Graphes d’états (suite et fin)

Cyril Terrioux

Laboratoire des Sciences de l’Information et des Systèmes

LSIS - UMR CNRS 6168

Plan

1. Méthodes de résolution arborescentes (suite et fin)
2. Méthodes incomplètes
3. Quelle méthode choisir ?

Plan

1. Méthodes de résolution arborescentes (suite et fin)
2. Méthodes incomplète
3. Quelle méthode choisir ?

Recherche à divergence limitée

Hypothèse : on dispose d'une bonne heuristique

Recherche à divergence limitée

Hypothèse : on dispose d'une bonne heuristique

Divergence : un choix de successeur ne respectant pas l'heuristique

Recherche à divergence limitée

Hypothèse : on dispose d'une bonne heuristique

Divergence : un choix de successeur ne respectant pas l'heuristique

Principe :

- ne générer que les états qui ont moins de i divergences

Recherche à divergence limitée

Hypothèse : on dispose d'une bonne heuristique

Divergence : un choix de successeur ne respectant pas l'heuristique

Principe :

- ne générer que les états qui ont moins de i divergences
- lancer itérativement plusieurs recherches avec un i croissant

Recherche à divergence limitée

Hypothèse : on dispose d'une bonne heuristique

Divergence : un choix de successeur ne respectant pas l'heuristique

Principe :

- ne générer que les états qui ont moins de i divergences
- lancer itérativement plusieurs recherches avec un i croissant

Idée : les états avec peu de divergences sont les plus prometteurs

Recherche à divergence limitée

Avantage :

- coût en mémoire limité

Recherche à divergence limitée

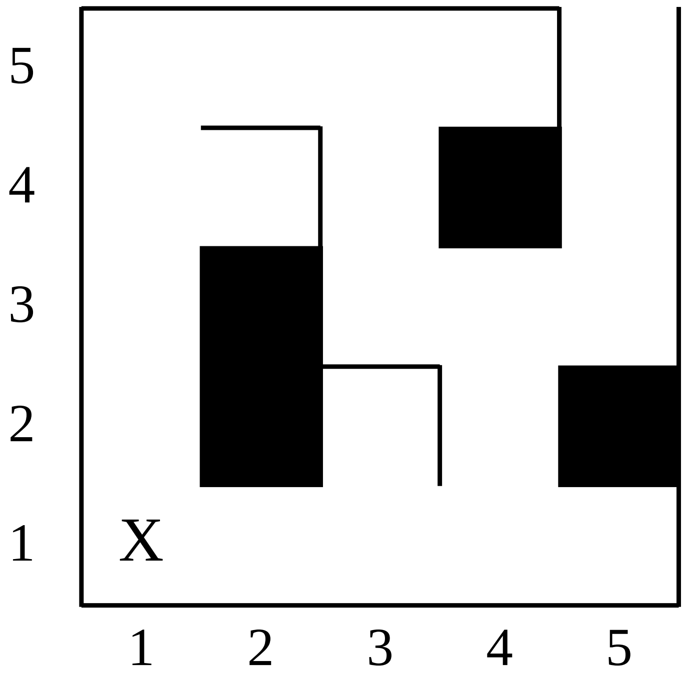
Avantage :

- coût en mémoire limité

Inconvénients :

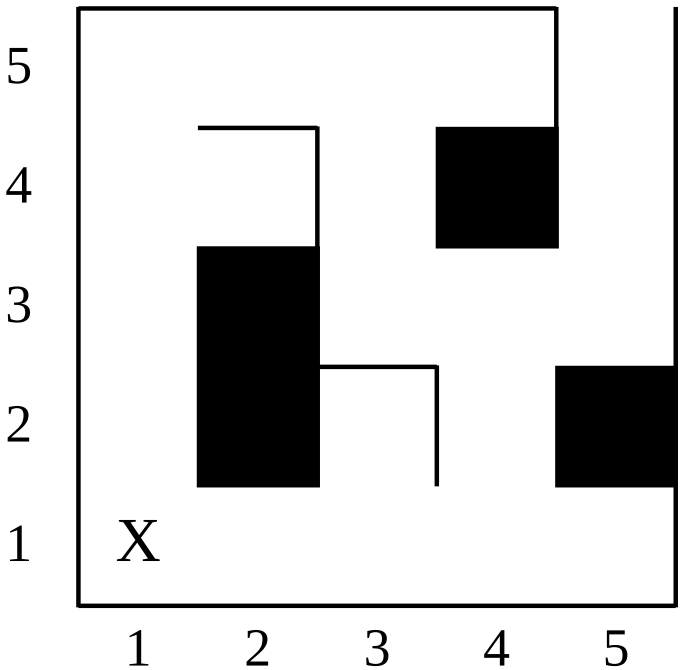
- risque de parcours intégral de l'espace de recherche
- des redondances : un état peut être examiné plusieurs fois

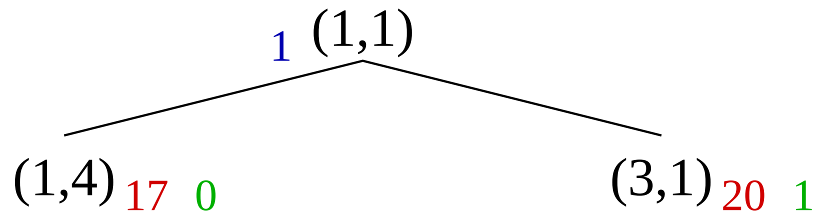
Exemple $i = 1$



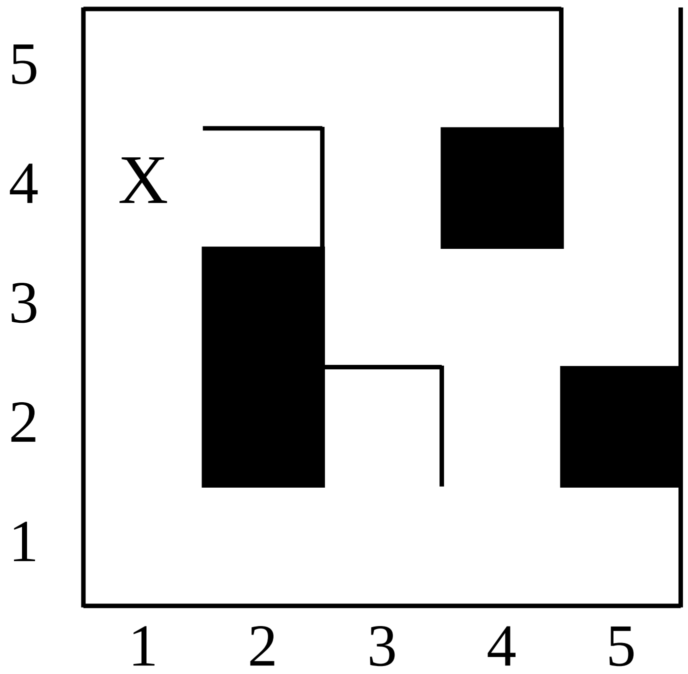
1 (1,1)

