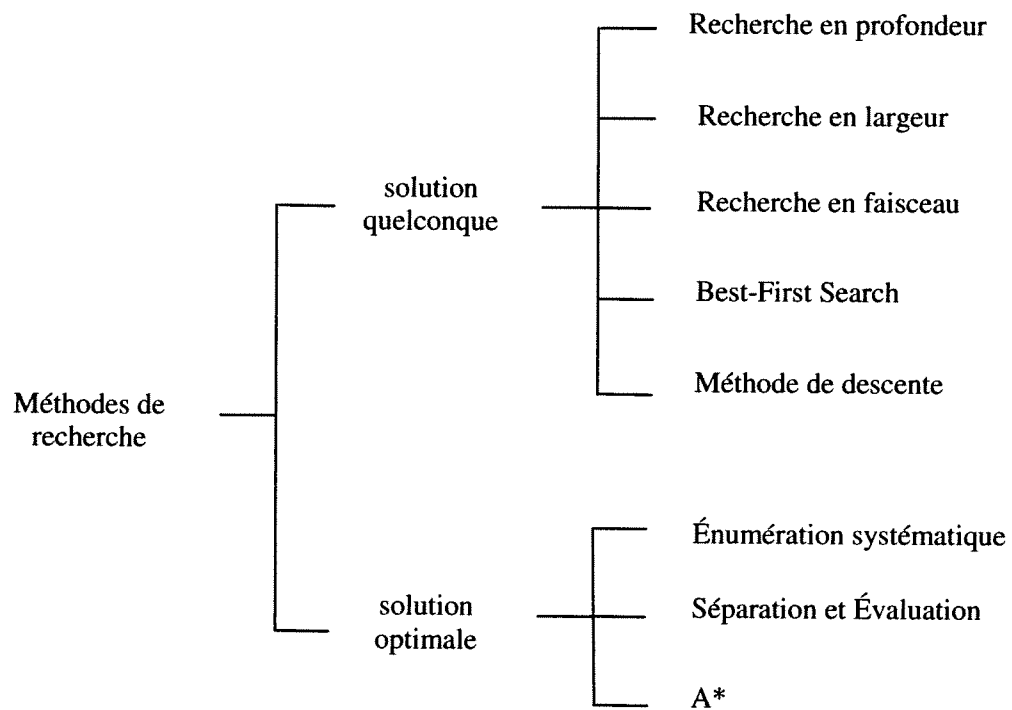
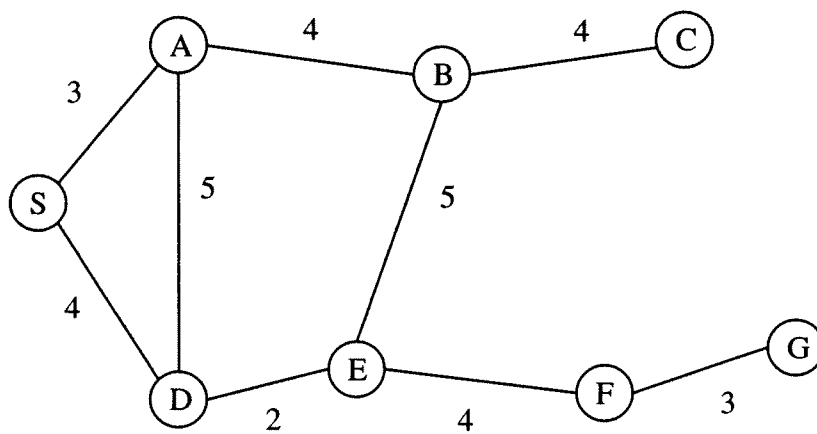


