Quadrature n° 72 (2009) 22–28

DOI: 10.1051/quadrature/2009009



# Algèbres tropicales et plus court chemin

*par* Antoine Delignat-Lavaud

**Résumé.**
Chercher le plus court chemin sur un graphe est un problème algorithmique complexe. Ce texte de TIPE propose de revisiter ce problème à la lueur de la géométrie tropicale et d'obtenir la résolution *via* un système linéaire sur une certaine algèbre. La fin du texte est consacrée à une implémentation de cette solution théorique.

Le problème du plus court chemin, consistant, étant donné un ensemble de sommets reliés par des arêtes pondérées par leur « coût de franchissement », à trouver un chemin le moins coûteux possible entre deux d'entre eux, est fondamental en théorie des graphes. Plusieurs algorithmes classiques permettent d'y répondre efficacement, mais ils se contentent en général de traiter le cas où le « coût de franchissement » des arêtes est un réel ou un réel positif [2].

Or, dans de nombreuses situations concrètes, on voudrait déterminer un plus court chemin alors que ces « coûts » dépendent du temps ou bien des déplacements précédents (par exemple, dans un réseau de transports en commun où les déplacements sont conditionnés par des horaires).

Le but de cet article est de montrer qu'on peut généraliser les algorithmes classiques de plus court chemin en exprimant ce problème comme un système linéaire par changement d'algèbre sous-jacente : on introduit pour cela les structures algébriques dites tropicales, obtenues par substitution des opérations usuelles (la somme et le produit) par des lois aux propriétés plus faibles (en particulier min et max, pour lesquelles les réels ne disposent pas de symétriques), mais permettant de définir les opérateurs linéaires.

## I Présentation des structures tropicales

L'étude formelle des structures utilisant min ou max comme loi est relativement récente, et fut motivée par ses applications en théorie des graphes, des langages et en automatique discrète [1, 6]. Le qualificatif « tropical » est donné en l'honneur du mathématicien Imre Simon, d'origine brésilienne [4].

Les propriétés caractéristiques des semi-anneaux et dioïdes varient légèrement selon les auteurs. Les conventions utilisées ici sont celles de [5].

**Définition 1 (semi-anneau).** On appelle semi-anneau tout ensemble $\mathcal{E}$ muni d'une loi additive $\oplus$ associative et commutative de neutre $\mathbb{0}$ et d'une loi multiplicative $\otimes$ associative et unifère de neutre $\mathbb{1}$ qui vérifient :

(i) $\forall (a,b,c) \in \mathcal{E}^3, (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ et $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$ ;

(ii) $\forall a \in \mathcal{E}, a \otimes \mathbb{0} = \mathbb{0} \otimes a = \mathbb{0}$ ($\mathbb{0}$ est absorbant pour $\oplus$).

Dans la suite, on supposera $\oplus$ sélective : $\forall x, y \in \mathcal{E}, x \oplus y \in \{x, y\}$.

**Définition 2 (dioïde).** Un semi-anneau $(\mathcal{E}, \oplus, \otimes)$ est un dioïde si la loi $\oplus$ est idempotente, c'est-à-dire si :

$$\forall x \in \mathcal{E},\ x \oplus x = x.$$

Une propriété fondamentale des dioïdes est qu'ils sont totalement ordonnés, c'est-à-dire qu'on peut toujours comparer deux éléments ; dans notre problème, c'est ce qui permet d'affirmer qu'un chemin est « plus court » qu'un autre.

**Proposition 3 (ordre canonique).** *Tout dioïde $(\mathcal{D}, \oplus, \otimes)$ est naturellement muni d'une relation d'ordre total définie par :*

$$\forall x, y \in \mathcal{D},\quad x \preceq y \iff x \oplus y = x.$$

*Démonstration.* Soient $x, y, z \in \mathcal{D}$. Par idempotence, on a toujours $x \preceq x$ : $\preceq$ est réflexive. Si $x \preceq y$ et $y \preceq x$,

$x = x \oplus y = y \oplus x = y$ : $\preceq$ est antisymétrique. Enfin, si $x \preceq y$ et $y \preceq z$,

$$x \oplus z = (x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus y = x.$$

On a donc $x \preceq z$ : $\preceq$ est transitive. □

**Example 1.** On note $\mathbb{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \min, +)$. Alors $\mathbb{R}_{\min}$ est un dioïde, que l'on appelle par convention l'algèbre (min, +). L'ordre sur ce dioïde est l'ordre habituel sur les réels : $\min(x, y) = x \Leftrightarrow x \preceq y$.

