

L. PIBOULEAU

S. DOMENECH

**Une procédure arborescente pour la séparation
de mélanges complexes dans l'industrie chimique.
Application à un train de colonnes de rectification**

RAIRO. Recherche opérationnelle, tome 19, n° 1 (1985), p. 35-56

http://www.numdam.org/item?id=RO_1985__19_1_35_0

© AFCET, 1985, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

**UNE PROCÉDURE ARBORESCENTE
POUR LA SÉPARATION
DE MÉLANGES COMPLEXES
DANS L'INDUSTRIE CHIMIQUE.
APPLICATION
A UN TRAIN DE COLONNES DE RECTIFICATION (*)**

par L. PIBOULEAU ⁽¹⁾ et S. DOMENECH ⁽¹⁾

Résumé. — *Le problème de la séparation optimale de mélanges chimiques complexes est traité dans cet article à l'aide d'une procédure arborescente. Étant donné un mélange chimique dont l'état est connu, cette procédure détermine un ensemble d'agencements de séparateurs, permettant d'obtenir les produits désirés, au moindre coût total, qui est la somme des coûts opératoires et des coûts d'investissement. Cet ensemble de solutions comprend celle de coût minimal, ainsi que les agencements dont les coûts sont inférieurs ou égaux à une limite supérieure, déterminée a priori en fonction du coût minimal.*

Une application est présentée dans le cadre de la rectification industrielle de mélanges complexes pour déterminer la structure d'un train de colonnes de coût total minimal. Dans ce cas, la fonction d'évaluation est définie à l'aide d'une règle heuristique. Deux exemples numériques sont détaillés afin d'illustrer la procédure d'optimisation.

Mots clés : Procédure arborescente; borne heuristique; séparation; rectification.

Abstract. — *In this paper, the problem of the optimal separation of complex multicomponent mixtures, is solved with a Branch and Bound procedure. For a given multicomponent feed at a known state, the procedure finds a set of separators arrangements giving the goal products, with a total lower cost, where the total cost is the sum of the annual cost of utilities with the annualized cost of equipment. This set of solutions contains the one of minimal cost, and all the sequences whose costs are within a specified factor of the minimal cost.*

An application is given in the field of the complex multicomponent rectification, in order to synthesize the sequence of rectification columns with the minimal cost. In that case, the lower bound is defined with the help of an heuristic rule. In order to illustrate the optimization procedure two numerical examples are detailed.

Keywords: Branch and Bound procedure; heuristic bound; separation; rectification.

(*) Reçu en mars 1983.

(¹) Institut du Génie chimique, U.A.-C.N.R.S. n° 192, Chemin de la Loge, 31078 Toulouse Cedex, France.

INTRODUCTION

Les procédés de séparation de mélanges complexes constituent une partie importante de l'investissement d'un grand nombre d'installations chimiques industrielles. La conception de tels procédés, qui sont de grands consommateurs d'énergie et qui nécessitent des investissements lourds, doit être abordée d'une façon rigoureuse, afin que les prix de revient des produits obtenus soient compétitifs. Dans les conditions économiques présentes, une approche systématique du problème de la conception de procédés de séparation de mélanges complexes, nous paraît être importante. Ce problème d'optimisation combinatoire peut se formuler comme suit : étant donnée une alimentation dont l'état est connu, trouver l'agencement de séparateurs permettant d'obtenir les produits désirés, au moindre coût total. Le coût total est le cumul des coûts de chaque séparateur du procédé, le coût d'un séparateur étant la somme des coûts opératoires et des coûts d'investissement rapportés à une période de temps donnée. L'évaluation du coût d'un séparateur nécessite son dimensionnement en fonction de son alimentation et des spécifications imposées. En effet, le coût d'investissement est une fonction du diamètre, de la hauteur du séparateur, des caractéristiques internes de ce dernier, comme par exemple les plateaux, le garnissage ou le solvant, ainsi que des échangeurs de récupération de chaleur utilisés. D'autre part, le coût opératoire est la somme des coûts énergétiques liés au chauffage ou au refroidissement de certaines parties du séparateur; par exemple pour une colonne de rectification, il s'agit de la somme des coûts de l'énergie à apporter au bouilleur et au condenseur. Par ailleurs, des facteurs d'ajustement permettant par exemple de tenir compte du matériau de construction, de la nature des fluides caloporteurs et de l'inflation, apparaissent dans l'expression de ces coûts.

Nous présentons un algorithme qui génère automatiquement, à partir d'une alimentation donnée d'état connu, une séquence de séparations de coût total minimal – dite séquence optimale – permettant d'obtenir les produits voulus. Cet algorithme peut aussi générer un ensemble de séquences voisines de la séquence optimale, dont les coûts totaux sont compris dans un intervalle d'amplitude fixée, tel que le coût minimal soit la borne inférieure de l'intervalle. Le calcul du coût d'investissement pour un séparateur donné, comporte en général des imprécisions, soit parce que le dimensionnement de l'appareil a été effectué par une méthode rapide et approximative (méthodes Short-Cut en distillation), soit parce que le coût est obtenu par lissage de courbes. Le fait de disposer d'un ensemble de séquences voisines de la séquence optimale peut permettre de tenir compte de ces imprécisions de calcul, en utilisant sur ce nombre réduit de séquences des méthodes de dimensionnement et de calcul des coûts plus rigoureuses, afin d'obtenir des critères de choix plus précis.

TRAVAUX ANTÉRIEURS

La majorité des travaux antérieurs s'appuie sur l'hypothèse fondamentale suivante :

(H1) toute séparation est tranchée, c'est-à-dire qu'elle partage un courant d'alimentation (qui est unique pour chaque séparateur) en deux courants de sortie, tel qu'un constituant donné de l'alimentation d'un séparateur n'apparaisse que dans un seul courant de sortie.