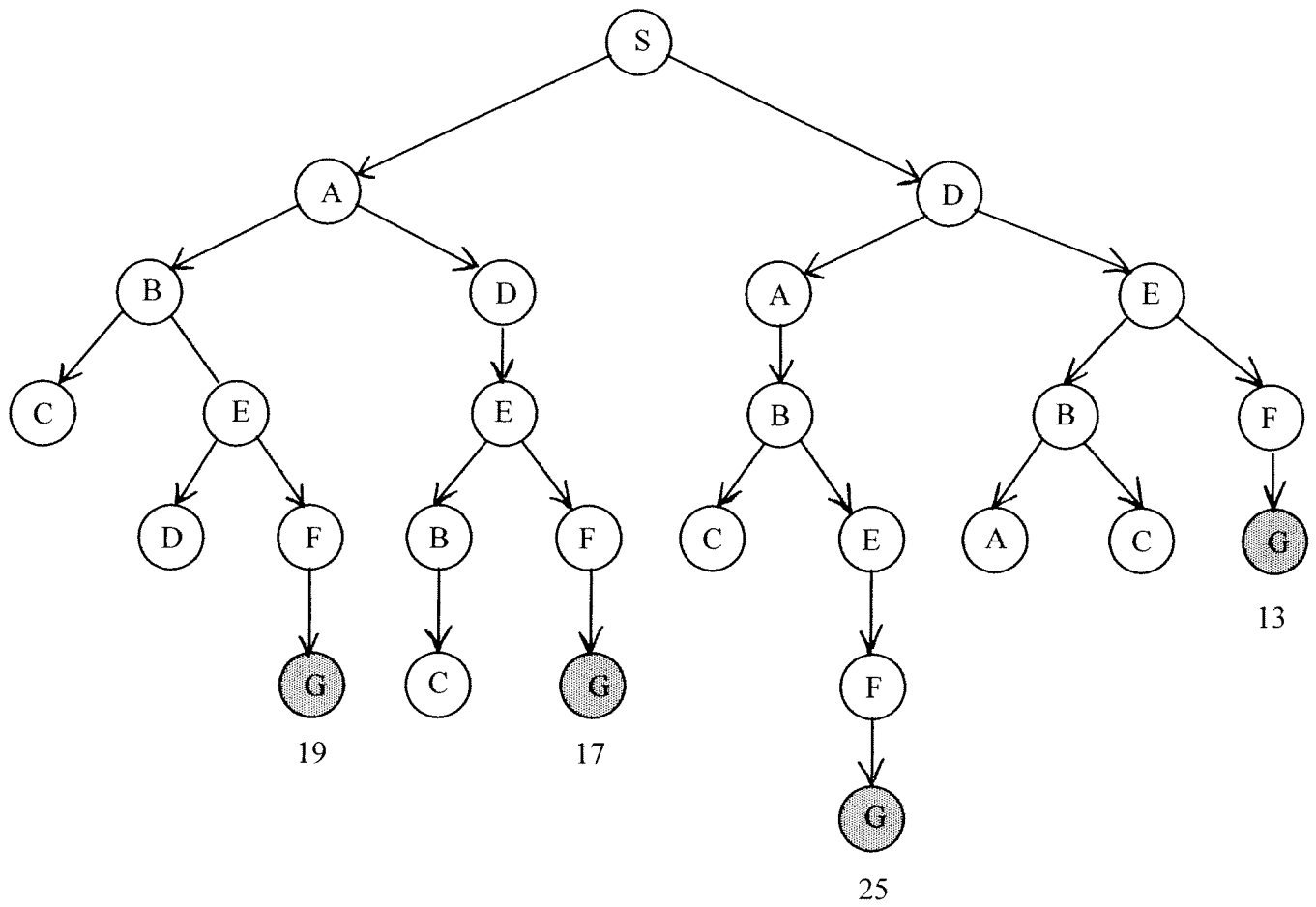
Méthodes de recherche



Problème : Identifier une chaîne élémentaire menant de S à G dans le graphe



Espace de recherche



Recherche en Profondeur (Depth-First Search)

1. $Q \leftarrow S$
2. Tant que Q n'est pas vide et qu'un noeud cible n'a pas été atteint:
Déterminer d'abord si le premier élément de Q est un noeud cible.
 - 2a. Si c'est le cas, on ne fait rien
 - 2b. Sinon, enlever cet élément de Q et ajouter ses enfants dans l'ordre gauche-droite au début de Q.
3. Si un noeud cible a été atteint alors Succès; sinon Échec

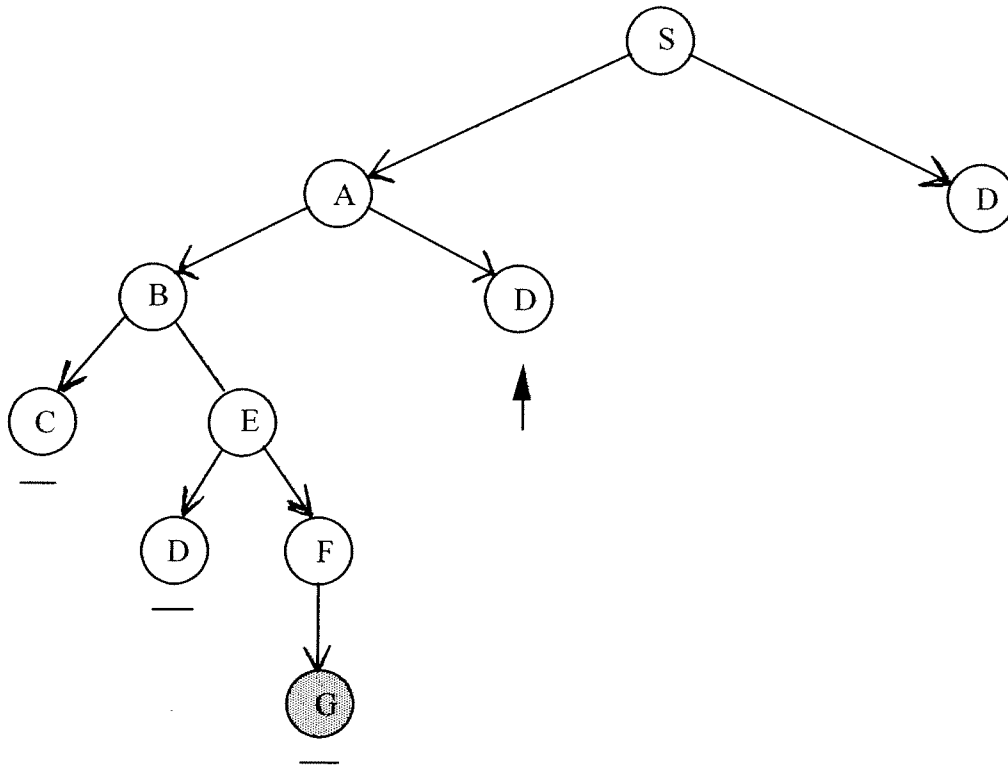
Évolution de Q

- (a) S
- (b) AD
- (c) BDD
- (d) CEDD
- (e)

