

50376  
1981  
226

50376  
1981  
226

N° d'ordre : 922

THESE

PRESENTEE

A L'UNIVERSITE DE LILLE

POUR OBTENIR

LE TITRE DE DOCTEUR <sup>3<sup>e</sup> cycle</sup> ~~INGENIEUR~~

Spécialité : PHYSIQUE NUCLEAIRE

Option : PHYSIQUE DES REACTEURS



**Théophile MISSE NSEKE**

**UNE METHODE D'OPTIMISATION BOOLEENNE  
DU REPOSITIONNEMENT D'UN REACTEUR NUCLEAIRE**

Soutenu le 15 octobre 1981

**MM. BREZINSKI**

Président

**DUMAS  
FIOROT  
KAVENOKY  
MIGNOT**

Examineurs

Le présent travail a été effectué au Centre d'Etudes Nucléaires de Saclay dans le Service d'Etude des Réacteurs et de Mathématiques Appliquées (S.E.R.M.A.).

Je remercie,

Monsieur C. BREZINSKI de me faire l'honneur de présider le Jury de ma thèse,

Monsieur F. MIGNOT d'avoir accepté d'être mon Directeur de Thèse,

Monsieur J.C. FIOROT d'avoir accepté de juger ce travail,

Je remercie tout particulièrement,

Monsieur A. KAVENOKY de m'avoir permis d'entreprendre ce travail en m'accueillant au sein de la Section de Physique Mathématique du SERMA qu'il dirige,

Monsieur M. DUMAS qui par ses conseils et ses encouragements m'a permis de mener à bien ce travail ; qu'il trouve ici ma sincère reconnaissance,

L'ensemble du groupe de Physique Mathématique pour son aimable collaboration.

## TABLE DES CHAPITRES

INTRODUCTION

CHAPITRE I - ENONCE DU PROBLEME ET SA FORMULATION EN VARIABLES BOOLEENNES

CHAPITRE II - LINEARISATION DE  $K_{\text{eff}}(u)$  et  $p(u)$

CHAPITRE III - UNE METHODE D'ENUMERATION IMPLICITE : L'ALGORITHME DE GEOFFRION

CHAPITRE IV - MISE EN OEUVRE DE L'ALGORITHME DE GEOFFRION

CHAPITRE V - RESULTATS NUMERIQUES

CONCLUSION

TABLE DES ANNEXES

ANNEXE A - LA CONTRAINTE DU FILTRE DE BALAS

ANNEXE B - QUELQUES NOTIONS ELEMENTAIRES DE LA THEORIE DES GRAPHES

ANNEXE C - VOISINAGE DISCRET DE L'OPTIMUM CONTINU DONNE PAR LA MAP

ANNEXE D - QUELQUES ORGANIGRAMMES

REFERENCES

## INTRODUCTION

Nous avons tenté au cours de ce travail de généraliser les méthodes développées par DUMAS [5] pour résoudre le problème du repositionnement optimal du combustible dans le coeur d'un réacteur nucléaire.

Rappelons brièvement comment se pose ce problème. Dans le coeur d'un réacteur nucléaire, les assemblages combustibles évoluent pendant des périodes de fonctionnement appelées cycles. A la fin de chaque cycle, les éléments les plus usés sont remplacés par des assemblages neufs. Au début du cycle suivant, les assemblages sont repositionnés dans le coeur. Les ingénieurs désirent alors une configuration de chargement ayant le facteur de multiplication effectif,  $k_{eff}$ , maximum - plus ce facteur est élevé, plus loin on repousse la limite neutronique - pour un facteur de forme,  $f$ , de la distribution de puissance -  $f$  étant le rapport de la puissance maximale à la puissance moyenne - inférieur à une borne supérieure donnée. Parfois - et ce pour faire face aux perturbations observées lors du fonctionnement - les ingénieurs aimeraient disposer d'une configuration de chargement ayant le plus petit facteur de forme possible. Ils exigent éventuellement que cette configuration ait un  $k_{eff}$  supérieur à une borne inférieure donnée.

Optimiser le repositionnement du combustible, c'est donc résoudre un problème d'affectation car il s'agit d'attribuer à chaque emplacement du coeur un et un seul assemblage.

Malheureusement, ce problème d'affectation est non linéaire car  $k_{eff}$  et  $f$  sont obtenus à partir des équations de la diffusion multigroupe.

Ce problème d'affectation est d'autant plus difficile que ni le  $k_{eff}$  ni le facteur de forme ne s'expriment de manière analytique en fonction d'un paramètre caractérisant les assemblages ou leur place.

De nombreux auteurs se sont attaqués à la résolution de ce problème. Pour la plupart, ils s'affranchissent de la non analyticit  de  $k_{eff}$  en attribuant au coeur une sym trie radiale et en traduisant les  volutions des neutrons   l'aide de l' quation de la diffusion   un groupe. Ils aboutissent alors   un probl me d'affectation quadratique pour lequel il existe des algorithmes performants.

Malheureusement, leurs simplifications sont abusives car l'exp rience montre que l' quation de la diffusion   un groupe d'nergie - voir [4] par exemple - est insuffisante pour traduire les  volutions des neutrons.

Au SERMA (Service d'Etudes des R acteurs et de Math matiques Appliqu es)   Saclay, un code de r solution des  quations de la diffusion multigroupe   l'aide de la m thode des  l ments finis a  t  mis au point. Le code Neptune [13]. Ce code a permis d'obtenir une repr sentation fid le du coeur, ainsi qu'un calcul aussi d taill  que pr cis du  $k_{eff}$  et de la distribution de puissance. DUMAS [5] a montr  d'autre part que m me disposant d'un param tre exprimant continuellement les mouvements des assemblages, ni le  $k_{eff}$ , ni le facteur de forme de la distribution de puissance ne poss de les propri t s de convexit  requises pour assurer la convergence des algorithmes de programmation math matique vers un optimum global. De plus on peut noter que m me si

l'on ne désire qu'un optimum local, le coût élevé en temps de calcul de l'évaluation du  $K_{\text{eff}}$  et  $f$  ( $10^5$  plus que cosinus par exemple) hypothèque lourdement l'utilisation de toute méthode nécessitant leur calcul fréquent.

La MAP (Method of Approximale Programming) [9] qui ne nécessite que la résolution de programmes linéaires continus s'impose. Elle ne permet cependant que de trouver des solutions "idéales" au problème. Voir ROBEAU [18]. Elle suppose aussi que l'on dispose de développements au premier ordre du  $K_{\text{eff}}$  et de la distribution de puissance. Bien que les résultats de MIGNOT [15] ne s'appliquent pas dans ce cas, l'utilisation formelle de ses résultats permet d'obtenir une linéarisation qui, compte tenu des précisions demandées conduit à des résultats satisfaisants.

Appliquant une méthode de BRANCH and BOUND pour approcher la solution "idéale" de la MAP, DUMAS [5] a obtenu des repositionnements remarquables. Cependant, son étude repose sur l'hypothèse que les assemblages sont de même type.

En nous situant dans le cadre de son étude, nous avons essayé de généraliser ses résultats au repositionnement d'un coeur contenant des assemblages de divers types.

Si donc,  $n$  désigne le nombre total de positions dans le coeur ainsi que le nombre d'assemblages à repositionner, nous représentons la position d'un assemblage  $i$  par  $n$  variables booléennes  $u_{ij}$  telles que :

$$u_{ij} = 1 \quad \text{si l'assemblage } i \text{ est à la place } j$$

$$u_{ij} = 0 \quad \text{si non}$$

Nous remplaçons ainsi la détermination d'une configuration parmi  $n!$  par la détermination de  $n^2$  variables booléennes.

Nous affranchissant de la contrainte d'intégrité, nous calculons des développements au premier ordre du  $K_{\text{eff}}$  et de la distribution de puissance.

Ces développements nous permettent alors soit de faire une recherche directe en résolvant une suite de programmes linéaires booléens au voisinage des points de linéarisation, ce qui nous donne une configuration au moins sous optimale du problème initial, soit d'appliquer la MAP et d'approcher l'optimum continu obtenu de manière discrète.

La formulation proposée et les méthodes de résolutions envisagées soulèvent de nombreux problèmes théoriques et numériques. Nous espérons que la solution que nous leur avons apportée est satisfaisante.

## CHAPITRE I - ENONCE ET FORMULATION DU PROBLEME

Le coeur du réacteur est représenté par un domaine à deux dimensions  $\Omega$ , formé de zones rectangulaires, les assemblages  $\omega_k$ . Voir figure I.1.

En régime permanent, les équations de la diffusion multigroupe traduisent le bilan neutronique.

Chaque assemblage étant représenté pour un groupe d'énergie donné par :

son coefficient de diffusion  $D_g(\omega_k)$ ,

sa section efficace macroscopique totale  $\Sigma_{tg}$ ,

de fission  $\Sigma_f(\omega_k)$ ,

de ralentissement du groupe

$g$  vers le groupe  $g'$   $\Sigma_R(g \rightarrow g')(\omega_k)$ ,

son spectre de fission  $\chi_g(\omega_k)$ ,

$m$  désignant le nombre total de groupes d'énergie,  $\Gamma$  la frontière du coeur (le réflecteur), les équations de la diffusion s'écrivent :

$$(E) \quad \left\{ \begin{array}{l} -\operatorname{div}(D_g \operatorname{grad} \phi_g) + \Sigma_{tg} \phi_g - \sum_{g'=g+1}^m \Sigma_R(g \rightarrow g') \phi_{g'} = \frac{\chi_g}{K_{\text{eff}}} \sum_{g'=1}^m (v \Sigma_f)_{g'} \phi_{g'} \\ \phi_g / \Gamma = 0 \\ g = 1, m \end{array} \right.$$

$K_{\text{eff}}$  est appelé le facteur de multiplication effectif,  $\phi_g$  étant le flux de neutrons du groupe  $g$  dans le coeur.

L'état du coeur étant caractérisé par son  $K_{\text{eff}}$ , et sa distribution de puissance  $p$ ,  $P = (p_k)_{k=1,n}$  définie pour chaque assemblage  $k$  par :

$$p_k = \frac{\operatorname{mes}(\Omega)}{\operatorname{mes}(\omega_k)} \frac{\int_{\Omega} \sum_{q=1}^m (K \Sigma_f)_q \phi_q d\omega_k}{\int_{\Omega} \sum_{q=1}^m (K \Sigma_f)_q \phi_q d\Omega}$$

$n$  étant le nombre d'emplacements.



### 1.1. Enoncé du problème

En début de cycle on dispose d'un lot de  $n$  assemblages aux caractéristiques physiques fixées que l'on désire affecter à  $n$  emplacements du coeur. Une configuration du coeur sera une affectation des  $n$  assemblages dans les  $n$  emplacements. Il y a  $n!$  configurations possibles. Chaque configuration sera caractérisée par son  $K_{eff}$  et sa distribution de puissance calculée à l'aide des équations de la diffusion.

Le repositionnement optimal du combustible en début de cycle sera celui donné par la configuration qui réalise l'un des deux critères suivants :

critère 1 : minimiser le facteur de forme  $f$ ,  $f = \max_k p_k$  parmi toutes les configurations possibles,

critère 2 : maximiser le  $K_{eff}$ , pour un facteur de forme donné parmi toutes les configurations possibles.

### 1.2. Hypothèses

a) Les assemblages sont considérés comme homogènes. Nous supposons ainsi que les assemblages sont usés uniformément. Cela nous permet de définir la position d'un assemblage dans le coeur par l'emplacement qu'il occupe.

Cette hypothèse n'est pas tout à fait conforme à la réalité. En réalité les assemblages ne sont pas usés uniformément. Il faudrait pour tenir compte de l'usure faire intervenir des rotations de  $\pi$  ou  $\pi/2$  autour du centre de l'assemblage en le divisant par 2 ou 4 parties homogènes.

Cette possibilité n'a pas été étudiée.

b) Au cours du cycle on peut intervenir sur l'évolution du combustible au moyen d'un contrôle.

c) Nous dissocions la gestion du contrôle de celle du combustible, c'est-à-dire que si le repositionnement en début de cycle est optimal au sens qu'il vérifie l'un des critères 1 ou 2, alors le cycle sera dit optimal.

Cette hypothèse est réaliste. On observe d'une part que la durée du cycle est grande quand le  $K_{eff}$  en début de cycle est grand, d'autre part qu'une gestion du contrôle déterminée par la méthode de HALING [10] ne dégrade pas trop la distribution de puissance, et qu'elle est d'autant meilleure que le facteur de forme est petit.

## 2. FORMULATION DU PROBLEME

Pour les deux critères 1,2, la seule variable des problèmes d'optimisation est la

configuration du coeur. Celle-ci est une permutation d'ordre  $n$ . Nous la représenterons par une matrice de permutation d'ordre  $n$ ,

$$u = (u_{ij})_{\substack{j=1,n \\ i=1,h}} \quad \text{avec}$$

$$u_{ij} = 1 \quad \text{si l'assemblage } j \text{ est à l'emplacement } i$$

$$u_{ij} = 0 \quad \text{si non}$$

Nous allons formuler les problèmes de manière à y faire apparaître explicitement la variable  $u$ .

### 2.1. Formulation en variables booléennes

Soit  $\mathcal{U}$  l'ensemble des matrices de permutation d'ordre  $n$

$$\mathcal{U} = \{ u = (u_{ij}) \in \mathbb{R}^{n^2}, \quad \sum_{i=1}^n u_{ij} = 1 \quad j = 1, n, \quad ; \quad \sum_{j=1}^n u_{ij} = 1 \quad i = 1, n, \quad ; \quad u_{ij} = 0 \text{ ou } 1 \}$$

Posons pour chaque assemblage  $\omega_j$

$$D_g^j = D_g(\omega_j), \quad \Sigma_{tg}^j = \Sigma_{tg}(\omega_j), \quad \chi_g^j = \chi_g(\omega_j) \dots \text{etc}$$

Désignons par  $d_g^i, \sigma_{tg}^i, x_g^i \dots$  etc les caractéristiques physiques de l'assemblage affecté à l'emplacement  $i$ . Nous avons :

$$d_g^i = \sum_{j=1}^n D_g^j u_{ij}, \quad \sigma_{tg}^i = \sum_{j=1}^n \Sigma_{tg}^j u_{ij}, \quad \dots \text{etc}$$

Soit  $(x, y) \in \Omega \subset \mathbb{R}^2$ . Considérons  $\psi_i(x, y)$  fonction caractéristique de l'emplacement  $i$  :

$$\psi_i(x, y) = \begin{cases} 0 & \text{si } (x, y) \notin i \\ 1 & \text{si non} \end{cases}$$

Les coefficients des équations de la diffusion s'écrivent alors en fonction de la configuration  $u$  :

$$D_g(u, x, y) = D_g(u) = \sum_{i=1}^n \Psi_i(x, y) d_g^i = \sum_{i=1}^n \Psi_i(x, y) \sum_{j=1}^n D_g^j u_{ij} ;$$

$$\Sigma_{tg}(u, x, y) = \Sigma_{tg}(u) = \sum_{i=1}^n \Psi_i(x, y) \Sigma_{tg}^i = \sum_{i=1}^n \Psi_i(x, y) \sum_{j=1}^n \Sigma_{tg}^j u_{ij} ;$$

... etc

A l'aide de ces notations, nous pouvons faire intervenir directement la configuration  $u$  comme paramètre de l'équation E.

Nous obtenons le système d'équations aux dérivées partielles E(u) :

$$E(u) \left\{ \begin{array}{l} - \operatorname{div} (D_g(u) \operatorname{grad} \phi_g(u) + \Sigma_{tg}(u) \phi_g(u) - \sum_{g'=g+1}^{\Sigma_R} \Sigma_R(g \rightarrow g')(u) \phi_{g'}(u) = \\ = \frac{X_g(u)}{K_{\text{eff}}(u)} \sum_{g'=1}^m (\nu \Sigma f)_{g'}(u) \phi_{g'}(u) \quad \text{dans } \Omega \\ \phi_g(u)/\Gamma = 0 \quad \quad \quad g = 1, m. \end{array} \right.$$

De la même manière, la puissance spécifique s'écrira :

$$p(u) = (p_R(u))_{k=1;n}$$

$$P_k(u) = \frac{\operatorname{mes}(\Omega)}{\operatorname{mes}(\omega_k)} \frac{\int_R \sum_{g=1}^m (K_g \Sigma_{fg})(u) \phi_g(u) d\omega_R}{\int_{\Omega} \sum_{g=1}^m (K \Sigma_f)_g(u) \phi_g(u) d\Omega} \quad k = 1, n .$$

La recherche de la meilleure configuration s'écrit alors suivant le

critère 1  $C1 : \operatorname{Min} (\max_{i=1,n} p_i(u))$   
 $u \in \mathcal{U}$

critère 2  $\left\{ \begin{array}{l} C2 : \operatorname{Max} K_{\text{eff}}(u) \\ p_j(u) \leq p^{\max} \quad j = 1, n \\ u \in \mathcal{U} \end{array} \right.$

## 2.2. Remarque

Le problème C2 n'a pas de solution si  $p^{\max}$  est trop petit. C'est-à-dire quand  $p^{\max} < \min_{u \in \mathcal{U}} \max_{i=1,n} p_i(u)$ .

Lorsque C1 et C2 admettent une configuration optimale, celle-ci n'est pas toujours unique.

C1 a toujours au moins une solution.

## 3 - METHODE DE RESOLUTION

Les difficultés que nous avons soulevées dans l'introduction nous ont amené à choisir la méthode suivante pour résoudre C1 et C2. Voir figure I.2.

