

# Recherche Opérationnelle :

## TP 3 et 4

Sandra U. Ngueveu (ngueveu@laas.fr), Arthur Claviere, Aloïs Duguet,  
Alexandre Dupaqueis, Quentin Fabry

2020

### Table des matières

<b>1</b>	<b>Programmation dynamique avec Bellman-Ford</b>	<b>2</b>
1.1	Calcul du plus court chemin entre deux sommets d'un graphe . . . . .	2
1.2	Calcul du plus long chemin entre deux sommets d'un graphe (sera aussi utilisé comme brique de base pour le TP 5-6) . . . . .	2
<b>2</b>	<b>Extensions et adaptations</b>	<b>2</b>
2.1	Construction d'un réseau de transmission à vitesse maximale . . . . .	2
2.2	Fiabilité de procédé de fabrication de semi-conducteurs . . . . .	3
<b>3</b>	<b>BONUS : optimisation d'énergie pour un véhicule électrique hybride</b>	<b>4</b>
<b>4</b>	<b>ANNEXE : rappel de l'exemple du cours</b>	<b>5</b>

# 1 Programmation dynamique avec Bellman-Ford

## 1.1 Calcul du plus court chemin entre deux sommets d'un graphe

Implémenter en Julia l'algorithme de Bellman-Ford permettant de calculer le plus court chemin entre un sommet source  $s$  et tous les autres sommets d'un graphe quelconque. A titre d'exemple vous pouvez vous baser sur l'exemple donné en cours et rappelé en annexe. Tester ensuite avec différents jeux de données de votre choix.

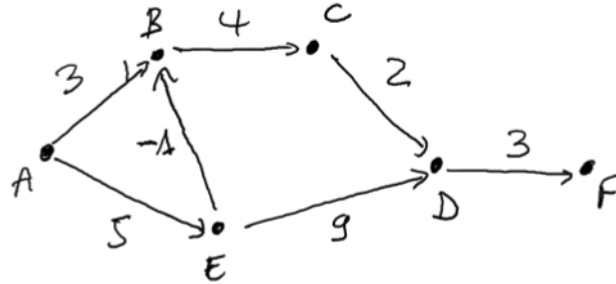


FIGURE 1 – Exemple de graphe quelconque

## 1.2 Calcul du plus long chemin entre deux sommets d'un graphe (sera aussi utilisé comme brique de base pour le TP 5-6)

Quelles modifications faut-il appliquer à la relation de récurrence et à l'algorithme de programmation dynamique de Bellman-Ford pour calculer le plus long chemin avec au maximum  $n - 1$  arêtes entre deux sommets d'un graphe à  $n$  sommets ? Passer plusieurs fois par la même arête ou le même sommet est autorisé. Implémenter et tester avec différents jeux de données de votre choix.

# 2 Extensions et adaptations

## 2.1 Construction d'un réseau de transmission à vitesse maximale

La société ALPHA désire relier par un réseau, un processeur central  $P$  à divers processeurs (éventuellement par des intermédiaires). Une vitesse de transmission  $l_{ij}$  en bauds est associée à chaque liaison possible (bidirectionnelle) reliant le processeur  $i$  au processeur  $j$ . La vitesse de transmission d'un chemin du processeur central  $P$  au processeur  $K$  est égale au plus petit des  $l_{ij}$  rencontrés sur ce chemin. ALPHA souhaite trouver les chemins de vitesses maximales reliant le processeur central  $P$  à tous les autres processeurs  $K$  du réseau. Cela nécessite donc de modifier votre algorithme pour fournir les chemins maximisant la plus petite vitesse rencontrée sur les chemins.

A titre d'exemple, considérons un réseau comportant 1 processeur central et 6 processeurs secondaires, avec les capacités des liaisons (vitesses) données par le TABLEAU 1.

	P	1	2	3	4	5	6
P		5			1		3
1	5		4				
2		4		1	2	2	
3			1			2	
4	1		2			3	5
5			2	2	3		
6	3				5		

TABLEAU 1 – Exemple de vitesses de connexion

Quelles modifications faut-il appliquer à la relation de récurrence et à l'algorithme de Bellman-Ford pour calculer le chemin de vitesse maximale entre deux sommets d'un graphe ? Implémenter et tester sur l'exemple fourni puis avec différents jeux de données de votre choix.

## 2.2 Fiabilité de procédé de fabrication de semi-conducteurs

L'entreprise de semi-conducteurs GAMMA souhaite déterminer le procédé de fabrication le plus sûr pour leurs nouveaux processeurs SX-42. Il y a en effet plusieurs façons de faire pour transformer la matière première jusqu'au produit fini (enchaînement d'opérations de gravure, de vernissage, de nettoyage, de dopage, etc). ; Le processus de fabrication de processeurs est extrêmement sensible et si à une étape il y a le moindre problème alors la plaque de silicium est perdue et détruite. La FIGURE 2 décrit les étapes intermédiaires possibles, et la probabilité de succès de chaque étape.

