A MIP Formulation of the International Timetabling Competition 2019 Problem

Holm, Dennis Søren; Mikkelsen, Rasmus Ørnstrup; Sørensen, Matias; Stidsen, Thomas Jacob Riis

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Holm, D. S., Mikkelsen, R. Ø., Sørensen, M., & Stidsen, T. J. R. (2020). *A MIP Formulation of the International Timetabling Competition 2019 Problem*. Technical University of Denmark.

A MIP Formulation of the International Timetabling Competition 2019 Problem

Dennis S. Holm · Rasmus Ø. Mikkelsen · Matias Sørensen · Thomas R. Stidsen

September 2020

Keywords Mixed Integer Programming · University Timetabling · International Timetabling Competition 2019 · ITC2019

Dennis S. Holm
E-mail: dsho@dtu.dk; dennisholm@get2net.dk

Rasmus Ø. Mikkelsen
E-mail: rasmi@dtu.dk

Matias Sørensen
E-mail: sorensen.matias@gmail.com

Thomas J. R. Stidsen
Tel.: +45 45254449
E-mail: thst@dtu.dk

Dennis S. Holm · Rasmus Ø. Mikkelsen · Thomas R. Stidsen
Technical University of Denmark
DTU Management
Akademivej
Building 358
2800 Kgs. Lyngby

1 Introduction

This report considers the problem presented in the International Timetabling Competition 2019 (ITC2019). The ITC2019 problem description presents a generalized model for the University Course Timetabling problem combined with an XML data format. The generalized model aims to include most of the aspects that universities worldwide might consider when constructing a timetable. The model has been simplified to some extent, but the complexity of the problem remains. To understand the origin of the data, the data format, and to get a more in-depth description of the generalized model, it is encouraged to read the ITC2019 problem description (Müller et al., 2018).

When considering an operational research problem like the ITC2019, it can be beneficial to describe the problem with a mathematical model. It is an excellent way to check if the problem formulation has been wholly understood. It also provides some idea of the problem's complexity. Additionally, if a commercial solver can solve the mathematical model, no further work needs to be done. Moreover, a mathematical model can serve as a basis for developing matheuristics and can also validate solutions from other solution methods, i.e., metaheuristics.

In this report, the ITC2019 problem is described using a linear Mixed Integer Programming (MIP) model. The MIP model has been verified by comparing objective penalties and feasibility of solutions with the validator provided for the ITC2019 (www.itc2019.org/validator).

2 Notations

This section contains an overview of the notation for the MIP model. Table 1 shows the sets that are used. Table 2 contains the notation of the parameters that are given in the XML data. Table 3 shows other modelling notation.

Symbol	Description
Δ	Set of distribution constraints
$\mathcal{P}$	Set of penalty variables
T	Set of time slots
$\mathcal{D}$	Set of days
$\mathcal{W}$	Set of weeks
$\mathcal{K}$	Set of courses
$\mathcal{C}$	Set of classes
$\mathcal{T}$	Set of times
$\mathcal{R}$	Set of rooms
$\mathcal{S}$	Set of students
$\mathcal{C}_\delta$	Set of classes for a distribution constraint $\delta \in \Delta$
$\mathcal{C}_s$	Set of classes a student $s \in \mathcal{S}$ can attend
$\mathcal{C}_\zeta$	Set of classes of a subpart $\zeta \in \mathrm{Z}_\omega$
$\mathcal{R}_c$	Set of available rooms for a class $c \in \mathcal{C}$
$\mathcal{T}_c$	Set of available times for a class $c \in \mathcal{C}$
$\mathcal{K}_s$	Set of courses a student $s \in \mathcal{S}$ must attend
$\mathcal{S}_c$	Set of students that can attend a class $c \in \mathcal{C}$
$\mathcal{S}_k$	Set of students that must attend a course $k \in \mathcal{K}$
Ω_k	Set of configurations of a course $k \in \mathcal{K}$
Z_ω	Set of subpart of a configuration $\omega \in \Omega_k$

Table 1 Set notations.

Symbol	Description
c_δ	Cost of the distribution constraint $\delta \in \Delta$
$p_{c,t}$	Penalty of assigning time $t \in \mathcal{T}$ to class $c \in \mathcal{C}$
$p_{c,r}$	Penalty of assigning room $r \in \mathcal{R}$ to class $c \in \mathcal{C}$
ψ_t	Objective weight of the time penalties
ψ_r	Objective weight of the room penalties
ψ_δ	Objective weight of the distribution constraint penalties
ψ_s	Objective weight of the student conflict penalties
$\mathtt{D}$	Distribution constraint parameter
$\mathtt{G}$	Distribution constraint parameter
$\mathtt{M}$	Distribution constraint parameter
$\mathtt{R}$	Distribution constraint parameter
$\mathtt{S}$	Distribution constraint parameter
t^{start}	The starting time slot of time $t \in \mathcal{T}$, $t^{\text{start}} \in \mathrm{T}$
t^{length}	The duration in time slots of time $t \in \mathcal{T}$
t^{end}	The ending time of time $t \in \mathcal{T}$, $t^{\text{start}} + t^{\text{length}} = t^{\text{end}} \in \mathrm{T}$
t^{days}	The set of days of time $t \in \mathcal{T}$, $t^{\text{days}} \subseteq \mathcal{D}$
$t^{\text{days.first}}$	The first day of time $t \in \mathcal{T}$, $t^{\text{days.first}} \in \mathcal{D}$
t^{weeks}	The set of weeks of time $t \in \mathcal{T}$, $t^{\text{weeks}} \subseteq \mathcal{D}$
$t^{\text{weeks.first}}$	The first week of time $t \in \mathcal{T}$, $t^{\text{weeks.first}} \in \mathcal{W}$

Table 2 Parameter notations. Parameters are given in the data sets.

Symbol	Description
c_i	A specific class with ID $i \in \mathbb{Z}^+$
c_i^{parent}	The parent class of class c_i, $c_i^{\text{parent}} \in \mathcal{C}$
c^{limit}	The student limit of class c
$\tilde{r}$	A 'dummy' room, which only exists in the model and does not follow the rules of a regular room.
r_i	A room of class c_i, $r_i \in \mathcal{R}_{c_i}$
t_i	A time of class c_i, $t_i \in \mathcal{T}_{c_i}$
$\bar{t}$	Another time different from $t \in \mathcal{T}$, $\bar{t} \in \mathcal{T}$
$\bar{\tau}$	Another time slot different from $\tau \in \mathrm{T}$, $\bar{\tau} \in \mathrm{T}$
c_p	Cost of the penalty $p \in \mathcal{P}$
T'	Set of start time slots, $\mathrm{T}' \subseteq \mathrm{T}$
T''	Set of end time slots, $\mathrm{T}'' \subseteq \mathrm{T}$
M	Big-M

Table 3 Other modelling notation.

3 Decision variables

The model includes two main decision variables; the scheduling variable $x_{c,t,r}$ and the student sectioning variable $e_{s,c}$. The scheduling variable is binary with indices, classes, times, and rooms. It is defined as

$$x_{c,t,r} = \begin{cases} 1 & \text{if class } c \in \mathcal{C} \text{ is scheduled in time } t \in \mathcal{T}_c \text{ in room } r \in \mathcal{R}_c \\ 0 & \text{otherwise} \end{cases}$$

and is defined for all $c \in \mathcal{C}$, $t \in \mathcal{T}_c$ and $r \in \mathcal{R}_c$. If the class does not need to be assigned a room we set $\mathcal{R}_c = \{\tilde{r}\}$, where $\tilde{r}$ is a 'dummy' room and $\tilde{r} \notin \mathcal{R}$.
The scheduling variables $x_{c,t,r}$ (class-time-room) leads to the following auxiliary variables $y_{c,t}$ (class-time), $z_{c,d}$ (class-day) and $w_{c,r}$ (class-room).

$$y_{c,t} = \begin{cases} 1 & \text{if class } c \in \mathcal{C} \text{ is scheduled in time } t \in \mathcal{T}_c \\ 0 & \text{otherwise} \end{cases}$$

$$z_{c,d} = \begin{cases} 1 & \text{if class } c \in \mathcal{C} \text{ is scheduled on day } d \\ 0 & \text{otherwise} \end{cases}$$

$$w_{c,r} = \begin{cases} 1 & \text{if class } c \in \mathcal{C} \text{ is scheduled in room } r \in \mathcal{R}_c \\ 0 & \text{otherwise} \end{cases}$$

The student sectioning variable $e_{s,c}$ is also binary with indices; students and classes. It is defined as

$$e_{s,c} = \begin{cases} 1 & \text{if student } s \in \mathcal{S} \text{ is attending class } c \in \mathcal{C}_s \\ 0 & \text{otherwise} \end{cases}$$

and is defined for all $s \in \mathcal{S}$ and $c \in \mathcal{C}_s$.

4 Constraints

The constraints are divided into three categories; the primary constraints that ensure general timetable feasibility, the distribution constraints that define the feasibility between the scheduling of classes, and the student sectioning constraints that ensure that the students are assigned the classes according to the enrollment and the structure of the courses.