### 3.1. Linéarisation de C2

$$C2 \quad \left\{ \begin{array}{l} \max K_{\text{eff}}(u) \\ p_j(u) \leq p^{\max} \quad j = 1, n \\ u \in \mathcal{U} \end{array} \right.$$

Soit  $u^0$  une configuration donnée,  $u$  un élément de  $\tilde{\mathcal{U}}$ . Ecrivons les développements limités au premier ordre de  $K_{\text{eff}}$  et de la puissance :

$$K_{\text{eff}}(u) = K_{\text{eff}}(u^0) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial K_{\text{eff}}(u^0)}{\partial u_{ij}} (u_{ij} - u^0_{ij})$$

$$P_1(u) = P_1(u^0) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial P_1}{\partial u_{ij}} (u^0) (u_{ij} - u^0_{ij}) \quad l = 1, n.$$

Remplaçons C2 par le problème linéarisé suivant

$$\left\{ \begin{array}{l} \max K_{\text{eff}}(u^0) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial K_{\text{eff}}(u^0)}{\partial u_{ij}} (u_{ij} - u^0_{ij}) \\ P_1(u^0) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial P_1}{\partial u_{ij}} (u^0) (u_{ij} - u^0_{ij}) \leq p^{\max} \quad l = 1, n. \\ u \in \tilde{\mathcal{U}} \end{array} \right.$$

Pour obtenir une configuration  $u^1$ , nous résolvons ce problème sur un ensemble  $u_1 \in \mathcal{U}$  pour lequel cette linéarisation reste valable.

Nous obtenons ainsi un programme linéaire à variables booléennes.

Soit le problème LC2

$$\text{LC2} \left\{ \begin{array}{l} \max K_{\text{eff}}(u^\circ) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial K_{\text{eff}}(u^\circ)}{\partial u_{ij}} (u_{ij} - u^\circ_{ij}) \\ P_1(u^\circ) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial P_1(u^\circ)}{\partial u_{ij}} (u_{ij} - u^\circ_{ij}) \leq P^{\max} \quad l = 1, n. \\ u \in \mathcal{U}_1 \subset \mathcal{U} \end{array} \right.$$

### 3.2. Linéarisation de C1

$$\text{C1} \left\{ \begin{array}{l} \min \max p_j(u) \\ u \in \mathcal{U} \quad j = 1, n \end{array} \right.$$

C1 est équivalent au problème

$$\left\{ \begin{array}{l} \min y \\ P_1(u) \leq y \quad l = 1, n \\ u \in \mathcal{U} \\ y \in \mathbb{R}_+ \end{array} \right.$$

De même que pour C2, nous remplaçons  $C_1$  par son expression linéarisée sur  $\tilde{\mathcal{U}}$ , à partir de la configuration  $u^\circ$

$$\left\{ \begin{array}{l} \min y \\ P_1(u^\circ) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial P_1(u^\circ)}{\partial u_{ij}} (u_{ij} - u^\circ_{ij}) \leq y \quad l = 1, n \\ u \in \tilde{\mathcal{U}} \quad y \in \mathbb{R} \end{array} \right.$$

En résolvant ce problème sur  $\mathcal{U}_1 \subset \mathcal{U}$ , sous ensemble sur lequel la linéarisation reste valable, nous obtenons le programme linéaire à variables mixtes suivant :

$$\text{LC1} \left\{ \begin{array}{l} \min y \\ P_1(u^0) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial P_1(u^0)}{\partial u_{ij}} (u_{ij} - u_{ij}^0) \leq y \quad l = 1, n \\ u \in \mathcal{U}_1 \subset \mathcal{U} \\ y \in \mathbb{R}_+^* \end{array} \right.$$

### 3.3. Résolution de C1 (resp C2)

Nous utilisons la méthode itérative suivante : A partir de  $u^0$ , nous construisons une suite de configurations  $u^k$ . Chaque configuration  $u^k$  est la solution optimale de LC1 (resp LC2), les termes des développements étant calculés pour  $u^{k-1}$ . La configuration  $\bar{u}$  de la suite  $u^k$  sera dite solution optimale de C1 (resp C2) si le problème linéarisé LC1 (resp LC2) en ce point n'a pas de solution améliorant notablement la fonction objectif.

### 3.4. Remarques

Nous n'avons à notre disposition aucun théorème nous permettant d'affirmer que  $\bar{u}$  est la configuration optimum global de C1 (resp C2).

Comme nous le verrons dans les exemples de résolution, au chapitre V, nous pouvons obtenir des résultats satisfaisants même si tous les termes de la suite  $u^k$  ne sont pas les optimum globaux des problèmes du type LC1 (resp LC2).

### 3.5. Autre méthode

Si à chaque étape de la méthode itérative précédente nous résolvons un programme linéaire continu en nous affranchissant de la condition booléenne sur  $u$ , alors le processus itératif que nous obtenons n'est autre que la MAP  $\underline{g}$ .

Plaçons nous à une étape  $k$  de ce processus. Soit  $\hat{u}^k$  la solution continue optimale du programme linéaire.  $\hat{u}^k$  est un élément de  $\hat{\mathcal{U}}$  enveloppe convexe de  $\mathcal{U}$ . Par rapport à la configuration de départ  $u^0$ , la configuration idéale obtenue à l'étape  $k$  sera donnée par l'élément  $u^k$  de  $\mathcal{U}$  telle que

$$u^k = \hat{u}^k \text{ ou } \hat{u}^{k-1} \text{ ou } \dots \text{ ou } u^0$$

Nous estimons alors que le processus a convergé lorsque l'on aura  $\|u^k - u^{k+1}\| \ll \epsilon$  avec pour  $u \in \hat{\mathcal{U}}$   $\|u\|^2 = \sum_{i,j} u_{ij}^2$ .

Soit  $\tilde{u}$  la solution optimale donnée par la MAP. Nous l'approchons de manière discrète selon les problèmes que nous avons à résoudre de la manière suivante :

### 3.5.1. Problème C1

Nous cherchons la configuration  $u$  dans un voisinage de  $\tilde{u}$ , qui soit solution optimale du problème LC1 suivant :

$$\left\{ \begin{array}{l} \min y \\ P_i(\tilde{u}) + \sum_{1,k}^n \frac{\partial P_i(u)}{\partial u_{1k}} (\tilde{u}_{1k} - \tilde{u}_{1k}) \leq y \quad i = 1, n \\ u \in V(\tilde{u}) \end{array} \right.$$

### 3.5.2. Problème C2

De même, nous cherchons la configuration  $\hat{u}$  dans un voisinage de  $\tilde{u}$  solution optimale du problème LC2

$$\left\{ \begin{array}{l} \max K_{\text{eff}}(\hat{u}) + \sum_{1,k}^n \frac{\partial K_{\text{eff}}(\hat{u})}{\partial u_{1k}} (\hat{u}_{1k} - \tilde{u}_{1k}) \\ P_i(\hat{u}) + \sum_{1,k}^n \frac{\partial P_i(\hat{u})}{\partial u_{1k}} (\hat{u}_{1k} - \tilde{u}_{1k}) \leq p^{\max} \quad i = 1, n \\ u \in V(\hat{u}) \end{array} \right.$$

### 3.5.3. Remarques

Notre souci est de trouver une configuration discrète  $u$ , située dans  $V(\tilde{u})$ , qui améliore le mieux les fonctions objectif linéarisées au point  $\tilde{u}$ .

La détermination de  $V(\tilde{u})$  est faite dans l'Annexe C.

### 3.6. Conclusion

La mise en oeuvre des deux méthodes décrites ci-dessus nécessite :

le calcul de  $K_{\text{eff}}(u)$  et  $p(u)$  pour  $u \in \tilde{u}$

le calcul des développements limités au premier ordre de  $K_{\text{eff}}(u)$  et  $p(u)$

la détermination du voisinage sur lequel la linéarisation est valable

la mise en oeuvre d'algorithmes de programmation linéaire continue et discrète.

Dans les chapitres suivants nous exposons de quelle façon nous avons résolu ces difficultés.

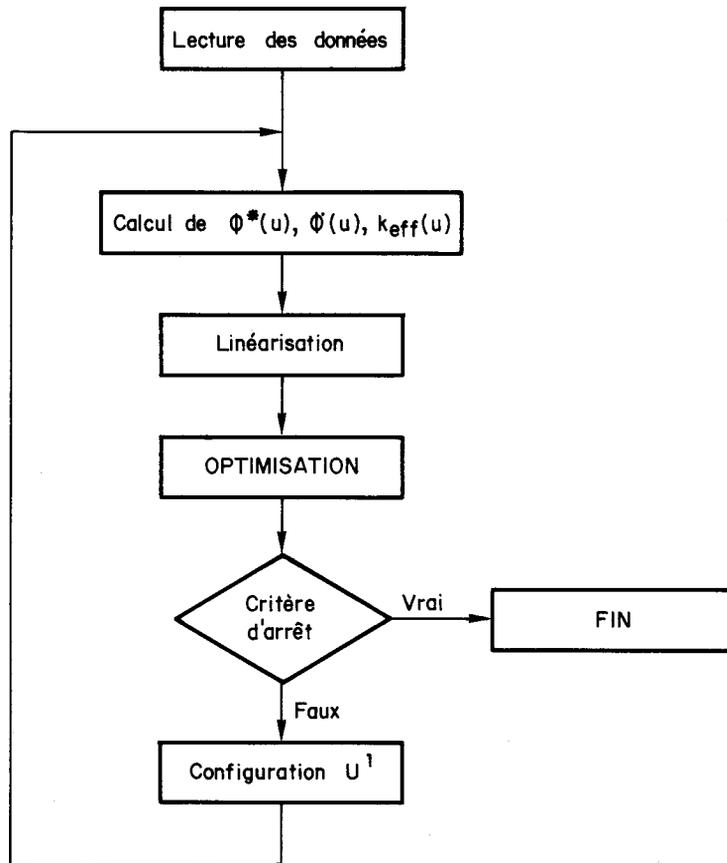


Figure I.2 . Schéma de principe de la méthode.



## CHAPITRE II - LINÉARISATION DE $K_{\text{EFF}}(U)$ ET $P(U)$

Soit  $\tilde{U}$  l'enveloppe convexe de  $U$ .

$$\tilde{U} = \left\{ u \in \mathbb{R}^{n^2}, \sum_{i=1}^n u_{ij} = 1 \quad j = 1, n; \sum_{j=1}^n u_{ij} = 1 \quad i = 1, n; 0 \leq u_{ij} \leq 1 \right\}$$

Considérons le système d'équations aux dérivées partielles  $E(u)$  pour  $u \in \tilde{U}$ . Les coefficients  $Dg(u), \Sigma_{ag}(u), \dots$  sont dans  $L_{\infty}(\Omega)$ . Nous savons d'après PLANCHARD [17] qu'il admet une solution unique  $(\phi(u), \lambda(u))$ .  $\lambda(u) = \frac{1}{K_{\text{eff}}(u)}$  étant la plus petite valeur propre simple du système et  $\phi(u)$  le vecteur propre positif associé.

### 1. DEVELOPPEMENT AU PREMIER ORDRE DE $K_{\text{eff}}(u)$ ET $p(u)$

Ecrivons le système  $E(u)$  sous la forme

$$u \in \tilde{U} \quad E(u) \quad \left\{ \begin{array}{l} A(u) \phi(u) = \lambda(u) B(u) \\ \phi(u)/\Gamma = 0 \end{array} \right.$$

où  $A(u), B(u)$  sont des opérateurs linéaires en  $u$  de  $(H^2(\Omega) \cap H_0^1(\Omega))^m$  dans  $(H^{-1}(\Omega))^m$ .

Pour  $m = 2$ , ils s'écrivent simplement

$$A(u) = \begin{pmatrix} -\nabla D_1(u) \nabla + \Sigma_{t1}(u) & 0 \\ -\Sigma_R(1 \rightarrow 2)(u) & -\nabla D_2(u) \nabla + \Sigma_{t2}(u) \end{pmatrix}$$

$$B(u) = \begin{pmatrix} \chi_1(u) (V \Sigma_f)_1(u) & \chi_1(u) (V \Sigma_f)_2(u) \\ \chi_2(u) (V \Sigma_f)_1(u) & \chi_2(u) (V \Sigma_f)_2(u) \end{pmatrix}$$

Soient  $a(u, \varphi, \psi)$  et  $b(u, \varphi, \psi)$  les formes bilinéaires associées à  $A(u)$  et  $B(u)$

$$a(u, \varphi, \psi) = \sum_{g=1}^m \int_{\Omega} Dg(u) \nabla \varphi_g \nabla \psi_g + \Sigma_{ag}(u) \varphi_g \psi_g - \sum_{g'=g+1}^m \Sigma_R(g \rightarrow g')(u) \varphi_{g'} \psi_g \, d\Omega$$

$$b(u, \Psi, \psi) = \sum_{g=1}^m \int_{\Omega} \chi_g(u) \sum_{g'=1}^m (V_{\Sigma_{f'}}, (u)) \Psi_{g'} \psi_g d\Omega$$

### 1.1. Remarque

Les formes bilinéaires  $a(u, \Psi, \psi)$ ,  $b(u, \Psi, \psi)$  sont dérivables sur  $\tilde{u}$  par rapport à  $u$ . En effet les coefficients  $D_g(u)$ ,  $\Sigma_{ag}(u)$  ... etc sont linéaires en  $u$ , donc continuellement différentiables.

On a par exemple :

$$\frac{\partial}{\partial u_{ij}} D_g(u) = \frac{\partial}{\partial u_{ij}} \left( \sum_{l=1}^n \Psi_l(x, g) \sum_{k=1}^n D_g^k u_{lk} \right)$$

Comme  $i$  et  $j$  sont fixés on a

$$\frac{\partial}{\partial u_{ij}} D_g(u) = D_g^j$$

Il est donc facile de déterminer les opérateurs  $\frac{\partial A}{\partial u_{ij}}$  et  $\frac{\partial B}{\partial u_{ij}}$ .

### 1.2. Développement au premier ordre de $K_{eff}(u)$

Les opérateurs  $A(u)$  et  $B(u)$  n'étant pas auto-adjoints,  $B(u)$  étant de plus singulier, nous ne pouvons démontrer comme dans MIGNOT [15] que  $\lambda(u)$  et  $\phi(u)$  sont différentiables en  $u$ . Nous admettons donc ce résultat. Par un calcul formel, nous allons déterminer la variation de  $\lambda(u)$  en fonction de  $u$ .

Soit  $\phi^*(u)$ , le flux adjoint de  $\phi(u)$ .  $A^*(u)$ ,  $B^*(u)$  les opérateurs adjoints de  $A(u)$  et  $B(u)$ .

Nous avons :

$$\lambda(u) = \frac{\langle \phi^*(u), A(u) \phi(u) \rangle}{\langle \phi^*(u), B(u) \phi(u) \rangle}$$

Calculons la variation  $\delta\lambda(u)$  de  $\lambda(u)$ . De manière formelle sachant que  $A(u) \phi(u) = \lambda(u) B(u) \phi(u)$  nous avons :

$$\left[ A(u) - \lambda(u) B(u) \right] \delta\phi(u) + \left[ \delta A(u) - \lambda(u) \delta B(u) \right] \phi(u) = \delta\lambda(u) B(u) \phi(u)$$

Soit en utilisant la dualité  $\langle V, V' \rangle$  avec  $V = \left[ H^2(\Omega) \cap H'_0(\Omega) \right]^2$  nous avons :

$$\langle \phi^*(u), [A(u) - \lambda(u) B(u)] \delta\phi(u) \rangle + \langle \phi^*(u), (\delta A(u) - \lambda(u) \delta B(u)) \phi(u) \rangle =$$

$$\delta\lambda(u) \langle \phi^*(u), B(u) \phi(u) \rangle$$

comme  $A^*(u) \phi^*(u) = \lambda(u) B^*(u) \phi^*(u)$  nous en déduisons

$$\delta\lambda(u) = \frac{\langle \phi^*(u), (\delta A(u) - \lambda(u) \delta B(u)) \phi(u) \rangle}{\langle \phi^*(u), B(u) \phi(u) \rangle}$$

comme  $\lambda(u) = \frac{1}{K_{\text{eff}}(u)}$  nous en déduisons :

$$K_{\text{eff}}(u) = K_{\text{eff}}(u^0) - \left[ K_{\text{eff}}(u^0) \right]^2 \times \sum_{i=1}^n \sum_{j=1}^n \frac{\langle \phi^*(u), \frac{\partial A}{\partial u_{ij}} (A(u) - \lambda(u) B(u)) \phi(u) \rangle}{\langle \phi^*(u), B(u) \phi(u) \rangle}$$

### 1.3. Développement au premier ordre de $p(u)$

$\omega_j$  désignant un emplacement du coeur, la puissance spécifique en cet emplacement s'écrit :

$$P_j(u) = \frac{\langle \Sigma_1^j(u), \phi(u) \rangle_{L^2(\Omega)}}{\langle \Sigma_2(u), \phi(u) \rangle_{L^2(\Omega)}}$$

$$\text{avec } \langle \Sigma_1^j(u), \phi(u) \rangle_{L^2(\Omega)} = \frac{1}{\text{mes}(\omega_j)} \int_{\omega_j} \sum_{g=1}^m (K\Sigma_f)_g(u) \phi_g(u) d\omega$$

$$\langle \Sigma_2(u), \phi(u) \rangle_{L^2(\Omega)} = \frac{1}{\text{mes}(\Omega)} \int_{\Omega} \sum_{g=1}^m (K\Sigma_f)_g(u) \phi_g(u) d\Omega$$

Par un calcul formel, nous obtenons la variation  $\delta P_j(u)$  de  $P_j$

$$\delta P_j(u) = \frac{\langle \delta \Sigma_1^j(u) - P_j \delta \Sigma_2(u), \phi(u) \rangle_{L^2(\Omega)} + \langle \Sigma_1^j(u) - P_j \Sigma_2(u), \delta \phi(u) \rangle_{L^2(\Omega)}}{\langle \Sigma_2(u), \phi(u) \rangle_{L^2(\Omega)}}$$

à éliminer  $\delta \phi(u)$  de l'expression ci-dessus. Considérons alors  $S_j^*$ , solution du système d'équations aux dérivées partielles :

$$\left\{ \begin{array}{l} (A^*(u) - \lambda(u) B^*(u)) S_j^*(u) = \Sigma_1^j(u) - P_j \Sigma_2(u) \\ \langle S_j^*(u), B(u) \phi(u) \rangle = 0 \end{array} \right.$$

$S_j^*$  est appelé importance source de l'élément situé à l'emplacement  $j$ .