On peut construire de la même façon $\mathbb{R}_{\max}$, $\mathbb{Z}_{\max}$, $\mathbb{N}_{\min}$... Dans le cas de l'algèbre (min, +), on utilisera ainsi les notations suivantes :

| | | |
|---|---|---|
| min | $\rightarrow$ | $\oplus$ |
| + | $\rightarrow$ | $\otimes$ |
| $\infty$ | $\rightarrow$ | $\varepsilon$ ou $\mathbb{0}$ |
| 0 | $\rightarrow$ | $e$ ou $\mathbb{1}$ |

**Example 2 (dioïde des booléens).** L'ensemble $\mathbb{B} = \{\mathbb{0}, \mathbb{1}\}$ muni des lois $\vee$, $\wedge$ définies par :

| $\vee$ | $\mathbb{0}$ | $\mathbb{1}$ |
|---|---|---|
| $\mathbb{0}$ | $\mathbb{0}$ | $\mathbb{1}$ |
| $\mathbb{1}$ | $\mathbb{1}$ | $\mathbb{1}$ |

| $\wedge$ | $\mathbb{0}$ | $\mathbb{1}$ |
|---|---|---|
| $\mathbb{0}$ | $\mathbb{0}$ | $\mathbb{0}$ |
| $\mathbb{1}$ | $\mathbb{0}$ | $\mathbb{1}$ |

forme un dioïde commutatif appellé dioïde des booléens ou algèbre de Boole.

Il est tout à fait possible d'envisager le calcul matriciel dans ce type de structures. On définit ainsi les matrices et les opérateurs matriciels sur un dioïde $(\mathcal{D}, \oplus, \otimes)$ comme sur un anneau.

**Définition 4 (calcul matriciel).** Soient $A, B \in \mathfrak{M}_n(\mathcal{D})$, avec $A = \left(a_{i,j}\right)_{i,j}$, $B = \left(b_{i,j}\right)_{i,j}$, on pose :

$$A \oplus B = \left(a_{i,j} \oplus b_{i,j}\right)_{i,j} \text{ et } A \otimes B = \left(\bigoplus_{k=1}^{n} a_{i,k} \otimes b_{k,j}\right)_{i,j}.$$

Outre les matrices, on définit aussi les endomorphismes de dioïde [5], qui forment à leur tour un dioïde.

**Définition 5 (dioïde des endomorphismes).** On suppose que $(\mathcal{D}, \oplus, \otimes)$ est un dioïde. On définit alors $\mathcal{F}$ comme ensemble des applications $f : \mathcal{D} \longrightarrow \mathcal{D}$ qui vérifient les propriétés suivantes :

(i) $\forall a, b \in \mathcal{D},\ f(a \oplus b) = f(a) \oplus f(b)$

(ii) $f(\mathbb{0}) = \mathbb{0}$.

On munit $\mathcal{F}$ de la loi $\hat{\oplus}$, définie par :

$$\forall f, g \in \mathcal{F},\ \forall x \in \mathcal{D},\ (f \hat{\oplus} g)(x) = f(x) \oplus g(x)$$

dont le neutre est l'application $\hat{\mathbb{0}} : x \mapsto \mathbb{0}$. Enfin, on utilise la composition $\circ$ de neutre Id comme produit (associatif mais non commutatif).

Alors, $(\mathcal{F}, \hat{\oplus}, \circ)$ est un dioïde.

*Démonstration.* $(\mathcal{F}, \hat{\oplus})$ est bien un monoïde commutatif. $(\mathcal{F}, \circ)$ est un monoïde et $\forall f \in \mathcal{F}$, $f \circ \hat{\mathbb{0}} = \hat{\mathbb{0}} \circ f = \hat{\mathbb{0}}$. Reste à vérifier la distributivité de $\circ$ sur $\hat{\oplus}$ :

$$\begin{aligned}\forall f, g, h \in \mathcal{F},\ \forall x \in \mathcal{D},\\ (f \circ (g \hat{\oplus} h))(x) &= f(g(x) \oplus h(x))\\ &= ((f \circ g) \hat{\oplus} (f \circ h))(x)\\ \forall f, g, h \in \mathcal{F},\ \forall x \in \mathcal{D},\\ ((g \hat{\oplus} h) \circ f)(x) &= (g \hat{\oplus} h)(f(x))\\ &= ((g \circ f) \hat{\oplus} (h \circ f))(x).\end{aligned}$$

Enfin, on a $\forall f,\ f \hat{\oplus} f = f$. $(\mathcal{F}, \hat{\oplus}, \circ)$ est donc un dioïde. □

