Aula 16/03/2020
Xpress Mosel

Conjuntos e vetores

- Já utilizamos a boa prática de nomear conjuntos para utilizá-los ao definir vetores

  declarations
- RI = 1..N → nomeamos o conjunto
- x: array(RI) of mpvar → definimos o vetor
- end-declarations

Vetores

- Porém, para modelarmos os problemas de fluxo a seguir, precisaremos de um tipo especial de vetor, denominado vetor dinâmico.
- No Mosel, um vetor é dinâmico se é indexado por um conjunto dinâmico.
- Se um vetor dinâmico estiver sendo utilizado para representar dados densos, deveria ser evitado, pois utiliza mais memória e é mais lento do que um vetor estático.

Vetores

- A declaração de um vetor dinâmico se dá de duas formas:

  1. declaração explícita com a utilização da palavra reservada dynamic

  2. declaração implícita

Vetores

- Exemplo de declaração explícita de um vetor dinâmico

```
declarations
  I=1..1000
  J=1..500
  A:dynamic array(I,J) of real
  x: array(I,J) of mpvar
end-declarations
initializations from "mydata.txt"
  A
end-initializations
C:= sum(i in I,j in J | exists(A(i,j))) A(i,j)*x(i,j) = 0
```

Vetores

- Exemplo de declaração implícita de um vetor dinâmico

```
declarations
  S: set of string
  A,B: array(S) of real ! O tamanho de S não é conhecido ainda
  x: array(S) of mpvar
end-declarations

initializations from "mydata.dat"
  A
end-initializations

forall(s in S) create(x(s))
```

Vetores

- Ao trabalhar com vetores esparsos, a sequência de índices nos laços deve corresponder à sequência dada em sua declaração.
- O intervalo “range” significa que você usará um intervalo de inteiros
- Também é uma declaração implícita de vetor dinâmico

Vetores

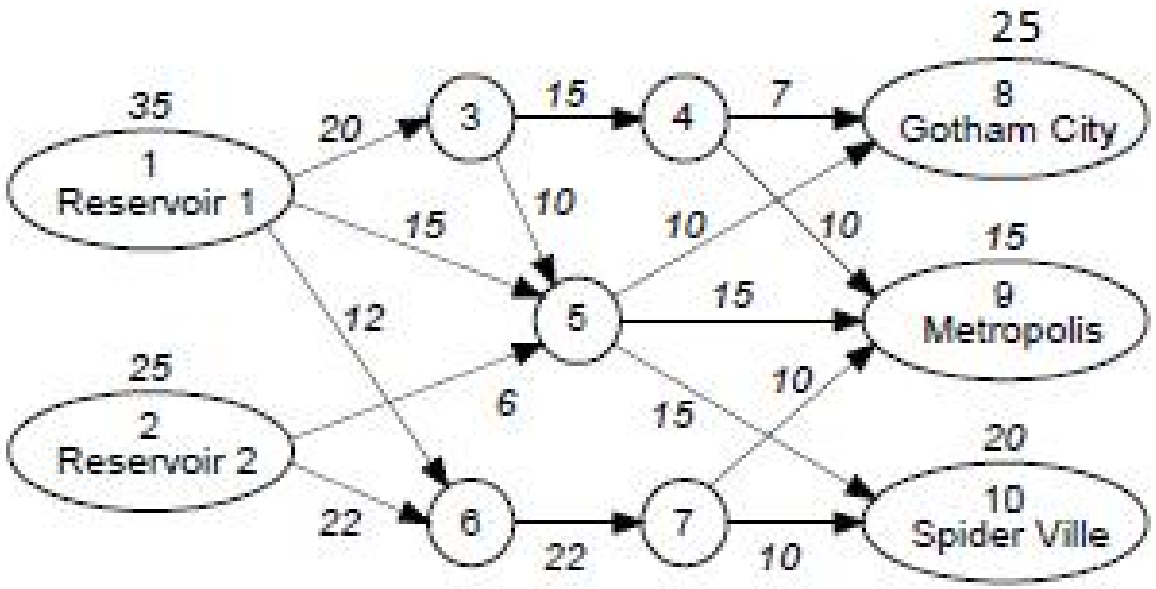
- Exemplo:

