MAXPLUS ALGEBRA IN SCILAB AND APPLICATIONS

G. COHEN, S. GAUBERT & J.P. QUADRAT

This work has been partly supported by the ALAPEDES project of the European TMR programme.

CONTENTS

1. STRUCTURES

- A semiring $\mathcal{K}$ is a set endowed with two operations denoted $\oplus$ and $\otimes$ where $\oplus$ is associative, commutative with zero element denoted ε, $\otimes$ is associative, admits a unit element denoted e, and distributes over $\oplus$; zero is absorbing ($\varepsilon \otimes a = a \otimes \varepsilon = \varepsilon$ for all $a \in \mathcal{K}$). This semiring is commutative when $\otimes$ is commutative.
- A module on a semiring is called a semimodule.
- A dioid $\mathcal{K}$ is a semiring which is idempotent ($a \oplus a = a, \ \forall a \in \mathcal{K}$).
- A [commutative, resp. idempotent] semifield is a [commutative, resp. idempotent] semiring whose nonzero elements are invertible.
- We denote $\mathcal{M}_{np}(\mathcal{K})$ the semimodule of (n, p)-matrices with entries in the semiring $\mathcal{K}$. When $n = p$, we write $\mathcal{M}_n(\mathcal{K})$. It is a semiring with matrix product :

$$[AB]_{ij} \overset{\text{def}}{=} [A \otimes B]_{ij} \overset{\text{def}}{=} \bigoplus_k [A_{ik} \otimes B_{kj}] \ .$$

All the entries of the zero matrix are ϵ. The diagonal entries of the identity matrix are e, the other entries being ϵ.

$\mathcal{K}$	$\oplus$	$\otimes$	ε	e	name
$\mathbb{R}^+$	$+$	$\times$	0	1	$\mathbb{R}^+$
$\mathbb{R}^+$	$\sqrt[p]{a^p + b^p}$	$\times$	0	1	$\mathbb{R}_p^+$
$\mathbb{R}^+$	max	$+$	0	1	$\mathbb{R}_{\max,\times}$
$\mathbb{R} \cup \{+\infty\}$	max	$+$	$-\infty$	0	$\mathbb{R}_{\max}$
$\mathbb{R} \cup \{-\infty, +\infty\}$	max	$+$	$-\infty$	0	$\overline{\mathbb{R}}_{\max}$
$\mathbb{R} \cup \dot{\mathbb{R}}$	Argmax($\lvert a\rvert, \lvert b\rvert$)	$\times$	0	1	$\mathbb{S}$
$[a, b]$	max	min	b	a	$[a, b]_{\max,\min}$
$\{0, 1\}$	and	or	0	1	$\mathbb{B}$
$\mathcal{P}(\Sigma^*)$	$\cup$	prod. lat.	$\emptyset$	–	$\mathbb{L}$

In $\mathbb{R}_{\max}$ we have :

- $3 \oplus 2 = 3$,
- $3^2 = 3 \otimes 3 = 3 + 3 = 6$,
- $3/2 = 3 - 2 = 1$.

In $\mathbb{S}$ we have

- $\ominus 2 \triangleq \oplus - 2$
- $\dot{2} = 2 \ominus 2 = (2, -2)$
- $3 \ominus 2 = 3$
- $-3 \oplus 2 = -3$
- $\dot{2} \oplus 3 = 3$
- $\dot{2} \ominus 3 = -3$
- $\dot{2} \oplus 1 = \dot{2} \ominus 1 = \dot{2}$

1.2. MATRICES AND GRAPHS

- With a matrix C in $\mathcal{M}_n(\mathcal{K})$, we associate a precedence graph $\mathcal{G}(C) = (\mathcal{N}, \mathcal{P})$ with nodes $\mathcal{N} = \{1, 2, \cdots, n\}$, and arcs $\mathcal{P} = \{xy \mid x, y \in \mathcal{N},\ C_{xy} \neq \varepsilon\}$.
- The weight of a path π, denoted $\pi(C)$, is the $\otimes$-product of the weights of its arcs. For example we have $xyz(C) = C_{xy} \otimes C_{yz}$.
- The length of the path π (is $\pi(1)$ when $\otimes$ is $+$ (its weight when the arc weigths are all equal to 1)).
- The set of all paths with ends xy and length l is denoted $\mathcal{P}^l_{xy}$. Then, $\mathcal{P}^*_{xy}$ is the set of all paths with ends xy and $\mathcal{P}^*$ the set of all paths.

$$\mathcal{P}^* \stackrel{\text{def}}{=} \bigcup_{l=0}^{\infty} \mathcal{P}^l.\ \ \mathcal{C} = \bigcup_{x} \mathcal{P}^*_{xx}.\ \ \rho \subset \mathcal{P}^*,\ \ \rho(C) \stackrel{\text{def}}{=} \bigoplus_{\pi \in \rho} \pi(C)\ .$$

- We define the star operation by $C^* \stackrel{\text{def}}{=} \bigoplus_{i=0}^{\infty} C^i$.

PROPOSITION 1. *For $C \in \mathcal{M}_n(\mathcal{K})$ we have*

$$\mathcal{P}^l_{xy}(C) = C^l_{xy},\ \mathcal{P}^*_{xy}(C) = C^*_{xy}\ .$$

- If $\mathcal{K} = \mathbb{R}^+$ and $Ce = e$, the equation $p^{n+1} = p^n C$ is the forward Kolmogorov equation.
- If $\mathcal{K} = \mathbb{R}^+$ and $Ce = e$, C^*_{xy} is the probability to reach y starting from x.
- If $\mathcal{K} = \mathbb{R}_{\max}$, the equation $v^{n+1} = v^n C$ is the forward dynamic programming equation.
- If $\mathcal{K} = \mathbb{R}_{\max}$, the eigen equation $\lambda v = vC$ is the ergodic (*average cost by unit of time*) dynamic programming equation.
- If $\mathcal{K} = \mathbb{R}_{\max}$ and $\lambda \leq e$, $C^* = e \oplus C \oplus \cdots C^{n-1}$ and C^*_{xy} is the maximal weight of the paths joining x to y which is finite.

THEOREM 2. *If $\mathcal{K} = \mathbb{R}_{\max}$ and C irreducible, C admits a unique eigenvalue :*

$$\lambda = \bigoplus_{\pi \in \mathcal{C}} \frac{\pi(C)}{\pi(1)} \ ,$$

the columns $(C_\lambda^+)_{.x}$ with $(C_\lambda^+)_{xx} = e$, $C_\lambda = C/\lambda$ and $C^+ \triangleq CC^*$ generate the corresponding eigensemidodule.

PROPOSITION 3. *If $\mathcal{K} = \mathbb{R}_{\max}$, $\lambda(C) \leq e$,*

$$x = Cx \oplus b$$

has a smallest solution given by

$$x = A^* b$$

moreover if $\lambda < e$ the solution is unique.

1.3. COMBINATORICS - CRAMER FORMULAS

THEOREM 4. *The solution of the system* $Ax \oplus b' = A'x \oplus b$ *in* $\mathbb{R}^{+}_{\max,\times}$ *exists and is unique and given by*[1]

$$x = (A \ominus A')^{\sharp}(b \ominus b') / \det\left(A \ominus A'\right) \ ,$$

$$\det(A) = \bigoplus_{\sigma} \operatorname{sgn}(\sigma) \bigotimes_{i=1}^{n} A_{i\sigma(i)} \ , \quad A^{\sharp}_{ij} = \operatorname{cofactor}_{ji}(A) \ ,$$

when and only when $x \geq 0$.

$$\begin{cases} \max(x_1, 3x_2) = 5, \\ \max(4x_1, 2x_2) = 6, \end{cases} \qquad \det(A) = 2 \ominus 12 = \ominus 12, \quad \det\begin{bmatrix} 5 & 3 \\ 6 & 2 \end{bmatrix} = \ominus 18,$$

$$\det\begin{bmatrix} 1 & 5 \\ 4 & 6 \end{bmatrix} = \ominus 20, \quad x_1 = 3/2, \ x_2 = 5/3, \quad \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} 3/2 \\ 5/3 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix} \ .$$

[1] The computation are done in $\mathbb{S}$.