## II Problème du plus court chemin

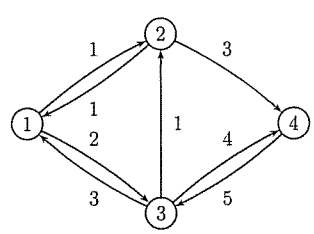
Dans la suite, on se place dans un dioïde $(\mathcal{D}, \oplus, \otimes)$. On donne une définition « orientée valuation » d'un graphe :

**Définition 6 (graphe).** Un graphe $\mathcal{G} = (S, \delta)$ est la donnée d'un ensemble fini de sommets $S = \{1, \ldots, n\}$ et d'une application de valuation $\delta : S \times S \longrightarrow \mathcal{D}$, telle que $\delta(i, j)$ représente la valuation de l'arc reliant $i$ à $j$ ou $\mathbb{0}$ si un tel arc n'existe pas.

Pour simplifier, on prendra toujours $\delta(i, i) = \mathbb{1}$, ce qui intuitivement signifie que rester sur le sommet $i$ ne « coûte » rien. Cette définition met en évidence l'association canonique entre un graphe et sa matrice d'adjacence généralisée :

**Définition 7 (matrice d'adjacence).** On appelle matrice d'adjacence la matrice $M \in \mathfrak{M}_n(\mathcal{D})$ définie par $M = (\delta(i, j))_{i,j}$. Réciproquement, si $A \in \mathfrak{M}_n(\mathcal{D})$, on note $G(A)$ le graphe orienté valué associé à $A$.

**Example 3.** Dans le dioïde $\mathbb{N}_{\min}$ :




$$\begin{pmatrix} 0 & 1 & 2 & \infty \\ 1 & 0 & \infty & 3 \\ 3 & 1 & 0 & 4 \\ \infty & \infty & 5 & 0 \end{pmatrix}.$$

Dans cette matrice, on retrouve l'élément neutre $\mathbb{1} = 0$ pour $\otimes = +$ sur la diagonale et l'élément neutre $\mathbb{0} = \infty$ pour $\oplus = \min$ entre les couples de sommets non reliés par des arcs dans le graphe.

**Définition 8 (chemin, poids).** Un chemin $\gamma$ dans un graphe $G(A)$ est une suite de sommets $(i = i_0, i_1, \ldots, i_{k-1}, i_k = j)$. On note $\Gamma_{i,j}^k$ l'ensemble des chemins reliant $i$ à $j$ en exactement $k$ arcs, $\Gamma_{i,j}^{(k)}$ les chemins joignant $i$ et $j$ en $k$ arcs au plus et $\Gamma_{i,j}$ l'ensemble des chemins quelconques entre $i$ et $j$.

On appelle poids d'un chemin le produit de la valuation des arcs qu'il emprunte :

$$\forall \gamma \in \Gamma_{i,j}^k,\ \omega(\gamma) = \bigotimes_{p=0}^{k-1} \delta(i_p, i_{p+1}).$$

On peut à présent définir le problème du plus court chemin dans sa forme la plus générale.

**Définition 9 (plus court chemin généralisé).** Le problème du plus court chemin consiste à calculer, pour tout sommet de départ $i$ et tout sommet d'arrivée $j$, la somme (si elle existe) des poids des chemins reliant $i$ à $j$, c'est-à-dire :

$$\bigoplus_{\gamma \in \Gamma_{i,j}} \omega(\gamma).$$

Ces définitions correspondent à l'idée intuitive que l'on a d'un plus court chemin dans l'algèbre $(\min, +)$ : c'est le minimum pris sur les chemins de la somme des valuations des arcs empruntés. Comme on a supposé la somme sélective (la valeur d'une somme est égale à un de ses termes), on cherche aussi un chemin $\gamma_0 \in \Gamma_{i,j}$ dont le poids est minimum.

Montrons à présent que la résolution de ce problème est équivalente à un calcul de puissance de matrice dans $\mathcal{D}$.

