Projet RO

Badr Sajid , Mehdi Wissad

Département Sciences du Numérique - Deuxième année
2020-2021

1 Programmation dynamique avec Bellman-Ford

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

Nous avons appliqué l'algorithme de Bellman-Ford , pour cela nous avons modélisé f^k par un vecteur de n élément avec n le nombre de noeud. Pour chaque noeud on a défini ses prédécesseurs dans une liste , on a aussi utilisé une fonction intermédiaire qui nous le permet.Pour la boucle principale on fait une modification au fur et à mesure sur le f^k et on s'arrête quand $f^k = f^{k+1}$ ou bien quand $k >= n + 1$.
La relation de récurrence est la suivante :

$$f_i^k = min_{\forall j \in Pred(i)} f_j^{k-1} + c_{ji}$$

Le résultat après le test avec les données du programme est le suivant :

A $->$ A = 0.0 passant par A
A $->$ B = 3.0 passant par A
A $->$ C = 7.0 passant par B
A $->$ D = 9.0 passant par C
A $->$ E = 5.0 passant par A
A $->$ F = 12.0 passant par D

Le chemin à suivre est : A $->$ B $->$ C $->$ D $->$ F et le coût est 12 pour aller de A à F par exemple.le résultat est bien cohérent avec la donnée des problèmes.

1.2 Calcul du plus long chemin entre deux sommets d'un graphe

Pour l'algorithme il reste le même , mais la seul différence et qu'on travail avec le plus court chemin appliqué à -graphe (la matrice avec les coefficients négatifs).

Le résultat après le test avec les données du programme est le suivant :

A $->$ A = 0.0 passant par A
A $->$ B = 4.0 passant par E
A $->$ C = 8.0 passant par B
A $->$ D = 14.0 passant par E
A $->$ E = 5.0 passant par A
A $->$ F = 17.0 passant par D

Le chemin à suivre est : A $->$ E $->$ D $->$ F et le coût est 17 pour aller de A à F par exemple.le résultat est bien cohérent avec la donnée des problèmes car après calcul c'est le même résultat.

2 Extensions et adaptations

2.1 Construction d'un réseau de transmission à vitesse maximal

C'est le même algorithme de Bellman-Ford associé au plus court chemin en changeant la relation de récurrence qui devient :

$$f_i^k = max_{\forall j \in Pred(i)} min(f_i^{k-1}, cji).$$

le résultat est le suivant :

P − > P = 5.0 passant par 1
P − > 1 = 5.0 passant par P
P − > 2 = 4.0 passant par 1
P − > 3 = 2.0 passant par 5
P − > 4 = 3.0 passant par 6
P − > 5 = 3.0 passant par 4
P − > 6 = 3.0 passant par P

C'est bien le résultat souhaité en faisant un calcul.

2.2 Fiabilité de procédé de fabrication de semi-conducteurs:

2.2.1 En utilisant le problème du plus long chemin:

Pour calculer la probabilité de succès,on doit alors calculer le produit des probabilités qui interviennents dans le chemin suivi. Avec la propriètè de la croisance du log , on a $max(log(a*b) = max(log(a) + log(b)))$ ainsi avec cette formule , on peut alors utilisé Bellman-Ford pour le plus long chemin appliqué au log de la matrice (graphe qui contient les poids contenus dans les arcs) le résultat est le suivant:

- La probabilité de succès est : 0.904 (9.636% de déchets)
- Le procédé de fabrication le plus sûr est :

M − > S1 − > A1 − > D1 − > Fg − > P

2.2.2 En s'adaptant au problème:

On va prendre l'algorithme du plus court chemin en changeant, la relation de concurrence qui devient la suivante :

$$f_i^k = max_{\forall j \in Pred(i)} f_j^{k-1} * c_{ji}$$

En changeant aussi f^0 on instancie f_i^0 par $-\infty$ sauf $f_{source}^0 = 1$.
le résultat est le suivant:

- La probabilité de succès est : 0.904 (9.636% de déchets)
- Le procédé de fabrication le plus sûr est :

M − > S1 − > A1 − > D1 − > Fg − > P