4

October 1, 2023

[ ]:
```
using JuMP
```

[ ]:
```
using CPLEX
```

[ ]:
```
model=Model(CPLEX.Optimizer)
```

```
A JuMP Model
Feasibility problem with:
Variables: 0
Model mode: AUTOMATIC
CachingOptimizer state: EMPTY_OPTIMIZER
Solver name: CPLEX
```

[ ]:
```
n = 2
```

```
2
```

[ ]:
```
K = 6
```

```
6
```

x_1 is amount of Acid A produced and x_2 is amount of acid B produced.

[ ]:
```
@variable(model, x[1:2],lower_bound=0)
```

```
2-element Vector{VariableRef}:
 x[1]
 x[2]
```

[ ]:
```
Time=[3 4; 3 2]
```

```
2×2 Matrix{Int64}:
 3  4
 3  2
```

1 Operation 1

[ ]:
```
@constraint(model, sum(x.*Time[:,1]) <= 20)
```

$$3x_1 + 3x_2 \leq 20$$

2 Operation 2

[ ]:
```
@constraint(model, sum(x.*Time[:,2]) <= 18)
```

$$4x_1 + 2x_2 \leq 18$$

[ ]:
```
@variable(model, c_sold,lower_bound=0)
```

$$c\_sold$$

[ ]:
```
@variable(model, c_destroyed,lower_bound=0)
```

$$c\_destroyed$$

[ ]:
```
@constraint(model, c_sold <= K) # Limiting amount of C sold
```

$$c\_sold \leq 6$$

$c_{sold} + c_{destroyed} = c_{produced} = n * x_2$

[ ]:
```
@constraint(model, c_sold + c_destroyed == n*x[2])
```

$$-2x_2 + c\_sold + c\_destroyed = 0$$

[ ]:
```
P=[80 60 20 15]
```

```
1×4 Matrix{Int64}:
 80  60  20  15
```

[ ]:
```
total_profit = sum(P[1:2].*x) + P[3]*c_sold - P[4] * c_destroyed
```

$$80x_1 + 60x_2 + 20c\_sold - 15c\_destroyed$$

[ ]:
```
@objective(model,Max,total_profit)
```

$$80x_1 + 60x_2 + 20c\_sold - 15c\_destroyed$$

[ ]:
```
@show model
```

```
model = A JuMP Model
Maximization problem with:
Variables: 4
Objective function type: AffExpr
`AffExpr`-in-`MathOptInterface.EqualTo{Float64}`: 1 constraint
`AffExpr`-in-`MathOptInterface.LessThan{Float64}`: 3 constraints
`VariableRef`-in-`MathOptInterface.GreaterThan{Float64}`: 4 constraints
Model mode: AUTOMATIC
CachingOptimizer state: EMPTY_OPTIMIZER
Solver name: CPLEX
Names registered in the model: c_destroyed, c_sold, x
```

```
A JuMP Model
Maximization problem with:
Variables: 4
Objective function type: AffExpr
`AffExpr`-in-`MathOptInterface.EqualTo{Float64}`: 1 constraint
`AffExpr`-in-`MathOptInterface.LessThan{Float64}`: 3 constraints
`VariableRef`-in-`MathOptInterface.GreaterThan{Float64}`: 4 constraints
Model mode: AUTOMATIC
CachingOptimizer state: EMPTY_OPTIMIZER
Solver name: CPLEX
Names registered in the model: c_destroyed, c_sold, x
```

[ ]:
```
optimize!(model)
```

```
CPLEX Error  3003: Not a mixed-integer problem.
Version identifier: 22.1.1.0 | 2022-11-26 | 9160aff4d
Tried aggregator 1 time.
LP Presolve eliminated 1 rows and 0 columns.
Aggregator did 1 substitutions.
Reduced LP has 2 rows, 3 columns, and 6 nonzeros.
Presolve time = 0.01 sec. (0.00 ticks)

Iteration log . . .
Iteration:     1   Dual infeasibility =             0.000000
```

Amount of A and B produced

[ ]:
```
@show value.(x)
```

```
value.(x) = [3.0, 3.0]
```

```
2-element Vector{Float64}:
 3.0
 3.0
```

Amount of C produced

[ ]:
```
@show value.(n*x[2])
```

```
value.(n * x[2]) = 6.0
```

```
6.0
```

Amount of C sold

[ ]:
```
@show value.(c_sold)
```

```
value.(c_sold) = 6.0
```

```
6.0
```

Total Profit

[ ]:
```
@show objective_value(model)
```

```
objective_value(model) = 540.0
```

```
540.0
```