arc from the node  $(c, p)$  to the node  $(r, p)$  is  $W^{\mathbf{RC}}(S_c - C_r)$ .

Note that the solutions  $\bar{x}$  and  $\bar{z}$  are feasible if and only if there exists an integral solution to the maximum flow problem in the graph with total amount  $\sum_{c \in C} L_c$ . Since we know the amount of flow that any feasible solution must contain we can solve the problem as a minimum cost flow problem where the supply of node  $(u)$  and the demand of node  $(v)$  is  $\sum_{c \in C} L_c$  and all other nodes have a demand and supply of zero. For every course  $c \in C$ , period  $p \in P$  and room  $r \in R$  let the cost of the flow on the arc  $(c, p) \rightarrow (r, p)$  be the cost of violating the **RC** constraint, i.e.,  $W^{\mathbf{RC}}(S_c - C_r)^+$ , and let the cost be zero on all other arcs.

Let the following non-negative variables be defined:

- $f_{c,p}^u$ : The amount of flow on the arc  $(u) \rightarrow (c, p)$
- $f_{c,p,r}$ : The amount of flow on the arc  $(c, p) \rightarrow (r, p)$

$f_{r,p}^v$ : The amount of flow on the arc  $(r, p) \rightarrow (v)$

The minimum cost flow formulation is given in as follows:

$$\min W^{\mathbf{RC}} \sum_{c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot f_{c,p,r} \quad (37)$$

$$\text{s.t.} \quad - \sum_{c \in \mathcal{C}, p \in \mathcal{P}} f_{c,p}^u = - \sum_{c \in \mathcal{C}} L_c \quad (38)$$

$$f_{c,p}^u - \sum_{r \in \mathcal{R}} f_{c,p,r} = 0 \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (39)$$

$$\sum_{c \in \mathcal{C}} f_{c,p,r} - f_{r,p}^v = 0 \quad \forall r \in \mathcal{R}, p \in \mathcal{P} \quad (40)$$

$$\sum_{r \in \mathcal{R}, p \in \mathcal{P}} f_{r,p}^v = \sum_{c \in \mathcal{C}} L_c \quad (41)$$

$$0 \leq f_{c,p}^u \leq \bar{x}_{c,p} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (42)$$

$$0 \leq f_{c,p,r} \leq \bar{z}_{c,r} \quad \forall c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R} \quad (43)$$

$$0 \leq f_{r,p}^v \leq 1 \quad \forall r \in \mathcal{R}, p \in \mathcal{P} \quad (44)$$

$$f_{c,p}^u \in \mathbb{Z} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (45)$$

$$f_{c,p,r} \in \mathbb{Z} \quad \forall c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R} \quad (46)$$

$$f_{r,p}^v \in \mathbb{Z} \quad \forall r \in \mathcal{R}, p \in \mathcal{P} \quad (47)$$

The constraints (38), (39), (40) and (41) correspond to the balance constraints of the nodes, i.e., ingoing flow minus outgoing flow must equal the demand in the node which is  $-\sum_c L_c$  in the source node,  $\sum_c L_c$  in the sink node and zero in all the other nodes. Constraints (42), (43) and (44) are the capacity constraints of the arcs.

Model (37) – (47) contains three-index integer variables just as in model (1) – (17) which is what we wanted to get rid of to begin with. However, the minimum cost flow integrality theorem states that if all arc capacities, supplies, and demands are integers and there exists a feasible flow, then there exists a minimum cost flow to the LP-relaxation with integer values (Ahuja et al 1993, Theorem 9.10). So we can disregard the integrality requirements on the flow variables since  $x$  and  $z$  are binary variables.

It should be noted that due to the structure of the graph and the equilibrium constraints (39) and (40) the flow requirement constraint (41) to the sink is redundant and can be removed from the model. Furthermore since  $\sum_{p \in \mathcal{P}} \bar{x}_{c,p} = L_c$  for every course  $c \in \mathcal{C}$  in any feasible solution then the flow requirement is fulfilled if and only if equality is met in constraints (42). This means that we can remove constraints (42) and (38) and replace  $f_{c,p}^u$  with  $\bar{x}_{c,p}$  throughout the model. Lastly since every variable  $f_{r,p}^v$  occur exactly once in the constraints (40) then we can replace the variable with  $\sum_{c \in \mathcal{C}} f_{c,p,r}$  in the model. Furthermore we can aggregate all the upper bounds (43) into one constraint:

$$\sum_{p \in \mathcal{P}} f_{c,p,r} \leq |P| \cdot z_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (48)$$

Since we know that for any course  $c \in C$  and room  $r \in R$  the sum  $\sum_{p \in \mathcal{P}} f_{c,p,r}$  can never exceed  $L_c$  then we can strengthen this constraint by replacing the coefficient of  $z_{c,r}$  with  $L_c$ . Then the models (23) – (33) and (34) – (36) can be combined into the new formulation given in the following:

$$\begin{aligned}
\min \quad & W^{\mathbf{RC}} \sum_{c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot f_{c,p,r} \\
& + W^{\mathbf{IL}} \sum_{q \in \mathcal{Q}, p \in \mathcal{P}} s_{q,p} + W^{\mathbf{MWD}} \sum_{c \in \mathcal{C}} w_c \\
& + W^{\mathbf{RStab}} \sum_{c \in \mathcal{C}} \left( \sum_{r \in \mathcal{R}} z_{c,r} - 1 \right) \tag{49} \\
\text{s.t.} \quad & (24) - (33) \tag{50} \\
& (35) - (36) \tag{51} \\
& \sum_{r \in \mathcal{R}} f_{c,p,r} = x_{c,p} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \tag{52} \\
& \sum_{p \in \mathcal{P}} f_{c,p,r} \leq L_c \cdot z_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \tag{53} \\
& \sum_{c \in \mathcal{C}} f_{c,p,r} \leq 1 \quad \forall r \in \mathcal{R}, p \in \mathcal{P} \tag{54} \\
& f_{c,p,r} \geq 0 \quad \forall c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R} \tag{55}
\end{aligned}$$

Even though the minimum cost flow problem has the integrality property, this does not imply that the  $f_{c,p,r}$  variables will take integer values when solving model (49) – (55). So, when the model is solved, we check whether the variables are fractional. If they are, we solve the minimum cost flow problem for the  $x_{c,p}$  and  $z_{c,r}$  variables by some polynomial algorithm returning an integer solution, e.g., the Cycle-Canceling Algorithm (Ahuja et al 1993, proof of Theorem 9.10, section 9.6).

### 3.2 Multi-Commodity Flow

In this section, we consider the room assignment part of the problem once again. However, this time we not only formulate a model to decide which rooms the courses should be scheduled in but also how many times during the week the courses should be scheduled in each room. We show how a multi-commodity flow problem can be used to connect this formulation to the time scheduling formulation in model (23) – (33).

We introduce the integer variable  $y_{c,r}$  to identify the number of times that course  $c \in C$  is assigned to room  $r \in R$ . The formulation is given as follows:

$$\begin{aligned}
\min \quad & W^{\mathbf{RC}} \cdot \sum_{c \in \mathcal{C}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot y_{c,r} \\
& + W^{\mathbf{RStab}} \cdot \sum_{c \in \mathcal{C}} \left( \sum_{r \in \mathcal{R}} z_{c,r} - 1 \right) \tag{56}
\end{aligned}$$

$$\text{s.t. } \sum_{r \in \mathcal{R}} y_{c,r} = L_c \quad \forall c \in \mathcal{C} \quad (57)$$

$$y_{c,r} \leq L_c \cdot z_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (58)$$

$$y_{c,r} \geq z_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (59)$$

$$\sum_{r \in \mathcal{R}} z_{c,r} \geq 1 \quad \forall c \in \mathcal{C} \quad (60)$$

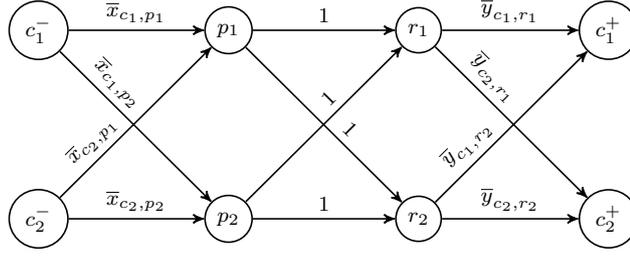
$$y_{c,r} \in \mathbb{Z}^+ \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (61)$$

$$z_{c,r} \in \mathbb{B} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (62)$$

Constraints (57) ensure that the total number of times that a course  $c \in \mathcal{C}$  is occupying some rooms is equal to the number of lectures to be taught. Constraints (58) ensure that for some course  $c \in \mathcal{C}$  and some room  $r \in \mathcal{R}$ ,  $z_{c,r}$  is set to one if  $y_{c,r} > 0$ . Constraints (59) and (60) ensure that at least one room is selected for each course and that at least one lecture is put into each selected room.

