

Incredible Chairs 2

Incredible Chairs 2 is the extension of Incredible Chairs, such that production planning can easily be done with more chairs (products) and more production lines.

Problem

- Maximize the profit by optimizing the chair production plan with 10 chairs and 5 production lines

Sets

- $c \in Chairs = \{A, B, C, D, E, F, G, H, I, J\}$
- $p \in ProductionLines = \{1, 2, 3, 4, 5\}$

Parameters

- $Profit_c$: Profit for production of chair c
- $Capacity_p$: Capacity of production line p
- $RecourseUsage_{c,p}$: Resource usage (hours) for production of one unit of chair c on production line p

Decision variables

- Amount of chair c produced: x_c

Model

Objective:

- Total profit maximized:

$$\sum_c Profit_c \cdot x_c$$

Constraints:

- Production line capacity limitation:

$$\sum_c RecourseUsage_{c,p} \cdot x_c \leq Capacity_p \quad \forall p$$

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

`IncredibleChairs2.jl`

from the book web-site, is given below:

```

*****
# Incrediblex Chairs 2 assignment, Simple LP
using JuMP
using HiGHS
*****

*****
# Data
Chairs=["A", "B", "C", "D", "E", "F", "G", "H", "I", "J"]
C=length(Chairs)
ProductionLines=[1, 2, 3, 4, 5]
P=length(ProductionLines)

Profit=[6, 5, 9, 5, 6, 3, 4, 7, 4, 3]
Capacity=[47, 19, 36, 13, 46]
RecourseUsage=[
6 4 2 3 1 10 2 9 3 5;
5 6 1 1 7 2 9 1 8 6;
8 10 7 2 9 6 9 6 5 6;
8 4 8 10 5 4 1 5 3 5;
1 4 7 2 4 1 2 3 10 1]
*****

*****
# Model
IC2 = Model(HiGHS.Optimizer)

@variable(IC2,x[1:C]>=0)

```

```

@objective(IC2, Max, sum( Profit[c]*x[c] for c=1:C ) )

# Production lines
@constraint(IC2, [p=1:P],
    sum( RecourseUsage[p,c]*x[c] for c=1:C) <= Capacity[p]
)
*****

*****
# Solve
solution = optimize!(IC2)
println("Termination status: $(termination_status(IC2))")
*****

*****
if termination_status(IC2) == MOI.OPTIMAL
    println("Optimal objective value: $(objective_value(IC2))")
    for c=1:C
        if value(x[c])>0.001
            println("No of chairs of type ", Chairs[c], " produceed: ", value(x[c]))
        end
    end
else
    println("No optimal solution available")
end
*****

```

Comments to the Julia/JuMP program:

- Notice that because of the parameters, this generic model can be used for many different production problems.