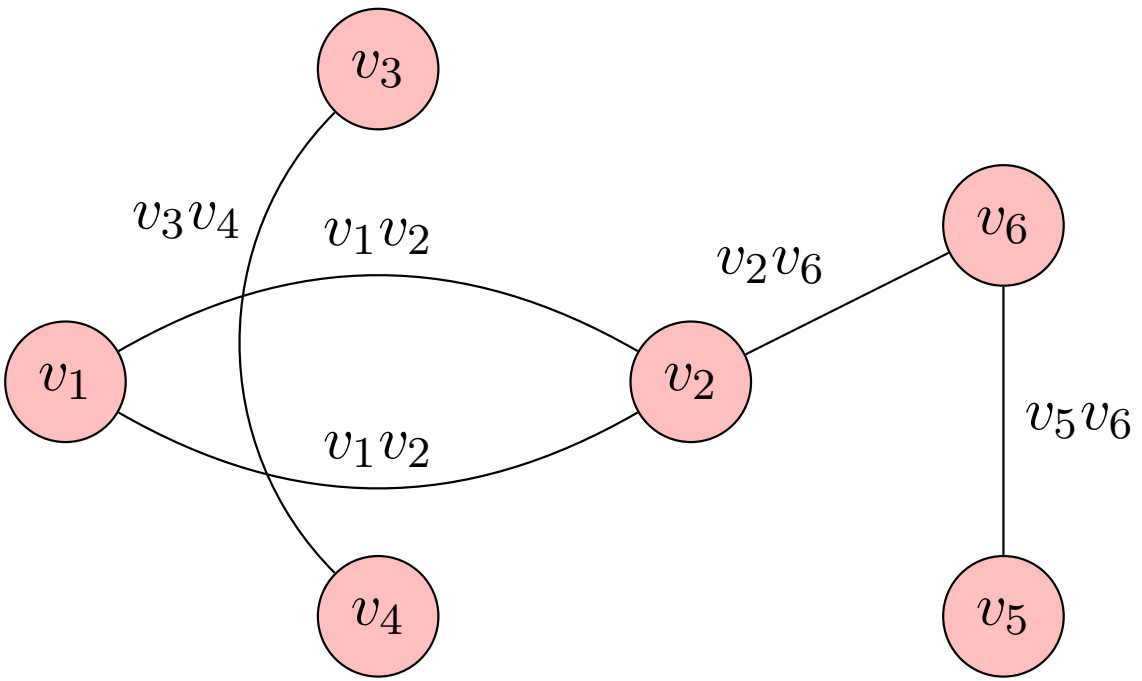
2. Plus Courts Chemins et Programmation Dynamique

4 Octobre 2023

Rappel sur les graphes

Graphe non orienté : $G = (V, E)$

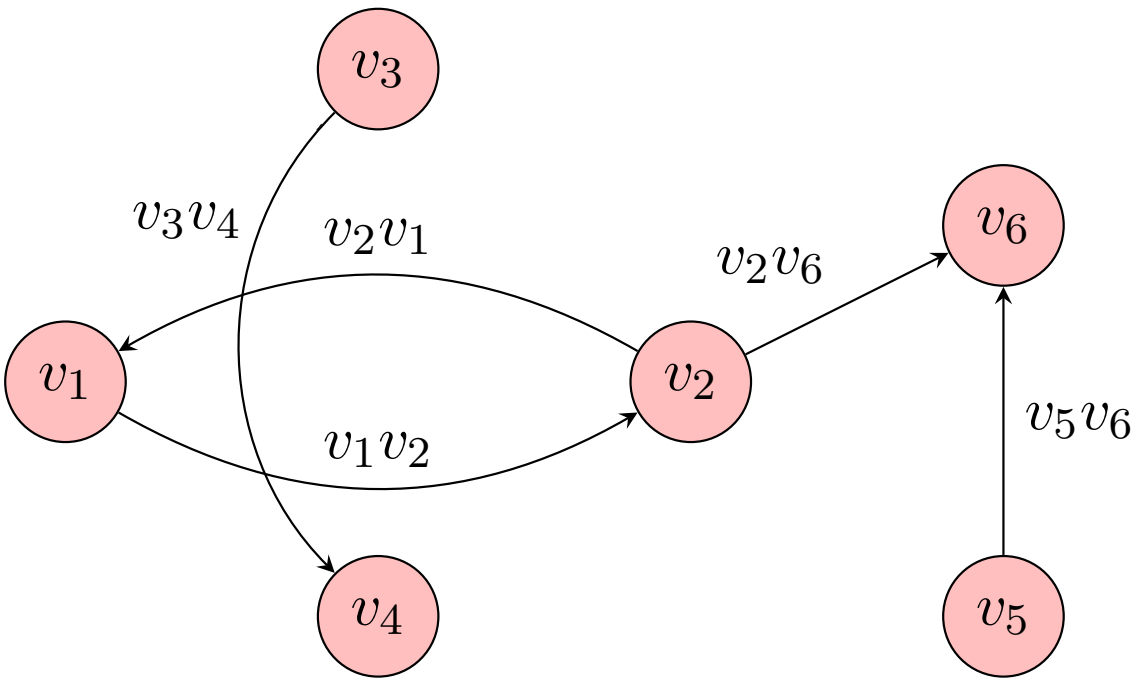
- V : ensemble de **sommets/noeuds**
- E : ensemble d'**arêtes** (paires non ordonnées de sommets)