Une colonne de distillation opérant avec un fort taux de séparation d'un ensemble de constituants est un exemple de séparation tranchée. La plupart des auteurs utilisent la notion de « liste ordonnée » (Hendry et Hughes [7]) qui contient les constituants à traiter rangés par ordre décroissant suivant la valeur de la propriété physique exploitée dans la méthode de séparation considérée (par exemple, la volatilité relative pour la distillation). Cette liste ordonnée permet d'identifier facilement toutes les séparations tranchées possibles au niveau de deux constituants adjacents. L'espace de recherche qui est l'ensemble de toutes les séquences de séparations possibles, peut être représenté par une arborescence, composée d'opérateurs (les différents types de séparateurs) et d'opérandes (les constituants séparés). En considérant par exemple la liste ordonnée (1, 2, 3, 4), l'espace de recherche de toutes les séquences permettant d'obtenir les quatre produits purs 1, 2, 3 et 4, à l'aide d'un seul type de séparateur, est représenté sur la figure 1. Le séparation (1) est le sommet de départ D , les sommets (2) à (8) sont les sommets intermédiaires, les sommets (9) à (13) sont les sommets terminaux T .

A partir de cette structure arborescente, les méthodes de génération de séquences de séparation, essayent de déterminer un chemin, partant du sommet D et aboutissant à un sommet T , ayant un coût total minimal, le coût étant la somme du coût de passage de chaque sommet du chemin vers son successeur. Pour un mélange à N constituants qui doit être séparé en N produits purs, l'utilisation de M méthodes de séparations, conduit à un nombre de séquences possibles donné par Thompson et King [20] :

$$R = \frac{[2(N-1)]!}{N! (N-1)!} M^{(N-1)}. \quad (1)$$

Beaucoup de méthodes traitent le cas où $M=1$, mais R n'en demeure pas moins une fonction fortement croissante de N . Il s'ensuit que toutes les techniques proposées pour traiter ce problème, visent à obtenir une solution optimale ou sous-optimale en explorant au minimum l'espace de recherche. Durant la dernière décennie, de nombreuses méthodes que l'on peut sérier

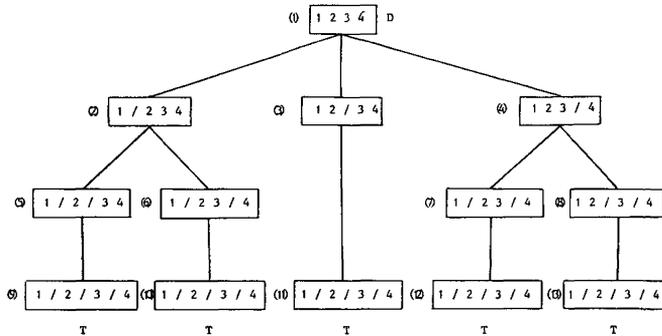


Figure 1. — Espace de recherche pour un exemple comportant quatre constituants.

en trois catégories (Nishida *et al.* [13], Minderman et Tedder [9]) ont été proposées et sont présentées dans le tableau I.

TABLEAU I

Classification des méthodes de génération de séquences de séparations.

| | Références |
|--|---|
| Méthode heuristiques | Heaven [6], Thompson et King [20], Hendry et Hughes [7], Doukas et Luyben [2], Minderman et Tedder [9]. |
| Méthodes évolutives | Stephanopoulos et Westerberg [19], Seader et Westerberg [18], Nath et Motard [10]. |
| Méthodes algorithmiques de type paramétrique . . . | Petlyuk <i>et al.</i> [14], Maikov <i>et al.</i> [8], Umeda <i>et al.</i> [21], Nishida <i>et al.</i> [11], Nishida et Powers [12]. |
| Méthodes algorithmiques fondées sur la théorie des graphes | Westerberg et Stephanopoulos [22], Rodrigo et Seader [16], Gomez et Seader [3], Rathore <i>et al.</i> [15]. |

● Les méthodes heuristiques, qui utilisent des règles issues d'un apprentissage expérimental, ont pu être développées grâce au travail de Heaven [6]. Cet auteur, après une étude systématique de nombreux cas, a pu dégager des règles heuristiques qui sont fréquemment utilisées actuellement dans le domaine de la distillation. Ces règles ont été étendues au cas de plusieurs

autres types de séparateurs ou au problème de récupération d'énergie [20, 7, 2, 9].

- Les méthodes évolutives, qui sont aussi des méthodes heuristiques, tentent d'identifier la meilleure séquence par une succession d'améliorations locales apportées à une solution initiale. Elles ont été séparées du premier type de méthodes heuristiques, car elles présentent une stratégie commune, consistant à opérer en trois étapes :

(a) génération d'une séquence admissible, à partir de règles heuristiques, pour initialiser le processus;

(b) définition de règles d'évolution pour modifier les séquences obtenues au cours de la recherche;

(c) utilisation d'une méthodologie d'évolution, jusqu'à ce qu'aucune diminution du critère ne soit possible.

- Les méthodes algorithmiques, qui utilisent des techniques classiques développées en programmation mathématique, sont en principe infaillibles mais souvent très lourdes à développer. Afin de diminuer les difficultés de mise en œuvre, ces méthodes utilisent souvent des règles heuristiques, mais elles diffèrent des techniques purement heuristiques, les développements théoriques étant plus importants que la part des éléments heuristiques. Ces méthodes peuvent être divisées en deux sous-groupes :

1° les méthodes paramétriques qui déterminent les paramètres principaux caractérisant les séquences, et résolvent ensuite un problème d'optimisation continu à grande échelle;

2° les méthodes qui utilisent la théorie des graphes en résolvant des problèmes d'optimisation discrets; elles permettent en général de rejeter des séquences avant qu'elles ne soient entièrement développées. Certains auteurs ([22, 16]) commencent, à l'aide de règles heuristiques, par développer une séquence qui donne une borne supérieure du coût optimal. Westerberg et Stephanopoulos [22] utilisent ensuite une méthode Branch and Bound, dans laquelle, la borne inférieure du coût réel d'une séquence est obtenue par minimisation des souslagrangiens qui sont les composantes du lagrangien associé au problème. Les paramètres de Lagrange sont relatifs aux contraintes de connexion entre séparateurs (l'une des sorties d'un séparateur est l'entrée de l'un des deux séparateurs successeurs). Rodrigo et Seader [16] choisissent pour prochain élément d'une séquence, le séparateur de plus faible coût, ce qui est contraire au principe d'optimalité, et arrêtent l'exploration de la séquence lorsque son coût excède la borne supérieure. Rathore *et al.* [15] ont utilisé la programmation dynamique combinée avec une stratégie de bornage. Gomez et Seader [3] associent à chaque sommet de l'espace de recherche, une

fonction heuristique qui est un minorant du coût des séquences complètes passant par ce sommet; les sommets sont ensuite séparés dans l'ordre des valeurs croissantes des fonctions heuristiques. Les auteurs montrent, sous certaines hypothèses, que leur algorithme converge, c'est-à-dire qu'il permet de trouver la séquence optimale.