1.4. ORDER - RESIDUATION

- A dioid is complete when the $\otimes$ is distributive with the infinite $\oplus$.
- A complete dioid is a lattice ($\oplus$ upper bound, $\wedge$ lower bound).
- $\mathcal{D}$ and $\mathcal{C}$ complete dioids $f : \mathcal{D} \to \mathcal{C}$. f is residuable if $\{x \mid f(x) \leq y\}$ admits an maximal element denoted by $f^\sharp(y)$.
- f residuable $\Leftrightarrow$ $f \circ f^\sharp \leqslant I_{\mathcal{C}}$ and $f^\sharp \circ f \geqslant I_{\mathcal{D}}$.

PROPOSITION 5. *In* $\mathbb{R}_{\max}$ $f(x) = A \otimes x$ *is residuable and*

$$f^\sharp(y)_j = (A \backslash y)_j \triangleq \bigwedge_i y_i / A_{ij} \ .$$

Thanks to this proposition the linear system $Ax = b$ can be solved much more easily than with the Cramer formula :

- compute the upper solution of $Ax \leq b$,
- check if it is a solution.

1.5. GEOMETRY - IMAGE, KERNEL, INDEPENDENCE

X and Y semodules, $F : X \to Y$ a linear map.

- $\mathrm{Im}(F) = \{F(x) \mid x \in X\}$.
- $\ker(F) = \left\{\left(x^1, x^2\right) \in X^2 \mid F\left(x^1\right) = F\left(x^2\right)\right\}$. It is a congruence that is an equivalent relation $\mathcal{R} \subset X \times X$ which is a semimodule.

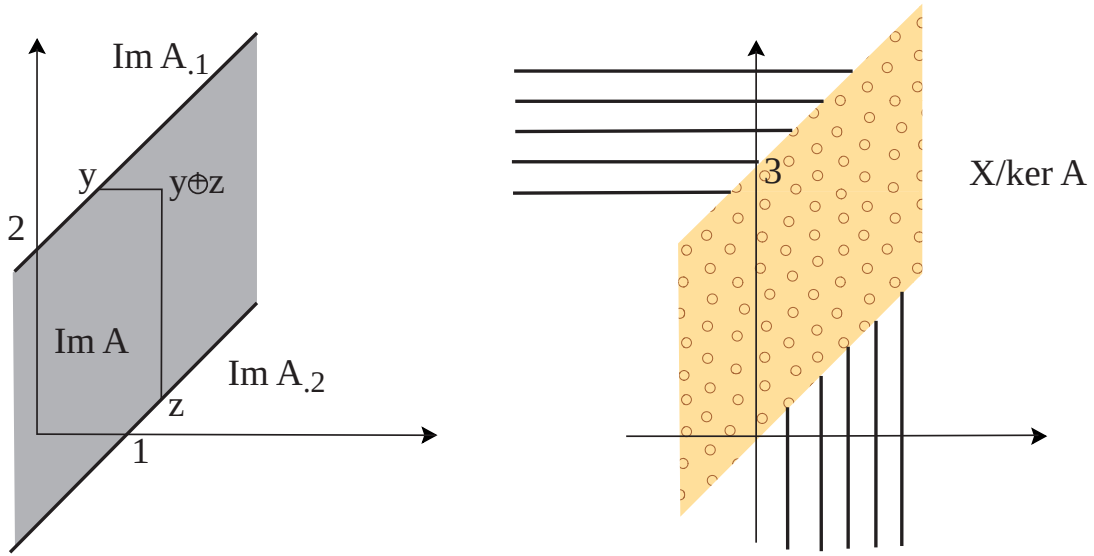


FIGURE 1. Image and Kernel.

- A generating family $\{x_i\}_{i\in I}$ of a semimodule X is a subset of X :

$$\forall x \in X \;\; \exists\, \{\alpha_i\}_{i\in I} \in \mathcal{K} : \;\; x = \bigoplus_{i\in I} \alpha_i x_i .$$

- "Convex" semimodule admits a unique generating family (the set of the extremal points).
- The family $\{x_i\}_{i\in I}$ is independent if

$$\bigoplus_{i\in I} \alpha_i x_i = \bigoplus_{i\in I} \beta_i x_i \Longrightarrow \alpha_i = \beta_i, \quad \forall i \in I \;\; .$$

- An independent generating family is called a basis. A semimodule admitting a basis is called free.

$$p_1 = \begin{bmatrix} \varepsilon \\ e \\ e \end{bmatrix}, \;\; p_2 = \begin{bmatrix} e \\ \varepsilon \\ e \end{bmatrix}, \;\; p_3 = \begin{bmatrix} e \\ e \\ \varepsilon \end{bmatrix}, \quad p_1 \oplus p_2 = p_2 \oplus p_3 \;\; .$$

2. MAXPLUS SCALARS AND MATRICES IN SCILAB

The maxplus toolbox is an external contribution of Scilab which must be builded (compiled) and loaded. It is available as a contribution to scilab at address www-rocq.inria.fr/scilab. It adds maxplus arithmetic to scilab.

Let us show first the max-plus types of Scilab and their interaction with standard objects.

```
-->a=2
 a  =
    2.
-->typeof(a)
 ans  =
   constant
```

A maxplus matrix is created by the instruction maxplus which has the abbreviation #.

```
-->b=#(3)
 b  =
 3
-->typeof(b)
 ans  =
   maxplus full
```

We can change a maxplus matrix in a standard matrix by the instruction plustimes.

```
-->plustimes(b)
 ans  =
    3.
-->typeof(ans)
 ans  =
   constant
```

The maxplus zero is $-\infty$. It is printed with the character dot.

```
-->%0
 %0  =
 .
```

```
-->typeof(%0)
 ans  =
   maxplus full
 -->%inf
 %inf  =
   Inf
-->typeof(%inf)
 ans  =
   constant
```

The maxplus unity is equal to 0.

```
-->%1
 %1  =
 0
```

The maxplus operations overload the standard operations.

```
-->b
 b  =
 3
-->typeof(b)
 ans  =
   maxplus full
-->b + %0
 ans  =
 3
-->b * %1
 ans  =
 3
-->b + b
 ans  =
 3
-->b * b
 ans  =
 6
-->b / b
```

```
 ans  =
 0
-->b & %0
 ans  =
-->b == b
 ans  =
  T
-->b <> b
 ans  =
  F
-->b >= b
 ans  =
  T
-->b > b
 ans  =
  F
```

As soon as an operand has a maxplus the result inherit of a maxplus type.

```
-->b+3
 ans  =
 3
-->typeof(ans)
 ans  =
    maxplus full
-->b*3
 ans  =
 6
```

We have different way to create maxplus matrices :

- from a max-plus scalar

```
-->c=[b,4;5,6]
 c  =
!3  4  !
!5  6  !
-->typeof(c)
```

```
 ans  =
   maxplus full
```

