MATRICES de la THÉORIE DES GRAPHES en J

ROBERT COQUIDÉ

Lors d'une étude, les questions auxquelles permettent parfois de répondre les méthodes de la *Recherche Opérationnelle* et, en particulier, celles de la *Théorie des Graphes*, sont de différents types :

1) Problème d'existence : *Existe-t-il une solution* ?
2) Problème de construction : *Peut-on élaborer explicitement une solution* ?
3) Problème de dénombrement : *Quel est le nombre des solutions possibles* ?
4) Problème d'énumération : *Peut-on élaborer explicitement l'ensemble de toutes les solutions* ?
5) Problème d'amélioration : *Une solution étant connue, peut-on en déduire une autre qui soit "meilleure"* ? Il faut pouvoir "*comparer*" les 2 solutions selon des critères prédéfinis. Cela ne suppose pas l'existence d'un optimum, ni une mesure de proximité de l'optimum, s'il existe. Une "*méthode heuristique*" permet parfois de répondre à ce type de question.
6) Problème d'optimisation : *Peut-on fournir la solution optimale* ? Cette question suppose une "*valuation*" des solutions, l'*existence* de cet optimum, la possibilité de le "*comparer*" à 1 solution quelconque selon des critères prédéfinis définissant, si possible, une *relation d'ordre total*.

Certaines méthodes de la *Théorie des Graphes* utilisent, pour élaborer des réponses, des "*tableaux matriciels*" particuliers et des "*produits matriciels*" (cas particuliers des produits intérieurs) spécialement adaptés.

Pour illustrer cet article (et tenter de le rendre moins indigeste !) nous prendrons l'exemple de 5 villages de montagne (graphe orienté de 5 sommets **A B C D E**), les éventuelles et tortueuses routes à sens unique ou non (les arcs du graphe) qui les relient, les Km de route parcourus (valeurs des arcs), les chemins possibles (respectant les sens uniques) permettant de relier 2 villages, les circuits possibles (chemins dont l'origine – sommet de départ – et l'extrémité – sommet d'arrivée – sont confondus).

Ne pas confondre la longueur d'un chemin (nombre d'arcs parcourus) et sa valeur (somme des Km parcourus le long des arcs empruntés).

Un chemin ou un circuit sont dits élémentaires s'ils ne passent pas 2 fois par le même sommet (hormis l'origine ou l'extrémité du circuit).

Un chemin ou un circuit sont dits simples s'ils n'empruntent pas 2 fois le même arc (élémentaire implique simple ; réciproque fausse).

Un chemin ou un circuit sont dits hamiltonniens s'ils passent une fois et une seule par chaque sommet (hormis l'origine ou l'extrémité du circuit).

Un chemin ou un circuit sont dits eulériens s'ils empuntent une fois et une seule chaque arc du graphe.

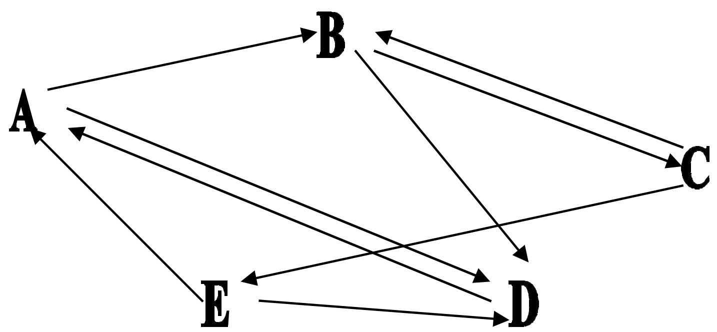
I) MATRICE BOOLÉENNE – PRODUIT MATRICIEL BOOLÉEN

[mb1 =: 0 EDIAG ? MBOOL 5 5

0 1 0 1 0 NB. on crée une matrice booléenne
0 0 1 1 0 NB. aléatoire en imposant des **0** sur la
0 1 0 0 1 NB. diagonale (ce qui évite les **boucles**
1 0 0 0 0 NB. ou **circuits** de **longueur 1**).
1 0 0 1 0

C'est la matrice d'incidence du graphe qui suit :

Un **1** signifie l'existence d'un **arc** (route directe) entre 2 villages (ou chemin de **longueur 1**). Un **0** signifie l'inexistence d'un tel **arc**.



Graphe de matrice d'incidence mb1

Voici comment lire une telle matrice booléenne:

mb1	A	B	C	D	E
A	0	1	0	1	0
B	0	0	1	1	0
C	0	1	0	0	1
D	1	0	0	0	0
E	1	0	0	1	0

[mb2 =: mb1 PMB mb1 NB. utilisation du produit
1 0 1 1 0 NB. matriciel booléen **PMB**
1 1 0 0 1
1 0 1 1 0 NB. dans **mb2**, un **1** signale l'existence d'un chemin au moins de
0 1 0 1 0 NB. **longueur 2** (2 arcs) entre 2 villages . Un **0** traduit l'inexistence
1 1 0 1 0 NB. d'un tel chemin. La diagonale concerne les circuits.