● Les méthodes s'appuyant sur la théorie des graphes semblent être les plus appropriées pour traiter le problème de la génération automatique de séquences de séparations, pour les trois raisons principales suivantes :

(a) elles peuvent être aisément mises en œuvre sur un ordinateur;

(b) elles sont plus rigoureuses que les méthodes purement heuristiques et évolutives;

(c) elles exploitent le caractère discret du problème de séparations tranchées; on n'est donc pas confronté, comme dans les méthodes paramétriques, à la résolution de problèmes continus à grande échelle. Il est à noter que ces dernières méthodes n'ont été utilisées que sur des problèmes de faibles dimensions.

A condition de disposer d'une bonne fonction d'évaluation, l'efficacité des méthodes arborescentes, pour explorer les solutions d'un problème combinatoire n'est plus à démontrer (Roy [17]). Cependant, l'obtention rapide de la fonction d'évaluation constitue la difficulté majeure des méthodes arborescentes, lorsqu'elles sont utilisées dans le cadre de la séparation de mélanges chimiques complexes. Par exemple dans l'algorithme de Westerberg et Stephanopoulos [22], pour déterminer la fonction d'évaluation, il est nécessaire de calculer les paramètres de Lagrange et de minimiser des souslagrangiens, ce qui semble être un problème aussi complexe que celui que la méthode arborescente doit résoudre. Toutefois, dans certains cas particuliers, il est possible de déterminer facilement une fonction d'évaluation, permettant de guider l'exploration de l'arborescence des solutions, et donc de définir un algorithme qui converge vers la solution optimale du problème.

PRÉSENTATION DE L'ALGORITHME

L'algorithme utilise une Procédure par Séparation et Évaluation Progressive (P.S.E.P.) (Gondran et Minoux [4], Roy [17]), pour explorer l'arborescence des solutions du problème de séparation de listes. Toute procédure de type P.S.E.P. s'appuie sur les deux principes fondamentaux suivants :

— un principe de séparation permettant d'engendrer à partir de l'ensemble D des solutions réalisables, une arborescence dont D est le sommet racine, et les autres sommets constituant des sous-ensembles de plus en plus petits de D ;

— un principe de déplacement dans l'arborescence servant, à l'aide d'une fonction d'évaluation, à éliminer des sous-ensembles de D obtenus après chaque séparation, et à sélectionner d'autres sous-ensembles sur lesquels le principe de séparation sera utilisé.

Pour définir l'algorithme, nous utiliserons les trois hypothèses fondamentales suivantes :

- (H 1) toute séparation est tranchée;
- (H 2) l'ordre des constituants dans la liste ordonnée est le même dans toute l'arborescence;
- (H 3) un seul type de séparateur est utilisé.

Les hypothèses (H 1) et (H 2) sont généralement admises dans la littérature. Nous avons introduit l'hypothèse (H 3) pour avoir une définition unique de la borne inférieure, qui est la fonction d'évaluation utilisée dans l'algorithme, car cette définition dépend essentiellement du type de séparateur envisagé. Il est à noter que la suppression de (H 3) n'entraînerait que peu de modifications à l'algorithme, mais se poserait alors un problème délicat de définition de la fonction d'évaluation. L'algorithme est général dans le sens où il peut s'appliquer à tout problème de séparation de listes, pour lequel on sait définir une fonction d'évaluation. Il permet de générer une séquence de coût minimal \hat{C} , appelée séquence optimale, ainsi qu'un ensemble de séquences voisines de la séquence optimale, appelées aussi séquences satisfaisantes [17], dont les coûts C_V sont tels que :

$$C_V \leq (1 + \alpha) \hat{C}, \quad (2)$$

où α est un scalaire positif quelconque donné.

Dans l'algorithme, nous utiliserons une borne C_{sup} du coût de la séquence optimale,

$$\hat{C} \leq C_{\text{sup}}, \quad (3)$$

en désignant par C_{inf} une borne inférieure du coût C d'une séquence complète quelconque, si on a :

$$C_{\text{inf}} > (1 + \alpha) C_{\text{sup}}, \quad (4)$$

il vient, d'après (3) :

$$C > (1 + \alpha) \hat{C}. \quad (5)$$

La relation (5) montre que la séquence de coût C n'est ni optimale, ni satisfaisante. Ce principe sera utilisé dans l'algorithme P.S.E.P. pour éviter

de séparer certains sommets de l'arborescence. Toutefois, la difficulté majeure de la méthode de recherche réside dans la détermination des bornes C_{inf} et C_{sup} , de façon à réduire le plus possible l'arborescence des solutions.

Le principe de séparation permettant d'engendrer à partir du mélange complexe à traiter, qui est le sommet racine D , une arborescence, est relativement simple; pour un sommet donné, il consiste à déterminer toutes les combinaisons de deux constituants clés adjacents, présentes dans ce sommet, et à séparer le mélange au niveau de ces deux constituants clés. Si le sommet ne contient que des mélanges binaires (respectivement des produits purs), c'est un sommet terminal binaire (respectivement terminal unaire). Les hypothèses (H 1) et (H 2) permettent de montrer que le principe de séparation adopté, vérifie les trois propriétés (finitude, conservation, arrêt) relatives à un principe de séparation (Roy [17]).

La fonction d'évaluation choisie associée à un sommet j de l'arborescence, notée $g_j^*(D, T)$ est une fonction minorante des coûts de toutes les séquences complètes passant par le sommet j . La détermination de la fonction $g_j^*(D, T)$ est assez délicate, car il faut pouvoir calculer cette fonction minorante pour tout sommet j , en particulier non terminal, et ce sans connaître les valeurs C des coûts des séquences complètes passant par le sommet j . La fonction minorante associée à un sommet j est la somme de deux termes p et f , où p (respectivement f) représente le passé (respectivement le futur) du sommet j .

Le terme p , qui ne pose pas de difficulté de calcul, est égal au coût de passage de la racine D jusqu'au sommet j , et s'exprime sous la forme :

$$p = g(D, i) + C(i, j), \quad (6)$$

où $g(D, i)$ est le coût de la séquence partielle débutant en D et aboutissant sur le prédécesseur (unique) i du sommet j , et $C(i, j)$ est le coût de passage de i à j .