**Théorème 10 (interprétation du produit matriciel).** *Chaque terme $(i, j)$ de la $k$-ième puissance de la matrice $A$ associée au graphe $\mathcal{G}$ est la somme des poids des chemins reliant $i$ à $j$ en $k$ arcs exactement :*

$$\forall k \in \mathbb{N},\ \forall (i,j) \in [\![1,n]\!]^2,\ (A^k)_{i,j} = \bigoplus_{\gamma \in \Gamma_{i,j}^k} \omega(\gamma).$$

*Démonstration.* Par récurrence sur $k \in \mathbb{N}$. C'est vrai pour $k = 1$ par définition de la matrice d'adjacence. Soit $k \geq 2$, on suppose la formule vérifiée au rang $k-1$. Alors $\forall (i,j) \in [\![1,n]\!]^2$,

$$\begin{aligned}
(A^k)_{i,j} &= \bigoplus_{p=1}^{n} (A^{k-1})_{i,p} \otimes a_{p,j} \\
&= \bigoplus_{p=1}^{n} \left( \bigoplus_{\gamma \in \Gamma_{i,p}^{k-1}} \omega(\gamma) \otimes a_{p,j} \right) \\
&= \bigoplus_{\gamma \in \Gamma_{i,p}^{k-1}} \left( \bigoplus_{p=1}^{n} \omega(\gamma) \otimes a_{p,j} \right) \\
(A^k)_{i,j} &= \bigoplus_{\gamma \in \Gamma_{i,j}^k} \omega(\gamma).
\end{aligned}$$

□

Ainsi, la résolution du problème du plus court chemin revient à calculer la matrice :

$$\left( \bigoplus_{\gamma \in \Gamma_{i,j}} \omega(\gamma) \right)_{i,j} = \left( \bigoplus_{p=0}^{+\infty} \bigoplus_{\gamma \in \Gamma_{i,j}^p} \omega(\gamma) \right)_{i,j} = \bigoplus_{p=0}^{+\infty} A^k.$$

**Définition 11 (semi-inverse).** On appelle semi-inverse d'une matrice $A$ la somme (si elle existe) :

$$A^* = \bigoplus_{p=0}^{+\infty} A^p.$$

Montrons à présent comment le problème du plus court chemin peut s'écrire comme un système linéaire du type « point fixe » dans un dioïde.

**Définition 12 (cycle absorbant).** Un cycle est un chemin $\gamma \in \Gamma_{i,i}$. On dit que le cycle $\gamma$ est absorbant si $\omega(\gamma) < \mathbb{1}$.

Par exemple, dans l'algèbre $(\min, +)$, un cycle absorbant $\gamma$ vérifie $\omega(\gamma) < 0$. En empruntant un nombre arbitraire de fois un tel cycle, on crée un chemin dont le poids tend vers $-\infty$ ! Il est donc impératif d'écarter ce type de situation pour espérer trouver un chemin de poids minimal.

**Théorème 13.** *Soit $\mathcal{G}$ un graphe à $n$ sommets sans circuit absorbant. Alors pour tout $k \geq n$, $A^k = A^{n-1}$. En particulier,*

$$A^* = \bigoplus_{p=0}^{n-1} A^p = (A \oplus I)^{n-1}.$$

*Démonstration.* On note $\Lambda_{i,j}$ l'ensemble des chemins élémentaires de $i$ à $j$, c'est-à-dire passant au plus

une fois par chaque sommet du graphe ou encore ne contenant pas de cycle. Alors

$$\bigoplus_{\gamma\in\Gamma_{i,j}} \omega(\gamma) = \bigoplus_{\gamma\in\Lambda_{i,j}} \omega(\gamma).$$

En effet, si $\gamma_{i,j} \in \Gamma_{i,j}$ admet un cycle, c'est-à-dire se décompose en $\gamma_{i,j} = \alpha_{i,p} \cdot \theta_{p,p} \cdot \beta_{p,j}$ avec $\theta_{p,p} \in \Gamma_{p,p}$ et $\alpha_{i,p} \cdot \beta_{p,j} \in \Lambda_{i,j}$, alors $\theta_{p,p} \geq 0$ donc $\omega(\gamma_{i,j}) \geq \omega(\alpha_{i,p} \cdot \beta_{p,j})$.

Or, un chemin élémentaire passe par au plus $n-1$ arcs : $\forall k \geq n$, $\Lambda_{i,j}^{(k)} = \Lambda_{i,j}^{n-1}$, et donc $\forall k \geq n$, $A^k = A^{n-1}$. Ainsi,

$$A^* = \bigoplus_{p=0}^{+\infty} \bigoplus_{\gamma\in\Gamma_{i,j}^p} \omega(\gamma) = \bigoplus_{p=0}^{n-1} \bigoplus_{\gamma\in\Lambda_{i,j}^p} \omega(\gamma) = \bigoplus_{p=0}^{n-1} A^p.$$