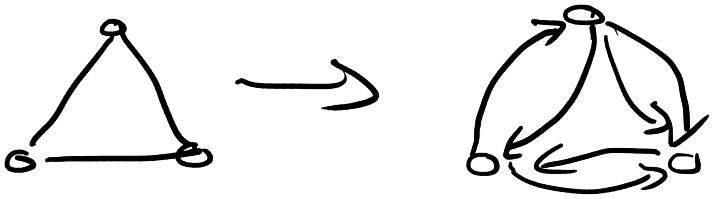
Rappel sur les graphes

Graphe orienté : $D = (V, A)$

- V : ensemble de **noeuds/arêtes**
- A : ensemble d'**arcs** (paires ordonnées de sommets)

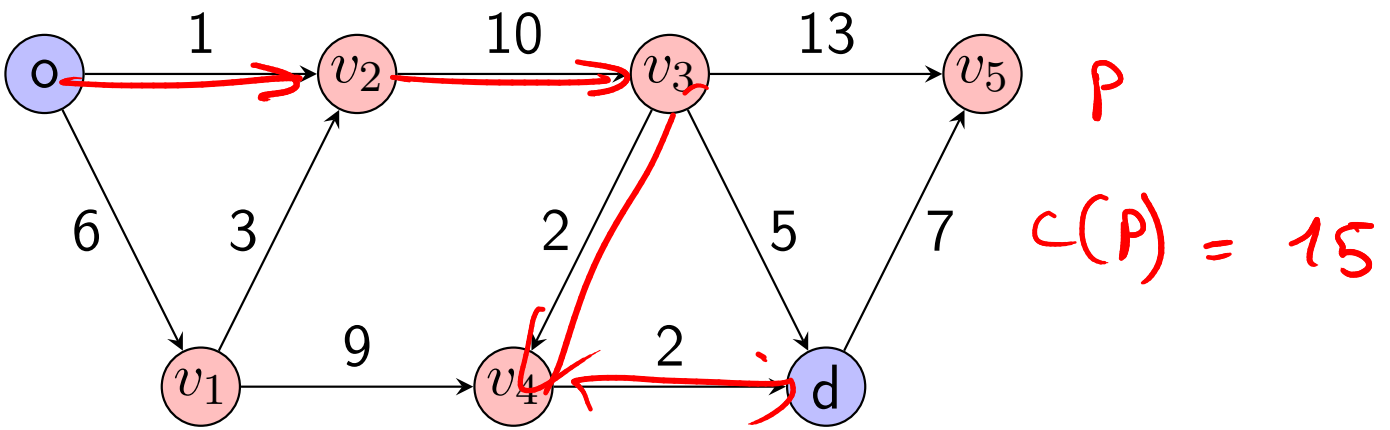


Problème de plus court chemin



Plus court chemin

- **Instance**. Un graphe orienté $D = (V, A)$, une fonction de coût $c : A \to \mathbb{R}$, et deux sommets o et d.
- **Question**. o-d chemin (élémentaire) P de coût minimum $\sum_{a \in P} c(a)$.