La détermination du terme f dépend du domaine dans lequel l'algorithme doit être utilisé. Nous poserons d'une façon générale $f = C^*(j, T)$, avec la contrainte :

$$C^*(j, T) \leq C(j, T_k), \quad \forall j, \quad \forall k \in I_j, \quad (7)$$

où I_j est l'ensemble des indices des sommets terminaux unaires que l'on peut atteindre à partir de j .

$C(j, T_k)$ est le coût d'une séquence partielle débutant en j et aboutissant au sommet terminal unaire T_k .

La relation (7) signifie que $C^*(j, T)$ est un minorant des coûts de toutes les séquences partielles débutant en j et aboutissant à des sommets terminaux. En rassemblant les relations (6) et (7), on définit la fonction d'évaluation $g_j^*(D, T)$ par :

$$g_j^*(D, T) = g(D, i) + C(i, j) + C^*(j, T). \quad (8)$$

On initialise le processus en posant :

$$g(D, D) = 0. \quad (9)$$

Nous supposons que la fonction g^* , donnée par la relation (8), vérifie les trois propriétés suivantes :

(P1) g^* est une fonction croissante au cours de la génération d'une séquence, c'est-à-dire : $g_j^*(D, T) \geq g_i^*(D, T)$ pour tous les sommets i et j , avec j successeur de i ;

(P2) si j est un sommet terminal unaire, c'est-à-dire $j = T_k$, $k \in I_j$, alors : $g_j^*(D, T) = g(D, j)$;

(P3) si j est un sommet terminal binaire, père d'un sommet terminal unaire T_k , alors : $g_j^*(D, T) = g(D, T_k)$.

De par sa construction, la fonction g^* donnée par la relation (8) et possédant les propriétés (P2) et (P3), vérifie les deux propriétés caractéristiques d'une fonction d'évaluation, qui sont la minoration et la coïncidence pour les nœuds terminaux (Roy [17]).

Disposant d'un principe de séparation et d'une fonction d'évaluation possédant tous deux les propriétés requises, il faut maintenant définir une stratégie de déplacement dans l'arborescence. La stratégie utilisée est une procédure du type Séparation et Évaluation Progressive; elle consiste donc à séparer en premier lieu le sommet pendant de l'arborescence, correspondant à la valeur minimale de la fonction d'évaluation.

Nous utilisons dans l'algorithme décrit ci-après une borne supérieure C_{sup} , définie par la relation (3), afin d'introduire deux critères de rejet (C1) et (C2), qui permettront d'éliminer le sommet i de l'arborescence si :

$$(C1) \quad g(D, i) \geq (1 + \alpha) C_{\text{sup}},$$

ou si :

$$(C2) \quad g_i^*(D, T) \geq (1 + \alpha) C_{\text{sup}}.$$

A partir d'une valeur de la borne C_{sup} arbitrairement grande, le calcul de cette borne en cours de recherche ne pose pas de problème : C_{sup} est prise égale au minimum des coûts des séquences qui ont été entièrement générées, jusqu'à cette étape de la recherche, multiplié par le facteur $(1 + \alpha)$, de façon

à obtenir toutes les séquences satisfaisantes. On peut remarquer que la borne C_{sup} décroît continuellement lors de la progression de la recherche, les critères de rejet (C1) et (C2) permettront donc d'éliminer un nombre de plus en plus important de sommets.

ÉTAPES LOGIQUES DE L'ALGORITHME P.S.E.P.

Pour un ensemble D à N éléments de liste, rangés dans l'ordre croissant 1 à N , les étapes logiques de l'algorithme P.S.E.P. sont les suivantes.

Étape 1. Phase d'initialisation

Soit D la racine de l'arborescence.

Ranger D dans la liste LO des sommets candidats à une séparation.

Initialiser C_{sup} et le coût optimal \hat{C} avec un réel arbitrairement grand, ainsi que le coût $g(D, D)$ avec la valeur 0.

Étape 2. Phase de sélection

Si la liste LO est vide, la procédure d'exploration est terminée; aller à l'Étape 4.

Sinon choisir dans cette liste, le sommet i dont la fonction d'évaluation $g_i^*(D, T)$ est minimale.

Étape 3. Phase de séparation, d'évaluation et d'examen

(a) Si i est un sommet terminal (unaire ou binaire), le ranger dans la liste LOPT des sommets terminaux. Faire, d'après la propriété (P2) ou (P3) $g(D, i) = g_i^*(D, T)$, et si le coût \hat{C} est strictement supérieur à $g_i^*(D, T)$, actualiser \hat{C} avec la valeur de $g_i^*(D, T)$, et redéfinir C_{sup} . Puis supprimer i de la liste LO et revenir à l'Étape 2.

(b) Si le sommet i vérifie le critère de rejet (C1), éliminer i de la liste LO et revenir à l'Étape 2.

(c) Si le sommet i vérifie le critère de rejet (C2), la procédure d'exploration est terminée; aller à l'Étape 4.

(d) Dans tous les autres cas, appliquer le principe de séparation et d'évaluation afin de définir les sommets fils de i , puis supprimer i de la liste LO et revenir à l'Étape 2.

Etape 4. Phase finale d'actualisation

A ce niveau, la procédure d'exploration est terminée, il faut donc mettre à jour la liste LOPT, qui contient tous les sommets terminaux trouvés lors de la recherche, de telle sorte qu'elle ne contienne plus que les solutions satisfaisantes. Il faut donc supprimer de LOPT tous les sommets terminaux j tels que leurs coûts $g(D, j)$ soient supérieurs à $(1 + \alpha) C_{\text{sup}}$.

PROPRIÉTÉS DE L'ALGORITHME

Roy [17] a montré qu'un algorithme de type P.S.E.P., qui utilise un principe de séparation et une fonction d'évaluation possédant les propriétés décrites ci-dessus, est un algorithme convergent.

Le fait d'utiliser les deux critères de rejet (C1) et (C2), définis à l'aide d'une borne supérieure continuellement décroissante, permettant donc de rejeter un nombre de plus en plus important de sommets au cours de l'exploration, accroît la vitesse de convergence de l'algorithme.