Remarques

- Recherche aveugle, révocable
- Q est une pile
- Il y a un danger de s'enfoncer profondément dans un sous-arbre touffu qui ne contient pas de solutions (e.g. si C donne accès à un sous-réseau ne contenant pas de noeuds cibles)
- En pratique: on fixe une limite sur la profondeur de la recherche

Recherche en Profondeur (Depth-First Search)



Recherche en Profondeur (Depth-First Search)

Remarques

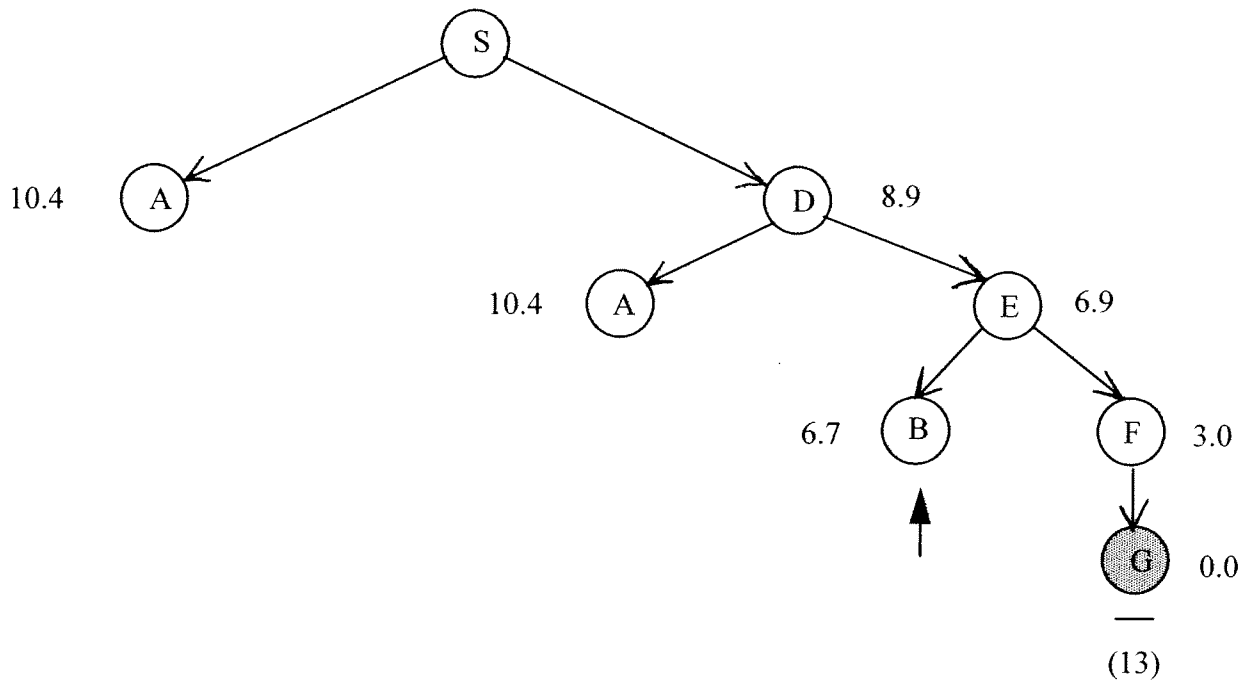
- Amélioration heuristique: ordonner les enfants non pas dans un ordre systématique gauche-droite, mais en fonction des connaissances du problème, de façon à explorer les alternatives les plus intéressantes en premier.

Exemple

1. $Q \leftarrow S$
2. Tant que Q n'est pas vide et qu'un noeud cible n'a pas été atteint:
Déterminer d'abord si le premier élément de Q est un noeud cible.
 - 2a. Si c'est le cas, on ne fait rien
 - 2b. Sinon, enlever cet élément de Q, trier les enfants en fonction de la distance estimée restant à parcourir, et ajouter les enfants dans cet ordre au début de Q
3. Si un noeud cible a été atteint alors Succès; sinon Échec

N.B. la distance restant à parcourir peut être estimée par la distance euclidienne ou distance à vol d'oiseau entre le noeud courant et le noeud cible.