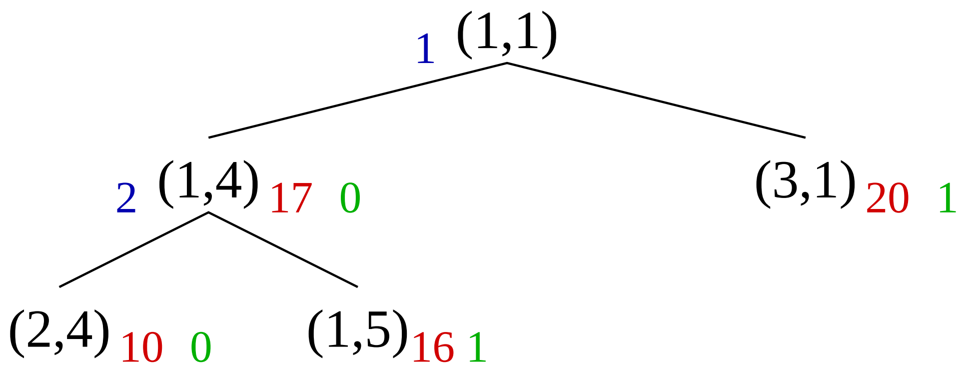
Exemple $i = 1$



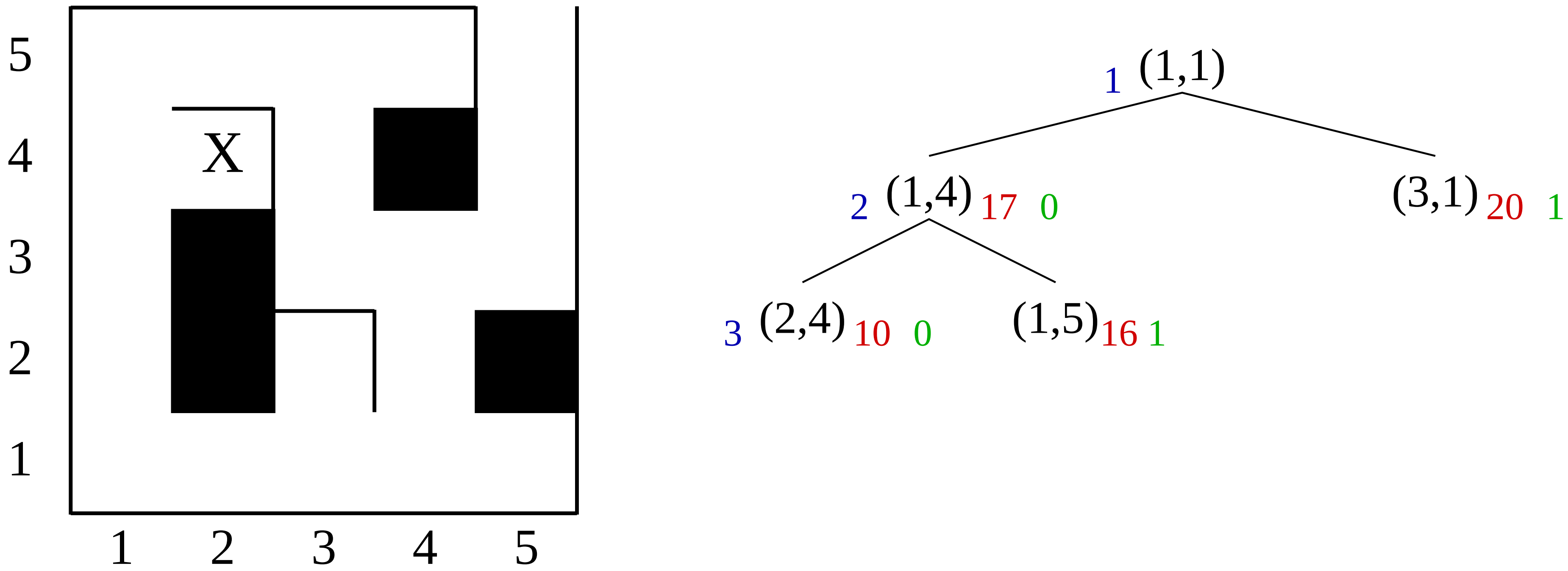


Exemple $i = 1$

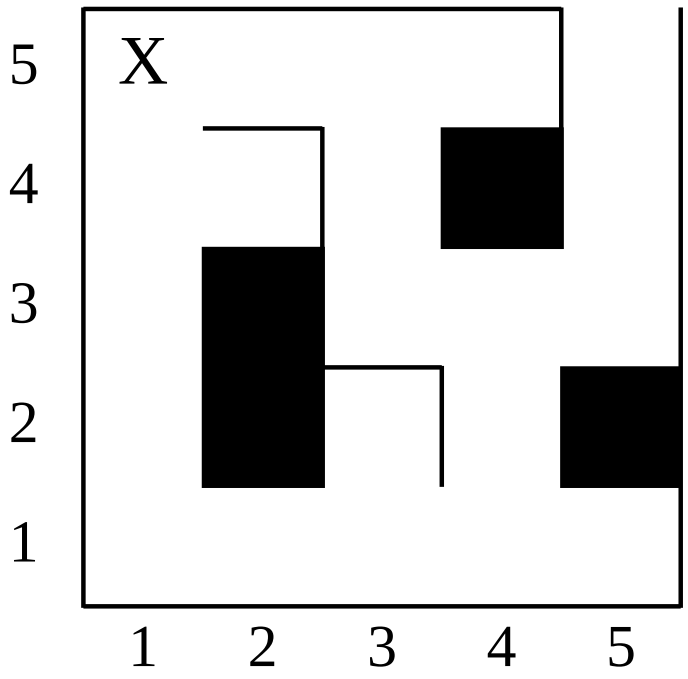


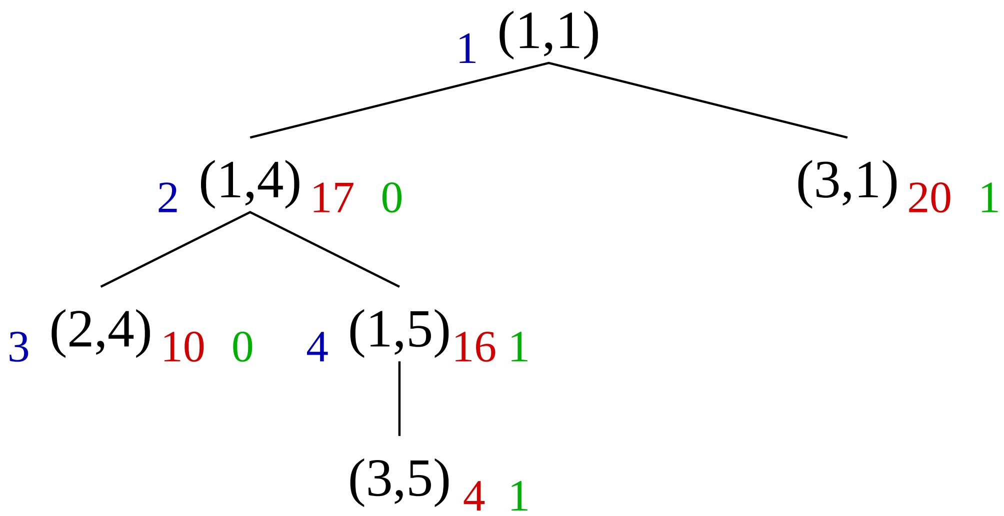


Exemple $i = 1$

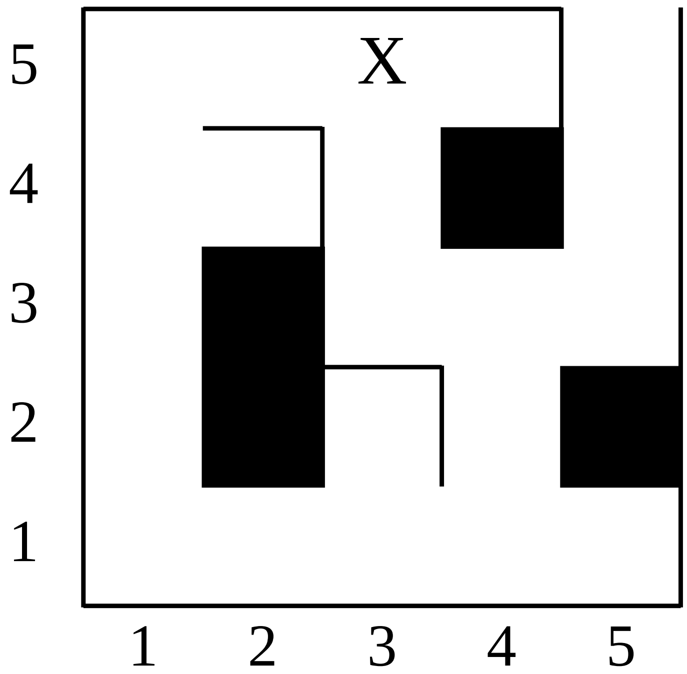


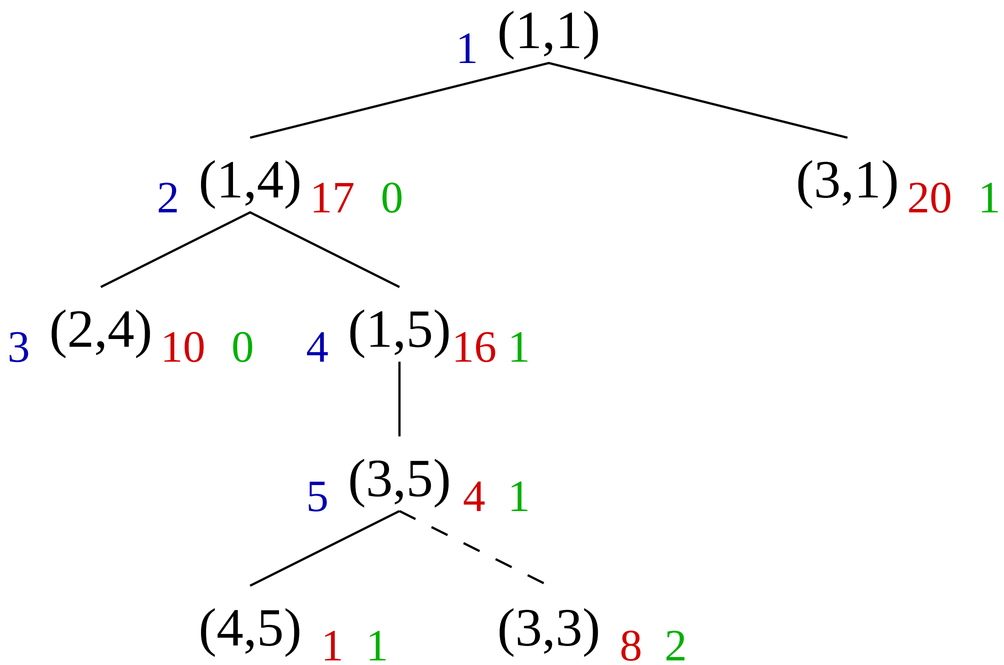
Exemple $i = 1$



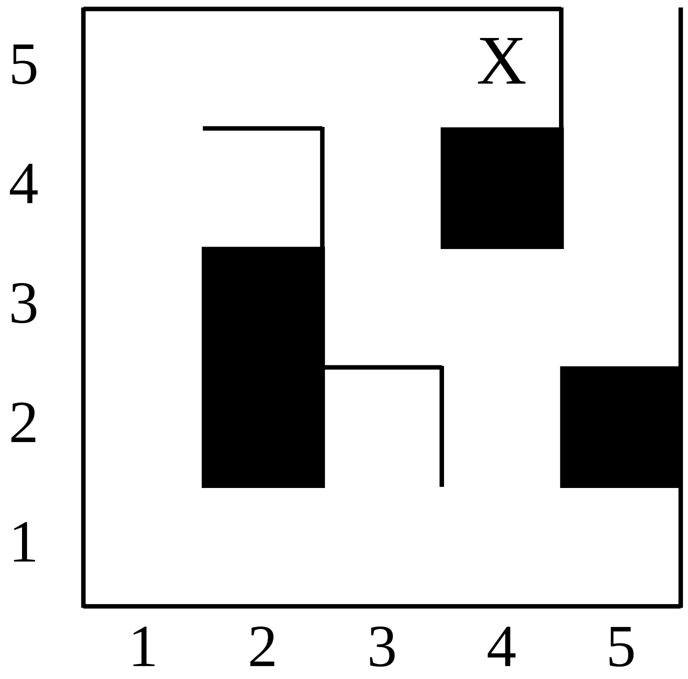


Exemple $i = 1$

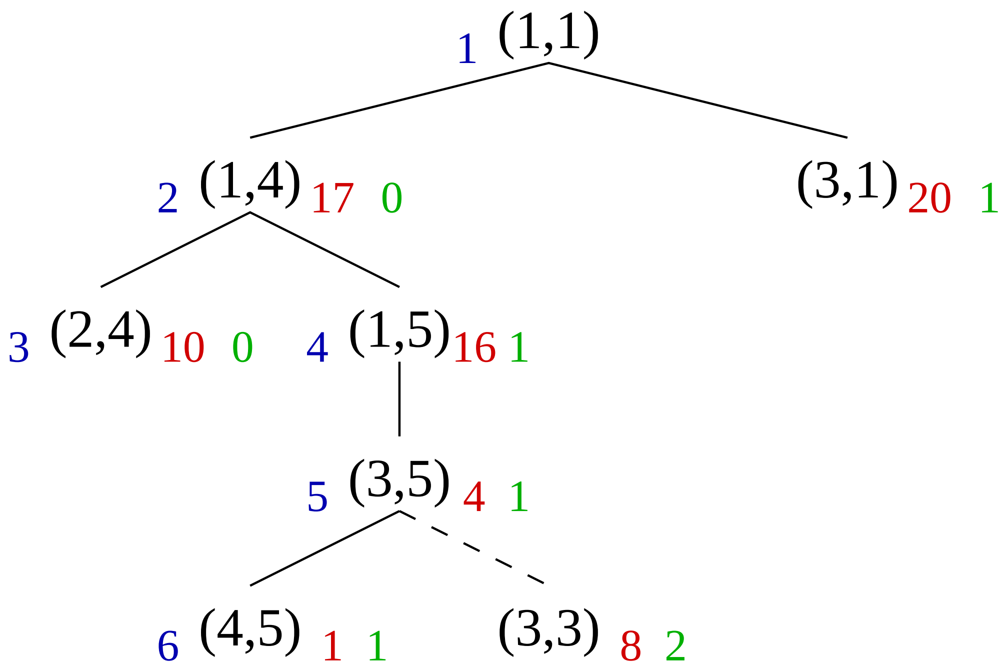

X
5
4
3
2
1
1 2 3 4 5


1 (1,1)
2 (1,4) 17 0
(3,1) 20 1
3 (2,4) 10 0
4 (1,5) 16 1
5 (3,5) 4 1
(4,5) 1 1
(3,3) 8 2

Exemple $i = 1$

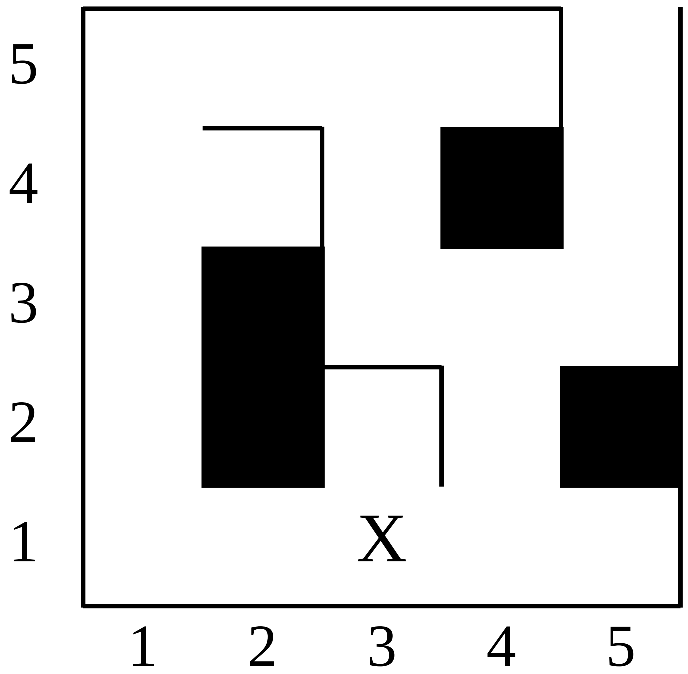


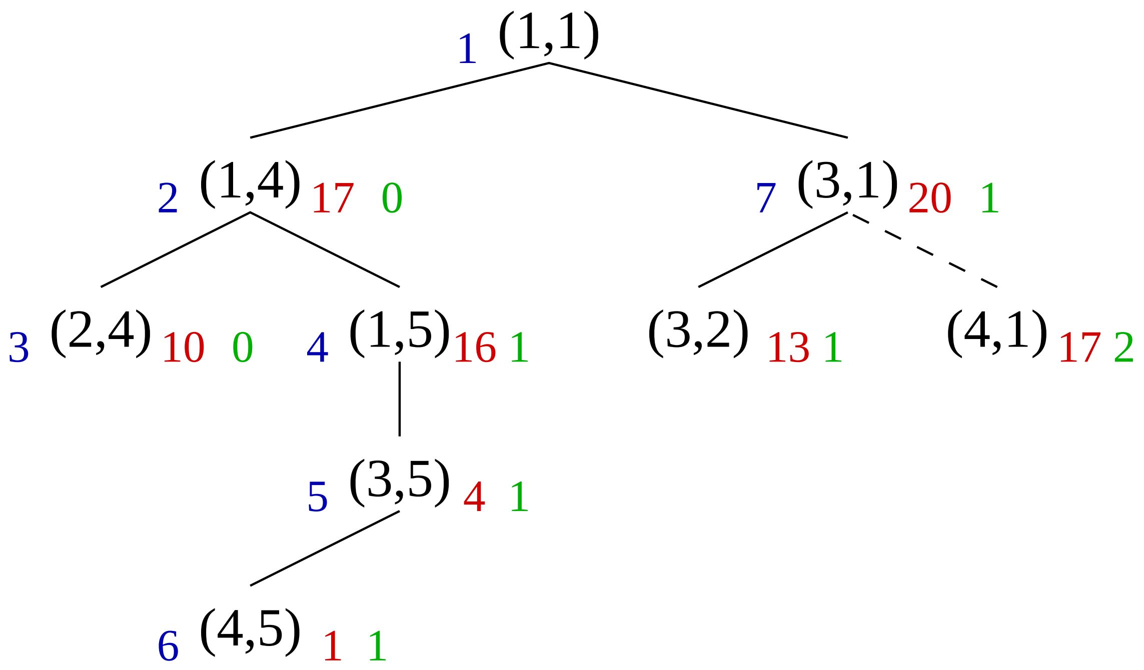
X
5
4
3
2
1
1 2 3 4 5



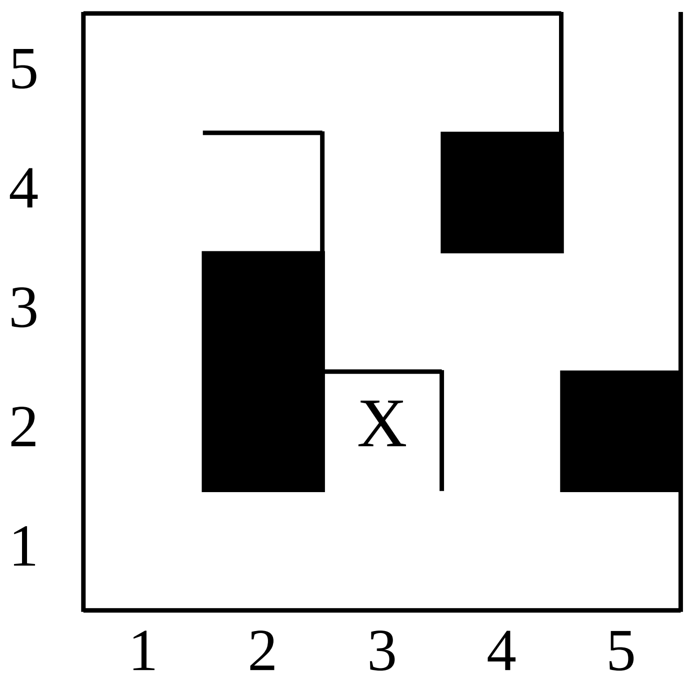
1 (1,1)
2 (1,4) 17 0
(3,1) 20 1
3 (2,4) 10 0
4 (1,5) 16 1
5 (3,5) 4 1
6 (4,5) 1 1
(3,3) 8 2

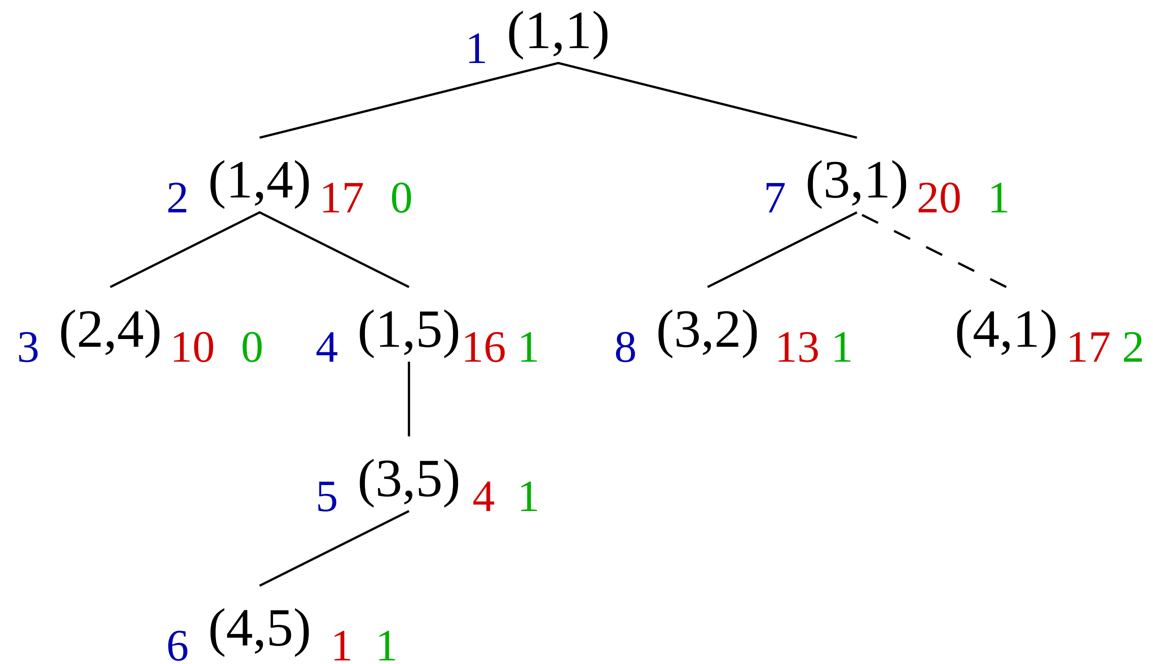
Exemple $i = 1$


5
4
3
2
1
X
1 2 3 4 5


1 (1,1)
2 (1,4) 17 0
7 (3,1) 20 1
3 (2,4) 10 0
4 (1,5) 16 1
(3,2) 13 1
(4,1) 17 2
5 (3,5) 4 1
6 (4,5) 1 1

Exemple $i = 1$


5
4
3
2
1
X
1 2 3 4 5


1 (1,1)
2 (1,4) 17 0
7 (3,1) 20 1
3 (2,4) 10 0
4 (1,5) 16 1
8 (3,2) 13 1
(4,1) 17 2
5 (3,5) 4 1
6 (4,5) 1 1