[mb3 =: mb1 PMB mb2
1 1 0 1 1
1 1 1 1 0
1 1 0 1 1 NB. **mb3** : existence des chemins et circuits de longueur **3**
1 0 1 1 0
1 1 1 1 0

[mb4 =: mb1 PMB mb3

1 1 1 1 0
1 1 1 1 1
1 1 1 1 0 NB. **mb4** : existence des chemins et circuits de longueur **4**
1 1 0 1 1
1 1 1 1 1

[mb5 =: mb1 PMB mb4

1 1 1 1 1
1 1 1 1 1
1 1 1 1 1 NB. **mb5** : existence des chemins et circuits de longueur **5**
1 1 1 1 0
1 1 1 1 1

NB. matrice d'incidence réflexive

[mbr1 =: 1 EDIAG mb1

1 1 0 1 0
0 1 1 1 0
0 1 1 0 1 NB. **mbr1** comme **mb1** mais avec des **1** non significatifs
1 0 0 1 0 NB. sur la diagonale
1 0 0 1 1

[mbr2 =: mbr1 PMB mbr1

1 1 1 1 0
1 1 1 1 1 NB. **mbr2** : un **1** (non diagonal) signifie l'existence d'un
1 1 1 1 1 NB. chemin au moins de longueur au plus **2**.
1 1 0 1 0 NB. un **0** signifie l'inexistence d' un tel chemin
1 1 0 1 1

[mbr3 =: mbr2 PMB mbr1

1 1 1 1 1
1 1 1 1 1
1 1 1 1 1 NB. **mbr3** : existence des chemins de longueur au plus **3**
1 1 1 1 0 NB. ici pas de chemin de longueur au plus **3** de **D** à **E**
1 1 1 1 1

[mbr4 =: mbr3 PMB mbr1

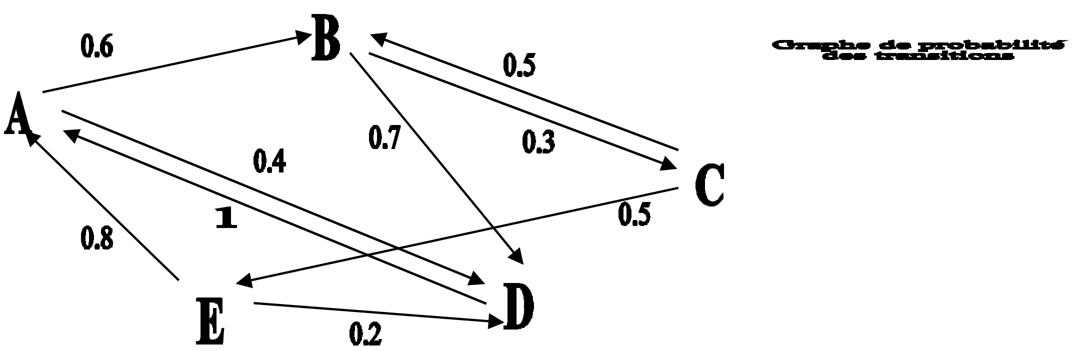
1 1 1 1 1 NB. **mbr4** (4=nb. de sommets – 1) : matrice de la fermeture
1 1 1 1 1 NB. **réflexo-transitive** du graphe étudié. C'est le graphe
1 1 1 1 1 NB. d'une **clique** (que des **1**). Signifie que notre graphe est
1 1 1 1 1 NB. "**fortement connexe**": possibilité de déplacement de tout
1 1 1 1 1 NB. sommet vers tout autre par un chemin au moins.

II) MATRICE STOCHASTIQUE - PRODUIT MATRICIEL CLASSIQUE

Imaginons un voyageur de commerce oeuvrant dans ces 5 villages. Soit **mp1** une matrice de 5 lignes et 5 colonnes contenant les **probabilités conditionnelles** de déplacement de ce voyageur vers l'un de ces villages. Sachant où il se trouve présentement, il peut, chaque jour, soit rester dans le même village que la veille, soit changer de village. Les éléments de cette matrice sont positifs ou nuls et inférieurs ou égaux à 1; la somme des éléments d'une ligne est égale à 1.