Reste à vérifier que $A^* = (A \oplus I)^{n-1}$. Écrivons la forme développée du produit $(A \oplus I)^{n-1} = (A \oplus I) \otimes \cdots \otimes (A \oplus I)$ :

$$(A \oplus I)^{n-1} = A^{n-1} \oplus \underbrace{A^{n-2} \oplus \cdots \oplus A^{n-2}}_{n-1 \text{ fois}} \oplus \cdots \oplus \underbrace{A \oplus \cdots \oplus A}_{n-1 \text{ fois}} \oplus I.$$

Par idempotence, on a pour tout $k \in \mathbb{N}$, $A^k \oplus A^k = A^k$, d'où

$$(A \oplus I)^{n-1} = \bigoplus_{p=0}^{n-1} A^p = A^*.$$

□

D'après ce qui précède, la ligne $i_0$ de la semi-inverse de $A$ a pour composantes les poids des chemins minimaux d'origine $i_0$. La solution du problème du plus court chemin à origine unique $i_0$ est donc $B \otimes A^*$ où $B$ est le vecteur $(\delta_{i_0,j})_{j\leq n}$. On peut remarquer que c'est une solution du système linéaire :

$$X = X \otimes A \oplus B.$$

En effet, pour le problème du plus court chemin, $A \oplus I = A$ ($A$ a des $\mathbb{1}$ sur la diagonale), et donc

$$\begin{aligned} (B \otimes A^*) \otimes A \oplus B &= B \otimes \left(A^{n-1} \otimes A\right) \oplus B \quad (1) \\ &= B \otimes (A^n \oplus I) \quad (2) \\ &= B \otimes A^*. \quad (3) \end{aligned}$$

Dans (1), on a utilisé l'associativité du produit matriciel, dans (2) la distributivité de $\otimes$ sur $\oplus$ et dans (3) le théorème 13. De la même façon, $A^*$ est solution du système $X = X \otimes A \oplus I$. L'avantage majeur de cette interprétation est son lien étroit avec les méthodes de résolution des systèmes linéaires [5].

# III Algorithmes de résolution

## III.1 Approche naïve

Une manière évidente de calculer $A^*$ est de procéder par produits matriciels successifs. Comme chaque produit matriciel nécéssite $O(n^3)$ opérations $\oplus$ et $\otimes$, cette méthode a une complexité en $O(n^4)$. On peut améliorer cette méthode de deux façons : d'une part en utilisant un algorithme d'exponentiation rapide, et d'autre part en utilisant un algorithme de produit matriciel rapide (tous deux sont expliqués dans [2]).

De même, si l'on ne cherche qu'une ligne $i_0$ de $A^*$ (plus court chemin à origine unique), on peut la construire itérativement en $O(n^3)$ par la récurrence :

$$\begin{cases} X_0 = (\delta_{i_0,i})_{i\leq n} \\ \forall n \in \mathbb{N},\ X_{n+1} = X_n \otimes A \oplus X_0. \end{cases}$$

## III.2 Algorithme de Dantzig

L'idée de Dantzig [3] consiste à construire la pseudo inverse $A^*$ en partant du graphe réduit à un sommet et en ajoutant successivement les sommets du graphe. En effet, une fois les plus courts chemins connus dans un graphe, il est relativement aisé de mettre à jour ces chemins lorsque l'on ajoute un sommet.

Cette approche ressemble à la méthode d'inversion par blocs de Strassen. Dans la suite, on notera $M^{[i]}$ la matrice des $i$ premières lignes et colonnes extraite de $M$. Les graphes sont supposés sans circuits absorbants. On a donc pour $k \in [\![2, n]\!]$ :

$$A^{[k]} = \left[\begin{array}{c|c} A^{[k-1]} & C_{k-1} \\ \hline L_{k-1} & \mathbb{1} \end{array}\right].$$

**Théorème 14.**

$$A^{*[k]} = \left[\begin{array}{c|c} A^{*[k-1]} \oplus \left(A^{*[k-1]} \otimes C_{k-1}\right) \otimes \left(L_{k-1} \otimes A^{*[k-1]}\right) & A^{*[k-1]} \otimes C_{k-1} \\ \hline L_{k-1} \otimes A^{*[k-1]} & \mathbb{1} \end{array}\right].$$

D'après cette décomposition, on peut calculer itérativement $A^* = A^{*[n]}$, sachant que $A^{*[1]} = (\mathbb{1})$. On commence par calculer les deux vecteurs ligne et colonne supplémentaires, puis on réinjecte leur produit dans le bloc supérieur gauche. L'algorithme qui en

découle s'en déduit naturellement :