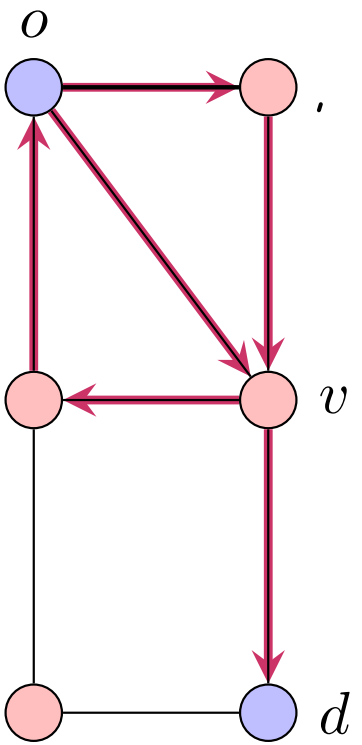


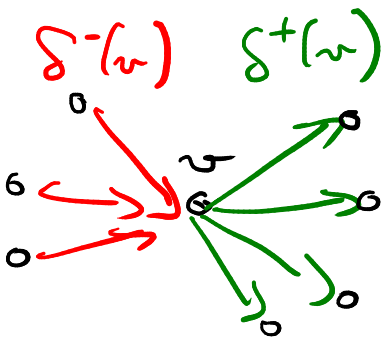
Une version non orientée peut aussi être considérée.

Formulation PLNE

$$\min_x \quad \sum_{a \in A} c_a x_a$$

$$x_a \in \{0,1\}, \quad \forall a \in A$$





$$\delta^+(v) = \{a = (v,u) \in A\}, \quad \delta^-(v) = \{a = (u,v) \in A\}$$

$$\begin{cases} \underbrace{\sum_{a \in \delta^+(o)} x_a} - \sum_{a \in \delta^-(o)} x_a = 1 \\ \text{nb d'arcs sortants de } O \text{ dans le chemin} \\ \sum_{a \in \delta^-(d)} x_a - \sum_{a \in \delta^+(d)} x_a = 1 \end{cases}$$

$$\sum_{a \in \delta^+(v)} x_a = \sum_{a \in \delta^-(v)} x_a \qquad \forall v \neq o, d$$

Formulation PLNE

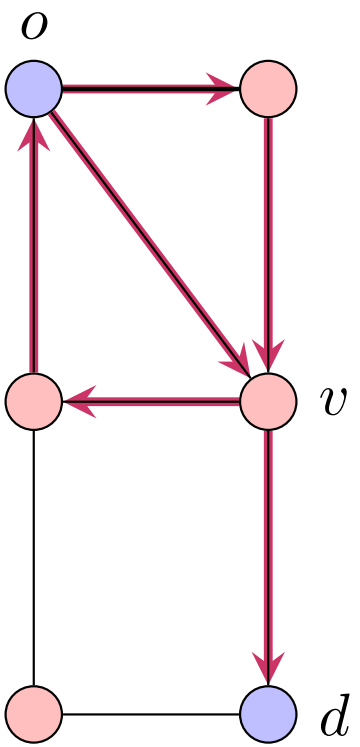
$$\min_x \quad \sum_{a \in A} c_a x_a$$

$$\text{s. t.} \quad \sum_{a \in \delta^+(v)} x_a - \sum_{a \in \delta^-(v)} x_a = 0, \qquad \forall v \notin \{o, d\}$$

$$\sum_{a \in \delta^+(o)} x_a - \sum_{a \in \delta^-(o)} x_a = 1,$$

$$\sum_{a \in \delta^+(d)} x_a - \sum_{a \in \delta^-(d)} x_a = -1$$

$$x_a \in \{0, 1\}, \qquad \forall a \in A$$



$$\delta^+(v) = \{a = (v, u) \in A\}, \quad \delta^-(v) = \{a = (u, v) \in A\}$$

Complexité du problème ?

