

C. LAQUERBE

P. FLOQUET

S. DOMENECH

L. PIBOULEAU

## **Conception de séquences de séparation par un algorithme génétique**

*RAIRO. Recherche opérationnelle*, tome 31, n° 4 (1997),  
p. 375-397

[http://www.numdam.org/item?id=RO\\_1997\\_\\_31\\_4\\_375\\_0](http://www.numdam.org/item?id=RO_1997__31_4_375_0)

© AFCET, 1997, 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/>

## CONCEPTION DE SÉQUENCES DE SÉPARATION PAR UN ALGORITHME GÉNÉTIQUE (\*)

par C. LAQUERBE, P. FLOQUET, S. DOMENECH et L. PIBOULEAU <sup>(1)</sup>

Communiqué par Jacques A. FERLAND

**Résumé.** – *Cet article a trait à la mise en œuvre d'un algorithme génétique non élitiste avec tirage sur roue de loterie pour la conception de séquences de séparation. Après un rappel de la combinatoire du problème, on détaille le type de codage retenu fondé sur la notion de pile, pour réaliser la séparation d'une liste de plusieurs éléments en présence de contraintes de faisabilité ainsi que les opérateurs génétiques mis en œuvre. Après avoir traité un exemple didactique permettant d'appréhender les valeurs des paramètres fondamentaux de l'algorithme, on présente deux autres exemples de grande taille inhérents à la séparation de 14 et de 16 hydrocarbures par une cascade de colonnes de rectification. Les résultats obtenus, comparés avec ceux que fournissent une méthode de recuit simulé et une procédure de recherche aléatoire pure, permettent de constater l'efficacité de l'algorithme génétique.*

**Mots clés :** Algorithmes génétiques, Codage, Séparation, Rectification.

**Abstract.** – *This paper deals with the implementation of a non elitist genetic algorithm with roulette wheel selection for the synthesis of separation sequences. After a brief recall of problem combinatorics, the coding strategy based on the notion of stack for separating a list of elements with feasibility constraints, as well as the genetic operators used, are detailed. A didactic example is first presented for estimating the numerical values of the algorithm basic parameters, and then two large scale examples concerning the separation of mixtures involving 14 and 16 hydrocarbons by means of distillation sequences, are reported. The obtained results, compared with those given by a simulated annealing procedure and a purely random search, clearly show the genetic algorithm efficiency.*

**Keywords:** Genetic algorithms, Coding, Separation, Rectification.

### NOMENCLATURE

#### Majuscules

CS	Codage secondaire des structures de séparation
CΣ	Codage somme des structures de séparation
$P(t)$	Population d'individus ou chromosomes, fonction de la génération

---

(\*) Reçu en février 1995.

<sup>(1)</sup> Laboratoire de Génie Chimique, UMR 5503 CNRS-ENSIGC-INPT, 18, Chemin de la Loge, 31078 Toulouse Cedex, France.

TS <sub>2</sub>	Vecteur composé de tous les séparateurs à 2 sorties pour un mélange à $n$ constituants donné
TS <sub>3</sub>	Vecteur composé de tous les séparateurs à 3 sorties pour un mélange à $n$ constituants donné
Minuscules	
$dg$	Taux de survie de la population
$dm$	Taux de mutation de la population
$n$	Nombre de constituants à séparer
$np$	Taille de la population
Symboles	
+ ou <b>1</b>	Opérateur de codage des structures
– ou <b>-1</b>	Opérateur de codage des structures
*	Caractère se substituant au symbole <b>1</b> ou <b>-1</b>

## 1. INTRODUCTION

En génie des procédés, les problèmes de conception d'unités complexes présentent souvent un caractère combinatoire important, rendant impraticables les approches classiques : recherche exhaustive, programmation non linéaire ou mixte. Nous présentons dans ce travail, un exemple d'application des algorithmes génétiques au domaine de la conception de séquences de colonnes de rectification complexes.

Introduite par Holland dès les années 60, l'idée fondamentale des algorithmes génétiques est, suivant le modèle de sélection et d'évolution naturelle, de faire évoluer des populations d'individus – tous solutions potentielles d'un problème donné – à partir de générations précédentes d'individus au moyen de deux processus élémentaires : la sélection et la reproduction (Goldberg (8)).

Contrairement aux méthodes d'optimisation classiques qui ne s'intéressent qu'à l'évolution d'un seul individu, les algorithmes génétiques traitent une population d'individus (ou chromosomes) qui évoluent dans le temps. C'est à la fois leur force – possibilité de trouver un optimum global – et leur faiblesse – temps de calcul fonction de la taille de la population. Pour simuler les mécanismes naturels de l'évolution, il est nécessaire de disposer :

- d'un codage des individus, en général des séquences de bits ou des vecteurs d'entiers ou de réels ;
- d'une fonction d'adaptation, ou « fitness », qui correspondra pour les problèmes d'optimisation au critère à améliorer ou à une fonction déduite de ce critère ;

– de mécanismes de sélection, de reproduction et de mutation pour le passage d'une génération à la suivante.

Ces trois points seront détaillés ci-après pour le problème de conception de séquences de séparation et des résultats illustreront la démarche.

## 2. PRÉSENTATION DU PROBLÈME

Les procédés de séparation de mélanges complexes constituent une partie importante de l'investissement d'un grand nombre d'installations chimiques industrielles. Il s'agit d'opérations permettant d'obtenir, à partir d'un mélange comportant plusieurs constituants, chacun des produits purs, ce qui correspond à des séparations tranchées. La propriété physique exploitée pour cette technique de séparation est la volatilité relative qui permet d'obtenir de bas en haut des produits de plus en plus légers. Il est aussi possible d'établir une liste ordonnée des constituants à séparer (du plus léger au plus lourd) qui permet d'identifier facilement les séparations tranchées possibles au niveau de deux constituants adjacents. La conception de tels procédés, qui sont de grands consommateurs d'énergie et qui nécessitent des investissements lourds (une colonne industrielle peut mesurer plusieurs dizaines de mètres de hauteur), doit être abordée de 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, peut conduire à des gains très substantiels. Rappelons que les colonnes de séparation traditionnellement utilisées dans l'industrie sont de deux types :

– colonnes de rectification classiques à une entrée et deux sorties (sortie haute ou distillat : constituants légers ; sortie basse ou résidu : constituants lourds) ;

– colonnes de rectification complexes avec un soutirage intermédiaire situé entre le distillat et le résidu (constituants intermédiaires).

Le problème de la conception optimale de séquences de séparation consiste donc à définir la meilleure façon – au sens d'un critère technico-économique, somme des coûts d'investissements et des frais de fonctionnement annualisés – de séparer un mélange multiconstituants. Le coût de fonctionnement, constitué par la consommation des séquences en utilités (vapeurs de chauffe ou bien eau de refroidissement), représente en général une fraction importante (souvent supérieure à 50 %) du coût global des cascades de séparation par rectification. Mais d'après Stephanopoulos *et al.* (12) les

séquences optimales ou quasi optimales générées sans intégration énergétique semblent offrir les meilleures possibilités de croisement thermique des courants présents, c'est-à-dire que dans la majorité des cas une optimisation avec intégration énergétique fournit la même structure de séquence qu'une optimisation sans intégration énergétique. Ceci permet donc de réaliser ultérieurement une intégration énergétique satisfaisante.