Given a solution  $\bar{x}$  to model (23) – (33) and a solution  $(\bar{y}, \bar{z})$  to model (56) – (62) then a new problem emerges; is the combined solution  $(\bar{x}, \bar{y}, \bar{z})$  feasible, i.e., is there a feasible mapping from the assigned rooms in  $\bar{y}$  to the assigned periods in  $\bar{x}$  such that no room is occupied by two courses in the same period and no course is giving two lectures in the same period? To check this, we formulate the problem as a *multi-commodity flow problem*. For each course  $c \in \mathcal{C}$  we have a commodity where we need to send flow from a source node ( $c^-$ ) to a sink node ( $c^+$ ). The demand for the commodity is  $L_c$ , i.e., the amount of flow that needs to be sent from the source to the sink is the number of lectures for the course. For each period  $p \in \mathcal{P}$  we have a node and for each room  $r \in \mathcal{R}$  we also have a node. For each course  $c \in \mathcal{C}$  we have an outgoing arc from node ( $c^-$ ) to each node ( $p$ ) with a capacity of  $\bar{x}_{c,p}$  for the corresponding period  $p \in \mathcal{P}$ . The capacity ensures that the amount of flow send out of the node ( $c^-$ ) to the node ( $p$ ) does not exceed  $\bar{x}_{c,p}$ , i.e., a course can only send flow to (be scheduled in) the periods that it has been assigned to. Since each commodity has their distinct source node, this ensures that the amount of the commodity corresponding to course  $c$  is only send to periods where the course is assigned. Furthermore we have an ingoing arc from each node ( $r$ ) to node ( $c^+$ ) with a capacity of  $\bar{y}_{c,r}$  for the corresponding room  $r \in \mathcal{R}$ . The capacity ensures that the total amount of flow through rooms does not exceed the number of times the course has been scheduled in the rooms. Lastly for each period and room pair  $(p, r) \in \mathcal{P} \times \mathcal{R}$  there is an arc from node ( $p$ ) to node ( $r$ ) with a capacity of one to ensure that the total amount of flow across all commodities does not exceed one. This capacity corresponds to the constraint that any room cannot be occupied by more than one lecture in any period. An example of the graph for a test instance containing two courses, two rooms and two periods is given in Fig. 3.

We can either state the multi-commodity flow problem as an arc formulation or a path formulation. We use the path formulation. In the graph it can be seen that there is a path  $(c_1^-) \rightarrow (p) \rightarrow (r) \rightarrow (c_2^+)$  for every course  $c_1 \in \mathcal{C}$ , every period  $p \in \mathcal{P}$ , every room  $r \in \mathcal{R}$  and every course  $c_2 \in \mathcal{C}$ . These are all the paths in the graph and each commodity corresponding to a course  $c \in \mathcal{C}$  must select  $L_c$  paths to send flow. The path  $(c_1^-) \rightarrow (p) \rightarrow (r) \rightarrow (c_2^+)$ , for some course  $c_1 \in \mathcal{C}$ , some period  $p \in \mathcal{P}$ , some room  $r \in \mathcal{R}$  and some course  $c_2 \in \mathcal{C}$ , can only be selected by a commodity corresponding to course  $c \in \mathcal{C}$  if  $c = c_1 = c_2$ . Let the integer variable



**Fig. 3** Illustration of the multi-commodity flow graph of an instance with two courses, two rooms and two periods. The labels above the arcs are the corresponding capacities.

$f_{c,p,r}$  correspond to the amount of flow of the commodity corresponding to course  $c \in \mathcal{C}$  that is sent on the path  $(c^-) \rightarrow (p) \rightarrow (r) \rightarrow (c^+)$  for period  $p \in \mathcal{P}$  and room  $r \in \mathcal{R}$ . Then the mathematical formulation can be described as follows:

$$\sum_{r \in \mathcal{R}} f_{c,p,r} \leq \bar{x}_{c,p} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (63)$$

$$\sum_{c \in \mathcal{C}} f_{c,p,r} \leq 1 \quad \forall p \in \mathcal{P}, r \in \mathcal{R} \quad (64)$$

$$\sum_{p \in \mathcal{P}} f_{c,p,r} \leq \bar{y}_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (65)$$

$$\sum_{p \in \mathcal{P}, r \in \mathcal{R}} f_{c,p,r} = L_c \quad \forall c \in \mathcal{C} \quad (66)$$

$$f_{c,p,r} \in \mathbb{Z}^+ \quad \forall c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R} \quad (67)$$

Constraints (63), (64) and (65) ensure that the capacities of the arcs are not violated. Constraint (66) ensures that the demand is fulfilled for each commodity and (67) ensures that the flow is integral.

The multi-commodity flow problem is known to be NP-hard, however, in the next Proposition 1 we show that it suffices to solve the LP-relaxation of the path formulation in model (63) – (67).

**Proposition 1** *Let  $A$  be the set of feasible period-room solutions and let  $o(\bar{x}, \bar{y}, \bar{z})$  denote the objective value of the (possibly infeasible) period-room solution  $(\bar{x}, \bar{y}, \bar{z})$ . Consider the solution space of the LP-relaxation  $\mathcal{F}_{LP}$  of model (63) – (67) for  $(\bar{x}, \bar{y}, \bar{z})$ . We then have the following:*

- (a)  $\mathcal{F}_{LP} = \emptyset \implies (\bar{x}, \bar{y}, \bar{z}) \notin A$
- (b)  $\mathcal{F}_{LP} \neq \emptyset \implies \exists y' : (\bar{x}, y', \bar{z}) \in A \wedge$   
 $o(\bar{x}, y', \bar{z}) \leq o(\bar{x}, \bar{y}, \bar{z})$

*Proof (Proposition 1a)* We make the proof by showing that the contrapositive statement holds:

$$(\bar{x}, \bar{y}, \bar{z}) \in A \implies \mathcal{F}_{LP} \neq \emptyset$$

Assume that  $(\bar{x}, \bar{y}, \bar{z}) \in A$  and consider some feasible assignment for this solution. Let the variable  $\bar{f}_{c,p,r}$  take value one if course  $c \in \mathcal{C}$  is assigned to period  $p \in \mathcal{P}$  and room  $r \in \mathcal{R}$  in the considered assignment. Since we are considering a feasible assignment and it is based on the solution  $(\bar{x}, \bar{y}, \bar{z})$  then the following conditions must be met:

$$\sum_{r \in \mathcal{R}} \bar{f}_{c,p,r} = \bar{x}_{c,p} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (68)$$

$$\sum_{c \in \mathcal{C}} \bar{f}_{c,p,r} \leq 1 \quad \forall p \in \mathcal{P}, r \in \mathcal{R} \quad (69)$$

$$\sum_{p \in \mathcal{P}} \bar{f}_{c,p,r} = \bar{y}_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (70)$$

$$\bar{f}_{c,p,r} \in \mathbb{B} \quad \forall c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R} \quad (71)$$

So the variables  $\bar{f}$  must fulfill the constraints (63)–(65) and (67). Furthermore since

$\sum_{p \in \mathcal{P}} \bar{x}_{c,p} = \sum_{r \in \mathcal{R}} \bar{y}_{c,r} = L_c$  for each course  $c \in \mathcal{C}$  then  $\sum_{p \in \mathcal{P}, r \in \mathcal{R}} \bar{f}_{c,p,r} = L_c$  must hold and thus constraint (66) is also fulfilled. This means that  $\bar{f}$  must be a feasible solution to model (63) – (67) and so  $\mathcal{F}_{LP} \neq \emptyset$ .  $\square$

*Proof (Proposition 1b)* Consider a solution  $(\bar{x}, \bar{y}, \bar{z})$  with objective value  $o(\bar{x}, \bar{y}, \bar{z})$ . Assume that  $\mathcal{F}_{LP} \neq \emptyset$ . Then there must exist some (possibly fractional) solution  $\bar{f}$  that fulfills the constraints in model (63) – (67).

Since  $\sum_{p \in \mathcal{P}} \bar{x}_{c,p} = \sum_{r \in \mathcal{R}} \bar{y}_{c,r} = L_c$  for every course  $c \in \mathcal{C}$  then the only way to fulfill constraints (66) is to fulfill the constraints (63) and (65) by equality, i.e., the solution  $\bar{f}$  must fulfill the following conditions:

$$\sum_{r \in \mathcal{R}} \bar{f}_{c,p,r} = \bar{x}_{c,p} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (72)$$

$$\sum_{c \in \mathcal{C}} \bar{f}_{c,p,r} \leq 1 \quad \forall p \in \mathcal{P}, r \in \mathcal{R} \quad (73)$$

$$\sum_{p \in \mathcal{P}} \bar{f}_{c,p,r} = \bar{y}_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (74)$$

$$\bar{f}_{c,p,r} \geq 0 \quad \forall c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R} \quad (75)$$

Now we create a new solution  $(\bar{x}, y', \bar{z})$  where the values  $\bar{x}$  and  $\bar{z}$  are fixed and only the values of the  $y$ -variables might be changed while ensuring that the objective value does not increase,

i.e.,  $o(\bar{x}, y', \bar{z}) \leq o(\bar{x}, \bar{y}, \bar{z})$ . Note that since  $\bar{x}$  and  $\bar{z}$  are kept fixed the difference in the objective value of the two solutions  $(\bar{x}, \bar{y}, \bar{z})$  and  $(\bar{x}, y', \bar{z})$  must be on the **RC** constraint.

