Class Jobs 2

The class jobs problem is then extended with an extra requirement: The jobs should be done before a timelimit, given in hours, and the teacher rates the time it will take each of the children to complete the jobs. This changes the problem in such a way that the Linear Programming version will not achieve integer value optimal solution variables. The job is to first formulate the problem as a Linear Programming model and then afterwards, change the type of variables, to binary variables. Here we will only specify the variable with binary variables.

Problem

- Maximize the aggregated wishes of the children for the jobs, not taking into account the timelimit.

Sets

- $c \in Children = \{1, 2, 3, 4, 5\}$
- $j \in Jobs = \{1, 2, 3, 4, 5\}$

Parameters

- $Wish_{j,c}$: How much child c wish job j, 1 for worst, 5 for best
- $TimeReq_{j,c}$: How much time, in hours, job j takes for child c
- $TimeLimit = 3$: The timelimit where all the jobs should be finished.

Decision variables

- Assignment variable of, if 1 child c performs job j $x_{j,c} \in \{0, 1\}$.

Model

Objective:

- Maximize the wish fulfillment:

$$\sum_{j,c} Wish_{j,c} \cdot x_{j,c}$$

Constraints:

- Each child c should be assigned to one job:

$$\sum_{j} x_{j,c} = 1 \ \forall \ c$$

- Each job j should be assigned to one child:

$$\sum_{c} x_{j,c} = 1 \ \forall \ j$$

- For each job, ensure that the job is done inside the timelimit:

$$\sum_{c} TimeReq_{j,c} x_{j,c} \leq TimeLimit \ \forall \ j$$

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

`ClassJobs2.jl`

from the book web-site, is given below:

```
#************************************************************************
# Incredible Chairs 2 assignment, LP
using JuMP
using HiGHS
#************************************************************************

#************************************************************************
# Parameters
Children=[1 2 3 4 5]
C=length(Children)
Jobs=[1 2 3 4 5]
J=length(Jobs)
Wish=[
    1 3 2 5 5;
```

```
    5 2 1 1 2;
    1 5 1 1 1;
    4 5 4 4 4;
    3 5 3 5 3]

TimeReq=[
    1 2 1 4 4;
    6 2 4 2 2;
    3 3 2 4 4;
    1 1 4 4 2;
    7 2 2 3 1
]
TimeLimit=3
#************************************************************************

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

#@variable(CJ,x[j=1:J,c=1:C] >= 0)
@variable(CJ,x[j=1:J,c=1:C], Bin)

# Maximize aggregated Wish
@objective(CJ, Max, sum( Wish[j,c]*x[j,c] for j=1:J,c=1:C ) )

# One job pr. child
@constraint(CJ, [c=1:C],
            sum( x[j,c] for j=1:J) == 1
            )

# One child pr. job
@constraint(CJ, [j=1:J],
            sum( x[j,c] for c=1:C) == 1
            )

# timelimit pr. job pr. child
@constraint(CJ, [j=1:J],
            sum( TimeReq[j,c]*x[j,c] for c=1:J) <= TimeLimit
            )

#************************************************************************

#************************************************************************
# Solve
solution = optimize!(CJ)
println("Termination status: $(termination_status(CJ))")
```

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

#*********************************************************************
if termination_status(CJ) == MOI.OPTIMAL
    println("Optimal objective value: $(objective_value(CJ))")
    println("x: ",value.(x))
else
    println("No optimal solution available")
end
#*********************************************************************
```

Comments to the Julia/JuMP program:

- Notice: There are two definitions of the x variables, one with positive continuous variables, one with binary variables. Switching between a Linear Programming model and a Mixed Integer Programming model, is then simply commenting out one line and not the other.