Nous avons :

$$(A^*(u) - \lambda(u) B^*(u)) S_j^*(u), \delta\phi(u) = \langle S_j^*(u), (A(u) - \lambda(u) B(u)) \phi(u) \rangle$$

$$\langle (A(u) - \lambda(u) B(u)) \delta\phi(u) = (\delta A(u) - \lambda(u) \delta B(u)) \phi(u) + \delta\lambda(u) B(u) \phi(u)$$

$$\text{Comme } \langle S_j^*(u), B(u) \phi(u) \rangle = 0$$

L'expression  $\langle \Sigma_1^j(u) - P_j \Sigma_2(u), \delta\phi(u) \rangle$  vaut :

$$\langle \Sigma_1^j(u) - P_j \Sigma_2(u), \delta\phi(u) \rangle = \langle S_j^*(u), (\delta A(u) - \lambda(u) \delta B(u)) \phi(u) \rangle$$

La variation  $\delta P_j(u)$  s'écrit alors :

$$\delta P_j(u) = \frac{\langle \Sigma_1^j(u) - P_j \delta \Sigma_2(u) \rangle + \langle S_j^*(u), (\delta A(u) - \lambda(u) \delta B(u)) \phi(u) \rangle}{\langle \Sigma_2(u) \phi(u) \rangle}$$

Nous en déduisons le développement au premier ordre de  $P_j(u)$

$$P_j(u) = P_j(u^0) + \sum_{i=1}^n \sum_{k=1}^n \frac{\partial P_j(u^0)}{\partial u_{ik}} (u_{ik} - u_{ik}^0)$$

$$\text{avec } \frac{\partial P_j}{\partial u_{ik}} = \left\{ \langle \sigma_{1ik}^j(u^0), \phi(u^0) \rangle - P_j(u^0) \langle \sigma_{2ik}(u^0), \phi(u^0) \rangle + \frac{\partial A}{\partial u_{ik}}(u^0) S_j^*(u^0), \phi(u^0) \right. \\ \left. - \lambda(u^0) \langle \frac{\partial B}{\partial u_{ik}}(u^0) S_j^*(u^0), \phi(u^0) \rangle / \langle \Sigma_2(u^0), \phi(u^0) \rangle \right.$$

$$\text{où } \sigma_{1ik}^j = \frac{\partial}{\partial u_{ik}} \Sigma_1^j(u) \quad \text{et} \quad \sigma_{2ik} = \frac{\partial}{\partial u_{ik}} \Sigma_2(u)$$

#### 1.4. Calcul de $S_j^*(u^0)$ , $\phi^*(u^0)$ , $\phi(u^0)$ , $\lambda(u^0)$

Solutions d'un système d'équations aux dérivées partielles, ces termes sont calculés à l'aide de la méthode des éléments finis, mise au point par LAUTARD [14].

Pour calculer ces termes, il nous a fallu simplement faire quelques modifications - notamment le calcul des coefficients  $D_g(u)$ ,  $\Sigma_{ag}(u)$  - dans les modules du système Neptune [13] qui résolvent ces équations.

#### 1.5. Calcul des $\frac{\partial}{\partial u_{ik}} K_{eff}(u^0)$ et $\frac{\partial}{\partial u_{ik}} P_j(u^0)$

Ces calculs ne sont pas plus compliqués que les précédents. En réalité, pour chaque  $i$  fixé, les expressions sont identiques à celles obtenues par DUMAS [6]. Là encore, quelques modifications dans les programmes ont suffi pour obtenir les expressions des dérivées.

## 2. RESULTATS NUMERIQUES

Dans toute la suite, les valeurs de  $K_{eff}(u)$  et  $p(u)$  obtenu après résolution de  $E(u)$  par la méthode des éléments finis seront considérées comme exactes.

### 2.1. Précision des développements linéaires

Considérons deux configurations  $u^0$  et  $u^1$  de coefficient de multiplication effectif respectif  $K^0 = K_{eff}(u^0)$  et  $K^1 = K_{eff}(u^1)$  et de distribution de puissance  $p^0 = p(u^0)$  et  $p^1 = p(u^1)$ .

Désignons par  $\tilde{K}$  la valeur de  $K_{eff}(u)$  obtenue à partir des expressions linéarisées.

$$\tilde{K} = K^0 + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial K_{eff}(u^0)}{\partial u_{ij}} (u_{ij} - u_{ij}^0)$$

De même si  $\tilde{p}$  désigne la distribution de puissance linéarisée :

$$\tilde{p}_k = p_k^0 + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial p_k(u^0)}{\partial u_{ij}} (u_{ij} - u_{ij}^0) \quad k = 1, n$$

Le tableau II.1 donne la variation  $|\tilde{K} - K^1|$  quand on passe de  $u^0$  à  $u^1$  en échangeant deux éléments de burn-up voisin.

Le tableau II.2 donne la variation  $|\tilde{p}_j - p_j^1|$  dans le même cas.

Lorsqu'on procède à l'échange de deux éléments de burn-up très différents, la précision diminue. Le tableau II.3 donne l'erreur  $|\tilde{K} - k^1|$  commise sur le  $K_{eff}$ , le tableau II.4 celle commise sur la puissance.

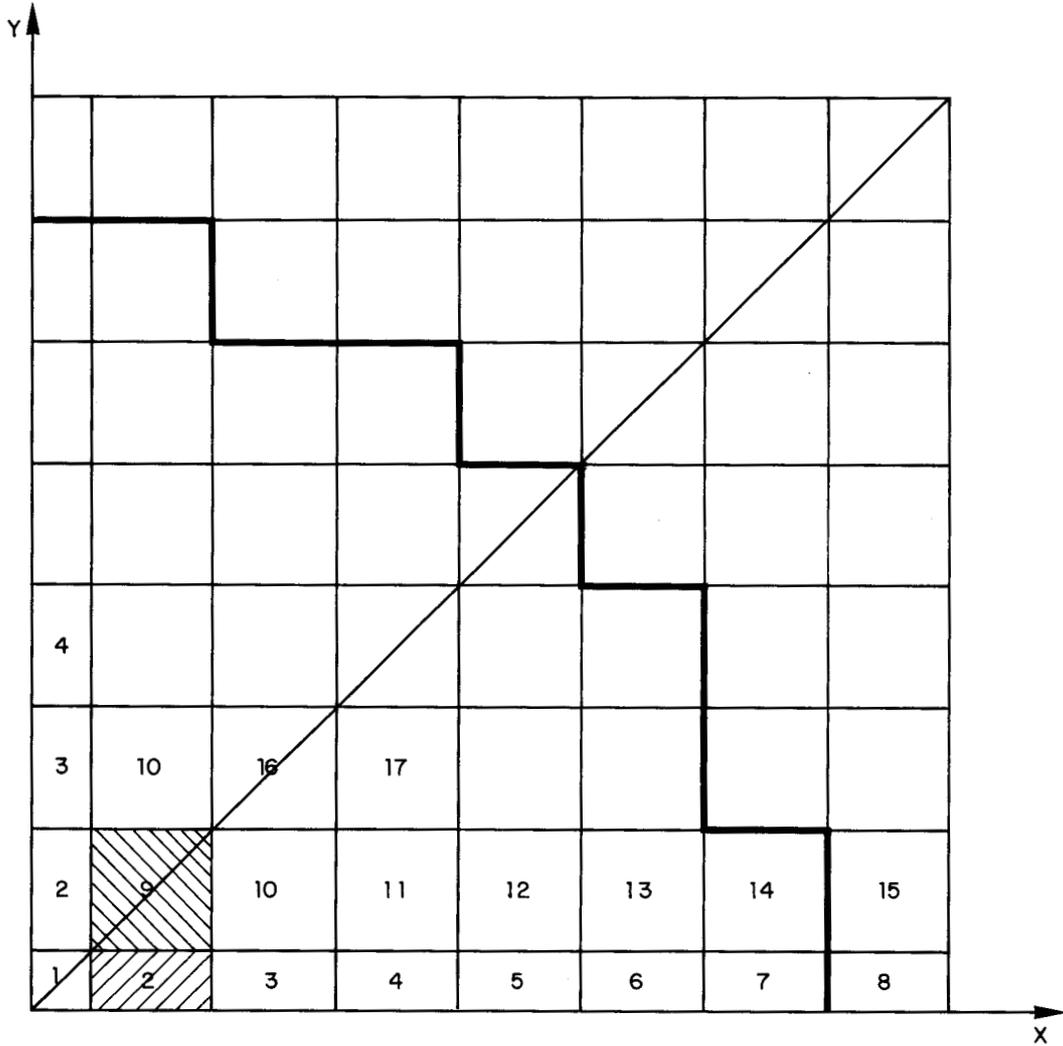


Figure II.1 \_ Représentation du cœur à symétrie par rapport à la première diagonale.

Seuls les éléments 2 et 9 sont échangés.

Les puissances des tableaux étant calculées sur les emplacements 1,2,9.

Le tableau II.5 compare les précisions de la méthode DUMAS [5] et la nôtre.

Le tableau II.6 donne l'évolution de l'erreur sur le  $K_{\text{eff}}$  et sur  $p$  en fonction du nombre d'assemblages échangés.

$K^0$	$\frac{\partial K}{\partial u_{22}}$	$\frac{\partial K}{\partial u_{29}}$	$\frac{\partial K}{\partial u_{92}}$	$\frac{\partial K}{\partial u_{99}}$	$\beta$	$K^1$	$ K^1 - K^0 $
1.10350	$0.16144 \cdot 10^{-2}$	$0.05616 \cdot 10^{-2}$	$0.14386 \cdot 10^{-2}$	$0.04674 \cdot 10^{-2}$	1.10342	1.10344	$2 \cdot 10^{-5}$

Tableau II.1 : Echange de deux assemblages, variations de  $K_{\text{eff}}(u)$

$P_j^0$	$\frac{\partial P_j}{\partial u_{22}}$	$\frac{\partial P_j}{\partial u_{29}}$	$\frac{\partial P_j}{\partial u_{92}}$	$\frac{\partial P_j}{\partial u_{99}}$	$\beta_j$	$P_j^1$	$ P_j^1 - \beta_j $
1.7175	0.07673	0.08816	- 0.06431	- 0.11645	1.6841	1.6855	$14 \cdot 10^{-4}$
1.8805	1.89654	1.80067	- 0.07167	- 0.12855	1.84153	1.8427	$12 \cdot 10^{-4}$
1.793	0.00971	- 0.06906	1.76950	1.69913	1.8041	1.8051	$10 \cdot 10^{-4}$

Tableau II.2 : Echange de deux assemblages, variations de  $p(u)$

Les assemblages échangés ont des burn up voisins. Ils diffèrent de moins de 1000 MWj/t.

La précision du calcul du  $K_{\text{eff}}$  est de l'ordre de la précision du calcul en éléments finis.

L'erreur relative sur la distribution de puissance est de l'ordre de  $10^{-3}$ .

$K^0$	$\frac{\partial K}{\partial u_{22}}$	$\frac{\partial K}{\partial u_{29}}$	$\frac{\partial K}{\partial u_{92}}$	$\frac{\partial K}{\partial u_{99}}$	$\beta$	$K^1$	$ K^1 - \beta $
1.09883	0.00143	- 0.00377	0.001332	- 0.00332	1.09822	1.09873	$5 \cdot 10^{-4}$

Tableau II.3 : Echange de deux éléments, variations de  $K_{\text{eff}}(u)$



$P_j^0$	$\frac{\partial P_j}{\partial u_{22}}$	$\frac{\partial P_j}{\partial u_{29}}$	$\frac{\partial P_j}{\partial u_{g2}}$	$\frac{\partial P_j}{\partial u_{g9}}$	$P_j^2$	$P_j^1$	$ P_j^1 - P_j^2 $
1.4540	0.12350	- 0.35529	- 0.00377	0.28379	1.355	1.299	$44 \cdot 10^{-3}$
1.5961	1.6608	1.12839	- 0.00334	- 0.30351	1.363	1.412	$49 \cdot 10^{-3}$
1.4473	0.03869	0.27135	1.5377	1.18137	1.493	1.525	$32 \cdot 10^{-3}$

Tableau II.4 : Echange de deux éléments, variations de  $K_{\text{eff}}(u)$ .

Les burn-up des assemblages échangés diffèrent de 6500 MWj/t. La précision du calcul du  $K_{\text{eff}}$  est de 50 pcm. (1 pcm =  $10^{-5}$ ). L'erreur sur la distribution de puissance est de l'ordre de  $5 \cdot 10^{-2}$ .

$\tilde{K}$	$ K_1 - \tilde{K} $	$\tilde{P}_1$	$ P_1 - \tilde{P}_1 $	$\tilde{P}_2$	$ P_2 - \tilde{P}_2 $	$\tilde{P}_g$	$ P_g - \tilde{P}_g $
1.09822	$5 \cdot 10^{-4}$	1.256	$43 \cdot 10^{-3}$	1.374	$38 \cdot 10^{-3}$	1.512	$13 \cdot 10^{-3}$
1.09822	$5 \cdot 10^{-4}$	1.255	$44 \cdot 10^{-3}$	1.363	$49 \cdot 10^{-3}$	1.493	$32 \cdot 10^{-3}$

Tableau II.5 : Comparaison de notre linéarisation avec celle de DUMAS [5].

La première ligne donnant les résultats de DUMAS [5].



L'erreur sur le  $K_{\text{eff}}$  est la même (50 pcm). La distribution de puissance linéarisée de DUMAS [6] étant légèrement plus précise, l'ordre de grandeur de l'erreur étant la même,  $5.10^{-2}$ .

n	$K^1$	$\tilde{K}$	$ \tilde{K}-K^1 $	$f^1$	$\tilde{f}$	$ \tilde{f}-f^1 $
2	1.10344	1.10342	$2.10^{-5}$	1.525	1.493	$32.10^{-3}$
4	1.09782	1.09715	$67.10^{-5}$	1.535	1.498	$37.10^{-3}$
9	1.09761	1.09748	$13.10^{-5}$	1.497	1.489	$8.10^{-3}$
10	1.11066	1.11062	$4.10^{-5}$	1.305	1.300	$5.10^{-3}$

Tableau II.6 : Variations de  $K_{\text{eff}}(u)$  et  $p(u)$  en fonction du nombre d'assemblages échangés.

Dans ce cas, aucune limite n'a été fixée au paramètre définissant l'état de l'assemblage (burn-up ou  $K_{\infty}$ ).

## 2.2. Temps de calcul

La durée de la linéarisation dépend essentiellement du nombre de puissances spécifiques calculées, ainsi que du nombre d'assemblages à répositionner.

Pour  $n = 20$ , et si l'on calcule 21 puissances spécifiques, la dimension de la matrice des dérivées est de  $21 \times 400$ , soit 8400 termes. Le calcul dure 5 s 62 sur IBM 30-33.

Pour  $n = 32$  et si l'on calcule 47 puissances spécifiques, la dimension de la matrice des dérivées est de  $47 \times 1024$ . Soit 48 128 termes.

Le calcul dure 34 s 41 sur IBM 90-33.

Ces temps sont relativement peu importants eu égard au nombre de termes calculés.

### CHAPITRE III - RÉSOLUTION D'UN PROGRAMME LINÉAIRE À VARIABLES BOOLÉENNES PAR UNE MÉTHODE D'ÉNUMÉRATION IMPLICITE

Tout programme linéaire en nombre entiers, à variables booléennes peut se ramener à la forme suivante :

$$(P) \quad \left\{ \begin{array}{l} \min \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j + b_i \geq 0 \quad i = 1, m. \quad (1) \\ x_j = 0 \text{ ou } 1 \quad j = 1, n. \quad (2) \end{array} \right.$$

où  $b = (b_i) \in \mathbb{R}^m$ ,  $A = (a_{ij}) \in \mathbb{R}^{m \times n}$   $c = (c_j) \in \mathbb{R}_+^n$ .

Tout vecteur  $x$  de  $\mathbb{R}^n$  à composantes binaires est appelé solution du problème (P).

Toute solution de (P) vérifiant les contraintes (1) sera appelée solution réalisable de (P).

Toute solution réalisable qui minimise l'objectif sera appelée solution optimale de (P).

Au lieu de procéder à une énumération complète des  $2^n$  solutions de (P), l'énumération implicite est une stratégie permettant au cours même de l'énumération de générer des informations conduisant à n'étudier explicitement qu'une "petite" partie des  $2^n$  solutions de P.

Parmi les diverses méthodes de résolutions de programmes mathématiques à variables bivalentes, nous avons choisi la méthode d'énumération implicite proposée par BALAS [1].

Parmi toutes les versions de cet algorithme - voir TAHA [2] nous avons retenu celle de GEOFFRIDON [7] car elle est de formulation simple et l'introduction d'une contrainte additionnelle le rend plus efficace.

#### 3.1. Procédure d'énumération implicite

On appellera solution partielle de P, tout sous ensemble  $S \mid |S| = s \leq n$ , fixant les valeurs de  $s$  variables  $x_j$ .

On appellera variable libre, toute variable  $x_j$  dont la valeur n'est pas fixée par S.

Nous conviendrons que le symbole  $j$  indique que  $x_j = 1$  et le symbole  $-j$  que  $x_j = 0$ . Ainsi par exemple pour  $n = 4$  si  $S = \{3, -2, 1\}$  alors nous aurons :

$$x_3 = 1, \quad x_2 = 0 \quad \text{et} \quad x_1 = 1$$

la variable  $x_4$  étant libre.