Recherche en Profondeur (avec amélioration heuristique)



Recherche en Largeur (Breadth-First Search)

1. $Q \leftarrow S$
2. Tant que Q n'est pas vide et qu'un noeud cible n'a pas été atteint:
Déterminer d'abord si le premier élément de Q est un noeud cible.
 - 2a. Si c'est le cas, on ne fait rien
 - 2b. Sinon, enlever cet élément de Q et ajouter ses enfants dans l'ordre gauche-droite à la fin de Q.
3. Si un noeud cible a été atteint alors Succès; sinon Échec

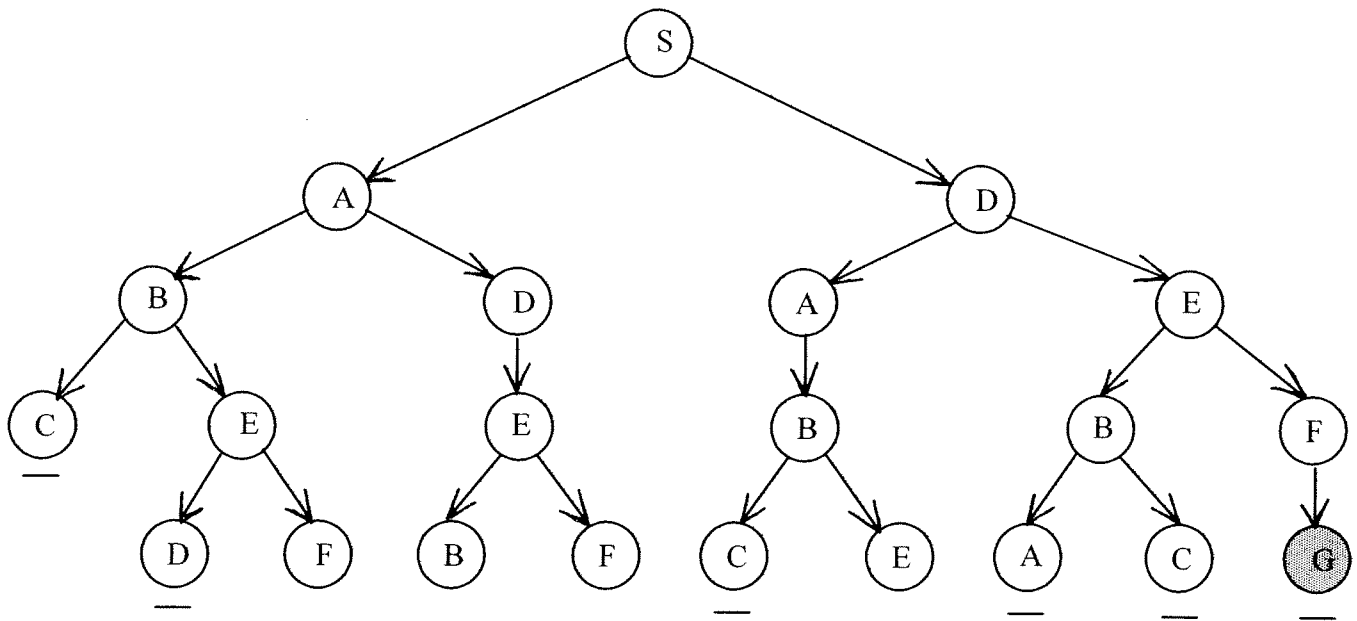
Évolution de Q

- (a) S
- (b) AD
- (c) DBD
- (d) BDAE
- (e)

Remarques

- Recherche aveugle, irrévocable
- Q est une queue
- Peut être coûteux si toutes les solutions admissibles sont à peu près au même niveau (comme dans notre exemple)
- Si le coût sur toutes les branches est le même, alors la recherche en largeur trouve la solution optimale.

Recherche en Largeur (Breadth-First Search)