Complexité du problème

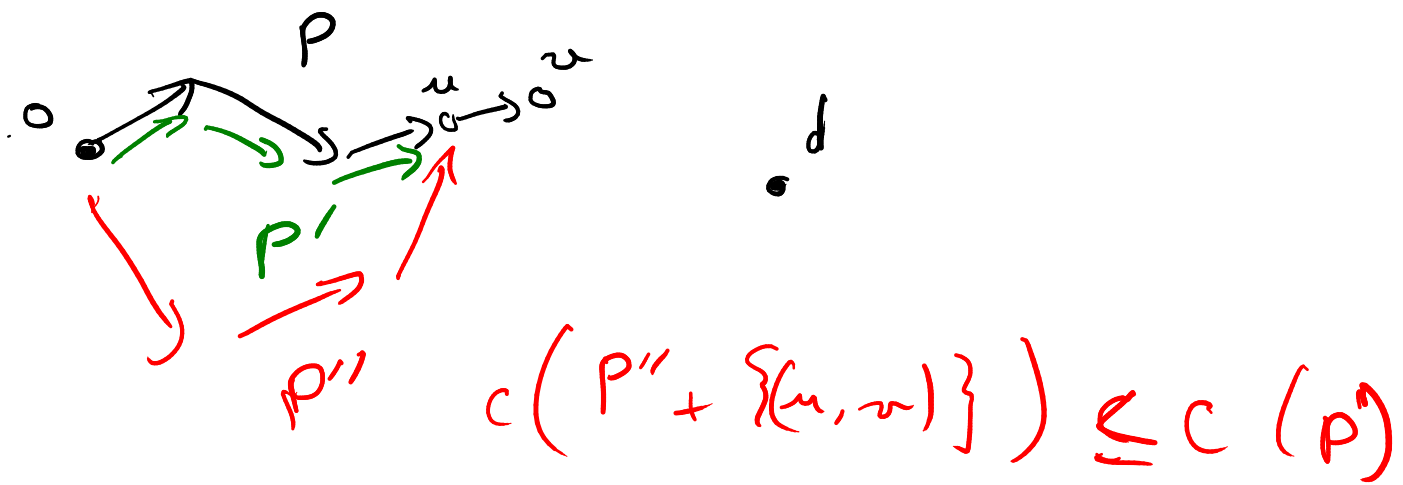
- NP-difficile dans le cas général
- Polynomial dans plusieurs cas particuliers
  1. **Pas de fonction de coût** ($c = 1$) : parcours en largeur
  2. **Acyclique** :
     - Orienté : ordre topologique
     - Non orienté (forêt) : au plus un o-d chemin élémentaire
  3. **Sans cycle absorbants** :
     - Orienté : programmation dynamique
     - Non orienté : T-joints
  4. **Poids positifs** : algorithme de Dijkstra

Cycle absorbant (=cycle de coût négatif)

Un cycle absorbant est un cycle C tel que $\sum_{a \in C} c_a < 0$.

Principe d'optimalité de Bellman

Proposition. Soit P un o-v chemin avec $k > 0$ arcs, u le sommmet juste avant v dans P, et P' le o-u sous chemin de P (en enlevant l'arc (u, v)). Si P est un plus court chemin de o à v, alors P' est un plus court o-u parmi les o-u chemins avec $k - 1$ arcs.



Preuve

Equation de Bellman

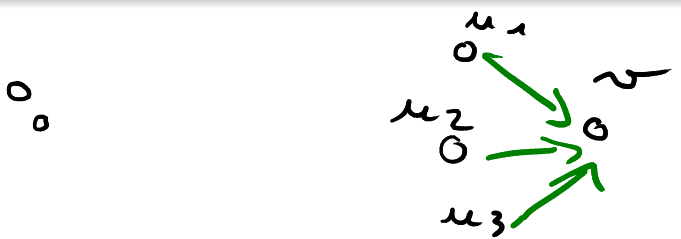
Contexte : graphe orienté sans cycle absorbant

Fonction valeur : $f : V \times \mathbb{Z}_+ \to \mathbb{R} \cup \{+\infty\}$ telle que :