Algorithme 1. Algorithme de Dantzig.

```
1  programme Dantzig(A : matrice, n : entier)
2   variables i, j, k : entier,
3
4   pour k de 2 à n faire
5   // Calcul de la ligne et de la colonne rajoutées
6    pour i de 1 à k-1 faire
7     A_{i,k} ← ⊕_{j<k} a_{i,j} ⊗ a_{j,k}
8     A_{k,i} ← ⊕_{j<k} a_{k,j} ⊗ a_{j,i}
9    fin pour
10
11   pour i de 1 à k-1 faire
12    pour j de 1 à k-1 faire
13     A_{i,j} ← A_{i,j} ⊕ A_{i,k} ⊗ A_{k,j};
14    fin pour
15   fin pour
16  fin pour
17
18  retourner A;
19 fin programme
```

L'algorithme de Dantzig nécessite $\frac{2n^3}{3} + o(n^3)$ opérations $(\oplus, \otimes)$, ce qui est nettement mieux que le calcul de $A^*$ par produits matriciels.

## III.3 Algorithme de Dijkstra

Plutôt que de calculer complètement la semi-inverse $A^*$, il est souvent suffisant de n'en calculer qu'une ligne (dont l'indice correspond à un sommet de départ $i_0$ fixé). Pour le calcul d'une ligne de $A^*$, on dispose (à condition d'être dans un dioïde *sélectif* où $\mathbb{1}$ est minimal) d'un algorithme glouton dû à Dijkstra particulièrement efficace, bien que sa complexité dépende de la structure de données choisie pour l'implémenter.

Algorithme 2. Algorithme de Dijkstra.

```
1  programme dijkstra(A : matrice, i_0, n : entier)
2  variables π[1..n] : vecteur,
3            i,j : entier, T : ensemble;
4
5  T ← {1,...,n};
6  π[1..n] ← 0; π[i_0] ← 1;
7
8  tant que T ≠ ∅ faire
9   i ← indice_du_min(π);
10  T ← T\{i}
11
12  si T = ∅ retourner π;
13  pour j dans T faire
14   π[j] ← π[j] ⊕ π[i] ⊗ A[i][j];
15  fin pour
16 fin tant que
17
18 retourner π;
19 fin programme
```

La preuve de l'algorithme repose sur l'invariant de boucle suivant : la plus petite composante $\pi[i]$ de $\pi$ (au sens de $\preceq$) vaut $A^*[i_0, i]$. À chaque étape, on calcule ainsi une composante de la $i_0$-ème ligne de $A^*$, et le problème est réduit d'une dimension (un sommet est supprimé de $T$).

**Théorème 15 (invariant de l'algorithme de Dijkstra).**

$$\exists i \in T, \bigoplus_{j \in T} \pi[j] = \pi[i] = A^*_{i_0,i}.$$

*Démonstration.* Lors de l'*initialisation*, on vérifie que la plus petite composante est $\pi[i_0] = \mathbb{1}$, qui est le plus court chemin de $i_0$ à $i_0$ (soit $A^*_{i_0,i_0}$).

Vérifions maintenant la *conservation* : on suppose qu'à l'itération courante, $\exists i \in T, \pi[i] = A^*_{i_0,i}$. On effectue $T \leftarrow T \setminus \{i\}$ et $\forall j \in T, \pi[j] \leftarrow \pi[j] \oplus \pi[i] \otimes A[i][j]$.

$$\begin{aligned}\pi[k] &= \bigoplus_{j \in T\setminus\{i\}} \pi[j] = \bigoplus_{j \in T\setminus\{i\}} (\pi[j] \oplus \pi[i] \otimes A[i][j]) \\ &= \pi[i'] \oplus \bigoplus_{j \in T\setminus\{i\}} \pi[i] \otimes A[i][j]\end{aligned}$$

avec $k \neq i$ (car $i \notin T$) et où $\pi[k]$ représente le poids d'un chemin de $i_0$ à $k$. Montrons que $\pi[k] = A^*_{i_0,k}$. Soit $\gamma$ le chemin de poids minimal reliant $i_0$ et $k$. Supposons d'abord que $\gamma$ passe par $i$. Alors, le chemin de $i_0$ à $i$ étant minimal,

$$\begin{aligned}\omega(\gamma) &= \pi[i] \otimes \left( \bigoplus_{k \in T\setminus\{i\}} A_{i,k} \right) \\ &= \bigoplus_{k \in T\setminus\{i\}} \pi[i] \otimes A_{i,k} = \pi[k] = A^*_{i_0,k}.\end{aligned}$$

Dans le cas où $\gamma$ ne passe pas par $i$, on a $\omega(\gamma) = \pi[i']$ et $\pi[k] = A^*_{i_0,k}$. L'algorithme *termine* lorsque $\pi = A^*[i_0]$. □