4.1 Primary constraints

The primary constraints ensure that all classes are scheduled and that no room can be double booked.

$$\sum_{t \in \mathcal{T}_c} \sum_{r \in \mathcal{R}_c} x_{c,t,r} = 1 \qquad \forall\, c \in \mathcal{C}$$

This constraint defines that all classes must be assigned a time and a room (possibly the dummy-room if no room is required).

$$\sum_{\substack{c \in \mathcal{C}, \\ \bar{t} \in \mathcal{T}: \\ t.\mathtt{Overlap}(\bar{t})}} x_{c,\bar{t},r} + M \sum_{c \in \mathcal{C}} x_{c,t,r} \leq M \qquad \forall\, r \in \mathcal{R}, t \in \mathcal{T}$$

The constraint ensures that if a class is scheduled in room r in time t, then there can be no classes scheduled in any overlapping times in the same room. The `Overlap` function is described further in appendix A. The big-M is equal to the number of classes $M = |\mathcal{C}|$.

Furthermore we have the following constraints to control the auxiliary variables

$$\sum_{r \in \mathcal{R}_c} x_{c,t,r} = y_{c,t} \qquad \forall\, c \in \mathcal{C}, t \in \mathcal{T}_c$$

$$\sum_{\substack{t \in \mathcal{T}_c : d \in t^{\text{days}}, \\ r \in \mathcal{R}_c}} x_{c,t,r} = z_{c,d} \qquad \forall\, c \in \mathcal{C}, d \in \mathcal{D}$$

$$\sum_{t \in \mathcal{T}_c} x_{c,t,r} = w_{c,r} \qquad \forall\, c \in \mathcal{C}, r \in \mathcal{R}_c$$

4.2 Distribution constraints

In this section, the modeling of the distribution constraints of different types is described. Each subsection considers a single distribution constraint of the given type. The constraints presented are therefore generated for each distribution constraint of that type. If any auxiliary variables are defined, they have an extra dimension δ that is not written explicitly, such that auxiliary variables from two different distribution constraints are not mixed. Each type of distribution constraint can be either soft or hard. Each soft constraint will include a penalty variable p, which dimensions will not be stated explicitly. The cost of the penalty variable is denoted by c_p. Each distribution constraint has a set of classes $\mathcal{C}_\delta$. The soft distribution constraints have a penalty c_δ. It will be stated in the section if the distribution constraint has parameters.

4.2.1 SameStart

The `SameStart` constraint says that if a class c_i is assigned time t_i, which starts at time slot τ, then c_j cannot be assigned a time that starts at a different time slot. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \sum_{\substack{t_i \in \mathcal{T}_{c_i}: \\ t_i^{\text{start}} = \tau}} y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_j^{\text{start}} \neq \tau}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, \tau \in \bigcup_{t \in \mathcal{T}_{c_i}} t^{\text{start}}$$

$$\text{Soft:} \sum_{\substack{t_i \in \mathcal{T}_{c_i}: \\ t_i^{\text{start}} = \tau}} y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_j^{\text{start}} \neq \tau}} y_{c_j,t_j} - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, \tau \in \bigcup_{t \in \mathcal{T}_{c_i}} t^{\text{start}}$$

4.2.2 SameTime

The `SameTime` constraint says that if a class c_i is assigned time t_i, then c_j cannot be assigned a time that is not taught at the same time. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg((t_i^{\text{start}} \leq t_j^{\text{start}} \wedge t_j^{\text{end}} \leq t_i^{\text{end}}) \\ \vee (t_j^{\text{start}} \leq t_i^{\text{start}} \wedge t_i^{\text{end}} \leq t_j^{\text{end}}))}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg((t_i^{\text{start}} \leq t_j^{\text{start}} \wedge t_j^{\text{end}} \leq t_i^{\text{end}}) \\ \vee (t_j^{\text{start}} \leq t_i^{\text{start}} \wedge t_i^{\text{end}} \leq t_j^{\text{end}}))}} y_{c_j,t_j} - 1 \leq p \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.3 DifferentTime

The `DifferentTime` constraint says that if a class c_i is assigned time t_i, then c_j cannot be assigned a time that overlaps the same time of the day. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \qquad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg((t_i^{\text{end}} \leq t_j^{\text{start}}) \\ \vee (t_j^{\text{end}} \leq t_i^{\text{start}}))}} y_{c_j,t_j} \qquad \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \qquad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg((t_i^{\text{end}} \leq t_j^{\text{start}}) \\ \vee (t_j^{\text{end}} \leq t_i^{\text{start}}))}} y_{c_j,t_j} \qquad - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.4 SameDays

The `SameDays` constraint says that if a class c_i is assigned time t_i, with day-set t_i^{days}, then c_j cannot be assigned a time with day-set t_j^{days} if the smaller of the day-sets is not included in the larger. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \qquad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{days}} \nsubseteq t_j^{\text{days}} \wedge \\ t_j^{\text{days}} \nsubseteq t_i^{\text{days}}}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \qquad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{days}} \nsubseteq t_j^{\text{days}} \wedge \\ t_j^{\text{days}} \nsubseteq t_i^{\text{days}}}} y_{c_j,t_j} - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.5 DifferentDays

The `DifferentDays` constraint says that if a class c_i is assigned time t_i, with day-set t_i^{days}, then c_j cannot be assigned a time with day-set t_j^{days} if the two sets have any days in common. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \qquad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{days}} \cap t_i^{\text{days}} \neq \emptyset}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \qquad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{days}} \cap t_i^{\text{days}} \neq \emptyset}} y_{c_j,t_j} - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.6 SameWeeks

The `SameWeeks` constraint says that if a class c_i is assigned time t_i, with week-set t_i^{weeks}, then c_j cannot be assigned a time with week-set t_j^{weeks} if the smaller of the week-sets is not included in the larger. Here we have $p \in \{0,1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{weeks}} \not\subseteq t_j^{\text{weeks}} \wedge \\ t_j^{\text{weeks}} \not\subseteq t_i^{\text{weeks}}}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{weeks}} \not\subseteq t_j^{\text{weeks}} \wedge \\ t_j^{\text{weeks}} \not\subseteq t_i^{\text{weeks}}}} y_{c_j,t_j} - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.7 DifferentWeeks

The `DifferentWeeks` constraint says that if a class c_i is assigned time t_i, with week-set t_i^{weeks}, then c_j cannot be assigned a time with day-set t_j^{weeks} if the two sets have any weeks in common. Here we have $p \in \{0,1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{weeks}} \cap t_j^{\text{weeks}} \neq \emptyset}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{weeks}} \cap t_j^{\text{weeks}} \neq \emptyset}} y_{c_j,t_j} - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.8 Overlap

The `Overlap` constraint says that if a class c_i is assigned time t_i, then c_j cannot be assigned a time that does **not** overlap t_i. Here we have $p \in \{0,1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg((t_j^{\text{start}} < t_i^{\text{end}}) \wedge \\ (t_i^{\text{start}} < t_j^{\text{end}}) \wedge \\ (t_i^{\text{days}} \cap t_j^{\text{days}} \neq \emptyset) \wedge \\ (t_i^{\text{weeks}} \cap t_j^{\text{weeks}} \neq \emptyset))}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg((t_j^{\text{start}} < t_i^{\text{end}}) \wedge \\ (t_i^{\text{start}} < t_j^{\text{end}}) \wedge \\ (t_i^{\text{days}} \cap t_j^{\text{days}} \neq \emptyset) \wedge \\ (t_i^{\text{weeks}} \cap t_j^{\text{weeks}} \neq \emptyset))}} y_{c_j,t_j} - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.9 NotOverlap

The `NotOverlap` constraint says that if a class c_i is assigned time t_i, then c_j cannot be assigned a time that overlaps t_i. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ (t_j^{\text{start}} < t_i^{\text{end}}) \wedge \\ (t_i^{\text{start}} < t_j^{\text{end}}) \wedge \\ (t_i^{\text{days}} \cap t_i^{\text{days}} \neq \emptyset) \wedge \\ (t_i^{\text{weeks}} \cap t_i^{\text{weeks}} \neq \emptyset)}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ ((t_j^{\text{start}} < t_i^{\text{end}}) \wedge \\ (t_i^{\text{start}} < t_j^{\text{end}}) \wedge \\ (t_i^{\text{days}} \cap t_i^{\text{days}} \neq \emptyset) \wedge \\ (t_i^{\text{weeks}} \cap t_i^{\text{weeks}} \neq \emptyset)}} y_{c_j,t_j} - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.10 SameRoom

The `SameRoom` constraint says that if a class c_i is assigned room r_i, then another class c_j cannot be assigned another room. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \quad w_{c_i,r_i} + \sum_{r_j \in \mathcal{R}_{c_j} \setminus \{r_i\}} w_{c_j,r_j} \leq 1 \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, r_i \in \mathcal{R}_{c_i}$$

$$\text{Soft:} \quad w_{c_i,r_i} + \sum_{r_j \in \mathcal{R}_{c_j} \setminus \{r_i\}} w_{c_j,r_j} - 1 \leq p \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j \in \mathcal{C}_\delta, r_i \in \mathcal{R}_{c_i}$$

4.2.11 DifferentRoom

The `DifferentRoom` constraint says that if a class c_i is assigned room r_i, then another class c_j cannot be assigned the same room. Here we have $p \in \{0,1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \qquad w_{c_i,r} + w_{c_j,r} \leq 1 \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, c_j \in \mathcal{C}_\delta, r \in \mathcal{R}_{c_i} \cup \mathcal{R}_{c_j}$$

$$\text{Soft:} \qquad w_{c_i,r} + w_{c_j,r} - 1 \leq p \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, c_j \in \mathcal{C}_\delta, r \in \mathcal{R}_{c_i} \cup \mathcal{R}_{c_j}$$