L’algorithme A*

Une variante de la recherche le meilleur d’abord

L'algorithme A*

Une variante de la recherche le meilleur d'abord

Objectif : calculer le plus court chemin de l'état initial à un état final

L'algorithme A*

Une variante de la recherche le meilleur d'abord

Objectif : calculer le plus court chemin de l'état initial à un état final

$\Rightarrow$ un algorithme voisin de l'algorithme de Dijkstra

L'algorithme A*

$Fermé \leftarrow \emptyset$, $Ouvert \leftarrow \{(e_{init}, h(e_{init}))\}$

TantQue $meilleur(Ouvert)$ n'est pas un but **Et** $Ouvert \neq \emptyset$

    $e \leftarrow meilleur(Ouvert)$, $Ouvert \leftarrow Ouvert - \{e\}$

    Pour chaque état e_i voisin de e

        Si $e_i \notin Fermé$ **Et** $(e_i, f_i) \notin Ouvert$

        Alors $Ouvert \leftarrow Ouvert \cup \{(e_i, f(e_i))\}$

        Si $(e_i, f_i) \in Ouvert$ **Et** $f(e_i) < f_i$

        Alors $f_i \leftarrow f(e_i)$

    $Fermé \leftarrow Fermé \cup \{e\}$

Si $Ouvert = \emptyset$

Alors il n'existe pas de but accessible

Sinon l'élément $meilleur(Ouvert)$ est un but

La fonction de coût f

$$f(e) = g(e) + h(e)$$

La fonction de coût f

$$f(e) = g(e) + h(e)$$

$g(e)$ = le coût du chemin de e_{init} à e

La fonction de coût f

$$f(e) = g(e) + h(e)$$

$g(e)$ = le coût du chemin de e_{init} à e

$h(e)$ = une évaluation heuristique du coût du plus court chemin de e à un état final

La fonction de coût f

$$f(e) = g(e) + h(e)$$

$g(e)$ = le coût du chemin de e_{init} à e

$h(e)$ = une évaluation heuristique du coût du plus court chemin de e à un état final

Choix du prochain état à étudier :

- l'état de $Ouvert$ qui possède la plus petite valeur pour f

La fonction de coût f

$$f(e) = g(e) + h(e)$$

$g(e)$ = le coût du chemin de e_{init} à e

$h(e)$ = une évaluation heuristique du coût du plus court chemin de e à un état final

Choix du prochain état à étudier :

- l'état de $Ouvert$ qui possède la plus petite valeur pour f
- en cas d'égalité, on choisit celui qui a la plus petite valeur pour g

La fonction heuristique h

On souhaite estimer :

$$h^*(e) = \min_{[e, e_{final}]} c(e, e_1) + c(e_1, e_2) + \ldots + c(e_i, e_{final})$$

La fonction heuristique h

heuristique parfaite : $\forall e, e', h(e) = h(e') \Leftrightarrow h^*(e) = h^*(e')$

La fonction heuristique h

heuristique parfaite : $\forall e, e', h(e) = h(e') \Leftrightarrow h^*(e) = h^*(e')$

$\Rightarrow$ convergence directe vers un état final

La fonction heuristique h

heuristique parfaite : $\forall e, e', h(e) = h(e') \Leftrightarrow h^*(e) = h^*(e')$

$\Rightarrow$ convergence directe vers un état final

heuristique monotone :

$$\forall e, e' \ successeur\ de\ e, h(e) - h(e') \leq c(e, e')$$

La fonction heuristique h

heuristique parfaite : $\forall e, e', h(e) = h(e') \Leftrightarrow h^*(e) = h^*(e')$

$\Rightarrow$ convergence directe vers un état final

heuristique monotone :

$$\forall e, e' \; successeur \; de \; e, h(e) - h(e') \leq c(e, e')$$

$\Rightarrow$ pour tout état e de $Fermé$, le chemin de e_{init} à e est optimal

La fonction heuristique h

heuristique parfaite : $\forall e, e', h(e) = h(e') \Leftrightarrow h^*(e) = h^*(e')$

$\Rightarrow$ convergence directe vers un état final

heuristique monotone :

$$\forall e, e' \ successeur \ de \ e, h(e) - h(e') \leq c(e, e')$$

$\Rightarrow$ pour tout état e de $Fermé$, le chemin de e_{init} à e est optimal

heuristique minorante : $\forall e, h(e) \leq h^*(e)$

La fonction heuristique h

heuristique parfaite : $\forall e, e', h(e) = h(e') \Leftrightarrow h^*(e) = h^*(e')$

$\Rightarrow$ convergence directe vers un état final

heuristique monotone :

$$\forall e, e' \ successeur \ de \ e, h(e) - h(e') \leq c(e, e')$$

$\Rightarrow$ pour tout état e de $Ferm\acute{e}$, le chemin de e_{init} à e est optimal

heuristique minorante : $\forall e, h(e) \leq h^*(e)$

$\Rightarrow$ le chemin trouvé est optimal

Complexité en temps de l'algorithme A*

- dans le pire des cas : $O(2^N)$ (N = Nombre d'états)

Complexité en temps de l'algorithme A*

- dans le pire des cas : $O(2^N)$ (N = Nombre d'états)
- si l'heuristique est minorante : $O(N^2)$

Complexité en temps de l'algorithme A*

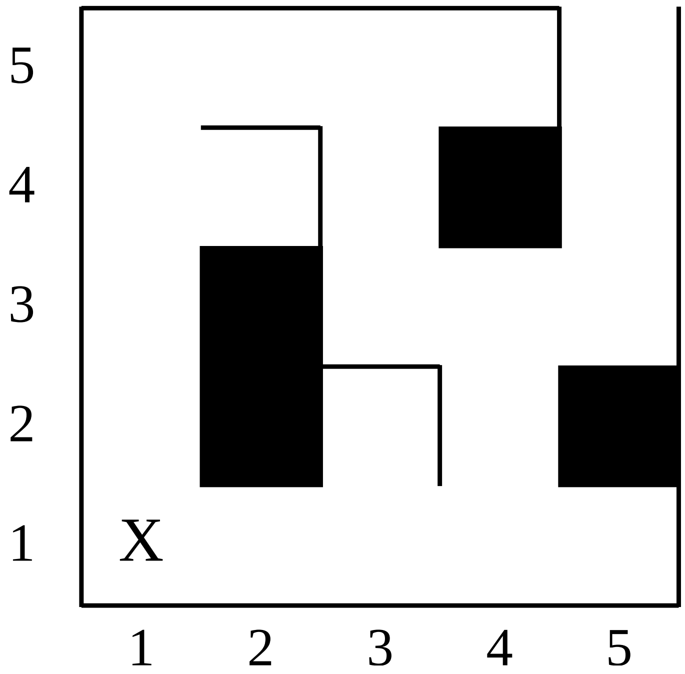
- dans le pire des cas : $O(2^N)$ (N = Nombre d'états)
- si l'heuristique est minorante : $O(N^2)$
- si l'heuristique est monotone : $O(N)$

Complexité en temps de l'algorithme A*

- dans le pire des cas : $O(2^N)$ (N = Nombre d'états)
- si l'heuristique est minorante : $O(N^2)$
- si l'heuristique est monotone : $O(N)$

Attention le nombre d'états peut être exponentiel !!!

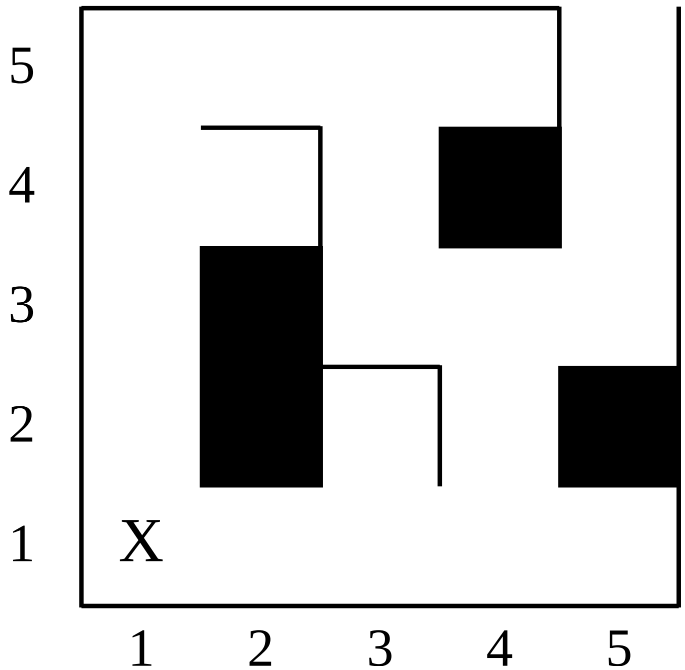
Exemple

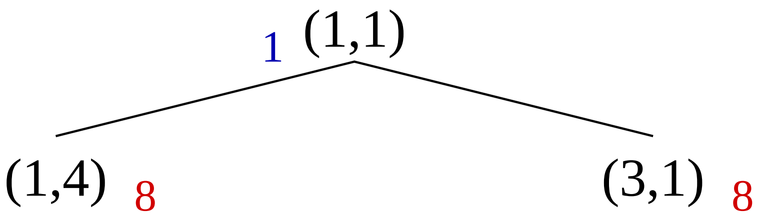


1 (1,1)

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

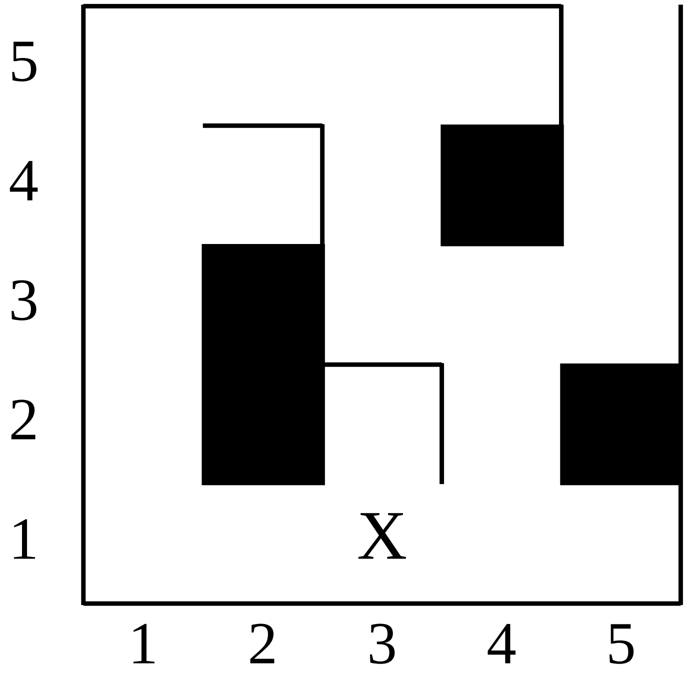
Exemple

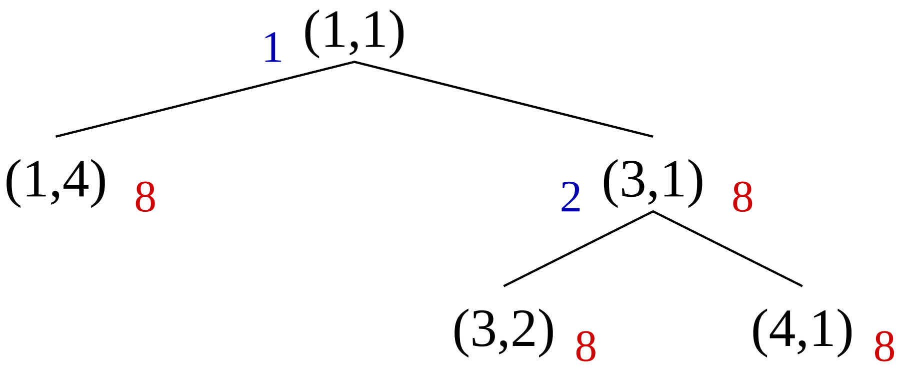




Distance de Manhattan : $h((x, y)) = |5 - x| + |5 - y|$

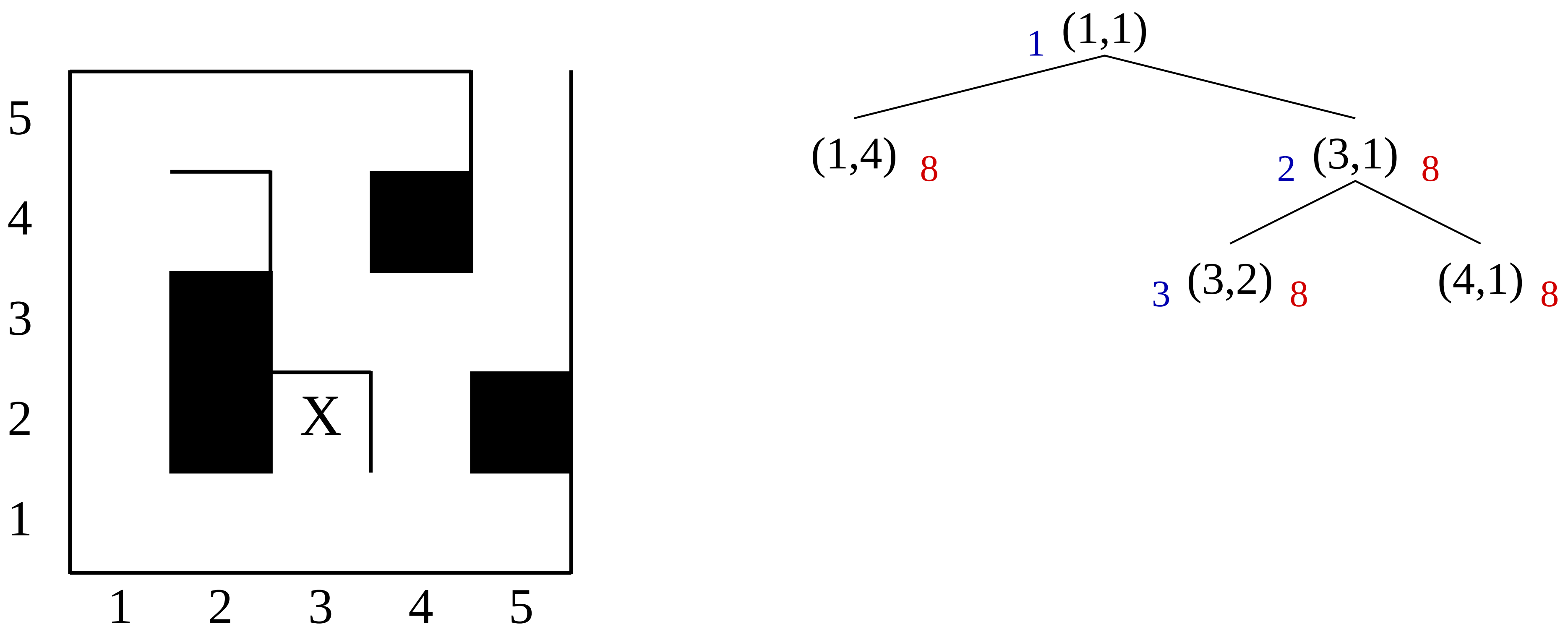
Exemple





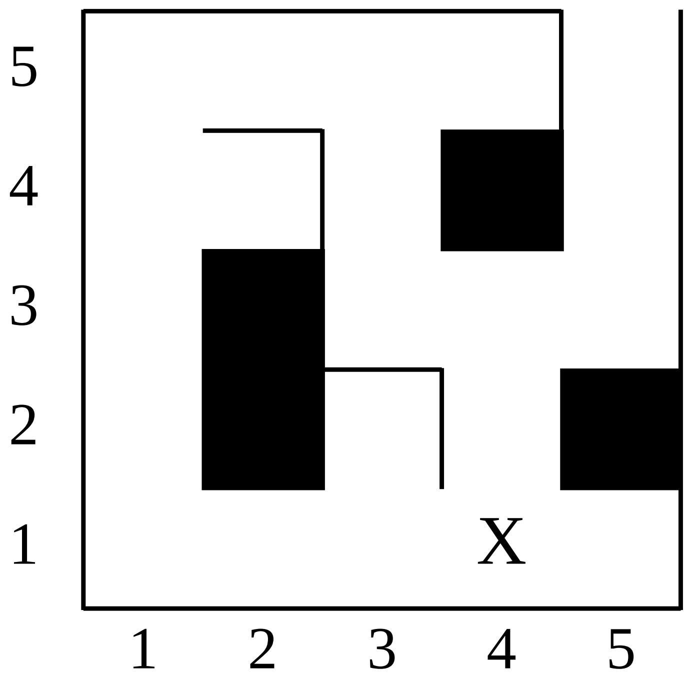
Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