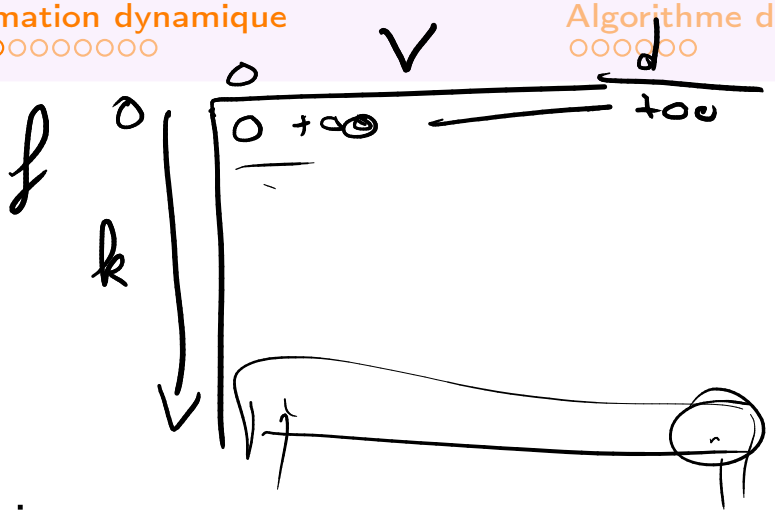
- $f(v,k) =$ coût du plus court chemin de o à v qui a au plus k arcs.
- $f(v,0) = 0$ si $v = o$, $+\infty$ sinon

Equation de Bellman/Equation de programmation dynamique

$$f(v,k+1) = \min_{\substack{a=(u,v) \\ a \in \delta^-(v)}} [c_a + f(u,k)]$$



Algorithme de Ford-Bellman



- Initialisation :
  - Tableaux 2d f et a
  - $f(v,0) \leftarrow 0$ si $v = o$, $+\infty$ sinon
- Pour k allant de 1 à $|V|-1$, $\forall v \in V$:
  - $f(v,k) \leftarrow \min\limits_{\substack{a=(u,v)\\ a\in\delta^-(v)}} [c_a + f(u,k-1)]$
  - $a(v,k) \leftarrow \arg\min\limits_{\substack{a=(u,v)\\ a\in\delta^-(v)}} [c_a + f(u,k-1)]$
- Retourner : $\min\limits_{k\in\{0,\dots,|V|-1\}} f(d,k)$

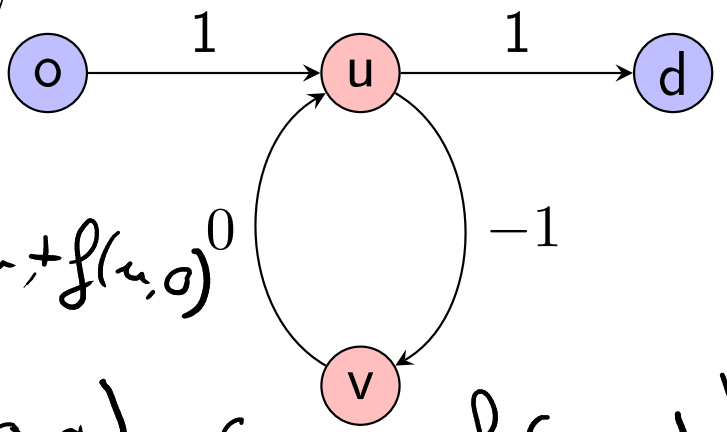
Proposition. L'algorithme retourne un plus court o-d chemin en $\mathcal{O}(|V| \cdot |A|)$

Reconstruction du chemin optimal

Pourquoi ça ne marche pas avec un cycle absorbant ?