4.2.12 SameAttendees

The `SameAttendees` constraint says that if a class c_i is scheduled at time t_i in room r_i, then another class c_j cannot be scheduled such that the times overlap (like the `Overlap` constraint) but also not such that the the classes overlap in a time-room sense. That means that the classes must be scheduled such that the travel time between the two assigned rooms does not exceed the duration between the two assigned times. Here we have $p \in \{0,1\}$ and $c_p = c_\delta$.

$$\text{Hard: } x_{c_i,t_i,r_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i.\texttt{Overlap}(t_j)}} y_{c_j,t_j} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg t_i.\texttt{Overlap}(t_j), \\ r_j \in \mathcal{R}_{c_j}: \\ t_i.\texttt{Overlap}(t_j, r_j, r)}} x_{c_j,t_j,r_j} \leq 1 \ \forall c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in t_{c_i}, r_i \in \mathcal{R}_{c_i}$$

$$\text{Soft: } x_{c_i,t_i,r_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i.\texttt{Overlap}(t_j)}} y_{c_j,t_j} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg t_i.\texttt{Overlap}(t_j), \\ r_j \in \mathcal{R}_{c_j}: \\ t_i.\texttt{Overlap}(t_j, r_j, r)}} x_{c_j,t_j,r_j} - 1 \leq p \ \forall c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in t_{c_i}, r_i \in \mathcal{R}_{c_i}$$

4.2.13 Precedence

The `Precedence` constraint says that if a class c_i is assigned time t_i, then another class c_j cannot be assigned a time that starts in an earlier week or on an earlier day of the week (if they start in the same week) or on an earlier time (if they start in the same week on the same day). Here we have $p \in \{0,1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \qquad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_j^{\text{weeks.first}} < t_i^{\text{weeks.first}} \vee \\ (t_j^{\text{weeks.first}} = t_i^{\text{weeks.first}} \wedge \\ (t_j^{\text{days.first}} < t_i^{\text{days.first}} \vee \\ (t_j^{\text{days.first}} = t_i^{\text{days.first}} \wedge t_j^{\text{start}} < t_i^{\text{start}})))}} y_{c_j,t_j} \leq 1 \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_j^{\text{weeks.first}} < t_i^{\text{weeks.first}} \vee \\ (t_j^{\text{weeks.first}} = t_i^{\text{weeks.first}} \wedge \\ (t_j^{\text{days.first}} < t_i^{\text{days.first}} \vee \\ (t_j^{\text{days.first}} = t_i^{\text{days.first}} \wedge t_j^{\text{start}} < t_i^{\text{start}})))}} y_{c_j,t_j} - 1 \leq p \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.14 WorkDay(S)

The `WorkDay(S)` constraint says that if a class c_i is assigned time t_i, then another class c_j cannot be assigned a time that overlaps any week and any day such that the time difference between earliest start time and latest end time is greater than the parameter `S`. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{weeks}} \cap t_i^{\text{weeks}} \neq \emptyset \wedge \\ t_i^{\text{days}} \cap t_i^{\text{days}} \neq \emptyset \wedge \\ \max(t_i^{\text{end}}, t_j^{\text{end}}) - \min(t_i^{\text{start}}, t_j^{\text{start}}) > \mathtt{S}}} y_{c_j,t_j} \leq 1 \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i^{\text{weeks}} \cap t_i^{\text{weeks}} \neq \emptyset \wedge \\ t_i^{\text{days}} \cap t_i^{\text{days}} \neq \emptyset \wedge \\ \max(t_i^{\text{end}}, t_j^{\text{end}}) - \min(t_i^{\text{start}}, t_j^{\text{start}}) > \mathtt{S}}} y_{c_j,t_j} - 1 \leq p \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.15 MinGap(G)

The `MinGap(G)` constraint says that if a class c_i is assigned time t_i, then another class c_j cannot be assigned a time that overlaps any week and any day such that the time between the earliest end time and the latest start time is less than `G`. Here we have $p \in \{0, 1\}$ and $c_p = c_\delta$.

$$\text{Hard:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg(t_i^{\text{weeks}} \cap t_i^{\text{weeks}} = \emptyset \vee \\ t_i^{\text{days}} \cap t_i^{\text{days}} = \emptyset \vee \\ t_i^{\text{end}} + \mathtt{G} \leq t_j^{\text{start}} \vee \\ t_j^{\text{end}} + \mathtt{G} \leq t_i^{\text{start}})}} y_{c_j,t_j} \leq 1 \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

$$\text{Soft:} \quad y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg(t_i^{\text{weeks}} \cap t_i^{\text{weeks}} = \emptyset \vee \\ t_i^{\text{days}} \cap t_i^{\text{days}} = \emptyset \vee \\ t_i^{\text{end}} + \mathtt{G} \leq t_j^{\text{start}} \vee \\ t_j^{\text{end}} + \mathtt{G} \leq t_i^{\text{start}})}} y_{c_j,t_j} - 1 \leq p \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in \mathcal{T}_{c_i}$$

4.2.16 MaxDays(D)

The `MaxDays(D)` constraint says that the given classes cannot be spread over more than `D` days. We define the auxiliary variable:

$$\gamma_d = \begin{cases} 1 & \text{if any class } c \in \mathcal{C}_\delta \text{ is scheduled on day } d \in \mathcal{D} \\ 0 & \text{otherwise} \end{cases}$$

For all days $d \in \mathcal{D}$ The variable γ_d is bounded by the constraint:

$$\sum_{c \in \mathcal{C}_\delta} z_{c,d} \leq M\gamma_d \qquad \forall d \in \mathcal{D}$$

Where $M = |\mathcal{C}_\delta|$. The `MaxDays(D)` constraints are shown below, where $p \in \mathbb{Z}^+$ and $c_p = c_\delta$.

$$\text{Hard:} \qquad \sum_{d \in \mathcal{D}} \gamma_d \leq \mathtt{D}$$

$$\text{Soft:} \qquad \sum_{d \in \mathcal{D}} \gamma_d - \mathtt{D} \leq p$$

4.2.17 MaxDayload(S)

The `MaxDayload(S)` says that the given classes cannot be scheduled such that the number of time slots on any day (day load) does not exceed `S`. We define the day load $\phi_{w,d} \in \mathbb{Z}^+$ on a day d in a week w.

$$\phi_{w,d} = \sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c : t^{week} = w \wedge t^{day} = d}} t^{\text{length}} y_{c,t} \qquad \forall w \in \mathcal{W}, d \in \mathcal{D}$$

Thus the constraints are as follows.

$$\text{Hard:} \qquad \phi_{w,d} \leq \mathtt{S} \qquad \forall w \in \mathcal{W}, d \in \mathcal{D}$$

$$\text{Soft:} \qquad \phi_{w,d} - \mathtt{S} \leq \iota_{w,d} \qquad \forall w \in \mathcal{W}, d \in \mathcal{D}$$

The variable $\iota_{w,d} \in \mathbb{Z}^+$ counts the number of exceeding time slots on day d in week w. Thus the penalty for this distribution constraint is set by:

$$\frac{c_\delta}{|\mathcal{W}|} \sum_{\substack{w \in \mathcal{W}, \\ d \in \mathcal{D}}} \iota_{w,d} - 0.999 \leq p$$

The penalty should be computed using integer division. Since the division by $|\mathcal{W}|$ can result in non-integer value we subtract 0.999 to bind p correctly. As the distribution constraints cost c_δ is included in the above constraint, we have $p \in \mathbb{Z}^+$ and $c_p = 1$

4.2.18 MaxBreaks(R,S)

The `MaxBreaks(R,S)` constraint says that there can be no more than `R` breaks during a day, on any day in any week. A break is defined by having more than `S` empty timeslots between two consecutive classes. Two consecutive classes are considered to be in the same block if there is no break between them. This means that on any day d in any week w, the number of blocks $\beta_{w,d} \in \mathbb{Z}^+$ must be less than $\mathtt{R} + 1$.

$$\text{Hard:} \qquad \beta_{w,d} - 1 \leq \mathtt{R} \qquad \forall w \in \mathcal{W}, d \in \mathcal{D}$$

$$\text{Soft:} \qquad \beta_{w,d} - 1 - \mathtt{R} \leq \eta_{w,d} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}$$

The variable $\eta_{w,d} \in \mathbb{Z}^+$ counts the number of exceeding time slots on day d in week w. Thus the penalty for this distribution constraint is set by:

$$\frac{c_\delta}{|\mathcal{W}|} \sum_{\substack{w \in \mathcal{W}, \\ d \in \mathcal{D}}} \eta_{w,d} - 0.999 \leq p$$

Since the distribution constraint cost c_δ is included in the above constraint, we have $p \in \mathbb{Z}^+$ and $c_p = 1$

Constraints to control $\beta_{w,d}$

To count the number of blocks on a day d in a week w, we define the variable $\alpha_{w,d,\tau} \in \{0, 1\}$. Let $\mathrm{T}' \subseteq \mathrm{T}$ be the set of time slots where any $c \in \mathcal{C}_\delta$ can start.

