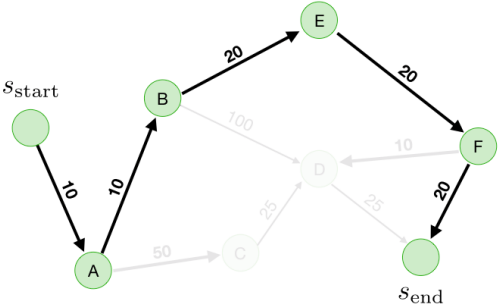


Remarque : la figure ci-dessus illustre une approche ascendante alors que la formule nous donne l'intuition d'une résolution avec une approche descendante.

□ **Types d'états** – La table ci-dessous présente la terminologie relative aux états dans le contexte de la recherche à coût uniforme :

État	Explication
Exploré \mathcal{E}	États pour lesquels le chemin optimal a déjà été trouvé
Frontière \mathcal{F}	États rencontrés mais pour lesquels on se demande toujours comment s'y rendre avec un coût minimal
Inexploré \mathcal{U}	États non rencontrés jusqu'à présent

□ **Recherche à coût uniforme** – La recherche à coût uniforme (*uniform cost search* ou *UCS* en anglais) est un algorithme de recherche qui a pour but de trouver le chemin le plus court entre les états s_{start} et s_{end} . Celui-ci explore les états s en les triant par coût croissant de $\text{PastCost}(s)$ et repose sur le fait que toutes les actions ont un coût non négatif.



Remarque 1 : UCS fonctionne de la même manière que l'algorithme de Dijkstra.

Remarque 2 : cet algorithme ne marche pas sur une configuration contenant des actions à coût négatif. Quelqu'un pourrait penser à ajouter une constante positive à tous les coûts, mais cela ne résoudrait rien puisque le problème résultant serait différent.

□ **Théorème de correction** – Lorsqu'un état s passe de la frontière \mathcal{F} à l'ensemble exploré \mathcal{E} , sa priorité est égale à $\text{PastCost}(s)$, représentant le chemin de coût minimal allant de s_{start} à s .

□ **Récapitulatif des algorithmes de parcours de graphe** – En notant N le nombre total d'états dont n sont explorés avant l'état final s_{end} , on a :

Algorithme	Acyclicité	Coûts	Temps/Espace
Programmation dynamique	oui	peu importe	$\mathcal{O}(N)$
Recherche à coût uniforme	non	$c \geq 0$	$\mathcal{O}(n \log(n))$

Remarque : ce décompte de la complexité suppose que le nombre d'actions possibles à partir de chaque état est constant.