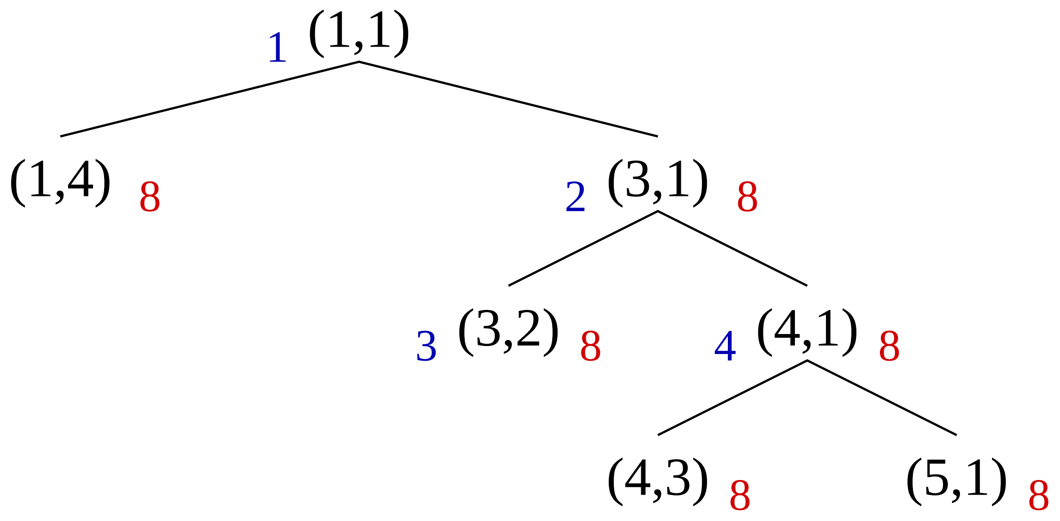
Exemple



Distance de Manhattan : $h((x, y)) = |5 - x| + |5 - y|$

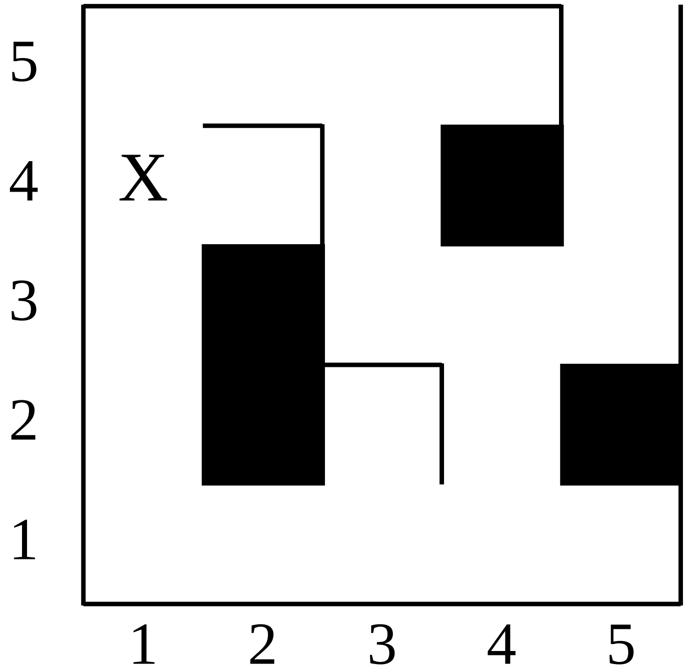
Exemple

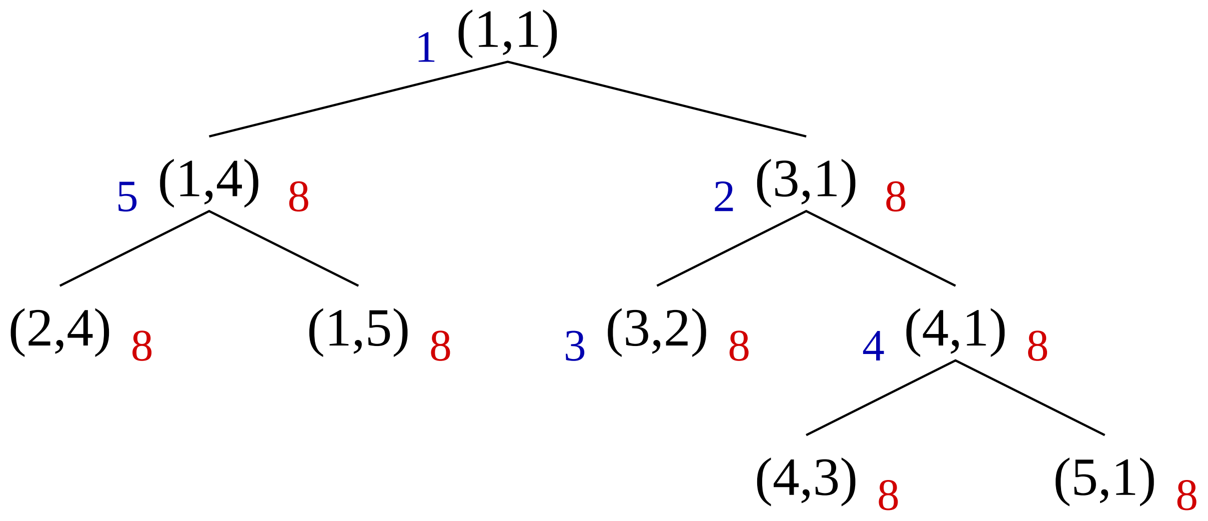




Distance de Manhattan : $h((x, y)) = |5 - x| + |5 - y|$

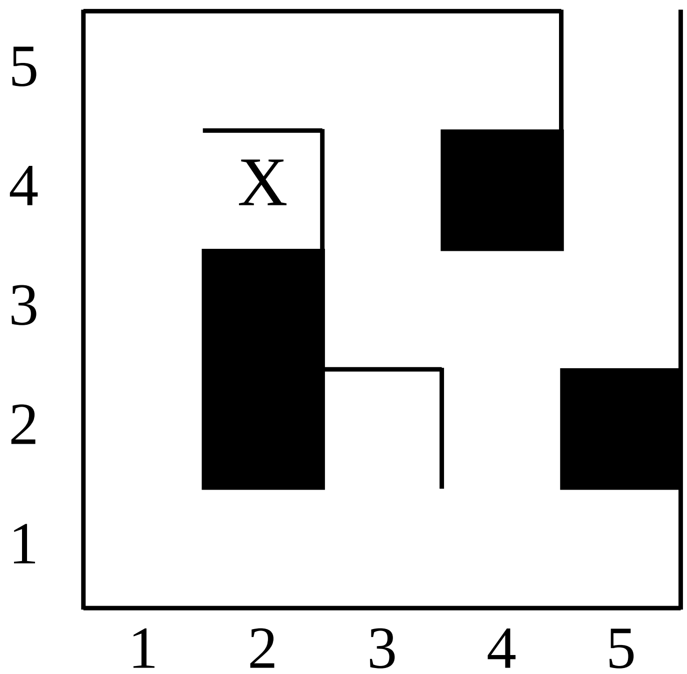
Exemple

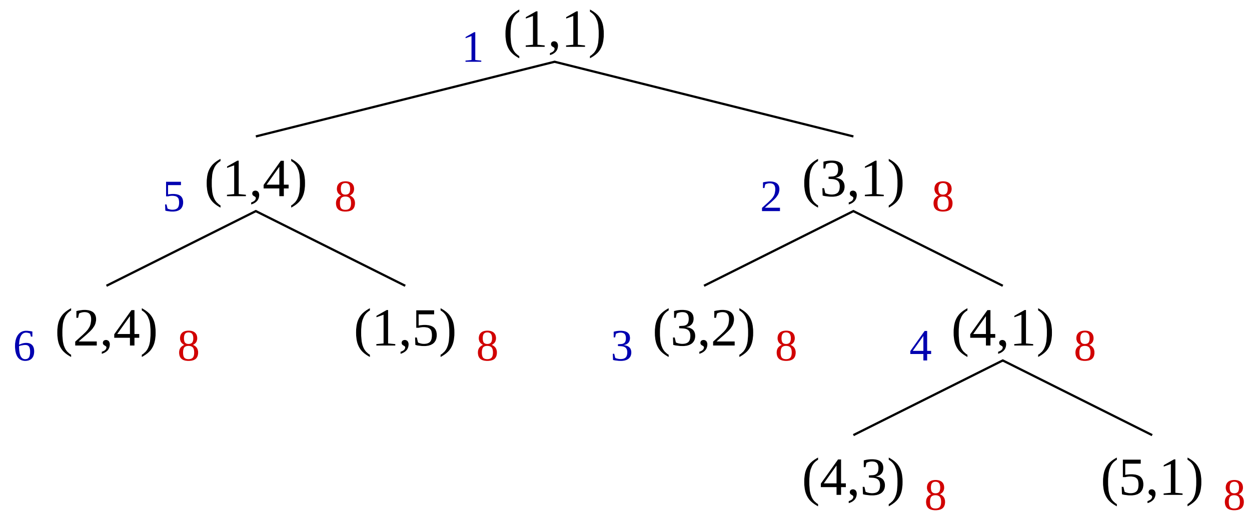




Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

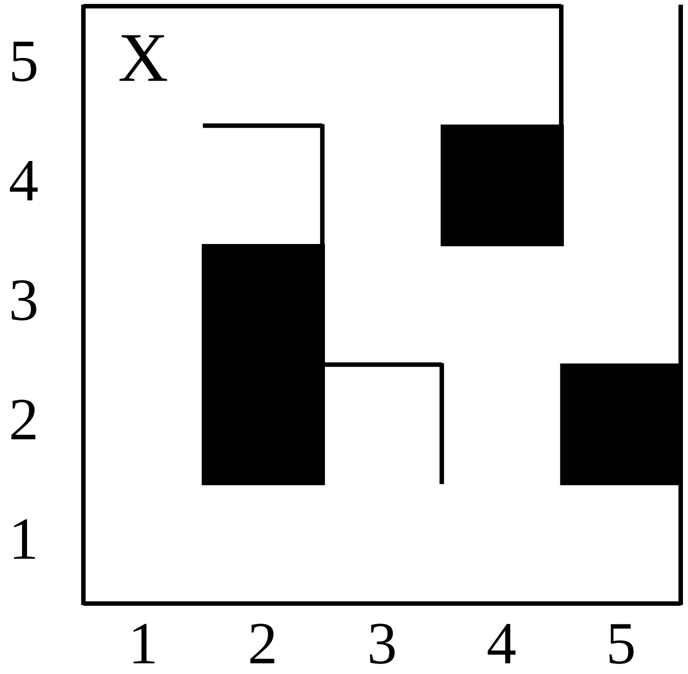
Exemple

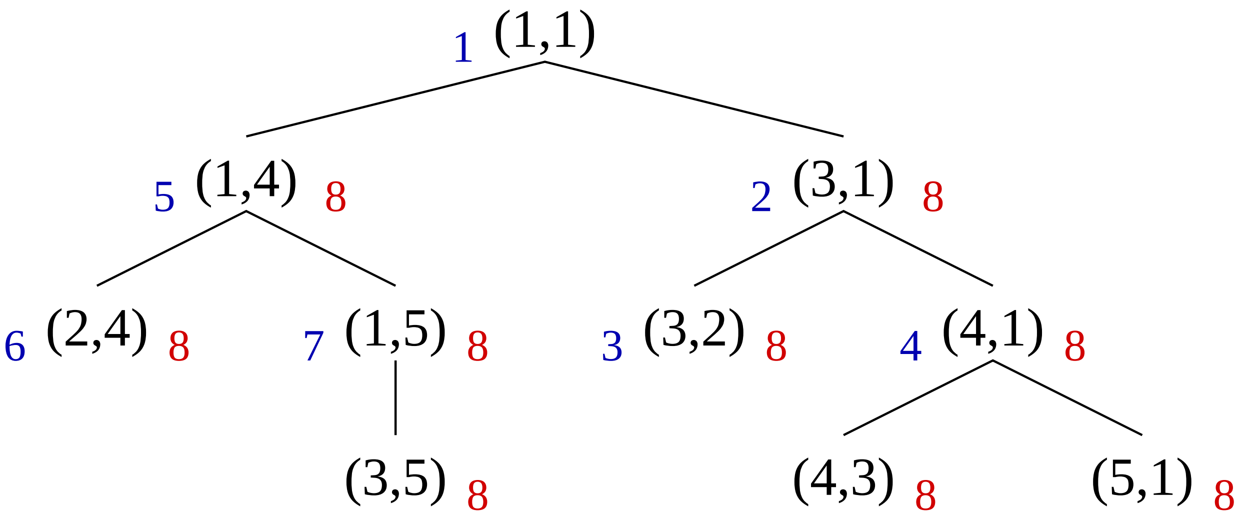




Distance de Manhattan : $h((x, y)) = |5 - x| + |5 - y|$

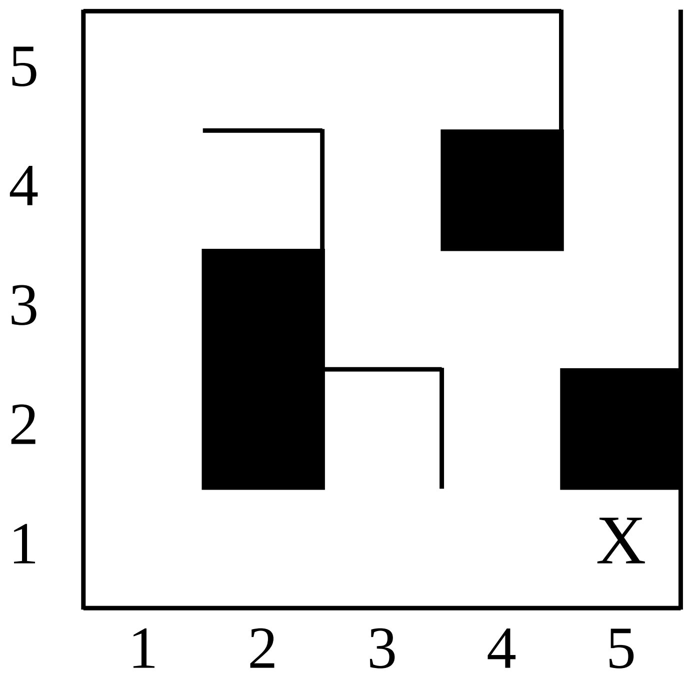
Exemple

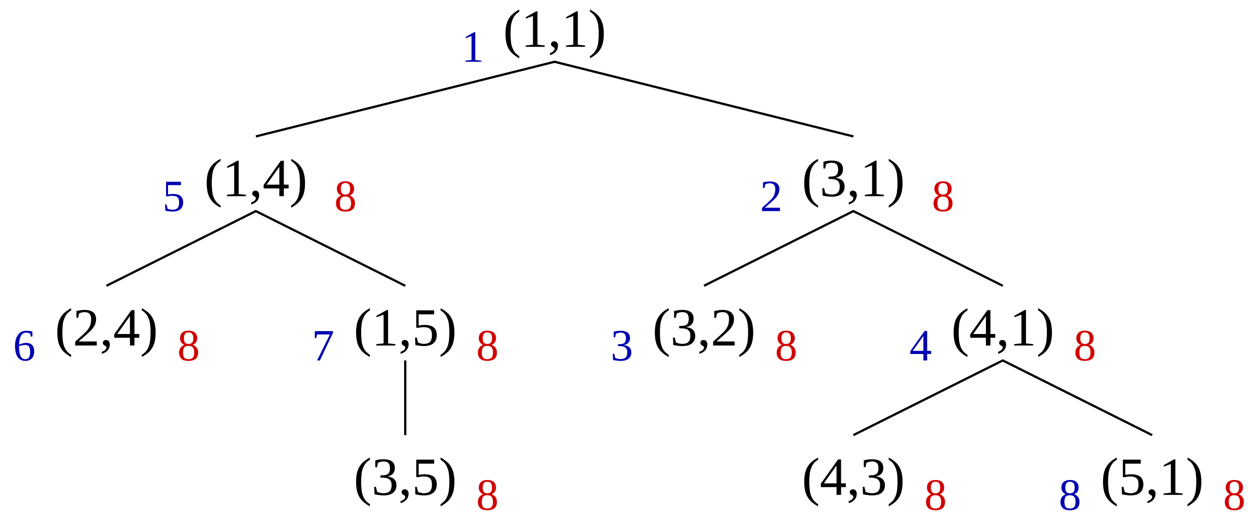




Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

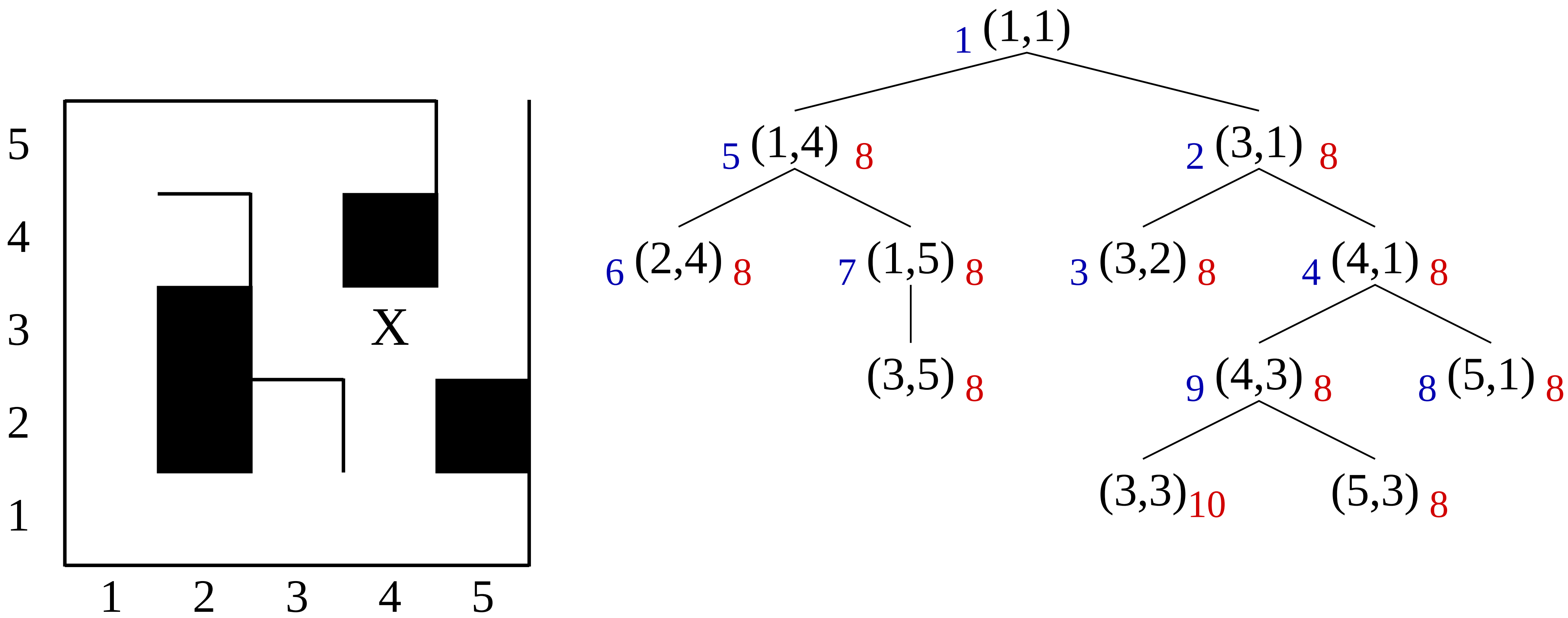
Exemple





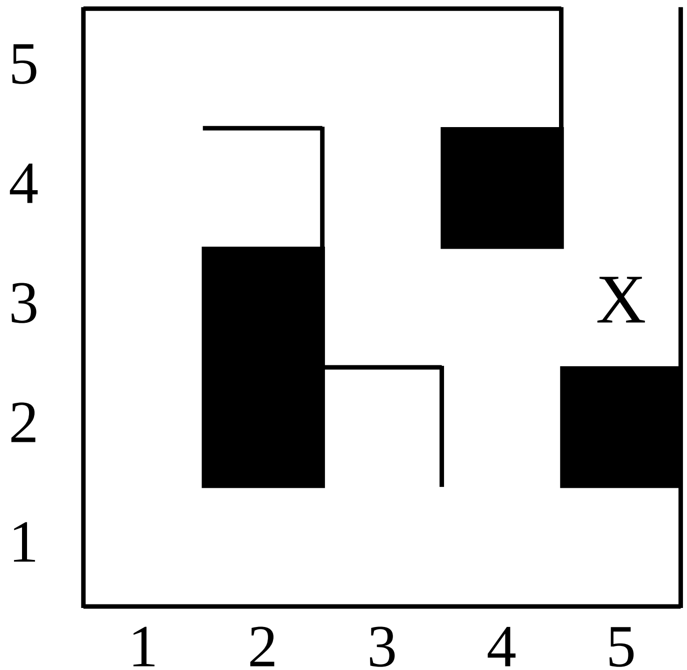
Distance de Manhattan : $h((x, y)) = |5 - x| + |5 - y|$

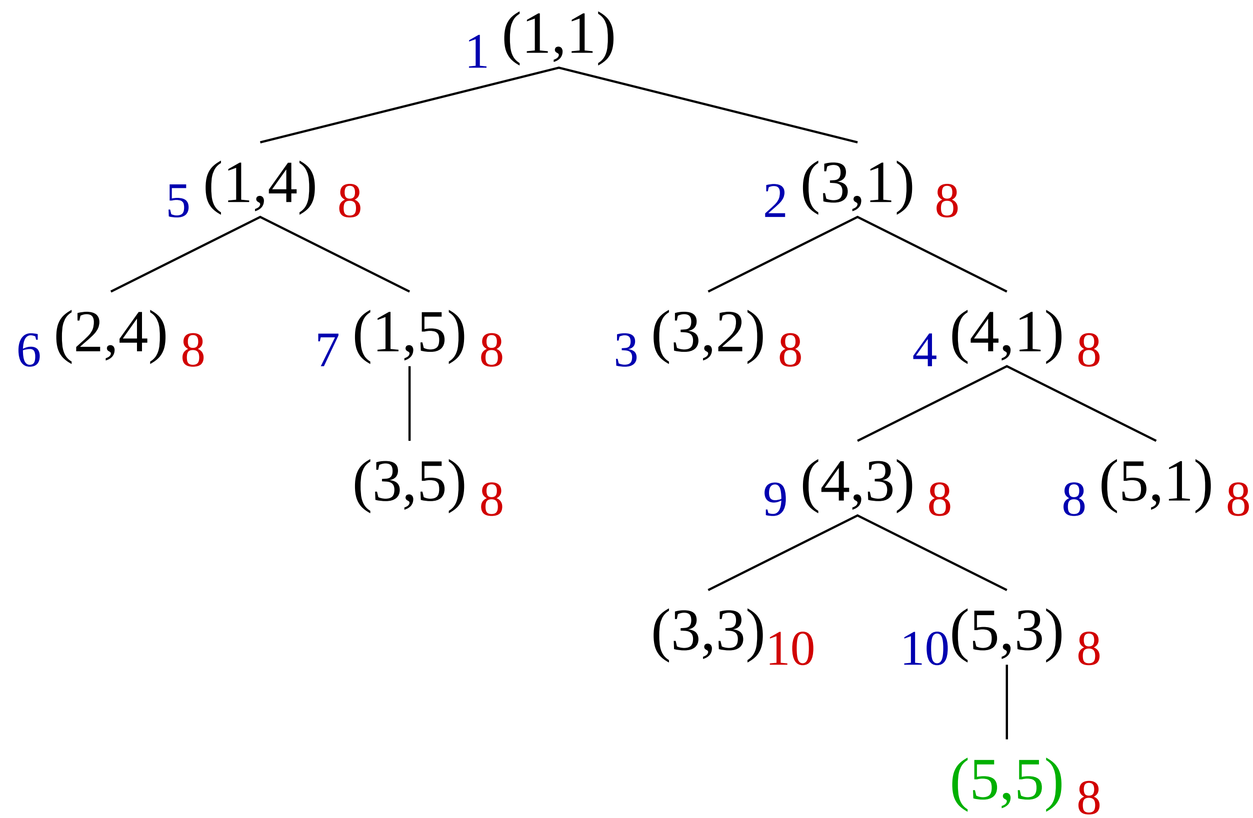
Exemple



Distance de Manhattan : $h((x, y)) = |5 - x| + |5 - y|$

Exemple





Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Plan

1. Méthodes de résolution arborescentes (suite et fin)
2. Méthodes incomplètes
3. Quelle méthode choisir ?

Les méthodes incomplètes

Effectuer une exploration non systématique

Les méthodes incomplètes

Effectuer une exploration non systématique

Principe :

- On part de l'état initial

Les méthodes incomplètes

Effectuer une exploration non systématique

Principe :

- On part de l'état initial
- A chaque étape, on choisit un seul état parmi tous les états voisins de l'état courant

Les méthodes incomplètes

Effectuer une exploration non systématique

Principe :

- On part de l'état initial