On appellera complétion d'une solution partielle  $S$ , toute solution définie à l'aide de  $S$  et d'une spécification des variables libres. Notons qu'un sous ensemble  $S$  a  $2^{n-s}$  complétions.

### 3.1.1. Principe de l'énumération implicite

L'énumération implicite consiste à générer une suite de solutions partielles  $S^P$ , en considérant à chaque fois toutes les complétions de chacune d'elles.

Toutes les complétions d'une solution partielle  $S^P$  auront été examinées si nous pouvons aboutir à l'un des résultats suivants :

- nous avons trouvé la meilleure complétion de  $S^P$ , c'est-à-dire celle qui minimise  $cx$  parmi toutes les complétions de  $S^P$  qui sont réalisables.  
Soit  $\bar{z}$  la meilleure valeur de  $cx$  connue jusqu'alors. Si la meilleure complétion de  $S^P$  améliore  $\bar{z}$  alors, la valeur de  $\bar{z}$  est mise à jour.
- nous n'avons trouvé aucune complétion réalisable de  $S^P$  qui améliore la valeur de  $\bar{z}$ .

Au paragraphe 3.2., nous rappellerons l'ensemble de tests proposés par BALAS [1] permettant d'examiner une solution partielle  $S$ .

### 3.1.2. Procédure

La procédure ci-dessous permet de générer une suite non redondante de solutions partielles  $S^P$ .

Dans (P) nous supposons  $c_j \geq 0$ .

Partons de  $S^0 = \emptyset$ . Si toutes les complétions de  $S^0$  ont été examinées alors l'énumération est terminée. Dans ce cas, soit (P) n'a aucune solution réalisable, soit la meilleure valeur de l'objectif ne peut être améliorée.

Si toutes les complétions de  $S^0$  ne peuvent être examinées, alors considérons la solution partielle  $S^1 = S^0 \cup \{j_1\}$ , où  $j_1$  est une variable libre quelconque. Nous cherchons à examiner toutes les complétions de  $S^1$ . Supposons que pour  $k = k_1$ , toutes les complétions de  $S^{k_1} = S^0 \cup \{j_1\} \cup \{j_2\} \cup \dots \cup \{j_{k_1}\}$  aient été examinées.

Afin de pouvoir déterminer quand les  $2^n$  solutions de (P) auront été examinées, nous devons pouvoir nous "souvenir" que toutes les complétions de  $S^{k_1}$  ont été examinées. Nous devons donc conserver  $S^{k_1}$  en mémoire.

Afin de ne pas avoir à examiner des solutions partielles déjà étudiées, la solution partielle que nous allons examiner,  $S^{k_1+1}$  sera la solution  $S^{k_1}$  dont un élément aura été remplacé par son complémentaire.

Pratiquement, pour nous "souvenir" que toutes les complétions de  $S^{k_1}$  ont été examinées, nous "soulignerons" son dernier élément. Pour éviter la redondance nous multiplierons ce dernier élément par  $-1$ . Ainsi donc

$$S^{k_1+1} = \{j_1, j_2, \dots, \underline{j_{-k_1}}\} .$$

Exemple :

$$k_1 = 3 \quad S^3 = \{3, 5, -2\}$$

$$k_1+1 = 4 \quad S^4 = \{3, 5, \underline{2}\}$$

Il est évident que si toutes les complétions de  $S^{k_1+1}$  sont examinées, alors toutes les complétions de la solution partielle  $S^{k_1+2} = S^{k_1} \setminus \{j_{k_1}\}$  ont été examinées.

Pour nous "souvenir" de ce fait, nous prendrons donc  $S^{k_1+2}$  comme étant  $S^{k_1}$  sans son dernier élément et dont l'avant dernier élément a été souligné puis multiplié par  $-1$ .

Exemple :

$$k_1 = 3 \quad S^3 = \{3, 5, -2\}$$

$$k_1+1 = 4 \quad S^4 = \{3, 5, \underline{2}\}$$

$$k_1+2 = 5 \quad S^5 = \{3, \underline{-5}\}$$

Supposons maintenant que toutes les complétions de  $S^{k_1+1}$  n'aient pas été étudiées. Alors nous sommes amenés à augmenter  $S^{k_1+1}$  en attribuant une valeur particulière à une variable libre. Nous continuons à augmenter les solutions partielles ainsi formées jusqu'à ce que pour  $k = k_2$ , nous obtenions une solution partielle  $S^{k_2}$  dont toutes les complétions peuvent être examinées. La suite de solutions partielles  $S^r$ ,  $r = k_1+1, \dots, k_2$ , est une suite non redondante car chacune d'elle contient un élément qui est le complémentaire d'un élément de  $S^{k_1}$ .

A cette étape de la procédure, supposons que toutes les complétions de  $S^{k_1}$  aient été examinées. Nous devons nous "souvenir" non seulement que toutes les complétions de  $S^{k_2}$  ont été examinées, mais aussi celles de  $S^{k_1}$ .

Afin de réduire les problèmes de stockage, nous prendrons  $S^{k_2+1}$  égal à  $S^{k_2}$  dont le dernier élément a été "souligné" et multiplié par  $-1$ .

Exemple :  $n = 5$

$$k_1 = 3 \quad S^3 = \{3, 5, -2\} \quad k_1+1 = 4 \quad S^4 = \{3, 5, \underline{2}\}$$

alors  $S^6 = S^{k_2} = S^{k_2+1} = \{3, 5, \underline{2}, -1\}$ . Si toutes les complétions de  $S^5$  sont examinées,

### 3.1.3. Schéma de la procédure

0 -  $S = \emptyset \quad \bar{z} = +\infty$  ( $\bar{z}$  meilleure valeur de l'objectif)

1 - S n'a aucune complétion réalisable ou aucune complétion améliorant la meilleure valeur de l'objectif. Aller à l'étape 3.

S a une meilleure complétion réalisable améliorant  $\bar{z}$ . Conserver cette complétion aller à l'étape 3.

2 - Augmenter S en attribuant une valeur binaire à une variable libre quelconque. Aller à l'étape 1.

3 - Localiser l'élément le plus à droite de S non souligné. S'il n'y en a aucun, l'énumération est terminée.

Si non remplacer cet élément par son complémentaire souligné. Exclure de S tous les éléments situés à droite de cet élément. Aller à l'étape 1.

### 3.2. Tests permettant d'examiner les complétions d'une solution partielle

Soit  $\bar{z}$  la meilleure valeur de l'objectif connue. Supposons  $c_j > 0$ ,  $x^S$  désignant une complétion de la solution partielle S, posons :

$$y^S = b + A x^S$$

#### 3.2.1. La solution partielle peut-elle être augmentée ?

Considérons l'ensemble  $T^S$  :

$$T^S = \{j \notin S / c^S x_j^S + c_j < \bar{z} \text{ et } \exists i \ y_i^S < 0 \text{ avec } a_{ij} > 0\}$$

$T^S$  est appelé l'ensemble des variables candidates.

En effet, l'introduction dans S d'une variable  $j \in T^S$  conduirait soit à une complétion dont la valeur n'améliore pas l'objectif, soit à une complétion irréalisable.

L'ensemble  $T^S$  ne contient donc que des variables susceptibles de nous fournir des complétions réalisables qui améliore l'objectif.

### 3.2.2. La solution partielle S n'a-t-elle aucune complétion réalisable ?

Une réponse à cette question permet de limiter l'examen des complétions de S. Aussi faut-il essayer d'y répondre précisément. Dans le cas général, GEOFFRION propose la solution suivante.

Si il existe une contrainte  $y_i^S$  non satisfaite, telle que  $y_i^S + \sum_{j \in TS} \max(0, a_{ij}) < 0$ , alors S n'a aucune complétion réalisable.

Nous présenterons au chapitre suivant une modification de ce test pour l'adapter à notre problème.

### 3.3. La contrainte additionnelle

GEOFFRION [8], introduit dans P, une contrainte spéciale dont le but est d'apporter plus d'informations sur les complétions de S que les tests précédents.

Cette contrainte est de la forme

$$\sum_{i=1}^m \mu_i (b_i + \sum_{j=1}^n a_{ij} x_j) + (\bar{z} - \sum_{j=1}^n c_j x_j) \geq 0$$

Elle est une combinaison linéaire à coefficients positifs, des contraintes initiales du problème.

#### 3.3.1. Remarque

Si la contrainte additionnelle n'a pas de solution alors, soit S n'a pas de complétion réalisable, soit S n'a pas de complétion améliorante.

Ceci permet de répondre au moins partiellement à la question 3.2.2., en examinant d'abord cette contrainte.

Les coefficients  $\mu$  sont les valeurs de la contrainte additionnelle la plus stricte au sens que nous allons préciser maintenant.

La contrainte  $\mu_1 (b + Ax) + (\bar{z} - cx) > 0$  sera dite plus stricte que la contrainte  $\mu_2 (b + Ax) + (\bar{z} - cx) > 0$  pour la solution partielle S si l'on a :

$$\max \left\{ \begin{array}{l} \mu_1 (b + Ax) + (\bar{z} - cx) / x_j = 0 \text{ ou } 1 \quad j \notin S \\ x_j = 0 \text{ ou } 1 \quad j \notin S \end{array} \right\} \leq \max \left\{ \begin{array}{l} \mu_2 (b + Ax) + (\bar{z} - cx) \\ x_j = 0 \text{ ou } 1 \quad j \notin S \end{array} \right\}$$

Les coefficients  $\mu$  que nous cherchons sont donc ceux de la contrainte qui réalise sur tous les  $\mu \geq 0$  le minimum de l'expression.

$$\max \{ \mu(b + Ax) + (\bar{z} - cx) / x_j = 0 \text{ ou } 1, j \notin S \}$$

### 3.3.2. Calcul de la contrainte additionnelle

Il s'agit de résoudre le problème

$$(I) \quad \min_{\mu \geq 0} \max \{ \mu(b + Ax) + (\bar{z} - cx) / x_j = 0 \text{ ou } 1, j \notin S \}$$

Ce problème s'écrit

$$\min_{\mu \geq 0} \sum_{i=1}^m \mu_i b_i^S + (\bar{z} - c_S x^S) + \max_{j \notin S} \left\{ \sum_{i=1}^m \mu_i a_{ij} \dots c_j \right\} x_j \quad x_j = 0 \text{ ou } 1 \}$$

$$\text{avec } b_i^S = b_i + A_i x^S$$

$$\text{Considérons le problème : } \max_{j \notin S} \left\{ \sum_{i=1}^m \mu_i a_{ij} - c_j / x_j = 0 \text{ ou } 1 \right\}$$

Remplaçons le par son dual

$$(II) \quad \left\{ \begin{array}{l} \min \sum_{j \notin S} \omega_j \\ \sum_{i=1}^m \mu_i a_{ij} + \omega_j \leq c_j \quad j \notin S \\ (\omega_j) \end{array} \right.$$

Le problème (I) s'écrit alors :

$$(III) \quad \left\{ \begin{array}{l} \min \sum_{i=1}^m \mu_i b_i^S + (\bar{z} - c_S x^S) - \sum_{j \notin S} \omega_j \\ \sum_{i=1}^m \mu_i a_{ij} + \omega_j \leq c_j \quad j \notin S \\ (\omega_j) \quad j \notin S \\ \mu_i \geq 0 \quad i = 1, m. \end{array} \right.$$

Il est aisé de voir que le problème III obtenu est le dual du problème (LPS)

$$(LPS) \quad \left\{ \begin{array}{l} \min \sum_{j \notin S} c_j x_j + \sum_{j \in S} c_j x_j \\ \sum_{j \notin S} a_{ij} x_j + b_i^S \geq 0 \quad i = 1, m, \\ 0 \leq x_j \leq 1 \quad j \notin S. \end{array} \right.$$

Le problème (LPS) est la relaxation du problème  $P^S$

$$P^S \quad \left\{ \begin{array}{l} \min \sum_{j \notin S} c_j x_j + \sum_{j \in S} c_j x_j \\ \sum_{j \notin S} a_{ij} x_j + b_i^S \geq 0 \quad i = 1, m. \\ x_j = 0 \text{ ou } 1 \quad j \notin S \end{array} \right.$$

En conclusion, les coefficients  $\mu$  sont les valeurs optimales des variables duales de LPS.

### 3.3.3. Remarques

Soit  $\bar{v}(LP^S)$  la valeur optimale de  $LP^S$ . Si  $\bar{v}(LP^S) \geq \bar{z}$  alors aucune complétion de  $S$ , réalisable ou non, ne peut améliorer l'objectif.

Si  $\bar{v}(LP^S) < \bar{z}$ , alors  $P^S$  peut avoir une solution réalisable améliorant  $\bar{z}$ .

Si dans ce second cas, la solution de LPS est entière, alors cette solution est la meilleure complétion de  $S$ . Sinon, la meilleure complétion de  $S$  si elle existe doit vérifier la contrainte spéciale, où les  $\mu$  sont les variables duales à l'optimum de  $LP^S$ .

### 3.4. Schéma de l'algorithme

$$0 - S = \emptyset \quad \bar{z} = +\infty$$

1 - Si  $x^S$  est réalisable aller en 5.

2 - Construire  $T^S$ . Si  $T^S$  est vide aller en 6.

3 - Résoudre  $LP^S$ . Si  $\bar{v}(LP^S) \geq \bar{z}$  aller en 6. Si  $\bar{v}(LP^S) < \bar{z}$  et si  $LP^S$  a une solution entière aller en 5, sinon former la contrainte spéciale.

- 4 - Augmenter  $S$  à l'aide de l'élément de  $T^S$ , qui réalise le maximum de la quantité  $\sum_{i=1}^m \min(y_i^S + a_{ij}, 0)$  sur  $T^S$ . Aller en 1.
- 5 - Si  $c_S x^S < \bar{z}$  alors faire  $\hat{x} = x^S$   $\bar{z} = c_S x^S$ .
- 6 - Localiser l'élément le plus à droite de  $S$  qui n'est pas souligné. Si il n'y en a aucun, aller en 7. Si non, souligner cet élément le remplacer par son complémentaire et aller en 1.
- 7 - Si  $\bar{z} = +\infty$ , alors le problème n'a pas de solution réalisable. Sinon écrire la solution. Fin.

### 3.5. Remarques

Un meilleur choix de  $S^0$ , si possible permet d'accélérer l'énumération.

Il en est de même, si on dispose au départ d'une valeur intéressante de  $\bar{z}$ .

### 3.6. Rapidité de la convergence

GEOFFRION donne dans [8], un ensemble de résultats de tests numériques sur des problèmes dont la taille atteint 90 variables.

Le calcul de la contrainte additionnelle permet d'affirmer que le temps de calcul, pour des problèmes de type recouvrement, knapsack, est une fonction polynomiale du nombre de variables.

C'est fort de cette conclusion que nous avons essayé de le mettre en oeuvre pour résoudre notre problème.

## CHAPITRE IV - MISE EN OEUVRE DE L'ALGORITHME DE GEOFFRION

Avant de poser les programmes mathématiques que nous nous proposons de résoudre, nous allons présenter la numérotation des variables dont nous nous servirons dans toute la suite. Cette numérotation devra tenir compte de la linéarisation et des symétries que peut présenter le coeur.

### 4.1. Prise en compte des symétries et de la linéarisation

Soit  $u$  une configuration d'un coeur donné  $u \in \mathcal{U}$   $u = (u_{ij})_{\substack{j=1,n \\ i=1,n}}$ ,

avec

$$\sum_{i=1}^n u_{ij} = 1 \quad j = 1, n, \quad \sum_{j=1}^n u_{ij} = 1 \quad i = 1, n, \quad u_{ij} = 0 \text{ ou } 1 \quad j, i = 1, n.$$

#### 4.1.1. Prise en compte des symétries

Si le coeur à repositionner présente une symétrie - ce qui sera le cas dans tous les problèmes que nous traiterons - alors les assemblages peuvent être répartis en deux groupes,  $G_1$  et  $G_2$ .  $G_1$  désignant l'ensemble des assemblages situés sur les axes de symétrie,  $G_2$ , les autres. Les éléments de  $G_1$  (resp  $G_2$ ) ne pourront être alors échangés qu'entre eux.

Soit alors :

$$n_1 = / G_1 / \quad \text{et} \quad n_2 = / G_2 / \quad \text{avec} \quad n = n_1 + n_2$$

Sous forme matricielle, la configuration  $u$  pourra s'écrire :

$$u = \left( \begin{array}{c|c} u_1 & 0 \\ \hline 0 & u'_2 \end{array} \right)$$

où  $u_1$  et  $u_2$  sont des matrices carrées d'ordre  $n_1$  et  $n_2$ .

#### 4.1.2. Prise en compte de la linéarisation

Nous avons vu au chapitre 2, qu'en procédant à l'échange d'assemblages de burn-up ou de  $K_\infty$  très différents, l'erreur commise sur le  $K_{\text{eff}}$  ou la puissance en prenant les expressions linéarisées devient très grande. Nous sommes donc amenés à fixer une borne supérieure  $|\Delta K_\infty|^{\text{max}}$  (par exemple) à la variation du  $K_\infty$  lorsque nous procédons à l'échange de deux assemblages.

Par exemple, soit  $K_\infty^i$  le  $K_\infty$  de l'assemblage  $i$ . Nous pourrions échanger l'assemblage  $i$  avec l'assemblage  $j$  si

$$|K_\infty^i - K_\infty^j| \leq |\Delta K_\infty|^{\text{max}}$$

Lorsque la relation ci-dessus n'est pas vérifiée, alors nous dirons que l'échange de l'assemblage  $i$  avec l'assemblage  $j$  est interdit. Les composantes  $u_{ij}$  de  $u$  correspondantes sont donc nulles.