Two cases can occur; either  $\bar{f}$  is integral, or some of the values are fractional. If  $\bar{f}$  is integral then every value must be either zero or one due to constraint (64) and therefore  $(\bar{x}, \bar{y}, \bar{z}) \in A$  and we simply set  $y' = \bar{y}$ .

Consider now the case where  $\bar{f}$  contains fractional values. Since  $\bar{y}_{c,r} \leq L_c \cdot \bar{z}_{c,r}$  for each course  $c \in \mathcal{C}$  and each room  $r \in \mathcal{R}$  then we must have that  $\bar{f}_{c,p,r} \leq \bar{z}_{c,r}$  for every period  $p \in \mathcal{P}$ . Consider now the minimum cost flow problem as described

in Section 3.1. A flow  $\bar{f}^{\text{MCF}}$  in the minimum cost flow graph is created in the following way; for each course  $c \in \mathcal{C}$ , period  $p \in \mathcal{P}$  and room  $r \in \mathcal{R}$  send flow on the path  $(u) \rightarrow (c, p) \rightarrow (r, p) \rightarrow (v)$  equal to  $\bar{f}_{c,p,r}$ . All the node balancing constraints must be satisfied as we are considering paths from the source to the sink. Furthermore the amount of flow must be  $\sum_{c \in \mathcal{C}} L_c$  since  $\sum_{p \in \mathcal{P}} \bar{x}_{c,p} = L_c$  for each course  $c \in \mathcal{C}$  and the condition (72) holds. Due to the condition (72) the capacity of the arc  $(u) \rightarrow (c, p)$  in the minimum cost flow graph illustrated in Fig. 2 cannot be violated as the total amount of flow on the arc is equal to  $\sum_{r \in \mathcal{R}} \bar{f}_{c,p,r}$  and the capacity of the arc is  $\bar{x}_{c,p}$ . For each course  $c \in \mathcal{C}$ , period  $p \in \mathcal{P}$  and room  $r \in \mathcal{R}$  the capacity of the arc  $(c, p) \rightarrow (r, p)$ , which is  $\bar{z}_{c,r}$ , cannot be violated since the flow send through that arc is equal to  $\bar{f}_{c,p,r}$  and we just argued that  $\bar{f}_{c,p,r} \leq \bar{z}_{c,r}$ . Lastly since  $\sum_{c \in \mathcal{C}} \bar{f}_{c,p,r} \leq 1$  then the capacity of the arc  $(r, p) \rightarrow (v)$  in the graph from Fig. 2 for room  $r \in \mathcal{R}$  and period  $p \in \mathcal{P}$  cannot be violated. So the flow  $\bar{f}^{\text{MCF}}$  is feasible for the minimum cost flow graph. Recall that the costs on the arcs in the minimum cost flow graph is zero on the arc  $(u) \rightarrow (c, p)$  for each course  $c \in \mathcal{C}$  and period  $p \in \mathcal{P}$ , zero on the arc  $(r, p) \rightarrow (v)$  for each room  $r \in \mathcal{R}$  and period  $p \in \mathcal{P}$  and  $W^{\text{RC}} \cdot (S_c - C_r)^+$  on the arc  $(c, p) \rightarrow (r, p)$  for each course  $c \in \mathcal{C}$ , period  $p \in \mathcal{P}$  and room  $r \in \mathcal{R}$ . This means that the total cost of the flow  $\bar{f}^{\text{MCF}}$  is:

$$W^{\text{RC}} \sum_{c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot \bar{f}_{c,p,r}$$

Due to the integrality property of the minimum cost flow there must exists a flow  $f'$  where the total amount of flow is the same as for the flow  $\bar{f}^{\text{MCF}}$ , the flow on all the arcs are integers and the cost of the flow  $f'$  must be less than or equal to the cost of the flow  $\bar{f}^{\text{MCF}}$ :

$$W^{\text{RC}} \sum_{c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot f'_{c,p,r} \leq W^{\text{RC}} \sum_{c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot \bar{f}_{c,p,r} \quad (76)$$

Create the new solution  $(\bar{x}, y', \bar{z})$  by setting  $y' = \sum_{p \in \mathcal{P}} f'_{c,p,r}$ . Now the difference in the objective value between the solution  $(\bar{x}, y', \bar{z})$  and the solution  $(\bar{x}, \bar{y}, \bar{z})$  can be calculated:

$$\begin{aligned} o(\bar{x}, y', \bar{z}) - o(\bar{x}, \bar{y}, \bar{z}) &= W^{\text{RC}} \sum_{c \in \mathcal{C}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot y'_{c,r} \\ &\quad - W^{\text{RC}} \sum_{c \in \mathcal{C}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot \bar{y}_{c,r} \\ &= W^{\text{RC}} \sum_{c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot f'_{c,p,r} \\ &\quad - W^{\text{RC}} \sum_{c \in \mathcal{C}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot \bar{y}_{c,r} \end{aligned}$$

Using (76) we can put an upper bound on the difference in the objective values:

$$o(\bar{x}, y', \bar{z}) - o(\bar{x}, \bar{y}, \bar{z}) \leq W^{\text{RC}} \sum_{c \in \mathcal{C}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot \left( \sum_{p \in \mathcal{P}} \bar{f}_{c,p,r} - \bar{y}_{c,r} \right)$$

Due to condition (74) it must therefore hold that  $o(\bar{x}, y', \bar{z}) \leq o(\bar{x}, \bar{y}, \bar{z})$ .  $\square$

Proposition 1 shows that we only need the linear relaxation of the multi-commodity flow problem and thus do not need to require integrality. Any solution to model (63) – (67) can only fulfill constraint (66) if equality is met in constraints (65). Then model (23) – (33) and model (56) – (62) can be combined into the following model:

$$\begin{aligned} \min \quad & W^{\mathbf{RC}} \sum_{c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R}} (S_c - C_r)^+ \cdot f_{c,p,r} \\ & + W^{\mathbf{IL}} \sum_{q \in \mathcal{Q}, p \in \mathcal{P}} s_{q,p} + W^{\mathbf{MWD}} \sum_{c \in \mathcal{C}} w_c \\ & + W^{\mathbf{RStab}} \sum_{c \in \mathcal{C}} \left( \sum_{r \in \mathcal{R}} z_{c,r} - 1 \right) \end{aligned} \quad (77)$$

$$\text{s.t.} \quad (24) \text{ — } (33) \quad (78)$$

$$(57) \text{ — } (62) \quad (79)$$

$$\sum_{p \in \mathcal{P}} f_{c,p,r} = y_{c,r} \quad \forall c \in \mathcal{C}, r \in \mathcal{R} \quad (80)$$

$$\sum_{r \in \mathcal{R}} f_{c,p,r} \leq x_{c,p} \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (81)$$

$$\sum_{c \in \mathcal{C}} f_{c,p,r} \leq 1 \quad \forall r \in \mathcal{R}, p \in \mathcal{P} \quad (82)$$

$$f_{c,p,r} \geq 0 \quad \forall c \in \mathcal{C}, p \in \mathcal{P}, r \in \mathcal{R} \quad (83)$$

It is not guaranteed that the  $f_{c,p,r}$  variables are integers in the solution obtained from model (77) – (83). If the solution returned by the model contains fractional values for the  $f_{c,p,r}$  variables we solve a minimum cost flow problem in the same way as in Section 3.1 using the values of the variables  $x_{c,p}$  and  $z_{c,p}$ . The minimum cost flow gives us a feasible integer solution with an objective value which is less than or equal to the objective value of the solution returned by model (77) – (83) as we showed in the proof of Proposition 1.

## 4 Computational Results

In this section, we perform computational experiments to evaluate the performance of the two new formulations for the CTT. We test the models on three data sets; TEST, COMP and DDS. The set TEST contains four data instances and were proposed by Di Gaspero and Schaefer (2003) (test1 – test4). The set COMP contains 21 data instances from the ITC2007 competition track 3 described in Di Gaspero et al (2007) (comp01 – comp21) mainly taken from the University of Udine. The set DDS contains seven data instances mainly taken from other Italian universities (DDS1 – DDS7). All the data sets can be retrieved from <http://tabu.diegm.uniud.it/ctt/index.php>. A benchmarking tool was provided as part of the ITC2007 competition, which calculates the amount of time that the algorithms were allowed to run in the competition on the machine the tool is executed on. This amount of time is usually referred to as one CPU unit. The

tool can be obtained from <http://www.cs.qub.ac.uk/itc2007>. We ran the tests in Windows 10 on a 3.40GHz Intel<sup>®</sup> Core<sup>™</sup> i5-3570K CPU with 8GB memory. Running the benchmarking tool returned 208 seconds as one CPU unit. The MIP solver used is Gurobi 6.0.2 provided by [Gurobi Optimization, Inc. \(2016\)](#). We have run all tests with the default parameters except that we have set the presolver to the most aggressive level (Presolve=2), in the hope that the presolver can decrease the size of the problem, and we set the number of threads to one (Threads=1).

