

Clase Modelado 1

2024-03-09

Modelado 1

Formulación

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

# Tener en cuenta que x_1 es en realidad x_0

T = 20; #18 posiciones en la línea temporal de 17 años

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

# Matriz de valor de mercado de tractor

inflacion = .15; # 15% de inflación anual

depreciacionInicial = .1; # 10% de depreciación anual

depreciacionCorriente = .07; # 7% de depreciación anual

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

# 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);

@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 17, ya que el Señor Marq
@constraint(model, x[1] == 1);

# Asegurando que en 5 años debo haber vendido al menos un tractor para todo 12 >= t >= 0
for t in 1:13
    @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 15 >= t >= 0
for t in 1:16
    @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[
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]*

optimization_result = optimize!(model);

value.(x)

```

Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms

Presolving model

64 rows, 102 cols, 291 nonzeros

53 rows, 82 cols, 280 nonzeros

48 rows, 61 cols, 247 nonzeros

36 rows, 54 cols, 208 nonzeros

34 rows, 49 cols, 210 nonzeros

Solving MIP model with:

34 rows

49 cols (49 binary, 0 integer, 0 implied int., 0 continuous)

210 nonzeros

	Nodes		B&B Tree		Objective Bounds			Dynamic C	
	Proc.	InQueue	Leaves	Expl.	BestBound	BestSol	Gap	Cuts	In
	0	0	0	0.00%	-240458.487147	-inf	inf	0	
T	0	0	0	0.00%	-240458.487147	-623400.100996	61.43%	0	

Solving report

Status Optimal

Primal bound -623400.100996

Dual bound	-623400.100996
Gap	0% (tolerance: 0.01%)
Solution status	feasible
	-623400.100996 (objective)
	0 (bound viol.)
	0 (int. viol.)
	0 (row viol.)
Timing	0.00 (total)
	0.00 (presolve)
	0.00 (postsolve)
Nodes	1
LP iterations	24 (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