$$\alpha_{w,d,\tau} = \begin{cases} 1 & \substack{\text{if a block starts in week } w \in \mathcal{W} \text{ on day } d \\ \text{at time slot } \tau \in \mathrm{T}'} \\ 0 & \text{otherwise} \end{cases}$$

and thus we have

$$\beta_{w,d} = \sum_{\tau \in \mathrm{T}'} \alpha_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}$$

To set $\alpha_{w,d,\tau}$ correctly, we need the auxiliary variable

$$\sigma_{w,d,\tau} = \begin{cases} 1 & \substack{\text{if any class } c \in \mathcal{C}_\delta \text{ starts in week } w \in \mathcal{W} \\ \text{on day } d \text{ at time slot } \tau \in \mathrm{T}'} \\ 0 & \text{otherwise} \end{cases}$$

Which is controlled by the constraints:

$$\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c : t^{\text{start}} = \tau}} y_{c,t} \geq \sigma_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}'$$

$$\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c : t^{\text{start}} = \tau}} y_{c,t} \leq M \sigma_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}'$$

Where $M = |\mathcal{C}_\delta|$. We define another auxiliary variable

$$\varepsilon_{w,d,\tau} = \begin{cases} 1 & \substack{\text{if any class } c \in \mathcal{C}_\delta \text{ is scheduled in week } w \in \mathcal{W} \\ \text{on day } d \in \mathcal{D} \text{ and overlaps any time slot } \{\tau - 1 - \mathtt{S}, \ldots, \tau - 1\}} \\ 0 & \text{otherwise} \end{cases}$$

Which is controlled by the constraints:

$$\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c : \\ t.\mathtt{Overlap}(\{\tau - 1 - \mathtt{S}, \ldots, \tau - 1\})}} y_{c,t} \geq \varepsilon_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}$$

$$\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c : \\ t.\mathtt{Overlap}(\{\tau - 1 - \mathtt{S}, \ldots, \tau - 1\})}} y_{c,t} \leq M \varepsilon_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}$$

Where $M = |\mathcal{C}_\delta|$. The auxiliary variables $\sigma_{w,d,\tau}$ and $\varepsilon_{w,d,\tau}$ sets $\alpha_{w,d,\tau}$ using the following constraints.

$$\sigma_{w,d,\tau} - \varepsilon_{w,d,\tau} \leq \alpha_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}'$$

$$\sigma_{w,d,\tau} \geq \alpha_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}'$$

$$\varepsilon_{w,d,\tau} + \alpha_{w,d,\tau} \leq 1 \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}'$$

Note that for a time slot $\bar{\tau}$ where $\varepsilon_{w,d,\bar{\tau}} = 0$ because $\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c : \\ t.\mathtt{Overlap}(\{\tau - 1 - \mathtt{S}, \ldots, \tau - 1\})}} y_{c,t}$ is fixed to 0 (there exists no $y_{c,t}$ satisfying the sum conditions), then

$$\sigma_{w,d,\bar{\tau}} = \alpha_{w,d,\bar{\tau}} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \bar{\tau} \in \mathrm{T} : \left| \bigcap_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c : \\ t.\mathtt{Overlap}(\{\bar{\tau} - 1 - \mathtt{S}, \ldots, \bar{\tau} - 1\})}} \{y_{c,t}\} \right| = 0$$

The `Overlap`-function tells if a time t overlaps a time period, similarly to the `Overlap` distribution constraint.

4.2.19 MaxBlock(M,S)

The `MaxBlock(M,S)` says that a block on any day in any week can be no longer than `M` time slots. Two consecutive classes are said to be in the same block if there are no more than `S` time slots between them. There exist special cases where a class by itself is longer than the maximum allowed time slots `M`. It is stated that such a class cannot be in a block with another class (or if the constraint is soft, we penalize when that happens). For the following constraints, we ignore the classes with length strictly larger than `M` (which is covered later). We define the variable $\rho_{w,d,\tau} \in \{0,1\}$

$$\rho_{w,d,\tau} = \begin{cases} 1 & \text{if a block longer than } \mathtt{M} \text{ starts in week } w \in \mathcal{W} \text{ on day } d \in \mathcal{D} \text{ at time slot } \tau \\ 0 & \text{otherwise} \end{cases}$$

$$\rho_{w,d,\tau} = \begin{cases} 1 & \substack{\text{if a block longer than } \mathtt{M} \text{ starts in week } w\in\mathcal{W} \\ \text{on day } d\in\mathcal{D} \text{ at time slot } \tau\in\mathrm{T}'} \\ 0 & \text{otherwise} \end{cases}$$

Then we can define the constraints

$$\text{Hard:} \quad \sum_{\substack{w\in\mathcal{W},\\ d\in\mathcal{D},\\ \tau\in\mathrm{T}}} \rho_{w,d,\tau} = 0$$

$$\text{Soft:} \quad \frac{c_\delta}{|\mathcal{W}|} \sum_{\substack{w\in\mathcal{W},\\ d\in\mathcal{D},\\ \tau\in\mathrm{T}}} \rho_{w,d,\tau} \leq p$$

Since the distribution constraints cost c_δ is included in the above constraint, we have $p \in \mathbb{Z}^+$ and $c_p = 1$

Constraints to control $\rho_{w,d,\tau}$

From section 4.2.18 we have defined $\alpha_{w,d,\tau}$ (any block starting at w, d, τ). Here we also need variable $\gamma_{w,d,\tau}$ defined as

$$\gamma_{w,d,\tau} = \begin{cases} 1 & \substack{\text{if a block ends in week } w \in \mathcal{W} \text{ on day } d \in \mathcal{D} \\ \text{at time slot } \tau\in\mathrm{T}''} \\ 0 & \text{otherwise} \end{cases}$$

We have that

$$\rho_{w,d,\tau} \leq \alpha_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}$$

$$\mathtt{M}\rho_{w,d,\tau} + \mathtt{S} \sum_{\bar{\tau}\in\mathrm{T}:\tau<\bar{\tau}\leq\tau+\mathtt{M}} \gamma_{w,d,\bar{\tau}} \leq \mathtt{M} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}$$

$$\alpha_{w,d,\tau} - \sum_{\bar{\tau}\in\mathrm{T}:\tau<\bar{\tau}\leq\tau+\mathtt{M}} \gamma_{w,d,\bar{\tau}} \leq \rho_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}$$

To control $\gamma_{w,d,\tau}$, we need two auxiliary variables. Let $\mathrm{T}'' \subseteq \mathrm{T}$ be the set of time slots where any $c \in \mathcal{C}_\delta$ can end.

$$\varphi_{w,d,\tau} = \begin{cases} 1 & \text{if any class } c \in \mathcal{C}_\delta \text{ ends in week } w \in \mathcal{W} \text{ on day } d \in \mathcal{D} \text{ at time slot } \tau \in \mathrm{T}'' \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c: \\ t^{\text{end}} = \tau}} y_{c,t} \geq \varphi_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}''$$

$$\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c: \\ t^{\text{end}} = \tau}} y_{c,t} \leq M\varphi_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}''$$

Where $M = |\mathcal{C}_\delta|$. We define the variable $\theta_{w,d,\tau}$

$$\theta_{w,d,\tau} = \begin{cases} 1 & \text{if a class } c \in \mathcal{C}_\delta \text{ is scheduled in week } w \in \mathcal{W} \text{ on day } d \in \mathcal{D} \text{ and overlaps } \{\tau+1, \ldots, \tau+1+\mathtt{S}\} \\ 0 & \text{otherwise} \end{cases}$$

and the constraints

$$\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c: \\ t.\mathtt{Overlaps}(\tau+1, \ldots, \tau+1+\mathtt{S})}} y_{c,t} \geq \theta_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}''$$

$$\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c: \\ t.\mathtt{Overlaps}(\tau+1, \ldots, \tau+1+\mathtt{S})}} y_{c,t} \leq M\theta_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}''$$

We can now define the constraints on $\gamma_{w,d,\tau}$.

$$\varphi_{w,d,\tau} - \theta_{w,d,\tau} \leq \gamma_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}''$$

$$\varphi_{w,d,\tau} \geq \gamma_{w,d,\tau} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}''$$

$$\theta_{w,d,\tau} + \gamma_{w,d,\tau} \leq 1 \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \tau \in \mathrm{T}''$$

Note that for a time slot $\bar{\tau}$ where $\theta_{w,d,\bar{\tau}} = 0$ because $\sum_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c: \\ t.\mathtt{Overlap}(\{\tau+1, \ldots, \tau+1+\mathtt{S}\})}} y_{c,t}$ is fixed to 0 (there exists no $y_{c,t}$ satisfying the sum conditions), then

$$\varphi_{w,d,\bar{\tau}} = \gamma_{w,d,\bar{\tau}} \qquad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, \bar{\tau} \in \mathrm{T} : \left| \bigcap_{\substack{c \in \mathcal{C}_\delta, \\ t \in \mathcal{T}_c: \\ t.\mathtt{Overlap}(\{\bar{\tau}+1, \ldots, \bar{\tau}+1+\mathtt{S}\})}} \{y_{c,t}\} \right| = 0$$