As mentioned in sections 3.1.3 and 3.2, it may be needed to run some minimum cost flow algorithm on the solutions returned by model (49) – (55) or model (77) – (83). However, in all our tests the final solutions did not contain any fractional variables, so the minimum cost flow algorithm was never put to use. We also mentioned in Section 2 that an algorithm for enumerating all clique inequalities was run. This algorithm takes less than a second even for the largest data instances we have tested, so we have neglected these enumerations from the time limits when solving the models.

In Table 1, we compare the size of the MIP models for the three formulations; the three-index formulation, the minimum cost flow based formulation, and the multi-commodity flow based formulation.

Table 1: Comparison of the MIP statistics for the three models; 3IDX (the three-index formulation), MIN (the minimum cost flow based formulation), and MULT (the multi-commodity flow based formulation). For each data instance and model we report the number of continuous variables (Cont.), the number of integer variables (Int.), the number of constraints (Rows.), and the number of non-zero entries in the constraint matrix (Non-Zeros). At the bottom of the table (Avg.), we report the average value of dividing the numbers from the 3IDX formulation with the numbers from MIN or MULT.

	Model	Cont.	Int.	Rows	Non-Zeros
comp01	3IDX	$5.8 \times 10^2$	$5.3 \times 10^3$	$1.7 \times 10^3$	$5.3 \times 10^4$
	MIN	$5.7 \times 10^3$	$1.0 \times 10^3$	$2.3 \times 10^3$	$2.3 \times 10^4$
	MULT	$5.7 \times 10^3$	$1.2 \times 10^3$	$2.9 \times 10^3$	$2.9 \times 10^4$
comp02	3IDX	$2.1 \times 10^3$	$2.6 \times 10^4$	$7.0 \times 10^3$	$4.2 \times 10^5$
	MIN	$2.7 \times 10^4$	$2.8 \times 10^3$	$7.1 \times 10^3$	$1.0 \times 10^5$
	MULT	$2.7 \times 10^4$	$4.2 \times 10^3$	$1.1 \times 10^4$	$1.3 \times 10^5$
comp03	3IDX	$2.0 \times 10^3$	$2.4 \times 10^4$	$6.4 \times 10^3$	$3.9 \times 10^5$
	MIN	$2.5 \times 10^4$	$2.5 \times 10^3$	$6.6 \times 10^3$	$9.3 \times 10^4$
	MULT	$2.5 \times 10^4$	$3.7 \times 10^3$	$1.0 \times 10^4$	$1.2 \times 10^5$
comp04	3IDX	$1.7 \times 10^3$	$3.0 \times 10^4$	$6.4 \times 10^3$	$3.6 \times 10^5$
	MIN	$3.0 \times 10^4$	$3.0 \times 10^3$	$6.4 \times 10^3$	$1.1 \times 10^5$
	MULT	$3.0 \times 10^4$	$4.4 \times 10^3$	$1.1 \times 10^4$	$1.4 \times 10^5$
comp05	3IDX	$4.5 \times 10^3$	$1.1 \times 10^4$	$8.6 \times 10^3$	$3.9 \times 10^5$
	MIN	$1.5 \times 10^4$	$1.6 \times 10^3$	$9.0 \times 10^3$	$7.6 \times 10^4$
	MULT	$1.5 \times 10^4$	$2.1 \times 10^3$	$1.0 \times 10^4$	$9.0 \times 10^4$
comp06	3IDX	$2.3 \times 10^3$	$3.9 \times 10^4$	$8.6 \times 10^3$	$5.4 \times 10^5$
	MIN	$3.9 \times 10^4$	$4.0 \times 10^3$	$8.6 \times 10^3$	$1.4 \times 10^5$
	MULT	$3.9 \times 10^4$	$6.0 \times 10^3$	$1.4 \times 10^4$	$1.9 \times 10^5$

Continued on next page

Table 1 – continued from previous page

	Model	Cont.	Int.	Rows	Non-Zeros
comp07	3IDX	$2.6 \times 10^3$	$5.5 \times 10^4$	$1.1 \times 10^4$	$7.2 \times 10^5$
	MIN	$5.5 \times 10^4$	$5.2 \times 10^3$	$1.1 \times 10^4$	$2.0 \times 10^5$
	MULT	$5.5 \times 10^4$	$7.8 \times 10^3$	$1.9 \times 10^4$	$2.6 \times 10^5$
comp08	3IDX	$1.8 \times 10^3$	$3.2 \times 10^4$	$6.9 \times 10^3$	$3.6 \times 10^5$
	MIN	$3.2 \times 10^4$	$3.2 \times 10^3$	$6.8 \times 10^3$	$1.1 \times 10^5$
	MULT	$3.2 \times 10^4$	$4.8 \times 10^3$	$1.1 \times 10^4$	$1.5 \times 10^5$
comp09	3IDX	$2.2 \times 10^3$	$2.8 \times 10^4$	$6.7 \times 10^3$	$3.9 \times 10^5$
	MIN	$2.9 \times 10^4$	$2.9 \times 10^3$	$6.7 \times 10^3$	$1.0 \times 10^5$
	MULT	$2.9 \times 10^4$	$4.2 \times 10^3$	$1.1 \times 10^4$	$1.4 \times 10^5$
comp10	3IDX	$2.2 \times 10^3$	$4.1 \times 10^4$	$9.1 \times 10^3$	$5.5 \times 10^5$
	MIN	$4.1 \times 10^4$	$4.2 \times 10^3$	$9.0 \times 10^3$	$1.5 \times 10^5$
	MULT	$4.1 \times 10^4$	$6.3 \times 10^3$	$1.5 \times 10^4$	$2.0 \times 10^5$
comp11	3IDX	$7.6 \times 10^2$	$6.4 \times 10^3$	$2.2 \times 10^3$	$7.3 \times 10^4$
	MIN	$7.0 \times 10^3$	$1.4 \times 10^3$	$3.2 \times 10^3$	$3.2 \times 10^4$
	MULT	$7.0 \times 10^3$	$1.6 \times 10^3$	$3.7 \times 10^3$	$3.9 \times 10^4$
comp12	3IDX	$4.9 \times 10^3$	$2.1 \times 10^4$	$1.1 \times 10^4$	$6.0 \times 10^5$
	MIN	$2.5 \times 10^4$	$2.8 \times 10^3$	$1.2 \times 10^4$	$1.2 \times 10^5$
	MULT	$2.5 \times 10^4$	$3.7 \times 10^3$	$1.5 \times 10^4$	$1.4 \times 10^5$
comp13	3IDX	$1.9 \times 10^3$	$3.2 \times 10^4$	$6.9 \times 10^3$	$3.9 \times 10^5$
	MIN	$3.2 \times 10^4$	$3.1 \times 10^3$	$6.7 \times 10^3$	$1.1 \times 10^5$
	MULT	$3.2 \times 10^4$	$4.7 \times 10^3$	$1.1 \times 10^4$	$1.5 \times 10^5$
comp14	3IDX	$1.9 \times 10^3$	$2.9 \times 10^4$	$6.9 \times 10^3$	$4.3 \times 10^5$
	MIN	$3.0 \times 10^4$	$3.1 \times 10^3$	$7.0 \times 10^3$	$1.1 \times 10^5$
	MULT	$3.0 \times 10^4$	$4.5 \times 10^3$	$1.1 \times 10^4$	$1.5 \times 10^5$
comp15	3IDX	$2.0 \times 10^3$	$2.4 \times 10^4$	$6.4 \times 10^3$	$3.9 \times 10^5$
	MIN	$2.5 \times 10^4$	$2.5 \times 10^3$	$6.6 \times 10^3$	$9.3 \times 10^4$
	MULT	$2.5 \times 10^4$	$3.7 \times 10^3$	$1.0 \times 10^4$	$1.2 \times 10^5$
comp16	3IDX	$2.3 \times 10^3$	$4.6 \times 10^4$	$9.2 \times 10^3$	$6.0 \times 10^5$
	MIN	$4.6 \times 10^4$	$4.3 \times 10^3$	$9.0 \times 10^3$	$1.6 \times 10^5$
	MULT	$4.6 \times 10^4$	$6.5 \times 10^3$	$1.6 \times 10^4$	$2.2 \times 10^5$
comp17	3IDX	$2.2 \times 10^3$	$3.4 \times 10^4$	$8.0 \times 10^3$	$4.8 \times 10^5$
	MIN	$3.5 \times 10^4$	$3.6 \times 10^3$	$8.0 \times 10^3$	$1.3 \times 10^5$
	MULT	$3.5 \times 10^4$	$5.3 \times 10^3$	$1.3 \times 10^4$	$1.7 \times 10^5$
comp18	3IDX	$1.8 \times 10^3$	$1.0 \times 10^4$	$4.4 \times 10^3$	$1.8 \times 10^5$
	MIN	$1.2 \times 10^4$	$1.5 \times 10^3$	$4.9 \times 10^3$	$4.9 \times 10^4$
	MULT	$1.2 \times 10^4$	$1.9 \times 10^3$	$6.2 \times 10^3$	$6.2 \times 10^4$
comp19	3IDX	$1.9 \times 10^3$	$2.3 \times 10^4$	$6.0 \times 10^3$	$3.2 \times 10^5$
	MIN	$2.4 \times 10^4$	$2.6 \times 10^3$	$6.1 \times 10^3$	$8.7 \times 10^4$
	MULT	$2.4 \times 10^4$	$3.7 \times 10^3$	$9.7 \times 10^3$	$1.2 \times 10^5$
comp20	3IDX	$2.5 \times 10^3$	$4.7 \times 10^4$	$1.0 \times 10^4$	$6.5 \times 10^5$
	MIN	$4.7 \times 10^4$	$4.6 \times 10^3$	$1.0 \times 10^4$	$1.7 \times 10^5$
	MULT	$4.7 \times 10^4$	$6.9 \times 10^3$	$1.7 \times 10^4$	$2.3 \times 10^5$
comp21	3IDX	$2.4 \times 10^3$	$3.6 \times 10^4$	$8.1 \times 10^3$	$5.2 \times 10^5$
	MIN	$3.6 \times 10^4$	$3.5 \times 10^3$	$8.1 \times 10^3$	$1.3 \times 10^5$
	MULT	$3.6 \times 10^4$	$5.2 \times 10^3$	$1.3 \times 10^4$	$1.8 \times 10^5$