**Proposition 16 (complexité de l'algorithme de Dijkstra).** *Pour un graphe à n sommets, l'algorithme de Dijkstra s'exécute en $O(n^2)$ opérations.*

*Démonstration.* La boucle externe est itérée sur $k$ avec Card $T = n - k + 1$. L'extraction de l'indice minimum se fait naïvement en $O(|T|)$ (il est possible de faire mieux selon la structure de données choisie). La boucle interne s'exécute également en $O(|T|)$. D'où une complexité globale :

$$\sum_{k=1}^{n} 2(n - k + 1) = O(n^2).$$

□

Sans chercher à être efficace, on peut très facilement implémenter, avec un logiciel de calcul comme Maple, la version générale de l'algorithme de Dijkstra, en utilisant les opérateurs inertes &+ et &< que l'on peut ensuite définir selon le dioïde sur lequel

on travaille :

Algorithme 3. Dijkstra, implémentation Maple.

```
1  dijkstra := proc(A, n, r)
2    local X, T, i, j, g, chemins;
3
4    X := [N$n]; X[r] := E;
5    T := {$1..n};
6    chemins := [[r]$n];
7
8    while nops(T)>0 do
9     g := N;
10    for j in T do
11     if X[j] &< g then g := X[j]; i := j; end;
12    od;
13    T := T minus {i};
14    for j in T do
15     if (X[i] &* A[i,j]) &< X[j] then
16      X[j] := X[i] &* A[i,j];
17      chemins[j] := [op(chemins[i]), j];
18     fi;
19    od;
20   od;
21   return X, chemins;
22  end proc:
```

# IV Quelques applications

## IV.1 Plus court chemin

Dans l'algèbre (min, +), tous ces algorithmes calculent les poids des plus courts chemins, mais aussi, comme on le voit dans l'implémentation ci-dessus (ligne 17), lors des sommes, l'indice minimal permettant effectivement de construire un plus court chemin. Voici un exemple d'exécution de la procédure :

```
'&<' := (x,y) -> evalb(x &+ y = x):
'&+' := min: '&*' := '+': E := 0: N := infinity:
```

$$A := \begin{bmatrix} 0 & 18 & \infty & 25 & \infty \\ 61 & 0 & 62 & \infty & \infty \\ \infty & 22 & 0 & \infty & 23 \\ \infty & 66 & 13 & 0 & \infty \\ 30 & \infty & 69 & 52 & 0 \end{bmatrix}$$

```
dijkstra(A, 5, 5);
```

[30, 48, 65, 52, 0], [[5, 1], [5, 1, 2], [5, 4, 3], [5, 4], [5]]

La première liste contient la cinquième ligne de $A^*$, la seconde est un vecteur contenant les plus courts chemins dont l'origine est le cinquième sommet, sous forme d'une suite de sommets.

## IV.2 Accessibilité

Tous ces algorithmes peuvent être mis en œuvre directement dans l'algèbre de Boole $(\mathbb{B}, \vee, \wedge)$. La semi-inverse $A^*$ est alors appellée *fermeture transitive* du graphe $G(A)$, dont les termes $(A^*)_{i,j}$ valent $\mathbb{1}$ s'il existe un chemin entre $i$ et $j$ et $\mathbb{0}$ sinon.

## IV.3 Cheminement avec temps de parcours variables

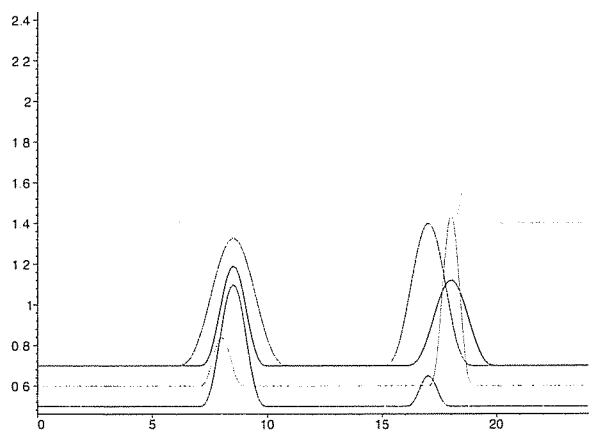
On a vu que les endomorphismes d'un dioïde forment un dioïde. Dans le cas de $\hat{\mathbb{R}}_{\min}$, ce dioïde $(\mathcal{F}, \min_{t_0}, \circ)$ est celui des fonctions croissantes de $\mathbb{R}^+$ telles que $f(\mathbb{0}) = \mathbb{0}$, où $\min_{t_0}$ est le minimum de la fonction prise au temps de départ $t_0$. Avec Maple, on déclarera par exemple les opérateurs :

```
'&+' := (f,g)->'if'(evalf(f(t0))<evalf(g(t0)), f, g):
'&*' := '@': '&<' := (x,y)->evalb(x &+ y = x):
 E := x->x: N := x->infinity:
```