La propriété de croissance (P1) de la fonction d'évaluation g^* n'est pas nécessaire pour la convergence de l'algorithme. On suppose toutefois que la fonction g^* vérifie cette propriété, afin que le critère de rejet (C2) permette d'éliminer un nombre de plus en plus important de sommets lors de la progression de la recherche, afin d'accroître encore la vitesse de convergence de l'algorithme.

APPLICATION A UN TRAIN DE COLONNES DE RECTIFICATION

Dans ce type d'application, afin de pouvoir reconstituer l'histoire des séparations, on utilise dans l'arborescence, les symboles $\mathbf{0}$ (respectivement -1 , -2 , etc.) pour représenter les points de séparation à l'étape courante (respectivement étapes précédentes). En considérant l'exemple présenté sur la figure 1, l'espace de recherche est donné par l'arborescence de la figure 2. Les parties en pointillés donnent les sommets terminaux que l'on obtiendrait si l'on appliquait le principe de séparation aux mélanges binaires. Les numéros des sommets figurent entre parenthèses.

Le procédé étudié ici est la rectification continue sans soutirage intermédiaire, à fort taux de séparation. Le problème majeur qui se pose dans cette application est la détermination du terme C^* vérifiant la relation (7) et possédant les propriétés (P2) et (P3) afin que l'algorithme soit convergent. On définit $C^*(j, T)$ en remarquant que pour passer du sommet non terminal

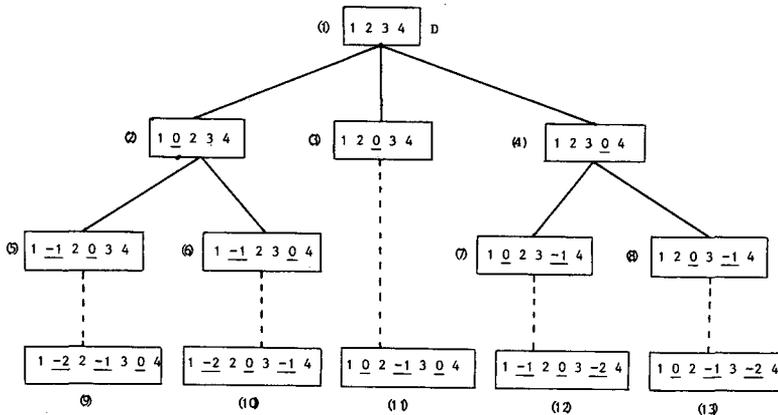


Figure 2. — Exemple d'espace de recherche pour l'algorithme.

j à un sommet terminal unaire T , il faut au moins effectuer toutes les séparations de binaires adjacents existant au niveau du sommet j . Nous avons utilisé la règle heuristique (RHH) citée par Heaven [6] :

(RHH) : « A separation is least expensive when conducted in the absence of nonkey components ».

Le terme C^* est défini par :

$$C^*(j, T) = \sum_{k \in I_j} C(X_k, T_k), \quad \forall j, \quad \forall k \in I_j \tag{10}$$

où :

I_j et T_k ont été définis plus haut [relation (7)];

X_k désigne un prédécesseur immédiat du sommet terminal unaire T_k (X_k contient au plus des mélanges binaires; il constitue donc un sommet terminal binaire).

La relation (10) signifie que $C^*(j, T)$ est égale à la somme des coûts de séparation des binaires adjacents existant au niveau du sommet j . Il découle de la règle (RHH) :

$$C^*(j, T) \leq C(j, T_k), \quad \forall j, \quad \forall k \in I_j \tag{11}$$

la relation (11) qui est analogue à la relation (7). En appliquant par exemple, la définition (10) au sommet 4 de la figure 2, on obtient :

$$C^*(4, T) \leq C(4, 12) \quad \text{et} \quad C^*(4, T) \leq C(4, 13),$$

avec :

$$C^*(4, T) = \text{coût pour séparer } (1, 2) + \text{coût pour séparer } (2, 3);$$

$$C(4, 12) = C(4, 7) + C(7, 12);$$

$$C(4, 13) = C(4, 8) + C(8, 13).$$

Nous allons maintenant examiner les propriétés de la fonction C^* . On définit le niveau 1 de l'espace de recherche associé à un sommet racine D à N constituants, par l'ensemble des sommets obtenus à partir de D après 1 séparation. Le niveau $(N-1)$ est donc l'ensemble des sommets terminaux unaires.

LEMME : *A un sommet quelconque de niveau p , correspondent $(N-p-1)$ séparations de binaires adjacents.*

Preuve : La relation est vraie pour $p=0$. Supposons-la vraie à un niveau p donné. Pour passer d'un sommet i du niveau p à un de ses successeurs j , du niveau $(p+1)$, on fait dans i une séparation au moins binaire. Le nombre de séparations de binaires adjacents que l'on peut faire dans j a diminué de 1 par rapport à sa valeur au sommet i , ce qui donne pour j un nombre égal à :

$$(N-p-1) - 1 = (N-(p+1) - 1).$$

THÉORÈME 1 : *C^* étant définie par la relation (10), la fonction g^* vérifie la propriété (P1).*

Démonstration : D'après le lemme, on a :

$$C^*(i, T) = C^*(j, T) + C(Xq, Tq),$$

où j est un successeur du sommet i et $C(Xq, Tq)$ est le coût de la séparation binaire présente dans i et donc absente dans j . En vertu de (RHH), il vient :

$$C(Xq, Tq) \leq C(i, j)$$

et donc :

$$C^*(i, T) \leq C^*(j, T) + C(i, j).$$

En ajoutant $g(D, i)$ aux deux membres de l'inégalité précédente, on obtient :

$$g_i^*(D, T) = g(D, i) + C^*(i, T) \leq g_j^*(D, T) = g(D, i) + C(i, j) + C^*(j, T),$$

ce qui démontre le théorème 1.

THÉORÈME 2 : *C^* étant définie par la relation (10), la fonction g^* vérifie les propriétés (P2) et (P3).*

Démonstration :

Cas 1 (sommet terminal unaire) :

$$\begin{aligned} j &= T_k, \quad \forall k \in I_j, \\ g_j^*(D, T) &= g(D, j) + C^*(j, T), \\ g(D, T_k) &= g(D, j) + C(j, T_k), \end{aligned}$$

Or, on a :

$$C(j, T_k) = C^*(j, T) = 0.$$

La propriété (P2) est vérifiée.