- from a standard matrix

```
-->d=[1,2;3,4]
 d  =
!   1.    2. !
!   3.    4. !
-->typeof(d)
 ans  =
    constant
-->e=#(d)
 e  =
!1  2  !
!      !
!3  4  !
-->typeof(e)
 ans  =
    maxplus full
```

- by extraction

```
-->f=e(1,:)
 f  =
!1  2  !
-->typeof(f)
 ans  =
   maxplus full
```

- by insertion

```
-->e(5,5)=6
 e  =
!1  2  .  .  .  !
!3  4  .  .  .  !
!.  .  .  .  .  !
!.  .  .  .  .  !
!.  .  .  .  6  !
-->typeof(e)
 ans  =
   maxplus full
```

There are special instructions to create important particular maxplus matrices.

```
-->%ones(2,5)
 ans  =
!0  0  0  0  0  !
!0  0  0  0  0  !
-->%eye(2,5)
 ans  =
!0  .  .  .  .  !
!.  0  .  .  .  !
-->g=%zeros(2,5)
 g  =
(2,5) zero sparse matrix
```

There exists sparse maxplus matrices.

```
-->typeof(g)
 ans  =
   maxplus sparse
```

We can change a sparse matrix in a full one.

```
-->full(g)
 ans  =
!.  .  .  .  .  !
!.  .  .  .  .  !
-->typeof(ans)
 ans  =
   maxplus full
```

We can change a full matrix in a sparse one.

```
-->%ones(2,5)
 ans  =
!0  0  0  0  0  !
!0  0  0  0  0  !
```

```
-->sparse(ans)
 ans  =
(2,5) sparse matrix

(    1,    1)    0.
(    1,    2)    0.
(    1,    3)    0.
(    1,    4)    0.
(    1,    5)    0.
(    2,    1)    0.
(    2,    2)    0.
(    2,    3)    0.
(    2,    4)    0.
(    2,    5)    0.
-->typeof(ans)
 ans  =
   maxplus sparse
```

The standard operations on matrices are overloaded (be careful with & which means here min element wise).

```
-->c
 c  =
!3  4  !
!5  6  !
-->d
 d  =
!   1.    2. !
!   3.    4. !
-->c + d
 ans  =
!3  4  !
!5  6  !
-->c * c
 ans  =
!9   10  !
```

```
!11  12  !
-->c / c
 ans  =
!0  -2  !
!2  0   !
-->d & c
 ans  =
!   1.    2. !
!   3.    4. !
-->star(c)
 ans  =
!Inf  Inf  !
!Inf  Inf  !
-->c == c
 ans  =

! T T !
! T T !
```

```
-->c <> c
 ans  =
! F F !
! F F !
-->d > c
 ans  =
! F F !
! F F !
```

The standard scilab column concatenation is overloaded.

```
-->h=[e,e]
 h  =
!1  2 . . . 1  2 . . .!
!3  4 . . . 3  4 . . .!
!.  . . . . .  . . . .!
!.  . . . . .  . . . .!
!.  . . . 6 .  . . . 6!
```

The row concatenation is overloaded.

```
-->i=[e;e]
 i  =
!1  2  .  .  .  !
!3  4  .  .  .  !
!.  .  .  .  .  !
!.  .  .  .  .  !
!.  .  .  .  6  !
!1  2  .  .  .  !
!3  4  .  .  .  !
!.  .  .  .  .  !
!.  .  .  .  .  !
!.  .  .  .  6  !
-->size(i)
 ans  =
!   10.    5. !
```

The standard extraction is overloaded.

```
-->i([1,3],:)
 ans  =
!1  2  .  .  .  !
!.  .  .  .  .  !
```

Spectral elements of a matrix can be computed efficiently by the Howard algorithm.

```
-->c
 c  =
!3  4  !
!5  6  !
-->[chi,v]=howard(c)
 v  =
!4  !
!6  !
```

```
 chi  =
!6  !
!6  !
```

chi is the eigenvalue, v is the eigenvector.

```
-->chi(1)*v==c*v
 ans  =
! T !
! T !
```

These spectral elements give the asymptotic behavior of maxplus dynamical system.

```
-->x=[%1;%0]
 x  =
!0  !
!.  !
```

```
-->[x,c*x,c*c*x,c*c*c*x]
 ans  =
!0  3  9   15 !
!.  5  11  17 !
```

In practice, Howard is a ”linear” algorithm with the number of arcs. Let us compute the time to compute spectral elements of a 10000x10000 matrix.

```
-->timer();
-->[chi,v]=howard(..
#(sprand(10000,10000,0.0005).
+0.001*speye(10000,10000)));
-->timer()
 ans  =
    3.32
```

3. Input-Output Max-Plus Linear Systems

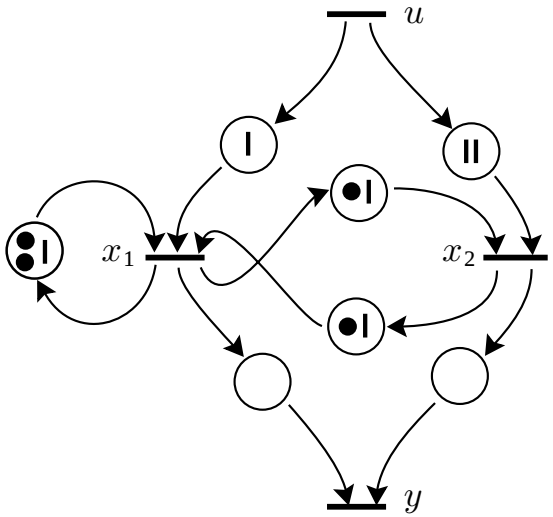


FIGURE 2. Event Graph

$$\begin{cases} x_k^1 = \max(1 + x_{k-2}^1, 1 + x_{k-1}^2, 1 + u_k) \\ x_k^2 = \max(1 + x_{k-1}^1, 2 + u_k) \\ y_k = \max(x_k^1, x_k^2) \end{cases} \qquad \begin{cases} x_t^1 = \min(x_{t-1}^1 + 2, x_{t-1}^2 + 1, u_{t-1}) \\ x_t^2 = \min(x_{t-1}^1 + 1, u_{t-2}) \\ y_t = \min(x_t^1, x_t^2) \end{cases}$$

3.1. Transfer Functions

$$D = \bigoplus_{k\in\mathbb{Z}} d_k \gamma^k,\ c_k \in \overline{\mathbb{Z}}_{\max}\ .\quad C = \bigoplus_{t\in\mathbb{Z}} c_t \delta^t,\ d_t \in \overline{\mathbb{Z}}_{\min}\ .$$

$$\gamma : (d_k)_{k\in\mathbb{Z}} \mapsto (d_{k-1})_{k\in\mathbb{Z}}\ .\quad \delta : (c_t)_{t\in\mathbb{Z}} \to (c_{t-1})_{t\in\mathbb{Z}}\ .$$

$$\begin{cases} X = \gamma A X \oplus BU\ , \\ Y = CX\ . \end{cases} \qquad \begin{cases} X = \delta \tilde{A} X \oplus \tilde{B} U\ , \\ Y = \tilde{C} X\ . \end{cases}$$

$$Y = C\,(\gamma A)^*\,BU\ .\qquad Y = \tilde{C}\left(\delta \tilde{A}\right)^*\tilde{B}U\ .$$

$C\,(\gamma A)^*\,B$, $\tilde{C}\left(\delta \tilde{A}\right)^*\tilde{B}$ are transfer functions of the system. They are rational maxplus [resp. minplus] series.