Prenons l'exemple d'un réseau routier soumis à des variations de trafic. À chaque route reliant $i$ à $j$, on associe une fonction $t_{i,j}$ croissante qui à une heure de départ associe une heure d'arrivée. Cette fonction permet de prendre en compte d'éventuelles heures creuses ou de pointe. Au graphe $\mathcal{G}$, on associe donc une matrice $T = (t_{i,j})_{i,j}$ à valeurs dans un dioïde sélectif dans lequel la fonction identité (temps de parcours nul) est minimale pour $\oplus$.

Les algorithmes de Dantzig et Dijkstra permettent donc le calcul du temps de cheminement entre deux sommets quelconques, pour une heure de départ $t_0$ fixée, et éventuellement de l'itinéraire associé. Cet itinéraire dépend de $t_0$ : il est possible qu'à certaines heures un itinéraire soit meilleur qu'un autre. Ceci est illustré par l'exemple suivant :

```
f := (x, hd, hf, c)->piecewise(x<hd, 0, x>hf, 0,
  c*sin(Pi*(x-hd)/(hf-hd))^3):
g := (x, hd, hf, c)->f(x-floor(x/24)*24, hd, hf, c):
f12 := x->x+0.5*(1+g(x,7,10,1.2)+g(x,16,18,0.3)):
f21 := x->x+0.6*(1+g(x,7,9,0.4)+g(x,17,19,1.4)):
f13 := x->x+1.3*(1+g(x,8,11,0.4)+g(x,15,17,0.5)):
f31 := x->x+1.4*(1+g(x,6,8,0.2)+g(x,18,20,0.4)):
f23 := x->x+0.7*(1+g(x,6,11,0.9)+g(x,15,19,1)):
f32 := x->x+0.7*(1+g(x,7,10,0.7)+g(x,16,20,0.6)):
plot(map(f->f-E,{f12,f21,f13,f31,f23,f32}),0..24);
```



```
A := matrix(3,3,[[E,f12,f13],[f21,E,f23],[f31,f32,E]]):
t0 := 8:
dijkstra(A, 3, 1);
```

$$[E, E@f12, E@f13], [[1], [1,2], [1,3]]$$

```
t0 := 12:
dijkstra(A, 3, 1);
```

$$[E, E@f12, E@f12@f23)], [[1], [1,2], [1,2,3]]$$

Dans cet exemple, pour aller du sommet 1 au sommet 3, il est plus court de prendre la route directe à 8h00, et de passer par le sommet 2 à midi.

## V Conclusion

Outre les cas simples précédents, rien n'empêche d'utiliser des dioïdes produits pour la recherche de chemins multi-critères, ou encore les endomorphismes de dioïdes sur des ensembles abstraits (par exemple, sur des horaires possibles de départ pour les transports en commun afin d'optimiser les correspondances).

Le changement de structure algébrique a permis de montrer que la résolution du problème du plus court chemin se résume à un simple système linéaire. En outre, le calcul de la semi-inverse se fait à l'aide de la série géométrique de la matrice du graphe : on calcule en quelque sorte $(I - A)^{-1}$ sans même que $-x$ ou même $x^{-1}$ n'existent dans le dioïde !

## Références


[1] G. Cohen, S. Gaubert et J.-P. Quadrat, « L'algèbre des sandwichs », *Pour la Science* **328** (février 2005) 56–63. http://www-rocq.inria.fr/metalau/quadrat/PourlaScience.pdf.

[2] T.H. Cormen, C.E. Leiserson, R.R. Rivest et C. Stein, Plus courts chemins à origine unique, *In Introduction à l'algorithmique*, Dunod, 2e édition, 2005, pp. 563–600.

[3] G.B. Dantzig, *All shortest routes in a graph*, Stanford University, 1966.

[4] A. Girault, Une introduction à l'algèbre (max, +). ftp://ftp.inrialpes.fr/pub/bip/pub/girault/Presentations/max-plus-2up.ps.gz.

[5] M. Gondran et M. Minoux, *Graphes, dioïdes et semi-anneaux*, Tec & Doc, janvier 2002.

[6] Page du projet MaxPlus à l'INRIA, http://www-rocq.inria.fr/MaxplusOrg.