#### 4.1.3. Matrices des cases permises

La détermination de la configuration  $u$  revient d'après 4.1.1. et 4.1.2. à déterminer les sous matrices  $u_1$  et  $u_2$ . Nous appellerons matrice des cases permises, du groupe  $G_1$  (resp  $G_2$ ), la matrice carrée d'ordre  $n_1$  (resp  $n_2$ ) dont tous les éléments sont égaux à 1 sauf ceux correspondants aux échanges interdits du fait de la précision composée pour la linéarisation qui sont fixées à 0. Ces matrices sont indicées en ligne par les numéros des emplacements et en colonne par les numéros des assemblages.

#### 4.2. Numérotation des variables

Soit  $N$  le nombre de composantes de  $u$  non nulles à priori. Partant de  $u_1 = u_{11}$ , en parcourant les lignes, nous les numérotons de 1 à  $N$ .  $N$  sera le nombre de variables du problème. Soit alors :

- pour chaque emplacement  $i$ ,  $\theta_i$  l'ensemble des indices  $j$  des variables  $u_j$  représentant les assemblages pouvant être affectés à cet emplacement
- pour chaque assemblage  $j$ ,  $\psi_j$  l'ensemble des indices  $i$  des variables  $u_i$  représentant les emplacements dans lesquels cet assemblage peut être placé.

Une configuration  $u$  du coeur sera donc un vecteur de  $\mathbb{R}^N$  tel que :

$$u = (u_j)_{j=1,N} \quad u_j = 0 \text{ ou } 1 \quad j = 1, N ;$$

$$\sum_{j \in \theta_L} u_j = 1 \quad L = 1, n ; \quad \sum_{j \in \psi_K} u_j = 1 \quad K = 1, n.$$

#### 4.2. Résolutions des problèmes LC1 et LC2

##### 4.2.1. Notations

$u^0$  désigne la configuration initiale,  $u$  la configuration désirée.

$m$  désigne le nombre d'emplacements où la puissance spécifique est calculée.

$N$  le nombre de variables obtenus en tenant compte des symétries et de la linéarisation.

$n$  le nombre d'assemblages et d'emplacements à affecter.

Les développements au premier ordre s'écrivent en prenant  $u$  dans  $\mathbb{R}^N$

$$K_{\text{eff}}(u) = K_{\text{eff}}(u^0) + \sum_{i=1}^N \frac{\partial K_{\text{eff}}(u^0)}{\partial u_i} (u_i - u_i^0)$$

$$P_j(u) = P_j(u^0) + \sum_{i=1}^N \frac{\partial P_j(u^0)}{\partial u_i} (u_i - u_i^0) \quad j = 1, m.$$

Posons :

$$A' = (a'_{lk})_{\substack{l=1, m \\ k=1, N}} \quad a'_{lk} = \frac{\partial P_l(u^0)}{\partial u_k} \quad l = 1, m ; \quad k = 1, N$$

$$c'_j = (c'_j)_{j=1, N} \quad c'_j = \frac{\partial K_{\text{eff}}(u^0)}{\partial u_j} \quad j = 1, N.$$

$$C = (c_j)_{j=1, N} \quad c_j = c'_j - \max_j c'_j \quad j = 1, N.$$

$$V = \{ u \in \mathbb{R}^N, u_j = 0 \text{ ou } 1, \sum_{j \in \theta_L} u_j = 1 \quad L = 1, n ; \sum_{j \in \psi_K} u_j = 1, K = 1, n \}$$

#### 4.2.2. Mise en forme du problème LC2

Avec les modifications nécessaires, ce problème s'écrit :

$$\left\{ \begin{array}{l} \max K_{\text{eff}}(u^0) + \sum_{j=1}^N c'_j (u_j - u_j^0) \\ P_i(u^0) + \sum_{j=1}^N a'_{ij} (u_j - u_j^0) \leq p^{\max} \quad i = 1, m. \\ u \in V \end{array} \right.$$

En posant :

$$b_i = p^{\max} - \sum_{j=1}^N a'_{ij} u_j^0 - P_i(u^0)$$

$$\alpha(u^0) = K_{\text{eff}}(u^0) - \sum_{j=1}^N c'_j u_j^0$$

$$A = -A' = (a'_{ij})_{\substack{i=1, m \\ j=1, N}}$$

Le problème LC2 se met sous la forme :

$$\begin{array}{l}
 \text{(LC2)} \quad \left\{ \begin{array}{l}
 \alpha(u^0) + \min \sum_{j=1}^N c_j u_j \\
 b_i + \sum_{j=1}^N a_{ij} u_j \geq 0 \quad i = 1, m. \quad (1) \\
 u \in V \quad (2)
 \end{array} \right.
 \end{array}$$

#### 4.2.3. Mise en forme du problème LC1

Avec les modifications nécessaires, ce problème s'écrit :

$$\begin{array}{l}
 \min y \\
 \left\{ \begin{array}{l}
 P_i(u^0) + \sum_{j=1}^N a'_{ij} (u_j - u_j^0) \leq y \quad i = 1, m. \\
 u \in V, \quad y \geq 0
 \end{array} \right.
 \end{array}$$

En posant  $b_i(y) = y - \sum_{j=1}^N a'_{ij} u_j^0$ , le problème LC1 se met sous la forme d'un programme linéaire à variables mixtes.

$$\begin{array}{l}
 \text{LC1} \quad \left\{ \begin{array}{l}
 \min y \\
 b_i(y) + \sum_{j=1}^N a_{ij} u_j \geq 0 \quad i = 1, m. \quad (1) \\
 u \in V \quad (2) \\
 y \geq 0 \quad (3)
 \end{array} \right.
 \end{array}$$

#### 4.2.4. Remarques

Afin d'utiliser l'algorithme de GEOFFRION sans augmenter la taille du tableau A, nous y introduisons un certain nombre de tests nous permettant de décrire au cours de l'énumération les éléments de V.

Nous prendrons en compte la contrainte (3) de LC1 en remarquant que si une solution partielle S n'a aucune complétion réalisable pour  $y = y_k$ , il en est de même pour  $y = y_k$ , avec :

$$y_k' \leq y_k \cdot$$

#### 4.4. Prise en compte de V dans la procédure de GEOFFRION

##### 4.4.1. Exemple

$$n = 5, \quad N = 12.$$

Les contraintes caractérisant V s'écrivent :

$$u_1 = 1 \quad (1)$$

$$u_2 + u_3 = 1 \quad (2)$$

$$u_4 + u_5 + u_6 = 1 \quad (3)$$

$$u_7 + u_8 + u_9 = 1 \quad (4)$$

$$u_{10} + u_{11} + u_{12} = 1 \quad (5)$$

$$u_1 + u_{10} = 1 \quad (6)$$

$$u_2 + u_{11} = 1 \quad (7)$$

$$u_3 + u_6 + u_9 + u_{12} = 1 \quad (8)$$

$$u_4 + u_7 = 1 \quad (9)$$

$$u_5 + u_8 = 1 \quad (10)$$

$$u_j = 0 \text{ ou } 1 \quad j = 1, 12.$$

Partons de  $S^0 = \emptyset$ . Cherchons à augmenter  $S^0$ .

$$\left\{ \begin{array}{l} (1) \\ (2) \end{array} \right. \rightarrow u_1 = 1 \quad \text{et } u_{10} = 0.$$

Afin de ne pas tester les complétions de  $S^0$  pour lesquelles  $u_1 = 0$  et  $u_{10} = 1$  - complétions qui ne nous conduisent pas à une permutation - nous prenons  $S^1 = \{\underline{1}, \underline{-10}\}$ . Le signe - indiquant que  $u_{10} = 0$ , la barre indiquant que la solution partielle dans laquelle  $u_{10}$  (par exemple) vaut 1 a été examinée.

Avec  $S^1$  ainsi formé, essayons à nouveau de l'augmenter.

Si nous prenons  $u_6 = 1$ , alors notre système n'a pas de solution. En effet, supposons  $u_6 = 1$ , alors

$$\left. \begin{array}{l} (3) \\ (8) \end{array} \right\} \rightarrow u_3 = u_4 = u_5 = u_9 = u_{12} = 0$$

Compte tenu de ce résultat

$$(2) \rightarrow u_2 = 1 \rightarrow u_{11} = 0$$

$$(5) \rightarrow u_{11} = 1$$

La relation 7 ne peut être vérifiée. Par conséquent  $u_6 = 0$ . De la même façon nous aurions pu montrer que  $u_9 = 0$ . La solution partielle  $S^2$  que nous prendrons alors sera :

$$S^2 = \{ \underline{1}, \underline{-10}, \underline{-6}, \underline{-9} \} .$$

Nous pouvons obtenir ces résultats plus aisément. Revenons à la matrice des cases permises associée à V. Soit  $D = (d_{jk})$ . Résoudre le système précédent revient en réalité à construire à partir de D une matrice de permutation D'.

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Comme D' n'a qu'un et un seul élément non nul par ligne et par colonne, nous avons nécessairement  $d'_{11} = 1$  et  $d'_{51} = 0$ . Soit  $u_1 = 1$  et  $u_{10} = 0$ .

Pour retrouver le fait que  $u_6 = u_9 = 0$  soit  $d'_{35} = d'_{45} = 0$ , nous faisons appel à la théorie des graphes (ROY [19]).

Considérons D comme la matrice caractéristique du graphe  $G = (X, A)$  avec  $X = \{1, 2, 3, 4, 5\}$ . G a 3 composantes fortement connexes (CFC),  $\{1\}$ ,  $\{2,5\}$ ,  $\{3,4\}$ .

Remarquons que  $d_{35}$ ,  $d_{45}$  sont des arcs reliant deux CFC distincts. Pour montrer que  $d'_{35} = d'_{45} = 0$ , considérons le graphe biparti  $G' = (X, Y, P)$  avec  $|X| = |Y| = 5$ , X ensemble des assemblages, Y ensemble des emplacements. Extraire de D, une matrice de permutation D', c'est trouver si, il existe un couplage maximum de X dans Y. Nous allons donc montrer que si  $d'_{35} = 1$ , il n'y a pas de couplage maximum de X dans Y.

Faisons alors  $d_{35} = 1$ . Nous avons aussi  $d_{11} = 1$ ,  $d_{51} = 0$ ,  $d_{53} = 1 \rightarrow d_{55} = d_{25} = d_{51} = 0$ .

Nous obtenons alors le graphe II ci-dessous à partir du graphe en supprimant les correspondants.



Théorème de Koenig - Hall : Soit  $G = (X, Y, \mathcal{A})$  un graphe triparti,  $|X| = 1$ . Une condition nécessaire et suffisante pour qu'il existe un couplage maximum de  $X$  dans  $Y$  est que

$$\forall T \subset X \quad |\Gamma(T)| \geq |T|$$

Preuve : Voir BERGE [3], page 128.

Appliquons le théorème de Koenig-Hall au graphe II. Si nous pensons par exemple

$T = \{5, 2\}$  alors  $\Gamma(T) = \{2\}$ . Comme  $|\Gamma(T)| = 1 < |T| = 2$  il n'existe pas de couplage de  $X$  dans  $Y$ . Nous devons par conséquent avoir  $d_{35} = 0 = d'_{35}$ .

De la même façon nous avons  $d_{45} = d'_{45} = 0$ .

#### 4.4.2. Génération des éléments de $U$

A l'aide de l'exemple précédent, nous avons introduit dans l'algorithme les modifications suivantes.

##### 4.4.2.1. Modification 1

Soit  $S^k$  une solution partielle. Nous sommes amenés à augmenter  $S^k$  à l'aide de la variable  $j_s$  de sorte que  $u_{j_0} = 1$ .  $S_1^{k+1} = S^k \cup \{j_0\}$

Si  $j_0 \in \theta_{L_0} \cap \psi_{K_0}$  alors  $\forall l \neq j_0 \in \theta_{L_0}, \forall k \neq j_0 \in \psi_{K_0} \quad u_l = u_k = 0$ .

Nous introduisons les indices tels que  $l$  et  $k$  dans  $S$  et nous les soulignons, car leur valeur à 1 ne conduit pas à une permutation du fait que  $u_{j_0} = 1$ .

Nous prendrons alors comme solution partielle  $S^{k+2} = S^{k+1} \cup \{l_1, l_2, \dots, l_m, k_1, \dots, k_p\}$

4.4.2.2. Modification 2  $(l_i)_{i=1, r} \in \theta_{L_0}, (k_i)_{i=1, p} \in \psi_{K_0}$ .

Soit S une solution partielle. Soit TS l'ensemble des variables candidates associées. Pour éliminer de la liste des variables candidates, les indices l qui ne nous conduiraient pas à une permutation si ils étaient mis à 1, nous allons procéder comme dans l'exemple ci-dessus.

Nous construisons D, la matrice des cases permises associées à S, dont tous les éléments sont nuls sauf ceux correspondant aux indices des variables candidates et ceux correspondant aux indices des éléments de S fixés à 1.

Comme dans l'exemple 4.3.1., nous décomposons le graphe associé à D en CFC et nous otions de la liste des variables candidates les indices l correspondants aux arcs reliant deux CFC distinctes.

4.4.2.3. Modification 3

Soit maintenant S une solution partielle, TS l'ensemble des variables candidates associées. Pour nous assurer que S a au moins une complétion qui soit une permutation, il nous suffit de regarder si pour chaque emplacement non saturé il existe au moins une variable candidate correspondant à un assemblage susceptible de lui être affecté.

Remarquons qu'avec la décomposition de G en CFC, cette condition est aussi nécessaire.

4.4.2.4. Remarques pratiques

. Pour ne pas alourdir le programme, nous avons choisi de calculer directement les indices reliant deux CFC. Pour cela à l'aide de l'algorithme de WARSHALL (voir J.P. STEEN [21]) nous calculons la matrice de la fermeture transitive  $M = (m_{ij})_{i=1, n}^{j=1, n}$  du graphe G. Deux sommets seront dans la même CFC lorsque  $\inf(m_{ij}, m_{ji}) = 1$ . Les indices l pour lesquels cette relation n'est pas vérifiée sont donc à éliminer de la liste des variables candidates.

. Dès qu'une variable candidate correspond soit à un assemblage unique à affecter à un emplacement, soit à un emplacement unique pour lequel il n'y a qu'un seul assemblage candidat, l'augmentation de S correspondante est faite.

4.5. Réalisabilité des complétions d'une solution partielle

En réponse à la question 3.2.2., GEOFFRION nous proposait de calculer

$$r_i^S = y_i^S + \sum_{j \in TS} \max(0, a_{ij}) \quad \text{pour } y_i^S < 0.$$

Si  $r_i^S < 0$  alors la solution partielle S n'a aucune complétion réalisable.

La borne  $\alpha = \sum_{j \in TS} \max(0, a_{ij})$  est trop grossière pour notre problème. La

meilleure borne serait l'optimum du problème d'affectation linéaire

$$\left\{ \begin{array}{l} \max \sum_{j \in T^S} a_{ij} u_j \\ \sum_{j \in \theta_L} u_j = 1 \quad L = 1, r \quad \text{non saturé} \\ \sum_{j \in \psi_K} u_j = 1 \quad K = 1, r \quad \text{non saturé} \\ u_j \in T^S \end{array} \right.$$

Pour des raisons de temps de calcul, nous prenons comme borne  $\beta = \sum_{L=1, r} \max_{j \in \theta_L} a_{ij}$  qui est plus grand que la borne idéale mais de calcul plus aisé !

Le test deviendra donc

$$r_1^S = y_i^S + \sum_{L=1, r} \max_{j \in \theta_L} a_{ij} \quad y_i^S < 0$$

Si  $r_1^S < 0$ , aucune complétion de S n'est une permutation réalisable.

#### 4.5.1. Remarque

$$\beta = \min \left( \sum_{L=1, r} \max_{j \in \theta_L} a_{ij}, \sum_{K=1, r} \max_{j \in \psi_K} a_{ij} \right)$$

serait une meilleure borne. Ce calcul n'a pas été programmé.

#### 4.6. Calcul de la contrainte additionnelle

Le calcul de la contrainte spéciale, tel qu'il est exposé au paragraphe 3.3.2. nécessite la prise en compte de la contrainte (2) de P.

Nous modifions ce calcul en prenant en compte la contrainte (2) de LC1 et LC2.

Soit donc S une solution partielle de LC1 ou LC2.  $\bar{z}$  la meilleure borne supérieure connue.

Reprenant la définition de GEOFFRION  $\lfloor \bar{\theta} \rfloor$ , nous dirons que la contrainte  $\mu_1(b + A_U) + (\bar{z} - c_U) > 0$  est plus stricte que la contrainte  $\mu_2(b + A_U) + (\bar{z} - c_U) > 0$  pour la solution partielle S si l'on a

$$\max \{ \mu_1 (b + A_U) + (\bar{z} - c_U) / \sum_{j \in \theta_L} u_j = 1, \sum_{j \in \psi_K} u_j = 1, j \notin S, L, K = 1, r' \}$$

$$\leq \max \{ \mu_1 (b + A_U) + (\bar{z} - c_U) / \sum_{j \in \theta_L} u_j = 1, \sum_{j \in \psi_K} u_j = 1, j \notin S; L, K = 1, r' \}$$

$r'$  nombre total d'emplacements non affectés.

Les coefficients  $\mu$  cherchés réalisent donc le minimum de l'expression

$$\sum_{i=1}^m \mu_i b_i^S + (\bar{z} - e_S K^S) + \max \left\{ \sum_{j \notin S} \left[ \sum_{i=1}^m \mu_i a_{ij} - c_j \right] u_j / \sum_{j \in \theta_L} u_j = 1, \right. \\ \left. \sum_{j \in \psi_K} u_j = 1, j \notin S, u_j = 0 \text{ ou } 1; L, K = 1, r' \right\}.$$

Considérons le problème

$$\max \left\{ \sum_{j \notin S} \left( \sum_{i=1}^m \mu_i a_{ij} - c_j \right) u_j / \sum_{j \in \theta_L} u_j = 1, \sum_{j \in \psi_K} u_j = 1, j \notin S; u_j = 0 \text{ ou } 1, \right. \\ \left. L, K = 1, r' \right\}$$

C'est un problème pour  $\mu$  fixé, d'affectation linéaire. Nous savons - voir SIMMONARD [20] - qu'il a la même solution optimale que le problème obtenu en relâchant la contrainte d'intégrité sur les  $u_j$ .