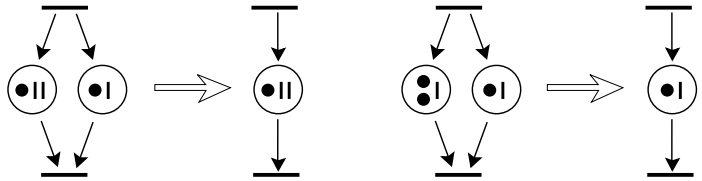


FIGURE 3. Event graph simplification.

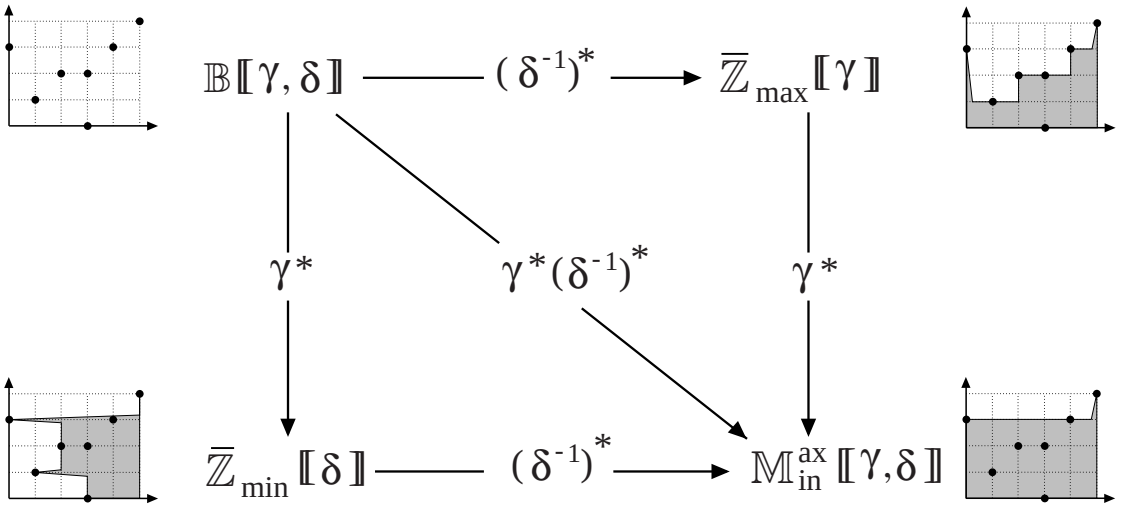


FIGURE 4. Modellings

$$\begin{cases} X = AX \oplus BU \ , \\ Y = CX \ , \end{cases} \qquad A = \begin{bmatrix} \gamma^2\delta & \gamma\delta \\ \gamma\delta & \varepsilon \end{bmatrix}, \ B = \begin{bmatrix} \delta \\ \delta^2 \end{bmatrix}, \quad C = \begin{bmatrix} e & e \end{bmatrix} \ .$$

$$Y = CA^* BU = \delta^2 (\gamma\delta)^* U \ .$$

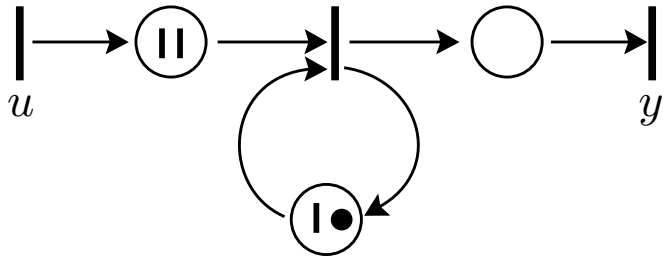


FIGURE 5. Equivalent system 2.

3.2. Rational Series.

$S \in \mathbb{M}_{\text{in}}^{\text{ax}}[\![\gamma, \delta]\!]$ is :

1. rational if it belongs to the closure $\{\varepsilon, e, \gamma, \delta\}$ with respect of finite number of operations $\oplus$, $\otimes$ and $*$;
2. realizable if it can be written :

$$S = C\,(\gamma A_1 \oplus \delta A_2)^*\,B\ ,$$

   with C, A_1, A_2, B boolean ;
3. periodic if it exists p, q polynomials and m monomial such that :

$$S = p \oplus q m^*\ .$$

Theorem 6.

Rational $\Leftrightarrow$ *Realizable* $\Leftrightarrow$ *Periodic.*

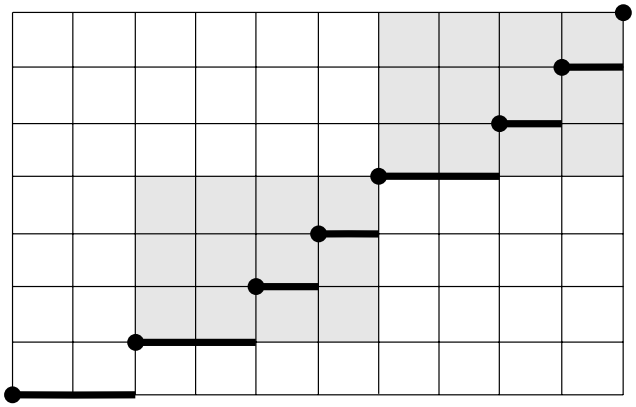


FIGURE 6. A rational series.

4. DYNAMICAL MAXPLUS LINEAR SYSTEMS IN SCILAB

In Scilab Maxplus linear dynamical systems are represented in implicit state form, that is using the canonical form

$$X(n) = DX(n) \oplus AX(n-1) \oplus BU(n), \quad Y(n) = CX(n)\ ,$$

and thus defined by the quadruple (A, B, C, D).

These quadruples can be manipulated with the standard Scilab operators which are once more overloaded.

Creation of system.

```
-->s1=mpsyslin(..
[1,2;3,4],[0;0],[0,0])
 s1  =
   |0  .|  |1  2|   |0|
x =|.  0|x+|3  4|x'+|0|u

y = | 0  0 |x
```

We have access to the different fields.

```
-->s1('D')
 ans  =
!0  .  !
!      !
!.  0  !

-->s1('X0')
 ans  =
```

(2,1) zero sparse matrix

The system use sparse matrix. But we can transform it in such way that it uses full matrices.

```
-->s1=full(s1)
 s1  =
   |0  .|  |1  2|   |0|
x =|.  0|x+|3  4|x'+|0|u

y = |0  0|x
-->s1('X0')
 ans  =
!.  !
!.  !
```

Using the star operation we can explicit the system.

```
-->explicit(s1)
 ans  =
    | 1  2 |    | 0 |
x = | 3  4 |x'+ | 0 |u

y = | 0  0 |x
```

In order to illustrate composition of systems let us define another dynamical system.

```
-->s2=mpsyslin(..
[1,2,3;4,5,6;7,8,9],..
[0;0;0],[0,0,0])
 s2  =
   |0 . .|  |1  2  3|   |0|
x =|. 0 .|x+|4  5  6|x'+|0|u
   |. . 0|  |7  8  9|   |0|

y = |0  0  0|x

-->s2=sparse(s2);
```

The maxplus linear system operators have the same syntax as the matrix ones.