$f(0,1) = \min_{(v,0)\in\delta^-(0)} \left(f(v,0) \right)$

$f(u,1) = c_{uu} + f(u,0)$
$= +\infty$



$f(0,1) = \min_{(u,0)\in\delta^-(0)} c_{uu} + f(u,0)$
$= 0$

$f(u,1) = \min(c_{0u} + f(0,0),\ c_{uu} + f(u,0)) = 1$

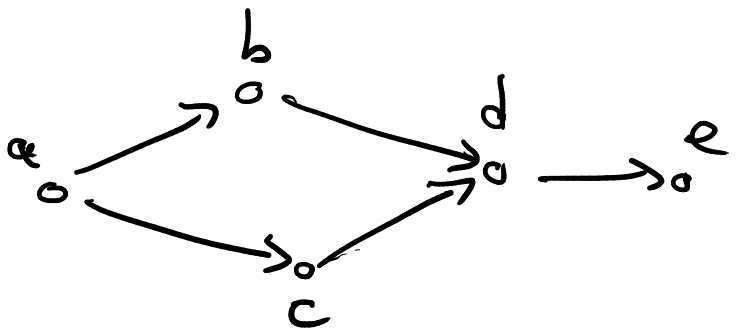
	0	v	u	d
0	0	$+\infty$	$+\infty$	$+\infty$
1	0	$+\infty$	1	$+\infty$
2	0	0	1	2
3	0	0	0	2
			−1	1

Graphe acyclique

- **Graphe acyclique** : ne contient pas de cycle.
- Un **ordre topologique** $<$ sur $D = (V, A)$ est un ordre sur V tel que : s'il existe un u-v chemin, alors $u < v$.

Proposition. Il existe un ordre topologique sur $D \Leftrightarrow$ D est acyclique

$\Longrightarrow$ **Contexte** : graphe orienté acyclique



ordre : [a, b, c, d, e] ou [a, c, b, d, e]

$D = (V, A)$

Calcul d'un ordre topologique sur un graphe acyclique

Initialisation : $L = [\,]$, $S = \{u \in V, \delta^-(u) = \emptyset\}$

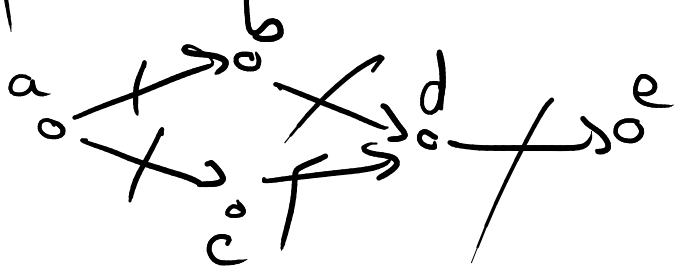
Tant que $S \neq \emptyset$.

$u \in S$.

$L \leftarrow L + [u]$

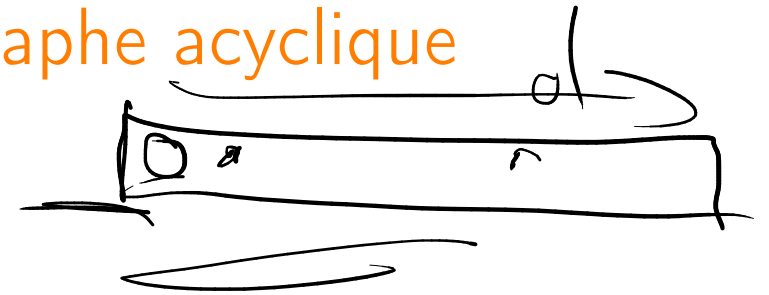
On retire de A tous les arcs de $\delta^+(u)$

On ajoute dans S tous les sommets v t.q $\delta^-(v) = \emptyset$



- $L = [\,]$, $S = \{a\}$
- $u = a$, $L = [a]$, $S = \{b, c\}$
- $u = c$, $L = [a, c]$, $S = \{b\}$
- $u = b$, $L = [a, c, b] = S = \{d\}$

Programmation dynamique sur un graphe acyclique



$f(v)$: longeur d'un plus court o-v chemin :

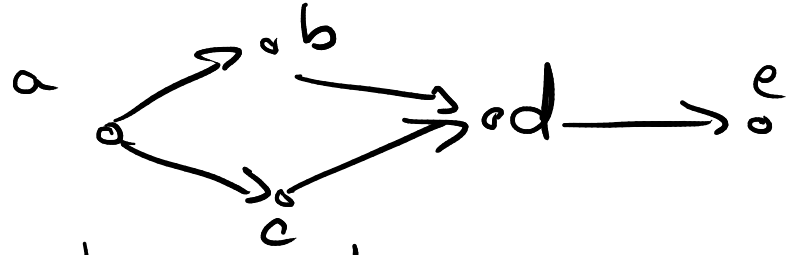
- $f(v) = 0$ si $v = o$
- $f(v) = \min_{a=(u,v)\in\delta^-(v)} f(u) + c_a$

$f(d)$

Algorithme

Calculer $f(v)$ en parcourant les sommets dans l'ordre topologique.