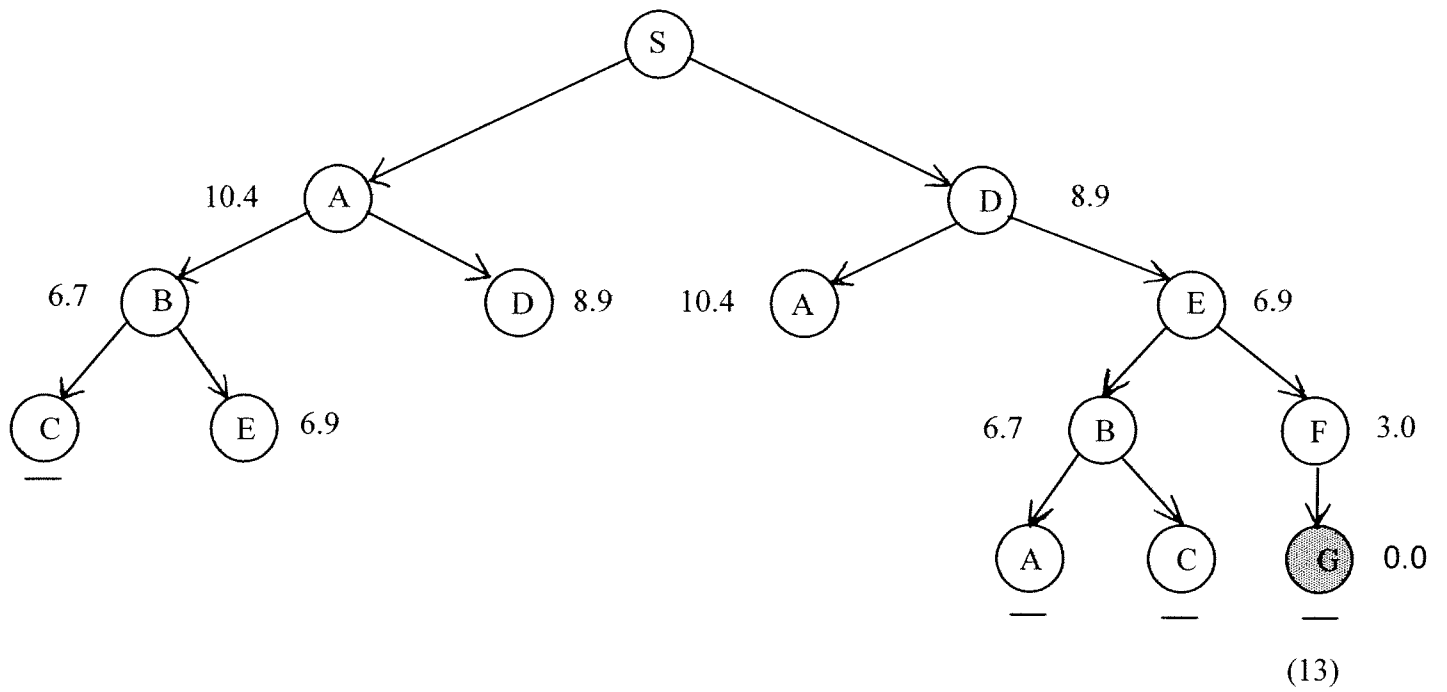
(13)

Recherche en Largeur (Breadth-First Search)

Remarques

- Recherche en faisceau: amélioration heuristique où on sélectionne à chaque niveau un nombre fixe de noeuds à partir desquels on poursuit la recherche.
 - réduit le nombre de noeuds à explorer
 - recherche non systématique: il est donc possible de passer "à côté" d'une solution admissible.
 - compromis entre le temps d'exécution et la qualité de la solution finale: largeur du faisceau

Recherche en Faisceau (Beam Search)



Largeur du Faisceau = 2

Heuristique: distance estimée restant à parcourir

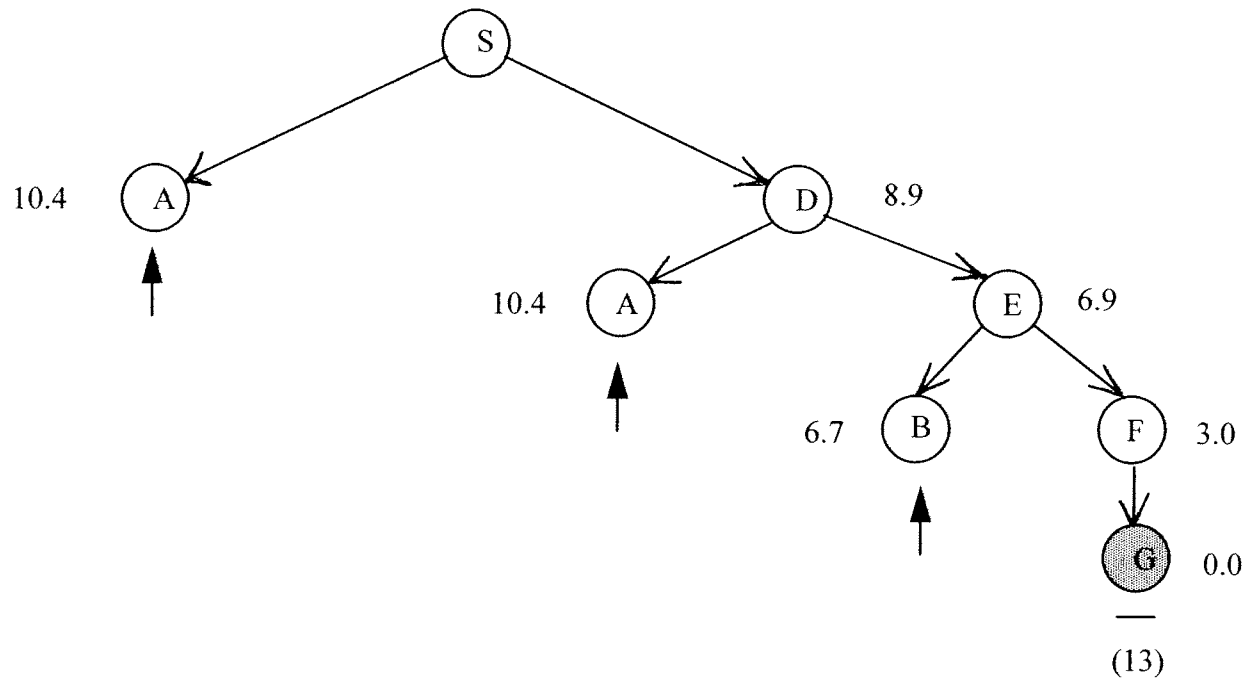
Best-First Search

1. $Q \leftarrow S$
2. Tant que Q n'est pas vide et qu'un noeud cible n'a pas été atteint:
Déterminer d'abord si le premier élément de Q est un noeud cible.
 - 2a. Si c'est le cas, on ne fait rien
 - 2b. Sinon, enlever cet élément de Q, ajouter ses enfants à Q, et trier Q (au complet) en fonction de la distance estimée restant à parcourir.
3. Si un noeud cible a été atteint alors Succès; sinon Échec.