Cas 2 (sommet terminal binaire).

j étant un prédécesseur immédiat d'un sommet terminal unaire T_k , il ne contient que des mélanges binaires. D'après (10), on a :

$$C^*(j, T) = C(j, T_k).$$

En ajoutant $g(D, j)$ aux deux membres de l'égalité, on constate que (P3) est vérifiée.

Ainsi, à partir de la règle (RHH), les théorèmes 1, 2 et la relation (11) montrent que l'algorithme appliqué à un train de colonnes de distillation, est convergent; le théorème 1 permet d'espérer une convergence assez rapide.

Remarque : Si à l'étape 3-d de l'algorithme, un successeur j de i ne diffère de i que par une séparation binaire, alors on ne considère pas j comme un successeur de i . En effet, les successeurs de j et ceux de i sont identiques; il est donc inutile de traiter deux fois les mêmes séquences.

Soit le sommet i suivant :

$$i = (a_1, a_2, \dots, a_m, -1, a_p, \dots, a_{p+q}, -2, a_l, a_{l+1}, -3, \dots).$$

Un successeur j_B ne différant de i que par la séparation binaire $a_1 \mathbf{0} a_{l+1}$ s'écrit :

$$j_B = (a_1, a_2, \dots, a_m, -2, a_p, \dots, a_{p+q}, -3, a_l \mathbf{0} a_{l+1}, -4, \dots).$$

Soit j_A un autre successeur de i différent de j_B qui s'écrit :

$$j_A = (a_1 \mathbf{0} a_2, \dots, a_m, -2, a_p, \dots, a_{p+q}, -3, a_l, a_{l+1}, -4, \dots).$$

Il existe toujours un successeur k_A (respectivement k_B) de j_A (respectivement j_B), pouvant s'écrire par exemple :

$$k_A = (a_1, -1, a_2, \dots, a_m, -3, a_p, \dots, a_{p+q}, -4, a_l, 0, a_{l+1}, -5, \dots),$$

$$k_B = (a_1, 0, a_2, \dots, a_m, -3, a_p, \dots, a_{p+q}, -4, a_l, -1, a_{l+1}, -5, \dots).$$

Les sommets k_A et k_B sont équivalents car ils sont situés sur les mêmes séquences de séparations, ce qui montre qu'il est inutile de considérer des successeurs du type j_B .

EXEMPLES D'APPLICATION

Le premier exemple est issu de l'article de Rathore *et al.* [15]. Il s'agit de générer la séquence d'un train de colonnes de rectification traitant le mélange de départ suivant :

| Constituant | Fraction molaire d'alimentation |
|---------------------------------|---------------------------------|
| Propane (1) | 0,05 |
| <i>i</i> -butane (2) | 0,15 |
| <i>n</i> -butane (3) | 0,25 |
| <i>i</i> -pentane (4) | 0,20 |
| <i>n</i> -pentane (5) | 0,35 |

Les valeurs des coûts de séparation sont données dans la référence [15]. Nous reportons ci-dessous le déroulement de l'algorithme sur cet exemple; les numéros des sommets figurent entre parenthèses.

La séparation du sommet (1) donne les valeurs reportées dans le tableau II. On sépare ensuite le sommet (2), les résultats sont présentés dans le tableau III. Les valeurs de g et g^* sont données en centaines de milliers de dollars par an.

Cas numéro 1 : $\alpha = 0$. Recherche de la séquence optimale

Dans le tableau III, le sommet (7), qui ne contient que des mélanges binaires, permet d'obtenir :

$$g(D, 7) = g(D, T) = g_7^*(D, T) = 4,1571$$

$$\hat{C} = XLS = 4,1571 \quad [\text{où } XLS \text{ désigne le produit } (1 + \alpha) C_{\text{sup}}].$$

On obtient pour séquence complète la séquence (7', 7, 2, 1) de coût total égal à 4,1571, le sommet (7') étant défini par les séparations suivantes :

$$(7') = 1 \quad -2 \quad 2 \quad 0 \quad 3 \quad -1 \quad 4 \quad 0 \quad 5.$$

Le seul sommet restant à séparer est le sommet (4), en effet $g_3^*(D, T)$, $g_5^*(D, T)$, $g_6^*(D, T)$ et $g_8^*(D, T)$ sont supérieurs à XLS. Les résultats sont consignés dans le tableau IV. Un de ses successeurs, 1 2 3 -1 4 0 5, qui ne diffère de (4) que par la séparation binaire (4, 5) n'a pas été séparé, en vertu de la remarque précédente.

TABLEAU II
Successeurs du sommet (1).

| Prédécesseur | Successeurs | g | g^* |
|---------------|--|--------------------------------------|--------------------------------------|
| (1) 1 2 3 4 5 | (2) 1 0 2 3 4 5 (3) 1 2 0 3 4 5 (4) 1 2 3 0 4 5 (5) 1 2 3 4 0 5 | 0,5715 1,6500 1,1490 2,6600 | 3,8055 4,1960 4,0516 4,4633 |

TABLEAU III
Successeurs du sommet (2).

| Prédécesseur | Successeurs | g | g^* |
|-----------------|--|----------------------------|----------------------------|
| (2) 1 0 2 3 4 5 | (6) 1 -1 2 0 3 4 5 (7) (*) 1 -1 2 3 0 4 5 (8) 1 -1 2 3 4 0 5 | 1,9055 1,5158 2,9895 | 4,1902 4,1571 4,5315 |

(*) Sommet terminal binaire.

TABLEAU IV
Successeurs du sommet (4).

| Prédécesseur | Successeurs | g | g^* |
|-----------------|---|------------------|------------------|
| (4) 1 2 3 0 4 5 | (9) (*) 1 0 2 3 -1 4 5 (10) (*) 1 2 0 3 -1 4 5 | 1,5443 2,3470 | 4,1856 4,3003 |

(*) Sommet terminal binaire.

