Accélérer l'exploration

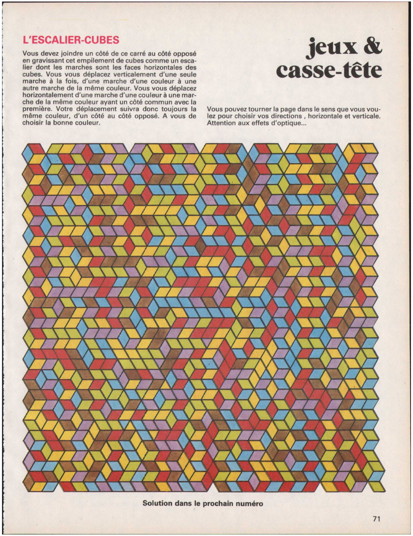
A-star

Jean-Marc.Vincent@univ-grenoble-alpes.fr[1]

[1] Laboratoire LIG
Équipe-Projet INRIA POLARIS
Université Grenoble-Alpes

Licence Informatique
Grenoble

Évasion


L'ESCALIER-CUBES

Vous devez joindre un côté de ce carré au côté opposé en gravissant cet empilement de cubes comme un escalier dont les marches sont les faces horizontales des cubes. Vous vous déplacez verticalement d'une seule marche à la fois, d'une marche d'une couleur à une autre marche de la même couleur. Vous vous déplacez horizontalement d'une marche d'une couleur à une marche de la même couleur ayant un côté commun avec la première. Votre déplacement suivra donc toujours la même couleur, d'un côté au côté opposé. A vous de choisir la bonne couleur.

jeux & casse-tête

Vous pouvez tourner la page dans le sens que vous voulez pour choisir vos directions , horizontale et verticale. Attention aux effets d'optique...

Solution dans le prochain numéro

71

Source : Jeux & Stratégie n 28 (1984)

ALGORITHME DE DIJKSTRA

Algorithme_Dijkstra($\mathcal{G}$,s)

Données: Un graphe $\mathcal{G}$ valué et un sommet s de $\mathcal{G}$

Résultat: Visite tous les sommets du graphe et calcule la valeur minimale des chemins de s à tout autre sommet du graphe

$d(s) = 0$
$S \leftarrow \{s\}$
for $y \in \mathcal{X} - S$
  $d(y) = v(s, y)$

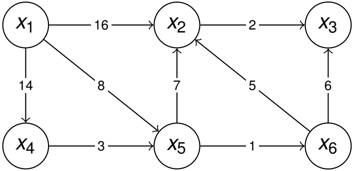
// S ensemble de sommets (chemins) dont la valeur minimale d est établie

while il existe $(x, y) \in \mathcal{A}$ **avec** $x \in S$ **et** $y \notin S$
  Choisir $y \notin S$ minimisant

$$\pi(y) = \min_{x \in S, (x,y) \in \mathcal{A}} d(x) + v(x, y)$$

  $S \leftarrow S \cup \{y\}$
  $d(y) = \pi(y)$
  for $z \in \mathcal{X} - S$
    $d(z) = \min(d(z), d(y) + v(y, z))$

Exemple



	S	$d(1)$	$d(2)$	$d(3)$	$d(4)$	$d(5)$	$d(6)$
0	$\{x_1\}$	0	16	$+\infty$	14	8	$+\infty$
1							
2							
3							
4							
5							

Heuristique

Trouver le plus court chemin de o à b

- Une **heuristique** H est une fonction qui associe une valeur H_x pour tout sommet x du graphe.
  Interprétation : H est une estimation de la valeur d'un chemin de x à au but b (fonction externe liée à la nature du problème).

