Résolution du taquin par utilisation d'heuristiques

Pierre Vigier

Lycée Louis-le-Grand

18 août 2014

Introduction

Définition (Problèmes P)

Les problèmes P constituent la classe des problèmes polynomiaux c'est-à-dire ceux pouvant se résoudre en un temps polynomial.

Définition (Problèmes NP)

Les problèmes NP sont les problèmes qui possèdent un algorithme proposant une solution et dont on peut vérifier qu'une solution convient en temps polynomial.

Définition (Problème NP-complets)

Les problèmes NP-complets sont les problèmes les plus difficiles de la classe NP.

Les problèmes NP-complets ne possèdent a priori pas d'algorithme de résolution en temps polynomial. Pourtant des problèmes importants sont NP-complets :

- l'optimisation de la VLSI ;
- le problème du voyageur du commerce ;
- le problème SAT.

Nous allons voir comment accélérer la résolution de ces problèmes par l'utilisation d'heuristiques. Pour cela, nous allons nous appuyer sur l'exemple de la résolution du taquin qui est un problème NP-complet.

Plan

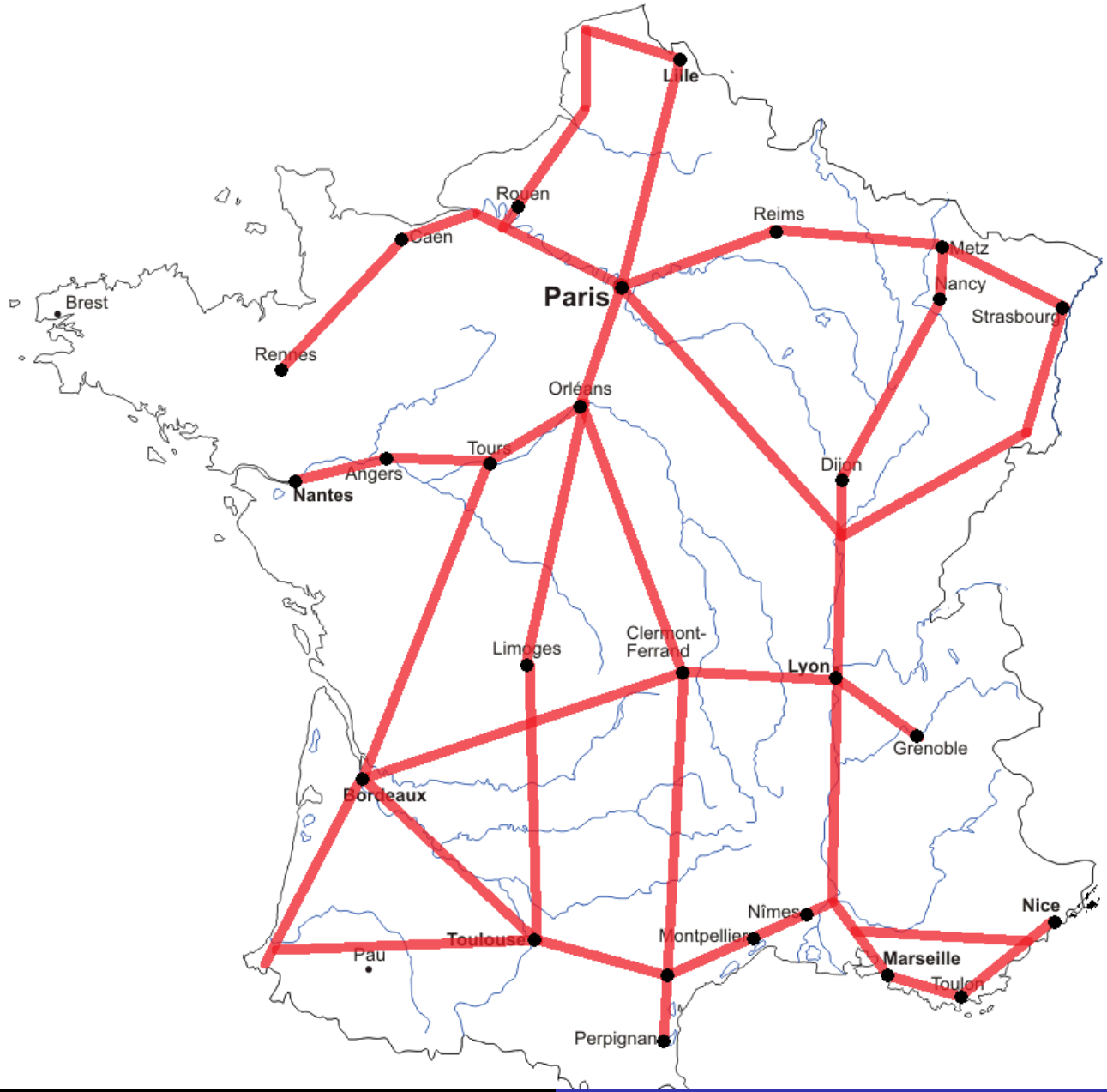
Une fonction classique d'exploration d'un graphe :

```
fonction explorer(état initial)
  initialiser la frontière avec l'état initial
  tant que la frontière n'est pas vide
    choisir un noeud dans la frontière et l'enlever
    si le noeud est l'état final alors
      retourner la solution correspondante
    pour tous noeud accessible depuis ce noeud
      si il n'est ni dans la frontière ni dans les
          noeuds déjà explorés alors
        on l'ajoute à la frontière
```

