Clase Modelado 1

2024-03-10

Modelado 1

1.- Formulación

Diseñe un modelo de programación lineal (mixto-entero) para resolver este problema. Explique claramente el significado de las variables, restricciones y constantes utilizadas en el modelo.

```
using JuMP, HiGHS;
model = Model(HiGHS.Optimizer);

# Tener en cuenta que x_1 es en realidad x_0
#  En este problema se considera la primera vez que se compró un tractor.
T = 20; #20 posiciones en la línea temporal de 19 años

R = 6 ;#6 posiciones en la línea temporal de 5 años máximos de depreciación

# Matriz de valor de mercado de tractor

inflacion = .05; # 5% de inflación anual

aumentoCosto = .15 ;# 15% de aumento anual en costo de mantenimiento

depreciacionInicial = .1; # 10% de depreciación anual

depreciacionCorriente = .07; # 7% de depreciación anual

pmTractorInicial = 43000/(1+inflacion)^2 ;# Precio inicial de tractor

costoInicial = 1300/(1+aumentoCosto)^2; # Costo inicial de mantenimiento
```

```
# Creamos matriz de precio de mercado de tractor
matrixPM = ones(1, T) * pmTractorInicial;

for i in 2:T
    matrixPM[i] = matrixPM[i-1] * (1+inflacion);
end
# Creamos matriz de valor en libros de tractor
matrixPB = ones(T, R);

for i in 1:T
    matrixPB[i, 1] = matrixPM[1, i]*(1-depreciacionInicial);
end

for i in 1:T
    for j in 2:R
        matrixPB[i, j] = matrixPB[i, j-1]*(1-depreciacionCorriente);
    end
end

# Creamos matriz de mantenimiento de tractor

matrixC = ones(T, R);

for i in 1:T
    matrixC[i, 1] = costoInicial*(1+aumentoCosto)^(i-1);
end

for i in 1:T
    for j in 2:R
        matrixC[i, j] = matrixC[i, j-1]*(1+aumentoCosto);
    end
end

@variable(model, x[1:T], Bin);

@variable(model, y[1:T,1:R], Bin);

@constraint(model, x[T] == 1); # Se deBe vender el tractor el año 19, ya que el Señor Marq
@constraint(model, x[1] == 1);

 # Asegurando que en 5 años debo haber vendido
```

```
 #al menos un tractor para todo 14 >= t >= 0
for t in 1:(T-5)
    @constraint(model, x[t] + x[t+1] + x[t+2] + x[t+3] + x[t+4] >= 1);
end

# Para asegurar que no compres tractor dos años consecutivos para todo 17 >= t >= 0
for t in 1:(T-2)
    @constraint(model, x[t] + x[t+1] <= 1) ;
end

# Suma de Yij debe ser igual a Xi
for i in 1:(T-1)
    @constraint(model, sum(y[i,j] for j in 1:R) == x[i]) ;
end

# Asumiendo que T y R definen los límites de tus índices
for z in 3:(T+R)
    @constraint(model, sum(y[i, j] for i in 1:T, j in 1:R if i + j == z) ==
    (z-1 <= T ? x[z-1] : 0));
end

# Restricción para no vender en el mismo momento de compra
for i in 1:T
    @constraint(model, y[i,1] == 0);
end

@constraint(model, sum(y[T,j] for j in 1:R) == 0) # El final no se contabiliza;


@objective(model, Max, sum(matrixPB[i,j]*y[i,j] for i in 1:T, j in 1:R) -
sum(matrixPM[i]*x[i] for i in 1:(T-1)) );
```

```
optimization_result = optimize!(model);

value.(x)
```

```
Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
68 rows, 102 cols, 305 nonzeros
56 rows, 71 cols, 285 nonzeros
43 rows, 64 cols, 260 nonzeros
```

```
43 rows, 63 cols, 273 nonzeros

Solving MIP model with:
   43 rows
   63 cols (63 binary, 0 integer, 0 implied int., 0 continuous)
   273 nonzeros

        Nodes      |    B&B Tree     |            Objective Bounds              |  Dynamic Co
     Proc. InQueue |  Leaves   Expl. | BestBound       BestSol              Gap |   Cuts   In

         0       0         0   0.00%   182682.084013   -inf                 inf        0
 T       0       0         0   0.00%   182682.084013   -81962.341236    322.89%        0

Solving report
  Status            Optimal
  Primal bound      -81962.3412359
  Dual bound        -81962.3412359
  Gap               0% (tolerance: 0.01%)
  Solution status   feasible
                    -81962.3412359 (objective)
                    0 (bound viol.)
                    0 (int. viol.)
                    0 (row viol.)
  Timing            0.00 (total)
                    0.00 (presolve)
                    0.00 (postsolve)
  Nodes             1
  LP iterations     40 (total)
                    0 (strong br.)
                    0 (separation)
                    0 (heuristics)

20-element Vector{Float64}:
  1.0
  0.0
  0.0
  0.0
  1.0
  0.0
  0.0
 -0.0
 -0.0
```

```
 1.0
-0.0
-0.0
-0.0
-0.0
 1.0
 0.0
 0.0
 0.0
 0.0
 1.0
```