Voici, par exemple, une telle matrice de probabilités (stochastique) **mp1** :

	mp1				
					NB. La ligne **B** s'interprète ainsi :
0	0.6	0	0.4	0	NB. quand il est en **B**, la probabilité qu'il
0	0	0.3	0.7	0	NB. soit le lendemain en **A** est 0.4;
0	0.5	0	0	0.5	NB. en **B** : 0 ; en **C** : 0 ; en **D** : 0.4 ;
1	0	0	0	0	NB. en **E** : 0.2 .
0.8	0	0	0.2	0	NB. sommes suivant les lignes égales à 1



Voici comment lire une telle matrice de probabilités (ou stochastique):

mp1	A	B	C	D	E
A	0	0.6	0	0.4	0
B	0	0	0.3	0.7	0
C	0	0.5	0	0	0.5
D	1	0	0	0	0
E	0.8	0	0	0.2	0

Remarque : **A B C D E** pourraient être les 5 états possibles d'un système, les éléments du tableau étant les probabilités de changements d'état à chaque top d'horloge.

Le produit matriciel classique **PMC** permet de calculer **mp2**, **mp3**, **mp4**… etc.. Ces matrices sont également stochastiques.

mpk permet de calculer, au jour **k**, les probabilités de présence du voyageur dans les différents villages.

[mp2 =: mp1 PMC mp1

0.4	0	0.18	0.42	0
0.7	0.15	0	0	0.15
0.4	0	0.15	0.45	0
0	0.6	0	0.4	0
0.2	0.48	0	0.32	0

[mp3 =: mp2 PMC mp1

0.42	0.33	0	0.16	0.09
0.12	0.42	0.045	0.415	0
0.45	0.315	0	0.16	0.075
0.4	0	0.18	0.42	0
0.32	0.12	0.144	0.416	0

[mp4 =: mp3 PMC mp1

0.232	0.252	0.099	0.417	0
0.415	0.0945	0.126	0.342	0.0225
0.22	0.27	0.0945	0.4155	0
0.42	0.33	0	0.16	0.09
0.416	0.264	0.036	0.212	0.072

[mp5 =: mp4 PMC mp1

0.417	0.1887	0.0756	0.2692	0.0495
0.36	0.312	0.02835	0.23665	0.063
0.4155	0.17925	0.081	0.277	0.04725
0.232	0.252	0.099	0.417	0
0.2696	0.2676	0.0792	0.3656	0.018

mp10 =: mp5 PMC mp5
mp20 =: mp10 PMC mp10
mp30 =: mp10 PMC mp20

[mpI =: mp1 PMC^:_ mp1 NB. **mp1** à la puissance **I**nfinie

0.340272	0.240192	0.0720576	0.311449	0.0360288	
0.340272	0.240192	0.0720576	0.311449	0.0360288	NB. valeurs
0.340272	0.240192	0.0720576	0.311449	0.0360288	NB. limites
0.340272	0.240192	0.0720576	0.311449	0.0360288	
0.340272	0.240192	0.0720576	0.311449	0.0360288	

Supposons que au jour **0** notre voyageur soit en **B**.

Son vecteur "probabilité de présence" est alors :
vp0 =: 0 1 0 0 0 NB. probabilité **1** en **B** , **0** ailleurs

[vp1 =: vp0 PMC mp1
0 0 0.3 0.7 0 NB. probabilités de présence jour 1

[vp2 =: vp0 PMC mp2
0.7 0.15 0 0 0.15 NB. probabilités de présence jour 2

[vp3 =: vp0 PMC mp3
0.12 0.42 0.045 0.415 0 NB. probabilités de présence jour 3

[vp4 =: vp0 PMC mp4
0.415 0.0945 0.126 0.342 0.0225 NB. probabilités de présence jour 4

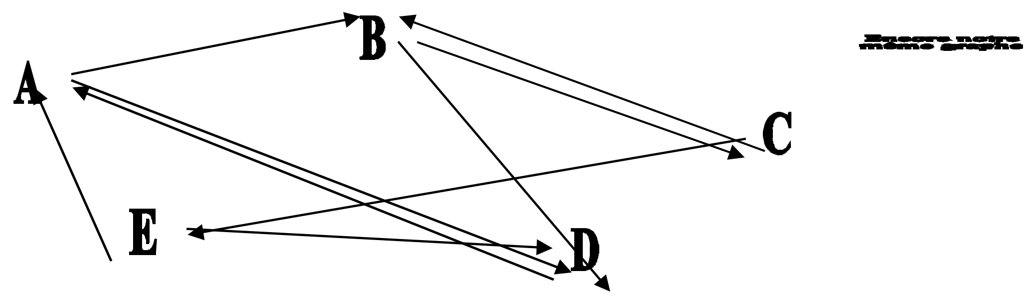
[vp5 =: vp0 PMC mp5
0.36 0.312 0.02835 0.23665 0.063 NB. probabilités de présence jour 5

[vpI =: vp0 PMC mpI
0.340272 0.240192 0.0720576 0.311449 0.0360288
NB. ce sont ici les probabilités-limites de présence de notre voyageur dans les 5 villages après un temps suffisamment long.