- Diagonal composition

```
-->s4=s1|s2
 s4  =
    | 0  .  .  .  .  |   | 1  2  .  .  .  |     | 0  .  |
    | .  0  .  .  .  |   | 3  4  .  .  .  |     | 0  .  |
x =| .  .  0  .  .  |x+ | .  .  1  2  3  |x'+  | .  0  |u
    | .  .  .  0  .  |   | .  .  4  5  6  |     | .  0  |
    | .  .  .  .  0  |   | .  .  7  8  9  |     | .  0  |

     | 0  0  .  .  .  |
y =  | .  .  0  0  0  |x
```

- Parallel composition

```
-->s3=s1+s2;
```

- Series composition

```
-->s1*s2;
```

- Input in common

```
-->[s1;s2];
```

- Output addition

```
-->[s1,s2];
```

- Feedback composition

```
-->s1/.s2 ;
```

- Extraction

```
-->s4=full(s4)
    | 0  .  .  .  .  |   | 1  2  .  .  .  |    | 0  .  |
    | .  0  .  .  .  |   | 3  4  .  .  .  |    | 0  .  |
x =| .  .  0  .  .  |x+ | .  .  1  2  3  |x'+ | .  0  |u
    | .  .  .  0  .  |   | .  .  4  5  6  |    | .  0  |
    | .  .  .  .  0  |   | .  .  7  8  9  |    | .  0  |

    | 0  0  .  .  .  |
y =| .  .  0  0  0  |x
-->s4(1,1)
    | 0  .  .  .  .  |    | 1  2  .  .  .  |    | 0  |
    | .  0  .  .  .  |    | 3  4  .  .  .  |    | 0  |
x =| .  .  0  .  .  |x +  | .  .  1  2  3  |x'+ | .  |u
    | .  .  .  0  .  |    | .  .  4  5  6  |    | .  |
    | .  .  .  .  0  |    | .  .  7  8  9  |    | .  |

y =| 0  0  .  .  .  |x
```

We can simulate a maxplus linear system.

```
-->y=simul(s1,[1:10])
 y  =
!1  5  9  13  17  21  25  29  33  37 !
```

5. Applications

Troughput of an event graph. $A\,(\gamma,\delta)$ irreducible,

$$\lambda = \max_{m\in C\in\mathcal{C}} \frac{m_\delta}{m_\gamma}\ , \quad m = \gamma^{m_\gamma}\delta^{m_\delta}\ .$$

Feedback design.

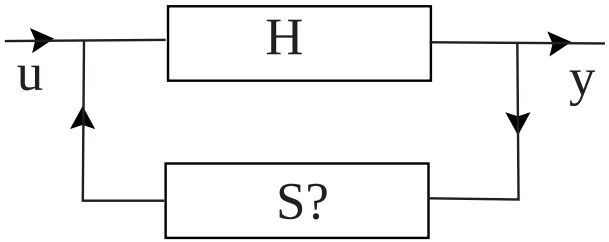


Figure 7. Feedback.

$$Y = H\,(U \oplus SY) = (HS)^*\,HU\ .$$

Latest entrance time to achieve an objective.

$$Z = CA^*BU \leqslant Y\ , \quad U = CA^*B\backslash Y\ , \quad \begin{cases} \xi = A\backslash\xi \wedge C\backslash Y\ , \\ Y = B\backslash\xi\ . \end{cases}$$

5.1. PRODUCTION SYSTEMS IN SCILAB

Periodic flowshop.

- Parts are carried on pallets. When the tasks on a part are finished the pallet start another cycle with another part of the same class. Each part visit machines in sequence never coming back to the same machine.
- To make a task we can have many machines (called a class). The machines visit the parts in sequence never coming back to the same part.
- We define the flowshop by a matrix describing the resources used and the processing times.
- Each line of the matrix is associated to a machine class.
- Each columns to a part class.
- The entries of the matrix are the processing times.
- If a part class does not need a machine class the corresponding entry is $-\infty$.

```
-->PT=[#(2),3.9,     ,1.2,1.2,1.2]
 PT  =
!2    3.9  0.95  1.1  0.7  1.4  !
!.    .    2     1.2  .    1.7  !
!3.7  .    2.2   .    6.4  .    !
!.    .    2     .    1    1    !
!1.7  3.1  3     .    1.3  .    !
!0.5  3.2  4.3   1.9  1.6  0.4  !
!1    1    1     1    1    1    !
!1.5  1.5  1.5   1.2  1.2  1.2  !
-->[nmach,npiece]=size(PT)
 npiece  =
    6.
 nmach  =
    8.
```

Let us give the machine number in each class.

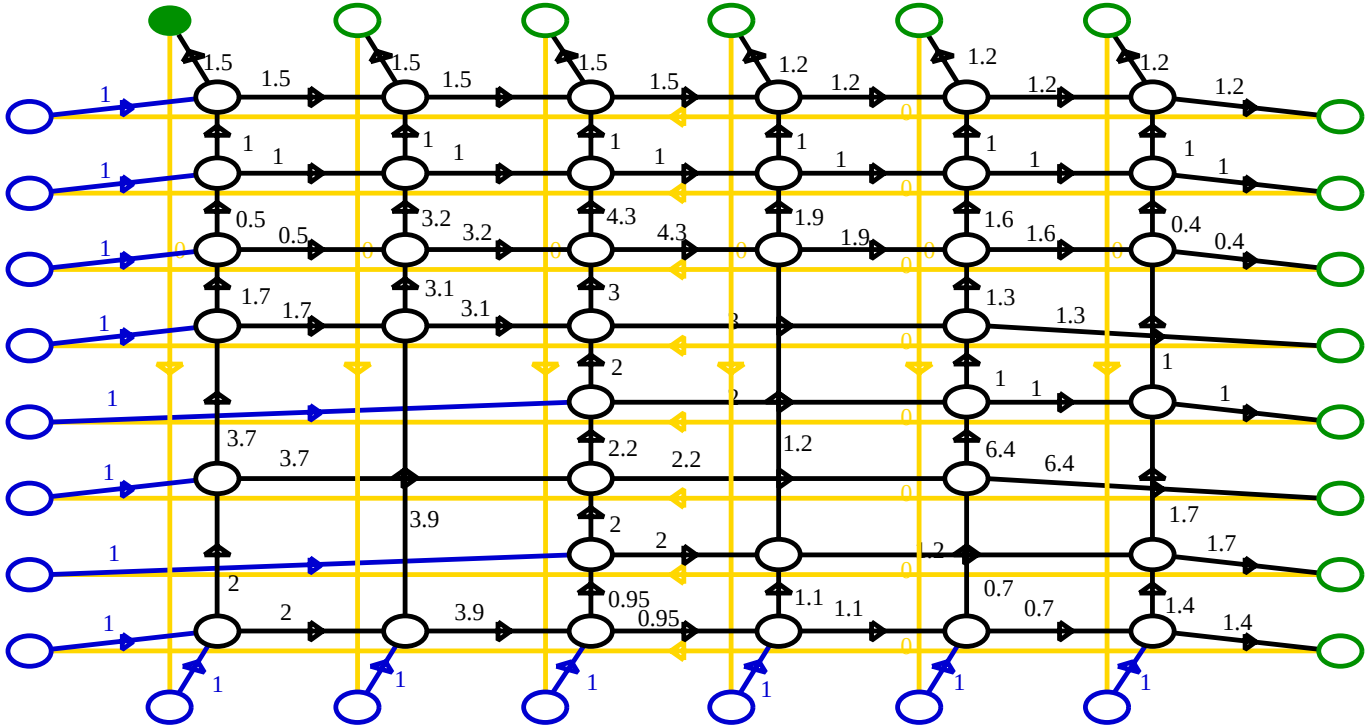
```
-->nm=ones(1,nmach)
 nm  =

!   1.    1.    1.    1.    1.    1.    1.    1. !
```

Let us give the piece number in each class.

```
-->np=ones(1,npiece)
 np  =
!   1.    1.    1.    1.    1.    1. !
```

Let us show a graphic representation of the corresponding cyclic flowshop.

```
-->[g,T,N]=flowshop_graph(PT,nm,np,50);
```