2.- Resolución

Resuelva el problema de programación lineal anterior y muestre en diferentes tablas que le ensenarán al Sr. Márquez cómo varían los costes y precios de venta, del tractor en cada año, así como los costes de mantenimiento. Explique dichas tablas e indique cómo se calcularía el coste total de mantenimiento del tractor a lo largo de los 17 años

```julia
using DataFrames

# Crear un DataFrame para mostrar los resultados
df1 = DataFrame(matrixPM, :auto)
df1_transposed = DataFrame(transpose(Matrix(df1)), :auto)

df2 = DataFrame(matrixPB,:auto)

df3 = DataFrame(matrixC,:auto)

df1_transposed
```

	x1
	Float64
1	39002.3
2	40952.4
3	43000.0
4	45150.0
5	47407.5
6	49777.9
7	52266.8
8	54880.1
9	57624.1
10	60505.3
11	63530.6
12	66707.1
13	70042.5
14	73544.6
15	77221.8
16	81082.9
17	85137.1
18	89393.9
19	93863.6
20	98556.8

```
df2
```

	x1	x2	x3	x4	x5	x6
	Float64	Float64	Float64	Float64	Float64	Float64
1	35102.0	32644.9	30359.8	28234.6	26258.2	24420.1
2	36857.1	34277.1	31877.7	29646.3	27571.1	25641.1
3	38700.0	35991.0	33471.6	31128.6	28949.6	26923.1
4	40635.0	37790.5	35145.2	32685.0	30397.1	28269.3
5	42666.8	39680.1	36902.5	34319.3	31916.9	29682.8
6	44800.1	41664.1	38747.6	36035.3	33512.8	31166.9
7	47040.1	43747.3	40685.0	37837.0	35188.4	32725.2
8	49392.1	45934.6	42719.2	39728.9	36947.9	34361.5
9	51861.7	48231.4	44855.2	41715.3	38795.2	36079.6
10	54454.8	50643.0	47097.9	43801.1	40735.0	37883.6
11	57177.5	53175.1	49452.8	45991.1	42771.8	39777.7
12	60036.4	55833.9	51925.5	48290.7	44910.4	41766.6
13	63038.2	58625.5	54521.8	50705.2	47155.9	43855.0
14	66190.1	61556.8	57247.8	53240.5	49513.7	46047.7
15	69499.6	64634.7	60110.2	55902.5	51989.3	48350.1
16	72974.6	67866.4	63115.8	58697.6	54588.8	50767.6
17	76623.4	71259.7	66271.5	61632.5	57318.3	53306.0
18	80454.5	74822.7	69585.1	64714.2	60184.2	55971.3
19	84477.2	78563.8	73064.4	67949.9	63193.4	58769.8
20	88701.1	82492.0	76717.6	71347.4	66353.0	61708.3

```
df3
```

	x1	x2	x3	x4	x5	x6
	Float64	Float64	Float64	Float64	Float64	Float64
1	982.987	1130.43	1300.0	1495.0	1719.25	1977.14
2	1130.43	1300.0	1495.0	1719.25	1977.14	2273.71
3	1300.0	1495.0	1719.25	1977.14	2273.71	2614.76
4	1495.0	1719.25	1977.14	2273.71	2614.76	3006.98
5	1719.25	1977.14	2273.71	2614.76	3006.98	3458.03
6	1977.14	2273.71	2614.76	3006.98	3458.03	3976.73
7	2273.71	2614.76	3006.98	3458.03	3976.73	4573.24
8	2614.76	3006.98	3458.03	3976.73	4573.24	5259.23
9	3006.98	3458.03	3976.73	4573.24	5259.23	6048.11
10	3458.03	3976.73	4573.24	5259.23	6048.11	6955.33
11	3976.73	4573.24	5259.23	6048.11	6955.33	7998.62
12	4573.24	5259.23	6048.11	6955.33	7998.62	9198.42
13	5259.23	6048.11	6955.33	7998.62	9198.42	10578.2
14	6048.11	6955.33	7998.62	9198.42	10578.2	12164.9
15	6955.33	7998.62	9198.42	10578.2	12164.9	13989.6
16	7998.62	9198.42	10578.2	12164.9	13989.6	16088.1
17	9198.42	10578.2	12164.9	13989.6	16088.1	18501.3
18	10578.2	12164.9	13989.6	16088.1	18501.3	21276.5
19	12164.9	13989.6	16088.1	18501.3	21276.5	24468.0
20	13989.6	16088.1	18501.3	21276.5	24468.0	28138.2