III) MATRICE AUX ARCS – PRODUIT MATRICIEL LATIN

[ma1 =: MARCS mb1 NB. matrice aux arcs ou des chemins
NB. de longueur 1

	AB		AD	
		BC	BD	
	CB			CE
DA				
EA			ED	



1) RECHERCHE DE TOUS LES CHEMINS

Utilisation de **PML : P**roduit **M**atriciel **L**atin

[ma2 =: ma1 PML ma1 NB. matrice des chemins de **2** arcs
NB. ou de longueur **2**

ADA		ABC	ABD	
BDA	BCB			BCE
CEA		CBC	CBD CED	
	DAB		DAD	
EDA	EAB		EAD	

[ma3 =: ma2 PML ma1 NB. matrice des chemins de longueur **3**

ABDA	ABCB ADAB		ADAD	ABCE
BCEA	BDAB	BCBC	BCBD BCED BDAD	
CBDA CEDA	CBCB CEAB		CEAD	CBCE
DADA		DABC	DABD	
EADA	EDAB	EABC	EABD EDAD	

[ma4 =: ma3 PML ma1 NB. matrice des chemins de longueur **4**

ABCEA ADADA	ABDAB	ABCBC ADABC	ABCBD ABCED ABDAD ADABD	
BCBDA BCEDA BDADA	BCBCB BCEAB	BDABC	BCEAD BDABD	BCBCE
CBCEA CEADA	CBDAB CEDAB	CBCBC CEABC	CBCBD CBCED CBDAD CEABD CEDAD	
DABDA	DABCB DADAB		DADAD	DABCE
EABDA EDADA	EABCB EADAB	EDABC	EADAD EDABD	EABCE

ma5 =: ma4 PML ma1 NB. etc….

2) RECHERCHE DES CHEMINS ET CIRCUITS SIMPLES

Utilisation de **PMLS : P**roduit **M**atriciel **L**atin **S**imple

NB. simple = pas 2 fois le même arc

[ma1 =: MARCS mb1

	AB		AD	
		BC	BD	
	CB			CE
DA				
EA			ED	

[mas2 =: ma1 PMLS ma1 NB. chemins et circuits simples de
NB. longueur **2**

ADA		ABC	ABD	
BDA	BCB			BCE
CEA		CBC	CBD CED	
	DAB		DAD	
EDA	EAB		EAD	

[mas3 =: mas2 PMLS ma1 NB. chemins et circuits simples de
NB. longueur 3

ABDA	ABCB ADAB			ABCE
BCEA	BDAB		BCBD BCED BDAD	
CBDA CEDA	CEAB		CEAD	CBCE
		DABC	DABD	
EADA	EDAB	EABC	EABD EDAD	

[mas4 =: mas3 PMLS ma1 NB. chemins et circuits simples de
NB. longueur 4

ABCEA		ADABC	ABCBD ABCED ABDAD ADABD	
BCBDA BCEDA	BCEAB	BDABC	BCEAD	
CBCEA CEADA	CBDAB CEDAB	CEABC	CBCED CBDAD CEABD CEDAD	
	DABCB			DABCE
EABDA	EABCB EADAB	EDABC	EDABD	EABCE

mas5 =: mas4 PMLS ma1
mas6 =: mas5 PMLS ma1 NB. etc.

LDIAG mas5 NB. Circuits simples de longueur **5**

ABCBDA ABCEDA	BCBDAB BCEDAB BDABCB	CBDABC CEDABC	DABCBD DABCED	EDABCE

Remarque : il n'y a, à une permutation circulaire près, que **2** circuits simples distincts de longueur **5**

