TD - Introduction à l'Intelligence Artificielle n° 5
(Correction)

# Formalisation de problèmes par graphes d'états

## Exercice 1 Le problème des cruches

On dispose de deux cruches (non graduées) dont on ne connait que les capacités respectives $P$ et $G$ (en litres) et d'une fontaine où l'on peut obtenir le l'eau à volonté. Le problème consiste à obtenir une quantité précise de $C$ litres dans l'une des deux cruches. Par exemple, on considère le cas où $P = 5$, $G = 7$, et $C = 4$.

1. Formaliser ce problème en terme de problème de recherche dans un graphe d'états. Spécifiez formellement les états et les opérateurs de changement d'état. Ceux-ci doivent être décrits par leurs conditions d'application et leurs effets.

**Correction :**
**Caractérisation de l'espace d'états**
Dans ce problème un état est complètement caractérisé par la description du volume d'eau contenu dans chacune des cruches. Comme les capacités des cruches correspondent à des entiers et que l'état objectif se caractérise aussi par des entier, on peut ici se limiter à un couple d'entiers.

$$\mathcal{E} = \{(p, g) \in 0..5 \times 0..7\}$$

où p et g représentent respectivement les contenus de la petite et de la grande cruche.
**Remarque :** tous les couples ne correspondent pas forcément à des états atteignables, mais cet espace couvre bien tous les états possibles.

**Caractérisation des opérateurs de changement d'état**
On peut ici vider, remplir chacune des cruches ou encore transvaser le contenu d'une cruche dans une autre. Dans le cas du tranvasement, deux cas sont à distinguer :

- celui du transvasement intégral
- celui du transvasement partiel, lorsque le contenu à transvaser dépasse la capacité de la cruche dans laquelle on transvase l'eau. Dans ce cas, il ne vaut pas perdre d'eau, sinon, on perdrait en même temps de l'information sur les capacités d'eau restantes. On ne peut donc remplir la cruche dans laquelle on transvase que jusqu'à son maximum.

**Discussion :** on peut s'interroger sur l'opportunité d'introduire des paramètres aux opérateurs. En effet remplir la petite cruche est très proche de remplir la petite cruche. Cependant, étant donné la dissymétrie existant entre les deux cruches, on sera ici obligé de commencer systématiquement (dans la description des précondition comme celles des effets) par distinguer le cas selon qu'il concerne la grande ou la petite cruche.

| Opérateur | Préconditions sur un état (p,g) | Effet |
|---|---|---|
| rp : remplirP | $p \neq 5$ | $(5, g)$ |
| rg : remplirG | $g \neq 7$ | $(p, 7)$ |
| vp : viderP | $p \neq 0$ | $(0, g)$ |
| vg : viderG | $g \neq 0$ | $(p, 0)$ |
| tpg : transvaserP>G | $p > 0 \wedge g < 7$ | $(max(0, p + g - 7), min(7, p + g))$ |
| tgp : transvaserG>P | $g > 0 \wedge p < 5$ | $(min(5, p + g), max(0, p + g - 5))$ |

2. Caractérisez la taille de l'espace d'états et le facteur de branchement de votre modélisation.

   **Correction :** Sensibiliser les étudiants au fait qu'il est bon d'avoir une idée de l'ordre de grandeur de la taille de l'espace d'états.

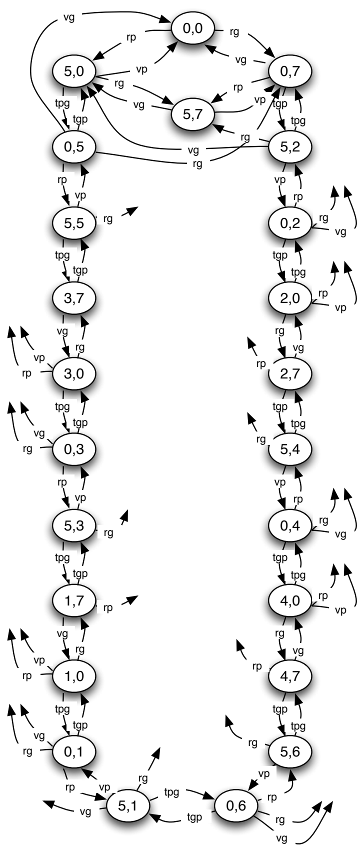
   Ici $|\mathcal{E}| = 6 * 8 = 48$
   Caractérisation du facteur moyen de branchement :