Ecrivons le dual de ce programme. Il s'écrit :

$$\min \left\{ - \sum_L V_L - \sum_K W_K / V_L + W_K \leq c_j - \sum_{i=1}^m \mu_i a_{ij} \quad (V_L, W_K, j \notin S, L, K=1, r') \right\}$$

Trouver les  $\mu$ , revient donc à résoudre le problème :

$$\left. \begin{array}{l} \min \sum_{i=1}^m \mu_i b_i^S + (\bar{z} - c_S u^S) - \sum_L V_L - \sum_K W_K \\ \sum_{i=1}^m \mu_i a_{ij} + V_L + W_K \leq c_j \quad j \notin S, L \in \theta_L, K \in \psi_K \\ (V_L) \quad L = 1, r', \\ (W_K) \quad K = 1, r', \\ \mu_i \geq 0 \quad i = 1, m. \end{array} \right\} \quad \text{(L'PS)}$$

Comme au paragraphe 3.3.2., ce problème est le dual du problème

$$\begin{array}{l}
 P, S \\
 \left\{ \begin{array}{l}
 \min \sum_{j \in S} c_j u_j + \sum_{j \notin S} c_j u_j \\
 b_i^S + \sum_{j \notin S} a_{ij} u_j \geq 0 \quad i = 1, m, \\
 \sum_{j \in \theta_L} u_j = 1 \quad L = 1, r', \\
 \sum_{j \in \psi_K} u_j = 1 \quad K = 1, r', \\
 0 \leq u_j \leq 1 \quad j \notin S.
 \end{array} \right.
 \end{array}$$

#### 4.6.1. Remarque

La contrainte additionnelle, telle que nous la présentons ne concerne que la puissance spécifique.

Dans le cas du problème LC1, nous prendrons  $\bar{z} = 0.$ , et  $c = 0.$  . Dans ce cas, la contrainte n'est calculée que pour un  $y$  fixé.

Dans le cas du problème LC2, lorsque  $\bar{z} = +\infty$ , nous calculons la contrainte additionnelle de BALAS. BALAS [22] et TAHA [22]. Nous présentons à l'annexe A, l'adaptation de ce calcul à notre cas.

## 5 - MISE EN ŒUVRE NUMÉRIQUE

Dans ce paragraphe, nous exposons les paramètres que nous avons choisi pour la résolution numérique du problème.

### 5.1. La solution de départ

Comme solution de départ, nous avons retenu la solution partielle,  $S$ , obtenue en prenant comme variables fixées à 0 ou à 1 dans la base optimale du programme linéaire  $P^S$  avec  $S = \emptyset$ .

En général, cette solution n'est pas éloignée de la solution optimale du problème. Ce choix s'est révélé efficace car dans les autres cas,  $S$  vide au départ par exemple, le temps de calcul augmente d'environ 20 %.

### 5.2. Augmentation de la solution partielle

A chaque itération, lors de l'examen des complétions d'une solution partielle, nous augmentons  $S$  d'une ou plusieurs unités :

- . Ainsi lorsque  $u_j \in \theta_L$  (ou  $u_j \in \psi_K$ ) tel que  $|\theta_L| = 1$  nous fixons d'office  $u_j$  à 1 et nous introduisons  $j$  dans  $S$ , et nous le soulignons.
- . Après le calcul de la contrainte additionnelle, lorsque la solution arrondie n'est pas réalisable, nous introduisons dans  $S$  toutes les variables dont la valeur a été fixée à 0 ou à 1 à l'optimum du programme linéaire résolvant  $P^S$ .
- . Dans tous les autres cas, nous utilisons la procédure d'augmentation proposée par BALAS [1].

### 5.3. La résolution de $P^S$

$P^S$  est résolu par la méthode duale-simpliciale du simplexe, telle qu'elle est exposée dans HUARD [11] ou SIMMONARD [20]. A chaque étape nous calculons tous les éléments du tableau simplifical, la variable entrant dans la base étant celle de plus petit pivot, la sortante le plus petit second membre.

En plus des habituelles règles de non cyclage, pour faire face aux problèmes de précision numériques nous utilisons en plus de la double précision, les recommandations de H. MUELLER MERBACH [16], qui préconise un calcul d'erreur sur chaque élément du tableau simplicial.

Bien que cette modification conduise à des temps de calculs plus longs, les résultats que nous obtenons sont plus stables numériquement et satisfaisants dans l'ensemble.

### 5.4. Fréquence de calcul de la contrainte additionnelle

Numériquement, l'expérience montre que le calcul d'une contrainte additionnelle coûte cher et elle n'apporte pas toujours des résultats décisifs permettant de conclure rapidement la recherche des complétions d'une solution partielle. Nous nous en sommes par conséquent tenus à la calculer le moins souvent possible, et uniquement lorsque le nombre de variables libres est encore grand.

Cependant, elle est toujours calculée pour  $S = \emptyset$ , ce qui permet d'obtenir des bornes globales sur le  $k_{\text{eff}}$  et  $f$ . Ces bornes nous servent alors de seuils limites pour l'optimisation.

### 5.5. Arrêt des calculs

Il arrive souvent, en optimisation, en nombre entier, que l'on passe plus de temps à vérifier qu'une solution est optimale qu'à la trouver. Afin d'éviter cet inconvénient, nous considérerons comme optimale, toute solution réalisable dont la valeur est à un pourcentage  $\alpha$  donné de la valeur de l'optimum continu obtenu en résolvant  $P^S$  avec  $S = \emptyset$ .

## 6. SCHEMA DE L'ALGORITHME

Avec les remarques que nous avons faites pour adapter la méthode de GEOFFRION au problème que nous avons à résoudre nous obtenons l'algorithme ci-dessous.

- 1 -  $S = \emptyset$   $\bar{z} = +\infty$
- 2 - Si  $u^S$  n'est pas une permutation aller en 3 - Si  $u^S$  est une permutation réalisable, aller en 9. Sinon aller en 10.
- 3 - Augmenter  $S$  à l'aide des indices  $j$ , des variables ayant des valeurs fixées pour que  $u^S$  soit le début d'une permutation. Calculer la liste des variables candidates. Si la liste est vide, aller en 10.
- 4 - Calculer la fermeture transitive de  $D$ , matrice des cases permises associées à  $S$ . Eliminer, si elles existent, les variables reliant deux composantes fortement connexes distinctes, de la liste des variables candidates. Si la nouvelle liste est vide aller en 10.
- 5 -  $S$  admet-elle une complétion qui soit une permutation ? Si oui, aller en 6. Si non aller en 10.
- 6 - Si  $S$  peut admettre des complétions vérifiant les contraintes, alors aller en 7. Si non aller en 10.
- 7 - Faut-il calculer la contrainte additionnelle ? Si non, aller en 8. Si la solution du programme linéaire est entière, ou si la situation arrondie obtenue est réalisable, aller en 9. Si  $S$  a été modifié aller en 2. Si  $V(LPS) \geq \bar{z}$  alors aller en 10.
- 8 - Augmenter  $S$  par un indice  $j$  d'une variable  $u_j$  vérifiant le critère de BALAS. Aller en 2.
- 9 -  $\hat{u} \leftarrow u^S$ . Si on résoud LC1, calculer le nouveau facteur de forme obtenu. Modifier les contraintes du problème.  
Si on résoud LC2, faire  $\bar{z}$  égal à l'optimum obtenu.  
Si l'optimum obtenu est à  $\alpha\%$  de l'optimum du premier programme linéaire aller en 11.
- 10 - Localiser l'élément le plus à droite de  $S$ . Il n'en existe pas aller en 11. Si non, le souligner. Extraire de  $S$  tous les autres éléments soulignés à droite du premier nom souligné. Si la solution partielle  $S$  ainsi obtenue est de cardinal inférieur à celui correspondant à la dernière contrainte calculée, retirer cette contrainte de la liste des contraintes du problème. Aller en 2.
- 11 - Si  $\bar{z} = +\infty$  ou si il n'y a pas de solution améliorant le facteur de forme, alors le problème n'a pas de solution. FIN. STØP. Si non, imprimer la solution optimale, le  $k_{\text{eff}}$  ou le facteur de forme optimal. FIN. STØP.

## 7. CONCLUSIONS

L'introduction du calcul de la CFC n'apporte pas une diminution importante de la durée de résolution d'un programme.

Plus on cherche à obtenir une valeur précise de l'optimum, plus le temps de calcul augmente.

Plus le nombre de variable diminue, plus l'algorithme converge rapidement.

Moins on calcule de contraintes additionnelles, plus vite l'algorithme converge.

De tous ces constats, nous avons estimé qu'il était bon de laisser à l'utilisateur le choix du pourcentage  $\alpha$ , ainsi que le nombre de variable à introduire, compte tenu de la linéarisation.

Au chapitre nous présentons les résultats numériques obtenus en appliquant la méthode préconisée au chapitre 2.

## CHAPITRE V - RÉSULTATS NUMÉRIQUES

Nous avons testé les deux méthodes exposées au chapitre 2 sur deux types de coeur de réacteur à eau pressurisée (PWR). L'un contenant des assemblages d'un seul type, l'autre des assemblages de deux types.

Les notations ci-dessous permettront nous l'espérons, une lecture compréhensible des tableaux et des figures.

### 1. NOTATIONS

Dans toute la suite, nous désignerons par :

G1 : l'ensemble des assemblages situés sur les axes de symétrie,

G2 : l'ensemble des assemblages non situés sur les axes,

$f^0$  : la valeur exacte du facteur de forme,

$K_{eff}^0$  : la valeur exacte du coefficient de multiplication effectif,

$\tilde{f}$  : la valeur du facteur de forme obtenue à partir du développement limité sur l'ensemble discret

$\tilde{K}_{eff}$  : la valeur du  $K_{eff}$  obtenue à partir du développement limité sur l'ensemble discret,

$K_{eff}^c, f^c$  : les valeurs de  $K_{eff}$  et de  $f$  calculées à l'aide des développements limités sur l'enveloppe convexe,

$|\Delta\tau|^{max}, |\Delta K_{\infty}|^{max}$  : les variations maximum de burn-up et de  $K_{\infty}$  tolérées ou pour des échanges,

Tepu : la durée en secondes sur IBM 3038 d'une itération

Nous illustrons chaque configuration optimale par sa distribution de puissance et sa distribution de burn-up (ou de  $K_{\infty}$ ).

### 2. REPOSITIONNEMENT D'UN COEUR CONTENANT DES ASSEMBLAGES DE MEME TYPE

$$G1 = \{2, 3, 4, 5, 6, 7, 9, 15, 19, 22\}$$

$$G2 = \{10, 11, 12, 13, 14, 16, 17, 18, 20, 21\}$$

La figure V.1 donne la numérotation des assemblages. La figure V.2 la distribution initiale de burn-up.

44
----

Numéro  
de l'élément

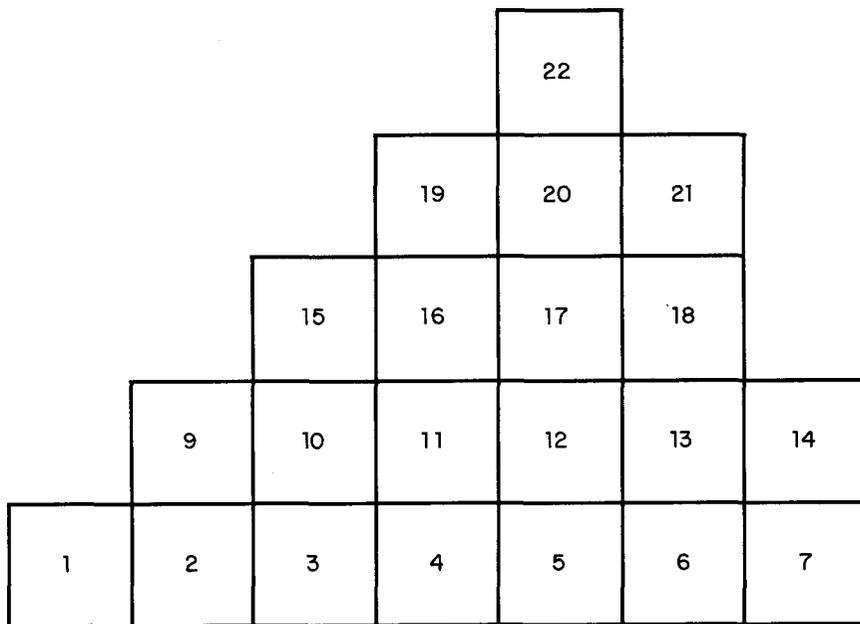


Figure V.1



8 888	Burn-up
-------	---------

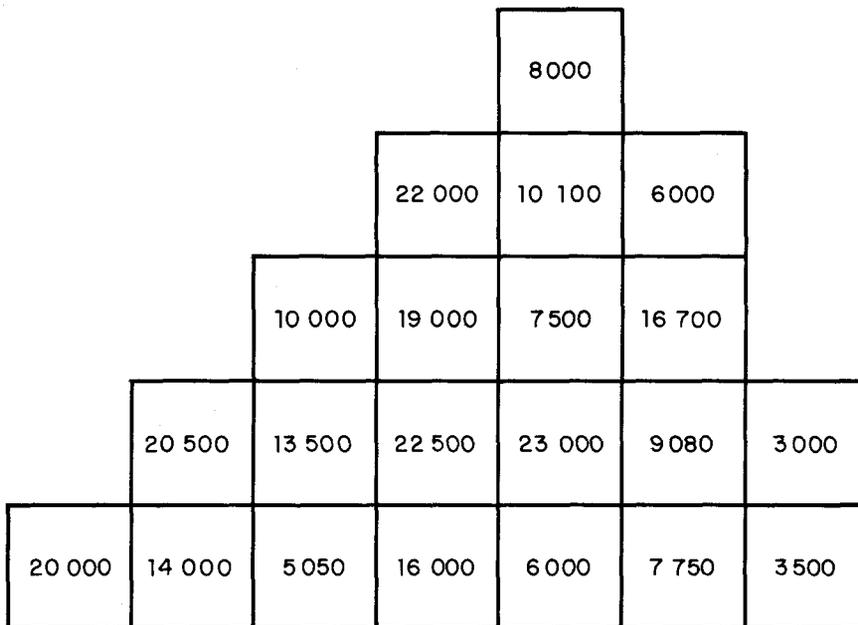


Fig V. 2 - Facteur de forme 1.782

Keff 1.09882



## 2.1. Résolution d'une suite de programmes linéaires à variables booléennes

### 2.1.1. Problème C2

$$\left\{ \begin{array}{l} \max K_{\text{eff}}(u) \\ p(u) \leq p^{\max} \\ u \in \mathcal{U} \end{array} \right.$$

$$p^{\max} = 1.30 \quad |\Delta\tau|^{\max} = 6000 \text{ MWj/t.}$$

En 5 itérations - voir tableau V.1 - nous obtenons un  $K_{\text{eff}}$  de 1.11406.

Nous réalisons un gain de 1523 pcm sur la configuration de départ. Voir figure V.3 et figure V.4.

Dans  $\bar{5}$  on obtient - voir tableau V.2 - un  $K_{\text{eff}}$  de 1.11618 soit un gain de 1736 pcm.

Les temps de calculs de notre méthode sont élevés, mais non prohibitifs.

### 2.1.2. Problème C1

$$\left\{ \begin{array}{l} \min \max_{i=1,21} p_i(u) \\ u \in \mathcal{U} \end{array} \right.$$

$$|\Delta\tau|^{\max} = 6000 \text{ MWj/t.}$$

Au bout de deux itérations - tableau V.3 - nous obtenons un facteur de forme de 1.221. Voir figure V.6. Il est meilleur que celui obtenu dans  $\bar{5}$  par la méthode de recherche directe.

Partant de la configuration de 1.221 de facteur de forme, en 5 itérations - voir tableau V.4, figure V.5 et figure V.7 - nous obtenons un facteur de forme de 1.205, en imposant  $|\Delta\tau|^{\max} = 4000 \text{ MWj/t.}$

Ce résultat est proche du meilleur résultat obtenu dans  $\bar{5}$  par la méthode dite de recherche aléatoire. On y obtient un facteur de forme de 1.203.