En choisissant de manière arbitraire les nœuds à développer, on doit parcourir tous les états possibles pour être sûr de trouver la solution optimale.

Tout le parcours est déterminé par la manière de choisir les nœuds dans la frontière. Les algorithmes de type best-first utilisent une fonction f qui à un nœud associe un réel et choisissent à chaque fois le nœud ayant la plus faible valeur dans la frontière.

$f(n)$	Parcours
$depth(n)$	en largeur
$-depth(n)$	en profondeur
$g(n)$	à coût uniforme


Lille
Rouen
Caen
Reims
Metz
Nancy
Strasbourg
Paris
Brest
Rennes
Orléans
Tours
Angers
Nantes
Dijon
Limoges
Clermont-
Ferrand
Lyon
Grenoble
Bordeaux
Nîmes
Nice
Montpellier
Toulouse
Pau
Marseille
Toulon
Perpignan

Plan

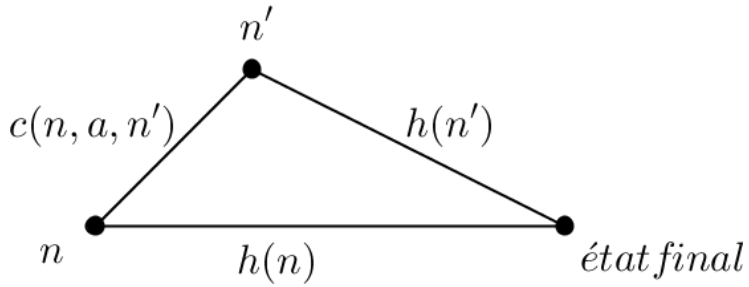
Posons $g(n)$ le coût pour aller de l'état initial au nœud n et $h(n)$ l'estimation du coût du chemin le moins coûteux de n à l'état final. Posons alors :

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

f(n) est alors le coût estimé de la solution la moins coûteuse.

Proposition

A* est optimal si h est consistante c'est-à-dire si pour chaque nœud n et chacun de ses successeurs n' et pour toute action a permettant d'aller de n à n' de coût $c(n, a, n')$, on a $h(n) \leq c(n, a, n') + h(n')$.



Preuve : Soit h consistante.

- Pour tout nœud n et n' un de ses successeurs et a une action permettant d'aller de n à n', on a $f(n) = g(n) + h(n) \leq g(n) + c(n, a, n') + h(n') = g(n') + h(n') = f(n')$ d'où f est croissante sur un chemin.
- Supposons qu'il existe un nœud n développé par A* tel que le chemin optimal vers celui-ci n'a pas été trouvé. Il existe alors n' appartenant à la frontière qui est sur un chemin optimal entre l'état initial et n. D'où d'après le point précédent $f(n') \leq f(n)$. D'où n' aurait été développé avant n. Contradiction.

Par conséquent, le premier nœud atteignant l'état final a été atteint via un chemin optimal. Donc la première solution trouvée est optimale.

Plan

Définition (Taquin)

Le taquin à $np - 1$ pièces se compose d'un plateau de $n \times p$ cases dont $np - 1$ sont occupées par des pièces numérotées et une est vide. Les pièces adjacentes à la case vide peuvent échanger leur place avec celle-ci. Le but du jeu est de passer d'une configuration initiale donnée à une configuration finale.