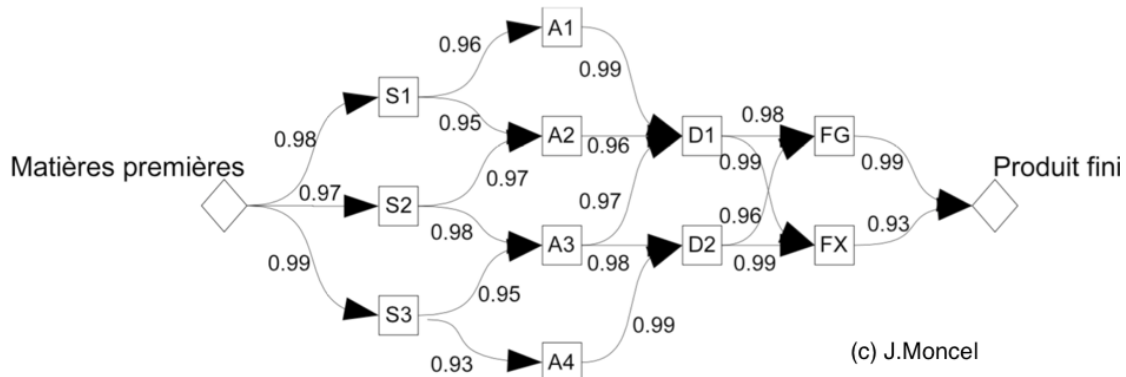


FIGURE 2 – Exemple de processus de fabrication

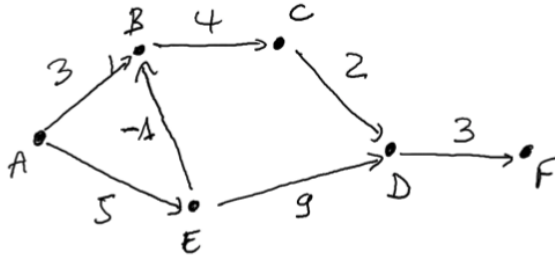
Ci-dessus le schéma descriptif des processus de fabrication possibles. Les produits semi-finis intermédiaires sont désignés par les abréviations S1, S2, S3, A1, A2, A3, A4, D1, D2, FG, FC. La probabilité de succès est indiquée pour chaque étape possible. Par exemple si l'on dispose de produits semi-finis de type S2, alors on a le choix entre deux processus. Le premier donnera des produits semi-finis de type A2, avec une probabilité de succès à 4s de 0.97 (il y aura donc 3% déchets). Le second processus fournira des produits semi-finis de type A3 avec une probabilité sur le période, avec une probabilité de succès de 0.98 (il y aura donc 2% de déchets). La problématique est de déterminer le procédé de fabrication le plus sûr, c-à-d comportant moins de déchets.

- (a) En utilisant le fait que pour tout  $a, b \in \mathbb{R}^+$  on a  $\log ab = \log a + \log b$ , montrer que l'on peut transformer la recherche du procédé de fabrication le plus sûr en un problème de plus long chemin dans un graphe à préciser. Implémenter et tester sur l'exemple fourni puis avec différents jeux de données de votre choix.
- (b) On peut aussi directement modifier l'algorithme de Bellman-Ford vu dans ce TP pour l'adapter à la recherche d'un plus sûr chemin, c.-à-d en un chemin  $x_1, \dots, x_k$  tel que le produit  $\prod_{i=1}^{k-1} p(x_i, x_{i+1})$  soit maximum. Implémenter et tester sur l'exemple fourni puis avec différents jeux de données de votre choix, et comparer au résultat précédent.

### 3 BONUS : optimisation d'énergie pour un véhicule électrique hybride

Appliquer la programmation dynamique pour obtenir des solutions au problème d'optimisation d'énergie pour véhicule électrique hybride donné en bonus sur Moodle, après discrétisation des niveaux d'énergie de la batterie.

#### 4 ANNEXE : rappel de l'exemple du cours



itération	A	B	C	D	E	F
0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	$3^A$	$+\infty$	$+\infty$	$5^A$	$+\infty$
2	0	$3^A$	$7^B$	$14^E$	$5^A$	$+\infty$
3	0	$3^A$	$7^B$	$9^C$	$5^A$	$17^D$
4	0	$3^A$	$7^B$	$9^C$	$5^A$	$12^D$
5	0	$3^A$	$7^B$	$9^C$	$5^A$	$12^D$

Données

- Ce graphe a  $n = 6$  sommets (A, B, ..., F) et  $m = 7$  arcs ((AB), (AE), ... (DF))
- Chaque arc  $ij$  a un coût  $c_{ij}$  (par exemple  $c_{AB} = 3$ )

Relation de récurrence

- Soit  $f_i^k$  la valeur du plus court chemin du sommet de départ (A) et un sommet  $i$  calculé à l'itération  $k$
- La relation de récurrence qui s'applique est  $f_i^k = \min_{j \in \text{Pred}(i)} f_j^{k-1} + c_{ji}$

Condition d'arrêt

- L'algorithme de Bellman-Ford s'arrête dès l'itération  $k$  qui vérifie soit  $\forall i f_i^k = f_i^{k-1}$ , soit  $k \geq n + 1$ .
- Remarque : Si le cas de figure  $k = n + 1$  se produit, alors cela démontre l'existence d'un cycle de longueur négative dans le graphe. Selon ce que modélise ce graphe, il se peut que cela permette de détecter une incohérence.

Solution obtenue

- Le plus court chemin entre A et F est A-B-C-D-F. Ce chemin a un coût de 12.

FIGURE 3 – Exemple : graphe et tableau des valeurs de  $f_i^k$  obtenu lors de l'application de l'algorithme de Bellman-Ford pour calculer le plus court chemin entre les noeuds A et F du graphe fourni