  declarations
- A1: array(1..3) of integer ! Vetor de tam. fixo
- R : range
- F = {"a","b","c"}
- A2: array(F) of real ! Vetor de tam. fixo
- A3: array(R) of integer ! vetor dinâmico implícito
- A4: dynamic array(R) of real ! vetor dinâmico explícito
- end-declarations

```
model cam_min
uses "mmxprs"; !gain access to the Xpress-Optimizer solver
parameters
   PROJECTDIR='C:\Users\anaflavia\ufpb\2017.2\Pesquisa Operacional\modelos mosel AF'
end-parameters
declarations
  nos: range
  origem=1
  destino=6
  meio=2..5
  tudo=1..6
  arcos:dynamic array(nos,nos) of integer
  x: dynamic array(nos,nos) of mpvar
end-declarations
initializations from 'caminho-min1.txt'
        arcos
end-initializations
forall(i,j in tudo | exists(arcos(i,j)))  create(x(i,j))
 - sum(i in meio) x(origem,i) = -1
sum (j in meio) x(j,destino) = 1
forall(i in meio) do
   sum(k in nos)x(k,i) = sum(j in nos) x(i,j)
end-do
custo:= sum(i in tudo, j in tudo) arcos(i,j)*x(i,j)
minimize(custo)
writeln("Begin running model")
forall(i in tudo, j in tudo | exists(arcos(i,j))) do
  writeln("valor(",i,j,")= ",getsol(x(i,j)))
end-do
writeln("FO: ",getobjval)
writeln("End running model")
exportprob(EP_MIN,"",custo)
end-model
```

```
arcos:[(1 2) 2
       (1 3) 1
       (2 4) 3
       (3 5) 2
       (4 6) 4
       (5 6) 3]
```

Exercício transbordo



```
model transbordo
uses "mmxprs";
!sample declarations section
declarations
OFE=1..2
FINAL=8..10
MEIO=3..7
TUDO=1..10
 nos: range
 flow: dynamic array(nos,nos) of mpvar
 custos: dynamic array(nos,nos) of real
 oferta: array(OFE) of real
 demanda: array(FINAL) of real
end-declarations
initializations from 'transbordo.txt'
  custos oferta demanda
 end-initializations
forall(i,j in nos | exists(custos(i,j)))  create(flow(i,j))
!nós de oferta
forall (k in OFE) do
  -sum(j in MEIO | exists(custos(k,j))) flow(k,j) <= - oferta(k)
end-do
forall (k in FINAL) do
 sum(j in MEIO | exists(custos(j,k)))flow(j,k) >= demanda(k)
end-do
forall(i in MEIO) do
   sum(k in nos | exists(custos(k,i)))flow(k,i) = sum(j in nos | exists(custos(i,j))) flow(i,j)
end-do
custo:=sum(k in TUDO, j in TUDO | exists(custos(k,j))) custos(k,j)*flow(k,j)
minimize(custo)
writeln("Begin running model")
forall(i in TUDO, j in TUDO | exists(custos(i,j))) do
  writeln("valor(",i,j,")= ",getsol(flow(i,j)))
end-do
writeln("FO: ",getobjval)
writeln("End running model")
exportprob(EP_MIN,"",custo)
end-model
```

custos:[(1 3) 20
(1 5) 15
(1 6) 12
(2 5) 6
(2 6) 22
(3 4) 15
(3 5) 10
(4 8) 7
(4 9) 10
(5 8) 10
(5 9) 15
(5 10) 15
(6 7) 22
(7 9) 10
(7 10) 10]

oferta:[35 25]

demanda:[25 15 20]

Begin running model
valor(13)= 0
valor(15)= 35
valor(16)= 0
valor(25)= 25
valor(26)= 0
valor(34)= 0
valor(35)= 0
valor(48)= 0
valor(49)= 0
valor(58)= 25
valor(59)= 15
valor(510)= 20
valor(67)= 0
valor(79)= 0
valor(710)= 0
FO: 1450
End running model