3

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
```

Our Decision Variable would be amount of fruits supplied by growers to each plant. For example $x_{1,2}$ denotes fruits supplied by first grower to plant B.

[ ]:
```
@variable(model, x[i=1:3,j=1:2],lower_bound=0) # Fixing minimum value of the␣
↪decision variable to 0
```

```
3×2 Matrix{VariableRef}:
 x[1,1]  x[1,2]
 x[2,1]  x[2,2]
 x[3,1]  x[3,2]
```

[ ]:
```
@constraint(model, sum(x[1,j] for j=1:2) == 200)
```

$$x_{1,1} + x_{1,2} = 200$$

[ ]:
```
@constraint(model, sum(x[2,j] for j=1:2) == 310)
```

$$x_{2,1} + x_{2,2} = 310$$

[ ]:
```
@constraint(model, sum(x[3,j] for j=1:2) == 420)
```

$$x_{3,1} + x_{3,2} = 420$$

[ ]:
```
@constraint(model, sum(x[i,1] for i=1:3) <= 460)
```

$$x_{1,1} + x_{2,1} + x_{3,1} \leq 460$$

[ ]:
```
@constraint(model, sum(x[i,2] for i=1:3) <= 560)
```

$$x_{1,2} + x_{2,2} + x_{3,2} \leq 560$$

[ ]:
```
BC=[1100,1000,900]
```

```
3-element Vector{Int64}:
 1100
 1000
  900
```

[ ]:
```
buying_cost=sum(BC[i] * sum(x[i,j] for j=1:2) for i=1:3)
```

$$1100x_{1,1} + 1100x_{1,2} + 1000x_{2,1} + 1000x_{2,2} + 900x_{3,1} + 900x_{3,2}$$

[ ]:
```
SC=[[3000,3500],[2000,2500],[6000,4000]]
```

```
3-element Vector{Vector{Int64}}:
 [3000, 3500]
 [2000, 2500]
 [6000, 4000]
```

[ ]:
```
shipping_cost= sum(sum(SC[i][j] * x[i, j] for j = 1:2) for i = 1:3)
```

$$3000x_{1,1} + 3500x_{1,2} + 2000x_{2,1} + 2500x_{2,2} + 6000x_{3,1} + 4000x_{3,2}$$

[ ]:
```
canning_cost = 26000*sum(x[i,1] for i=1:3) + 21000 * sum(x[i,2] for i=1:3)
```

$$26000x_{1,1} + 26000x_{2,1} + 26000x_{3,1} + 21000x_{1,2} + 21000x_{2,2} + 21000x_{3,2}$$

[ ]:
```
selling_price = 50000 * sum(sum(x[i, j] for j = 1:2) for i = 1:3)
```

$$50000x_{1,1} + 50000x_{1,2} + 50000x_{2,1} + 50000x_{2,2} + 50000x_{3,1} + 50000x_{3,2}$$

[ ]:
```
profit=selling_price-buying_cost-shipping_cost-canning_cost
```

$$19900x_{1,1} + 24400x_{1,2} + 21000x_{2,1} + 25500x_{2,2} + 17100x_{3,1} + 24100x_{3,2}$$

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

$$19900x_{1,1} + 24400x_{1,2} + 21000x_{2,1} + 25500x_{2,2} + 17100x_{3,1} + 24100x_{3,2}$$

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

```
A JuMP Model
Maximization problem with:
Variables: 6
Objective function type: AffExpr
`AffExpr`-in-`MathOptInterface.EqualTo{Float64}`: 3 constraints
`AffExpr`-in-`MathOptInterface.LessThan{Float64}`: 2 constraints
`VariableRef`-in-`MathOptInterface.GreaterThan{Float64}`: 6 constraints
Model mode: AUTOMATIC
CachingOptimizer state: EMPTY_OPTIMIZER
Solver name: CPLEX
Names registered in the model: x
```

```
model = A JuMP Model
Maximization problem with:
Variables: 6
Objective function type: AffExpr
`AffExpr`-in-`MathOptInterface.EqualTo{Float64}`: 3 constraints
`AffExpr`-in-`MathOptInterface.LessThan{Float64}`: 2 constraints
`VariableRef`-in-`MathOptInterface.GreaterThan{Float64}`: 6 constraints
Model mode: AUTOMATIC
CachingOptimizer state: EMPTY_OPTIMIZER
Solver name: CPLEX
Names registered in the model: 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 2 rows and 3 columns.
Aggregator did 3 substitutions.
All rows and columns eliminated.
Presolve time = 0.00 sec. (0.00 ticks)
```

[ ]:
```
@show value.(x) # Most optimal distibution
```

```
value.(x) = [60.0 140.0; 310.0 0.0; 0.0 420.0]
```

```
3×2 Matrix{Float64}:
  60.0  140.0
 310.0    0.0
   0.0  420.0
```

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

```
objective_value(model) = 2.1242e7
```

```
2.1242e7
```