

Chap 2 Les Systèmes de Production et I.A.(SP) (Systèmes d'états)

Introduction

Comment résoudre un problème en IA? Pour répondre, il faut savoir :

- D'où l'on part, c.-à-d. l'énoncé du problème avec son état initial
- Où on veut aller, c.-à-d. le but à atteindre
- Comment on avance, c.-à-d. comment changer d'état
- Calculer un chemin de l'état initial à un état but, c.-à-d. faire un plan
- Essayer ce plan, pour le vérifier

Dans de tels systèmes d'IA, on a une entité centrale (base de données globale - un état), manipulée par des opérations, suivant une certaine stratégie : c'est un Système de règles de Production (SP) ou problèmes d'états.

Définition: Les composantes principales d'un SP (systèmes d'états) sont:

- a) La base de données globale : BDG est la structure de donnée utilisée par le SP (A ne pas confondre avec BD). Elle décrit les données (états/configurations) du problème d'une manière simple (vecteur, matrice..) ou de manière très complexes (structures indexées.....).
 - b) Opérations manipulant la base : Les règles de production qui agissent sur la BDG. Elles ont la forme: Précondition \Rightarrow Action . La pré condition doit être vérifiée par la BDG actuelle pour que la règle puisse être appliquée. L'action modifie la BDG. Ces règles permettent de changer d'état.
 - c) La stratégie de contrôle qui choisit la règle applicable à appliquer.
- La résolution d'un tel problème par un SP consiste à trouver un chemin (séquence de règles) de l'état initial au but.

Quelques propriétés

- Toutes les règles peuvent accéder à la BDG.
- Une règle n'appelle pas une autre règle. La communication se fait uniquement par l'intermédiaire de la BDG à travers sa modification.
- On distingue plusieurs types de SP. Ils diffèrent par le mécanisme de Contrôle, les propriétés des règles et de la structure de leur BDG.

Exemple du PB du Taquin à 9 cases

Soit par exemple le problème du Taquin à 9 cases, qui consiste en une matrice 3x3 dont une case contient x et les autres cases un chiffre de 1 à 8. La case X peut être échangée avec une case adjacente.

Etant dans un état initial, le problème consiste à transformer cet état pour atteindre un état but par une séquence de déplacements de la case x.

2	8	3
1	6	4
7	x	5

Etat Initial

?
⇒

1	2	3
8	X	4
7	6	5

Etat But

Pb de représentation en IA :

Pour résoudre ce problème par un SP, il faut spécifier ses 3 composantes:

- La BDG qui décrit l'ensemble des états du problème (l'ensemble des configurations). Cet ensemble constitue ce qu'on appelle l'espace du problème (pour cet exemple le nombre d'états est de l'ordre de $9!=362880$ qui est relativement petite). La représentation informatique d'un état de ce problème peut être une matrice 3x3.

- Les règles de production sont les déplacements de la case « x ». Une règle transforme un état en un autre (chaque règle ayant une pré condition)

Règle	Pré condition
* ↑ Déplacer la case x vers le haut	x n'est pas sur la ligne 1
* ↓ Déplacer vers le bas	x n'est pas sur la ligne 3
* → déplacer la case x vers la droite	x n'est pas sur la colonne 3
* ← déplacer vers la gauche.	x n'est pas sur la colonne 1

- Le but est d'atteindre la configuration but. (atteindre un état satisfaisant une certaine configuration (par exemple un état telle que la somme des éléments d'une certaine ligne doit ≤ 6 ou un état donné).

La solution à ce problème est une séquence de déplacements qui transforme un état en un autre jusqu'à ce que la condition but soit satisfaite.

Remarque :

Dans certain cas on exige que la solution soit de coût minimal. Pour cela on associe des coûts aux déplacements.

Procédure de base d'un SP.

L'algorithme de base d'un SP peut être la procédure non-déterministe:

Procédure PRODUCTION

- 1) *Donnée := BDG initiale* /*description de l'état initial */
- 2) *Jusqu'à ce que Donnée satisfasse la condition but* /* but atteint */
- 3) *Faire début*
- 4) **Sélectionner** une règle R de l'ensemble des règles applicables
à Donnée /* dont la pré condition est vérifiée */
- 5) *Donnée := R(Donnée)* /* appliquer R à donnée */
- 6) *Fin*

« Donnée » est une variable contenant initialement la BDG initiale et contiendra au fur et à mesure la BDG obtenue après l'application d'une règle R au niveau 5.

Le Contrôle. La procédure précédente est non-déterministe au niveau de la ligne 4 « **Sélectionner** ». La sélection à ce niveau, l'enregistrement des séquences des règles essayées ainsi que les BDG qu'elles produisent constituent ce qu'on appelle **la stratégie de Contrôle**. On peut avoir des stratégies qui utilisent d'autres informations supplémentaires pour produire des solutions optimales. On distingue 2 types de stratégie de contrôle:

1) Irrévocable: c.-à-d. des stratégies sans que l'on puisse remettre en question l'application d'une règle. Elle est utilisée uniquement lorsque la connaissance locale infaisable est disponible pour transformer la connaissance globale. A 1ère vue elle semble inexploitable, pour cela elle n'est utilisée que dans les cas où l'application d'une règle n'empêche pas l'utilisation ultérieure des autres règles. Elle n'est pas complète, (on peut se bloquer sans que le but soit atteint même lorsque celui-ci est atteignable).

2) Retour Arrière chronologique: Un point de retour arrière est établi lorsqu'une règle est sélectionnée parmi d'autres. Lors d'un blocage, on peut revenir au dernier point de choix pour essayer un autre choix jusqu'à atteindre le but ou il n'y est plus de choix. Si aucune information n'est disponible sur le choix de la règle à appliquer lorsque plusieurs peuvent l'être, la 1ère dans l'ordre d'écriture est choisie.