3) RECHERCHE DES CHEMINS ET CIRCUITS ÉLÉMENTAIRES

Utilisation de **PMLE : P**roduit **M**atriciel **L**atin **E**lémentaire

NB. élémentaire = pas 2 fois le même sommet

[ma1 =: MARCS mb1

	AB		AD	
		BC	BD	
	CB			CE
DA				
EA			ED	

NB. chemins et circuits élémen-
NB. taires de longueur **2**

[mae2 =: ma1 PMLE ma1

ADA		ABC	ABD	
BDA	BCB			BCE
CEA		CBC	CBD CED	
	DAB		DAD	
EDA	EAB		EAD	

NB. chemins et circuits élémen-
NB. taires de longueur 3

[mae3 =: mae2 PMLE ma1

ABDA				ABCE
ABCE	BDAB		BCED	
CBDA CEDA	CEAB		CEAD	
		DABC	DABD	
	EDAB	EABC	EABD	

NB. chemins et circuits élémen-
NB. taires de longueur 4

[mae4 =: mae3 PMLE ma1

ABCEA		ABCED		
BCEDA	BCEAB		BCEAD	
	CEDAB	CEABC	CEABD	
				DABCE
		EDABC		EABCE

NB. chemins et circuits élémen-
NB. taires de longueur 5

[mae5 =: mae4 PMLE ma1

ABCEDA				
	BCEDAB			
		CEDABC		
			DABCED	
				EDABCE

Remarque : Les chemins de longueur **5** sont tous des circuits dans un graphe de **5** sommets. A une permutation circulaire près, il n'y a qu'un seul circuit de longueur **5 :** c'est un **circuit hamiltonnien** (pb. dit du voyageur de commerce).

IV) MATRICE DES PSEUDO-DISTANCES – PRODUIT MATRICIEL AUX VALEURS MINIMALES (pseudo-distances minimales)

mb1

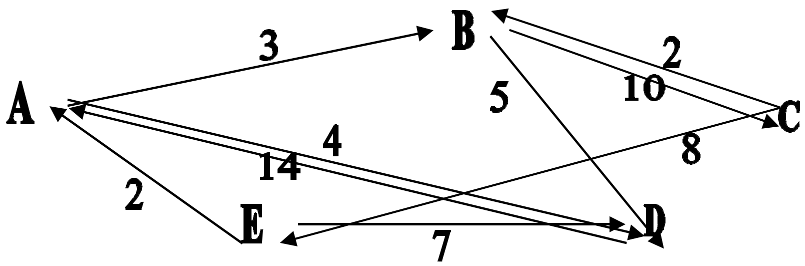
0 1 0 1 0
0 0 1 1 0 NB. Le jeu consiste à minimiser les coûts
0 1 0 0 1 NB. On utilise _ (infini)
1 0 0 0 0 NB. c'est l'élément neutre de **<.** (mini)
1 0 0 1 0

NB. Création d'une matrice de pseudo-distances

[md1 =: 15 MPSEUD mb1 NB. à partir d'une matrice booléenne

0 3 _ 4 _ NB. **md1** : pseudo-distances en **1** arc.
_ 0 10 5 _ NB. (en km par exemple)
_ 2 0 _ 8 NB. ce pourrait être des durées, des coûts,
14 _ _ 0 _ NB. des consommations de carburant ….
2 _ _ 7 0 NB. _ (infini) signifie pas d'arc

NB. les **0** diagonaux s'interprètent très facilement.



Interprétation de cette matrice :

md1	**A**	**B**	**C**	**D**	**E**
A	0	3	_	4	_
B	_	0	10	5	_
C	_	2	0	_	8
D	14	_	_	0	_
E	2	_	_	7	0

[md2 =: md1 PMVm md1

0 3 13 4 _ NB. **PMVm** : **P**roduit **M**atriciel aux
19 0 10 5 18 NB. **V**aleurs **m**inimales
10 2 0 7 8
14 17 _ 0 _ NB. **md2** : matrice des pseudo-distances en
2 5 _ 6 0 NB. **2** arcs au plus

[md3 =: md2 PMVm md1

0 3 13 4 21	
19 0 10 5 18	
10 2 0 7 8	NB. **md3** : pseudo-distances en **3** arcs au plus
14 17 27 0 _	NB. _ : pas de chemin en **3** arcs au plus de **D** à **E**
2 5 15 6 0	

[md4 =: md3 PMVm md1

0 3 13 4 21	
19 0 10 5 18	
10 2 0 7 8	NB. **md4** : pseudo-distances en **4** arcs au plus
14 17 27 0 35	
2 5 15 6 0	

[md5 =: md4 PMVm md1

0 3 13 4 21	
19 0 10 5 18	NB. **md4** : (4=nb. de sommets – 1) est la **matrice** des
10 2 0 7 8	NB. des **pseudo-distances minimales**.
14 17 27 0 35	NB. on a : **md4 = md5 = md6 = md7** ….
2 5 15 6 0	

md1 VALEUR CHEMIN 'BCAD'

_ NB. l'arc **CA** n'existe pas dans le graphe

md1 VALEUR CHEMIN 'BCEDA'

39 NB. somme des valeurs des arcs **BC CE ED DA**

md1 VALEUR CHEMIN 'ABCEDA'