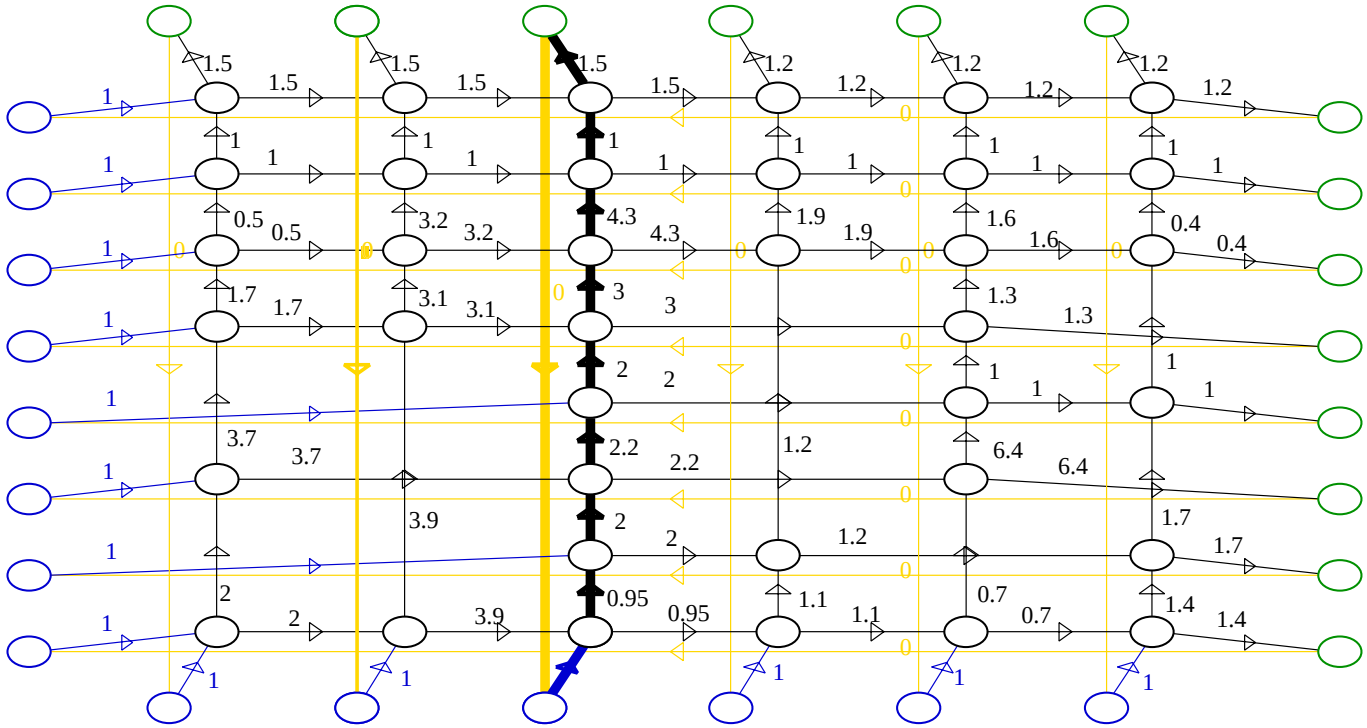


Let us compute the throughput of the flowshop by the Howard algorithm using the following theorem.

```
-->[chi,v]=semihoward(T,N);
-->chi'
 ans  =
         column  1 to 5
!16.95  16.95  16.95  16.95  16.95  !
-->v'
 ans  =
         column  1 to 5
!23.8  6.85  6.85  19.45  2.5  !
```

Let us show the critical circuit.

```
-->show_cr_graph(g);
```

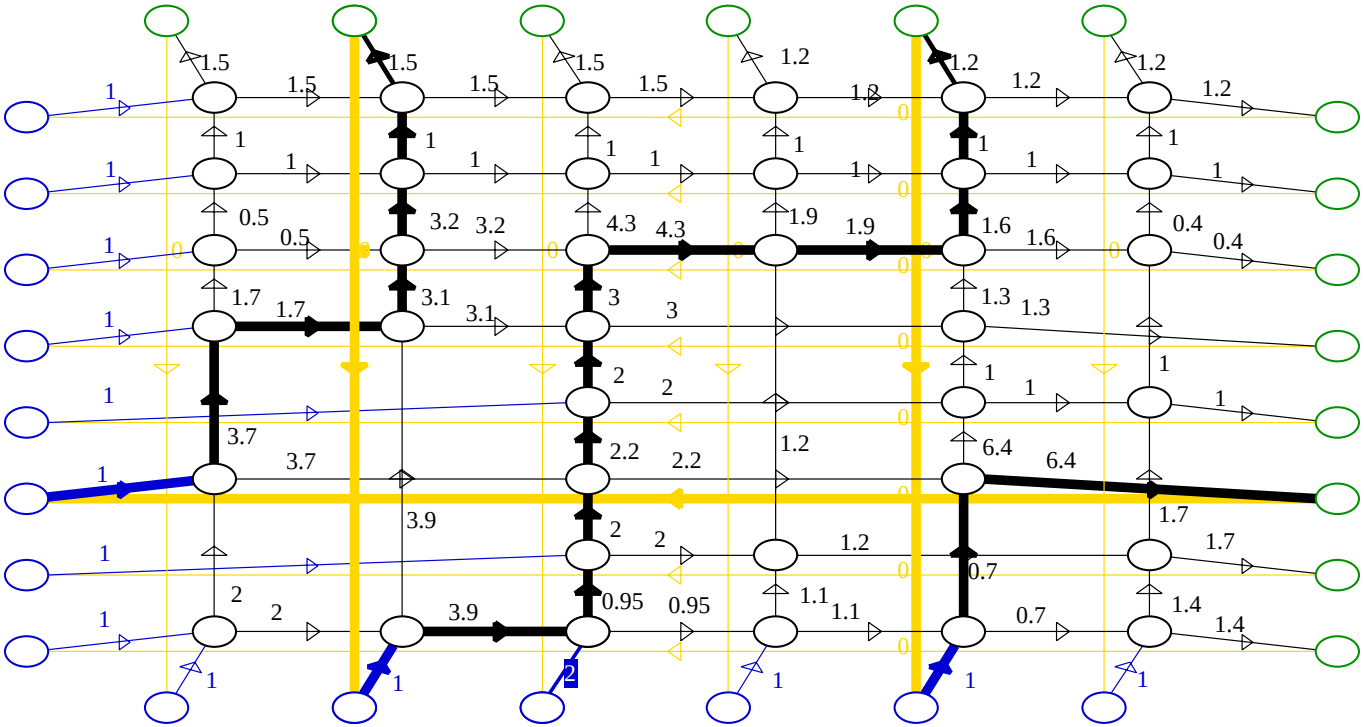


5.2. Optimization of the pallet numbers

Let us optimize by hand the number of pallet in the system. For that we add pallet in critical circuit with vertical arcs. The presence of such arcs means that machines are waiting for a part.

