Student Teacher Meetings

The student teacher meeting is a scheduling problem, i.e. a problem where events are scheduled to take place at a specific place at a specific time. This problem is actually rather hard to solve, and the problem really shows the difference between the open-source solvers and commercial solvers.

Problem

- Minimize the total meeting time for the students and parents.

Sets

- Students: $s \in Students = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$

- Teachers: $t \in Teachers = \{1, 2, 3, 4, 5, 6\}$

- Timeslots: $ts \in TimeSlots = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$

Parameters

- $StudentTeacherMeetings_{s,t}$: Incidence matrix, where $StudentTeacherMeetings_{s,t} = 1$ if student s has to have a meeting whith teacher t, otherwise 0.

Decision variables

- Should student s meet teacher t in timeslot ts: $x^s_{t,ts} \in \{0, 1\}$

- The latest meeting for student s: $y_s \geq 0$.

Model

Objective:

- Minimize the summed last meeting time for all students:

$$\sum_s y_s$$

Constraints:

- Ensure that all required student teacher meetings are scheduled:

$$\sum_{ts} x^s_{t,ts} = StudentTeacherMeetings_{s,t} \ \forall \ s,t$$

- Each student can only meet one teacher at a time:

$$sum_t x^s_{t,ts} = 1 \ \forall \ s,ts$$

- Each teacher can only meet one student at a time:

$$sum_s x^s_{t,ts} = 1 \ \forall \ t,ts$$

- Force the value of y_s to take the value of the last timeslot ts for each student:

$$ts \cdot \sum_t x^s_{t,ts} \le y_s \ \forall \ s,ts$$

Notice the trick in the last constraint: The value of the sum can only take the value 0 or 1. If it is 1 the student has a meeting at that timeslot. This value is multiplied by the index of the timeslot ts. And this is set as a lowerbound of y_s. The last meeting, hence will hence force the largest value.

The full model in Julia/JuMP, available with the name

`StudentTeacherMeetings.jl`

from the book web-site, is given below:

```
#*************************************************************************
# Student Teacher Meeting, "Mathematical Programming Modelling" (42112)

#*************************************************************************
# Intro definitions
using JuMP
using HiGHS
#*************************************************************************

#*************************************************************************
```

```julia
# Data
include("StudentTeacherMeetingData_15_6.jl")
#include("StudentTeacherMeetingData_20_6.jl")
#include("StudentTeacherMeetingData_25_10.jl")
#include("StudentTeacherMeetingData_25_8.jl")
#include("StudentTeacherMeetingData_30_10.jl")
#include("StudentTeacherMeetingData_40_10.jl")

# S: no of students   : in data file
# T: no of teachers   : in data file
# TS: no of timeslots : in data
println("Students: $(S) Teachers: $(T)  Timeslots: $(TS)")
total_no_meetings=sum(StudentTeacherMeetings[:,:])
println("Total no. of meetings: $(total_no_meetings)")
#**********************************************************************


#**********************************************************************
# Model
stmeet = Model(HiGHS.Optimizer)

# 1 if student s has a meeting with teacher t in time ts
@variable(stmeet, x[1:S,1:T,1:TS],Bin)

for s=1:S
    for t=1:T
        if StudentTeacherMeetings[s,t]==0
            for ts=1:TS
                fix(x[s,t,ts],0; force = true)
            end
        end
    end
end

# last meeting of student s
@variable(stmeet, y[1:S] >= 0)

# Minimize summed Inconvenience
@objective(stmeet, Min, sum( y[s] for s=1:S ) )

# meetings has to occur (or not occur)
@constraint(stmeet, [s=1:S,t=1:T], sum( x[s,t,ts] for ts=1:TS ) == StudentTeacherMeetings[s,

# only one meeting pr timeslot for each student s
@constraint(stmeet, [s=1:S,ts=1:TS], sum( x[s,t,ts] for t=1:T ) <= 1)

# only one meeting pr timeslot for each teacher t
```

3

```julia
@constraint(stmeet, [t=1:T,ts=1:TS], sum( x[s,t,ts] for s=1:S ) <= 1)

# force value of y to last meeting for each student
@constraint(stmeet, [s=1:S,ts=1:TS], ts*sum( x[s,t,ts] for t=1:T ) <= y[s])
#*************************************************************************


#*************************************************************************
# solve
optimize!(stmeet)
println("Termination status: $(termination_status(stmeet))")
#*************************************************************************


#*************************************************************************
println("-----------------------------------");
if termination_status(stmeet) == MOI.OPTIMAL
    println("RESULTS:")
    println("objective = $(objective_value(stmeet))")
    println("Minimal num of meetings: $(sum(StudentTeacherMeetings[:,:]))")
    println("Solve time: $(solve_time(stmeet))")
else
    println("  No solution")
end
println("-----------------------------------");
#*************************************************************************
```