   - si $(0 < p < 5)$ et $(0 < g < 7)$ soit donc $4 * 6 = 24$ états potentiels, les 6 opérateurs sont applicables
   - si $(p = 0$ ou $p = 5)$ et $(0 < g < 7)$ soit donc$2 * 6 = 12$ états potentiels, seuls 4 opérateurs sont applicables
   - si $(g = 0$ ou $g = 7)$ et $(0 < p < 5)$ soit donc $2 * 4 = 8$ états potentiels, seuls 4 opérateurs sont applicables
   - si $(p = 0$ et $g = 0)$ ou $(p = 5$ et $g = 7)$ soit donc 2 états potentiels, seuls 2 opérateurs sont applicables
   - si $(p = 0$ et $g = 7)$ ou $(p = 5$ et $g = 0)$ soit donc 2 états potentiels, seuls 3 opérateurs sont applicables

   En moyenne $b = (2 * 2 + 2 * 3 + 20 * 4 + 24 * 6)/48 = 4,875$

3. Construisez le graphe d'état complet pour ce problème et commentez.

   **Correction :** NB : dans quasiment n'importe quel état, il est possible en deux transitions de vider ou remplir complètement les deux cruches. Afin de ne pas trop alourdir le dessin, toutes les transitions de ce type ne seront pas forcément indiquées.




   On remarque que en pratique aucun des états tels que $(0 < p < 5)$ et $(0 < g < 7)$ n'est atteint en pratique. Le véritable facteur de branchement est donc : $3,75$.

**Exercice 2 Le problème des missionnaires et cannibales**

$N$ missionnaires et $N$ cannibales se trouvent au bord d'une rivière. Ils sont tous d'accord pour aller sur l'autre rive, mais les missionnaires ne se sentent pas très rassurés à l'idée de voyager en présence de cannibales. Ils veulent donc arranger la traversée de la rivière de manière à ce que à chaque endroit, le nombre de missionnaires sur l'une ou l'autre rive ne soit jamais inférieur au nombre de cannibales. De plus, le seul bateau disponible ne peut prendre plus de $k$ personnes à son bord en même temps. Comment tout le monde peut-il traverser la rivière sans que les missionnaires risquent d'être mangés ?

1. Formaliser ce problème en tant que problème de recherche dans un espace d'états. Donner l'état initial, l'état final, et préciser les opérateurs de changement d'états, leurs conditions d'application et leurs effets.

**Correction :** Au delà de son appartenance au folklore de l'IA, le principal intérêt de ce problème est de montrer que, pour un même problème, différentes formalisations sont possibles et que cela peut avoir une influence sur la taille des espaces d'états résultants (même si sur cet exemple le graphe d'états reste de taille constante).

Il est précisé dans l'énoncé que l'on ne doit pas considérer les individus pris un par un mais plutôt le nombre de missionnaires et de cannibales sur les rives et le bateau. Dire tout de même un mot sur le fait qu'une telle représentation induirait de nombreuse symétries, une explosion combinatoire de ta taille de l'espace d'états et du nombre de solutions (en fait toutes les permutations d'individus possibles, dans chaque ensemble de missionnaires et cannibales)

**Formalisation 1 :**
On peut représenter un état par un septuplet $(Mg, Cg, Mb, Cb, Md, Cd, pb)$ où :

- $Mg$ représente de le nombre de missionnaires sur la rive gauche, avec $0 \leq Mb \leq N$.
- $Cg$ représente de le nombre de cannibales sur la rive gauche, avec $0 \leq Cb \leq N$.
- $Mb$ représente de le nombre de missionnaires dans le bateau, avec $0 \leq Mb \leq k$.
- $Cb$ représente de le nombre de cannibales dans le bateau, avec $0 \leq Cb \leq k$.
- $Md$ représente de le nombre de missionnaires sur la rive droite, avec $0 \leq Md \leq N$.
- $Cd$ représente de le nombre de cannibales sur la rive droite, avec $0 \leq Cd \leq N$.
- $pb$ représente la position du bateau, avec $pb \in \{gauche, droite\}$.