Les résultats présentés dans le tableau IV permettent d'identifier les deux séquences terminales (9', 9, 4, 1) et (10', 10, 4, 1) de coûts respectifs 4,1856 et 4,3003, avec :

$$(9') = 1 \quad -1 \quad 2 \quad 0 \quad 3 \quad -2 \quad 4 \quad 0 \quad 5,$$

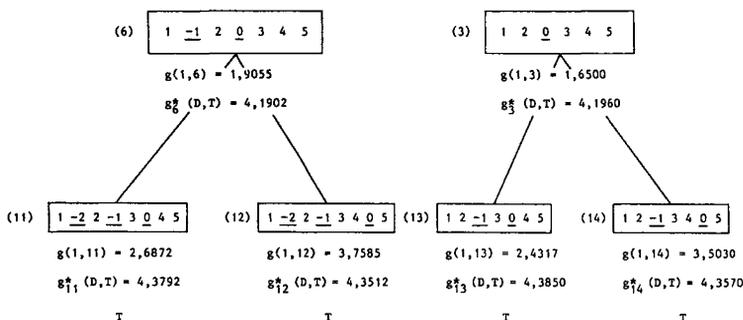


Figure 4. — Arborescence des solutions supplémentaires explorées pour l'exemple 1 avec $\alpha = 0,05$.

TABLEAU V
Successeurs du sommet (6).

| Prédécesseur | Successeurs | g | g^* |
|--------------------|--|------------------|------------------|
| (6) 1 -1 2 0 3 4 5 | (11) (*) 1 -2 2 -1 3 0 4 5 (12) (*) 1 -2 2 -1 3 4 0 5 | 2,6872 3,7585 | 4,3792 4,3512 |

(*) Sommet terminal binaire.

TABLEAU VI
Successeurs du sommet (3).

| Prédécesseur | Successeurs | g | g^* |
|-----------------|--|------------------|------------------|
| (3) 1 2 0 3 4 5 | (13) (*) 1 2 -1 3 0 4 5 (14) (*) 1 2 -1 3 4 0 5 | 2,4317 3,5030 | 4,3850 4,3570 |

(*) Sommet terminal binaire.

Dans le deuxième exemple, nous étudions la séparation par rectification du mélange à six constituants suivant :

| Constituant | Fraction molaire d'alimentation |
|-------------------------|---------------------------------|
| Éthane (1) | 0,05 |
| Propylène (2) | 0,15 |
| Propane (3) | 0,35 |
| Isobutane (4) | 0,20 |
| n-butane (5) | 0,15 |
| Pentane (6) | 0,10 |

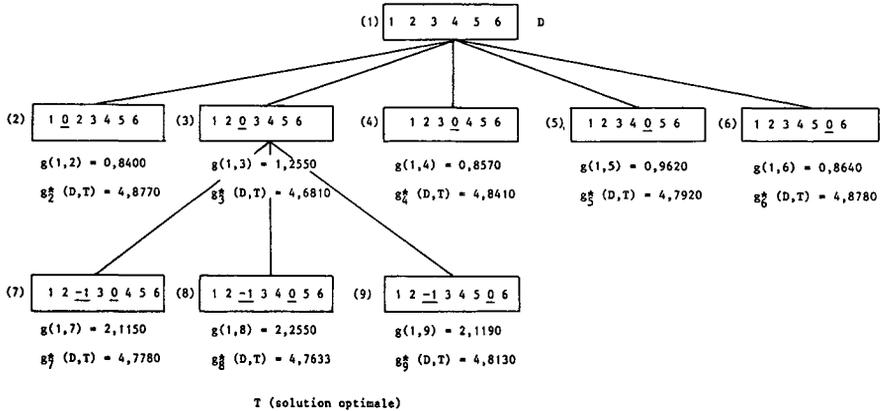


Figure 5. — Arborescence des solutions explorées pour le deuxième exemple.

Les caractéristiques de chaque colonne de rectification (hauteur, diamètre, nombre de plateaux, position du plateau d'alimentation, bilans thermiques au bouilleur et au condenseur) sont déterminées à l'aide d'une méthode rapide de type Short-Cut. La méthode de Guthrie [5] permet de déterminer le coût d'investissement d'un séparateur en fonction de ses caractéristiques de dimensionnement, ainsi que le coût opératoire en fonction des bilans thermiques. Nous avons étudié sur cet exemple l'influence des conditions opératoires (pression, taux de reflux) de chaque séparateur; nous avons constaté, d'une part que la valeur optimale du reflux est toujours comprise entre 1,4 et 1,5 fois le reflux minimal, et d'autre part que de faibles pressions opératoires (≤ 5 atm) conduisent toujours à des coûts totaux moindres.

Un exemple de recherche de la séquence optimale ($\alpha=0$), lorsque le reflux est égal à 1,5 fois le reflux minimal, est donné sur la figure 5.

CONCLUSION

L'algorithme présenté est un outil très intéressant pour traiter toutes sortes de problèmes de séparations tranchées de listes, pour lesquels on peut définir un principe de séparation et une fonction d'évaluation ayant les propriétés requises pour assurer la convergence.

A titre d'illustration, nous avons utilisé cet algorithme pour traiter le problème de la génération automatique de séquences de colonnes à distiller. Dans cette application, la fonction d'évaluation a été définie à l'aide d'une

règle heuristique, et nous avons montré que cette fonction possédait les propriétés nécessaires à une convergence relativement rapide. Nous avons obtenu pour chaque problème particulier, une séquence optimale de la cascade de colonnes, selon un critère économique, ainsi qu'un ensemble de séquences voisines de la séquence optimale dont l'intérêt a été signalé en introduction. Il ressort de notre expérience d'utilisation de l'algorithme, que cette méthode permet d'obtenir la solution optimale (nous n'avons pas de contre-exemple à ce jour), même lorsque le terme C^* ne vérifie pas la propriété fondamentale (7) en certains sommets de l'espace de recherche.

La programmation de cet algorithme nécessite un faible espace mémoire, puisque par exemple dans le cadre de la rectification, nous avons implanté l'ensemble du programme, comprenant, outre la méthode de recherche, la procédure de dimensionnement et d'évaluation du coût total d'un séparateur sur moins de 32 koctets de mémoire centrale.

A condition de pouvoir définir une fonction d'évaluation ayant les propriétés requises pour assurer la convergence, l'utilisation de l'algorithme nous paraît prometteuse pour le Génie chimique dans d'autres domaines que la distillation.

Dans le cas beaucoup plus complexe de séparations non tranchées de listes (un constituant de l'alimentation pouvant apparaître dans les deux courants de sortie, en proportion variable, dans le cas de la rectification), qui est un problème continu d'optimisation avec contraintes à grande échelle, cet algorithme peut être utilisé pour déterminer une solution initiale acceptable, point fondamental de toute méthode d'optimisation avec contraintes.