Proposition. L'algorithme retoune un plus court o-d chemin en $\mathcal{O}(|V| + |A|)$



$[a, b, c, d, e]$

| a b c d e

f : 0

$f(b) = c_{ab} + f(a)$ $\quad f(c) = c_{ac} + f(c)$

$f(e) = c_{de} + f(d)$

$f(d) = \min\left(c_{bd} + f(b),\ c_{cd} + f(c) \right)$

Programmation dynamique : généralisation

Système dynamique discret :

- Horizon temporel fini $[\![1, T]\!]$
- Espace d'état $\mathcal{S}$ fini
- Etat initial s_{ini}
- Espace d'action $\mathcal{U}$ fini
- Etat du système à l'instant t : s_t
- Fonction de transition f telle que $s_{t+1} = f(s_t, u_t)$
- Coût de l'action u en l'état s : $c(u, s)$
- Coût d'être en l'état s au temps T : $c_T(s)$

Programmation dynamique : généralisation

$S = V$

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sum_{t=0}^{T-1} c(u_t, s_t) + c_T(s_T) \\ \text{s. t.} \quad & s_0 = s_{ini}, \\ & s_{t+1} = f(s_t, u_t), && \forall t \in [\![0, T-1]\!] \\ & u_t \in \mathcal{U}(s_t) && \forall t \in [\![0, T-1]\!] \end{aligned}$$

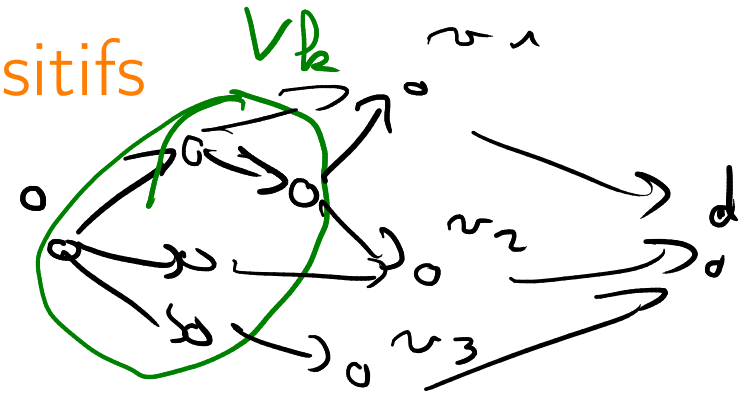
Fonction valeur : $V(t, s)$ coût d'une solution optimale du problème sur $[\![t, T]\!]$ si le système est en s au temps t

Equations de programmation dynamique

- $V(T, s) = c_T(s)$
- $V(t, s) = \min_{u \in \mathcal{U}} c(s, u) + V(t+1, f(s, u)), \ \forall t \in [\![0, T-1]\!]$

$\Longrightarrow$ calculable en $\mathcal{O}(|\mathcal{S}| \cdot |\mathcal{U}| \cdot T)$

Cas des graphes à poids positifs



Contexte : poids positifs, $\forall a \in A,\ c_a \geq 0$

Proposition. Soit $k < |V|$, et V_k un ensembles des k plus proches sommets de o. Soit $f(u)$ la longueur du plus court o-u chemin pour $u \in V_k$. Alors $V_{k+1} = V_k \cup \{v\}$ avec
$v \in \arg\min_{v \in V}[\min_{\substack{(u,v)\in\delta^-(v)\\ u\in V_k}} f(u) + c_{u,v}]$ est un ensemble de $k+1$ plus proches sommets de o.