$\mathcal{E} = \{(Mg, Cg, Mb, Cb, Md, Cd, pb)/0 \leq Mg, Md \leq N, 0 \leq Cg, Cd \leq N$ et $0 \leq Mb, Cb \leq k$
et $pb \in \{gauche, droite\}\}$

- Etat initial : $(N, N, 0, 0, 0, 0, gauche)$
- Etat terminal : $(0, 0, 0, 0, N, N, \_)$

Taille de l'espace d'états : $|\mathcal{E}| = (N+1)^6 * 2$

**Les opérateurs :**
Les trois opérations de base correspondent à embarquer, débarquer et traverser. Il faut cependant les dupliquer en fonction de la rive où l'on se trouve car les préconditions et effets ne sont pas les mêmes.

On peut se demander s'il n'est pas possible de paramètrer ces opérateurs en fonction de la rive de départ... mais on se rend compte qu'en fait, cela oblige à distinguer des le début quelle est la valeur du paramètre désignant la rive et a donner deux expressions distinctes des préconditons et postconditions selon les cas, ce qui au final est plus lourd que de rajouter les opérateurs symétriques.

On peut aussi s'interroger sur ce qui se passe vraiment lors d'un embarquement/débarquement... Est-ce qu'un cannibale ne va pas profiter du moment où un missionnaire descend/monte pour en croquer un autre ? Pour simplifier on peut supposer que tout le monde embarque et débarque en même temps. On supposera aussi que lors d'un débarquement, tous les passagers à bord du bateau débarquent, même si l'un d'entre eux doit remonter par la suite (sinon on retombe dans le cas précédent).

Enfin il est clair que l'opérateur modélisant l'embarquement nécessite des paramètres, pour préciser le nombre de missionnaires et de cannibales qui montent à bord du bateau.

- **embarquer_g**$(m, c)$ : $m$ missionnaires et $c$ cannibales embarquent dans le bateau. On suppose qu'il n'y a personne dans le bateau puisqu'à chaque débarquement, tout le monde descend.

  **preconditions :**

— $pb = gauche$ (le bateau doit être a gauche)
— $(0 \leq m \leq Mg) \wedge (0 \leq c \leq Cg)$ (on ne peut embarquer plus de personne que celles qui sont sur la rive)
— $(1 \leq m + c \leq k)$ (au moins une personne et pas plus que la capacité du bateau)
— $(m = 0 \vee m \geq c)$ (conditions de survie sur le bateau)
— $(Mg - m = 0 \vee Mg - m \geq Cg - c)$ (conditions de survie sur la rive gauche)

**postconditions :** on passe alors dans l'état $(Mg - m, Cg - c, m, c, gauche)$.

- **traverser_g_d** : traverser la rivière de gauche à droite

  **preconditions :**
  — $pb = gauche$ le bateau doit être a gauche
  — $Mb + Cb \geq 1$ le bateau de doit pas être vide

  **postconditions :** on passe dans l'état $(Mg, Cg, Mb, Cb, Md, Cd, droite)$

- **debarquer_d** : les passagers du bateau débarquent sur la rive droite. On suppose qu'il n'y a personne dans le bateau.

