Feuille de TD 4 : Le problème du plus court chemin

I. Algorithme de Bellman-Ford

Exercice 1 Exécuter l'algorithme de Bellman-Ford sur le graphe orienté de la figure 24.4, en prenant z pour origine. À chaque passage, relâcher les arcs dans le même ordre que sur la figure, et donner les valeurs de d et p après chaque passage. Ensuite, donner au poids de l'arc (z, x) la valeur 4, et exécuter à nouveau l'algorithme en prenant pour origine le sommet s.

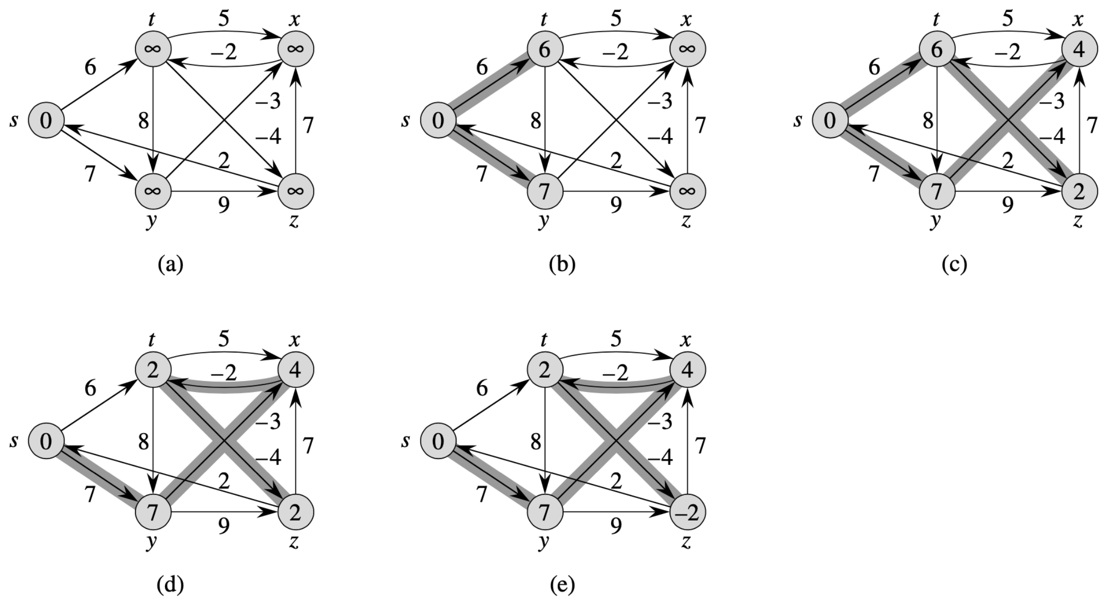

Figure 24.4 L'exécution de l'algorithme de Bellman-Ford. Le sommet origine est s. Les valeurs de d sont représentées dans les sommets, et les arcs en gris indiquent les valeurs prédécesseur : si l'arc (u, v) est en gris, alors $\pi[v] = u$. Dans cet exemple particulier, chaque passage relâche les arcs dans l'ordre $(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)$. **(a)** La situation juste avant le premier passage sur les arcs. **(b)–(e)** La situation après chacun des passages suivants. Les valeurs d et π donnée en partie (e) sont les valeurs finales. L'algorithme de Bellman-Ford retourne VRAI pour cet exemple.

Exercice 2 Démontrer le corollaire 24.3 ci-dessous.

Corollaire 24.3 *Soit $G = (S, A)$ un graphe orienté pondéré de fonction de pondération w : $A \to \mathbf{R}$ et d'origine s. Alors, pour chaque sommet $v \in S$, il existe un chemin de s vers v si et seulement si* BELLMAN-FORD *se termine avec $d[v] < \infty$ quand elle est exécutée sur G.*

Exercice 3 ÉtantdonnéungrapheorientépondéréG=(S,A)sanscircuitdelongueurstrictement négative, soit m le maximum, pour tous les couples de sommets u, $v \in S$, du nombre minimal d'arcs dans un plus court chemin de u à v. (Ici, « plus court » signifie de poids minimal et ne concerne pas le nombre d'arcs). Suggérer une modification simple à l'algorithme de Bellman-Ford, lui permettant de se terminer après $m + 1$ passages.

II. Algorithme de Dijkstra

Exercice 1 Exécuter l'algorithme de Dijkstra sur le graphe orienté de la figure 24.2, en prenant d'abord comme origine le sommet *s*, puis le sommet *z*. En s'inspirant de la figure 24.6, donner la valeur des attributs *d* et p ainsi que les sommets de l'ensemble *E* après chaque itération de la boucle **tant que**.