Heuristique

Trouver le plus court chemin de *o* à *b*

- Une **heuristique** H est une fonction qui associe une valeur H_x pour tout sommet x du graphe.
  Interprétation : H est une estimation de la valeur d'un chemin de x à au but b (fonction externe liée à la nature du problème).
- Une heuristique H est **admissible** (ou optimiste) si pour tout x

  $$H_x \leqslant d_{\min}(x, b)$$

  Interprétation : Au mieux la longueur d'un chemin de x à la destination (ou but) sera H_x

Heuristique

Trouver le plus court chemin de *o* à *b*

- Une **heuristique** H est une fonction qui associe une valeur H_x pour tout sommet x du graphe.
  Interprétation : H est une estimation de la valeur d'un chemin de x à au but b (fonction externe liée à la nature du problème).
- Une heuristique H est **admissible** (ou optimiste) si pour tout x

  $$H_x \leqslant d_{\min}(x, b)$$

  Interprétation : Au mieux la longueur d'un chemin de x à la destination (ou but) sera H_x
- Une heuristique H est **cohérente** si pour tout arc (x, y)

  $$H_x \leqslant v(x, y) + H_y \text{ ou encore } H_x - H_y \leqslant v(x, y)$$

  Interprétation : En passant par y on pourrait éventuellement faire mieux

Heuristique

Trouver le plus court chemin de o à b

- Une **heuristique** H est une fonction qui associe une valeur H_x pour tout sommet x du graphe.
  Interprétation : H est une estimation de la valeur d'un chemin de x à au but b (fonction externe liée à la nature du problème).
- Une heuristique H est **admissible** (ou optimiste) si pour tout x

  $$H_x \leqslant d_{\min}(x, b)$$

  Interprétation : Au mieux la longueur d'un chemin de x à la destination (ou but) sera H_x
- Une heuristique H est **cohérente** si pour tout arc (x, y)

  $$H_x \leqslant v(x, y) + H_y \text{ ou encore } H_x - H_y \leqslant v(x, y)$$

  Interprétation : En passant par y on pourrait éventuellement faire mieux

HEURISTIQUE

Trouver le plus court chemin de o à b

- Une **heuristique** H est une fonction qui associe une valeur H_x pour tout sommet x du graphe.
  Interprétation : H est une estimation de la valeur d'un chemin de x à au but b (fonction externe liée à la nature du problème).
- Une heuristique H est **admissible** (ou optimiste) si pour tout x

  $$H_x \leqslant d_{\min}(x, b)$$

  Interprétation : Au mieux la longueur d'un chemin de x à la destination (ou but) sera H_x
- Une heuristique H est **cohérente** si pour tout arc (x, y)

  $$H_x \leqslant v(x, y) + H_y \text{ ou encore } H_x - H_y \leqslant v(x, y)$$

  Interprétation : En passant par y on pourrait éventuellement faire mieux

Remarque : une heuristique cohérente est admissible (réciproque fausse)

$$c_{\min}(x, b) = < x = x_0, x_1, \cdots, x_k = b >$$

$$\begin{aligned} v(c_{\min}(x, b)) &= v(x_0, x_1) + v(x_1, x_2) + \cdots + v(x_{k-1}, x_k) \\ &\geqslant (H_{x_0} - H_{x_1}) + (H_{x_1} - H_{x_2}) + \cdots + (H_{x_{k-1}} - H_{x_k}) \\ &= H_{x_0} - H_{x_k} = H_x - H_b = H_x \text{ (car } H_b = 0 \text{)} \end{aligned}$$

ALGORITHME A^*

Algorithme_Astar($\mathcal{G}$,o,d,v)

Données: Un graphe $\mathcal{G}$ valué (positif), un sommet origine (resp but) o (resp b) de $\mathcal{G}$, une fonction heuristique H cohérente

Résultat: Construit un chemin de valeur minimale de o à b

$d(s) = 0$

$S \leftarrow \{s\}$

`// ensemble de sommets (chemins) dont la valeur minimale` d `est établie`

while il existe $(x, y) \in \mathcal{A}$ **avec** $x \in S$ **et** $y \notin S$

`Choisir` $y \notin S$ minimisant

$$\min_{x \in S, (x,y) \in \mathcal{A}} d(x) + v(x, y) + \mathbf{H_y}$$

$S \leftarrow S \cup \{y\}$

$d(y) = d(x) + v(x, y)$

PREUVE DE L'ALGORITHME A^*

Adaptation de la preuve de Dijkstra

$$\hat{d}(x) = d(x) + H_x$$

où $d(x)$ est le poids optimal actuellement connu pour un chemin menant de o à x, et H est une heuristique cohérente.
Nous remarquons que pour un sommet examiné x minimal (au sens de $\hat{d}$), nous avons pour tout successeur y de x :

$$\begin{aligned}
\hat{d}(y) &\stackrel{def}{=} d(y) + H_y \\
&= \min_x \{d(x) + v(x,y) + H_y\} \\
&= d(x^*) + v(x^*,y) + H_y = d(x^*) + H_{x^*} + v(x^*,y) - H_{x^*} + H_y \\
&= \hat{d}(x^*) + [\mathbf{v(x^*,y) - H_{x^*} + H_y}]
\end{aligned}$$

