

```
> restart : # Mangasarian-Stone-Pricing.mw
```

```
> # a and b are expected payoffs at equilibrium
```

```
> A :=
```

$$\begin{bmatrix} 6 & 0 & 0 \\ 10 & 5 & 0 \\ 8 & 8 & 4 \end{bmatrix}$$

$$A := \begin{bmatrix} 6 & 0 & 0 \\ 10 & 5 & 0 \\ 8 & 8 & 4 \end{bmatrix}$$

(1)

```
> B :=
```

$$\begin{bmatrix} 6 & 10 & 8 \\ 0 & 5 & 8 \\ 0 & 0 & 4 \end{bmatrix}$$

$$B := \begin{bmatrix} 6 & 10 & 8 \\ 0 & 5 & 8 \\ 0 & 0 & 4 \end{bmatrix}$$

(2)

```
> with(LinearAlgebra) :
```

```
> m := RowDimension(A); n := ColumnDimension(B)
```

$m := 3$

$n := 3$

(3)

```
> X := Vector(m, symbol=x)
```

$$X := \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

(4)

```
> Y := Vector(n, symbol=y)
```

$$Y := \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

(5)

```
> U := Vector(1..m, 1)
```

$$U := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

(6)

```
> V := Vector(1..n, 1)
```

$$V := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (7)$$

$$\begin{aligned} &> z := \text{Transpose}(X).(A+B).Y - a - b \\ & z := y_1(12x_1 + 10x_2 + 8x_3) + y_2(10x_1 + 10x_2 + 8x_3) + y_3(8x_1 + 8x_2 + 8x_3) - a - b \end{aligned} \quad (8)$$

$$> AY := A.Y - a \cdot U$$

$$AY := \begin{bmatrix} 6y_1 - a \\ 10y_1 + 5y_2 - a \\ 8y_1 + 8y_2 + 4y_3 - a \end{bmatrix} \quad (9)$$

$$> AY(1)$$

$$6y_1 - a \quad (10)$$

$$> BX := \text{Transpose}(B).X - b \cdot V$$

$$BX := \begin{bmatrix} 6x_1 - b \\ 10x_1 + 5x_2 - b \\ 8x_1 + 8x_2 + 4x_3 - b \end{bmatrix} \quad (11)$$

$$> BX(1)$$

$$6x_1 - b \quad (12)$$

$$> \text{Transpose}(U).X$$

$$x_1 + x_2 + x_3 \quad (13)$$

$$> \text{Transpose}(V).Y$$

$$y_1 + y_2 + y_3 \quad (14)$$

$$\begin{aligned} &> \text{for } i \text{ from } 1 \text{ to } m \text{ do} \\ & \quad eA[i] := AY(i) \leq 0 \\ & \text{od} \end{aligned}$$

$$\begin{aligned} & eA_1 := 6y_1 - a \leq 0 \\ & eA_2 := 10y_1 + 5y_2 - a \leq 0 \\ & eA_3 := 8y_1 + 8y_2 + 4y_3 - a \leq 0 \end{aligned} \quad (15)$$

$$\begin{aligned} &> \text{for } j \text{ from } 1 \text{ to } n \text{ do} \\ & \quad eB[j] := BX(j) \leq 0 \\ & \text{od} \end{aligned}$$

$$\begin{aligned} & eB_1 := 6x_1 - b \leq 0 \\ & eB_2 := 10x_1 + 5x_2 - b \leq 0 \\ & eB_3 := 8x_1 + 8x_2 + 4x_3 - b \leq 0 \end{aligned} \quad (16)$$

$$> \text{Transpose}(U).X = 1$$

$$x_1 + x_2 + x_3 = 1 \quad (17)$$

$$> \text{Transpose}(V).Y = 1$$

$$(18)$$

$$y_1 + y_2 + y_3 = 1 \quad (18)$$

```
> with(Optimization)
  [ImportMPS, Interactive, LPSolve, LSSolve, Maximize, Minimize, NLPsolve, QPSolve] (19)
```

```
> Cons := {seq(eA[i], i = 1 .. m), seq(eB[j], j = 1 .. n), Transpose(U).X = 1, Transpose(V).Y
  = 1, seq(X[i] ≥ 0, i = 1 .. m), seq(Y[j] ≥ 0, j = 1 .. n)};
```

```
Cons := {x1 + x2 + x3 = 1, y1 + y2 + y3 = 1, 0 ≤ x1, 0 ≤ x2, 0 ≤ x3, 0 ≤ y1, 0 ≤ y2, 0 ≤ y3, 6 x1
  - b ≤ 0, 6 y1 - a ≤ 0, 10 x1 + 5 x2 - b ≤ 0, 10 y1 + 5 y2 - a ≤ 0, 8 x1 + 8 x2 + 4 x3 - b
  ≤ 0, 8 y1 + 8 y2 + 4 y3 - a ≤ 0} (20)
```

```
> QPSolve(z, Cons, maximize)
[0., [a = 4.000000000000000266, b = 3.99999999999999734, x1 = (21)
```

```
-1.11022302462515654 10-16, x2 = -1.11022302462515654 10-16, x3
  = 0.99999999999999978, y1 = 1.11022302462515654 10-16, y2 = 0., y3
  = 1.000000000000000044 ]]
```

```
> #with(DirectSearch)
```

```
> #Search(z, Cons, maximize)
```

```
>
```

```
>
```