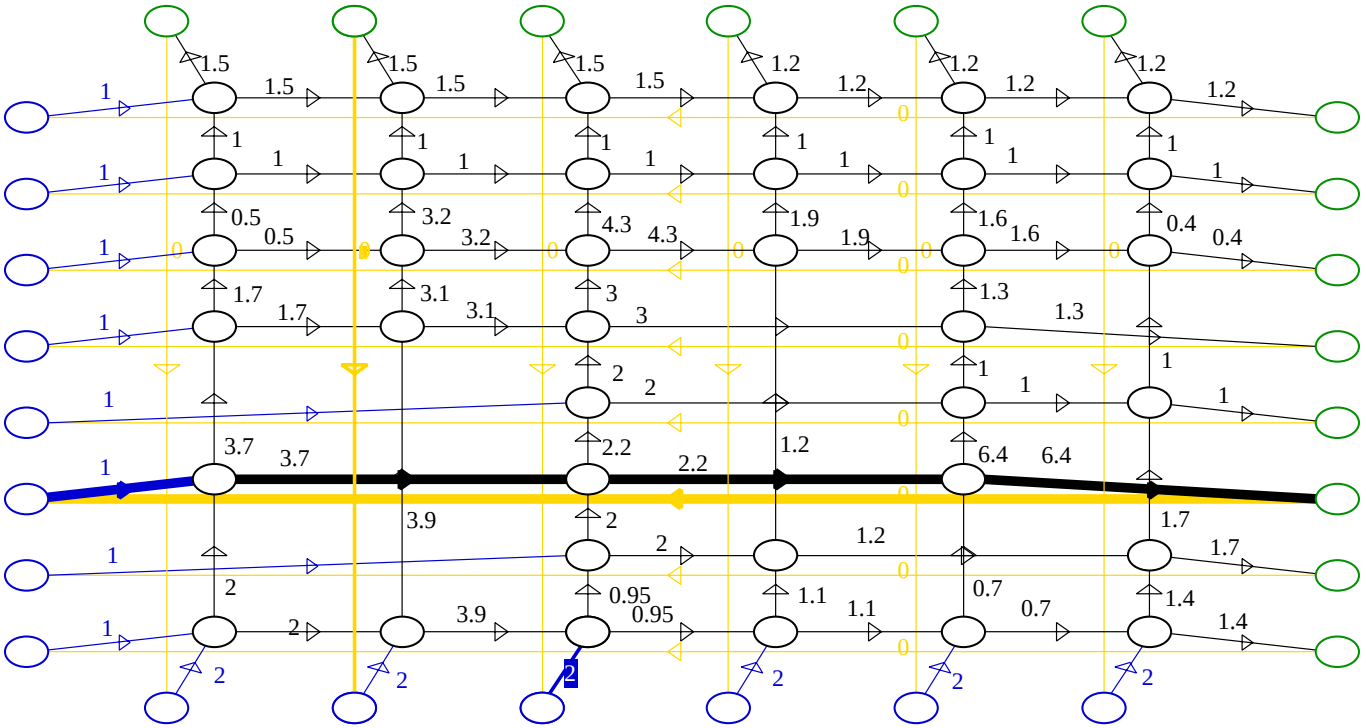
One pallet more for the third part class.

```
-->pnb=[1,16,28,46,58,74];
-->g('length')(pnb)=[1,1,2,1,1,1];
-->show_cr_graph(g);
```



Two pallets for all part classes.

```
-->g('length')(pnb)=[2,2,2,2,2,2];
-->show_cr_graph(g);
```



5.3. SIMULATION OF FLOWSHOP IN SCILAB

The flowshop is seen as a feedback system.

The feedback corresponds to :

- the arcs on machine saying that after achieving a task cycle the machines starts a new cycle,
- the arcs on pallets saying that as soon as all the tasks on a part has been achieved the pallets take another part.

The open-loop system corresponds to the other arcs of the flowshop.

To build the dynamic of the cyclic flowshop, we build the open-loop system, the feedback system and we compose the two using the implicit state form description of dynamical systems.

Implicit state representation of the open-loop flowshop.

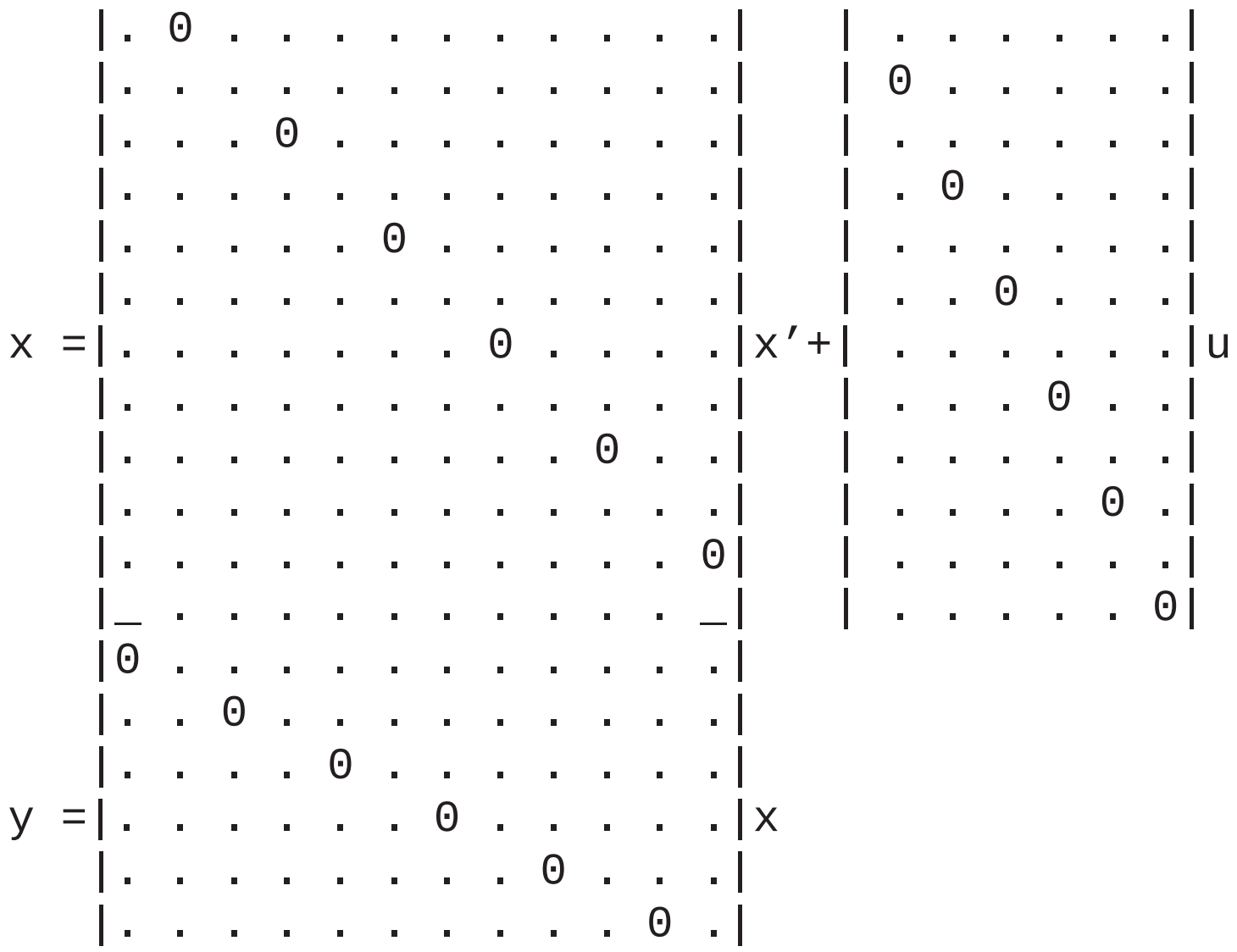
```
-->s=flowshop(PT)
 s  =
 x(n)=Dx(n)+Ax(n-1)+Bu(n)
 A=
(   48,   48) zero sparse matrix
 B=
(   48,   14) sparse matrix
 D=
(   48,   48) sparse matrix
 C=
(   14,   48) sparse matrix
```