Ainsi, l'algorithme A* correspond exactement à l'algorithme de Dijkstra avec la repondération suivante :

$$\hat{v}(x,y) = v(x,y) - H_x + H_y$$

Merci à Pierre-Édouard Portier

Pour conclure la preuve

Soit un chemin de o à b dans le graphe revalué $\hat{\mathcal{G}} = (\mathcal{X}, \mathcal{A}, \hat{\mathcal{V}})$

$$c(o,b) =< o = x_0, x_1, \cdots, x_k = b >$$

$$\begin{aligned}
\hat{v}(c(o,b)) &= \hat{v}(x_0,x_1) + \hat{v}(x_1,x_2) + \cdots + \hat{v}(x_{k-1},x_k) \\
&= v(x_0,x_1) + (H_{x_1} - H_{x_0}) + v(x_1,x_2) + (H_{x_2} - H_{x_1}) + \cdots + v(x_{k-1},x_k)(H \\
&= v(x_0,x_1) + v(x_1,x_2) + \cdots + v(x_k,x_{k-1}) + H_{x_k} - H_{x_0} \\
&= v(c(o,b)) + H_b - H_o = \mathbf{v(c(o,b)) - H_o}
\end{aligned}$$

en conséquence

$$\min_c \hat{v}(c(o,b)) = \min_c v(c(o,b)) - H_o$$

Un chemin de valeur minimale de o à b dans $\hat{\mathcal{G}}$ est également un chemin de valeur minimale dans $\mathcal{G}$

A Formal Basis for the Heuristic Determination of Minimum Cost Paths

PETER E. HART, MEMBER, IEEE, NILS J. NILSSON, MEMBER, IEEE, AND BERTRAM RAPHAEL

Abstract—**Although the problem of determining the minimum cost path through a graph arises naturally in a number of interesting applications, there has been no underlying theory to guide the development of efficient search procedures. Moreover, there is no adequate conceptual framework within which the various ad hoc search strategies proposed to date can be compared. This paper describes how heuristic information from the problem domain can be incorporated into a formal mathematical theory of graph searching and demonstrates an optimality property of a class of search strategies.**

I. Introduction

A. The Problem of Finding Paths Through Graphs

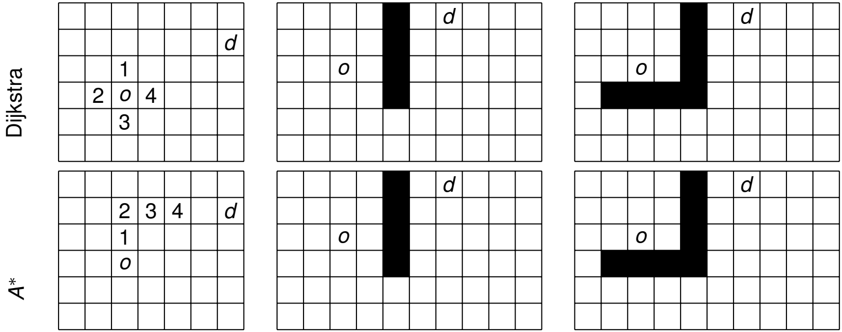
MANY PROBLEMS of engineering and scientific importance can be related to the general problem of finding a path through a graph. Examples of such problems include routing of telephone traffic, navigation through a maze, layout of printed circuit boards, and mechanical theorem-proving and problem-solving. These problems have usually been approached in one of two ways, which we shall call the *mathematical approach* and the *heuristic approach.*

1) The mathematical approach typically deals with the properties of abstract graphs and with algorithms that prescribe an orderly examination of nodes of a graph to establish a minimum cost path. For example, Pollock and Wiebenson[1] review several algorithms which are guaranteed to find such a path for any graph. Busacker and Saaty[2] also discuss several algorithms, one of which uses the concept of dynamic programming.[3] The mathematical approach is generally more concerned with the ultimate achievement of solutions than it is with the computational feasibility of the algorithms developed.