  **preconditions :**
  — $pb = droite$ le bateau doit être a droite
  — $Mb + Cb \geq 1$ le bateau de doit pas être vide
  — $(Md + Mb = 0) \vee (Md + Mb \geq Cd + Cb)$ conditions de survie sur la rive droite après débarquement

  **postconditions :** on passe dans l'état $(Mg, Cg, 0, 0, Mb + Md, Cb + Cd, droite)$

- il convient d'ajouter également les trois opérateurs symétriques

**Remarque :** Les étudiants ont tendance à se planter sur la condition de survie en oubliant le cas ou il n'y a aucun missionnaire.

**Formalisation 2 :**
On peut remarquer dans la représentation précédente qu'à chaque instant on a

$$Mg + Mb + Md = N \text{ et } Cg + Cb + Cd = N$$

On peut donc alléger la représentation des états en supprimant deux valeurs (par exemple $Md$ et $Cd$) puisqu'elles peuvent se déduire à partir des deux autres. Il suffit donc de représenter un état par un quintuplet $(Mg, Cg, Mb, Cb, pb)$.

$\mathcal{E} = \{(Mg, Cg, Mb, Cb, pb)/0 \leq Mg \leq N, 0 \leq Cg \leq N \text{ et } 0 \leq Mb, Cb \leq k \text{ et } pb \in \{gauche, droite\}\}$

- Etat initial : $(N, N, 0, 0, gauche)$
- Etat terminal : $(0, 0, 0, 0, \_)$

Taille de l'espace d'états : $|\mathcal{E}| = (N + 1)^4 * 2$

A priori, à par la description des post conditions, seuls les opérateurs debarquer_d et embarquer_d sont affectés.

**debarquer_d :** appliqué sur un état $(Mg, Cg, Mb, Cb, droite)$

**preconditions :**
- $pb = droite$ (le bateau doit être a droite)
- $Mb + Cb \geq 1$ ((le bateau de doit pas être vide)
- $(N - Mg = 0 \vee N - Mg \geq N - Cg)$ (survie sur la rive droite après débarquement)

**postconditions :** on passe dans l'état $(Mg, Cg, 0, 0, droite)$

On peut de plus remarquer que on peut très bien arriver dans une situation où des passagers embarquent, traversent,... mais ne peuvent débarquer car les conditions de survie à droite ne sont pas satisfaites. Pour remédier à ce problème, on peut vérifier dès le moment de l'embarquement à gauche que les passagers pourront bien tous débarquer à droite, ce qui revient à rajouter à l'opérateur embarquer_g la précondition de débarquer_d.

**embarquer_g$(m, c)$ :** $m$ missionnaires et $c$ cannibales embarquent dans le bateau.

**preconditions :**

- $pb = droite$ (le bateau doit être a droite)
- $Mb + Cb \geq 1$ ((le bateau de doit pas être vide)
- $(N - Mg = 0 \vee N - Mg \geq N - Cg)$ (conditions de survie sur la rive droite après débarquement)

$(0 \leq m \leq Mg) \wedge (0 \leq c \leq Cg) \wedge (1 \leq m + c \leq k) \wedge pb = gauche$ et les conditions de survie à gauche : $(Mg - m = 0 \vee Mg - m \geq Cg - c)$, sur le bateau : $(m = 0 \vee m \geq c)$ et à droite : $(N - Mg + m = 0 \vee N - Mg + m \geq N - Cg + c)$.

**postconditions :** on passe dans l'état $(Mg - m, Cg - c, m, c, gauche)$

2. Tenir compte des contraintes du problème pour simplifier la représentation et donner de nouveau l'état initial, l'état final, et les opérateurs de changement d'états.

**Correction :**
**Formalisation 3 :**
On peut noter à ce stade, qu'il n'est pas très intéressant de faire embarquer des passager de les faire traverser, puis de les faire revenir, puis redébarquer. Pour éviter de tels scénarios, on peut tout simplement factoriser les trois opérateurs pour ne plus en faire qu'un seul.