Les séquences sont représentées sous forme d'une liste ordonnée identifiant de manière unique une structure de colonnes de séparation. Des hypothèses classiques sont retenues :

- séparation tranchée entre constituants adjacents : chaque élément du mélange ne se retrouve que dans un courant de sortie des séparateurs et un seul ;
- les constituants sont rangés dans une liste ordonnée (par volatilité relative décroissante) invariante au cours de la séparation ;
- deux types de séparateurs : colonnes de rectification classiques ou complexes ;
- fonctionnement continu de la cascade de séparateurs sans intégration énergétique, ni division des courants ;

Le nombre de possibilités de structures de séquences de colonnes à caractéristiques opératoires fixées croît très fortement (Floquet *et al.* (6)) en fonction du nombre de constituants à séparer. Un exemple de l'ensemble des séquences de séparation possibles pour un mélange à 4 constituants est donné sur la figure 1.

Dans le cas de séquences ne comportant que des colonnes classiques et peu de constituants à séparer, ce problème peut être résolu par une méthode Branch and Bound (Pibouleau et Domenech (10)).

Dans le cas général où l'aspect combinatoire est très important, nous avons résolu le problème de la conception optimale, d'un point de vue topologique, des séquences de séparation au moyen d'un algorithme génétique standard pour lequel :

- le codage des structures de séparation est effectué sous la forme d'opérateurs élémentaires (+, -) agissant sur une pile ;
- la fonction d'adaptation correspond à l'évaluation technico-économique de la structure ;
- la stratégie de sélection s'effectue par tirage sur roue de loterie ;
- la reproduction se fait par croisement de deux structures, avec des contraintes de faisabilité ;

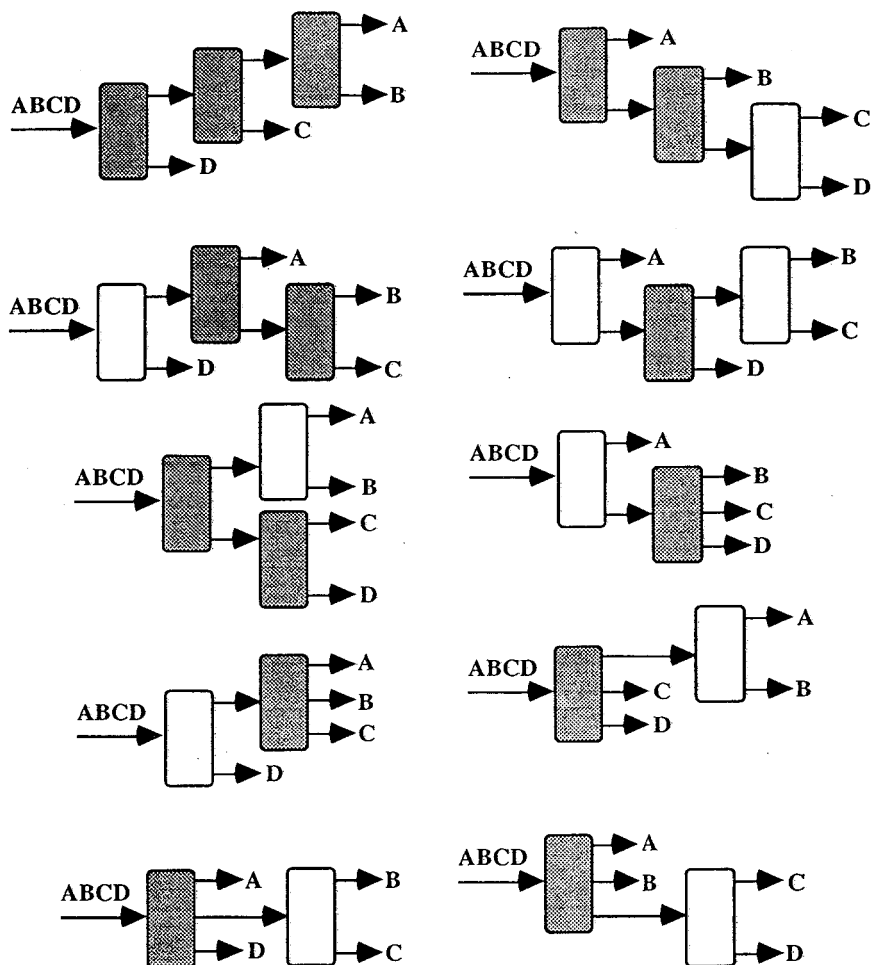


Figure 1. – Ensemble des séquences de séparations tranchées de 4 constituants.

– la mutation, définie par un taux de mutation, est réalisée aléatoirement sur des schémas de procédés.

La démarche est illustrée par trois exemples. Une étude portant sur l'évolution de la meilleure structure, de l'adaptation moyenne de la population et de son écart-type au cours de la recherche est présentée. Les paramètres caractéristiques de la méthode (taille de la population, taux de survie, taux de mutation) sont analysés. Enfin, une comparaison de cette méthode avec des approches alternatives, recuit simulé et recherche aléatoire, est présentée.

### 3. ASPECT COMBINATOIRE ET DÉNOMBREMENT DES STRUCTURES

Le nombre de séquences réalisant la séparation totale d'un mélange de  $n$  constituants en  $n$  produits purs et ne comportant que des séparateurs simples a été calculé par Thompson et King (13) :

$$S_n = \frac{(2(n-1))!}{n!(n-1)!}$$

Cette relation correspond à la définition des nombres de Catalan.

Lorsque les séparateurs simples ou complexes, avec un soutirage intermédiaire au maximum, sont acceptés le nombre de séquences est donné par (Floquet *et al.* (6)) :

$$S_n = \sum_{i=0}^{E\left(\frac{n-1}{2}\right)} \frac{(2n-2-i)!}{i!n!(n-2i-1)!}$$

Où  $E\left(\frac{n-1}{2}\right)$  est la partie entière de  $\left(\frac{n-1}{2}\right)$ .

Chaque terme de la sommation correspond au nombre de séquences comportant  $i$  séparateurs complexes. Ce nombre  $i$  ne peut excéder  $E\left(\frac{n-1}{2}\right)$ . En effet, en excluant l'alimentation générale, le nombre de courants présents dans une séquence comportant  $i$  séparateurs complexes et  $j$  séparateurs simples est de  $(2j+3i)$ . Ce nombre correspond aux  $n$  sorties de produits purs (hypothèse de séparation tranchée) et aux  $(i+j-1)$  courants de connexions entre les séparateurs. D'où :

$$2j+3i = n+i+j-1 \Rightarrow j+2i = n-1$$

Le nombre de séparateurs complexes présents dans une séquence est donc limité par  $E\left(\frac{n-1}{2}\right)$  (cas où  $j=0$ ).