Special case
A special case is where a class length is longer than $\mathtt{M}$. We define the set of

such classes as $\mathcal{C}_\delta^{>\mathtt{M}}$. For the hard constraint, we know that if a class $c \in \mathcal{C}_\delta^{>\mathtt{M}}$ is scheduled in a specific time, then any other class from $\mathcal{C}_\delta$ cannot use a time that will place the two in the same block. It implies that if the times share at least one week and at least one day, there cannot be less than $\mathtt{S}$ time slots between them.

$$\text{Hard:} \qquad y_{c_i,t_i} + \sum_{\substack{c_j \in \mathcal{C}_\delta \setminus \{c_i\}, \\ t_j \in \mathcal{T}_{c_j}: \\ t^{\text{weeks}} \cap t_j^{\text{weeks}} \neq \emptyset \wedge \\ t^{\text{days}} \cap t_j^{\text{days}} \neq \emptyset \wedge \\ (t_i^{\text{start}} - t_j^{\text{end}} < \mathtt{S} \vee \\ t_j^{\text{start}} - t_i^{\text{end}} < \mathtt{S})}} y_{c_j,t_j} \quad \leq 1 \qquad \forall\, c_i \in \mathcal{C}_\delta^{>\mathtt{M}}, t \in \mathcal{T}_{c_i}$$

For the soft constraint we need to set the $\rho_{w,d,\tau}$ variable if the time between the class $c_i \in \mathcal{C}_\delta^{>\mathtt{M}}$ and another class $c_j \in \mathcal{C}_\delta \setminus \{c_i\}$ is less than $\mathtt{S}$. That means that we must have a constraint for each week and day of any time of the class c_i.

$$\text{Soft:} \quad y_{c_i,t_i} + M \sum_{\substack{c_j \in \mathcal{C}_\delta \setminus \{c_i\}, \\ t_j \in \mathcal{T}_{c_j}: \\ w \in t_j^{\text{weeks}} \wedge \\ d \in t_j^{\text{days}} \wedge \\ (t_i^{\text{start}} - t_j^{\text{end}} < \mathtt{S} \vee \\ t_j^{\text{start}} - t_i^{\text{end}} < \mathtt{S})}} y_{c_j,t_j} \ \leq \rho_{w,d,t^{\text{start}}} \qquad \forall c_i \in \mathcal{C}_\delta^{>\mathtt{M}}, t \in \mathcal{T}_{c_i}, w \in t^{\text{weeks}}, d \in t^{\text{days}}$$

4.3 Student sectioning

Recall that $e_{s,c}$ is the decision variable that tells if student s is attending class c.

The number of students attending any class cannot exceed the limitation.

$$\sum_{s\in\mathcal{S}_c} e_{s,c} \leq c^{\text{limit}} \qquad \forall\, c \in \mathcal{C}$$

If a student attends a class given a parent class, the parent class must also be attended. Equality does not hold since classes can share the same parent.

$$e_{s,c_i} \leq e_{s,c_j} \qquad \forall\, s \in \mathcal{S}, c \in \mathcal{C}_s : c_i^{\text{parent}} = c_j$$

4.3.1 Students attending courses

For courses with only one configuration, the students who must attend the course must attend exactly one class from each configuration subparts.

$$\sum_{c\in\mathcal{C}_\zeta} e_{s,c} = 1 \qquad \forall\, k \in \mathcal{K}_s : |\Omega_k| = 1, \omega \in \Omega_k, \zeta \in \mathrm{Z}_\omega, s \in \mathcal{S}_k$$

For courses that have more than one configuration, we define the auxiliary variable $b_{s,\omega} \in \{0, 1\}$.

$$b_{s,\omega} = \begin{cases} 1 & \text{if student } s \in \mathcal{S} \text{ is attending a class in configuration } \omega \in \Omega_k \\ 0 & \text{otherwise} \end{cases}$$

The students must attend the courses by attending exactly one of the configurations.

$$\sum_{\omega\in\Omega_k} b_{s,\omega} = 1 \qquad \forall\, s \in \mathcal{S}, k \in \mathcal{K}_s : |\Omega_k| > 1$$

To attend a configuration, the students must attend exactly one class from each of the configuration subparts. Furthermore, if a student is not attending a configuration, no classes from the subparts of that configuration can be attended.

$$\sum_{c\in\mathcal{C}_\zeta} e_{s,c} = b_{s,\omega} \qquad \forall\, k \in \mathcal{K}_s : |\Omega_k| > 1, \omega \in \Omega_k, \zeta \in \mathrm{Z}_\omega, s \in \mathcal{S}_k$$

4.3.2 Student conflicts

For student conflicts we define the variable $\chi_{s,c_i,c_j} \in \{0, 1\}$

$$\chi_{s,c_i,c_j} = \begin{cases} 1 & \text{if there is a student conflict for student } s \in \mathcal{S} \text{ between classes } c_i \in \text{ and } c_j \in \mathcal{C}_s \\ 0 & \text{otherwise} \end{cases}$$

$$\chi_{s,c_i,c_j} = \begin{cases} 1 & \substack{\text{if there is a student conflict for student } s \in \mathcal{S} \\ \text{between classes } c_i \in \mathcal{C}_s \text{ and } c_j \in \mathcal{C}_s} \\ 0 & \text{otherwise} \end{cases}$$

This variable is dependent on variables f_{s,c_i,c_j} and o_{c_i,c_j}.

$$f_{s,c_i,c_j} = \begin{cases} 1 & \text{if student } s \in \mathcal{S} \text{ is attending both class } c_i \in \mathcal{C}_s \text{ and } c_j \in \mathcal{C}_s \\ 0 & \text{otherwise} \end{cases}$$

$$o_{c_i,c_j} = \begin{cases} 1 & \text{if classes } c_i \in \mathcal{C} \text{ and } c_j \in \mathcal{C} \text{ overlaps} \\ 0 & \text{otherwise} \end{cases}$$

Both have to be 1 for a student conflict to occur

$$o_{c_i,c_j} + f_{s,c_i,c_j} - 1 \leq \chi_{s,c_i,c_j} \qquad \forall\, s \in \mathcal{S}, (c_i, c_j) \in \mathcal{C}_s : i < j$$

Controlling the auxiliary variables

The variable f_{s,c_i,c_j} is dependent on the variables $e_{s,c}$

$$e_{s,c_i} + e_{s,c_j} - 1 \leq f_{s,c_i,c_j} \qquad \forall\, s \in \mathcal{S}, (c_i, c_j) \in \mathcal{C}_s : i < j$$

$$e_{s,c_i} \geq f_{s,c_i,c_j} \qquad \forall\, s \in \mathcal{S}, (c_i, c_j) \in \mathcal{C}_s : i < j$$

$$e_{s,c_j} \geq f_{s,c_i,c_j} \qquad \forall\, s \in \mathcal{S}, (c_i, c_j) \in \mathcal{C}_s : i < j$$

The variable o_{c_i,c_j} is dependent on the time and room of both classes. It is similar to the `SameAttendees` distribution constraint, but here it is split into two types of constraints, one for the times that overlap and one for the times that do not overlap, but the assigned rooms cause an overlap.

$$y_{c_i,t_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ t_i\texttt{.Overlap}(t_j)}} y_{c_j,t_j} - 1 \leq o_{c_i,c_j} \qquad \forall\, c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in t_{c_i}$$

$$x_{c_i,t_i,r_i} + \sum_{\substack{t_j \in \mathcal{T}_{c_j}: \\ \neg t_i\texttt{.Overlap}(t_j), \\ r_j \in \mathcal{R}_{c_j}: \\ t_i\texttt{.Overlap}(t_j,r_j,r_i)}} x_{c_j,t_j,r_j} - 1 \leq o_{c_i,c_j} \qquad \forall c_i, c_j \in \mathcal{C}_\delta : i < j, t_i \in t_{c_i}, r_i \in \mathcal{R}_{c_i}$$

5 Objectives

There are four categories of objectives: time, room, distribution constraints, and student conflicts. These categories have respective weights ψ_t, ψ_r, ψ_δ, and ψ_s that prioritize the types of objectives compared to each other.

5.1 Time

The time objective is related to a penalty on a class c being scheduled at a specific time t.

$$\psi_t \sum_{\substack{c \in \mathcal{C}, \\ t \in \mathcal{T}_c}} p_{c,t} y_{c,t}$$

5.2 Room

The time objective is related to a penalty on a class c being assigned a specific room r.

$$\psi_r \sum_{\substack{c \in \mathcal{C}, \\ r \in \mathcal{R}_c}} p_{c,r} w_{c,r}$$

5.3 Distribution constraint

Recall that each soft distribution constraint defined a penalty variable p, which was written without dimensions and a penalty cost c_p. We have the set of all penalty variables $\mathcal{P}$.

$$\psi_\delta \sum_{p \in \mathcal{P}} c_p p$$

5.4 Student conflicts

Each student conflict has a penalty of 1; thus, we penalize the total number of student conflicts.

$$\psi_s \sum_{\substack{s \in \mathcal{S}, \\ (c_i, c_j) \in \mathcal{C}_s}} \chi_{s,c_i,c_j}$$

6 Results

The results consist of two parts. Section 6.1 presents the sizes of the MIP models in number of constraints and variables. Section 6.2 presents the performance of solving the MIP.
The MIP is solved using Gurobi 9.0 with 8 threads on a 64bit computer running Scientific Linux 7.7. The machine is equipped with two Intel Xeon E5-2650 v4 CPUs clocked at 2.20GHz and 256GB of RAM.
The 30 instances from the ITC2019 competition are used to test the MIP. There is no data available for *pu-proj-fal19* as we could not construct the MIP model because of a lack of memory.