NOTATIONS

| | |
|---------------|--|
| C , | coût d'une séquence complète; |
| C_{inf} , | limite inférieure du coût d'une séquence complète; |
| \hat{C} , | coût total de la séquence optimale; |
| C_{sup} , | limite supérieure des coûts des séquences; |
| C_v , | coût total d'une séquence voisine de la séquence optimale; |
| $C(i, j)$, | coût pour passer du sommet i au sommet j ; |
| $C(j, T_k)$, | coût réel pour compléter une séquence débutant au sommet j et aboutissant au sommet terminal T_k ; |
| $C^*(j, T)$, | limite inférieure des coûts pour compléter les séquences initiées au sommet j et aboutissant à des sommets terminaux T ; |
| D , | sommet de départ; |
| $g(D, i)$, | coût réel de la séquence partielle conduisant du sommet D au sommet i ; |

| | |
|-----------------|---|
| $g_j^*(D, T)$, | fonction d'évaluation des sommets de l'arborescence; |
| I_j , | ensemble des indices des sommets terminaux unaires que l'on peut atteindre à partir du sommet j ; |
| i , | numéro d'un sommet courant; |
| j , | numéro d'un sommet successeur de i ; |
| LO, | liste contenant les numéros des sommets susceptibles d'être séparés; |
| LOPT, | liste contenant les numéros des sommets terminaux unaires et des sommets terminaux binaires trouvés lors de la recherche; |
| M , | nombre de méthodes différentes de séparations utilisées; |
| N , | nombre de constituants du sommet de départ D ; |
| R , | nombre de séquences de séparations possibles; |
| T , | sommet terminal; |
| XLS, | limite supérieure des coûts des séquences ($XLS = (1 + \alpha) \hat{C}$); |
| α , | amplitude de l'intervalle contenant les coûts des séquences voisines de la séquence optimale. |

BIBLIOGRAPHIE

1. BILLON, *Optimisation d'un train de distillation*, Document Rhône-Poulenc France transmis à l'Université, non publié, octobre 1981.
2. N. DOUKAS et W. L. LUYBEN, *Economics of Alternative Distillation Configurations for the Separation of Ternary Mixtures*, Ind. Eng. Chem. Process Des. Dev., vol. 17, 1978, p. 272.
3. A. GOMEZ et J. D. SEADER, *Separation Sequence Synthesis by a Predictor Based Ordered Search*, A.I.Ch.E. J., vol. 22, 1976, p. 970.
4. M. GONDRAAN et M. MINOUX, *Graphes et algorithmes*, Collection de la direction des Études et Recherches d'Électricité de France, Eyrolles, Paris, 1979.
5. K. M. GUTHRIE, *Chem. Eng.*, vol. 24, 1969, p. 114.
6. D. L. HEAVEN, *Optimum Sequencing of Distillation Columns in Multicomponent Fractionation*, M. S. Thesis, Univ. of California, Berkeley, 1969.
7. J. E. HENDRY et R. R. HUGHES, *Generating Separation Process Flowsheets*, Chem. Eng. Prog., vol. 68, 1972, p. 71.
8. V. P. MAIKOV, G. G. VILKOV et A. V. GALTISOV, *Optimum Design of Multicolumn Fractionating Plants from the Thermo-economic Standpoint*, Int. Chem. Eng., vol. 12, 1972, p. 426.
9. P. A. MINDERMAN et D. W. TEDDER, *Comparisons of Distillation Networks*, A.I.Ch.E. Symposium Series, 1982, p. 69.
10. R. NATH et R. L. MOTARD, *Evolutionary Synthesis of Separation Processes*, A.I.Ch.E. J., vol. 27, 1981, p. 578.
11. N. NISHIDA, A. ICHIKAWA et E. TAZAKI, *Synthesis of Optimal Process Systems with Uncertainty*, Ind. Eng. Chem. Process Des. Dev., vol. 13, 1974, p. 209.

12. N. NISHIDA et G. J. POWERS, *On the Computational Technique of Optimal Synthesis Problem Using Structural Parameters*, J. of Chem. Eng. of Jpn., vol. 11, 1978, p. 396.
13. N. NISHIDA, G. STEPHANOPOULOS et A. W. WESTERBERG, *A Review of Process Synthesis*, A.I.Ch.E. J., vol. 27, 1981, p. 321.
14. F. B. PETLYUK, V. M. PLATONOV et D. M. SLAVINKII, *Thermodynamically Optimal Method for Separating Multicomponent Mixtures*, Int. Chem. Eng., vol. 5, 1965, p. 555.
15. R. N. S. RATHORE, K. A. VANWORNER et G. J. POWERS, *Synthesis Strategies for Multicomponent Separation Systems with Energy Integration*, A.I.Ch.E. J., vol. 20, 1974, p. 491; *Synthesis in Distillation Systems with Energy Integration*, A.I.Ch.E. J., vol. 20, 1974, p. 940.
16. F. R. RODRIGO et J. D. SEADER, *Synthesis of Separation Sequences by Ordered Branch Search*, A.I.Ch.E. J., vol. 21, 1975, p. 885.
17. B. ROY, *Algèbre moderne et théorie des graphes*, t. 2, Dunod, Paris, 1970.
18. J. D. SEADER et A. W. WESTERBERG, *A Combined Heuristic and Evolutionary Strategy for Synthesis of Simple Separation Sequences*, A.I.Ch.E. J., vol. 23, 1977, p. 951.
19. G. STEPHANOPOULOS et A. W. WESTERBERG, *Studies in Process Synthesis-II*, Chem. Eng. Sc., vol. 31, 1976, p. 195.
20. R. W. THOMPSON et C. J. KING, *Systematic Synthesis of Separation Schemes*, A.I.Ch.E. J., vol. 18, 1972, p. 941.
21. T. UMEDA, A. HIRAI et A. ICHIKAWA, *Synthesis of Optimal Processing System by an Integrated Approach*, Chem. Eng. Sc., vol. 27, 1972, p. 795.
22. A. W. WESTERBERG et G. STEPHANOPOULOS, *Studies in Process Synthesis-I*, Chem. Eng. Sc., vol. 30, 1975, p. 963.