Dans toute la suite, nous utiliserons le taquin à 8 pièces dans nos exemples.

3	7	5
6		2
4	8	1

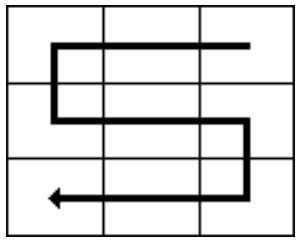
État initial

3	2	1
4	5	6
	8	7

État final

Plan

Associons à chaque configuration une permutation de S_{np-1} pour cela, numérotons les cases non vides dans cet ordre :



Par exemple à la configuration ci-dessous est associée la permutation $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 7 & 3 & 6 & 2 & 1 & 8 & 4 \end{pmatrix}$.

3	7	5
6		2
4	8	1

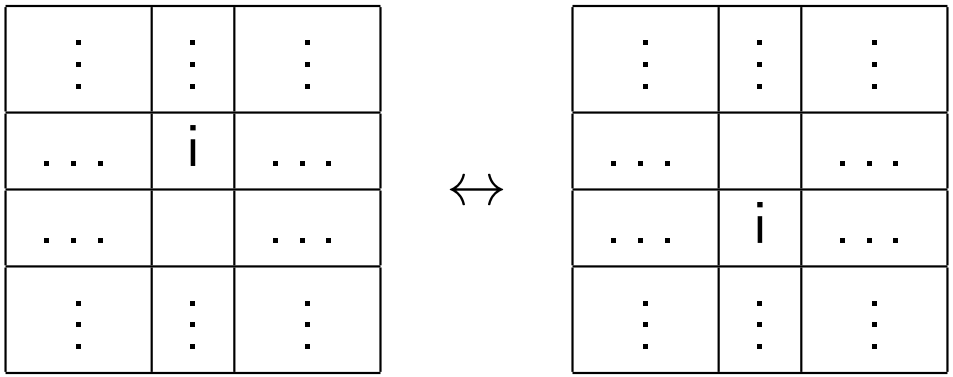
On remarque qu'à la configuration finale est associée l'identité et que deux configurations ont la même permutation si et seulement si on peut passer de l'une à l'autre en déplaçant la case vide le long de ce parcours.

Théorème

On peut passer d'une configuration à une autre si et seulement si les permutations associées ont des signatures égales.

Preuve : ($\Rightarrow$) Supposons qu'il existe une suite de mouvements allant d'une configuration à une autre. Remarquons que les mouvements ne changent pas la signature :

- les mouvements horizontaux ne changent pas la permutation donc a fortiori pas sa signature ;
- notons (i, j) les coordonnées de la case vide et regardons comment change la permutation lors des deux mouvement verticaux possibles.



On compose la permutation par un cycle de longueur $2(j-1)+1$ ou $2(p-j-1)+1$ selon le sens du parcours. D'où la signature est invariante.

Donc la suite de mouvements laisse la signature invariante d'où les deux configurations ont la même signature.

($\Rightarrow$) Remarquons qu'on peut obtenir les permutations $\begin{pmatrix} p-1 & p & p+1 \end{pmatrix}$ et $\begin{pmatrix} 1 & 2 & \dots & 2p-1 \end{pmatrix}$.

Or en composant $\begin{pmatrix} p-1 & p & p+1 \end{pmatrix}$ par des puissances de $\begin{pmatrix} 1 & 2 & \dots & 2p-1 \end{pmatrix}$, on obtient $\begin{pmatrix} i & i+1 & i+2 \end{pmatrix}$ pour tout $i \in [\![0; 2p-3]\!]$. Et en réopérant cette opération à chaque coude, on obtient les 3-cycles $\begin{pmatrix} i & i+1 & i+2 \end{pmatrix}$ pour tout $i \in [\![0; np-3]\!]$. Et le théorème suivant permet de conclure :

Théorème

Les 3-cycles $\begin{pmatrix} i & i+1 & i+2 \end{pmatrix}$ pour tout $i \in [\![0; n-2]\!]$ génèrent A_n.

Plan

Définition (Problème relaxé)

Un problème dont les restrictions sur les actions sont moindres est un problème relaxé.

