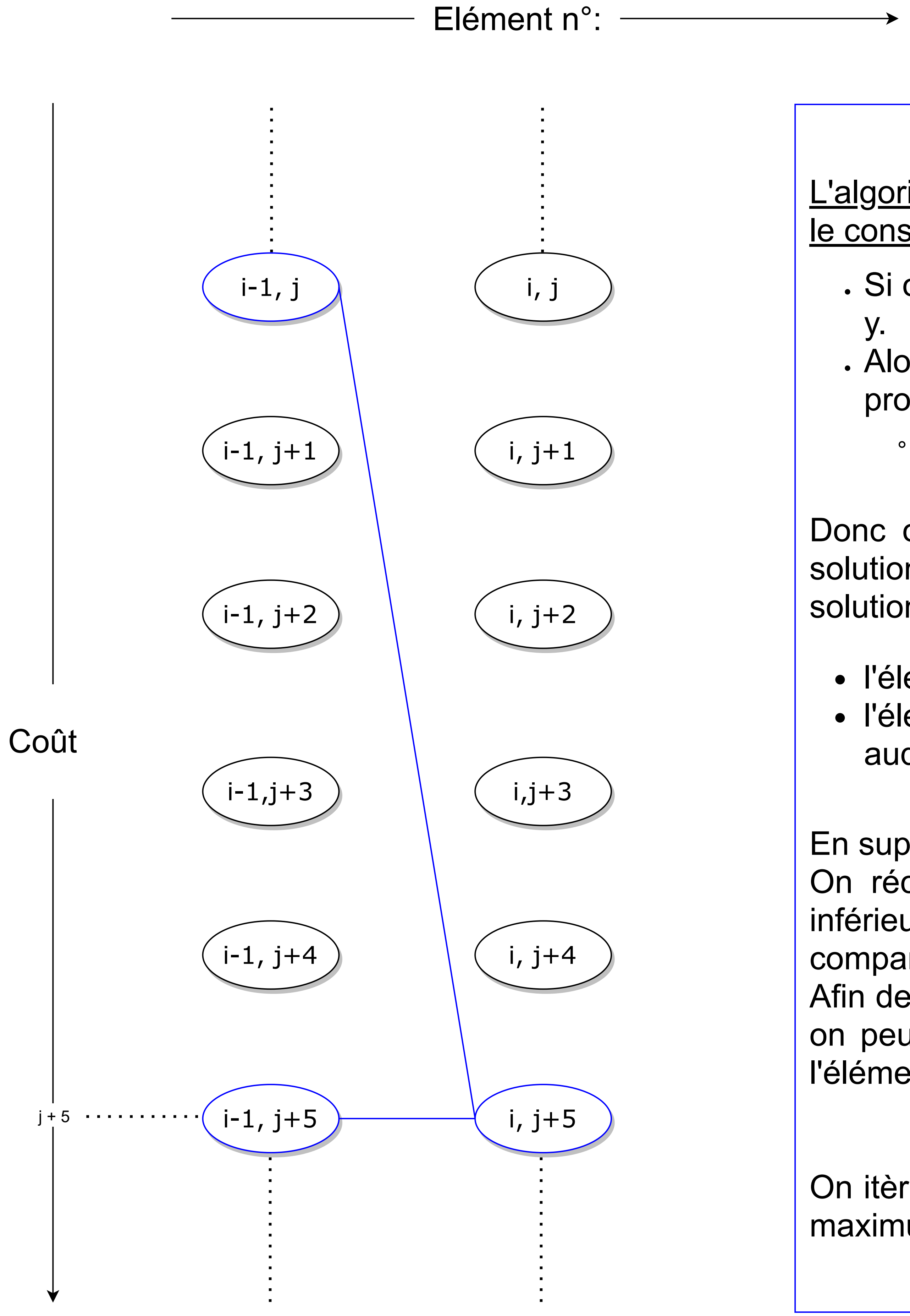

<u>L'algorithme **path_pattern** s'appuie sur la théorie des modèles de chemin et sur le constat suivant:</u>

- Si on a une solution optimale, nommé S, contenant x éléments dont l'élément y.
- Alors S contient une solution optimale de (x-1, à savoir "S - y") pour un problème d'un coût correspondant à:
  - Coût de S - coût de y

Donc on s'appuie sur les résultats de la colonne précédente pour trouver la solution optimale de la colonne en cours de traitement en choisissant la meilleure solution entre :

- l'élément de la colonne précédent (on ajoute pas l'élément i)
- l'élément de la colonne précédente à la position "i - le coût de l'élément" auquel on ajoute le bénéfice de l'élément i

En supposant que l'élément i a un coût de 5.
On récupère la valeur de l'élément de la colonne précédente avec un coût inférieur du coût de l'élément i. On y ajoute le bénéfice de l'élément i, que l'on compare au bénéfice de la colonne précédente pour le coût que l'on met à jour.
Afin de garder le meilleur résultat entre l'élément de la colonne précédent auquel on peut ajouter l'élément i et l'élément de la colonne précédente sans ajouter l'élément i.

On itère cette logique pour toutes actions, et pour chaque coût entre 1 et la coût maximum autorisé.

```
# MAX_WALLET_COST une constante correspondant au coût maximum autorisé
# NUMBER_SHARES le nombre d'action
# Soit une action i, avec un coût ci et un bénéfice bi
# Soit un tableau T[NUMBER_SHARES + 1][MAX_WALLET_COST] initialisé à 0

Pour tout i de 1 à nombre_d_action
    Pour tout c de 1 à MAX_WALLET_COST
        si ci <= c
            T[i][c] = max(
                T[i][c - ci] + bi,
                T[i - 1][c]
                )
        sinon
            T[i][c] = T[i - 1][c]

Retourner T[NUMBER_SHARES][MAX_WALLET_COST - 1]
```