42 NB. valeur du circuit hamiltonnien

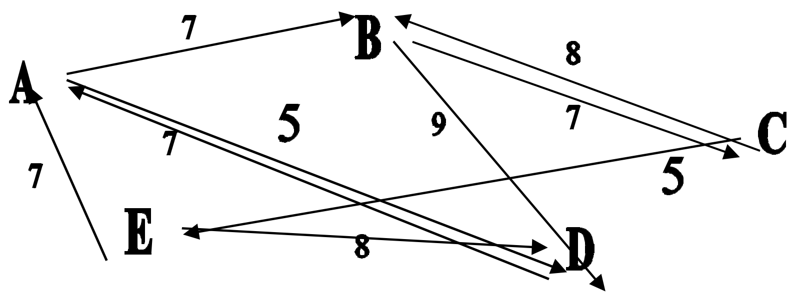
V) MATRICE DE GAINS – PRODUIT MATRICIEL AUX VALEURS MAXIMALES

mb1 NB. matrice d'incidence du graphe

0 1 0 1 0
0 0 1 1 0
0 1 0 0 1
1 0 0 0 0
1 0 0 1 0

[mg1 =: MGAINS mb1

__ 7 __ 5 __ NB. **mg1** : matrice de gains de transitions
__ __ 7 9 __ NB. __ (infini négatif) signifie pas de transition
__ 8 __ __ 5 NB. c'est l'élément neutre pour >. (maxi)
7 __ __ __ __ NB. Le jeu consiste à **maximiser** les gains
7 __ __ 8 __



Graphe des Gains

Interprétation de cette matrice de gains **mg1**

mg1	**A**	**B**	**C**	**D**	**E**
A	__	7	__	5	__
B	__	__	7	9	__
C	__	8	__	__	5
D	7	__	__	__	__
E	7	__	__	8	__

[mg2 =: mg1 PMVM mg1

12 __ 14 16 __
16 15 __ __ 12 NB. **mg2** : gains maxi sur des chemins de longueur **2** au plus
12 __ 15 17 __
__ 14 __ 12 __
15 14 __ 12 __

[mg3 =: mg2 PMVM mg1

23 22 __ 17 19
19 23 22 24 __ NB. **mg3** : gains maxi sur des chemins de longueur **3** au plus
24 23 __ 17 20
19 __ 21 23 __
19 22 21 23 __

[mg4 =: mg3 PMVM mg1

26 30 29 31 __
31 30 30 32 27 NB. **mg4** : gains maxi sur des chemins de longueur **4** au plus
27 31 30 32 __
30 29 __ 24 26
30 29 29 31 26

[mg5 =: mg4 PMVM mg1

38 37 37 39 34
39 38 37 39 35 NB. **mg5** : gains maxi sur des chemins de longueur **5** au plus
39 38 38 40 35
33 37 36 38 __
38 37 36 38 34

[mg6 =: mg5 PMVM mg1

46 45 44 46 42
46 46 45 47 42
47 46 45 47 43 NB. **mg6** : gains maxi sur des chemins de longueur **6** au plus
45 44 44 46 41
45 45 44 46 41

mg1 VALEUR CHEMIN 'BCAD'

__ NB. arc **CA** inexistant

mg1 VALEUR CHEMIN 'BCEDA'

27 NB. somme des gains des arcs **BC CE ED DA**

mg1 VALEUR CHEMIN 'ABCEDA'

34 NB. gain pour le circuit hamiltonnien

V) VERBES UTILISÉS DANS CET ARTICLE

PMC	**P**roduit **M**atriciel **C**lassique (ou **C**ommun)
PMB	**P**roduit **M**atriciel **B**ooléen
PMVm	**P**roduit **M**atriciel aux **V**aleurs **m**inimales
PMVM	**P**roduit **M**atriciel aux **V**aleurs **M**aximales
PML	**P**roduit **M**atriciel **L**atin
PMLE	**P**roduit **M**atriciel **L**atin **E**lémentaire
PMLS	**P**roduit **M**atriciel **L**atin **S**imple
MBOOL	**M**atrice **BOOL**éenne
MSTO	**M**atrice **STO**chastique
MPROB	**M**atrice de **PROB**abilités
MARCS	**M**atrice aux **ARCS**
MPSEUD	**M**atrice de **PSEU**do-**D**istances
MGAINS	**M**atrice de **GAINS**
MPERTES	**M**atrice de **PERTES**
CHEMIN	Indices des sommets d'un **CHEMIN**
VALEUR	**VALEUR** d'un chemin
EDIAG	**E**criture des éléments **DIAG**onaux d'une matrice
LDIAG	**L**ecture des éléments **DIAG**onaux d'une matrice
CHANGE	**CHANGE** une composante en une autre