$K_{\text{eff}}^0$	$\tilde{K}_{\text{eff}}$	$f^0$	$\tilde{f}$	$t_{\text{epu}}$
1.09882	1.09613	1.782	1.280	9.19
1.10266	1.10644	1.325	1.296	6.23
1.10732	1.10726	1.293	1.295	17.51
1.10822	1.11323	1.306	1.299	4.42
1.11406	-	1.295	-	-

Tableau V.1  $p^{\text{max}} = 1.30$   
 $|\Delta\tau|^{\text{max}} = 6000 \text{ MWj/t}$

$K_{\text{eff}}^0$	$\tilde{K}_{\text{eff}}$	$f^0$
1.10052	1.10601	1.336
1.00	1.10890	1.308
1.11097	1.11124	1.300
1.11265	1.11493	1.298
1.11514	1.11607	1.296
1.11618	1.11618	1.293

Tableau V.2  $p^{\text{max}} = 1.30$   
 $|\Delta\tau|^{\text{max}} = 6000 \text{ MWj/t}$



6666	Burn - up
------	-----------

				5050		
			10 000	3 000	22 500	
		16 000	13 500	7 500	23 000	
	22 000	19 000	16 700	9 080	6 000	10 100
20 000	7 750	6 000	20 500	14 000	3 500	8 000

Figure V. 3 - Facteur de forme 1.295

$K_{eff}$  1.11406



.0	.0	.0	.0	.0	.0	.0	.0
.6684	.4897	.0	.0	.0	.0	.0	.0
1.289	1.082	.5898	.3778	.0	.0	.0	.0
1.249	1.290	1.205	1.114	.6640	.0	.0	.0
1.071	1.119	1.219	1.295	1.114	.3778	.0	.0
1.286	1.044	1.092	1.219	1.205	.5898	.0	.0
1.269	1.024	1.044	1.119	1.290	1.082	.4897	.0
1.125	1.269	1.286	1.071	1.249	1.289	.6684	.0

Figure V-4

.0	.0	.0	.0	.0	.0	.0	.0
.7687	.5404	.0	.0	.0	.0	.0	.0
1.205	.9895	.9241	.7010	.0	.0	.0	.0
1.179	1.010	1.202	1.204	.7537	.0	.0	.0
1.006	1.029	1.131	1.170	1.204	.7010	.0	.0
1.178	.9995	1.200	1.131	1.202	.9241	.0	.0
1.100	.9773	.9995	1.029	1.010	.9895	.5404	.0
1.010	1.100	1.178	1.006	1.179	1.205	.7687	.0

Figure V-5



3 999	Burn-up
-------	---------

				3 500		
			14 000	9 080	3 000	
		7 750	19 000	7 500	13 500	
	20 500	23 000	16 700	22 500	10 100	6 000
20 000	16 000	6 000	22 000	10 000	8 000	5 050

Figure V. 6 - Facteur de forme 1.221

$K_{eff}$  1.09209



$f^S$	$f^E$	$\hat{f}$	$ f^C - \hat{f} /f^C$	$T_{cpu} (s)$
1.782	1.230	1.277	0.039	17.50
1.313	1.198	1.222	0.021	17.81
1.221	*	*	*	> 210

Tableau V.3 Problème min f

$$|\Delta\tau|^{\max} = 6000 \text{ MWj/t}$$

$f^S$	$\hat{f}^E$	$\hat{f}^C$	$ f^C - \hat{f} /f^C$	$T_{cpu} (s)$
1.221	1.217	1.192	0.020	9.04
1.215	1.214	1.187	0.022	30.70
1.214	1.211	1.184	0.022	6.8
1.208	1.203	1.180	0.011	101.00
1.205	-	1.181	-	120

Tableau V.4 Problème min f

$$|\Delta\tau|^{\max} = 4000 \text{ MWj/t}$$



8 888	Burn-up
-------	---------

				5 050			
			16 000	7 500	3 000		
		8 000	19 000	10 100	6 000		
	20 500	23 000	16 700	22 500	13 500	9 080	
20 000	14 000	7 750	22 000	10 000	6 000	3 500	

Figure V. 7 - Facteur de forme 1.205

$K_{eff}$  1.09168

BUS  
LILLE

### 2.1.3. Conclusion

Cette méthode est coûteuse en temps de calcul. A chaque itération, nous obtenons une configuration "optimale" étant donné la non convexité du domaine, elle n'est pas toujours réalisable pour le problème C2.

Les résultats obtenus sont voisins de ceux obtenus dans [5]. Les configurations optimales que nous obtenons diffèrent largement de celles de [5].

Afin de diminuer le temps de calcul, nous avons observé qu'il faut procéder de la manière suivante :

Faire les 2 ou 3 premières itérations avec un pas  $|\Delta K_\infty|^{\max}$  (ou  $|\Delta \tau|^{\max}$ ) assez grand, en exigeant que l'optimum des programmes linéaires booléens soit assez "grossier".

Faire les itérations suivantes en diminuant le pas  $|\Delta K_\infty|^{\max}$  (ou  $|\Delta \tau|^{\max}$ ), mais en exigeant une bonne précision pour l'optimum du programme linéaire booléen.

## 2.2. Approximation discrète de l'optimum continu

### 2.2.1. Problème C2

$$\left\{ \begin{array}{l} \max K_{\text{eff}}(u) \\ p(u) \leq p^{\max} \\ u \in \mathcal{U} \end{array} \right.$$

Nous appliquons la MAP avec  $p^{\max} = 1.25$

$$|\Delta K_\infty|^{\max} = 0.070$$

La configuration "idéale" présente un  $K_{\text{eff}}$  de 1.11703 voir tableau V.5, figure V.8.

Nous approchons l'optimum continu comme indiqué au chapitre 2; avec  $p^{\max} = 1.30$ .

L'optimum discret obtenu présente un  $K_{\text{eff}}$  égal à 1.11574, avec un facteur de forme de 1,291.

Il est atteint en 49,99 s. On a amélioré le  $K_{\text{eff}}$  initial de 1692 pcm. Voir figure V.9 et V.10.

DUMAS réalise un gain de 1964 pcm. Soit 272 pcm de mieux mais avec un facteur de forme de 1,309. Sa solution n'est donc pas réalisable.

$K_{\text{eff}}^0$	$f^0$	$\tilde{K}_{\text{eff}}$	$T_{\text{cpu}} \text{ (s)}$
1.09882	1.782	1.09748	11.31
1.10231	1.290	1.10588	13.12
1.10667	1.253	1.11244	17.19
1.11345	1.253	1.11648	21.18
1.11684	1.251	1.11702	21.85
1.11702	1.250	1.11702	23.26
1.11703	1.250	-	-

Tableau V.5

MAP  $|\Delta K_{\infty}|^{\text{max}} = 0.070$

$p^{\text{max}} = 1.250$

problème max  $K_{\text{eff}}$



.0	.0	.0	.0	.0	.0	.0	.0
.6221	.4549	.0	.0	.0	.0	.0	.0
1.246	1.070	.5886	.3241	.0	.0	.0	.0
1.250	1.250	1.139	.9192	.4799	.0	.0	.0
1.250	1.142	1.250	1.248	.9192	.3241	.0	.0
1.250	1.250	1.250	1.250	1.189	.5886	.0	.0
1.245	1.250	1.250	1.142	1.250	1.070	.4549	.0
1.128	1.245	1.250	1.250	1.250	1.246	.6221	.0

Figure V-8

.0	.0	.0	.0	.0	.0	.0	.0
.6521	.4348	.0	.0	.0	.0	.0	.0
1.177	.9983	.5880	.3542	.0	.0	.0	.0
1.217	1.284	1.270	1.051	.5977	.0	.0	.0
1.255	1.156	1.205	1.291	1.051	.3542	.0	.0
1.113	1.223	1.228	1.205	1.270	.5880	.0	.0
1.226	1.087	1.223	1.156	1.284	.9983	.4348	.0
1.114	1.226	1.113	1.255	1.217	1.177	.6521	.0

Figure V-9



6 666	Burn-up
-------	---------

				6 000		
			7 750	6 000	23 000	
		14 000	16 700	3 000	22 500	
	22 000	10 100	19 000	9 080	7 500	13 500
20 000	8 000	20 500	10 000	16 000	5 050	3 500

Figure V. 10 - Facteur de forme 1.291

$K_{eff}$  1.11574



2.2.2. Problème C1

$$\left\{ \begin{array}{l} \min \max_{i=1,21} p_i(u) \\ u \in \mathcal{U} \end{array} \right.$$

Nous appliquons la MAP en imposant  $[\Delta K_\infty]^{\max} = 0.070$ .

Nous obtenons - voir tableau V.6. - un facteur de forme de 1.144. Voir figure V.11.

Nous approchons de manière discrète cette configuration idéale.

Nous obtenons une configuration de facteur de forme 1.200 en 24,92 s. Voir figure V.13.

En 89 s nous obtenons une configuration de facteur de forme 1.199 - voir figure V.12 et figure V.14.

Le meilleur facteur de forme obtenu pour ce problème dans  $\llbracket 5 \rrbracket$  est 1.201. Voir figure V.15.

$f^0$	$K_{\text{eff}}^0$	$f^*$	$T_{\text{cpu}} \text{ (s)}$
1.782	1.09882	1.220	11.10
1.255	1.09506	1.185	14.19
1.188	1.09194	1.155	20.34
1.160	1.09260	1.144	21.96
1.148	1.09272	1.144	27.34
1.144	1.09300	1.144	22.71
1.144	1.09301	-	-

Tableau V.6

MAP

Problème Min  $f$ 

$$|\Delta K_{\infty}|^{\max} = 0.070$$



.0	.0	.0	.0	.0	.0	.0	.0
.7234	.5416	.0	.0	.0	.0	.0	.0
1.144	1.112	.9223	.6273	.0	.0	.0	.0
1.144	1.114	1.144	1.140	.7336	.0	.0	.0
1.144	1.038	1.029	1.144	1.140	.6273	.0	.0
1.144	1.020	1.144	1.029	1.144	.9223	.0	.0
1.144	1.144	1.020	1.038	1.114	1.112	.5416	.0
1.069	1.144	1.144	1.144	1.144	1.144	.7234	.0

Figure V-11

.0	.0	.0	.0	.0	.0	.0	.0
.7646	.5565	.0	.0	.0	.0	.0	.0
1.193	1.115	.9695	.6436	.0	.0	.0	.0
1.072	1.096	1.139	1.163	.7579	.0	.0	.0
1.196	1.026	1.020	1.101	1.163	.5486	.0	.0
1.088	1.020	1.199	1.020	1.139	.9695	.0	.0
.9880	1.151	1.020	1.026	1.096	1.115	.5565	.0
.9316	.9880	1.000	1.196	1.072	1.193	.7646	.0

Figure V-12

.0	.0	.0	.0	.0	.0	.0	.0
.7737	.5676	.0	.0	.0	.0	.0	.0
1.199	1.147	.9706	.6281	.0	.0	.0	.0
1.030	1.082	1.135	1.168	.7597	.0	.0	.0
1.059	.9894	1.025	1.145	1.168	.6281	.0	.0
1.145	1.032	1.200	1.025	1.135	.9706	.0	.0
1.027	1.183	1.032	.9894	1.082	1.147	.5676	.0
.9625	1.027	1.145	1.059	1.030	1.199	.7737	.0

Figure V-13



6 666
-------

Burn - up

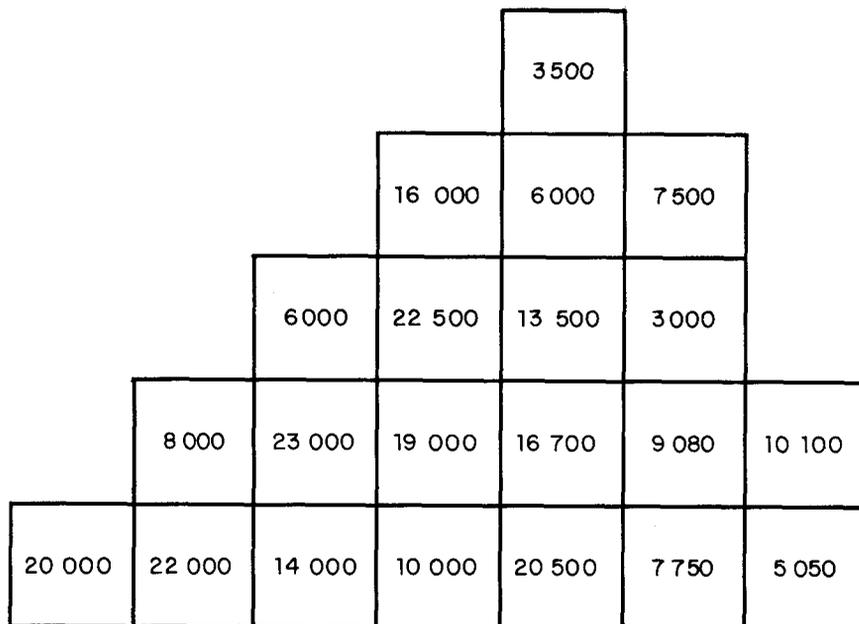


Figure V. 14 - Facteur de forme 1.199

$K_{eff}$  1.09301



8 888
-------

Burn - up

				5 050		
			8 000	13 500	6 000	
		6 000	23 000	10 100	3 000	
	14 000	22 500	19 000	16 700	9 080	7 500
20 000	22 000	7 750	16 000	20 500	10 000	3 500

Figure V.15 - Facteur de forme 1.201

K<sub>eff</sub> 1.09138

### 2.2.3. Conclusion

L'approximation discrète de l'optimum continu que donne la MAP, apparaît comme la méthode appropriée pour résoudre les problèmes C1 et C2.

Elle donne de bons résultats.

Les temps de calcul sont peu importants, du fait notamment qu'on ne résoud qu'un seul problème linéaire à variable booléenne.

Ici encore, les configurations optimales obtenues sont assez éloignées de celles données dans [5].

Les résultats sont à la précision des calculs, identiques mais meilleurs pour le problème C1.

### 3. REPOSITIONNEMENT D'UN COEUR CONTENANT DEUX TYPES D'ASSEMBLAGES

Le coeur à repositionner contient des assemblages d'uranium et de plutonium. Figure V.

#### 3.1. Validité de la linéarisation

Afin de déterminer les assemblages pouvant être permutés entre eux, nous avons observé les phénomènes suivants :

Si l'on échange un assemblage d'uranium et un assemblage de plutonium de même  $K_{\infty}$  on observe une variation de plus de 20% de la puissance exacte. De plus, on commet une erreur de plus de 30% si l'on remplace la puissance exacte par son expression linéarisée.

Nous en déduisons donc que le  $K_{\infty}$  n'est pas le paramètre adéquat pour caractériser de manière uniforme un lot contenant des assemblages de type différent.

En l'absence d'un tel paramètre, nous sommes amenés à conclure que les méthodes exposées ne s'appliquent plus car la linéarisation n'est plus valable.

Le coeur à répositionner constitue donc le cas "limite" à partir duquel, il n'est plus possible de recourir à la linéarisation pour résoudre le problème du repositionnement.

#### 3.2. Remarque

Bien entendu, il est toujours possible de "faire quelque chose". Ainsi par exemple, n'autoriser les échanges qu'entre assemblages du même type. Nous avons pu obtenir un facteur de forme de 1.250 voir figure V.17 et tableau V.7.

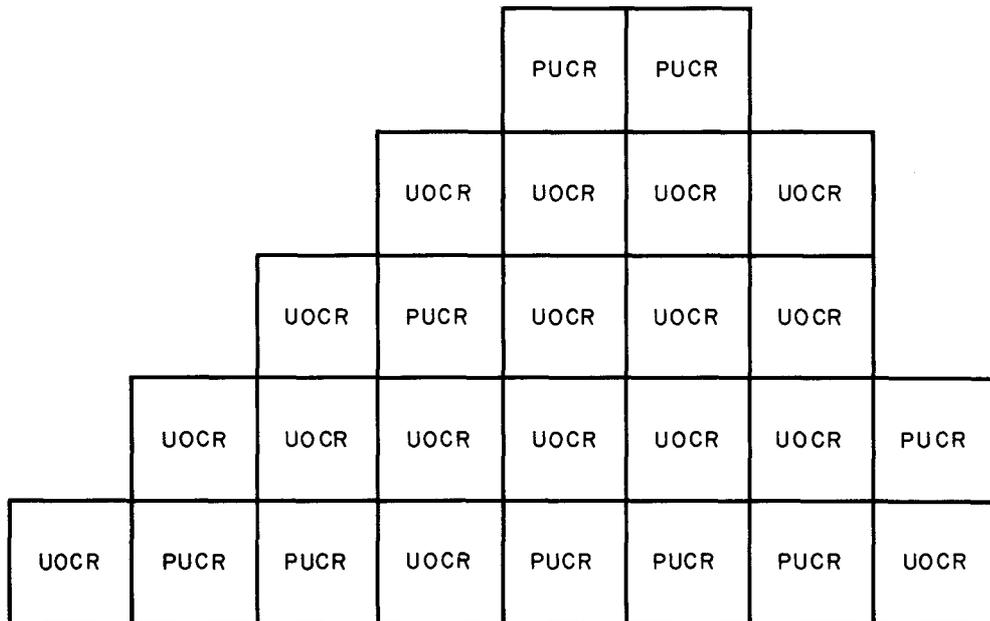


Figure V. 16

.0	.0	.0	.0	.0	.0	.0	.0	.0
.6744	.6740	.0	.0	.0	.0	.0	.0	.0
1.162	1.156	1.014	.6551	.0	.0	.0	.0	.0
1.241	1.025	1.236	1.136	.6390	.0	.0	.0	.0
1.187	1.113	.9358	1.032	.8207	.6390	.0	.0	.0
1.137	.9996	1.250	.8547	1.032	1.136	.6551	.0	.0
1.121	1.055	.8790	1.250	.9353	1.236	1.014	.0	.0
1.193	.9329	1.055	.9996	1.113	1.025	1.156	.6740	.0
.8275	1.193	1.121	1.137	1.187	1.241	1.162	.6744	.0

Figure V-17

$f^0$	$f^c$	$\lambda_f$	$ f^c - \lambda_f  / f^c$	$T_{cpu}$
1.258	1.245	1.249	0.003	2.41
1.253	1.250	1.251	0.008	2.13
1.250	1.250	1.250	0	2.13

Tableau V.7



## CONCLUSION

Nous avons montré qu'en définissant la position d'un assemblage  $i$  dans le coeur d'un réacteur nucléaire par  $n$  variables booléennes  $u_{ij}$  telles que

$$u_{ij} = 1 \quad \text{si } i \text{ est à la place } j$$

$$u_{ij} = 0 \quad \text{si non}$$

Le problème du repositionnement du combustible se formule aisément comme un problème d'affectation non linéaire.