Remarques

- Recherche informée
- Plus efficace que les recherches aveugles

Best-First Search



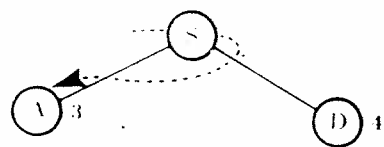
Énumération Systématique (British Museum)

- Énumérer toutes les solutions possibles
- Modification de la recherche en longueur ou en profondeur:
- On n'arrête pas dès qu'on a trouvé une solution, mais on poursuit jusqu'à ce que tout l'espace de recherche ait été exploré
- Pas applicable lorsque l'espace de recherche est étendu

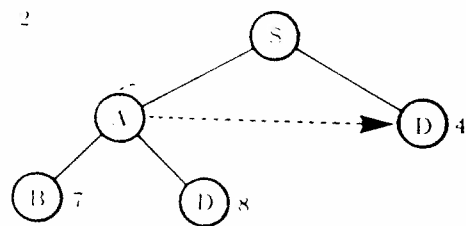
Séparation et Évaluation (Branch-and-Bound)

- Recherche informée qui permet d'obtenir une solution optimale sans une énumération explicite de toutes les solutions possibles (sauf en pire cas).
- Ressemble à Best-First Search.
- Mesure d'évaluation heuristique: distance parcourue jusqu'à présent.
- On retient toujours la valeur de la meilleure solution admissible rencontrée jusqu'à présent, afin d'éliminer des portions de l'espace de recherche qui ne peuvent contenir de solutions optimales.

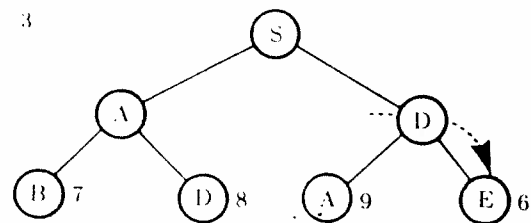
1



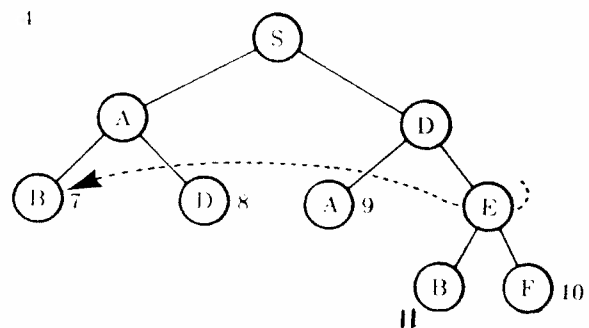
2



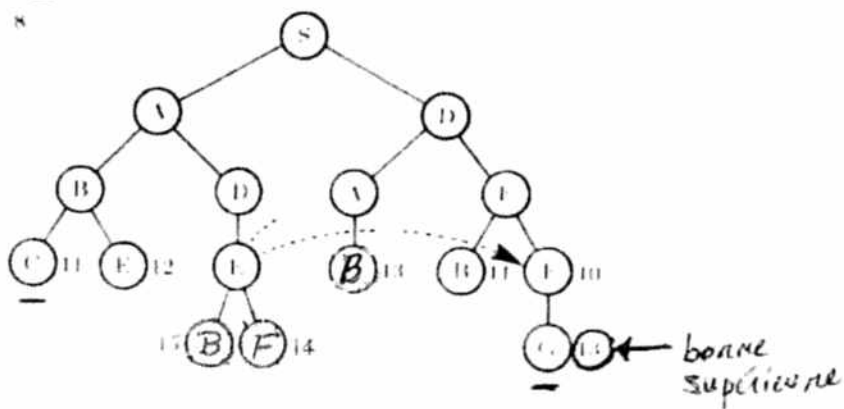
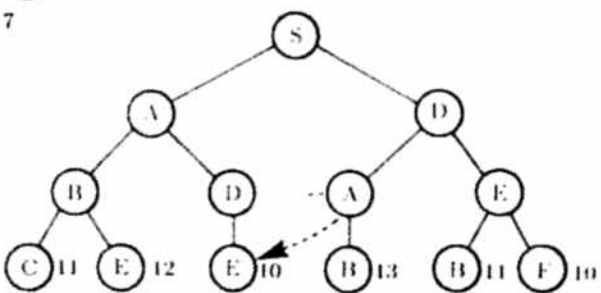
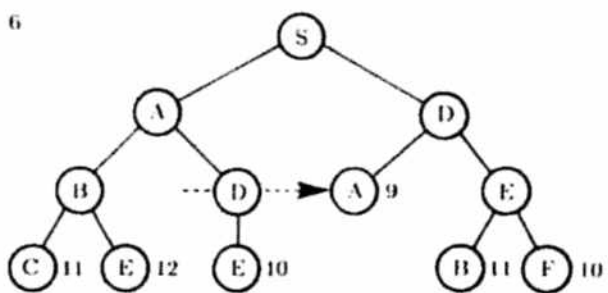
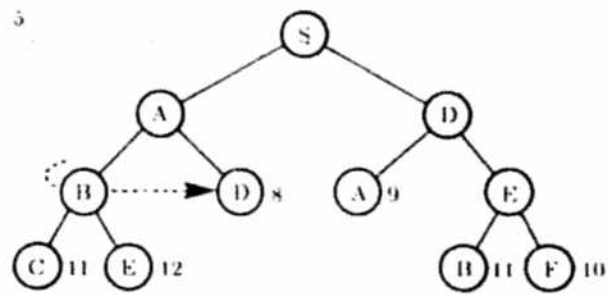
3