VI) PROGRAMMATION - FICHIER SCRIPT GRAPHES.IJS

```
 NB. AL : alphabet par défaut de 52 caractères :
AL =:'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
 NB. ------------- Créations de matrices ----------------------------------------------
MBOOL    =: ([:?]$2:) : ([>:0.001"_*[:?]$1000"_)   NB. matrice booléenne
 NB.   mb =: MBOOL 5 5                    avec  50 pour cent de 1
 NB.   mb =: 0.23  MBOOL 5 5              avec  23 pour cent de 1
MCLIQUE =: ,:&.>@([:,&.>/~]$AL"_ :[)      NB. que des 1
 NB.     ma =: MCLIQUE 5    ou  ma =: 'PQRST'   MCLIQUE  5
MARCS    =: $@]$0:{"1,.@,@]|."1 a:"_ ,.[:,AL"_ :[MCLIQUE#@]
 NB.     ma =: MARCS mb      ou   ma =: 'pqrstuvwxyz' MARCS mb
 NB. crée une matrice latine à partir d'une matrice booléenne
MPROB    =: >@(([:(%+/)]*1:+[:?([:#])#100"_ :[)&.>@([:<"1]))
 NB.      mp =: MPROB mb         ou        mp =: 10 MPROB mb
MSTO     =: (%"2+/"1)@:(?@((],])$(9: : [)))  NB. Matrice stochastique
 NB.     ms =:  MSTO 5 5          ou  ms =: 50 MSTO 5 5
MGAINS   =:  0 __"_ CHANGE]*1:+[:?([:$])$9: :[
 NB.      mg =:  MGAINS mb        ou     mg =:  50 MGAINS mb
MPERTES =:  0   _"_ CHANGE]*1:+[:?([:$])$9: :[
 NB.     m =:   MPERTES mb        ou   m =:   25  MPERTES mb
MPSEUD   =:  0:  EDIAG  MPERTES
 NB. crée aléatoirement une matrice de pseudo-distances compatible
 NB. avec la matrice booléenne mb  (avec des 0 sur la diagonale)
 NB. md =:        MPSEUD   mb                    valeur maximale 9 (par défaut)
 NB. md =:  25  MPSEUD   mb
 NB. ------------- Lecture et Modifications -------------------------------------------
LDIAG      =: (<0 1)&|:
 NB. lecture des éléments diagonaux d'une matrice
 NB. d =: LDIAG m
EDIAG      =: [`((<"1@(,.&i.~)@#)@])`]}
 NB. écriture des éléments diagonaux d'une matrice
 NB. ma =: a: EDIAG ma  ou    mb =: 0 1 1 0 0 EDIAG mb
CHANGE   =: 4 : '($y.)$({:x.)((({.x.)=v)#i.#v)}v=.,y.'
 NB. m =:  5  9   CHANGE m                     change les 5 en 9 dans m
NORM       =: $@]$a:"_(0&=@#@>@]#i.@#@,@])},@]
 NB.   ma =: NORM ma              remplace par a: les matrices à 0 ligne
 NB. ------------- Verbes divers utilisés pour les produits matriciels ---------------
PL1            =: ([,}.@:])"1       NB. produit latin de 2 chaînes de caractères
SB             =: ' '&~:#]          NB. suppression des blancs
PL2            =: SB @ PL1          NB. composition de SB et PL1
PL3            =: PL2/&.>           NB. prod. lat. de 2 matrices de chaînes
PL4            =: <@(,/)@>@ PL3        NB. matrices empaquetées
```

```
ORDRALPHA =: (/:~)&.>                    NB. lignes rangées par ordre alpha
SUPDUP       =: ~.&.>                     NB. suppres chaines dupliques
SUPLB         =: (+./"1@(' '&~:)#])&.>   NB. Suppres lignes blanches
PL5            =: ORDRALPHA @ SUPLB @ SUPDUP @ PL4
TESTPL       =: (a:"_ -:])+.a:"_ -:[
 NB. x TESTPL y    répond 1 (si y=a: ou x=a:),  0 (sinon)
PL               =: (PL5`(a:"_)@. TESTPL)"0      NB. produit latin
 NB. ------------- Chemins simples -----------------------------------------------------
SIMP          =: (*./@~:@(2&(,/\))"1#])&.>       NB. pas 2 fois le même arc
SIMPLE       =: NORM @ SIMP
 NB. ------------- Chemins élémentaires ------------------------------------------------
ELEM          =: (*./@(' '&=+.~:)"1#])&.>  NB. pas 2 fois le même sommet
TESTVIDE  =: a:&-:@]"0
CS              =: (],{.)"1&.>`(a:"_)@.TESTVIDE   NB. pour diagonaux
ELEMD       =: CS@ELEM@(}:"1&.>)
ELEMDIAG =: NORM @ ELEMD
ELEMENTAIRE=:(ELEMDIAG@LDIAG@])EDIAG(NORM@ELEM@])
 NB. ------------- Somme latine ----------------------------------------------------------
SL1            =: /:~@~.@,&.>
TESTSL       =: [:<./2:,((a:"_ -:])#0:),(a:"_ -:[)#1:
 NB. x TESTSL y    repond 0 (y=a:), 1 (x=a:) , 2 (sinon)
SL               =: ([`]`SL1 @. TESTSL)"0           NB. somme latine
 NB. ------------- Valeur d'un chemin --------------------------------------------------
CHEMIN     =: 2:<@;/\AL"_ :[i.]
 NB. CHEMIN 'BDFG'      ou    'PQRSTUVWXYZ' CHEMIN 'URTX'
 NB. fournit les indices des extrémités des arcs dans l'alphabet
VALEUR      =: [:+/]{[
 NB.               md   VALEUR   CHEMIN   'BDFG'
 NB. ou          md   VALEUR    'PQRSTUVWXYZ'   CHEMIN    'URTX'
 NB. ------------- Produits matriciels ---------------------------------------------------
PMC           =: +/ .*               NB. Produit Matriciel Classique
PMB           =: +./ .*.              NB. Produit Matriciel Booléen
PMVm        =: <./ .+               NB. Produit Matriciel aux Valeurs minimales
PMVM        =: >./ .+               NB. Produit Matriciel aux Valeurs Maximales
PML           =: SL/ .PL           NB. Produit Matriciel Latin
PMLE         =: ELEMENTAIRE @ PML         NB. Idem élémentaire
PMLS         =: SIMPLE @ PML                    NB. idem simple
```