Le nombre de séparateurs distincts comportant  $r$  sorties pour un mélange à  $n$  constituants, noté  $TS(n)_r$ , et permettant de générer l'ensemble des séquences est donné par les expressions suivantes :

$$TS(n)_2 = C_{n-1}^3 = \frac{n^3-n}{6} = \sum_{i=1}^{n-1} i(n-i)$$

$$TS(n)_3 = C_{n-2}^4$$

De manière générale :

$$TS(n)_r = C_{n-(r-1)}^{r+1} \quad \text{pour } n \geq r$$

Où  $C_p^q$  représente le nombre de combinaisons à  $q$  éléments pris parmi  $p$  sans répétition.

Sur la figure 1, les séparateurs distincts permettant de générer l'ensemble des séquences pour un mélange à 4 constituants sont représentés en grisé. Par comparaison avec l'exemple bien connu du voyageur de commerce, les vecteurs  $TS_2$  et  $TS_3$ , qui sont composés de tous les séparateurs distincts à 2 et 3 sorties et qui sont de taille  $TS(n)_2$  et  $TS(n)_3$ , respectivement, se comportent intuitivement comme la matrice des distances intervalles. Leur connaissance est nécessaire à la résolution du problème mais n'est pas suffisante vu la combinatoire du problème.

Le tableau 1 donne les valeurs de ces différentes expressions pour divers nombres de constituants à séparer.

TABLEAU 1  
*Nombre de séquences de séparations tranchées  
en fonction du nombre de constituants du mélange.*

Nombre de constituants à séparer	Nombre de séquences de séparation ne comportant que des colonnes de rectification classique	Nombre de séquences de séparation comportant des colonnes de rectification classiques et complexes	Nombre de séparateurs simples distincts $TS_2$	Nombre de séparateurs simples distincts $TS_3$
2	1	1	1	—
3	2	3	4	1
4	5	10	10	5
5	14	38	20	15
6	42	154	35	35
7	132	654	56	70
8	429	2871	84	126
9	1430	12925	120	210
10	4862	59345	165	330
14	742900	30142360	455	1365
16	9694845	717061952	680	2380



L'efficacité des diverses techniques d'optimisation peut être testée en analysant la part de l'espace de recherche réellement calculée et en s'intéressant au pourcentage d'exploration des vecteurs  $TS_2$  et  $TS_3$ .

#### 4. ALGORITHMES GÉNÉTIQUES

Les algorithmes génétiques traitent une population  $P(t)$  d'individus, appelés chromosomes, qui sont des solutions potentielles du problème posé et qui évoluent dans le temps (Alliot et Schiex (2)). Deux éléments : le codage des individus en entrée et la fonction d'adaptation (ou fitness) sont caractéristiques du problème à traiter. L'algorithme génétique qui est appliqué est entièrement indépendant du problème à résoudre et répond à la structure de tout programme évolutif. On peut ainsi décomposer tout algorithme génétique en 5 étapes :

1. Génération de la population initiale, soit aléatoirement, soit en appliquant des règles heuristiques.
2. Évaluation de l'adaptation de tous les individus de la population.
3. Sélection des individus survivants et se reproduisant éventuellement.
4. Reproduction des individus.
5. Retour à l'étape 2, tant que nécessaire.

L'étape 4 de reproduction se décompose pour chaque descendant en deux opérations génétiques élémentaires, mutation et croisement. Lors de la sélection, les chromosomes mis en jeu pour le croisement ainsi que la position de croisement (pour un croisement à un point), sont tirés aléatoirement. De même, la mutation d'un gène est déterminée aléatoirement (Goldberg (8)).

Divers paramètres sont à fixer pour le bon fonctionnement d'un algorithme génétique :

- taille de la population traitée (fixe au cours des générations);
- taux de survie de la population d'une génération à l'autre, il s'agit du pourcentage de chromosomes qui ne sont pas traités par les opérateurs de croisement ou de mutation;
- taux de mutation;
- test d'arrêt de l'algorithme.

Cependant, l'un des points les plus importants reste le codage du problème. Comme le souligne Goldberg (8), « le codage des données est un art et de cet art dépend le succès ou l'échec de la tentative ». Dans les paragraphes

suivants nous illustrerons ces divers points par une exemple didactique et nous détaillerons le codage des séquences de séparation.

## 5. APPLICATIONS DES ALGORITHMES GÉNÉTIQUES EN CONCEPTION DE SÉQUENCES DE SÉPARATION

### 5.1. Codage des séquences

Le codage des séquences de séparation classiques ou complexes ne peut s'effectuer directement par une suite de bits, d'entiers ou de réels. Qui plus est, un codage « en liste » devra respecter certaines contraintes :

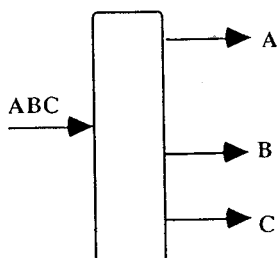
- la longueur des listes devra être fixe, afin de pouvoir employer un algorithme génétique classique ;
- le codage devra être tel que des schémas intéressants soient d'ordre et de longueur fondamentale courts. Une des conséquences de cette notion de « Building block » sera que des codages voisins devront représenter des structures de séparation topologiquement proches ;
- toutes les listes générées devront être admissibles, c'est-à-dire correspondre à une séparation faisable.

Divers types de codage ont déjà été formulés pour ce problème : représentation de Janicke (Bezzina *et al.* (3)), de Domenech *et al.* (5) ou de Floquet *et al.* (7), mais aucun ne semble pouvoir prendre en compte les contraintes ci-dessus. En définitive, aucun codage direct ne semble exploitable, c'est pourquoi il faut envisager un codage primaire représentant la structure et un codage intermédiaire des données pour le traitement par les opérateurs génétiques.

#### *Codage primaire des structures*

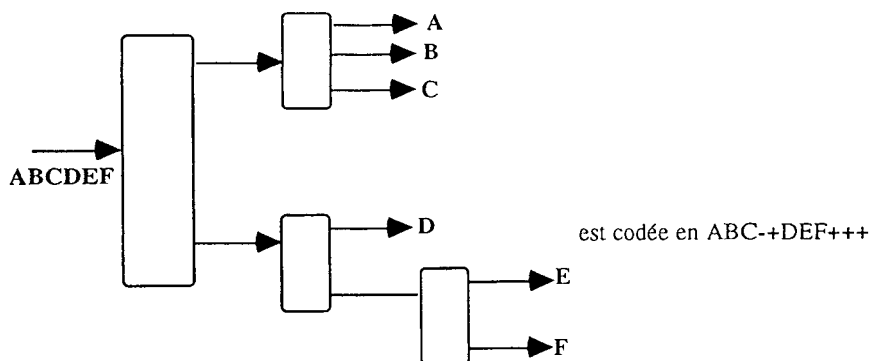
Le codage retenu est un codage sous forme de deux opérateurs élémentaires de séparation de courant, agissant sur la liste des constituants, qui se comporte comme une pile. Chaque opérateur agit sur les deux derniers éléments de la pile, le premier élément sortant au dessus de l'autre. Il s'agit du principe classique de la notation polonaise inversée. Ainsi l'opérateur + représente une séparation avec deux sorties. Par exemple AB+ représente une séparation avec A en haut, B en bas. De même, ABC++ représente la séparation de B (en haut) de C (en bas), précédée de la séparation A (en haut) du binaire BC (en bas).