Continued on next page

Table 1 – continued from previous page

	Model	Cont.	Int.	Rows	Non-Zeros
DDS1	3IDX	$5.2 \times 10^3$	$7.0 \times 10^4$	$2.2 \times 10^4$	$1.2 \times 10^6$
	MIN	$7.1 \times 10^4$	$7.3 \times 10^3$	$1.9 \times 10^4$	$2.6 \times 10^5$
	MULT	$7.1 \times 10^4$	$1.2 \times 10^4$	$3.1 \times 10^4$	$3.5 \times 10^5$
DDS2	3IDX	$5.6 \times 10^2$	$2.3 \times 10^4$	$3.4 \times 10^3$	$2.3 \times 10^5$
	MIN	$2.3 \times 10^4$	$2.5 \times 10^3$	$4.7 \times 10^3$	$7.6 \times 10^4$
	MULT	$2.3 \times 10^4$	$3.5 \times 10^3$	$6.7 \times 10^3$	$1.0 \times 10^5$
DDS3	3IDX	$6.9 \times 10^2$	$1.4 \times 10^4$	$2.7 \times 10^3$	$1.2 \times 10^5$
	MIN	$1.4 \times 10^4$	$2.0 \times 10^3$	$3.9 \times 10^3$	$5.3 \times 10^4$
	MULT	$1.4 \times 10^4$	$2.4 \times 10^3$	$5.2 \times 10^3$	$6.9 \times 10^4$
DDS4	3IDX	$6.2 \times 10^3$	$3.1 \times 10^5$	$2.9 \times 10^4$	$4.1 \times 10^6$
	MIN	$3.1 \times 10^5$	$1.7 \times 10^4$	$3.2 \times 10^4$	$1.1 \times 10^6$
	MULT	$3.1 \times 10^5$	$2.3 \times 10^4$	$5.2 \times 10^4$	$1.4 \times 10^6$
DDS5	3IDX	$3.2 \times 10^3$	$9.5 \times 10^4$	$1.3 \times 10^4$	$8.4 \times 10^5$
	MIN	$9.7 \times 10^4$	$7.2 \times 10^3$	$1.4 \times 10^4$	$3.2 \times 10^5$
	MULT	$9.7 \times 10^4$	$9.1 \times 10^3$	$2.0 \times 10^4$	$4.3 \times 10^5$
DDS6	3IDX	$2.1 \times 10^3$	$3.7 \times 10^4$	$8.4 \times 10^3$	$5.0 \times 10^5$
	MIN	$3.8 \times 10^4$	$3.9 \times 10^3$	$8.5 \times 10^3$	$1.4 \times 10^5$
	MULT	$3.8 \times 10^4$	$5.7 \times 10^3$	$1.4 \times 10^4$	$1.8 \times 10^5$
DDS7	3IDX	$2.3 \times 10^3$	$2.3 \times 10^4$	$5.3 \times 10^3$	$3.4 \times 10^5$
	MIN	$2.5 \times 10^4$	$2.9 \times 10^3$	$7.5 \times 10^3$	$1.0 \times 10^5$
	MULT	$2.5 \times 10^4$	$3.4 \times 10^3$	$8.8 \times 10^3$	$1.3 \times 10^5$
test1	3IDX	$7.7 \times 10^2$	$1.1 \times 10^4$	$2.8 \times 10^3$	$1.1 \times 10^5$
	MIN	$1.2 \times 10^4$	$1.4 \times 10^3$	$3.1 \times 10^3$	$4.1 \times 10^4$
	MULT	$1.2 \times 10^4$	$2.0 \times 10^3$	$4.7 \times 10^3$	$5.5 \times 10^4$
test2	3IDX	$8.3 \times 10^2$	$1.1 \times 10^4$	$3.0 \times 10^3$	$1.1 \times 10^5$
	MIN	$1.2 \times 10^4$	$1.5 \times 10^3$	$3.2 \times 10^3$	$4.2 \times 10^4$
	MULT	$1.2 \times 10^4$	$2.1 \times 10^3$	$5.1 \times 10^3$	$5.6 \times 10^4$
test3	3IDX	$1.3 \times 10^3$	$1.2 \times 10^4$	$4.0 \times 10^3$	$1.4 \times 10^5$
	MIN	$1.3 \times 10^4$	$1.6 \times 10^3$	$3.9 \times 10^3$	$4.6 \times 10^4$
	MULT	$1.3 \times 10^4$	$2.3 \times 10^3$	$6.1 \times 10^3$	$6.1 \times 10^4$
test4	3IDX	$1.5 \times 10^3$	$1.1 \times 10^4$	$4.1 \times 10^3$	$1.3 \times 10^5$
	MIN	$1.2 \times 10^4$	$1.6 \times 10^3$	$4.3 \times 10^3$	$4.5 \times 10^4$
	MULT	$1.2 \times 10^4$	$2.2 \times 10^3$	$6.0 \times 10^3$	$5.9 \times 10^4$
Avg.	MIN	$\times 16.3$	$\times 0.1$	$\times 1.1$	$\times 0.3$
	MULT	$\times 16.3$	$\times 0.2$	$\times 1.6$	$\times 0.4$

In Table 1, we see that the number of continuous variables is much larger in the flow based formulations compared to the three-index formulation. The number of integer variables for the minimum cost flow based formulation is 10% of the number of integer variables in the three-index formulation, and for the multi-commodity flow based formulation the number is 20%. The hope is that the much smaller number of integer variables will lead to fewer branching decisions in the branch-and-bound nodes and thus improve the performance of the MIP solver.

#### 4.1 Lower Bounds Results

The bounds obtained by the flow formulations are compared with the four approaches proposed by [Lach and Lübbecke \(2012\)](#), [Burke et al \(2010\)](#), [Hao and Benlic \(2011\)](#) and [Cacchiani et al \(2013\)](#). In Table 2 – 4 we report the lower bounds obtained on the first 14 COMP data sets for the latter mentioned four approaches and the flow-based formulations when running the approaches for one CPU unit, ten CPU units and forty CPU units. The reason that we only report the results for the first 14 data sets, is that these were the only publicly available data sets at the time where [Lach and Lübbecke \(2012\)](#) and [Burke et al \(2010\)](#) conducted their tests. In each table, we report the number of times that the approaches obtain a bound which is at least as good as the bound obtained by the other approaches and also the number of times that the approaches obtain a bound which is better than all the other approaches. For each data instance and approach, we calculate the gap between the lower bound (LB) obtained by the approach, and best-known upper bound (UB) reported at <http://tabu.diegm.uniud.it/ctt/index.php>. We calculate the gap as  $(UB - LB)/LB$ .

