Jewelry Production

Create a production plan for your aunts necklace production, to maximize her income.

Problem

- Maximize production profit for your aunt.

Sets

- $n \in Necklaces = \{1, 2, 3, 4, 5\}$
- $m \in Machine = \{1, 2, 3\}$

Parameters

- $Profit_n$: Profit for production of necklace n
- $MachineTime_{m,n}$: Number of minutes needed for each necklace n on machine m
- $AssemblyTime_n$: Number of minutes needed to assemble necklace n
- $DayMinutes$: No of minutes pr. day
- $NoAssemblyWorkers$: No of assembly workers employed

Decision variables

- Number of necklaces of type n produced: $x_n \geq 0$.

Model

Objective:

- Maximize the necklace production profit:

$$\sum_n Profit_n \cdot x_n$$

Constraints:

- Limit the production to the machine capacity:

$$\sum_n MachineTime_{m,n} \cdot x_n \leq DayMinutes \ \forall m$$

- Limit the production according to the assembly capacity:

$$\sum_n AssemblyTime_n \cdot x_n \leq DayMinutes \cdot NoAssemblyWorkers$$

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

`JewelleryProduction.jl`

from the book web-site, is given below:

```
#*****************************************************************************
# Jewellery Production assignment, LP
using JuMP
using HiGHS
#*****************************************************************************

#*****************************************************************************
# Data
Necklaces=["1","2","3","4","5"]
N=length(Necklaces)
Machines=["1","2","3"]
M=length(Machines)
Profit=[50 45 85 60 55]
AssemblyTime=[12 3 11 9 6]
MachineTime=[
 7  0  0  9  0;
 5  7 11  0  5;
 0  3  8 15  3]
DayMinutes=60*7.5
NoAssemblyWorkers=2
#*****************************************************************************

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

```
# Model
JP = Model(HiGHS.Optimizer)

@variable(JP,x[n=1:N]>=0)

# maximize profit
@objective(JP, Max, sum( Profit[n]*x[n] for n=1:N) )

# Machine capacity
@constraint(JP, [m=1:M],
            sum( MachineTime[m,n]*x[n] for n=1:N) <= DayMinutes
            )

# Assembly capacity
@constraint(JP,
            sum(AssemblyTime[n]*x[n] for n=1:N) <= DayMinutes*NoAssemblyWorkers
            )
#*****************************************************************************

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

#*****************************************************************************
if termination_status(JP) == MOI.OPTIMAL
    println("Optimal objective value: $(objective_value(JP))")
    for n=1:N
        println("Production of necklace $(Necklaces[n]) : ", value(x[n]))
    end
else
    println("No optimal solution available")
end
#*****************************************************************************
```

In a market economy, you cannot force people to buy your products. Hence, it is necessary to consider the demand. This is a simple extension.

More Parameters

- $Demand_n$: The maximal number of necklace n which can be sold.

Additional Constraints:

- Added constraint to limit according to the demand:

$$x_n \leq Demand_n \ \forall n$$

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

`JewelleryProduction2.jl`

from the book web-site, is given below:

```
#*****************************************************************
# Jewellery Production 2 assignment, LP
using JuMP
using HiGHS
#*****************************************************************

#*****************************************************************
# Data
Necklaces=["Necklace 1","Necklace 2","Necklace 3","Necklace 4","Necklace 5"]
N=length(Necklaces)
Machines=["Machine 1","Machine 2","Machine 3"]
M=length(Machines)
Profit=[50 45 85 60 55]
AssemblyTime=[12 3 11 9 6]
MachineTime=[
 7  0  0  9  0;
 5  7 11  0  5;
 0  3  8 15  3]
DayMinutes=60*7.5
NoAssemblyWorkers=3
Demand=[25 10 12 15 60]
#*****************************************************************

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

@variable(JP,x[n=1:N]>=0)

# maximize profit
@objective(JP, Max, sum( Profit[n]*x[n] for n=1:N) )

# Machine capacity
@constraint(JP, [m=1:M],
            sum( MachineTime[m,n]*x[n] for n=1:N) <= DayMinutes
            )
```

```
# Assembly capacity
@constraint(JP,
            sum(AssemblyTime[n]*x[n] for n=1:N) <= DayMinutes*NoAssemblyWorkers
            )

# Demand limits
@constraint(JP, [n=1:N],
            x[n] <= Demand[n]
            )


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

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

#*****************************************************************************
if termination_status(JP) == MOI.OPTIMAL
    println("Optimal objective value: $(objective_value(JP))")
    for n=1:N
        println("Production of necklace $(Necklaces[n]) : ", value(x[n]))
    end
else
    println("No optimal solution available")
end
#*****************************************************************************
```