2) The heuristic approach typically uses special knowledge about the domain of the problem being represented by a graph to improve the computational efficiency of solutions to particular graph-searching problems. For example, Gelernter's[4] program used Euclidean diagrams to direct the search for geometric proofs. Samuel[5] and others have used ad hoc characteristics of particular games to reduce

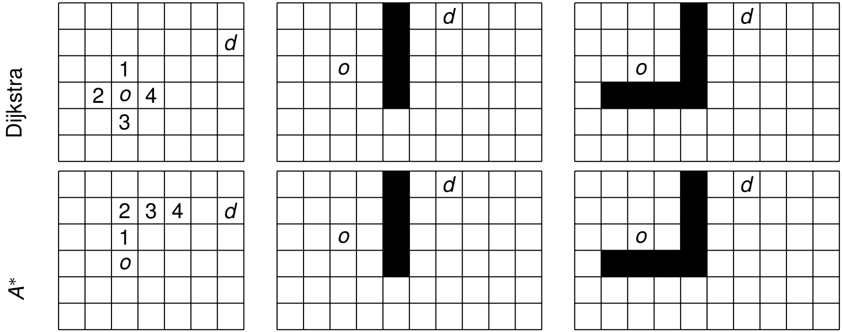
Manuscript received November 24, 1967.
The authors are with the Artificial Intelligence Group of the Applied Physics Laboratory, Stanford Research Institute, Menlo Park, Calif.

Sur la grille



Quelles seraient des heuristiques possibles ?

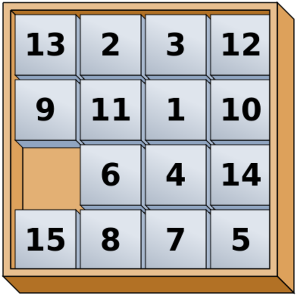
SUR LA GRILLE

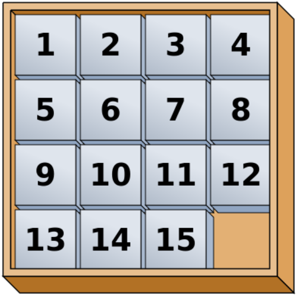


Quelles seraient des heuristiques possibles ?

$H_x = distance(x, d)$ avec une distance (Manhattan, ou euclidienne, ...)

Un peu de Taquin





1. $H^1_{configuration} = 15 -$ nombre de pièces bien placées.
2. $H^2_{configuration} = \sum_{i \in \text{pièces}}$ distance entre i et sa bonne position.

Ces heuristiques sont-elles admissibles ?
Peut-on comparer ces heuristiques ?

UGA Université Grenoble Alpes

Échelles de mots

Étant donnés 2 mots de même longueur, trouver une séquence de mots allant de l'un à l'autre en ne changeant qu'une seule lettre à la fois.
De bleu à noir
bleu -> blés -> bues -> buis -> luis -> lois -> loir -> noir
De Jean à Marc
jean -> jeun -> jeux -> feux -> feus -> fers -> fars -> mars -> marc.


Doublets de Lewis Carroll, repris par Knuth

structure de graphe : quelles heuristiques pourraient être utilisées ?

BIBLIOGRAPHIE

Ouvrages historiques

- Claude Berge, (1958). Théorie des graphes et ses applications. Paris : Dunod.
- Michel Gondran et Michel Minoux (1993) Graphes et algorithmes, Paris, Eyrolles
- J.A. Bondy U.S.R Murty, (2008). Graph Theory. Springer

Théorie des graphes

- Olivier Cogis et Claudine Schwartz, Théorie des graphes : problèmes, théorèmes, algorithmes, Paris, Cassini, 2018
- Alain Bretto, Alain Faisant et François Hennecart, (2022) Elements of Graph Theory : From Basic Concepts to Modern Developments, European Mathematical Society Press,
-

Algorithmique pour les graphes

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein (2010) Algorithmique - 3ème édition, Dunod (chap 22–27)
- Robert Sedgewick et Kevin Wayne (2011) Algorithms Addison-Wesley (chap 4)
- Jon Kleinberg et Eva Tardos (2013) Algorithm Design : Pearson New International Edition (chap 16–26)