Exemple: Supposons que dans notre problème du Taquin à 9 cases, le nombre maximum d'application de la règle est 6. On suppose que l'ordre

des règles est arbitraire par exemple : gauche, haut, droite et bas. On revient en arrière chaque fois qu'on produit une description déjà produite, chaque fois que le nombre maximum (fixé au préalable) de règles est atteint, ou chaque fois qu'il n'y a plus de règles (de choix) à appliquer.

Pour l'implémentation, on peut utiliser une pile de choix. Dans cette stratégie on ne mémorise que le chemin de l'état initial à l'état actuel. On oublie les états intermédiaires lors de retours arrière.

TD : Produire l'arbre de recherche

Le processus de retour arrière est plus efficace si la sélection de la règle n'est pas arbitraire. On peut utiliser une méthode appelée méthode du gradient où on attribue une fonction à valeur ≤ 0 (0 - donne le nombre de cases mal classées) à chaque point de choix. On choisit la règle qui croît cette fonction. (voir exemple)

3) Recherche avec graphe. On explore dans cette stratégie l'effet de plusieurs règles simultanément. On utilise un arbre (un graphe en général) de recherche, d'où l'explosion de l'arbre puisqu'on garde tous les chemins. A chaque noeud, avant d'atteindre le but, seront produits tous les chemins possibles.

TD : Produire l'arbre de recherche

Notions de chaînage avant et chaînage arrière :

Lors de la résolution du problème précédent, nous avons travaillé en chaînage avant c à d allant de l'état initial vers l'état final.

On distingue 3 types de chaînage :

- Le chaînage avant (c'est-à-dire allant de l'état initial vers le but comme l'exemple précédent)
- Le chaînage arrière (c'est-à-dire allant du but vers l'état initial). A partir d'un but B on produit un sous but B' qui si une règle est appliquée à ce sous but B' elle nous produit le but B
- Chaînage bidirectionnel (combinaison des 2 chaînages)

Exemples de Problèmes

Exemple 1

Soit le problème du voyageur du commerce qui consiste à minimiser le chemin pour partir d'une ville A, visiter 4 autres villes B, C, D, E en ne passant qu'une seule fois et revenir à la ville de départ A. On a les distances séparant chaque 2 villes.

Exemple 2: Soit le problème d'analyseur syntaxique et soit une grammaire G. Le but est de voir si un mot W appartient au langage L généré par G.

Systèmes de Productions spécialisés.

- 1) **SP commutatifs :** On dit qu'un SP est commutatif s'il vérifie les 3 conditions suivantes :
 - a) Chaque règle des règles applicables à une BDG B est aussi applicable à toute BDG résultante de l'application d'une règle R' à B
 - b) Si le but est satisfait par B alors il est satisfait par R(B) pour toute règle R appliquée à B
 - c) La BDG obtenue en appliquant à B toute séquence de règles qui est appliquée à B ne change pas quelque soit les permutations effectuées sur la séquence.

Exemple : Soit un SP avec 3 règles R1, R2 et R3 qui sont applicables à une BDG EI. Ce SP est commutatif.

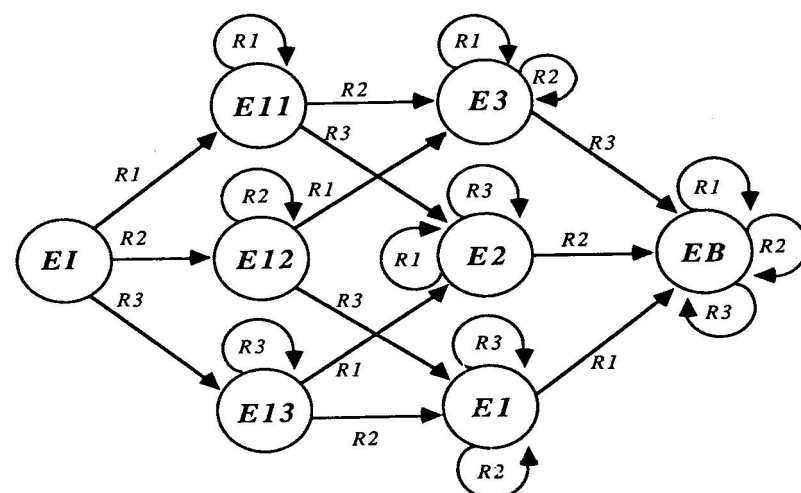


Fig. 1.8 Voies équivalentes dans un graphe.

Rmq : On peut utiliser la stratégie irrévocable pour les SP commutatifs.

2) SP décomposables : Dans certains problèmes il est plus commode et plus intéressant de décomposer le problème en sous problèmes. Pour cela on décompose la BDG en sous BDG qui peuvent être traitées séparément. Les règles peuvent être appliquées à ces s/BDG séparément ou en parallèle. Il faut aussi être capable de décomposer la condition d'arrêt. Les SP où on peut décomposer la BDG et le but sont appelés des SP décomposables.

Exercice : Modifier la procédure précédente pour l'adapter aux SP décomposables

Exemples de SP décomposables :

- 1) Découverte de structures chimiques organique (Dendral)
- 2) Intégration symbolique

Terminologie

- Etat : description du monde relativement au problème considéré
- Espace des états (ou espace de recherche) : ensemble de tous les états possibles
- Action : opération permettant de changer d'état
- Chemin : séquence d'états reliés par des actions
- Exploration : processus de calcul qui consiste à déterminer une séquence d'actions permettant d'atteindre une solution
- Stratégie : choix de la prochaine action à entreprendre dans le processus d'exploration