Workplan for Teaching Assistants

Man power planning, i.e. deciding which worker/employee does which job, is a classic Operations Research problem. The workplan for teaching assistant prolem, is for assigning TA's to certain timeslots, such that enough TA's are available to help the students at the different timeslots.

Problem

- Minimize the TA inconvenience

Sets

- Teaching Assistants: $ta \in TeachingAssistants = \{AL, FR, JE, MI\}$
- Periods: $p \in Periods = \{"9-10", "10-11", "11-12", "12-13", "13-14", "14-15", "15-16", "16-17"\}$
- Days: $d \in Days = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Parameters

- $Demand_{p,d}$: The number of TA's necessary in timeslot t on day d
- $Inconvenience^{ta}_{p,d}$: The inconvenience value for ta to work in timeslot t on day d

Decision variables

- Should ta work in timeslot t on day d: $x^{ta}_{p,d}$.

Model

Objective:

- Minimize the total TA inconvenience:

$$\sum_{ta,p,d} Inconvenience_{p,d}^{ta} \cdot x_{p,d}^{ta}$$

Constraints:

- Ensure that the TA demand is satisfied:

$$\sum_{ta} x_{p,d}^{ta} \geq Demand_{p,d} \; \forall \; p,d$$

- Ensure that each TA is working exactly the correct no of hours:

$$\sum_{p,d} x_{p,d}^{ta} = 52 \; \forall \; ta$$

The model is actually rather simple.

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

`WorkplanTeachingAssistant1.jl`

from the book web-site, is given below:

```
#*****************************************************************************
# Workplan Teaching Assistant Assignment 1 question, "Mathematical Programming Modelling" (
using JuMP
using HiGHS
#*****************************************************************************

#*****************************************************************************
# PARAMETERS
include("WorkplanData.jl")
#*****************************************************************************

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

# 1 if ta is working on time t on day d
@variable(taworkplan, x[ta=1:TA,p=1:P,d=1:D],Bin)

# Minimize summed Inconvenience
@objective(taworkplan, Min,
           sum( Inconvenience[ta,p,d]*x[ta,p,d]
                for ta=1:TA,p=1:P,d=1:D))
```

```
# satisfy deman for TAs
@constraint(taworkplan, [p=1:P,d=1:D],
            sum( x[ta,p,d] for ta=1:TA) >= Demand[p,d])

# max no of working hours is 52
@constraint(taworkplan, [ta=1:TA],
            sum( x[ta,p,d] for p=1:P,d=1:D) == 52)
#*****************************************************************************


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


#*****************************************************************************
println("---------------------------------------");
if termination_status(taworkplan) == MOI.OPTIMAL
    println("RESULTS:")
    println("objective = $(objective_value(taworkplan))")

    for ta=1:TA
        println("")
        println("")
        print("TA: $(TA_acronyms[ta])")
        for d=1:D
            print("\t$(d)")
        end
        println("")
        for p=1:P
            print("$(Periods[p])\t")
            for d=1:D
                if round.(Int8,value(x[ta,p,d]))==1
                    print("*\t")
                else
                    print("\t")
                end
            end
            println("")
        end
    end
else
    println("  No solution")
```

```
end
println("---------------------------------------");
#*****************************************************************************
```

Comments to the Julia/JuMP program:

- Notice the printout method in the bottom of the program. Implemented to give a simple, easy to read, plan.

Question 2 in this assignment is about ensuring that the TA's only have to work in consecutive timeslot on a day and at least 2 hours when they work. One new binary variable $y_{p,d}^{ta}$ is added, and it is 1 if TA ta works the first period p on day d. Furthermore 3 constraints ensure that the requirements are satisfied.

Decision variables

- Should ta start the work on day d in period p on day d: $y_{p,d}^{ta}$.

Constraints:

- Ensure that the TA can only start to work once a day:

$$\sum_{ta} y_{p,d}^{ta} \leq 1 \ \forall \ ta, d$$

- Ensure that a TA can only work if they work in the previous period or start working this period:

$$x_{p,d}^{ta} \leq x_{p-1,d}^{ta} | (p > 1) + y_{p,d}^{ta} \ \forall \ ta, p, d$$

- : If a TA works on a day, then the TA has to work at least 2 periods:

$$\sum_{p} x_{p,d}^{ta} \geq 2 \cdot \sum_{p} y_{p,d}^{ta} \ \forall \ ta, d$$

Notice the second constraint where the $x_{p,d}^{ta}$ upwards by the previous period for this ta or the start variable $y_{p,d}^{ta}$.
The full model in Julia/JuMP, available with the name

`WorkplanTeachingAssistant2.jl`

from the book web-site, is given below:

```julia
#**************************************************************************
# Workplan Teaching Assistant Assignment 1 question, "Mathematical Programming Modelling" (
using JuMP
using HiGHS
#**************************************************************************

#**************************************************************************
# PARAMETERS
include("WorkplanData.jl")
#**************************************************************************

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

# 1 if ta is working on time t on day d
@variable(taworkplan, x[ta=1:TA,p=1:P,d=1:D],Bin)

# 1 if ta STARTS working in period p on day d
@variable(taworkplan, y[ta=1:TA,p=1:P,d=1:D],Bin)

# Minimize summed Inconvenience
@objective(taworkplan, Min,
           sum( Inconvenience[ta,p,d]*x[ta,p,d]
                for ta=1:TA,p=1:P,d=1:D))

# satisfy deman for TAs
@constraint(taworkplan, [p=1:P,d=1:D],
            sum( x[ta,p,d] for ta=1:TA) >= Demand[p,d])

# max no of working hours is 52
@constraint(taworkplan, [ta=1:TA],
            sum( x[ta,p,d] for p=1:P,d=1:D) == 52)

# only start working once a day
@constraint(taworkplan, [ta=1:TA,d=1:D],
            sum( y[ta,p,d] for p=1:P) <= 1 )

# require connected work plans
@constraint(taworkplan, [ta=1:TA,d=1:D,p=1:P],
            x[ta,p,d] <= (p>1 ? x[ta,p-1,d] : 0) + y[ta,p,d])

# at least 2 hours of work pr. day
@constraint(taworkplan, [ta=1:TA,d=1:D],
            sum( x[ta,p,d] for p=1:P) >= 2*sum( y[ta,p,d] for p=1:P) )
```

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


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


#*****************************************************************************
println("-------------------------------------");
if termination_status(taworkplan) == MOI.OPTIMAL
    println("RESULTS:")
    println("objective = $(objective_value(taworkplan))")

    for ta=1:TA
        println("")
        println("")
        print("TA: $(TA_acronyms[ta])")
        for d=1:D
            print("\t$(d)")
        end
        println("")
        for p=1:P
            print("$(Periods[p])\t")
            for d=1:D
                if round.(Int8,value(x[ta,p,d]))==1
                    print("*\t")
                else
                    print("\t")
                end
            end
            println("")
        end
    end
else
    println("  No solution")
end
println("-------------------------------------");
#*****************************************************************************
```

Comments to the Julia/JuMP program:

- Notice how the conditional insertion of the $x^{ta}_{p-1,d}$, and $p-1$ identifies the previous period.

- The printout makes it easy to see that the new constraints are satisfied. But the added constraints makes the optimal inconvenience value worse.