6.1 Size of the MIP

The total number of constraints and variables are shown in Table 4. Appendix B gives specific details on the number of constraints and variables used to model the different parts of the MIP.

Instance	**Constraints**	**Variables**
agh-fis-spr17	10,716,234	5,270,957
agh-ggis-spr17	10,971,106	10,532,658
bet-fal17	7,427,922	4,770,806
iku-fal17	11,440,488	3,772,222
mary-spr17	1,716,685	1,287,460
muni-fi-spr16	4,958,093	4,350,531
muni-fsps-spr17	1,003,857	717,147
muni-pdf-spr16c	26,203,626	6,922,578
pu-llr-spr17	8,980,995	5,542,707
tg-fal17	597,723	232,749
agh-ggos-spr17	15,912,008	3,523,533
agh-h-spr17	12,642,901	1,881,442
lums-spr18	589,187	458,344
muni-fi-spr17	6,475,267	5,711,500
muni-fsps-spr17c	17,717,840	1,466,645
muni-pdf-spr16	10,554,793	6,901,019
nbi-spr18	1,955,308	631,823
pu-d5-spr17	31,570,758	30,624,014
pu-proj-fal19		
yach-fal17	4,407,666	1,336,902
agh-fal17	45,591,515	25,795,085
bet-spr18	10,006,170	6,165,343
iku-spr18	9,617,259	3,728,533
lums-fal17	525,567	448,222
mary-fal18	3,754,624	3,307,703
muni-fi-fal17	8,032,513	7,384,314
muni-fspsx-fal17	23,884,170	2,791,093
muni-pdfx-fal17	48,957,509	22,345,083
pu-d9-fal19	115,664,227	56,683,344
tg-spr18	645,009	130,533

Table 4 Total number of constraints and variables for each instance

6.2 Solving the MIP

The performance is tested with a 24 hour time limit, including reading and processing the data. Table 5 shows the performance of the MIP after 1 hour and 24 hours.

Time	**1 hour**			**24 hours**		
Instance	UB	LB	Gap	UB	LB	Gap
agh-fis-spr17	-	-	-	-	779	100%
agh-ggis-spr17	-	-	-	-	21,703	100%
bet-fal17	-	-	-	-	-	-
iku-fal17	-	-	-	-	10,947	100%
mary-spr17	-	2,435	100%	15,932	13,212	17.07%
muni-fi-spr16	-	-	-	-	3,276	100%
muni-fsps-spr17	-	851	100%	**868**	868	0%
muni-pdf-spr16c	-	-	-	-	9,196	100%
pu-llr-spr17	-	-	-	10,710	9,683	9.59%
tg-fal17	**4,215**	4,215	0%	**4,215**	4,215	0%
agh-ggos-spr17	-	-	-	-	-	-
agh-h-spr17	-	-	-	-	5	100%
lums-spr18	-	1	100%	95	24	74.74%
muni-fi-spr17	-	-	-	24,572	2,056	91.63%
muni-fsps-spr17c	-	-	-	-	923	100%
muni-pdf-spr16	-	-	-	-	-	-
nbi-spr18	18,979	17,438	8.12%	18,212	17,654	3.06%
pu-d5-spr17	-	-	-	-	4,147	100%
pu-proj-fal19						
yach-fal17	-	30	100%	19,046	516	97.29%
agh-fal17	-	-	-	-	-	-
bet-spr18	-	-	-	-	-	-
iku-spr18	-	-	-	-	14,006	100%
lums-fal17	-	196	100%	405	233	42.47%
mary-fal18	-	-	-	-	3,009	100%
muni-fi-fal17	-	-	-	-	1,486	100%
muni-fspsx-fal17	-	-	-	-	1,680	100%
muni-pdfx-fal17	-	-	-	-	-	-
pu-d9-fal19	-	-	-	-	-	-
tg-spr18	**12,704**	12,704	0%	**12,704**	12,704	0%

Table 5 Runtime stats after 1 hour and 24 hours. Optimal solutions are in bold. Dash means that the solver is running, but there was no value.

When solving the MIP, Gurobi will first try to reduce the MIP by a presolve procedure. The presolve procedure will try to make the MIP smaller and easier to solve. In Table 6, we present the number of constraints and variables removed by presolve and the size of the model after presolve. The Gurobi Presolve parameter is left at its default setting.

Instance	Removed constraints	Removed variables	Number of constraints after presolve	Number of variables after presolve	Time (sec)
agh-fis-spr17	3,303,883	2,688,938	7,412,351	2,582,019	2,956
agh-ggis-spr17	9,004,781	8,571,540	1,966,325	1,961,118	239
bet-fal17	1,208,055	788,486	6,219,867	3,982,320	680
iku-fal17	2,040,648	478,300	9,399,840	3,293,922	2,662
mary-spr17	942,132	761,533	774,553	525,927	62
muni-fi-spr16	1,843,183	1,644,797	3,114,910	2,705,734	100
muni-fsps-spr17	471,269	349,566	532,588	367,581	34
muni-pdf-spr16c	9,263,603	3,277,787	16,940,023	3,644,791	4,825
pu-llr-spr17	4,288,999	3,652,660	4,691,996	1,890,047	289
tg-fal17	477,409	106,435	120,314	126,314	22
agh-ggos-spr17	2,557,583	873,066	13,354,425	2,650,467	1,413
agh-h-spr17	1,830,684	295,957	10,812,217	1,585,485	5,379
lums-spr18	101,420	22,924	487,767	435,420	189
muni-fi-spr17	2,050,479	1,833,495	4,424,788	3,878,005	150
muni-fsps-spr17c	4,574,019	604,346	13,143,821	862,299	1,497
muni-pdf-spr16	2,739,129	2,298,089	7,815,664	4,602,930	698
nbi-spr18	734,527	431,815	1,220,781	200,008	52
pu-d5-spr17	23,427,922	22,939,347	8,142,836	7,684,667	473
pu-proj-fal19					
yach-fal17	1,086,201	326,048	3,321,465	1,010,854	194
agh-fal17	14,037,069	12,331,961	31,554,446	13,463,124	11,506
bet-spr18	1,662,060	971,980	8,344,110	5,193,363	660
iku-spr18	2,666,485	780,152	6,950,774	2,948,381	2,216
lums-fal17	109,252	16,101	416,315	432,121	134
mary-fal18	1,438,431	1,294,436	2,316,193	2,013,267	92
muni-fi-fal17	2,625,159	2,433,535	5,407,354	4,950,779	154
muni-fspsx-fal17	5,813,525	1,197,584	18,070,645	1,593,509	2,860
muni-pdfx-fal17	22,427,991	14,035,002	26,529,518	8,310,081	11,008
pu-d9-fal19	39,699,909	32,863,017	75,964,318	23,820,327	5,970
tg-spr18	573,696	58,854	71,313	71,679	29

Table 6 The number constraints and variable removed by Gurobi presolve and the MIP model sizes after presolve. Last column is the time spend in presolve.

7 Concluding remarks

We have presented a MIP formulation for the ITC2019 problem. This MIP formulation results in models with a vast number of constraints and variables. The performance was tested on the 30 instances of ITC2019 resulting in solutions to ten instances, including three optimal solutions. The results also provide lower bounds on 22 of the instances. One instance reached the memory limit and thus was not able to be computed. The results are, of course, dependant on different Gurobi parameter settings. For example, one could wish to use a more aggressive presolve strategy and a MIP focus parameter set to *feasibility* or *optimality*. Such settings might provide more reductions in presolve and might also provide feasible solutions to more instances with worse lower bounds as a consequence. However, before such parameter tuning is performed, it should be considered if the model can be formulated differently. Table 6 shows the number of constraints and variables removed by presolve. For some instances, Gurobi can remove numerous constraints and variables in very little time. This might indicate that the MIP formulation contains many redundant constraints and/or variables.

This basic MIP formulation can provide good solutions to smaller instances within a reasonable time. There is no time limit of the ITC2019, but the final data instances are released 10 days before the deadline. It is not expected that the MIP models will perform significantly better by increasing the time limit to 10 days. We believe that it is likely that an improved formulation of the MIP or a matheuristic based on the MIP can outperform these results.

References

Müller, T., Rudová, H., and Müllerová, Z. University course timetabling and International Timetabling Competition 2019. In Burke, E. K., Di Gaspero, L., McCollum, B., Musliu, N., and Özcan, E., editors, *Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2018)*, pages 5–31, 2018.

Appendices

A The `Overlap` function

The `Overlap` function takes different input parameters and gives a boolean result if the time and the input parameters overlap.