- A chaque étape, on choisit un seul état parmi tous les états voisins de l'état courant

Ce choix est souvent guidé par un critère de proximité

Les méthodes incomplètes

Effectuer une exploration non systématique

Principe :

- On part de l'état initial
- A chaque étape, on choisit un seul état parmi tous les états voisins de l'état courant

Ce choix est souvent guidé par un critère de proximité

Recherche locale ou par tâtonnement

Les méthodes incomplètes

$e \leftarrow e_{init}$
TantQue e n'est pas un but
  choisir e' parmi les voisins de e
  $e \leftarrow e'$
FinTantQue

Les méthodes incomplètes

$e \leftarrow e_{init}$
TantQue e n'est pas un but
    choisir e' parmi les voisins de e
    $e \leftarrow e'$
FinTantQue

La façon de choisir le prochain état caractérise la méthode.

Les méthodes incomplètes

Avantages :

- simples à implémenter

Les méthodes incomplètes

Avantages :

- simples à implémenter
- on peut rapidement trouver un chemin de e_{init} à un but

Les méthodes incomplètes

Avantages :

- simples à implémenter
- on peut rapidement trouver un chemin de e_{init} à un but

Inconvénients :

- on peut ne jamais atteindre un but

Les méthodes incomplètes

Avantages :

- simples à implémenter
- on peut rapidement trouver un chemin de e_{init} à un but

Inconvénients :

- on peut ne jamais atteindre un but
- on peut boucler à l'infini

Les méthodes incomplètes

Avantages :

- simples à implémenter
- on peut rapidement trouver un chemin de e_{init} à un but

Inconvénients :

- on peut ne jamais atteindre un but
- on peut boucler à l'infini
- on ne peut conclure que les buts sont inaccessibles

