Aula Mosel-Xpress
12/02/2020

Leitura de arquivo txt

- A aula de hoje vai se basear no exemplo apresentado no slide a seguir.
- Este é um exercício da primeira lista de modelagem.
- Depois de entender e executar este modelo no Xpress Mosel, coloque os outros modelos da lista e execute-os.

Uma grande fábrica de móveis dispõe, em estoque, de 250 m. de tábuas, 600 m. de pranchas e 500 m. de painéis de conglomerado. A fábrica normalmente oferece uma linha de móveis composta por um modelo de escrivaninha, uma mesa de reunião, um armário e uma prateleira. A Tabela 1 informa a quantidade de matéria-prima utilizada e o valor de revenda de cada produto. Exiba um modelo que maximize a receita com a venda dos móveis.

	Quantidade de material em m. por unidade de produto				Disponibilidade de recurso (m)
	Escrivaninha	Mesa	Armário	Prateleira	
Tábua	1	1	1	4	250
Prancha	0	1	1	2	600
Painéis	3	2	4	0	500
Valor de revenda	100	80	120	20	

Lendo vetores e matrizes de um arquivo txt

- Para ler um arquivo txt, é necessário colocar o caminho do local onde você vai ler o seu arquivo txt no módulo parameters.
- Parameters fica abaixo do uses mmxprs;
- Exemplo:

```
parameters
   PROJECTDIR='C:\Users\anauz\UFPB\2019.2'
end-parameters
```

Lendo vetores e matrizes de um arquivo txt

- Crio um arquivo txt com os dados no diretório informado, que neste caso é o mesmo diretório onde se encontra o programa.

  A:[1 1 1 4 0 1 1 2 3 2 4 0]

  b: [250 600 500]

  c: [100 80 120 20]

Lendo vetores e matrizes de um arquivo txt

- Depois do módulo projectdir vem o módulo declarations, onde você tem que declarar também as estruturas que serão lidas do arquivo.

```
declarations
rest=1..3 !numero de restrições
produtos=1..4
x: array(produtos) of mpvar
A:array(rest,produtos) of real
b:array(rest) of real
c:array(produtos) of real
end-declarations
```

Lendo vetores e matrizes de um arquivo txt

- Depois, do módulo declarations, precisamos ler o conteúdo do arquivo, para isso informamos o conteúdo do arquivo no módulo initializations

initializations from "dados_exe1.txt"

A b c

end-initializations

Escrevendo a solução em um arquivo txt

- Caso você queira que a solução seja impressa em um arquivo, após a função objetivo você deve escrever

  fopen("solução-designação-exemplo-livro.txt", F_OUTPUT+F_APPEND)
- Use a opção F_OUTPUT+F_APPEND se você quiser conservar todas as soluções encontradas e apenas F_OUTPUT no caso de você querer que o software escreva por cima, ou seja, você só terá o resultado da última vez que executou o programa.

```
model ModelName
uses "mmxprs"; !gain access to the Xpress-Optimizer solver
!optional parameters section
parameters
   PROJECTDIR='C:\Users\anauz\UFPB\2019.2' ! for when file is added to project
end-parameters
declarations
rest=1..3 !numero de restrições
produtos=1..4
x: array(produtos) of mpvar
A:array(rest,produtos) of real
b:array(rest) of real
c:array(produtos) of real
end-declarations
initializations from "dados_exe1.txt"
       A b c
end-initializations
forall(i in rest) do
  sum(j in produtos) A(i,j)*x(j) <= b(i)
end-do
obj:= sum(j in produtos) c(j)*x(j)
maximize(obj)
forall(j in produtos) do
writeln("x(", j,")",getsol(x(j)))
end-do
writeln("FO: ", getobjval)
writeln("End running model")
end-model
```

```
model ModelName
uses "mmxprs"; !gain access to the Xpress-Optimizer solver
parameters
   PROJECTDIR='C:\Users\anauz\UFPB\2019.2'
end-parameters
declarations
rest=1..3 !numero de restrições
produtos=1..4
x: array(produtos) of mpvar
A:array(rest,produtos) of real
b:array(rest) of real
c:array(produtos) of real
end-declarations
initializations from "dados_exe1.txt"
      A b c
end-initializations
forall(i in rest) do
  sum(j in produtos) A(i,j)*x(j) <= b(i)
end-do
obj:= sum(j in produtos) c(j)*x(j)
maximize(obj)
forall(j in produtos) do
writeln("x(", j,")",getsol(x(j)))
end-do
writeln("FO: ", getobjval)
end-model
```

```
x(1)= 0
x(2)= 250
x(3)= 0
x(4)= 0
FO: 20000
```

Impressão do modelo

- Você pode querer visualizar o modelo que está sendo gerado pelo programa.
- Para isto, você terá que digitar o comando

  exportprob(arguments1,arguments2,arguments3)
- arguments1: EP_MIN (minimization – padrão)

  EP_MAX (maximization)

  EP_MPS (formato MPS)

Impressão do modelo

- arguments 2: nome do arquivo de saída, ou seja, nome do arquivo onde você quer que o seu modelo seja impresso.

  Se você usar “”, o modelo será impresso no mesmo arquivo para onde você direcionou a saída do programa. Se você não direcionou a saída do programa para um arquivo, o modelo também será impresso na tela do computador .

Impressão do modelo

- arguments 3: nome dado à equação da função objetivo do problema
- Coloque o exportprob no modelo que você criou.