3.- Aumentando costos

```
# Creamos matriz estática nueva de mantenimiento de tractor
aumentoCosto2 = .03 ;# 3% de aumento anual en costo de mantenimiento
matrixCEst = ones(T, R);

for i in 1:T
    matrixCEst[i, 1] = .01*matrixPM[1,i]*(1+aumentoCosto)^(i-1);
end

for i in 1:T
    for j in 2:3
        matrixCEst[i, j] = matrixCEst[i, j-1]*(1+aumentoCosto);
    end
end

for i in 1:T
```

```julia
    for j in 4:R
        matrixCEst[i, j] = matrixCEst[i, j-1]*(1+aumentoCosto)*(1+aumentoCosto2);
    end
end

matrixC = cumsum(matrixCEst, dims=2);

@objective(model, Max, sum(matrixPB[i,j]*y[i,j] for i in 1:T, j in 1:R) -
sum(matrixPM[i]*x[i] for i in 1:(T-1)) -
sum(matrixC[i,j]*y[i,j] for i in 1:T, j in 1:R) );
```

```julia
optimization_result = optimize!(model);

value.(x)
```

```
Presolving model
68 rows, 102 cols, 305 nonzeros
56 rows, 71 cols, 285 nonzeros
43 rows, 64 cols, 260 nonzeros
43 rows, 61 cols, 268 nonzeros

Solving MIP model with:
   43 rows
   61 cols (61 binary, 0 integer, 0 implied int., 0 continuous)
   268 nonzeros

        Nodes      |    B&B Tree     |            Objective Bounds              |  Dynamic Co
     Proc. InQueue |  Leaves   Expl. | BestBound       BestSol              Gap |   Cuts   In

         0       0         0   0.00%   96649.992828    -inf                 inf        0
 T       0       0         0   0.00%   96649.992828    -161224.267873   159.95%        0

Solving report
  Status            Optimal
  Primal bound      -161224.267873
  Dual bound        -161224.267873
  Gap               0% (tolerance: 0.01%)
  Solution status   feasible
                    -161224.267873 (objective)
                    0 (bound viol.)
                    0 (int. viol.)
```

```
                  0 (row viol.)
  Timing          0.00 (total)
                  0.00 (presolve)
                  0.00 (postsolve)
  Nodes           1
  LP iterations   39 (total)
                  0 (strong br.)
                  0 (separation)
                  0 (heuristics)
```

```
20-element Vector{Float64}:
  1.0
  0.0
  0.0
  0.0
  1.0
  0.0
  0.0
 -0.0
 -0.0
  1.0
 -0.0
 -0.0
 -0.0
 -0.0
  1.0
  0.0
  0.0
  0.0
  0.0
  1.0
```

```
value.(y)
```

```
20×6 Matrix{Float64}:
 0.0  0.0   0.0   0.0   1.0  -0.0
 0.0  0.0   0.0   0.0   0.0   0.0
 0.0  0.0   0.0   0.0  -0.0  -0.0
 0.0  0.0   0.0   0.0   0.0  -0.0
 0.0  0.0   0.0   0.0   0.0   1.0
 0.0  0.0   0.0  -0.0  -0.0   0.0
```

```
0.0  0.0  -0.0   0.0   0.0   0.0
0.0  0.0   0.0   0.0   0.0   0.0
0.0  0.0  -0.0   0.0   0.0   0.0
0.0  0.0  -0.0   0.0   0.0   1.0
0.0  0.0   0.0  -0.0  -0.0   0.0
0.0  0.0  -0.0   0.0   0.0   0.0
0.0  0.0  -0.0   0.0   0.0   0.0
0.0  0.0  -0.0   0.0   0.0   0.0
0.0  0.0   0.0   0.0   0.0   1.0
0.0  0.0   0.0   0.0  -0.0   0.0
0.0  0.0   0.0  -0.0   0.0   0.0
0.0  0.0   0.0   0.0   0.0   0.0
0.0  0.0   0.0   0.0   0.0   0.0
0.0  0.0   0.0   0.0   0.0   0.0
```

4.- Resolución

Muestre cómo el cambio anterior en el coste del tractor afectaría en el modelo de programación lineal.