Le concept de soutirage intermédiaire implique la définition du second opérateur – qui combiné avec l'opérateur + représente une colonne à trois sorties. Le codage ABC–+ représente la séparation suivante :



Le codage en (–+) d'un séparateur complexe permet de conserver des chromosomes de longueur constante quelle que soit la structure codée et le nombre de séparateurs complexes qu'elle comporte.

La séquence plus complexe suivante :



### *Codage secondaire des structures*

L'hypothèse selon laquelle la liste des constituants est invariante (par rapport à la volatilité relative, qui est la propriété physico-chimique utilisée pour ordonner les constituants) durant la séparation, permet de remplacer les symboles d'identification des constituants (A, B, C...) par le nombre de constituants précédant immédiatement l'opérateur de séparation. Ainsi, AB+ est représenté par 2+. De même, ABC–+ par 3–0+ et ABC–+DEF++ par 3–0+3+0+0+.

Cependant, pour des raisons pratiques, les opérateurs sont dissociés des nombres de constituants mis en jeu par ces opérateurs. Les trois codages précédents deviennent respectivement : 2+, 30 -+ et 30300-++++.

On remplace, enfin, pour le codage informatique, l'opérateur + par 1 et l'opérateur - par -1. Ceci donne pour les exemples précédents : 21, 30-11 et 30300-11111

## 5.2. Traitement des contraintes

On note que la contrainte de n'avoir que des chaînes de longueur égale et fixe est vérifiée. En effet, pour ce codage, tous les chromosomes auront une longueur égale à  $2(n-1)$  pour un mélange de  $n$  constituants à séparer.

La contrainte de voisinage structurel est, elle aussi, vérifiée et son efficacité sera démontrée au paragraphe suivant.

Enfin, la contrainte de conservation des schémas réalisables lors des générations est plus délicate à traiter. Les opérateurs génétiques vont s'appuyer sur un codage de travail, dit « codage somme », dérivé du codage secondaire des structures. Les principes du codage somme sont les suivants :

$$\begin{cases} C\Sigma(1) = CS(1) \\ C\Sigma(j) = C\Sigma(j-1) + CS(j) - 1 \quad j = 2 \text{ à } (n-1) \end{cases}$$

$$\Leftrightarrow \begin{cases} CS(1) = C\Sigma(1) \\ CS(j) = C\Sigma(j) - C\Sigma(j-1) + 1 \quad j = 2 \text{ à } (n-1) \end{cases}$$

Les éléments  $C\Sigma(j)$ , pour  $j = n$  à  $2(n-1)$ , sont constitués par les opérateurs de séparation (1 ou -1) du codage secondaire.

Le codage somme des trois exemples précédents est le suivant : 21, 32-11 et 32432-11111.

La justification d'un tel codage provient de ce qu'il permet d'obtenir aisément des solutions réalisables à l'issue de l'application des opérateurs génétiques. Ceci est rapidement justifié au paragraphe 5.3 par un exemple. De plus le passage d'un codage à l'autre est facile.

## 5.3. Opérateurs génétiques

Les opérateurs génétiques employés sont les opérateurs habituels, mutation aléatoire et croisement à un point sur les structures, directement accessibles au niveau du codage somme. Par exemple, concernant le croisement en

seconde position suivant :

$\{AB + C + D + E + F+\}$	Codage primaire des parents	$\{ABCDEF + + + + +\}$
$[21   111 \ 11111]$	Codage secondaire des parents	$[60   000 \ 11111]$
$[22   222 \ 11111]$	Codage somme des parents	$[65   432 \ 11111]$

Le premier descendant est le suivant :  $[22 | 432 \ 11111]$  ce qui correspond à la séquence de codage secondaire  $[21300 \ 11111]$  et de codage primaire  $AB+C+DEF+ + +$ .

Pour le second descendant, le codage somme  $[65 | 222 \ 11111]$  correspond à un codage secondaire  $[60-211 \ 11111]$  non admissible, la présence de nombre négatifs étant impossible, dans la mesure où un opérateur ne peut être précédé que par un nombre positif de constituants. Nous appliquons alors une procédure de normalisation qui consiste à ajouter au plus grand élément en position antérieure, ce nombre négatif. Sur l'exemple précédent, le codage secondaire normalisé du second descendant est  $[40011 \ 11111]$ , soit en codage primaire  $ABCD+ + +E+F+$ .

La figure 2 montre une représentation graphique de ce croisement.

Si ce même croisement, c'est-à-dire au même point, avait été réalisé sur le codage secondaire des parents, on aurait obtenu comme descendants :  $21000 \ 11111$  et  $60111 \ 11111$  pour lesquels le nombre de constituants mis en jeu n'est plus respecté. On aboutit plus facilement à des solutions réalisables à partir de l'application des opérateurs génétiques sur le codage somme plutôt que sur le codage secondaire.

La procédure de normalisation est quant à elle facilitée sur le codage secondaire, mais elle exige un passage permanent du codage somme au

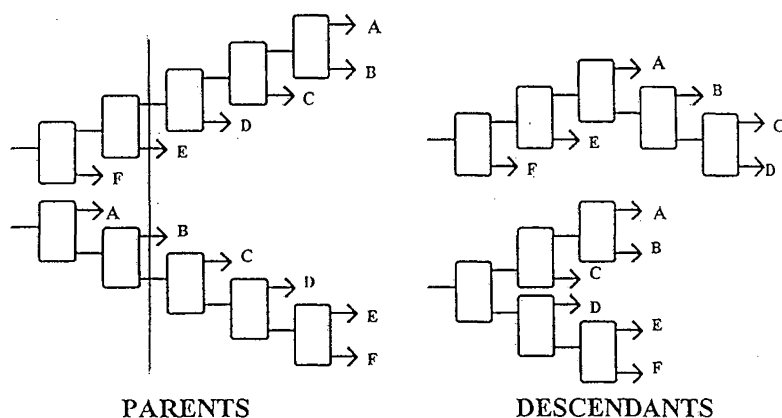


Figure 2. - Exemple de croisement de deux structures.

codage secondaire, pour lequel il est impossible d'appliquer les opérateurs génétiques.

Dans les exemples numériques présentés, le choix de la position du croisement est effectué aléatoirement parmi les  $(n - 2)$  positions possibles.

Pour ce qui est du phénomène de mutation, il est effectué aléatoirement sur les éléments du codage secondaire grâce aux deux procédures suivantes, pour les positions  $(n - 1)$  à  $2(n - 1)$  :

- un  $-1$  est changé en  $1$  dans le codage secondaire (on remplace une colonne complexe par deux colonnes simples),

- un  $1^*$ , où  $*$  joue aussi bien le rôle du symbole  $1$  ou du symbole  $-1$ , est changé en  $-1 \ 1$  dans le codage secondaire (on remplace deux colonnes simples par une colonne complexe ou bien on inverse une colonne simple et une colonne complexe).