Mais dans ce cas, on passe systématiquement d'un état où il n'y a personne sur le bateau à un autre état où il n'y a personne sur le bateau. On peut donc supprimer dans la représentation des états les deux paramètres indiquant le nombre de missionnaires et de cannibales dans le bateau et ne conserver que des triplets :

$$\mathcal{E} = \{(Mg, Cg, pb)/0 \leq Mg \leq N, 0 \leq Cg \leq N \text{ et } pb \in \{gauche, droite\}\}$$

- Etat initial : $(N, N, gauche)$
- Etat terminal : $(0, 0, \_)$

Taille de l'espace d'états : $|\mathcal{E}| = (N + 1)^2 * 2$

**Les opérateurs :**

**traverser_g_d($m, c$) :** $m$ missionnaires et $c$ cannibales traversent de droite à gauche

**preconditions :**

$pb = gauche$
$\wedge (0 \leq m \leq Mg) \wedge (0 \leq c \leq Cg) \wedge (1 \leq m + c \leq k)$
$\wedge ((Mg = 0) \vee (Mg \geq Cg))$ (conditions de survie à gauche)
$\wedge ((m = 0) \vee (m \geq c))$ (conditions de survie sur le bateau)
$\wedge ((N - Mg + m = 0) \vee (N - Mg + m \geq N - Cg + c))$. (conditions de survie à droite )

**postconditions :** on passe dans l'état $(Mg - m, Cg - c, droite)$

**traverser_d_g($m, c$) :** opérateur symétrique

On peut remarquer alors que dans chaque état, les conditions de survie doivent être simultanément être vérifiées à gauche comme à droite ce qui fait que la propriété suivante est toujours vérifiée :

$$(Mg = 0 \vee Mg \geq Cg) \wedge (N - Mg = 0 \vee N - Mg \geq N - Cg)$$

ce qui revient à dire que :

$$(Mg = 0 \vee Mg \geq Cg) \wedge (Mg = N \vee Mg \leq Cg)$$

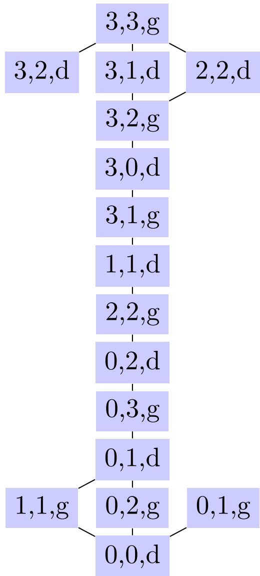
ce qui est équivalent à :

$$(Mg = 0) \vee (Mg = N) \vee (Mg = Cg)$$

On peut donc en déduire qu'il n'y a en fait au maximum que $2 * (N + 1) + N - 1 = 3N + 1$ états.

3. Construire le graphe d'état pour N= 3 et K=2.

**Correction :**




On peu noter un axe de symétrie dans le graphe entre les nœuds $(1,1,D)$ ET $(2,2,g)$.

## Exercice 3 Algorithme de recherche non informés

On condidère un problème abstrait formalisé en terme d'espace d'états et d'un ensemble d'opérateurs de changements d'états, décrits dans le tableau **??**, qui décrit, pour chaque état, les opérateurs applicables à partir de cet état et donne et les états atteignables après application de ces opérateurs.

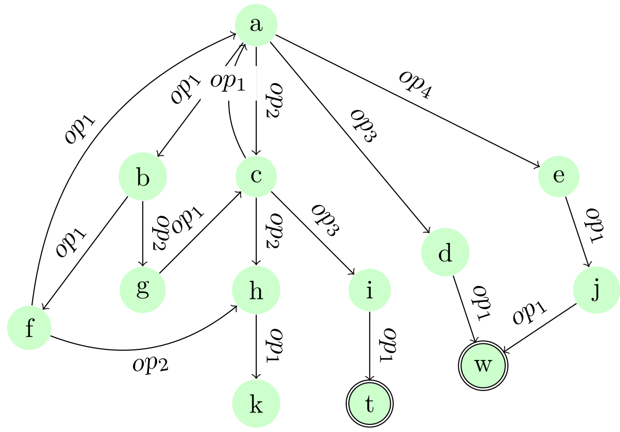
Le problème posé consiste à trouver une séquence d'opérateurs permettant de passer de l'état initial (état $a$) à l'un des états terminaux ($w$ et $t$).