No cambia nada

5.- Conclusiones

Martín tiene pensado continuar con el trabajo en la granja una vez que su padre se retire. El quiere aprovechar el trabajo de su hermana y evitarse problemas con el tractor una vez que su padre se jubile. Así le pide a su hermana que investigue si cambiarían las políticas de reemplazo cuando el cambio del tractor se realice de forma periódica en el tiempo. Para de esta manera poder seguir esa política periódica siempre y cuando los precios se ajusten a esos valores sin importar cuántos años esté al frente de la granja

```julia
using JuMP, HiGHS;
model = Model(HiGHS.Optimizer);

# Tener en cuenta que x_1 es en realidad x_0
#  En este problema se considera la primera vez que se compró un tractor.
T = 7; #20 posiciones en la línea temporal de 19 años

R = 6 ;#6 posiciones en la línea temporal de 5 años máximos de depreciación
```

```julia
# Matriz de valor de mercado de tractor

inflacion = .05; # 5% de inflación anual

aumentoCosto = .15 ;# 15% de aumento anual en costo de mantenimiento

depreciacionInicial = .1; # 10% de depreciación anual

depreciacionCorriente = .07; # 7% de depreciación anual

pmTractorInicial = 43000/(1+inflacion)^2 ;# Precio inicial de tractor

costoInicial = 1300/(1+aumentoCosto)^2; # Costo inicial de mantenimiento

# Creamos matriz de precio de mercado de tractor
matrixPM = ones(1, T) * pmTractorInicial;

for i in 2:T
    matrixPM[i] = matrixPM[i-1] * (1+inflacion);
end
# Creamos matriz de valor en libros de tractor
matrixPB = ones(T, R);

for i in 1:T
    matrixPB[i, 1] = matrixPM[1, i]*(1-depreciacionInicial);
end

for i in 1:T
    for j in 2:R
        matrixPB[i, j] = matrixPB[i, j-1]*(1-depreciacionCorriente);
    end
end


@variable(model, x[1:T], Bin);

@variable(model, y[1:T,1:R], Bin);

@constraint(model, x[T] == 1); # Se deBe vender el tractor el año 19, ya que el Señor Marq
@constraint(model, x[1] == 1);
```

```
 # Asegurando que en 5 años debo haber vendido
 #al menos un tractor para todo 14 >= t >= 0
for t in 1:(T-5)
    @constraint(model, x[t] + x[t+1] + x[t+2] + x[t+3] + x[t+4] >= 1);
end

# Para asegurar que no compres tractor dos años consecutivos para todo 17 >= t >= 0
for t in 1:(T-2)
    @constraint(model, x[t] + x[t+1] <= 1) ;
end

# Suma de Yij debe ser igual a Xi
for i in 1:(T-1)
    @constraint(model, sum(y[i,j] for j in 1:R) == x[i]) ;
end

# Asumiendo que T y R definen los límites de tus índices
for z in 3:(T+R)
    @constraint(model, sum(y[i, j] for i in 1:T, j in 1:R if i + j == z) ==
    (z-1 <= T ? x[z-1] : 0));
end

# Restricción para no vender en el mismo momento de compra
for i in 1:T
    @constraint(model, y[i,1] == 0);
end

@constraint(model, sum(y[T,j] for j in 1:R) == 0) # El final no se contabiliza;


# Creamos matriz estática nueva de mantenimiento de tractor
aumentoCosto2 = .03 ;# 3% de aumento anual en costo de mantenimiento
matrixCEst = ones(T, R);

for i in 1:T
    matrixCEst[i, 1] = .01*matrixPM[1,i]*(1+aumentoCosto)^(i-1);
end

for i in 1:T
    for j in 2:3
        matrixCEst[i, j] = matrixCEst[i, j-1]*(1+aumentoCosto);
    end
```

```
end

for i in 1:T
    for j in 4:R
        matrixCEst[i, j] = matrixCEst[i, j-1]*(1+aumentoCosto)*(1+aumentoCosto2);
    end
end

matrixC = cumsum(matrixCEst, dims=2);

@objective(model, Max, sum(matrixPB[i,j]*y[i,j] for i in 1:T, j in 1:R) -
sum(matrixPM[i]*x[i] for i in 1:(T-1)) -
sum(matrixC[i,j]*y[i,j] for i in 1:T, j in 1:R) );

optimize!(model)

value.(x)
```

```
Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
16 rows, 24 cols, 58 nonzeros
8 rows, 12 cols, 42 nonzeros
0 rows, 0 cols, 0 nonzeros
Presolve: Optimal

Solving report
  Status            Optimal
  Primal bound      -27967.9957135
  Dual bound        -27967.9957135
  Gap               0% (tolerance: 0.01%)
  Solution status   feasible
                    -27967.9957135 (objective)
                    0 (bound viol.)
                    0 (int. viol.)
                    0 (row viol.)
  Timing            0.00 (total)
                    0.00 (presolve)
                    0.00 (postsolve)
  Nodes             0
  LP iterations     0 (total)
                    0 (strong br.)
                    0 (separation)
```

```
                    0 (heuristics)

7-element Vector{Float64}:
 1.0
 0.0
 1.0
 0.0
 0.0
 0.0
 1.0
```