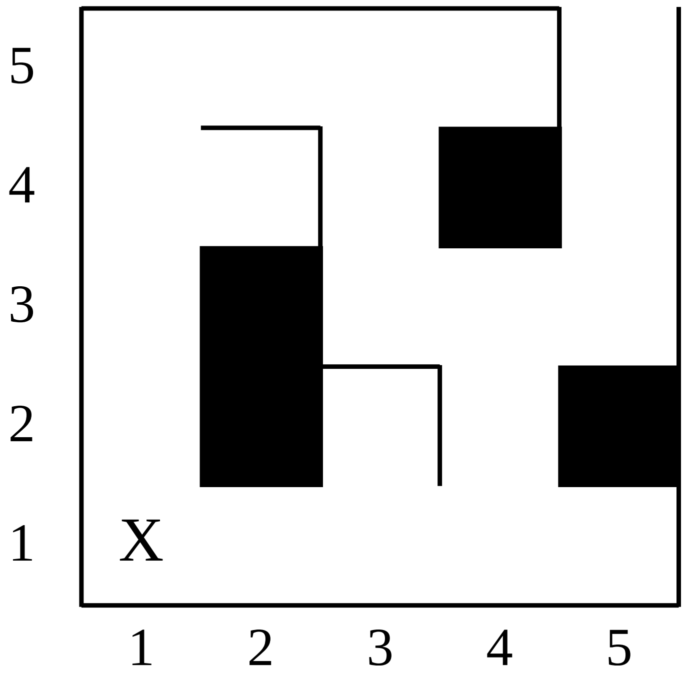
L'algorithme du British Museum

Choix aléatoire du prochain état

Si on rencontre une impasse, on repart de e_{init}

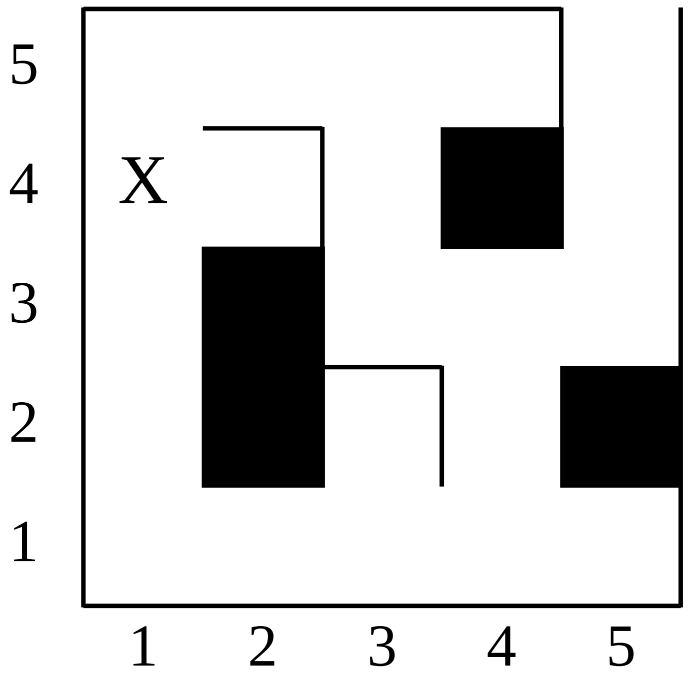
Très inefficace en pratique

Exemple



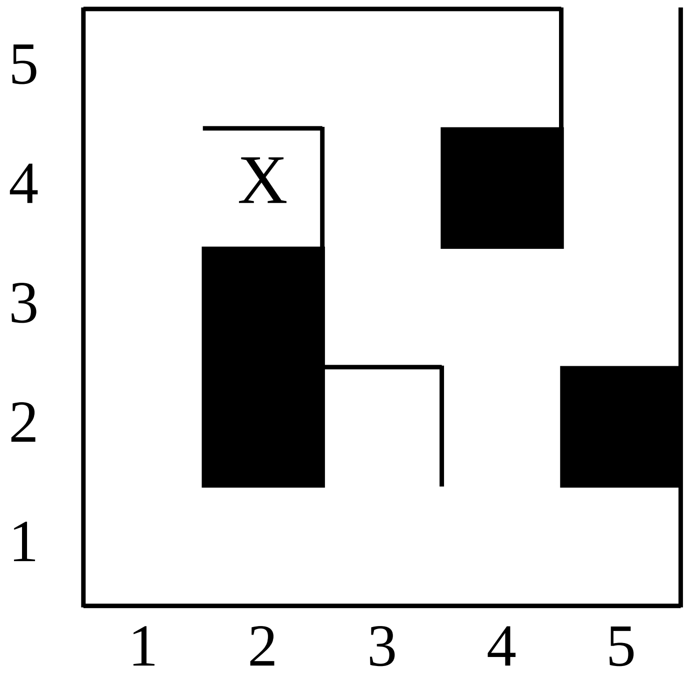
$(1, 1)$

Exemple



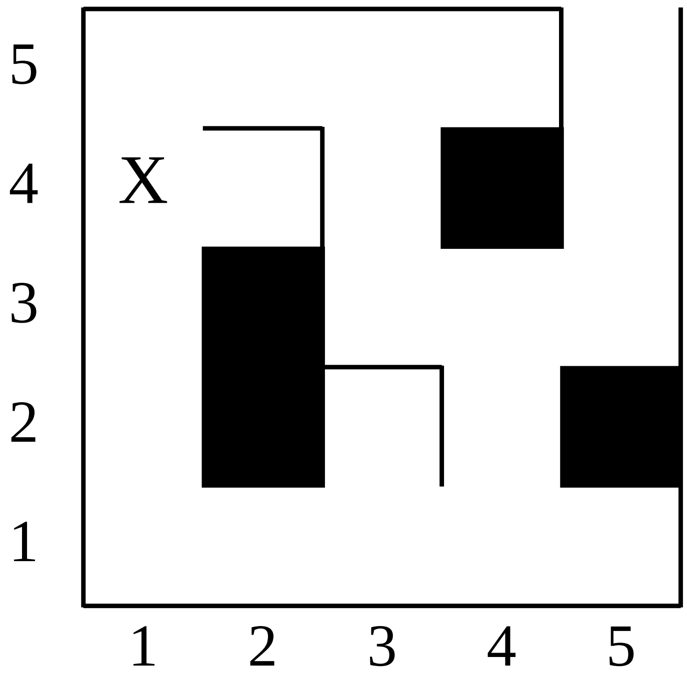
$$(1,1) \to (1,4)$$

Exemple



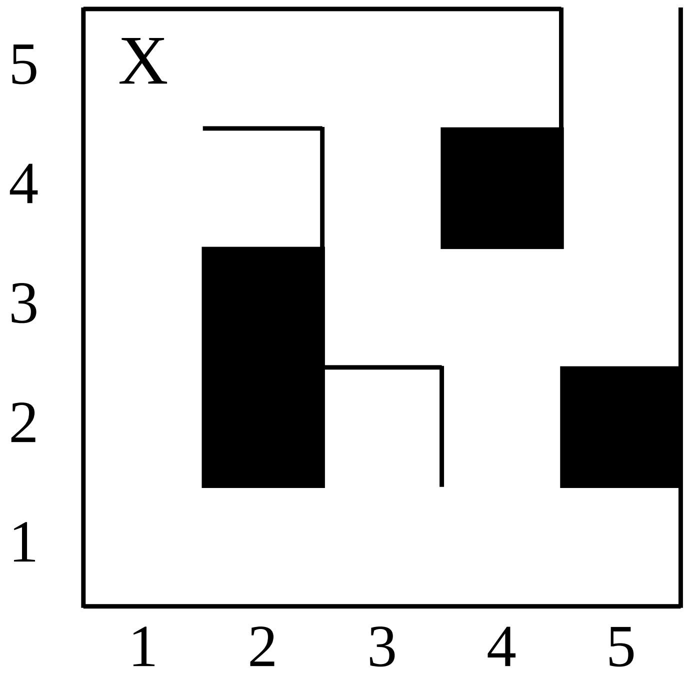
$$(1,1) \rightarrow (1,4) \rightarrow (2,4)$$

Exemple



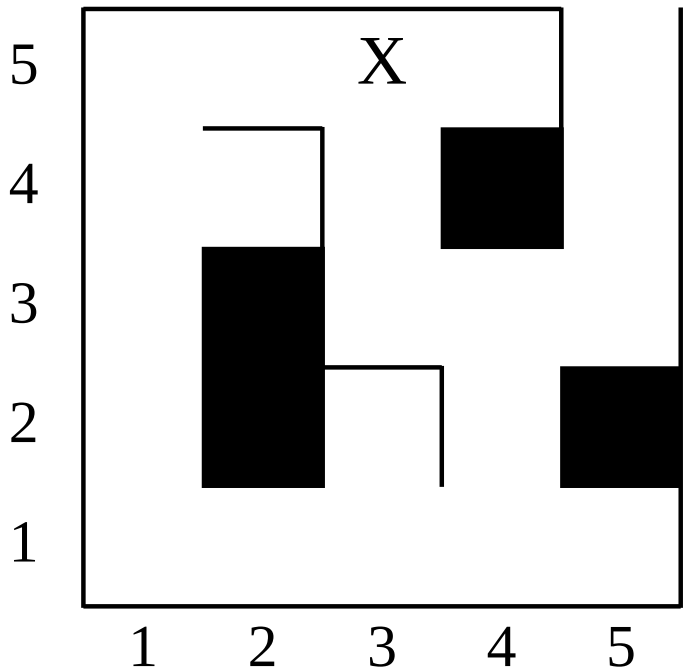
$$(1,1) \longrightarrow (1,4) \longrightarrow (2,4) \longrightarrow (1,4)$$

Exemple



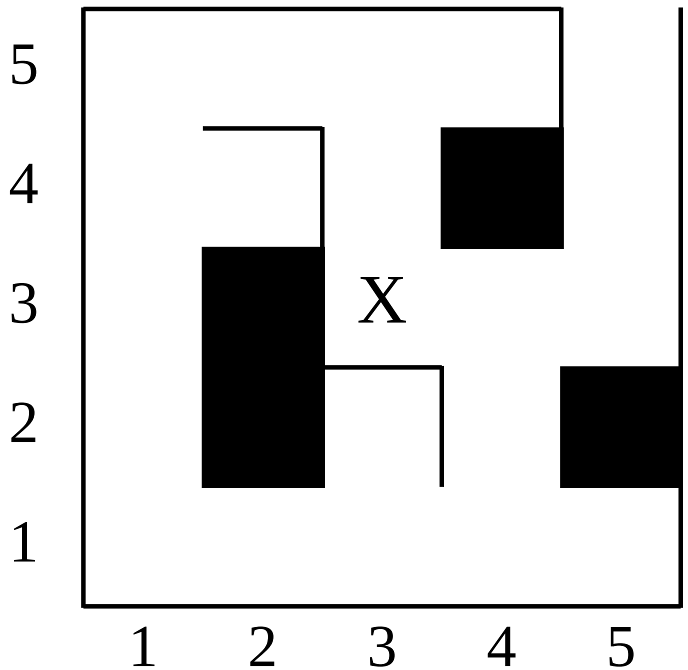
$$(1,1) \rightarrow (1,4) \rightarrow (2,4) \rightarrow (1,4) \rightarrow (1,5)$$

Exemple



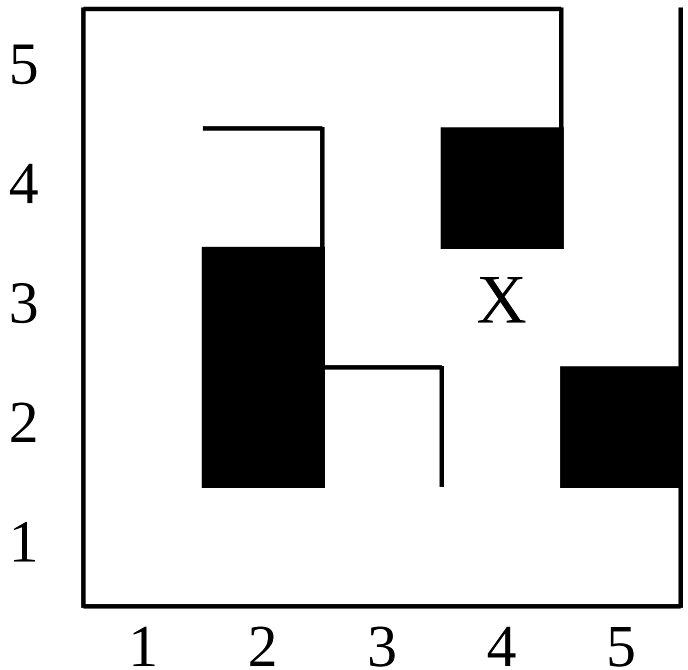
$(1,1) \rightarrow (1,4) \rightarrow (2,4) \rightarrow (1,4) \rightarrow (1,5)$
$\rightarrow (3,5)$

Exemple



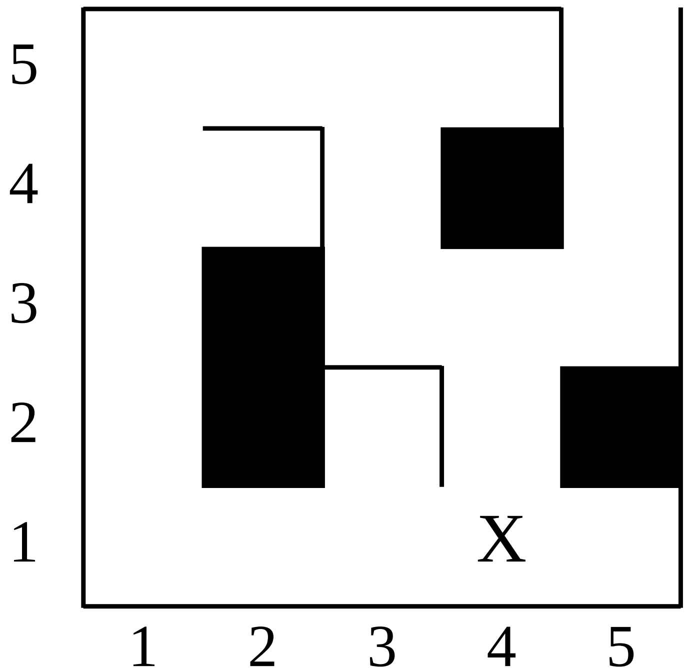
$$(1,1) \to (1,4) \to (2,4) \to (1,4) \to (1,5)$$
$$\to (3,5) \to (3,3)$$

Exemple



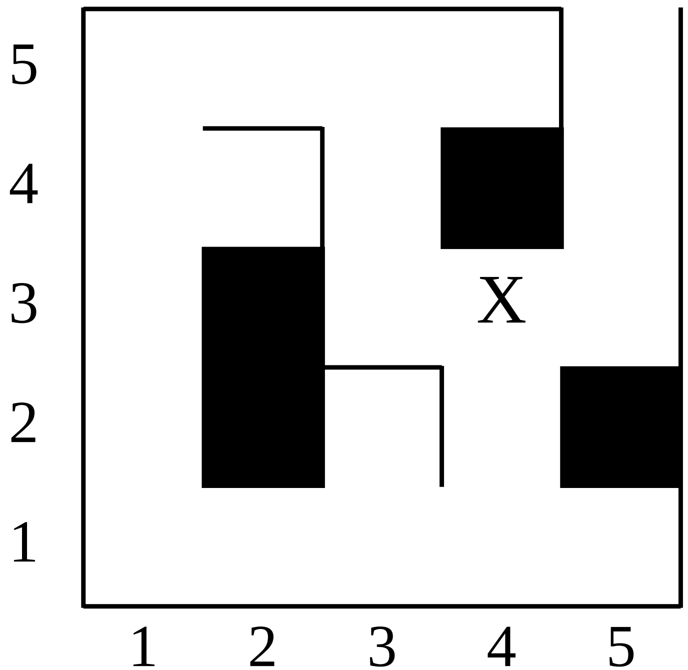
$(1,1) \rightarrow (1,4) \rightarrow (2,4) \rightarrow (1,4) \rightarrow (1,5)$
$\rightarrow (3,5) \rightarrow (3,3) \rightarrow (4,3)$

Exemple



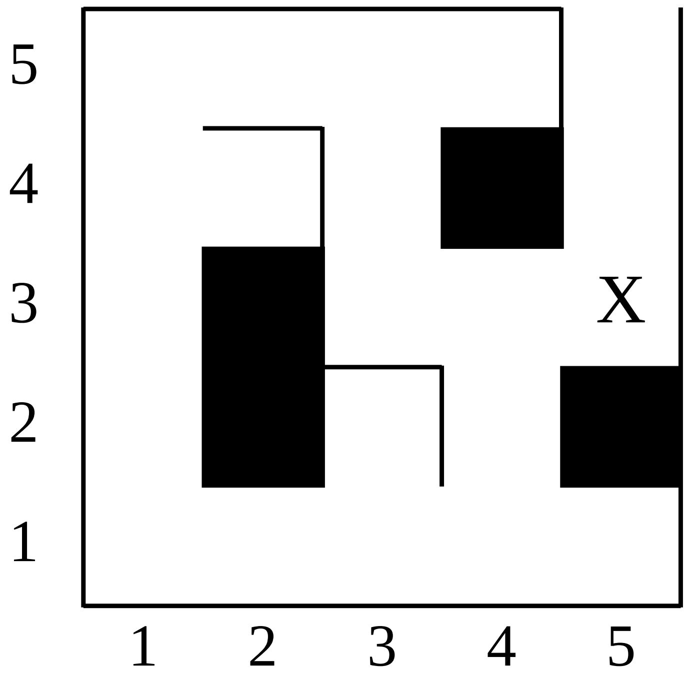
$$(1,1) \to (1,4) \to (2,4) \to (1,4) \to (1,5)$$
$$\to (3,5) \to (3,3) \to (4,3) \to (4,1)$$

Exemple



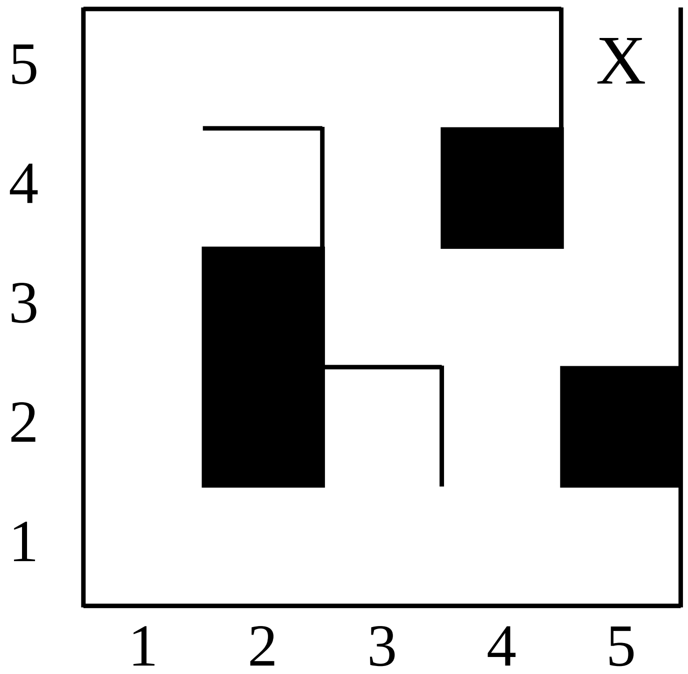
$(1,1) \rightarrow (1,4) \rightarrow (2,4) \rightarrow (1,4) \rightarrow (1,5)$
$\rightarrow (3,5) \rightarrow (3,3) \rightarrow (4,3) \rightarrow (4,1)$
$\rightarrow (4,3)$