The different types

t.`Overlap`$(\{\tau_{\min}, \ldots, \tau_{\max}\})$
- Returns *true* if the time slots of t overlaps the time slots from $\tau_{\min}$ to $\tau_{\max}$.

$$\tau_{\min} < t^{\text{end}} \quad \wedge \quad t^{\text{start}} < \tau_{\max}$$

t.`Overlap`$(\bar{t})$
- Returns *true* if the times t and $\bar{t}$ have at least one week and one day in common and the time slots overlaps.

$$\begin{aligned} t^{\text{weeks}} \cap \bar{t}^{\text{weeks}} \neq \emptyset \quad & \wedge \\ t^{\text{days}} \cap \bar{t}^{\text{days}} \neq \emptyset \quad & \wedge \\ \bar{t}^{\text{start}} < t^{\text{end}} \quad & \wedge \quad t^{\text{start}} < \bar{t}^{\text{end}} \end{aligned}$$

t.`Overlap`$(\bar{t}, \bar{r}, r)$ - Returns *true* if the times t and $\bar{t}$ have at least one week and one day in common and the time slots overlaps or it is not possible to get from one room to the other in the time difference.

$$\begin{aligned} t^{\text{weeks}} \cap \bar{t}^{\text{weeks}} \neq \emptyset \quad & \wedge \\ t^{\text{days}} \cap \bar{t}^{\text{days}} \neq \emptyset \quad & \wedge \\ [\bar{t}^{\text{start}} < t^{\text{end}} \quad & \wedge \quad t^{\text{start}} < \bar{t}^{\text{end}} \quad \vee \\ \max\{t^{\text{start}}, \bar{t}^{\text{start}}\} - \min\{t^{\text{end}}, \bar{t}^{\text{end}}\} & < \texttt{distance}(\bar{r}, r)] \end{aligned}$$

B Specified MIP size

Tables 7-11 gives the specified number of constraints used to model each of the distribution constraint types. Table 12 gives the number of constraint and variable used for student sectioning.

Instance	**SameStart**	**SameTime**	**SameDays**	**SameWeeks**	**SameRoom**
agh-fis-spr17	0 | 0	6,002 | 341	6,002 | 9,657	731 | 0	68 | 51
agh-ggis-spr17	0 | 0	8,281 | 397	15,871 | 21,048	0 | 0	1,094 | 126
bet-fal17	64 | 0	37 | 0	198 | 55,680	0 | 0	333 | 510
iku-fal17	208 | 0	3,324 | 0	13,628 | 1,714	0 | 0	4,959 | 1,677
mary-spr17	0 | 0	164 | 0	659 | 875	0 | 0	159 | 57
muni-fi-spr16	0 | 0	628 | 0	1,253 | 278	84 | 0	149 | 9
muni-fsps-spr17	0 | 0	2,308 | 1,192	2,558 | 1,222	0 | 0	116 | 15
muni-pdf-spr16c	0 | 0	3,626 | 573	3,904 | 0	0 | 0	132 | 28,555
pu-llr-spr17	1,736 | 0	261 | 0	296 | 563	0 | 0	427 | 277
tg-fal17	0 | 0	833 | 0	833 | 6,967	0 | 0	0 | 0
agh-ggos-spr17	0 | 0	19,938 | 0	21,962 | 516	44 | 0	789 | 1
agh-h-spr17	0 | 0	32,138 | 616	32,218 | 102,452	54 | 0	271 | 27
lums-spr18	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0
muni-fi-spr17	0 | 0	300 | 0	1,057 | 65	162 | 0	141 | 11
muni-fsps-spr17c	0 | 0	1,056 | 0	1,182 | 0	846 | 234	0 | 1,489
muni-pdf-spr16	0 | 0	1,723 | 40	2,289 | 5,891	0 | 0	619 | 2,387
nbi-spr18	0 | 0	947 | 0	992 | 275	0 | 0	0 | 0
pu-d5-spr17	4 | 0	75 | 178	2,016 | 323	0 | 0	671 | 1,965
pu-proj-fal19					
yach-fal17	0 | 0	0 | 0	23 | 0	0 | 0	24 | 1,145
agh-fal17	0 | 0	44,698 | 65,737	51,903 | 153,694	1,592 | 12	3,434 | 2,782
bet-spr18	2 | 0	316 | 0	831 | 69,564	0 | 0	249 | 1,417
iku-spr18	210 | 0	3,972 | 0	14,822 | 50	0 | 0	5,518 | 0
lums-fal17	0 | 0	77 | 0	118 | 0	0 | 0	0 | 0
mary-fal18	0 | 0	115 | 0	408 | 206	0 | 0	354 | 46
muni-fi-fal17	0 | 0	388 | 0	788 | 448	0 | 0	289 | 0
muni-fspsx-fal17	0 | 0	615 | 0	1,987 | 1,372	1,320 | 0	69 | 2,235
muni-pdfx-fal17	0 | 0	7,571 | 655	12,135 | 7,439	392 | 444	898 | 42,685
pu-d9-fal19	1,512 | 0	1,022 | 0	5,413 | 4,965	0 | 0	5,149 | 8,068
tg-spr18	0 | 0	180 | 17,591	350 | 17,591	0 | 0	1,081 | 3,871

Table 7 Number of constraints in the MIP to model each of the distribution constraint types. Data represented Hard | Soft.

Instance	DifferentTime	DifferentDays	DifferentWeeks	DifferentRoom
agh-fis-spr17	0 | 0	478 | 210	0 | 0	0 | 0
agh-ggis-spr17	0 | 0	78 | 14	0 | 0	0 | 0
bet-fal17	3 | 0	8,154 | 54,691	0 | 0	0 | 431
iku-fal17	0 | 0	105 | 0	0 | 0	0 | 0
mary-spr17	0 | 0	0 | 0	0 | 0	0 | 0
muni-fi-spr16	0 | 0	171 | 225	0 | 0	0 | 0
muni-fsps-spr17	0 | 0	50 | 0	0 | 0	0 | 0
muni-pdf-spr16c	0 | 0	0 | 0	178,622 | 1,512	0 | 0
pu-llr-spr17	38 | 19	77 | 51	0 | 0	0 | 0
tg-fal17	0 | 0	0 | 331	0 | 0	0 | 0
agh-ggos-spr17	0 | 0	25 | 0	0 | 0	0 | 0
agh-h-spr17	0 | 0	0 | 0	0 | 0	0 | 0
lums-spr18	0 | 0	0 | 0	0 | 0	0 | 0
muni-fi-spr17	0 | 0	146 | 162	0 | 0	8 | 0
muni-fsps-spr17c	0 | 0	0 | 0	118,192 | 0	0 | 0
muni-pdf-spr16	0 | 0	664 | 1,176	116 | 0	0 | 0
nbi-spr18	0 | 0	1,333 | 40	0 | 0	0 | 0
pu-d5-spr17	0 | 0	2,303 | 0	0 | 0	0 | 0
pu-proj-fal19				
yach-fal17	0 | 0	12,427 | 0	0 | 0	0 | 0
agh-fal17	0 | 0	1,101 | 0	0 | 0	0 | 0
bet-spr18	0 | 0	11,507 | 68,076	0 | 0	0 | 1,924
iku-spr18	0 | 0	1,117 | 0	0 | 0	0 | 0
lums-fal17	0 | 0	0 | 0	0 | 0	0 | 0
mary-fal18	0 | 0	0 | 0	0 | 0	0 | 0
muni-fi-fal17	0 | 0	106 | 272	0 | 0	0 | 0
muni-fspsx-fal17	0 | 0	8 | 60	156,245 | 0	0 | 0
muni-pdfx-fal17	0 | 0	582 | 357	225,539 | 0	0 | 0
pu-d9-fal19	467 | 28	47 | 68	0 | 0	0 | 0
tg-spr18	0 | 0	1,359 | 0	0 | 0	0 | 0

Table 8 Number of constraints in the MIP to model each of the distribution constraint types. Data represented Hard | Soft.

Instance	**Overlap**	**NotOverlap**	**SameAttendees**	**Precedence**
agh-fis-spr17	0 | 0	2,286 | 222	6,263,527 | 3,374	272 | 22,983
agh-ggis-spr17	0 | 0	64 | 5,427	498,878 | 0	991 | 8,125
bet-fal17	0 | 0	205 | 0	1,521,402 | 0	0 | 0
iku-fal17	1,072 | 0	539,015 | 1,043	10,604,393 | 2,920	1,571 | 21,855
mary-spr17	0 | 0	0 | 16,385	306,332 | 539,511	41 | 0
muni-fi-spr16	0 | 0	2,700 | 17,279	57,483 | 0	464 | 378
muni-fsps-spr17	0 | 0	0 | 6,215	94,501 | 0	957 | 0
muni-pdf-spr16c	0 | 0	0 | 29,354	18,099,414 | 0	91,893 | 1,265
pu-llr-spr17	0 | 0	3,111 | 5,643	37,641 | 0	0 | 126
tg-fal17	0 | 0	7,406 | 0	407,532 | 134,745	0 | 0
agh-ggos-spr17	0 | 0	0 | 0	3,161,309 | 0	989 | 67,366
agh-h-spr17	0 | 0	159,239 | 0	8,445,889 | 12	2,352 | 1,104
lums-spr18	0 | 0	108,029 | 33,772	392,946 | 0	0 | 0
muni-fi-spr17	0 | 0	3,539 | 16,670	73,599 | 0	1,208 | 84
muni-fsps-spr17c	0 | 0	0 | 9,994	2,979,413 | 0	16,321 | 0
muni-pdf-spr16	0 | 0	0 | 73,914	2,321,086 | 0	454 | 201
nbi-spr18	0 | 0	0 | 635	334,989 | 0	0 | 0
pu-d5-spr17	0 | 0	7,261 | 90,598	55,526 | 2	965 | 1,519
pu-proj-fal19				
yach-fal17	0 | 0	0 | 12,075	339,540 | 0	0 | 0
agh-fal17	0 | 0	10,151 | 1,402	10,410,795 | 234,335	6,080 | 190,573
bet-spr18	0 | 0	180 | 960	2,254,545 | 0	0 | 0
iku-spr18	0 | 0	766,819 | 0	8,521,716 | 45,806	1,799 | 17,797
lums-fal17	0 | 0	153,234 | 31,524	287,927 | 0	0 | 0
mary-fal18	0 | 0	0 | 13,907	111,775 | 11,781	198 | 0
muni-fi-fal17	0 | 0	1,901 | 12,557	53,199 | 0	421 | 1,129
muni-fspsx-fal17	0 | 0	29,711 | 26,530	3,425,746 | 11	26,794 | 532
muni-pdfx-fal17	0 | 0	372 | 72,744	25,140,403 | 2,653	115,569 | 6,390
pu-d9-fal19	0 | 0	3,937 | 78,834	418,931 | 0	76 | 759
tg-spr18	0 | 0	3,860 | 0	559,997 | 0	120 | 356

Table 9 Number of constraints in the MIP to model each of the distribution constraint types. Data represented Hard | Soft.

