

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

Introduction

Comment résoudre un problème ? Pour répondre à cette question il faut savoir :

- D'où l'on part, c-a-d Enoncé du problème avec son état initial
- Ou on veut aller, c-a-d le but
- Comment on avance, c-a-d changer d'état
- Calculer un chemin de l'état initial au but, faire un plan
- Essayer ce plan pour le vérifier
-

Dans les systèmes d'IA, on peut souvent faire une séparation entre les composantes classiques de ce système dans un haut niveau. A savoir une entité centrale (base de données globale), manipulée par des opérations suivant une certaine stratégie. C'est ce qu'on appelle un Système de Production (SP).

Définition: Les composantes principales d'un SP sont:

- La base de donnée globale (BDG)
- Opérations manipulant la base: les règles de production
- La stratégie de contrôle qui dirige les opérations.

a) La BDG est la structure de donnée utilisée par le SP (A ne pas confondre avec BD). Elle décrit les données du problème d'une manière simple tel qu'un vecteur, une matrice ou de manière très complexes telles que des structures indexées.....

b) Les règles de production agissent sur la BDG. Elles ont la forme:

Précondition \Rightarrow Action . La pré condition doit être vérifiée par la BDG pour que la règle puisse être appliquée. L'action modifie la BDG.

c) La stratégie de contrôle choisit la règle applicable à appliquer.

La résolution du SP consiste à trouver un chemin de l'état initial au but.

Quelques propriétés

- Cette séparation permet de concevoir ces systèmes de manière modulaire.
- 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

On peut avoir plusieurs problèmes en IA de ce type. Par exemple des problèmes nécessitant la génération de séquences d'opérations. 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. Etant dans un état initial, le problème consiste à transformer cet état pour atteindre un état but par des déplacements de la case x avec une case adjacente.

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 est l'ensemble des états du problème c-a-d l'ensemble des configurations. Cet ensemble constitue ce qu'on appelle l'espace du problème (pour notre exemple sa taille de l'ordre de 9! est relativement petite 362880). La représentation d'un état peut être une matrice 3x3.

- Les règles de production sont les déplacements de la case « x ». Un déplacement transforme un état en un autre par les 4 actions suivantes:

- * ↑ Déplacer la case x vers le haut
- * ↓ Déplacer vers le bas
- * → déplacer la case x vers la droite
- * ← déplacer vers la gauche.

Chaque règle a une precondition qui doit être satisfaite pour qu'elle puisse être appliquée. Par exemple la 1ere règle (→) a comme precondition le fait que la case x ne soit sur la dernière colonne de la matrice.

- Le but est d'atteindre une configuration satisfaisant une certaine condition. Elle peut être par exemple: atteindre état satisfaisant une certaine configuration (par exemple un état telle que la somme des éléments d'une certaine ligne doit ≤ 6). La condition du but constitue la condition d'arrêt.

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. Dans certains 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 pour résoudre un SP tel que l'exemple précédent peut être la procédure non-déterministe suivante:

Procédure PRODUCTION

- 1) *Donnée := BDG initiale*
- 2) *Jusqu'à ce que Donnée satisfasse la condition but*
- 3) *Faire début*
- 4) **Sélectionner** une règle *R* de l'ensemble des règles applicables
 à *Donnée*
- 5) *Donnée := R(Donnée)*
- 6) **Fin**

« Donnée » est une variable initialement contenant 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:

- Irréversible: c-à-d des stratégies sans que l'on puisse en question l'application d'une règle.
- par essais successifs (ou par tentatives): c-à-d des stratégies avec la possibilité de revenir sur ce choix. On distingue 2 classes:
 - * Retour Arrière chronologique: Un point de retour arrière est établi lorsqu'une règle est sélectionnée parmi d'autres. On peut revenir à chaque fois que possible au dernier point de choix jusqu'à ce qu'il n'y ait aucun autre point de choix ou on atteint le but.
 - * Recherche avec graphe: On explore dans cette stratégie l'effet de plusieurs règles simultanément.

Exemples de stratégies de contrôle

1) Irrévocable: cette stratégie est utilisée uniquement lorsque la connaissance locale infaible est disponible pour transformer la connaissance globale. A 1ere 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. Cette stratégie n'est pas complète, (on peut se bloquer sans que le but soit atteint même lorsqu'il existe).

2) Retour Arrière chronologique: Cette stratégie est utilisée dans les problèmes non-deterministes. Lorsqu'une règle est sélectionnée et on n'aboutit pas à la solution (ou on se bloque par un échec ou par une limitation sur le nombre de séquence possible), les étapes intermédiaires sont oubliées et une autre règle est sélectionnée dans le dernier point de choix. Si aucune information n'est disponible sur le choix de la règle à appliquer lorsque plusieurs peuvent l'être, la 1ere 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 enfin 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 ou 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.

Recherche avec graphe. 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

Problème de Représentation:

Pour avoir une solution performante, il faut avoir une bonne représentation pour les états du problème, de bons déplacements et de bonnes conditions d'arrêts (de buts). Il n'y a pas de règles générales pour avoir une bonne représentation. Elle nécessite de l'expérience.

Chaînage avant et chaînage arrière :

Lors de la résolution du problème précédent, nous avons travaillé en change avant c a d allant de l'état initial vers l'état final. On peut distinguer 3 types de chaînage :

- Le chaînage avant (c'est-a-dire allant de l'état initial vers le but comme l'exemple précédent)
- Le chaînage arrière (c'est-a-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 Probleme

Soit le problème du voyageur du commerce qui consiste a 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 de l'ensemble des règles applicables à une BDG B est aussi applicable à toute BDG résultant de l'application d'une règle 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 sont appliquées 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.

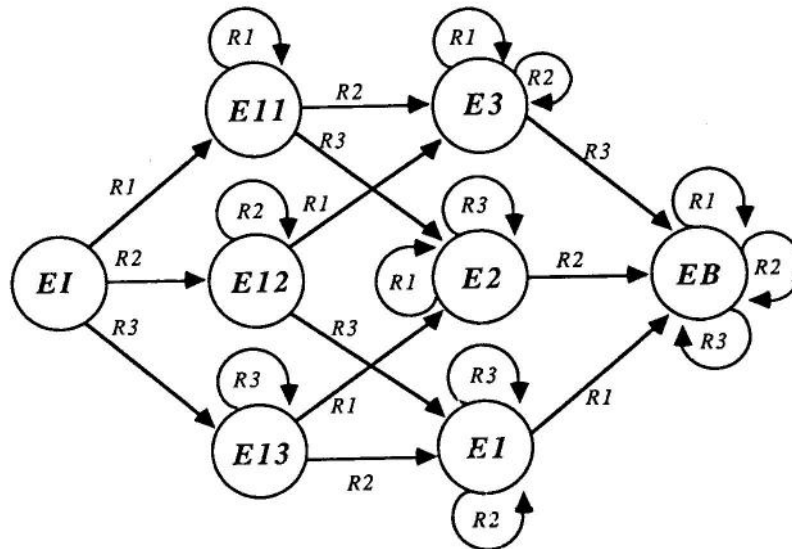


Fig. 1.8 Voies équivalentes dans un graphe.

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

b) 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 ou 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