Par exemple, l'application de la première règle sur la structure de codage [30300-11111] donne le code [3030011111], qui correspond à la séquence de séparation ABC++DEF+ + + schématisée sur la figure 3 a. De façon analogue, l'application de la seconde règle sur le deuxième  $1$  de ce même codage, fournit le code [30300-11-111], représentant la séquence de séparation ABC-+DEF -++ donnée sur la figure 3 b.

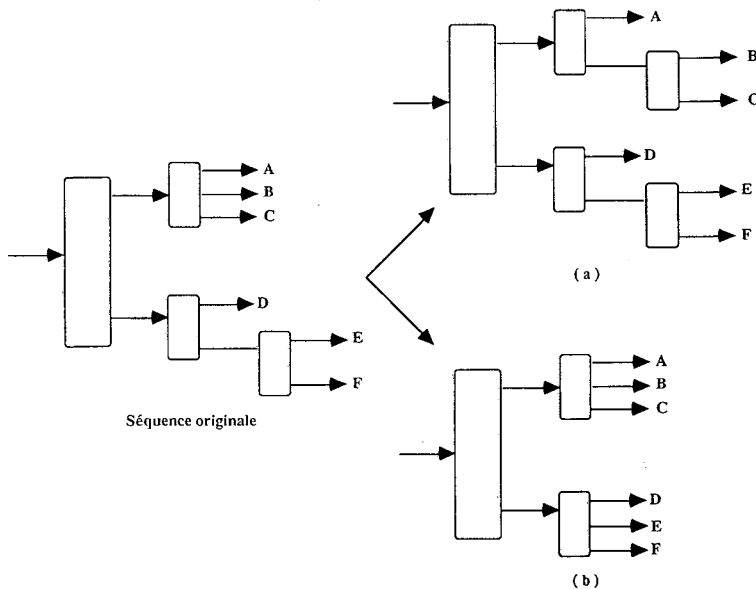


Figure 3. – Exemples de mutation. 3a Exemple de la première règle de mutation. 3b Exemple de la deuxième règle de mutation.

#### 5.4. Algorithme génétique choisi

Un algorithme génétique de type élitiste (Bicking *et al.* (4)) dans lequel seuls les individus les plus adaptés se reproduisent et sont conservés de générations en générations ne fonctionne pas sur ce type de problème. En effet, l'élitisme crée une harmonisation rapide de la population de structures qui étouffe les potentialités de la recherche. Nous avons donc retenu un algorithme génétique classique (Goldberg (8)) avec tirage sur roue de loterie, pour lequel des chromosomes ont une chance de croisement proportionnelle à leur adaptation et qui, par une fonction de partage (sharing) (Alliot et Schiex (2)), évite les problèmes de prédominance. Cette fonction donne deux fois plus de chance au meilleur élément de chaque population de se reproduire par rapport à un élément moyen de la population. Les étapes principales de l'algorithme sont illustrées sur la figure 4. Le seul test d'arrêt présent dans l'algorithme est le nombre (ITERMAX) de générations autorisées.

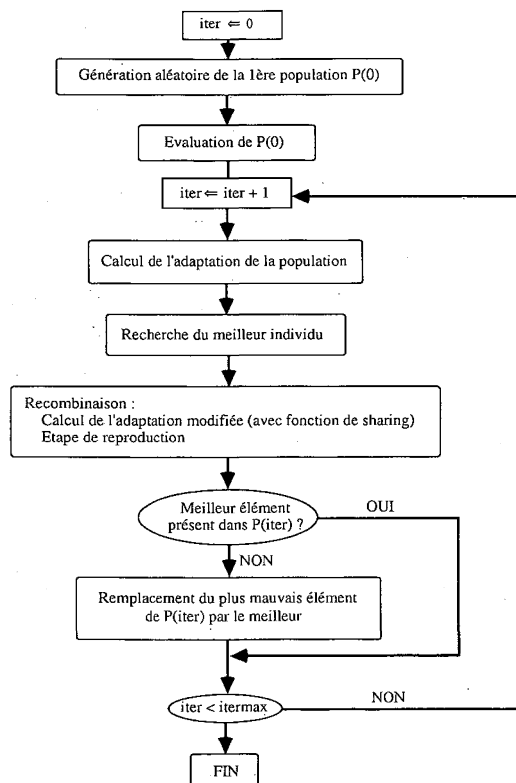


Figure 4. – Algorithme génétique avec tirage sur roue de loterie (Goldberg (7)).

## 6. EXEMPLES D'ILLUSTRATION

Par la suite, nous désignerons par « meilleure solution trouvée », la séquence de plus faible coût générée. C'est cette solution qui serait alors retenue pour chaque essai.

### 6.1. Exemple préliminaire

Il s'agit du problème de synthèse de la structure optimale d'un point de vue technico-économique d'une séquence de distillation d'un mélange équimolaire de 10 hydrocarbures ( $n$ -C7 à  $n$ -C16). Cet exemple sera noté **C10** par la suite. Le débit total du mélange est de 1 000 kmol/h, la conception et le dimensionnement des appareillages ont été effectués par une méthode Short Cut classique (Pibouleau *et al.* (9)) et le coût de chaque séquence évalué par des corrélations (Pibouleau et Domenech (11)).

Nous avons choisi de présenter cet exemple didactique afin de pouvoir générer de façon exhaustive, et en un temps de calcul raisonnable, la courbe de distribution pseudo-gaussienne des coûts, donnée sur la figure 5, ce qui permet de tester l'efficacité et la sûreté de la démarche de résolution adoptée.

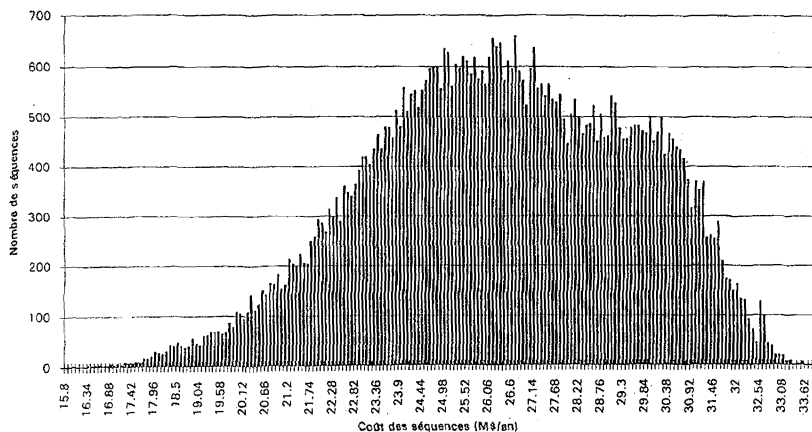


Figure 5. – Courbe de distribution des coûts des séquences de séparation exemple C10.