Instance	**WorkDay**	**MinGap**	**MaxDays**
agh-fis-spr17	$0 \mid 1,844$	$0 \mid 0$	$80 \mid 48$
agh-ggis-spr17	$6,449 \mid 2,104$	$0 \mid 0$	$0 \mid 24$
bet-fal17	$9,757 \mid 168$	$0 \mid 55,351$	$80 \mid 0$
iku-fal17	$10,027 \mid 642$	$0 \mid 0$	$0 \mid 0$
mary-spr17	$201 \mid 112$	$108 \mid 739$	$0 \mid 0$
muni-fi-spr16	$713 \mid 162$	$0 \mid 0$	$0 \mid 0$
muni-fsps-spr17	$130 \mid 8$	$0 \mid 0$	$0 \mid 0$
muni-pdf-spr16c	$28 \mid 61,001$	$0 \mid 0$	$0 \mid 0$
pu-llr-spr17	$17 \mid 459$	$0 \mid 44$	$0 \mid 0$
tg-fal17	$0 \mid 118$	$0 \mid 0$	$0 \mid 0$
agh-ggos-spr17	$1,821 \mid 75$	$0 \mid 0$	$16 \mid 16$
agh-h-spr17	$74 \mid 2,371$	$0 \mid 0$	$136 \mid 40$
lums-spr18	$0 \mid 0$	$0 \mid 0$	$0 \mid 0$
muni-fi-spr17	$638 \mid 58$	$0 \mid 0$	$8 \mid 0$
muni-fsps-spr17c	$100 \mid 0$	$0 \mid 0$	$0 \mid 0$
muni-pdf-spr16	$488 \mid 13,459$	$33 \mid 210$	$0 \mid 0$
nbi-spr18	$0 \mid 0$	$0 \mid 275$	$0 \mid 0$
pu-d5-spr17	$1,366 \mid 143$	$85 \mid 54$	$0 \mid 0$
pu-proj-fal19			
yach-fal17	$0 \mid 0$	$23 \mid 0$	$0 \mid 0$
agh-fal17	$5,127 \mid 6,424$	$16 \mid 104$	$352 \mid 216$
bet-spr18	$13,363 \mid 252$	$0 \mid 68,769$	$120 \mid 0$
iku-spr18	$11,174 \mid 0$	$0 \mid 0$	$0 \mid 0$
lums-fal17	$0 \mid 0$	$0 \mid 0$	$0 \mid 0$
mary-fal18	$202 \mid 45$	$0 \mid 116$	$0 \mid 0$
muni-fi-fal17	$391 \mid 263$	$0 \mid 0$	$16 \mid 0$
muni-fspsx-fal17	$592 \mid 52$	$0 \mid 1,364$	$0 \mid 0$
muni-pdfx-fal17	$1,069 \mid 21,761$	$0 \mid 1,132$	$0 \mid 64$
pu-d9-fal19	$3,370 \mid 1,430$	$199 \mid 95$	$0 \mid 0$
tg-spr18	$0 \mid 0$	$0 \mid 0$	$0 \mid 0$

Table 10 Number of constraints in the MIP to model each of the distribution constraint types. Data represented Hard | Soft.

Instance	MaxDayLoad	MaxBreaks	MaxBlock
agh-fis-spr17	0 | 0	0 | 93,490	0 | 0
agh-ggis-spr17	0 | 0	0 | 5,904	0 | 0
bet-fal17	32 | 0	0 | 0	749,119 | 0
iku-fal17	0 | 0	0 | 0	0 | 0
mary-spr17	0 | 0	0 | 0	0 | 0
muni-fi-spr16	114 | 0	0 | 0	0 | 14,325
muni-fsps-spr17	0 | 0	0 | 0	0 | 0
muni-pdf-spr16c	50 | 1,289	0 | 0	6,992 | 211,584
pu-llr-spr17	0 | 0	0 | 6,032	0 | 0
tg-fal17	0 | 0	0 | 0	0 | 0
agh-ggos-spr17	0 | 0	9,898 | 0	0 | 0
agh-h-spr17	546 | 71	0 | 431,581	0 | 0
lums-spr18	0 | 0	0 | 0	0 | 0
muni-fi-spr17	70 | 0	0 | 0	91,638 | 24,893
muni-fsps-spr17c	0 | 0	0 | 0	0 | 0
muni-pdf-spr16	286 | 1,541	0 | 0	26,352 | 554,275
nbi-spr18	0 | 0	0 | 0	0 | 0
pu-d5-spr17	0 | 0	41,250 | 0	0 | 0
pu-proj-fal19			
yach-fal17	2,935 | 0	0 | 0	0 | 0
agh-fal17	975 | 0	165 | 792,455	0 | 0
bet-spr18	0 | 0	0 | 0	892,608 | 0
iku-spr18	0 | 0	0 | 0	0 | 0
lums-fal17	0 | 0	0 | 0	0 | 0
mary-fal18	0 | 0	0 | 0	0 | 0
muni-fi-fal17	104 | 53	0 | 0	148,171 | 27,939
muni-fspsx-fal17	0 | 0	0 | 0	0 | 0
muni-pdfx-fal17	428 | 1,614	0 | 0	6,145 | 562,289
pu-d9-fal19	0 | 31	56,607 | 3,016	0 | 0
tg-spr18	0 | 152	0 | 0	0 | 0

Table 11 Number of constraints in the MIP to model each of the distribution constraint types. Data represented Hard | Soft.

The number of constraints and variables used to model the student sectioning is presented in Table 12. The constraints and variables are split into two categories. One category is the student sectioning part. This part considers the constraints and variables used in assigning the students to the correct classes. The other part, student conflicts, is the constraints and variables used to model when the students' assignment leads to conflict.

	Student Sectioning			**Student Conflicts**	
Instance	$e_{s,c}$	$b_{s,\omega}$	**Constraints**	**Variables**	**Constraints**
agh-fis-spr17	65,376	0	53,277	3,566,172	3,986,367
agh-ggis-spr17	116,552	0	57,723	10,061,361	10,257,182
bet-fal17	94,893	15,990	62,440	3,575,690	4,843,252
iku-fal17	-	-	-	-	-
mary-spr17	28,230	0	6,046	507,735	806,431
muni-fi-spr16	64,506	14	7,454	4,205,866	4,834,674
muni-fsps-spr17	20,667	776	7,704	644,280	867,989
muni-pdf-spr16c	80,918	5,246	56,494	3,735,530	7,157,505
pu-llr-spr17	236,980	20,353	103,598	5,108,450	8,773,805
tg-fal17	-	-	-	-	-
agh-ggos-spr17	56,818	0	45,621	2,111,511	12,415,681
agh-h-spr17	11,530	0	4,954	310,548	2,996,086
lums-spr18	-	-	-	-	-
muni-fi-spr17	70,257	60	8,373	5,504,735	6,232,852
muni-fsps-spr17c	17,190	0	10,135	973,959	14,490,356
muni-pdf-spr16	103,864	2,822	22,982	5,468,914	7,420,773
nbi-spr18	29,986	0	4,984	422,248	1,565,002
pu-d5-spr17	442,232	9,886	302,079	29,956,847	31,033,720
pu-proj-fal19					
yach-fal17	28,779	150	25,665	1,183,118	3,986,219
agh-fal17	305,022	108	198,075	20,701,326	32,709,410
bet-spr18	105,021	14,998	80,461	4,814,300	6,473,132
iku-spr18	-	-	-	-	-
lums-fal17	-	-	-	-	-
mary-fal18	83,731	0	12,970	2,986,780	3,564,777
muni-fi-fal17	90,997	0	9,991	7,141,048	7,758,047
muni-fspsx-fal17	44,916	194	18,373	2,155,918	20,066,497
muni-pdfx-fal17	241,256	14,776	135,103	16,817,990	22,120,722
pu-d9-fal19	941,391	57,306	513,531	54,867,830	114,406,242
tg-spr18	-	-	-	-	-

Table 12 Overview of the number of variables and constraints that are connected to student sectioning and student conflicts.