- Les solutions optimales d'un problème sont aussi des solutions du problème relaxé car elles sont réalisables mais il peut exister de meilleurs solutions.
- La fonction qui renvoie le coût de la solution optimale d'un problème relaxé est consistante donc elle fait une bonne heuristique. Nous allons donc rechercher les problèmes relaxés qui ont de telles fonctions simples à calculer.

Règles du taquin

Une pièce peut se déplacer de la case A vers la case B si A et B sont adjacentes et B est vide.

On en déduit trois problèmes relaxés :

1. Une pièce peut se déplacer de la case A vers la case B.
2. Une pièce peut se déplacer de la case A vers la case B si A et B sont adjacentes.
3. Une pièce peut se déplacer de la case A vers la case B si B est vide.

On peut alors associer une heuristique à chacun de ses problèmes relaxés :

1. $h1$ est le nombre de différences entre l'état initial et l'état final ;
2. $h2$ est la somme des distances de Manhattan que doivent parcourir chaque pièce pour aller de leur position initiale à leur position finale ;
3. pour calculer $h3$, on exécute un algorithme simple qui donne la solution optimale puis on renvoie sa longueur.

```
tant que la position finale n'est pas atteinte
  si la case vide n'est pas à sa position finale
    on l'échange avec la pièce qui devrait être à
        sa place
  sinon
    on l'échange avec une pièce mal placée
```

Plan

Il existe deux moyens de comparer des heuristiques :

- On peut donner une preuve mathématique qu'une heuristique en majore une autre.
- On peut associer mesurer son efficacité sur des exemples.

d	A*(h1)	A*(h2)	A*(h3)
2	6	6	6
4	11	10	10
6	17	14	16
8	33	20	30
10	69	32	62
12	168	57	143
14	395	105	326
16	946	210	749
18	2205	380	1815
20	5224	659	3991
22	11821	1194	9119
24	26118	2232	19471

Définition (Facteur de branchement effectif)

Soit d la profondeur d'une solution et k le nombre de nœuds explorés, on définit b le facteur de branchement effectif tel que $1 + b + \ldots + b^d = k$.

d	A*(h1)	A*(h2)	A*(h3)	d	A*(h1)	A*(h2)	A*(h3)
2	1.73	1.73	1.73	14	1.40	1.24	1.38
4	1.36	1.32	1.34	16	1.42	1.27	1.40
6	1.29	1.22	1.27	18	1.44	1.28	1.42
8	1.30	1.19	1.28	20	1.45	1.28	1.42
10	1.33	1.20	1.31	22	1.45	1.29	1.43
12	1.38	1.22	1.35	24	1.46	1.30	1.44

Si on note h une heuristique et k le coût réel de la solution optimale d'un nœud n à l'état final, on définit :

Définition (Rapport d'une heuristique)

Si N est une variable aléatoire discrète suivant une loi uniforme sur les nœuds d'un graphe fini, le rapport de h est $r = \mathbb{E}(\frac{h(N)}{k(N)})$.

Lemme

$$\sum_{d_a=0}^{d_f-1} b^{\left\lceil \frac{d_f(1-r)+2rd_a}{1+r} - 1 \right\rceil - d_a} + 1 \leq n \leq \sum_{d_a=0}^{d_f} b^{\left\lfloor \frac{d_f(1-r)+2rd_a}{1+r} \right\rfloor - d_a}$$

Corollaire

$$b_{eff} \leq b^{\frac{1-r}{1+r}}$$

- Conditions sur l'heuristique pour que le facteur d'embranchement converge.
- Conditions et explication de l'exponentielle lorsqu'on fait l'importance de l'heuristique.
- A-t-on des relations d'ordre ou d'équivalence entre les mesures ?
- Peut-on généraliser les résultats obtenus sur des arbres à embranchement fixe ?

Conclusion

- Sur un exemple aussi simple que le taquin, nous avons vu que l'utilisation d'heuristiques permettait d'accélérer la recherche de solution optimale.
- La méthode des problèmes relaxés pour générer des heuristiques est générale et transposable.
- L'algorithme A* et les heuristiques sont très utilisés dans la recherche du plus court chemin en général et plus particulièrement en intelligence artificielle.

Bibliographie

- Stuart Russel, Peter Norvig. Intelligence artificielle. Pearson, 2010, 1200p.
- Michel Coste. Jeu de taquin et générateurs du groupe alterné. [en ligne]. Université de Rennes 1. Disponible sur : `http://agreg-maths.univ-rennes1.fr/documentation/docs/taquin.pdf`.