Dans tous les cas traités (la population initiale est générée aléatoirement), avec divers paramétrages de l'algorithme génétique, la solution optimale, qui ici est bien le minimum du cas traité et qui correspond à un coût de 15,848 M\$/an et au schéma de la figure 6, est obtenue. Ce coût optimal



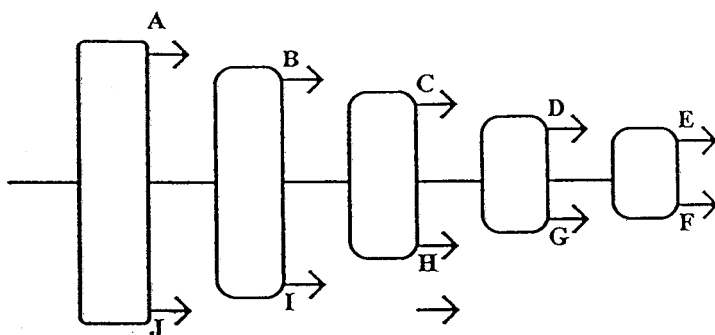


Figure 6. - Séquence solution de l'exemple C10.

correspond à une réduction de l'ordre de 50 % par rapport à la valeur moyenne de l'initialisation et de 38 % par rapport au coût moyen.

Afin d'étudier les paramètres sensibles de l'algorithme génétique au problème de conception, nous avons fait varier la taille de la population (20, 50, 100 et 200 individus) et le taux de survie (10 %, 20 %, 40 %, 60 %, 80 %) pour ce même exemple avec seulement des colonnes sans soutirage intermédiaire. Nous avons étudié la variation du meilleur élément de la population, l'adaptation moyenne et l'écart-type par rapport à la moyenne en fonction de la taille de la population (resp. du taux de survie).

L'évolution de l'écart-type au cours des générations reste sensiblement constante quelle que soit la taille de la population initiale ce qui montre qu'il n'y a pas dégénérescence vers un seul type d'individu. L'adaptation moyenne de l'ensemble des individus décroît de manière quasi monotone au cours des générations pour toutes les tailles de population étudiées, ainsi que le meilleur élément ce qui démontre la stabilité de la méthode à taux de survie constant (égal à 50 % dans tous les cas traités). On obtient des résultats similaires, lorsqu'avec une taille de population initiale fixée (égale à 60), on fait évoluer le taux de survie de 0,1 à 0,8.

Nous avons réalisé une étude systématique concernant la rapidité de l'algorithme et la qualité de la solution en faisant évoluer la taille  $n_p$  de la population, le taux de survie  $d_g$  et le taux de mutation  $d_m$ . Les meilleurs paramètres de fonctionnement correspondent à  $n_p=60$ ,  $d_g=0,5$  et  $d_m=0,1$ .

La détermination de la meilleure structure avec des colonnes complexes a été effectuée avec ces valeurs des paramètres.

Les figures 7a, 7b et 7c montrent l'évolution du meilleur élément, de l'adaptation moyenne des individus et de leur écart-type au cours des

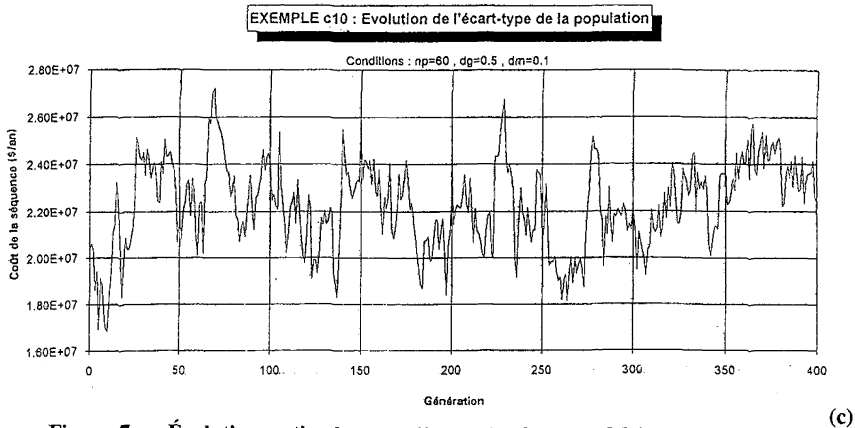
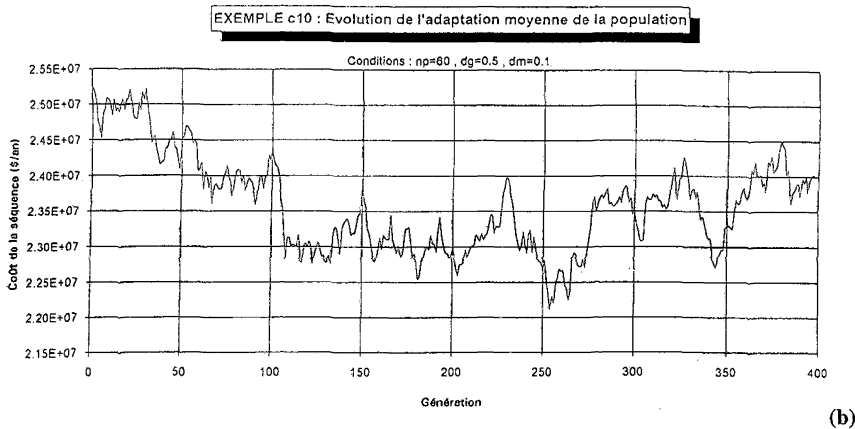
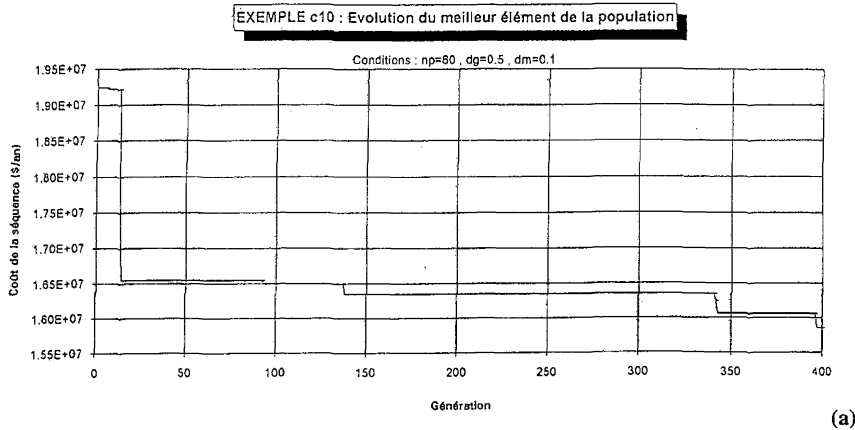


Figure 7. - Évolution optimale pour l'exemple C10. 7a Meilleur élément, 7b adaptation moyenne, 7c écart-type.

génération. On remarque ainsi que si l'optimum global (fig. 6) est obtenu en 400 générations, une séquence dont le coût est voisin à 4 % de celui de la séquence optimale est obtenue très rapidement (une quinzaine de générations environ), ce qui prouve une certaine sûreté de l'algorithme.

L'étude de l'écart-type permet de vérifier la diversité de la population nécessaire à une bonne évolution sans dégénérescence vers un seul type d'individu non optimal. Le suivi de l'adaptation moyenne des diverses générations montre quant à elle le gain réalisé à chaque étape.