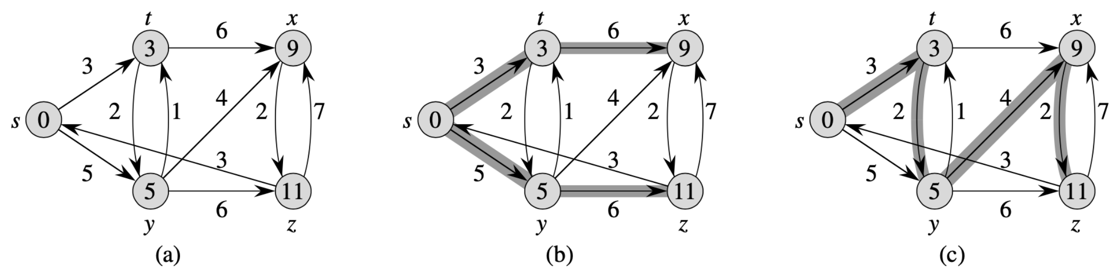

Figure 24.2 **(a)** Un graphe orienté pondéré avec des poids des plus courts chemins à partir de l'origine *s*. **(b)** Les arcs en gris forment une arborescence de plus courts chemins de racine *s*. **(c)** Une autre arborescence de plus courts chemins de même racine.

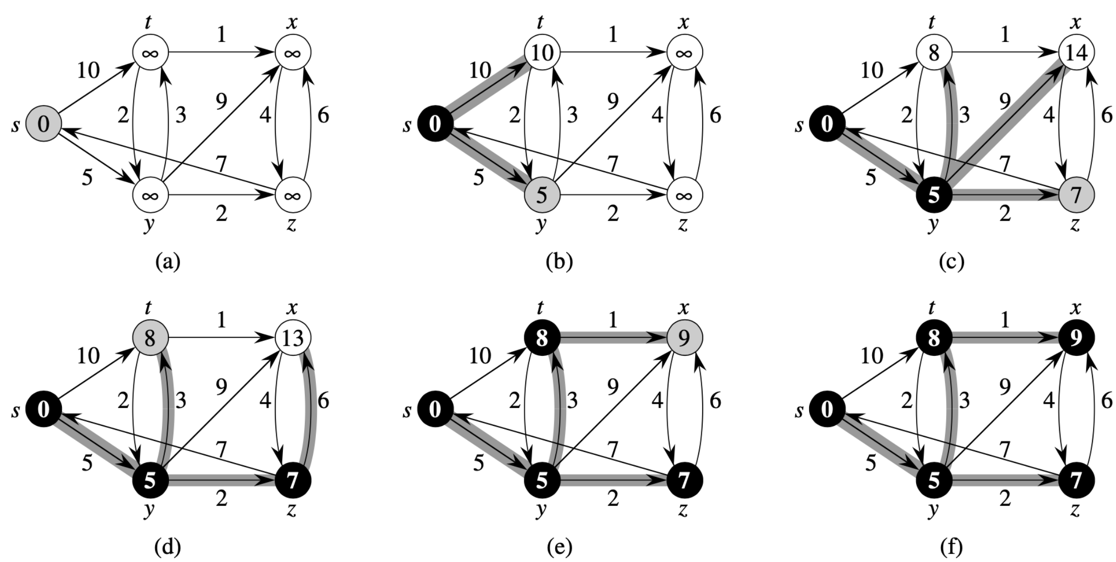

Figure 24.6 L'exécution de l'algorithme de Dijkstra. L'origine *s* est le sommet le plus à gauche. Les estimations de plus court chemin sont représentées à l'intérieur des sommets, et les arcs en gris indiquent les prédécesseurs. Les sommets noirs sont dans l'ensemble *E*, et les sommets blancs sont dans la file de priorités min *F* = *S* − *E*. **(a)** La situation juste avant la première itération de la boucle **tant que** des lignes 4–8. Le sommet gris contient la valeur minimale de *d* et est choisi comme sommet *u* à la ligne 5. **(b)–(f)** La situation après les itérations successives de la boucle **tant que**. A chaque étape, le sommet gris est celui choisi comme sommet *u* à la ligne 5 de l'itération suivante. Les valeurs *d* et π représentées en (f) sont les valeurs finales.

Exercice 2 Supposons que la ligne 4 de l'algorithme de Dijkstra soit remplacée par celle-ci :

4 **tant que** $|F| > 1$

Cette modification permet à la boucle **tant que** de s'exécuter $|S| - 1$ fois au lieu de $|S|$ fois. L'algorithme ainsi modifié est-il correct ?