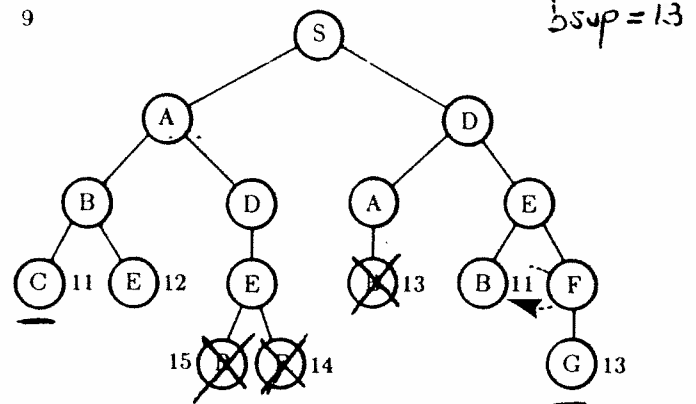
4



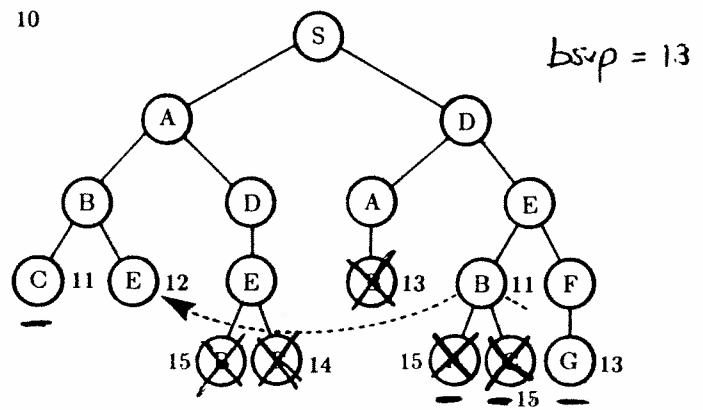
11



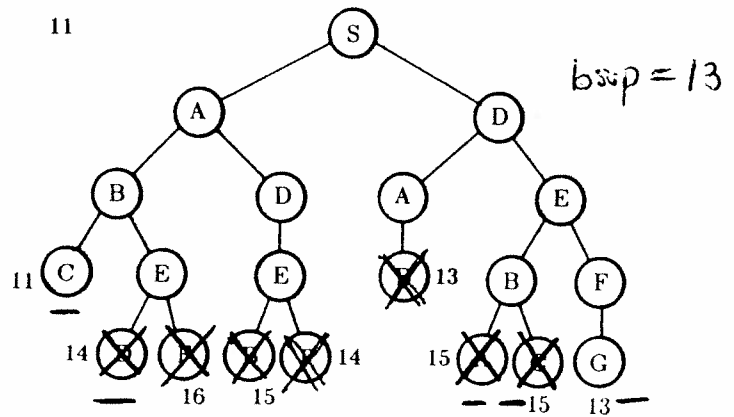
9



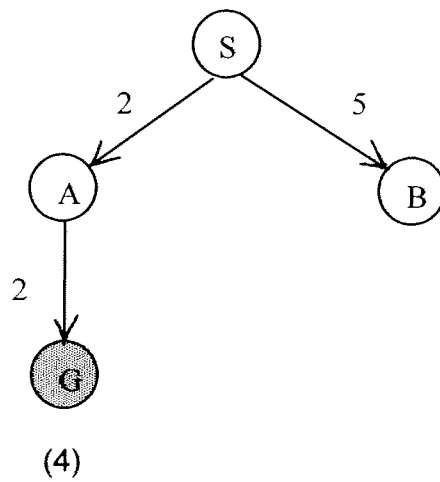
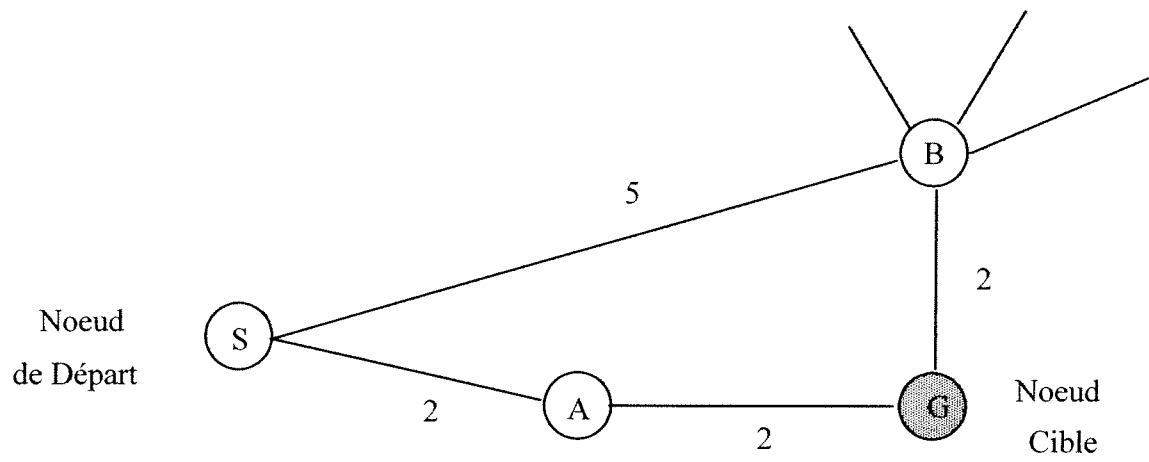
10



11

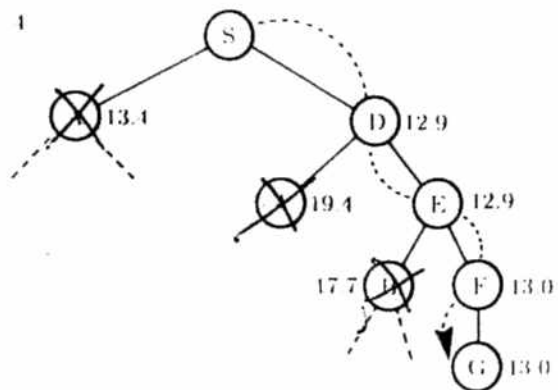
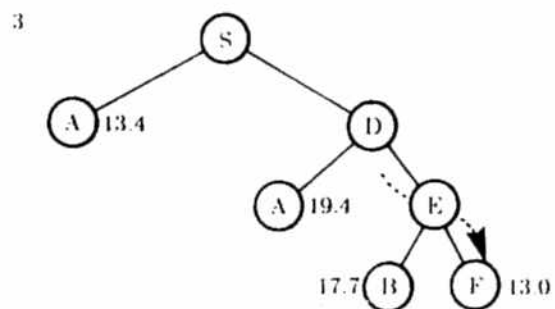
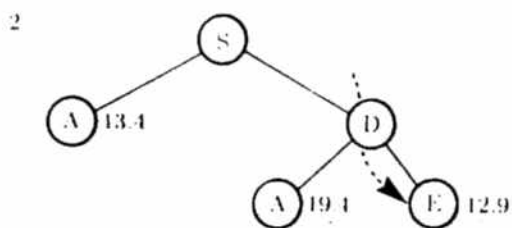
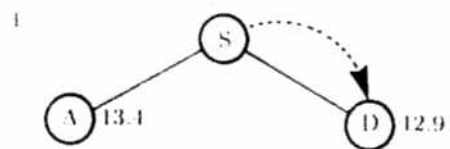


Séparation et Évaluation (Branch-and-Bound)



A*

- Recherche informée qui améliore Séparation et Évaluation.
- On ajoute à la distance accumulée un estimé de la distance à parcourir (de telle sorte que la distance obtenue constitue une borne inférieure de la distance réelle)
- L'ajout de l'estimé permet d'élaguer plus rapidement certaines branches de l'arbre de recherche, sans compromettre l'optimalité de la méthode.
- Séparation et Évaluation est un cas spécial de A*, où l'estimé de la distance à parcourir est fixé à 0.



Espace de recherche

- On considère l'espace de recherche suivant:
(plutôt qu'un espace de recherche de type "constructif", tel que celui qui nous a permis de trouver un chemin dans un graphe)
 - Noeud: solution admissible quelconque
 - Arc: opérateur (transformation) qu'on peut appliquer à une solution admissible afin d'obtenir une nouvelle solution admissible

Méthode de descente (Hill-climbing)

- Recherche informée, irrévocable.
- Semblable à la recherche en profondeur avec ordonnancement heuristique des enfants (sauf que dans ce dernier cas, la recherche est révocable).

1. solution courante ← solution initiale

2. Répéter :

Générer un ensemble de solutions voisines de la solution courante (à l'aide des opérateurs disponibles)

2a. Si la meilleure solution dans ce voisinage est meilleure que la solution courante, elle devient la nouvelle solution courante

2b. Sinon on arrête

3. Retourner la solution courante