VI) CONCLUSION

Les méthodes matricielles de la théorie des graphes ne sont pas "légères". Elles ont le bon goût de répondre aux questions "*globales*" du type *: "Quels sont* ***tous*** *les chemins élémentaires de longueur* 3 ?" ou "*Quelles sont* ***toutes*** *les pseudo-distances minimales en 4 arcs au plus* ?", par exemple. Si on ne s'intéresse qu'à des questions sur les chemins existant entre 2 sommets précis, d'autres méthodes plus connues et plus "*légères*" sont préférables.

Les généralisations du "produit matriciel classique" basé sur 2 lois de composition interne (le produit et la somme) sont riches de possibilités.

Les verbes proposés ici sont utilisables pour des graphes orientés quelconques. Les exemples proposés sur un graphe de 5 sommets (facile à imprimer et à lire) sont transposables sur un graphe beaucoup plus important (temps de réponse trop souvent proportionnel au cube du nombre de sommets… d'où … manque de légèreté !).

Toutefois, quelques remarques permettent de limiter les calculs. Par exemple, pour calculer la matrice (md42) des pseudo-distances minimales sur un graphe de 43 sommets, on calculera :
md2 =: md1 **PMVm** md1 , md4 =: md2 **PMVm** md2 , md8 , md16 , md32 et md64 (car md42 = md43 = md44 = … = md64 = …); ceci nécessite d'effectuer seulement 6 calculs avec le "*produit matriciel aux valeurs minimales*" **PMVm**.

Les propriétés matricielles des graphes ne sont pas épuisées ici. D'autres matrices peuvent être utilisées pour les "*graphes bi-partis*", les "*multigraphes*" …etc. D'autres questions concernant les "*composantes connexes*", les sous-ensembles "*intérieurement stables*" ou "*extérieurement stables*", les "*noyaux*", les "*colorations*" de sommets ou d'arcs … peuvent être abordées au moyen de l'outil matriciel, ainsi que des problèmes de "*flots*" ou "d'*affectation*"…. Ce qui laisse matière pour de futurs articles … si celui-ci n'augmente pas trop la proportion de lecteurs allergiques à la théorie des graphes !

Le Produit Matriciel Latin repose sur le produit latin de 2 chaînes (**PL1**). Il effectue la concaténation de 2 chemins pour en faire un plus long ; ex :

'ABDFG' PL1 'GEFABDH'

ABDFGEFABDH

Le reste n'est qu'habillage : suppression des chemins dupliques, sélection des chemins simples ou élémentaires, rangement dans l'ordre alphabétique …. **PL** est le produit latin pour des matrices dont chaque ligne contient un chemin sous forme d'une chaîne de caractères.

La somme latine **SL** empile de tels chemins dans une matrice.

Enfin **PML =: SL/ .PL** est le produit matriciel latin. Il est déduit du produit matriciel classique **PMC =: +/ .*** en remplaçant le classique produit * par **PL** et la somme + par **SL**.

Cette application, écrite en 2 pages de **J** *sous* **forme tacite** *(commentaires ... trop brefs ...compris), met à nouveau en lumière la puissance, la souplesse, la concision de ce langage.*