**Table 2** Comparison of the lower bounds obtained for the different approaches when given one CPU time unit; LL12 ([Lach and Lübbecke 2012](#)), BMPR10 ([Burke et al 2010](#)), HB11 ([Hao and Benlic 2011](#)), CCRT13 ([Cacchiani et al 2013](#)), MIN (the minimum cost flow based formulation) and MULT (the multi-commodity flow based formulation). The numbers reported in bold font are the values where the specific models obtained a value which is at least as good as the other formulations. The numbers underlined are the values where the specific models obtained a value which is better than for the other formulations. The UB column reports the best-known upper bounds from <http://tabu.diegm.uniud.it/ctt/index.php>. The two second last lines (Best) denotes the number of times that a specific algorithm obtained a lower bound which was at least as good as the other algorithms (in bold font) and strictly better than the other algorithms (underlines in bold font). The last line (Gap) reports the average gap between the lower bounds obtained by the specific algorithm and the best-known upper bound, where the gap for each instance is calculated as the best-known upper bound subtracted by the lower bound obtained by the algorithm, and then divided by the upper bound.

Instance	UB	BMPR10	LL12	HB11	CCRT13	MIN	MULT
comp01	5	0	4	4	<b>5</b>	<b>5</b>	<b>5</b>
comp02	24	0	0	<b>10</b>	0	0	6
comp03	64	25	0	26	24	<b>27</b>	26
comp04	35	<b>35</b>	22	<b>35</b>	<b>35</b>	24	24
comp05	284	119	92	19	6	<b>132</b>	121
comp06	27	<b>13</b>	7	12	0	12	12
comp07	6	<b>6</b>	0	5	0	0	0
comp08	37	<b>37</b>	30	<b>37</b>	<b>37</b>	26	27
comp09	96	68	37	39	<b>92</b>	49	46
comp10	4	3	2	<b>4</b>	0	<b>4</b>	<b>4</b>
comp11	0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
comp12	298	<b>101</b>	29	43	0	85	85
comp13	59	52	33	46	<b>57</b>	39	38
comp14	51	<b>41</b>	40	<b>41</b>	32	40	<b>41</b>
Best		<b>2</b> <b>3</b>	<b>1</b> <b>0</b>	<b>6</b> <b>1</b>	<b>6</b> <b>2</b>	<b>5</b> <b>2</b>	<b>4</b> <b>0</b>
Gap		37.33%	56.46%	34.99%	50.37%	41.95%	41.75%

When considering the number of times that the approaches obtained the best bound in Table 2–4, we see that the proposed formulations can compete with

**Table 3** Comparison of the lower bounds obtained for the different approaches when given ten CPU time units. The notation follows that of Table 2.

Instance	UB	BMPR10	LL12	HB11	CCRT13	MIN	MULT
comp01	5	4	4	4	<b>5</b>	<b>5</b>	<b>5</b>
comp02	24	0	8	12	<b>16</b>	8	8
comp03	64	33	0	34	<b>52</b>	37	35
comp04	35	<b>35</b>	28	<b>35</b>	<b>35</b>	<b>35</b>	<b>35</b>
comp05	284	111	25	69	6	<b>173</b>	172
comp06	27	<b>15</b>	10	12	11	13	13
comp07	6	<b>6</b>	2	<b>6</b>	<b>6</b>	0	<b>6</b>
comp08	37	<b>37</b>	34	<b>37</b>	<b>37</b>	<b>37</b>	<b>37</b>
comp09	96	65	41	67	<b>92</b>	71	71
comp10	4	<b>4</b>	<b>4</b>	<b>4</b>	2	<b>4</b>	<b>4</b>
comp11	0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
comp12	298	95	32	78	0	<b>129</b>	116
comp13	59	52	39	53	<b>57</b>	54	54
comp14	51	42	41	43	<b>48</b>	43	42
Best		<b>6</b> <u>1</u>	<b>2</b> <u>0</u>	<b>5</b> <u>0</u>	<b>10</b> <u>5</u>	<b>7</b> <u>2</u>	<b>6</b> <u>0</u>
Gap		28.84%	45.60%	27.00%	26.62%	27.37%	22.61%

**Table 4** Comparison of the lower bounds obtained for the different approaches when given forty CPU time units. The notation follows that of Table 2.

Instance	UB	BMPR10	LL12	HB11	CCRT13	MIN	MULT
comp01	5	<b>5</b>	4	4	<b>5</b>	<b>5</b>	<b>5</b>
comp02	24	1	11	12	<b>16</b>	8	8
comp03	64	33	25	36	<b>52</b>	38	37
comp04	35	<b>35</b>	28	<b>35</b>	<b>35</b>	<b>35</b>	<b>35</b>
comp05	284	114	108	80	166	<b>186</b>	181
comp06	27	<b>16</b>	10	<b>16</b>	11	<b>16</b>	<b>16</b>
comp07	6	<b>6</b>	<b>6</b>	<b>6</b>	<b>6</b>	0	<b>6</b>
comp08	37	<b>37</b>	<b>37</b>	<b>37</b>	<b>37</b>	<b>37</b>	<b>37</b>
comp09	96	66	46	67	<b>92</b>	74	73
comp10	4	<b>4</b>	<b>4</b>	<b>4</b>	2	<b>4</b>	<b>4</b>
comp11	0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
comp12	298	95	53	84	100	<b>142</b>	140
comp13	59	54	41	55	57	56	<b>59</b>
comp14	51	42	46	43	<b>48</b>	44	43
Best		<b>7</b> <u>0</u>	<b>4</b> <u>0</u>	<b>6</b> <u>0</u>	<b>9</b> <u>4</u>	<b>8</b> <u>2</u>	<b>8</b> <u>1</u>
Gap		26.45%	32.47%	25.06%	20.20%	24.82%	19.89%

most of the approaches, except for the proposed method by [Cacchiani et al \(2013\)](#) which seems to perform better on most instances. Considering the average gap, we see that for the short running time, the approaches by [Burke et al \(2010\)](#) and [Hao and Benlic \(2011\)](#) have a better gap than the other approaches. As the running time increases, the average gap by [Cacchiani et al \(2013\)](#) and the multi-commodity flow based formulation becomes lower than all the other approaches. The performance of [Cacchiani et al \(2013\)](#) is especially evident in Table 5 where we compare all the three data sets between the flow formulations and the approach proposed by [Cacchiani et al \(2013\)](#) where the time limit is forty CPU units. Here we see that [Cacchiani et al \(2013\)](#) obtains a better bound more often than the flow formulations as well as a better average gap. However the flow formulations generate better lower bounds on two of the instances at ten and forty CPU units;

comp05 and comp12, where the bound generated for comp12 is an improvement of the best-known bound.

**Table 5** Comparison of the lower bounds obtained for the different model formulations when given forty CPU time units; CCRT13 (Cacchiani et al 2013), MIN (the minimum cost flow based formulation) and MULT (the multi-commodity flow based formulation). The notation follows that of Table 2.

Instance	UB	CCRT13	MIN	MULT
comp01	5	<b>5</b>	<b>5</b>	<b>5</b>
comp02	24	<b>16</b>	8	8
comp03	64	<b>52</b>	38	37
comp04	35	<b>35</b>	<b>35</b>	<b>35</b>
comp05	284	166	<b>186</b>	181
comp06	27	11	<b>16</b>	<b>16</b>
comp07	6	<b>6</b>	0	<b>6</b>
comp08	37	<b>37</b>	<b>37</b>	<b>37</b>
comp09	96	<b>92</b>	74	73
comp10	4	2	<b>4</b>	<b>4</b>
comp11	0	<b>0</b>	<b>0</b>	<b>0</b>
comp12	298	100	<b>142</b>	140
comp13	59	57	56	<b>59</b>
comp14	51	<b>48</b>	44	43
comp15	64	<b>52</b>	38	37
comp16	18	<b>13</b>	<b>13</b>	11
comp17	56	<b>48</b>	43	44
comp18	61	<b>52</b>	36	30
comp19	57	48	<b>56</b>	55
comp20	4	<b>4</b>	0	0
comp21	74	<b>68</b>	56	57
DDS1	48	40	<b>46</b>	44
DDS2	0	<b>0</b>	<b>0</b>	<b>0</b>
DDS3	0	<b>0</b>	<b>0</b>	<b>0</b>
DDS4	17	<b>17</b>	15	15
DDS5	0	<b>0</b>	<b>0</b>	<b>0</b>
DDS6	0	<b>0</b>	<b>0</b>	<b>0</b>
DDS7	0	<b>0</b>	<b>0</b>	<b>0</b>
test1	224	<b>224</b>	<b>224</b>	<b>224</b>
test2	16	<b>16</b>	<b>16</b>	<b>16</b>
test3	67	<b>59</b>	<b>59</b>	<b>59</b>
test4	73	<b>46</b>	44	43
Best		<b>25</b>	<b>19</b>	<b>16</b>
		<b>11</b>	<b>4</b>	<b>1</b>
Gap		13.99%	21.97%	19.73%

#### 4.2 Comparing Upper-Bound Formulations

Since Lach and Lübbecke (2012) and Burke et al (2010) obtain both lower and upper bounds we also compare these with the bounds obtained by the flow formulations. The results for forty CPU units are given in Table 6. Here we see that the flow formulations obtain better lower bounds in most cases. As for the upper bounds, Lach and Lübbecke (2012) and the minimum cost flow formulation appear to perform similarly. The minimum cost flow formulation obtains upper bounds that are at least as good as the other formulations in most cases, but Lach and Lübbecke (2012) obtain upper bounds that are better than the other formulations

in more cases. However, the minimum cost flow formulation has the best average performance, as we see it has a lower average gap than the other formulations.