| **Etat** | **Operateur** | **Successeur** |
|---|---|---|
| $a$ | $op_1$ | $b$ |
| | $op_2$ | $c$ |
| | $op_3$ | $d$ |
| | $op_4$ | $e$ |
| $b$ | $op_1$ | $f$ |
| | $op_2$ | $g$ |
| $c$ | $op_1$ | $a$ |
| | $op_2$ | $h$ |
| | $op_3$ | $i$ |
| $d$ | $op_1$ | $w$ |
| $e$ | $op_1$ | $j$ |
| $f$ | $op_1$ | $a$ |
| | $op_2$ | $h$ |
| $g$ | $op_1$ | $c$ |
| $h$ | $op_2$ | $k$ |
| $i$ | $op_2$ | $t$ |
| $j$ | $op_1$ | $w$ |

TABLE 1 – Operateurs de changement d'états

1. Dessiner le graphe d'états. Cette opération est-elle réalisable dans tous les cas ?

**Correction :**




Naturellement, il n'est envisageable de dessiner le graphe d'états que si le nombre d'états et d'opérateurs est fini.

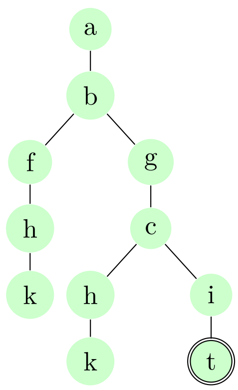
2. Trouver tous les chemins solution et leur coût (i.e. la longueur des chemins solution).

   **Correction :**
   - a-d-w : 2
   - a-e-j-w : 3
   - a-c-i-t : 3
   - a-b-g-c-i-t : 5
   - et tous ceux qui contiennent des boucles

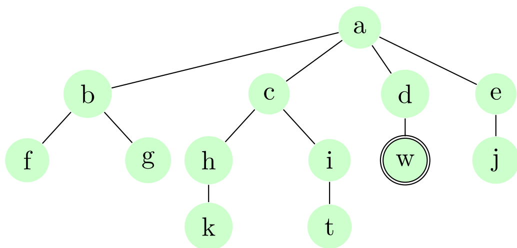
3. Parmi les algorithmes de recherche vus en cours, quels sont ceux qui sont applicables sur le problème ? Faire tourner le plus simple d'entre eux. En cas de doute, on développera les fils dans l'ordre où ils apparaissent dans le tableau **??**.

   **Correction :** Bien insister se l'ordre de développement des nœuds, sinon les résultats peuventvarier.

   **recherche en profondeur** : convient seulement s'il est capable de détecter (et d'éviter) les boucles.
   Solution trouvée : a-b-g-c-i-t.
   Arbre exploré par cet algorithme :




   **recherche en largeur** : convient.
   Solution trouvée : a-d-w.

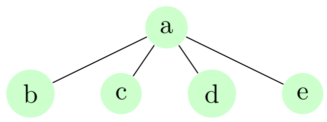



   NB : en fait pour la condition d'arrêt de la recherche en largeur, tout dépend du moment où est fait le test de détection d'un état terminal. Si on on a programmé l'algo pour le tester au moment où l'on fait apparaitre un nouveau nœud, on peut s'arrêter dès la profondeur 2. La même remarque vaut pour IDS.

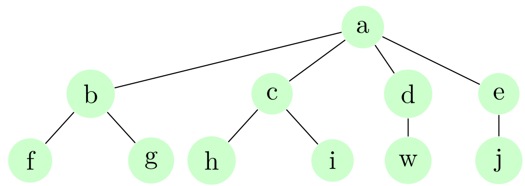
**recherche en profondeur itérative** :

Solution trouvée : a-d-w.

**Profmax = 1 :**




**Profmax = 2 :**




**Profmax = 3 :**