Exemple



$$(1,1) \rightarrow (1,4) \rightarrow (2,4) \rightarrow (1,4) \rightarrow (1,5)$$
$$\rightarrow (3,5) \rightarrow (3,3) \rightarrow (4,3) \rightarrow (4,1)$$
$$\rightarrow (4,3) \rightarrow (5,3)$$

Exemple



$$(1,1) \rightarrow (1,4) \rightarrow (2,4) \rightarrow (1,4) \rightarrow (1,5)$$
$$\rightarrow (3,5) \rightarrow (3,3) \rightarrow (4,3) \rightarrow (4,1)$$
$$\rightarrow (4,3) \rightarrow (5,3) \rightarrow (5,5)$$

Le Hill-Climbing (ou escalade)

Choix du prochain état : un de ceux qui améliorent le critère de proximité

Le Hill-Climbing (ou escalade)

Choix du prochain état : un de ceux qui améliorent le critère de proximité

critère de proximité = une heuristique estimant l’éloignement de l’état considéré à un but

Le Hill-Climbing (ou escalade)

Choix du prochain état : un de ceux qui améliorent le critère de proximité

critère de proximité = une heuristique estimant l'éloignement de l'état considéré à un but

Inconvénient : la méthode peut atteindre un optimum local

Le Hill-Climbing (ou escalade)

Choix du prochain état : un de ceux qui améliorent le critère de proximité

critère de proximité = une heuristique estimant l'éloignement de l'état considéré à un but

Inconvénient : la méthode peut atteindre un optimum local

optimum local = état pour lequel aucun état voisin n'améliore le critère

Le Hill-Climbing (ou escalade)

Choix du prochain état : un de ceux qui améliorent le critère de proximité

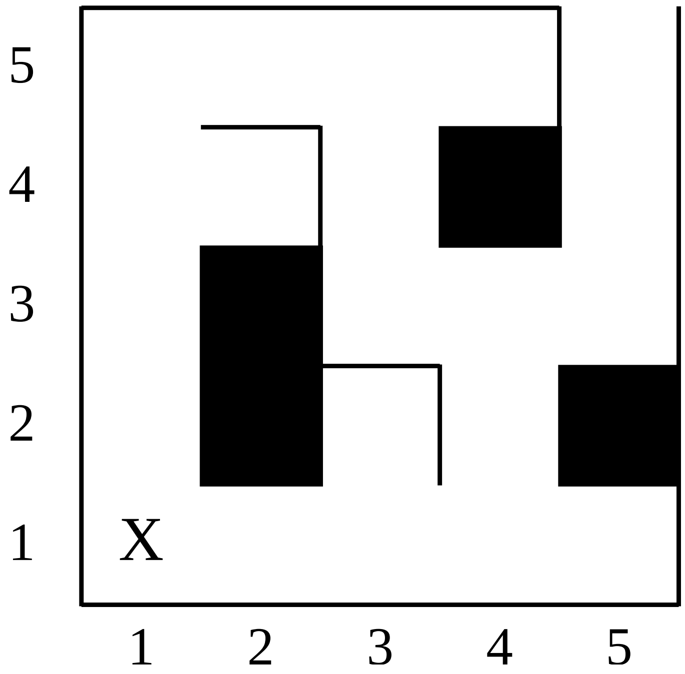
critère de proximité = une heuristique estimant l'éloignement de l'état considéré à un but

Inconvénient : la méthode peut atteindre un optimum local

optimum local = état pour lequel aucun état voisin n'améliore le critère

Efficacité très variable suivant le nombre d'optima locaux

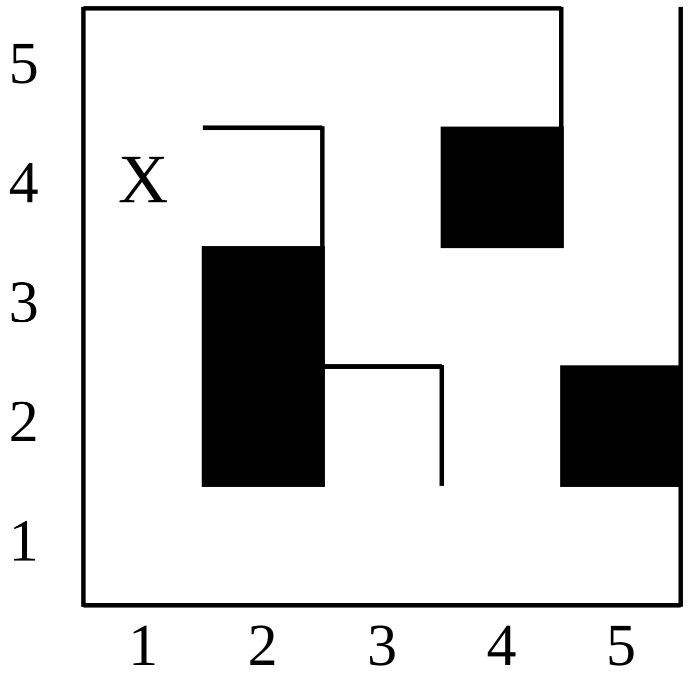
Exemple



$(1, 1)$

Distance de Manhattan : $h((x, y)) = |5 - x| + |5 - y|$

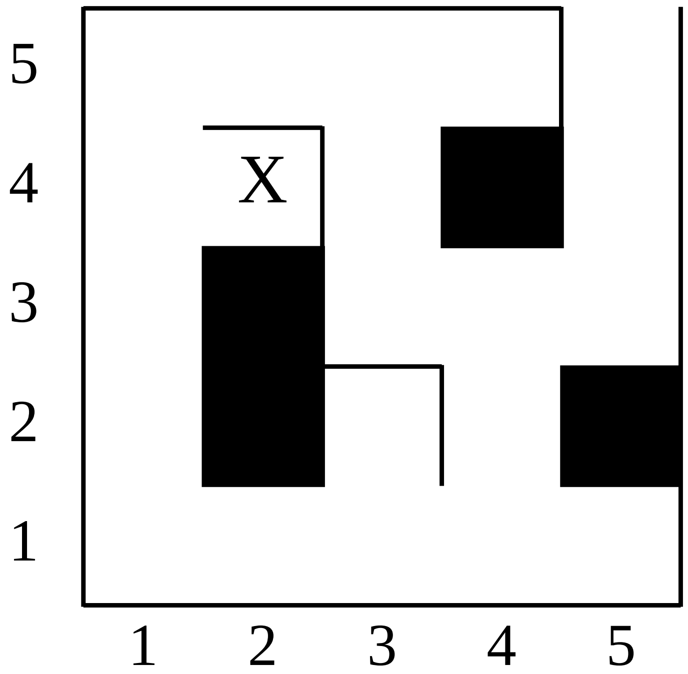
Exemple



$$(1,1) \rightarrow (1,4)_5$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

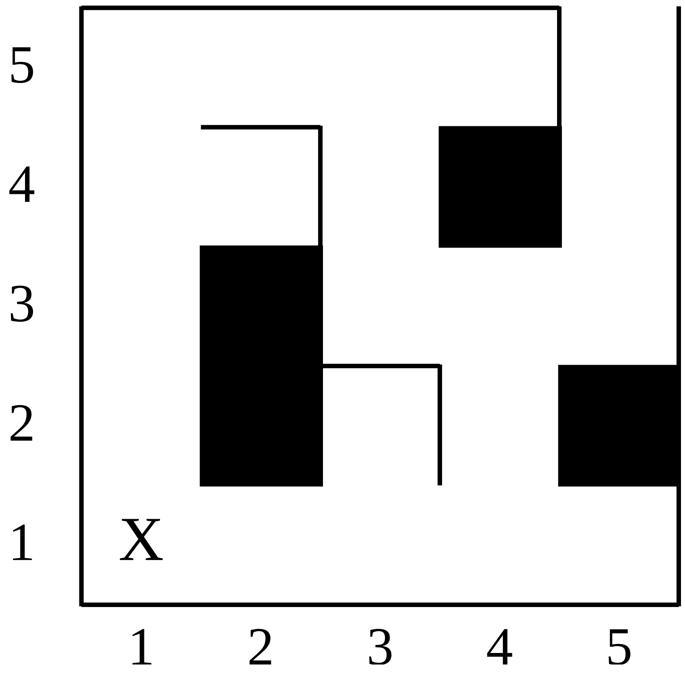
Exemple



$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple

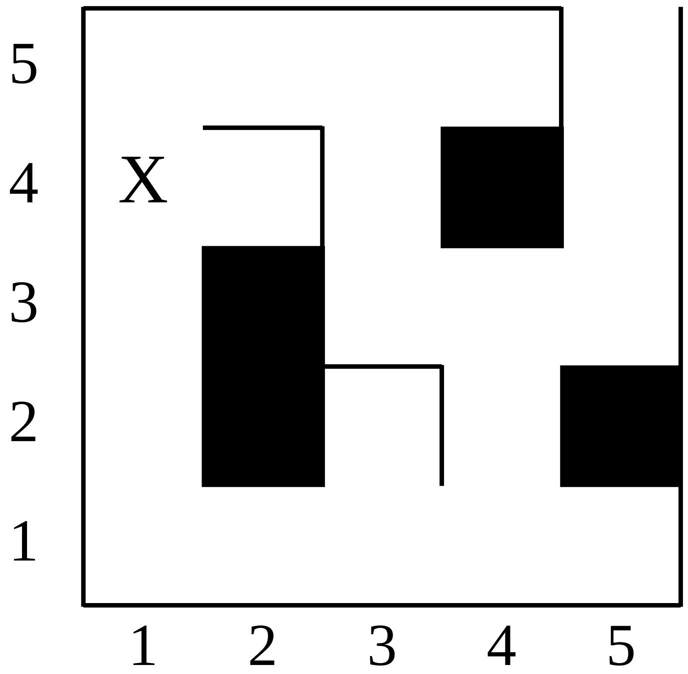


$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1)$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple

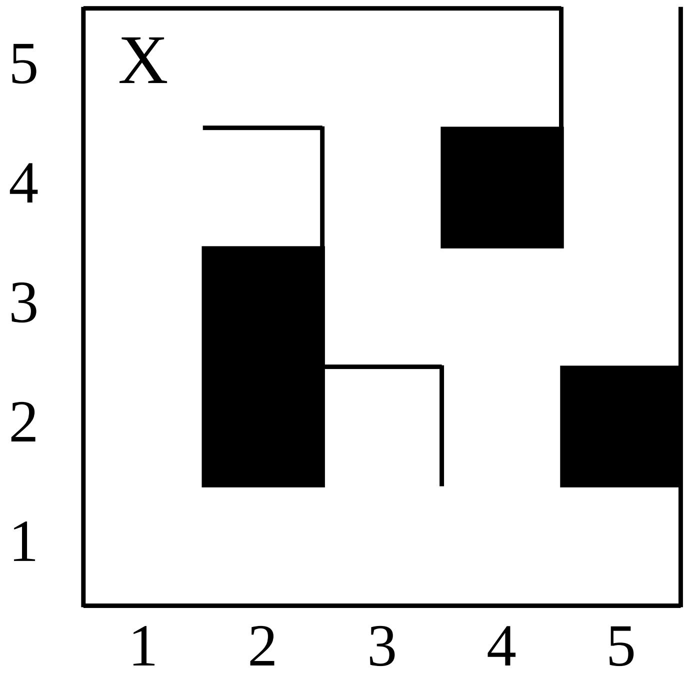


$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1) \rightarrow (1,4)_5$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple

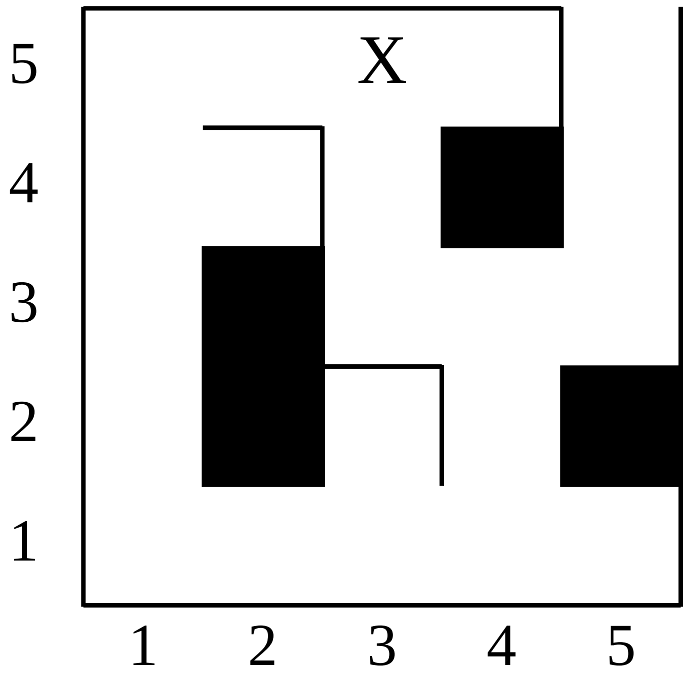


$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple

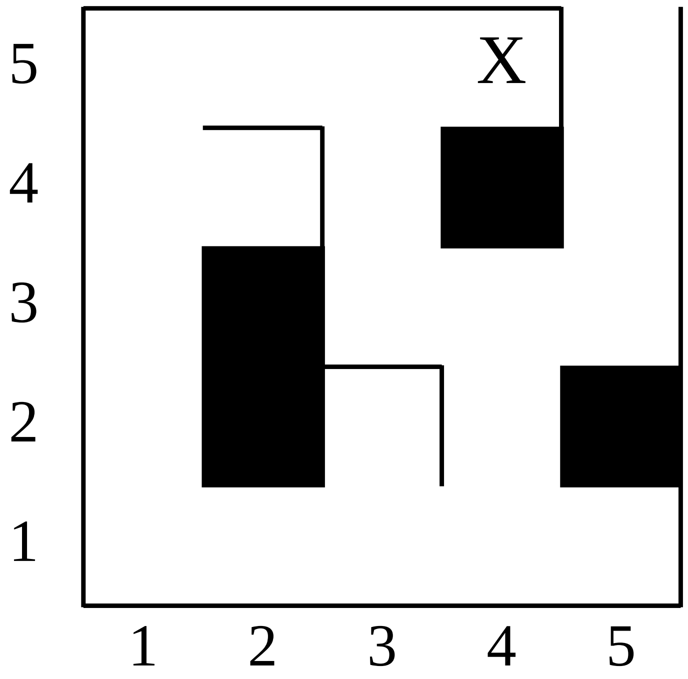


$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4 \rightarrow (3,5)_2$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple

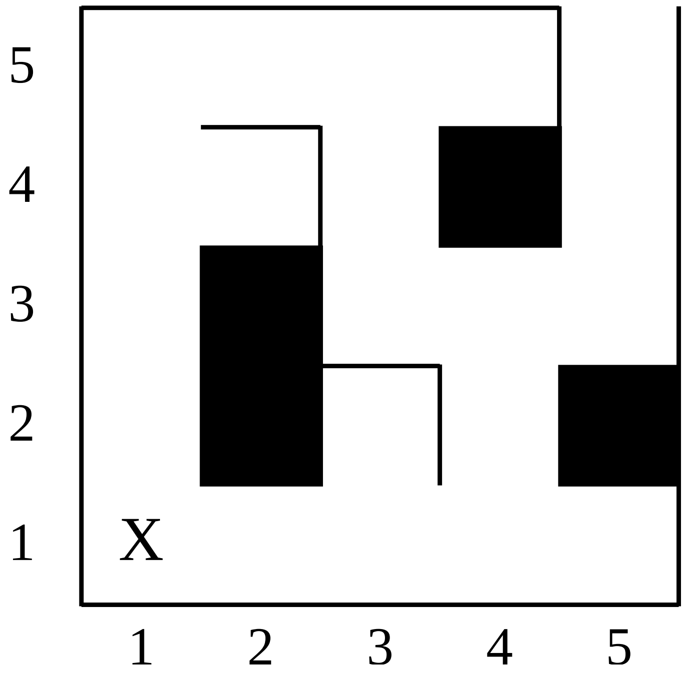


$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4 \rightarrow (3,5)_2 \rightarrow (4,5)_1$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple



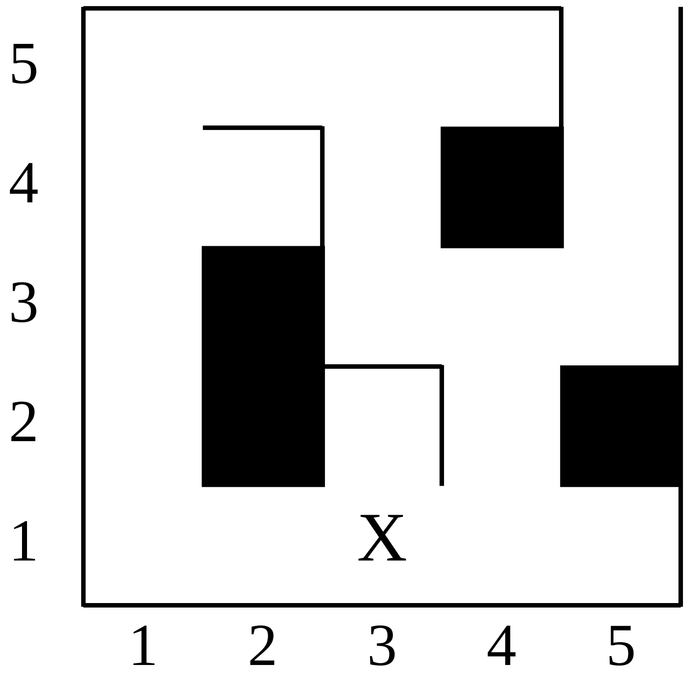
$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$

$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4 \rightarrow (3,5)_2 \rightarrow (4,5)_1$

$(1,1)$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple



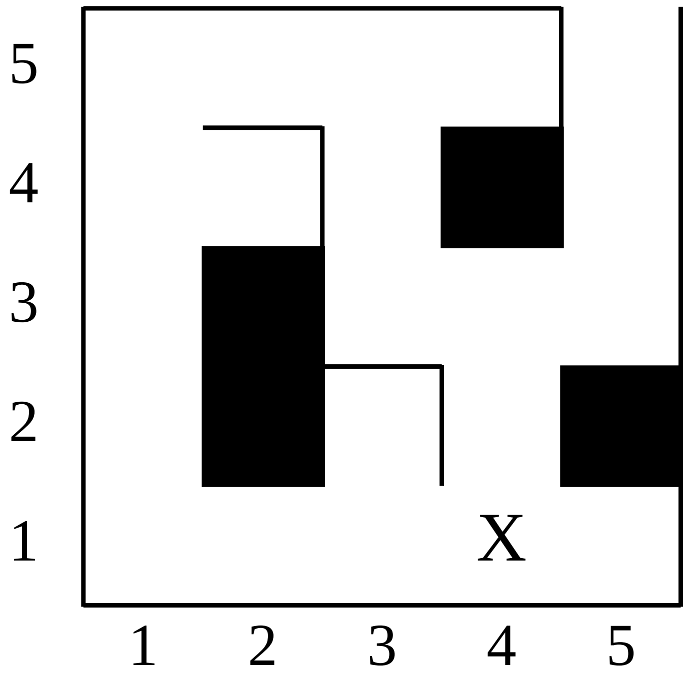
$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4 \rightarrow (3,5)_2 \rightarrow (4,5)_1$$

$$(1,1) \rightarrow (3,1)_6$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple



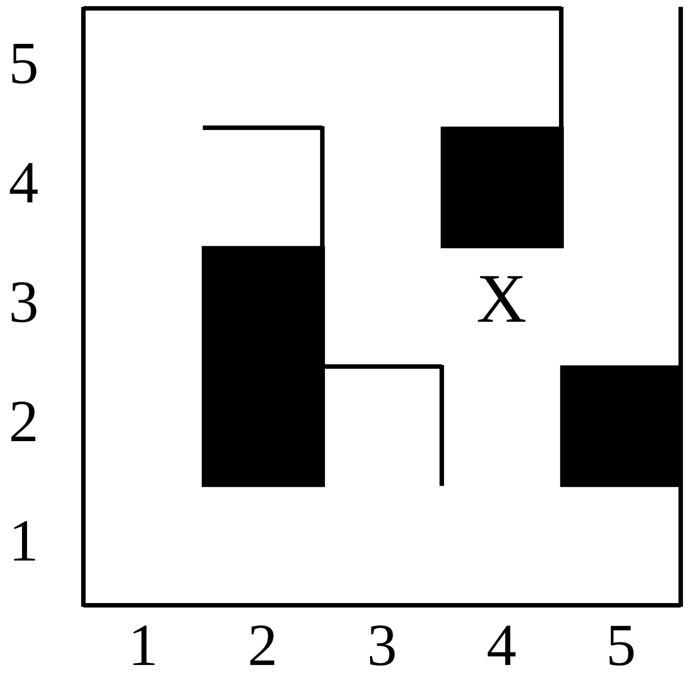
$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4 \rightarrow (3,5)_2 \rightarrow (4,5)_1$$

$$(1,1) \rightarrow (3,1)_6 \rightarrow (4,1)_5$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple



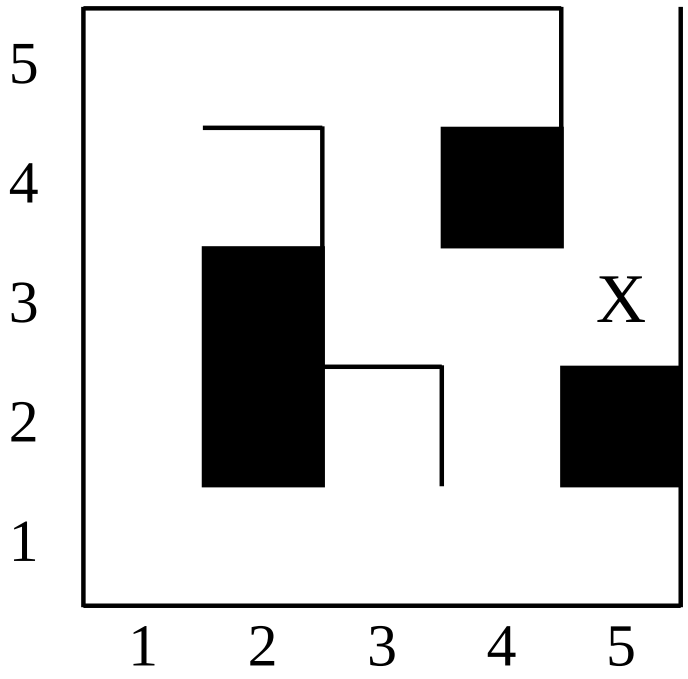
$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4 \rightarrow (3,5)_2 \rightarrow (4,5)_1$$

$$(1,1) \rightarrow (3,1)_6 \rightarrow (4,1)_5 \rightarrow (4,3)_3$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple



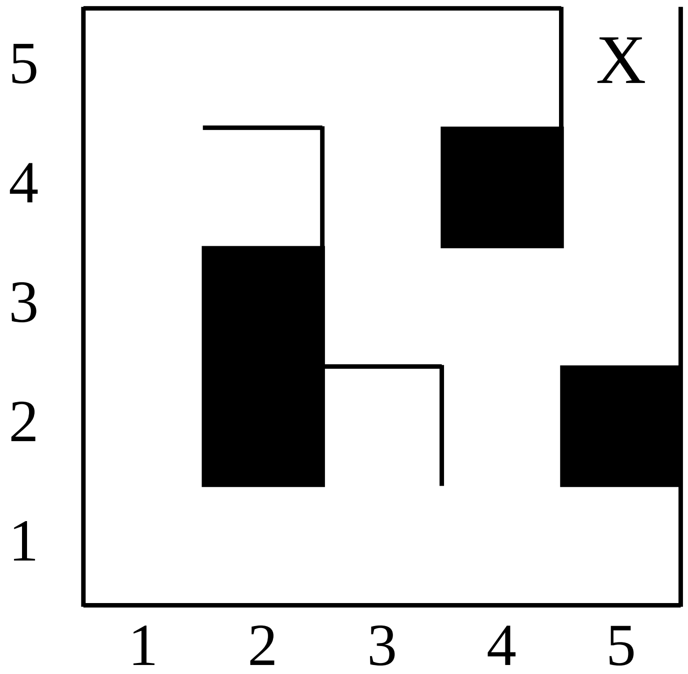
$$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$$

$$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4 \rightarrow (3,5)_2 \rightarrow (4,5)_1$$

$$(1,1) \rightarrow (3,1)_6 \rightarrow (4,1)_5 \rightarrow (4,3)_3 \rightarrow (5,3)_2$$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

Exemple



$(1,1) \rightarrow (1,4)_5 \rightarrow (2,4)_4$

$(1,1) \rightarrow (1,4)_5 \rightarrow (1,5)_4 \rightarrow (3,5)_2 \rightarrow (4,5)_1$

$(1,1) \rightarrow (3,1)_6 \rightarrow (4,1)_5 \rightarrow (4,3)_3 \rightarrow (5,3)_2$
$\rightarrow (5,5)_0$

Distance de Manhattan : $h((x,y)) = |5-x| + |5-y|$

La recherche taboue

Objectif : éviter de boucler

La recherche taboue

Objectif : éviter de boucler

Emploi d'une liste dite "taboue" contenant les états déjà étudiés

La recherche taboue

Objectif : éviter de boucler

Emploi d'une liste dite "taboue" contenant les états déjà étudiés

Choix du prochain sommet : le meilleur voisin n'appartenant pas à la liste taboue

La recherche taboue

Objectif : éviter de boucler

Emploi d'une liste dite "taboue" contenant les états déjà étudiés

Choix du prochain sommet : le meilleur voisin n'appartenant pas à la liste taboue

Inconvénient : le coût en mémoire peut devenir prohibitif

La recherche taboue

Objectif : éviter de boucler

Emploi d'une liste dite "taboue" contenant les états déjà étudiés

Choix du prochain sommet : le meilleur voisin n'appartenant pas à la liste taboue

Inconvénient : le coût en mémoire peut devenir prohibitif

$\Rightarrow$ en pratique, on emploie une liste de taille limitée

Le recuit simulé

Objectif : sortir du piège des optima locaux

Le recuit simulé

Objectif : sortir du piège des optima locaux

Choix du prochain état e':

- soit un état voisin qui améliore le critère de proximité

Le recuit simulé

Objectif : sortir du piège des optima locaux

Choix du prochain état e':

- soit un état voisin qui améliore le critère de proximité
- soit un état voisin qui détériore le critère avec une probabilité de $e^{-\frac{\Delta f}{T}}$

Le recuit simulé

Objectif : sortir du piège des optima locaux

Choix du prochain état e':

- soit un état voisin qui améliore le critère de proximité
- soit un état voisin qui détériore le critère avec une probabilité de $e^{-\frac{\Delta f}{T}}$

Δf = différence de valeur entre e et e'

Le recuit simulé

Objectif : sortir du piège des optima locaux

Choix du prochain état e':

- soit un état voisin qui améliore le critère de proximité
- soit un état voisin qui détériore le critère avec une probabilité de $e^{-\frac{\Delta f}{T}}$

Δf = différence de valeur entre e et e'

T = température qui décroît au cours du temps

Autres méthodes

- Les algorithmes génétiques
- La réparation locale

Autres méthodes

- Les algorithmes génétiques
- La réparation locale

Ne sont pas applicables aux graphes d'états !!!

Commentaires

Chacune de ces méthodes possède des paramètres :

Commentaires

Chacune de ces méthodes possède des paramètres :

- heuristique d'évaluation de l'éloignement

Commentaires

Chacune de ces méthodes possède des paramètres :

- heuristique d'évaluation de l'éloignement
- vitesse à laquelle décroît la température

Commentaires

Chacune de ces méthodes possède des paramètres :

- heuristique d'évaluation de l'éloignement
- vitesse à laquelle décroît la température
- longueur de la liste taboue

Commentaires

Chacune de ces méthodes possède des paramètres :

- heuristique d’évaluation de l’éloignement
- vitesse à laquelle décroît la température
- longueur de la liste taboue

Leur efficacité dépend du choix judicieux des valeurs des paramètres.

Plan

1. Méthodes de résolution arborescentes (suite et fin)
2. Méthodes incomplètes
3. Quelle méthode choisir ?

Plusieurs problèmes possibles

- Problème de décision
  Peut-on aller de l'état initial à un état final ?

Plusieurs problèmes possibles

- Problème de décision
  Peut-on aller de l’état initial à un état final ?
- Problème du plus court chemin
  Trouver un plus court chemin pour aller de l’état initial à un état final.

Plusieurs problèmes possibles

- Problème de décision
  Peut-on aller de l'état initial à un état final ?
- Problème du plus court chemin
  Trouver un plus court chemin pour aller de l'état initial à un état final.
- Problème d'optimisation
  Trouver le meilleur chemin pour aller de l'état initial à un état final.

Plusieurs problèmes possibles

- Problème de décision
  Peut-on aller de l'état initial à un état final ?
- Problème du plus court chemin
  Trouver un plus court chemin pour aller de l'état initial à un état final.
- Problème d'optimisation
  Trouver le meilleur chemin pour aller de l'état initial à un état final.
- Problème de dénombrement
  Combien existe-t-il de chemin pour aller de l'état initial à un état final ?

Quelle méthode pour quel problème

- Problème de décision : méthodes complètes

Quelle méthode pour quel problème

- Problème de décision : méthodes complètes
- Problème du plus court chemin : A*

Quelle méthode pour quel problème

- Problème de décision : méthodes complètes
- Problème du plus court chemin : A*

Qu’en est-il des méthodes incomplètes ?

Quelle méthode pour quel problème

- Problème de décision : méthodes complètes
- Problème du plus court chemin : A*

Qu'en est-il des méthodes incomplètes ?

A utiliser pour les problèmes de décision
(surtout quand l'espace d'états est de taille importante)

Quelle méthode pour quel problème

- Problème de décision : méthodes complètes
- Problème du plus court chemin : A*

Qu'en est-il des méthodes incomplètes ?

A utiliser pour les problèmes de décision
(surtout quand l'espace d'états est de taille importante)

Attention, on a aucune garantie de trouver un chemin !

Choix parmi les méthodes complètes

- Recherche en profondeur

Choix parmi les méthodes complètes

- Recherche en profondeur
- Recherche en largeur

Choix parmi les méthodes complètes

- Recherche en profondeur
- Recherche en largeur
- Recherche le meilleur d'abord

Choix parmi les méthodes complètes

- Recherche en profondeur
- Recherche en largeur
- Recherche le meilleur d'abord
- Iterative Deepening : quand le temps est limité

Choix parmi les méthodes complètes

- Recherche en profondeur
- Recherche en largeur
- Recherche le meilleur d'abord
- Iterative Deepening : quand le temps est limité
- Recherche à divergence limitée :
  quand on dispose d'une très bonne heuristique