**Table 6** Comparison of the lower (LB) and upper bounds (UB) obtained for the different model formulations when given forty CPU time units; LL12 (Lach and Lübbecke 2012), BMPR10 (Burke et al 2010), MIN (the minimum cost flow based formulation) and MULT (the multi-commodity flow based formulation). The numbers reported in bold font are the values where the specific models obtained a value which is at least as good as the other formulations. The numbers underlined are the values where the specific models obtained a value which is better than for the other formulations. The two second last lines (Best) denotes the number of times that a specific algorithm obtained a bound which was at least as good as the other algorithms (in bold font) and strictly better than the other algorithms (underlines in bold font). The last line (Gap) reports the average gap between the lower and upper bounds obtained by the specific algorithm, where the gap for each instance is calculated as  $(UB - LB)/UB$ .

Instance	BMPR10		LL12		MIN		MULT	
	LB	UB	LB	UB	LB	UB	LB	UB
comp01	<b>5</b>	9	4	12	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>
comp02	1	63	<b>11</b>	46	8	<b>45</b>	8	59
comp03	33	123	25	<b>66</b>	<b>38</b>	123	37	99
comp04	<b>35</b>	36	28	38	<b>35</b>	<b>35</b>	<b>35</b>	<b>35</b>
comp05	114	629	108	368	<b>186</b>	<b>355</b>	181	377
comp06	<b>16</b>	<b>46</b>	10	51	<b>16</b>	92	<b>16</b>	92
comp07	<b>6</b>	45	<b>6</b>	<b>25</b>	0	179	<b>6</b>	-
comp08	<b>37</b>	41	<b>37</b>	44	<b>37</b>	<b>37</b>	<b>37</b>	41
comp09	66	105	46	<b>99</b>	<b>74</b>	105	73	103
comp10	<b>4</b>	23	<b>4</b>	<b>16</b>	<b>4</b>	18	<b>4</b>	68
comp11	<b>0</b>	12	<b>0</b>	7	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
comp12	95	785	53	548	<b>142</b>	<b>423</b>	140	500
comp13	54	67	41	66	56	66	<b>59</b>	<b>59</b>
comp14	42	55	<b>46</b>	<b>53</b>	44	55	43	74
Best	<b>7</b>	<b>1</b>	<b>6</b>	<b>5</b>	<b>10</b>	<b>7</b>	<b>8</b>	<b>4</b>
	<b>0</b>	<b>1</b>	<b>2</b>	<b>5</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>1</b>
Gap	58.07%		60.54%		41.62%		45.04%	

### 4.3 Comparing with the Three-Index Formulation

In this section, we compare the flow formulations with the three-index formulation.

In Table 7 we provide the results of both the three-index formulation (model (1) – (17)) and the flow based formulations. Here we see that the flow formulations clearly outperform the three-index formulation and we obtain a new lower bound in one of the instances compared to the best-known bound. This improvement makes the models riveting as some of the other approaches based on the three-index formulation from the literature might also benefit from this reformulation.

On a final note, these computational results show that the minimum cost flow based formulation in general has better performance than the multi-commodity flow based formulation as it obtains better bounds in more cases and it has a lower average gap.

**Table 7** Comparison of the lower and upper bounds obtained for the different model formulations when given forty CPU time units; 3IDX (the three-index formulation), MIN (the minimum cost flow based formulation) and MULT (the multi-commodity flow based formulation). The notation follows that of Table 6.

Instance	3IDX		MIN		MULT	
	LB	UB	LB	UB	LB	UB
comp01	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>
comp02	0	109	<b>8</b>	<b>45</b>	<b>8</b>	59
comp03	33	136	<b>38</b>	123	37	<b>99</b>
comp04	<b>35</b>	41	<b>35</b>	<b>35</b>	<b>35</b>	<b>35</b>
comp05	161	427	<b>186</b>	<b>355</b>	181	377
comp06	12	98	<b>16</b>	<b>92</b>	<b>16</b>	<b>92</b>
comp07	3	<b>118</b>	0	179	<b>6</b>	-
comp08	<b>37</b>	45	<b>37</b>	<b>37</b>	<b>37</b>	41
comp09	66	157	<b>74</b>	105	73	<b>103</b>
comp10	<b>4</b>	39	<b>4</b>	<b>18</b>	<b>4</b>	68
comp11	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
comp12	108	629	<b>142</b>	<b>423</b>	140	500
comp13	51	126	56	66	<b>59</b>	<b>59</b>
comp14	41	144	<b>44</b>	<b>55</b>	43	74
comp15	33	136	<b>38</b>	123	37	<b>99</b>
comp16	8	102	<b>13</b>	61	11	<b>42</b>
comp17	41	175	43	123	<b>44</b>	<b>109</b>
comp18	24	133	<b>36</b>	<b>78</b>	30	108
comp19	53	114	<b>56</b>	<b>57</b>	55	<b>57</b>
comp20	<b>0</b>	146	<b>0</b>	<b>50</b>	<b>0</b>	96
comp21	49	235	56	156	<b>57</b>	<b>133</b>
DDS1	45	92	<b>46</b>	<b>70</b>	44	76
DDS2	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
DDS3	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
DDS4	<b>15</b>	67	<b>15</b>	<b>17</b>	<b>15</b>	12079
DDS5	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
DDS6	<b>0</b>	51	<b>0</b>	<b>2</b>	<b>0</b>	27
DDS7	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
test1	<b>224</b>	233	<b>224</b>	<b>224</b>	<b>224</b>	<b>224</b>
test2	<b>16</b>	<b>19</b>	<b>16</b>	<b>19</b>	<b>16</b>	<b>19</b>
test3	<b>59</b>	83	<b>59</b>	<b>75</b>	<b>59</b>	<b>75</b>
test4	43	109	<b>44</b>	<b>91</b>	43	107
Best	<b>15</b>	<b>8</b>	<b>28</b>	<b>24</b>	<b>21</b>	<b>19</b>
	<b>0</b>	<b>1</b>	<b>11</b>	<b>12</b>	<b>4</b>	<b>7</b>
Gap		53.57%		39.51%		41.88%

## 5 Perspectives

We proposed two mixed integer programming models for the curriculum based course timetabling problem (CTT). We based both our models on network flow problems; the minimum cost flow problem and the multi-commodity flow problem. These models are exact, meaning that they will obtain optimal solutions given enough computational resources. We showed that these models have far fewer integer variables than the standard three-index formulation for the CTT. By experimental results, we also showed that the models outperform the three-index formulation regarding finding feasible solutions when solved by a MIP solver.

Regarding lower bounds, the formulations are competitive with most of the mixed integer programming based approaches from the literature and improve one of the currently best known lower bounds on the benchmarking instances from the second international timetabling competition.

Regarding upper bounds, the minimum cost flow based formulation performs better than other state-of-the-art MIP based approaches.

Furthermore, some of the approaches from the literature are based on models similar to the three-index formulation. Since we showed that our formulations outperform the three-index formulation, we believe that these approaches can also benefit from adopting the proposed network flow formulations.