v le sommet de $V \setminus V_k$
qui minimise $\min_{\substack{u \in V_k\\ (u,v)\in A}} c_{u,v} + f(u)$

Algorithme de Dijkstra

$= f(v)$ à la fin

Initialisation : $d_v = \begin{cases} 0 & \text{si } v = 0 \\ +\infty & \text{sinon} \end{cases}$ $\quad U = \emptyset = V_0$

Tant que $V \setminus U \neq \emptyset$

$v \in V \setminus U$ qui minimise d_v

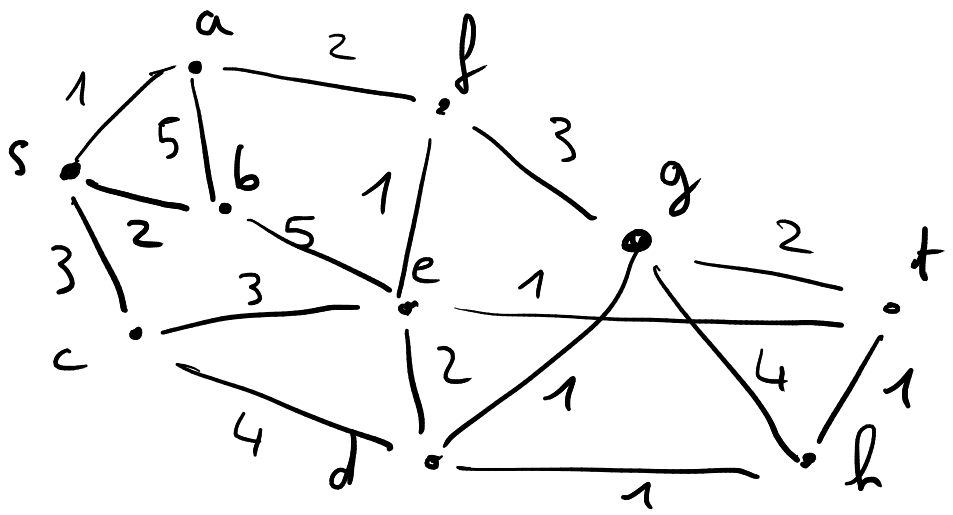
Ajouter v à U : $U \leftarrow U + \{v\} = V_{k+1}$

$\forall w$ t.q. $(v, w) \in \delta^+(v)$

$d_w \leftarrow \min(d_w, d_v + c_{v,w})$

$\begin{cases} N^-(v) = \{u \in V \mid (u, v) \in A\} \\ N^+(v) = \{u \in V \mid (v, u) \in A\} \end{cases}$

Exemple



$V_0 = \emptyset$

$V_1 = \{s\}$

$V_2 = \{s, a\}$

$V_3 = \{s, a, b\}$

$V_4 = \{s, a, b, c\}$

$V_5 = \{s, a, b, c, f\}$

$V_6 = \{s, a, b, c, f, e\}$

$V_7 = \{s, a, b, c, f, e, t\}$

$d_a = \min\{d_a, d_s + c_{a,s}\}$

$= \min\{+\infty, 1\}$

$d_f = \min\{d_f, d_a + c_{f,a}\}$

$= \min\{+\infty, 1+2\}$

$d_b = \min\{d_b, d_a + c_{b,a}\}$

$= \min\{2, 1+5\} = 2$

s	a	b	c	d	e	f	g	h	t
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
0	1 (s)	2	3	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
0	1	2	3	$+\infty$	7 (6)	3 (a)	$+\infty$	$+\infty$	$+\infty$
				7 (c)	6 (c)				
				7	4 (f)		6		
				6					5 (e)

$\{e, f, a, s\}$

Algorithme $A^\star$

Instance : $D = (V, A)$, $(c_a)_{a \in A}$, $o, d \in V$

$(b_v)_{v \in V}$

b_v : borne inférieure sur la valeur d'un plus court v-d chemin

Si on a un o-v chemin P, $c_P + b_v$ est une borne inf sur la valeur d'un o-d chemin contenant P.

Si on a un o-d P' de coût $c_{P'}$

Si $c_P + b_v > c_{P'}$

5.3

5.16

5.19