## 6.2. Exemples de grande taille

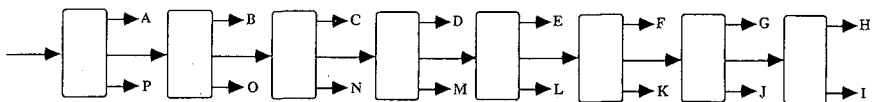
Nous avons réalisé une étude similaire sur des exemples plus complexes. Les deux exemples retenus sont :

- exemple **C14** : mélange équimolaire de 14 hydrocarbures ( $n$ -C5 à  $n$ -C18) avec un débit de 1000 kmol/h. Ce problème comporte 30142360 solutions.

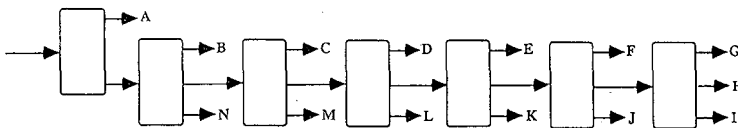
- exemple **C16** : mélange équimolaire de 16 hydrocarbures ( $n$ -C5 à  $n$ -C20) avec un débit de 1000 kmol/h. Ce problème comporte 717061952 solutions.

L'espace des solutions de l'exemple C14 est 500 fois plus important que celui de l'exemple préliminaire, celui de l'exemple C16 l'est 10000 fois plus.

Les meilleures séquences trouvées sur l'ensemble des essais ainsi que leurs coûts sont reportés sur la figure 8. On remarque pour les deux exemples la présence de nombreuses colonnes avec un soutirage intermédiaire ce qui réduit le nombre d'opérations de séparation.



Exemple C16 - Coût : 22,78 M\$



Exemple C14 - Coût : 18,86 M\$

Figure 8. - Meilleures séquences trouvées pour les exemples C14 et C16.

Pour prendre en compte l'aspect aléatoire de la méthode, des séries de 50 essais ont été réalisées pour chaque exemple, avec un taux de mutation égal à 0,1, et on s'intéressera donc aux valeurs moyennes et à l'intervalle de confiance symétrique au seuil de 5 % sur la meilleure solution trouvée à chaque essai. Rappelons que cet intervalle de confiance est donné par la relation classique :

$$\bar{x} \pm \frac{s}{\sqrt{ness}} t_{1-\frac{\alpha}{2}}$$

où  $\bar{x}$  est la moyenne des coûts des meilleures séquences trouvées et  $s$  leur écart-type. Enfin,  $t_{1-\frac{\alpha}{2}}$  est la valeur donnée par la table de Student au niveau de probabilité  $(1 - \frac{\alpha}{2})$  avec  $ness$  degrés de liberté ; pour  $\alpha = 5\%$  et  $ness = 50$ ,  $t_{1-\frac{\alpha}{2}} \approx 2$ .

Les résultats obtenus sont reportés sur le tableau 2. On constate évidemment que le coût de la meilleure séquence trouvée ainsi que le coût moyen des meilleures séquences de chacun des 50 essais croissent en fonction de la taille du problème. Il en est de même de l'intervalle de confiance. *A contrario*, le taux d'exploration des vecteurs TS<sub>2</sub> et TS<sub>3</sub>

TABLEAU 2

*Résultats obtenus pour l'algorithme génétique  
(taux de mutation égal à 0,1 - 50 essais).*

Taille de la population	Taux de Survie	Nombre de générations	Coût de la meilleure séquence trouvée (M\$)	Coût moyen des meilleures séquences trouvées (M\$)	Intervalle de confiance (% moy.)	Nombre de séquences calculées	Exploration de TS <sub>2</sub> (%)	Exploration de TS <sub>3</sub> (%)
<b>Ex. C10</b>								
60	0,5	100	16,0	16,6	0,7	3661	72	55
60	0,5	30	<b>15,84</b>	17,36	1,0	1139	56	36
60	0,25	30	<b>15,84</b>	16,9	1,0	1620	60	41
<b>Ex. C14</b>								
60	0,5	60	20,78	22,8	1,4	2216	33	16
100	0,5	100	19,98	22,35	1,3	6096	46	25
100	0,5	500	<b>18,86</b>	21,5	1,4	30075	59	37
100	0,25	100	19,98	21,7	1,0	8705	50	29
<b>Ex. C16</b>								
60	0,5	200	26,41	29,6	1,5	7254	36	17
100	0,5	200	22,78	28,2	1,4	12090	42	21
200	0,5	500	24,54	28,37	2,7	70182	47	26

décroît, ce qui conforte la capacité des algorithmes génétiques à traiter des problèmes fortement combinatoires. Par ailleurs, la faible valeur de l'intervalle de confiance dans tous les cas et le petit écart constaté entre le coût moyen des meilleures séquences trouvées à chaque essai et celui de la meilleure séquence trouvée sur les 50 essais confirment la stabilité de la procédure. Enfin, le taux d'exploration de l'espace total des solutions est au plus égal à 0,001 pour l'ensemble C14 et à 0,0001 pour le cas C16.

### 6.3. Autres approches envisagées

Nous avons mis en œuvre une méthode de recuit simulé et une méthode de recherche aléatoire afin de comparer les résultats avec ceux de l'algorithme génétique.

Un algorithme de recuit simulé (Floquet *et al.* (7)) a été appliqué avec génération aléatoire du point de départ. L'algorithme utilisé est un algorithme classique avec un schéma de refroidissement géométrique reliant la température du  $k$ -ième palier à celle du  $(k-1)$ -ième par :  $T_k = 0.95 T_{k-1}$ . D'autres schémas de décroissance de la température sont disponibles dans la littérature (Aarts *et al.* (1), Van Laarhoven (14)) et pourraient donner de meilleurs résultats. Les résultats donnés dans le tableau 3 sont issus de séries de 50 essais.

TABLEAU 3  
Résultats obtenus pour le recuit simulé (50 essais).

Température d'initialisation	Coût de la meilleure séquence trouvée (M\$)	Coût moyen des meilleures séquences trouvées (M\$)	Intervalle de confiance (% moy.)	Nombre de séquences calculées	Exploration de TS <sub>2</sub> (%)	Exploration de TS <sub>3</sub> (%)
<b>Ex. C10</b>						
200	16,14	18,84	2,3	941	12	5
10000	<b>15,84</b>	19,25	2,5	1327	13	5
<b>Ex. C14</b>						
200	22,23	25,0	1,8	1364	7	1
10000	19,91	24,56	1,7	2028	8	2
<b>Ex. C16</b>						
200	26,02	31,65	2,3	1542	5	1
10000	25,54	31,04	2,6	1796	5	1

La seconde méthode envisagée est une recherche aléatoire pure. Pour cela, un tirage aléatoire de 2000 individus (ordre de grandeur du nombre de séquences calculées par la méthode du recuit simulé) a été réalisé pour les mêmes exemples. Les moyennes sont également calculées sur 50 essais et un récapitulatif des résultats est fourni par le tableau 4.