The machine controller.

```
-->nm
 nm  =
!   1.    1.    1.    1.    1.    1.    1.    1. !
-->fbm=shift(nm(1),0) ;
-->for i=1:nmach-1, fbm=fbm|shift(nm(i),0) ; end ;
```

The pallet controller.

```
-->np
 np  =
!   1.    1.    1.    1.    1.    1. !
-->//
-->fbp=shift(np(1),0);
-->for i=1:npiece-1, fbp=fbp|shift(np(i),0) ; end ;

-->fbp
 fbp  =
```


x =
y =
x′+
u
x

The complete feedback system.

```
-->sb=s/.(fbp|fbm);
```

Reducing a system and putting it in explicit form.

```
-->sbs=explicit(sb);
```

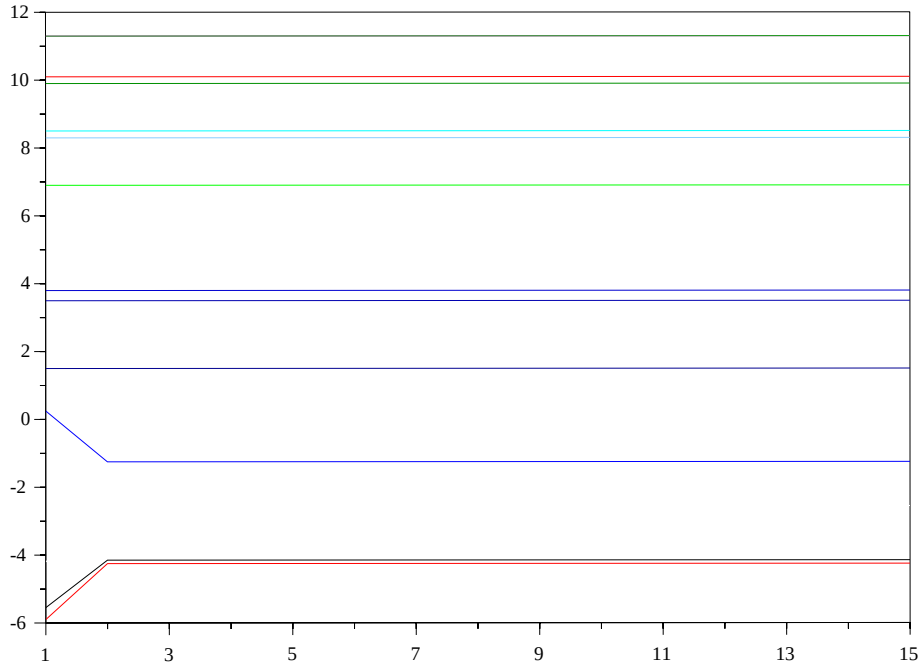
Simulation of the feedback system

```
-->u=ones(nmach+npiece,1)*(1:100);
-->y=simul(sbs,u);
-->y(:,100)'
 ans  =
         column 1 to 5
!1690.85  1693.75  1701.9  1703.5  1705.1  !
```

Plotting the transient part of the outputs without the stationary drift term.

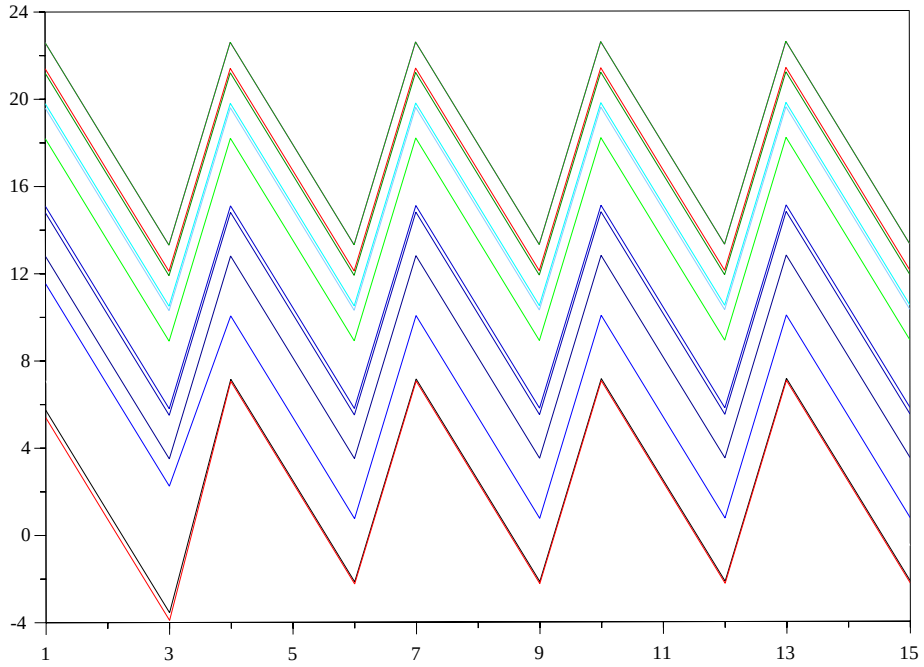
Periodicity 1 case.

```
-->chi=howard(sbs('A'));
-->chit=plustimes(chi(1))*[1:100];
-->y=plustimes(y)-ones(nmach+npiece,1)*chit;
-->xbasc(); plot2d(y(:,[1:15])');
```



Periodicity 3 case.

```
-->np=3*ones(1,6); nm=3*ones(1,8);
-->xbasc(); plot2d(y(:,[1:15])');
```



REFERENCES

[1] M. Akian : “Densities of idempotent measures and large deviations”, AMS 99,

[2] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat : “Synchronization and Linearity”, Wiley (1992).

[3] T.S. Blyth, M.J Janowitz “Residuation theory” Pergamon Press, Oxford (1972).

[4] P. Butkovic “Strong regularity of matrices - a survey of results”, Disc. Ap. Math. n. 48, p.45-68 (1994).

[5] G. Cohen, S. Gaubert and J.P. Quadrat : “Linear Projectors in the max-plus Algebra” IEEE Mediterranean Conference on Control, Chypre (August 1997).

[6] R. Cuninghame-Green : “Minimax Algebra”, L.N. on Economics and Math. Systems N. 166, Springer Verlag (1979).

[7] M. Gondran, M. Minoux : “Graphs and Algorithms” J. Wiley & Sons (1986)

[8] J.S. Golan :“The theory of semirings with applications in mathematics and theoretical computer sciences” Pitman (1992)..

[9] J. Gunawerdena (editor) (1998). *Idempotency.* Cambridge University Press.

[10] M.L. DubreuilL-Jacotin, L. Lesieur R. Croisot “‘Théorie des treillis des structures algébriques ordonnées et des treillis géométriques”, Gauthier-Villars,(1953).

[11] S. Gaubert : “Théorie des systèmes linéaires dans les dioides”, Thesis dissertation, École des Mines de Paris, (1992).

[12] V.P. Maslov and S.N. Samborskii : “Idempotent Analysis”, AMS (1992).

[13] V.P. Maslov : “Méthodes Opératorielles”, Éditions MIR, Moscou (1987).

[14] E. Pap : “Null-Additive Set Functions”, Mathematics and Its applications 337, Kluwer academic Publishers, Dordrecht (1995).
[15] U. Zimmerman : “Linear and combinatorial optimization inn ordered algebrai structures” Annals of discrete math. N.10, North-Holland, (1981).

Other informations and articles about this max-plus algebra are available from the following web pages :

- http://www-rocq.inria.fr/scilab/cohen,
- http://amadeus.inria.fr/gaubert,
- http://www-rocq.inria.fr/scilab/quadrat,
- http://www.cs.rug.nl/ rein/alapedes/alapedes.html