## References

- Ahuja RK, Magnanti TL, Orlin JB (1993) *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA
- Bagger N, Kristiansen S, Sørensen M, Stidsen T (2015) Flow Formulation-based Model for the Curriculum-based Course Timetabling Problem, pp 825–848. *Proceedings of the 7th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2015)*
- Bettinelli A, Cacchiani V, Roberti R, Toth P (2015) An overview of curriculum-based course timetabling. *TOP* pp 1–37
- Bonutti A, De Cesco F, Di Gaspero L, Schaerf A (2012) Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research* 194(1):59–70
- Bron C, Kerbosch J (1973) Algorithm 457: Finding all cliques of an undirected graph. *Commun ACM* 16(9):575–577, DOI 10.1145/362342.362367, URL <http://doi.acm.org/10.1145/362342.362367>
- Burke EK, Mareček J, Parkes AJ, Rudová H (2008) Penalising patterns in timetables: Novel integer programming formulations. In: Kalcsics J, Nickel S (eds) *Operations Research Proceedings 2007, Operations Research Proceedings*, vol 2007, Springer Berlin Heidelberg, pp 409–414, 10.1007/978-3-540-77903-2\_63
- Burke EK, Mareček J, Parkes AJ, Rudová H (2010) Decomposition, reformulation, and diving in university course timetabling. *Computers & Operations Research* 37(3):582–597
- Burke EK, Mareček J, Parkes AJ, Rudová H (2012) A branch-and-cut procedure for the udine course timetabling problem. *Annals of Operations Research* 194(1):71–87
- Cacchiani V, Caprara A, Roberti R, Toth P (2013) A new lower bound for curriculum-based course timetabling. *Computers & Operations Research* 40(10):2466 – 2477
- Di Gaspero L, Schaerf A (2003) Multi-neighbourhood local search with application to course timetabling. In: Burke E, De Causmaecker P (eds) *Practice and Theory of Automated Timetabling IV*, *Lecture Notes in Computer Science*, vol 2740, Springer Berlin Heidelberg, pp 262–275, DOI 10.1007/978-3-540-45157-0\_17, URL [http://dx.doi.org/10.1007/978-3-540-45157-0\\_17](http://dx.doi.org/10.1007/978-3-540-45157-0_17)
- Di Gaspero L, McCollum B, Schaerf A (2007) The second international timetabling competition (itc-2007): Curriculum-based course timetabling (track 3). Tech. rep., School of Electronics, Electrical Engineering and Computer Science, Queenes University SARC Building, Belfast, United Kingdom
- Gurobi Optimization, Inc (2016) Gurobi optimizer reference manual. URL <http://www.gurobi.com>
- Hao JK, Benlic U (2011) Lower bounds for the ITC-2007 curriculum-based course timetabling problem. *European Journal of Operational Research* 212(3):464 – 472
- Kleinberg J, Tardos E (2005) *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
- Lach G, Lübbecke M (2008) Optimal university course timetables and the partial transversal polytope. In: McGeoch C (ed) *Experimental Algorithms, Lecture Notes in Computer Science*, vol 5038, Springer Berlin / Heidelberg, pp 235–248
- Lach G, Lübbecke M (2012) Curriculum based course timetabling: new solutions to udine benchmark instances. *Annals of Operations Research* 194:255–272
- McCollum B, Schaerf A, Paechter B, McMullan P, Lewis R, Parkes AJ, Gaspero LD, Qu R, Burke EK (2010) Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing* 22(1):120–130

## A Notation

In this section we provide an overview of all the notation that we used throughout the paper. We divided the notation into five lists; abbreviations, functions, sets, parameters, variables and floor graph notations. In each list we have ordered the notation alphabetically. For each notation, we have provided a short description and a reference to the page number where the notation is first used.

### Abbreviations

<b>A</b>	Abbreviation of the Availability ( <b>A</b> ) constraint .....	2
<b>BMPR10</b>	The results reported by <a href="#">Burke et al (2010)</a> .....	22
<b>C</b>	Abbreviation of the Conflicts ( <b>C</b> ) constraint .....	2
<b>CCRT13</b>	The results reported by <a href="#">Cacchiani et al (2013)</a> .....	22
<b>CTT</b>	Abbreviation of <i>Curriculum-based Course Timetabling</i> .....	1
$\mathcal{FLP}$	The LP-relaxation of model (63) – (67) .....	15
<b>HB11</b>	The results reported by <a href="#">Hao and Benlic (2011)</a> .....	22
<b>IL</b>	Abbreviation of the Isolated Lectures ( <b>IL</b> ) constraint .....	3
<b>ITC2007</b>	The second international timetabling competition (ITC2007) held in 2007 .....	2
<b>L</b>	Abbreviation of the Lectures ( <b>L</b> ) constraint .....	2
<b>MIN</b>	The results obtained by the minimum cost flow based formulation .....	22
<b>MWD</b>	Abbreviation of the Minimum Working Days ( <b>MWD</b> ) constraint .....	3
<b>MULT</b>	The results obtained by the multi-commodity flow based formulation .....	22
<b>RC</b>	Abbreviation of the Room Capacity ( <b>RC</b> ) constraint .....	3
<b>RO</b>	Abbreviation of the Room Occupancy ( <b>RO</b> ) constraint .....	2
<b>RStab</b>	Abbreviation of the Room Stability ( <b>RStab</b> ) constraint .....	3

### Functions

$(\cdot)^+$	A function returning the the input if it is a non-negative value; zero is returned otherwise .....	5
$o(\cdot)$	The objective value of the solution given as input .....	15

### Sets

$A$	The set of feasible period-room solutions in the multi-commodity flow formulation .	15
$\mathcal{C}$	The set of courses .....	5
$\mathcal{C}_l$	The set of courses that is taught by lecturer $l \in \mathcal{L}$ .....	5
$\mathcal{C}_q$	The set of courses that belongs to curriculum $q \in \mathcal{Q}$ .....	5
$\mathcal{C}_\gamma$	The set of courses that belongs to clique $\gamma \in \Gamma$ .....	7
$\mathcal{D}$	The set of days .....	5
$\Gamma$	The set of cliques in the <i>course conflict graph</i> .....	7
$\mathcal{L}$	The set of lecturers .....	5
$\mathcal{P}$	The set of periods .....	5
$\mathcal{P}_d$	The set of periods in $\mathcal{P}$ that belongs to day $d \in \mathcal{D}$ .....	5
$\Phi_p$	The set of periods that are adjacent to the period $p \in \mathcal{P}$ .....	5
$\mathcal{Q}$	The set of curricula .....	5
$\mathcal{R}$	The set of rooms .....	5
$\mathbb{Z}^+$	The set of non-negative integers, i.e., $\mathbb{Z}^+ = \{x \in \mathbb{Z} : x \geq 0\}$ .....	5
$\emptyset$	The empty set .....	15

### Parameters

$C_r$	The capacity of room $r \in \mathcal{R}$ .....	5
$F_{c,p}$	Binary value indicating whether or not course $c \in \mathcal{C}$ can be assigned to period $p \in \mathcal{P}$ .....	5
$L_c$	The number of lectures to schedule for course $c \in \mathcal{C}$ .....	5
$M_c$	The desired number minimum working days for course $c \in \mathcal{C}$ .....	5
$S_c$	The number of students attending course $c \in \mathcal{C}$ .....	5
$\mathcal{T}$	The maximum number of periods for any day, i.e., $\mathcal{T} = \max_{d \in \mathcal{D}} \{ \mathcal{P}_d \}$ .....	7
$W^{\mathbf{IL}}$	The cost of violating the Isolated Lectures ( <b>IL</b> ) constraint .....	5
$W^{\mathbf{MWD}}$	The cost of violating the Minimum Working Days ( <b>MWD</b> ) constraint .....	5
$W^{\mathbf{RC}}$	The cost of violating the Room Capacity ( <b>RC</b> ) constraint .....	5
$W^{\mathbf{RStab}}$	The cost of violating the Room Stability ( <b>RStab</b> ) constraint .....	5

*Variables*

$f_{c,p,r}$	The amount of flow on the arc $(c,p) \rightarrow (r,p)$ in the minimum cost flow graph. A Non-negative continuous variable indicating by how big a fraction course $c \in \mathcal{C}$ is assigned to period $p \in \mathcal{P}$ and room $r \in \mathcal{R}$ in the combined model .....	11
$f_{c,p}^u$	The amount of flow on the arc $(u) \rightarrow (c,p)$ in the minimum cost flow graph .....	11
$f_{r,p}^v$	The amount of flow on the arc $(r,p) \rightarrow (v)$ in the minimum cost flow graph .....	12
$s_{q,p}$	Binary variable indicating whether or not curriculum $q \in \mathcal{Q}$ has an isolated lecture in period $p \in \mathcal{P}$ .....	5
$t_{c,d}$	Binary variable indicating whether or not course $c \in \mathcal{C}$ is assigned to at least one period from the set $\mathcal{P}_d$ for day $d \in \mathcal{D}$ .....	5
$w_c$	Non-negative integer variable indicating by how many days the Minimum Working Days ( <b>MWD</b> ) constraint is violated for course $c \in \mathcal{C}$ .....	5
$x_{c,p}$	Binary variable indicating whether or not course $c \in \mathcal{C}$ has a lecture assigned in period $p \in \mathcal{P}$ .....	9
$x_{c,p,r}$	Binary variable indicating whether or not course $c \in \mathcal{C}$ is assigned to period $p \in \mathcal{P}$ and room $r \in \mathcal{R}$ .....	5
$y_{c,r}$	Integer variable indicating the number of times that course $c \in \mathcal{C}$ is assigned to room $r \in \mathcal{R}$ .....	13
$z_{c,r}$	Binary variable indicating whether or not course $c \in \mathcal{C}$ has at least one lecture assigned in room $r \in \mathcal{R}$ .....	5

*Flow graph notations*

$(\cdot)$	A node in a graph, e.g., a node corresponding to a course and period pair is denoted $(c,p)$ . The source node and sink node of the minimum cost flow graph are denoted $(u)$ and $(v)$ .....	10
$(\cdot) \rightarrow (\cdot)$	An arc in a graph .....	11
$(\cdot) \rightarrow \dots \rightarrow (\cdot)$	A path in a graph .....	14