TABLEAU 4  
*Résultats obtenus pour la recherche aléatoire (2000 tirages – 50 essais).*

Cas traité	Coût de la meilleure séquence trouvée (M\$)	Coût moyen des meilleures séquences trouvées (M\$)	Intervalle de confiance (% moy.)	Nombre de séquences calculées	Exploration de TS <sub>2</sub> (%)	Exploration de TS <sub>3</sub> (%)
Ex. C10	16,65	19,95	2,5	2000	15	6
Ex. C14	20,59	25,83	2,2	2000	7	2
Ex. C16	27,15	31,9	2,0	2000	6	1

La comparaison des résultats fournis par les diverses approches montrent que l'algorithme génétique est plus fiable que les autres méthodes. En effet, pour chaque série d'essais, la valeur moyenne des meilleures solutions trouvées sur les 50 essais est plus proche du coût de la meilleure séquence trouvée sur l'ensemble des essais. De plus, nous avons constaté que l'algorithme génétique détectait plus souvent la solution de coût le plus faible. La bonne performance relative de la recherche aléatoire, concernant la meilleure solution trouvée et les exemples de faible combinatoire C10 et C14, tient principalement au fait que 100000 tirages ( $50 \times 2000$ ) ont été effectués, ce qui correspond à une large exploration de l'espace de recherche. La compétitivité relative de la recherche aléatoire montre également que les paramètres de réglage (taille de population, température initiale...) ont une influence sur la qualité de la solution trouvée aussi bien pour l'algorithme génétique que pour le recuit simulé.

Les algorithmes génétiques paraissent avoir un caractère aléatoire moindre que la méthode du recuit simulé au regard de l'intervalle de confiance qui est plus restreint pour les algorithmes génétiques. Ceci peut s'expliquer principalement par la notion de parallélisme implicite qu'ils présentent en travaillant sur des populations de solutions potentielles et non sur un unique individu.

Pour un nombre de séquences réellement calculées sensiblement équivalent, l'algorithme génétique permet une meilleure exploration de l'espace de recherche au regard du nombre de séparateurs distincts mis en jeu dans ces séquences (exploration plus importante des vecteurs  $TS_2$  et  $TS_3$ ).

Concernant les temps de calcul, ceux-ci sont directement corrélés avec le nombre de séquences calculées. La tâche la plus pénalisante du point de vue du temps calcul est la phase de simulation et d'évaluation économique de la séquence. A titre d'exemple, le temps CPU nécessaire pour le calcul d'un millier de séquences est de 3s sur une station IBM RISC6000.

Il faut noter enfin qu'une optimisation globale des séquences de séparation, c'est-à-dire de leurs structures et de leurs conditions opératoires, semble réalisable par des algorithmes génétiques. En effet, il apparaît actuellement très délicat de définir un codage pour la méthode du recuit simulé qui permettrait de prendre en compte des variables mixtes, et de définir des transformations conduisant à des configurations voisines d'une configuration donnée. Pour ce qui est des algorithmes génétiques, il semble possible de mêler des gènes codant des variables entières de topologie à des gènes codant des variables réelles traduisant des conditions opératoires, sous forme de binaires par exemple. Des opérateurs génétiques similaires pourraient de plus être appliqués aisément.

## 7. CONCLUSIONS

Le génie des procédés, et plus particulièrement la conception optimale des procédés, paraît être un domaine d'application privilégié des algorithmes génétiques.

En effet, en synthèse de procédés, il est extrêmement rare de disposer de l'expression analytique du problème d'optimisation à résoudre, dans la mesure où il se formule à partir de modèles de simulation numériques des diverses opérations unitaires constituant l'unité à concevoir, et donc l'analyse des propriétés globales ou même locales de convexité est impossible. On peut donc légitimement penser, et l'expérience l'a montré, que toute solution numérique d'un problème de conception optimale de procédé sera fortement dépendante de l'initialisation de procédures de programmation non linéaire ou de programmation en variables mixtes. De plus, l'obtention d'une solution globale n'est jamais garantie.

Par ailleurs, ces problèmes de conception sont fortement combinatoires, même dans des cas classiques (séquences de séparation, réseaux d'échangeurs

de chaleur, intégration thermique dans un procédé donné, ordre de lancement des produits dans un atelier de production par lots), ce qui rend très délicate, voire même impossible, eu égard au temps de calcul, l'utilisation de procédures d'énumération de type branch and bound ou de techniques de programmation mixte et d'optimisation globale.

Il faut néanmoins remarquer que l'implémentation d'algorithmes génétiques soulève des difficultés inhérentes au codage des solutions admissibles, vis-à-vis de contraintes physico-chimiques et/ou technologiques. Le cas des séquences de séparation a montré que ces problèmes de codage constituaient une tâche non triviale. Toutefois, compte tenu des retombées économiques que l'on peut attendre de l'obtention d'une solution globale, cet effort de codage sera largement récompensé.

#### BIBLIOGRAPHIE

1. E. H. L. AARTS et J. KORST, *Simulated annealing and Boltzmann machines*, John et Wiley & fils, 1989.
2. J. M. ALLIOT et T. SCHIEUX, *Intelligence artificielle et informatique théorique*, Cepadues, 1993.
3. M. BEZZINA, L. PIBOULEAU et S. DOMENECH, A statistical approach to the synthesis of separation sequences, *Comp. & Chem. Eng.*, 1989, 13, (6), p. 651-660.
4. F. BICKING, C. FONTEIX, J. P. CORRIOU et I. MARC, Global optimization by artificial life: A new technique using genetic algorithms, *RAIRO RO*, 1994, 28, (1), p. 23-36.
5. S. DOMENECH, L. PIBOULEAU et P. FLOQUET, Dénombrement de cascades de colonnes de rectification complexes, *Chem. Eng. J.*, 1991, 45, p. 149-164.
6. P. FLOQUET, S. DOMENECH, L. PIBOULEAU et S. M. ALY, Some complements in combinatorics of sharp separation system synthesis, *AIChE J*, 1993, 39, (6), p. 975-978.
7. P. FLOQUET, L. PIBOULEAU et S. DOMENECH, Separation sequence synthesis: how to use Simulated Annealing?, *Comp. & Chem. Eng.*, 1994, 18, (11/12), p. 1141-1148.
8. D. A. GOLDBERG, *Algorithmes génétiques*, Addison-Wesley, 1994.
9. L. PIBOULEAU, A. SAID et S. DOMENECH, Synthesis of optimal and near-optimal distillation sequences by a bounding strategy, *Chem. Eng. J.*, 1983, 27, p. 9-19.
10. L. PIBOULEAU et S. DOMENECH, Une procédure arborescente pour la séparation de mélanges complexes dans l'industrie chimique, *RAIRO RO*, 1985, 19, (1), p. 35-56.
11. L. PIBOULEAU et S. DOMENECH, Discrete and continuous approaches to the optimal synthesis of distillation sequences, *Comp. & Chem. Eng.*, 10, 1986, p. 479-491.
12. G. STEPHANOPOULOS, B. LINHOFF et A. SOPHOS, Synthesis of heat integrated distillation sequences, *IchemE Symp. Serie*, 1981, 74, p. 111.
13. R. W. THOMPSON et C. J. KING, Systematic synthesis of separation schemes, *AIChE J.*, 1972, 18, p. 941.
14. P. J. M. Van Laarhoven et E. H. L. AARTS, *Simulated annealing: Theory and applications*, Kluwer Acad. Pub., 1989.