En nous appuyant sur les principes de linéarisation utilisés par DUMAS, nous avons pu développer des méthodes de résolution nécessitant la mise en oeuvre d'algorithme de programmation linéaire à variables booléennes. C'est dans ce cadre que nous avons adapté l'algorithme de BALAS-GEOFFRION à la résolution de programmes d'affectation linéaire avec contraintes.

La méthode directe - résolution d'une suite de programmes linéaires booléens - fournit de "bons" résultats à condition de limiter le domaine linéarisé de manière à diminuer le temps de calcul.

La MAP, suivie de l'approximation discrète est, comme l'observait déjà DUMAS [5], la méthode la plus efficace. Elle fournira toujours de meilleurs résultats en des temps de calcul peu élevés.

Les modules correspondants à ces méthodes ont été développés dans le code NEPTUNE. Ils permettent de proposer des configurations de chargement pour des coeurs contenant des assemblages d'enrichissements différents.

Contrairement à nos espérances, nous ne pouvons tirer plus de la linéarisation. Le problème du repositionnement d'un coeur contenant deux types d'assemblages, uranium d'une part, plutonium d'autre part, reste un problème ouvert.

## ANNEXE A - LA CONTRAINTE DU FILTRE DE BALAS [2\_7]

Le calcul de la contrainte additionnelle de GEOFFRION ne présente d'intérêt que si l'on dispose d'une solution réalisable. En général au cours des premières itérations, on ne dispose pas d'une telle solution.

Pour faire face à une telle situation, nous calculons la contrainte du filtre de BALAS. Mais cette fois, nous sommes obligés de tenir compte explicitement des contraintes d'égalité.

Dans ces conditions, le problème relaxé LPS associé au problème  $P^S$  défini au chapitre IV s'écrit :

$$\text{LPS} \quad \left\{ \begin{array}{l} \max \quad \sum_{j \notin S} c_j x_j \\ \sum_{j \notin S} a_{ij} x_j - b_i^S \leq 0 \quad i = 1, m \\ x_j \geq 0 \quad j \notin S \end{array} \right.$$

après des transformations évidentes.

Définition :

La contrainte  $\sum_{i=1}^m \mu_i (\sum_{j \notin S} a_{ij} x_j - b_i^S) \leq 0$  sera dite plus stricte quand  $\mu = \mu^1$  que lorsque  $\mu = \mu^2$  si on a

$$z(\mu^1) < z(\mu^2), \text{ avec :}$$

$$\forall \mu \geq 0 \quad z(\mu) = \max \left\{ \sum_{j \notin S} c_j x_j \mid x_j \in DS(\mu) \right\}$$

$$\text{et } DS^1 = \left\{ x_j, j \notin S / \sum_{j \notin S} a_{ij} x_j - b_i^S \leq 0 \quad i = 1, m ; x_j \geq 0 \right\}$$

$$DS(\mu) = \left\{ x_j \geq 0, j \notin S / \sum_{i=1}^m (\sum_{j \notin S} a_{ij} x_j - b_i^S) \mu_i \leq 0 \right\}$$

Suivant cette définition, la contrainte la plus stricte sera obtenue pour  $\mu = \mu^0$  tel que  $z(\mu^0) = \min_{\mu \geq 0} z(\mu)$ .

Lemme :  $\mu^0$  est à une constante multiplicative près, la variable duale optimale de LPS.

Preuve : Soit  $\bar{\mu}$  la variable duale optimale de LPS. Par définition de  $\mu^0$ , nous avons :

$$z(\mu^0) \leq z(\bar{\mu}). \text{ Comme}$$

$$\forall \mu \geq 0 \quad DS1 \subset DS(\mu)$$

alors

$$\forall \mu \geq 0 \quad z(\mu) \geq \max \left\{ \sum_{j \notin S} c_j x_j / \sum_{j \notin S} a_{ij} x_j - b_i^S \leq 0 \quad i = 1, m, x_j \geq 0 \right\}$$

Soit  $\bar{x}$  la variable optimale de LPS.  $\bar{x} \in DS(\bar{\mu})$ . Soit  $x \in DS(\bar{\mu})$ . Nous avons :

$$\begin{aligned} \sum_{j \notin S} c_j x_j - \sum_{j \notin S} c_j \bar{x}_j &= \sum_{j \notin S} c_j x_j - \sum_{i=1}^m \bar{\mu}_i b_i^S \\ &\leq \sum_{j \notin S} \left( \sum_{i=1}^m \bar{\mu}_i a_{ij} \right) x_j - \sum_{i=1}^m \bar{\mu}_i b_i^S \leq 0 \end{aligned}$$

car d'après les conditions d'optimalité

$$\forall j \notin S \quad c_j - \sum_{i=1}^m \bar{\mu}_i a_{ij} \leq 0$$

$$\sum_{j \notin S} c_j \bar{x}_j = \sum_{i=1}^m \bar{\mu}_i b_i^S$$

Nous en déduisons donc que

$$\forall x \in DS(\bar{\mu}) \quad \sum_{j \notin S} c_j x_j \leq \sum_{j \notin S} c_j \bar{x}_j$$

Comme  $\bar{x} \in DS(\bar{\mu})$ , nous avons :

$$z(\bar{\mu}) = \max \left\{ \sum_{j \notin S} c_j x_j / \sum_{j \notin S} a_{ij} x_j - b_i^S \leq 0 \quad i = 1, m, x_j \geq 0 \right\}$$

Par conséquent :

$$\forall \mu \geq 0 \quad z(\mu) \geq z(\bar{\mu})$$

soit donc

$$z(\bar{\mu}) = z(\mu^0) \text{ ou encore } \bar{\mu} = k \mu^0 \quad k > 0$$

Conclusion pratique :

Pour déterminer la contrainte additionnelle dans le cas où nous ne disposons pas de solution réalisable, nous calculons la contrainte du filtre de Balas, en tenant compte explicitement des contraintes d'égalité.

Les coûts réduits des variables hors base nous donnent les coefficients de cette contrainte.

## ANNEXE B - RAPPELS DE QUELQUES DÉFINITIONS EN THÉORIE DES GRAPHES

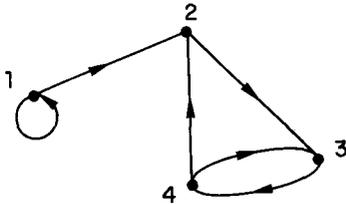
Nous rappelons dans cette annexe quelques définitions simples de la théorie des graphes. Pour une étude plus générale voir ROY [19].

Définition 1 - La matrice caractéristique  $A$  d'un graphe  $G = (X, U)$  est carrée d'ordre  $n = |X|$ , indicée en lignes et en colonnes par les sommets et telle que

$$a_{ij} = 1 \text{ si } (i, j) \in U$$

$$= 0 \text{ si non}$$

Exemple :



Définition 2 - La fonction successeur  $\Gamma$  est une application multivoque de  $X$  dans  $\mathcal{P}(X)$  telle que

$$\Gamma : X \rightarrow \mathcal{P}(X)$$

$$x \rightarrow \{ y \in X / (x, y) \in U \}$$

Définition 3 - La fermeture transitive  $\hat{\Gamma}$  de  $\Gamma$  est une application multivoque de  $X$  dans  $\mathcal{P}(X)$  telle que

$$\hat{\Gamma} : X \rightarrow \mathcal{P}(X)$$

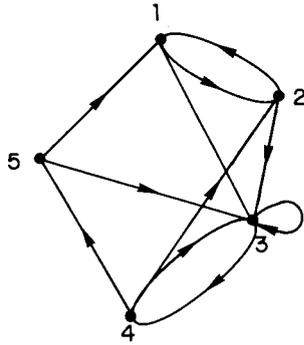
$$x \rightarrow \hat{\Gamma}(x) = \Gamma(x) \cup \Gamma^2(x) \cup \dots \cup \Gamma^p(x)$$

avec  $\Gamma^2(x) = \Gamma(\Gamma(x))$  et  $\Gamma^p(x) = \Gamma(\Gamma^{p-1}(x))$

$\hat{\Gamma}(x)$  est appelée la fermeture transitive du sommet  $x$ .

Définition 3 - Un graphe  $G = (X, U)$  est dit fortement connexe si pour tout sommet  $x$  et  $y$  de  $X$ , il existe au moins un chemin de  $x$  à  $y$ .

Exemple :



Définition 4 - On appelle composante fortement connexe d'un graphe  $G = (X, U)$ , un sous-graphe porté par une  $\mathcal{R}$  d'équivalence de la relation d'équivalence  $\mathcal{R}$  définie par :

$$\forall x \in X \quad \forall y \in X \quad x \mathcal{R} y \leftrightarrow \exists \text{ un chemin de } x \text{ à } y \text{ et un chemin de } y \text{ à } x.$$

Remarque : Soit  $M = (m_{ij})$  la matrice de la fermeture transitive de  $G$ . Les sous-matrices de  $M$  correspondant aux composantes fortement connexes sont des matrices pleines formées de 1. Deux sommets  $i$  et  $j$  seront donc dans la même composante fortement connexe si on a  $\min(m_{ij}, m_{ji}) = 1$ .

## ANNECE C - DÉTERMINATION D'UN VOISINAGE $\mathcal{U}_1$ DE $\bar{u}$ OPTIMUM DE LA MAP

Un assemblage étant déterminé par son  $K_\infty$ , comme dans DUMAS [5\_7], nous dirons que la configuration  $u$  discrète, la plus proche de  $\bar{u}$  est celle qui réalise

$$\min_{u \in \mathcal{U}} \max_i | K_\infty^i(u) - K_\infty^i(\bar{u}) | \quad (1)$$

où  $K_\infty^i(u)$  désigne le  $K_\infty$  de l'élément numéro  $i$  dans la configuration  $u$ .

Classons pour une configuration donnée, les  $K_\infty$  par ordre décroissant.

Lemme : La permutation qui réalise (1) est la permutation identité.

Preuve : Supposons le classement fait.

$$K_\infty^1(u) \geq K_\infty^2(u) \geq \dots \geq K_\infty^n(u)$$

$$K_\infty^1(\bar{u}) \geq K_\infty^2(\bar{u}) \geq \dots \geq K_\infty^n(\bar{u})$$

Posons :

$$B = (b_{ij})_{\substack{j=1,n \\ i=1,n}} = ( | K_\infty^i(u) - K_\infty^j(\bar{u}) | )$$

Une configuration  $u$  étant une permutation d'ordre  $n$ , pour en extraire une de  $B$ , il suffit de choisir un et un seul élément par ligne et par colonne.

Soit  $i_0$  tel que  $b_{i_0 i_0} = \max_i | K_\infty^i(\bar{u}) - K_\infty^i(u) |$

Supposons que l'on ait :  $K_\infty^{i_0}(\bar{u}) > K_\infty^{i_0}(u)$ .

Alors pour :  $i \leq i_0 \rightarrow K_\infty^i(u) \geq K_\infty^{i_0}(u)$

$j \geq i_0 \rightarrow K_\infty^{i_0}(u) \geq K_\infty^j(u)$

Par conséquent, pour  $i \leq i_0$  et  $j \geq i_0$  nous avons :

$$a_{ij} = K_\infty^i(\bar{u}) - K_\infty^j(u) \geq K_\infty^{i_0}(\bar{u}) - K_\infty^j(u) \geq K_\infty^{i_0}(u) - K_\infty^j(u) > b_{i_0 i_0} \quad (2)$$

Cherchons maintenant à construire à partir de B une matrice de permutation telle que

$$\max_i | K_{\infty}^i(u) - K_{\infty}^j(\bar{u}) | \leq \max_i K_{\infty}^i(\bar{u}) - K_{\infty}^i(e) = b_{i_0 i_0} \quad \text{où } e \text{ est la permutation}$$

identité.

D'après (2), il nous faut choisir dans la ligne  $i_0$ , un élément  $b_{i_0 j}$  avec  $j < i_0$ . La colonne  $j$  étant saturée, il ne nous reste donc plus que  $i_0 - 2$  éléments d'indice inférieur à  $i_0$  dans les  $i_0 - 1$  lignes de B. Les éléments  $b_{ij}$  à choisir seront donc tels que  $i < i_0$  et  $j > i_0$ . Dans ces conditions  $b_{ij} > b_{i_0 i_0}$  pour la permutation  $u$  ainsi formée. On ne peut donc trouver mieux que la permutation identité.

Nous aurions obtenu le même résultat avec  $K_{\infty}^{i_0}(\bar{u}) < K_{\infty}^{i_0}(u)$ .

La matrice  $D = (d_{ij})$  des cases permises sera alors telle que :

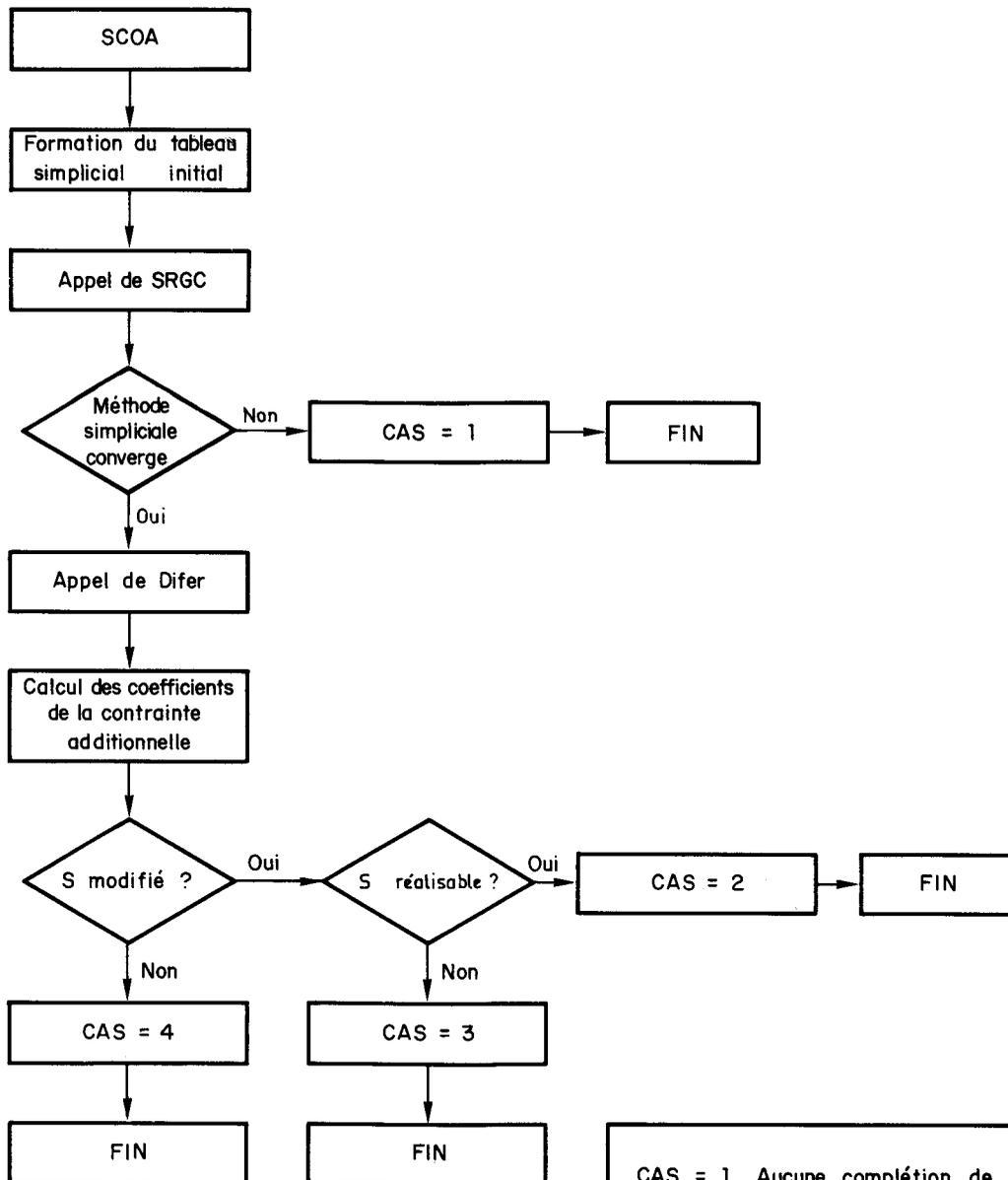
$$d_{ij} = 1 \quad \text{si } b_{ij} \leq b_{i_0 i_0}$$

$$d_{ij} = 0 \quad \text{si non.}$$

## ANNEXE D - QUELQUES ORGANIGRAMMES

- 1 - LILI : Lecture des sections efficaces des assemblages
- 2 - BOOL : Calcul des développements au premier ordre de  $K_{eff}$  et de la distribution de puissance
- 3 - TINF : Calcul du  $K_{\infty}$  de chaque assemblage
- 4 - MGCB : Optimisation : soit du  $K_{eff}$  soit du facteur de forme
- 5 - LISA : Détermination du nombre de variables du problème linéarisé
- 6 - CATY : Fabrication du tableau des contraintes et de la fonction économique pour l'optimisation en variables discrètes
- 7 - CRYS : Fabrication du tableau des contraintes et de la fonction économique pour l'optimisation en variables continues
- 8 - GEOFR : Résolution d'un programme mathématique à variables bivalentes, par l'algorithme de GEOFFRION
- 9 - SRGC : Résolution d'un programme linéaire à variables continues par la méthode duale simpliciale
- 10 - SCOA : Calcul de la contrainte additionnelle
- 11 - DIFER : Arrondi de la solution optimale du programme linéaire
- 12 - SOFI : Calcul des combinaisons linéaires des sections efficaces
- 13 - ADOC : Approximation discrète de l'optimum de la MAP
- 14 - ARTUR : Calcul de  $S^*$  et des  $\frac{\partial}{\partial u_{ij}}$

## CALCUL DE LA CONTRAINTE ADDITIONNELLE



CAS = 1 Aucune complétion de S  
ne peut améliorer l'objectif.

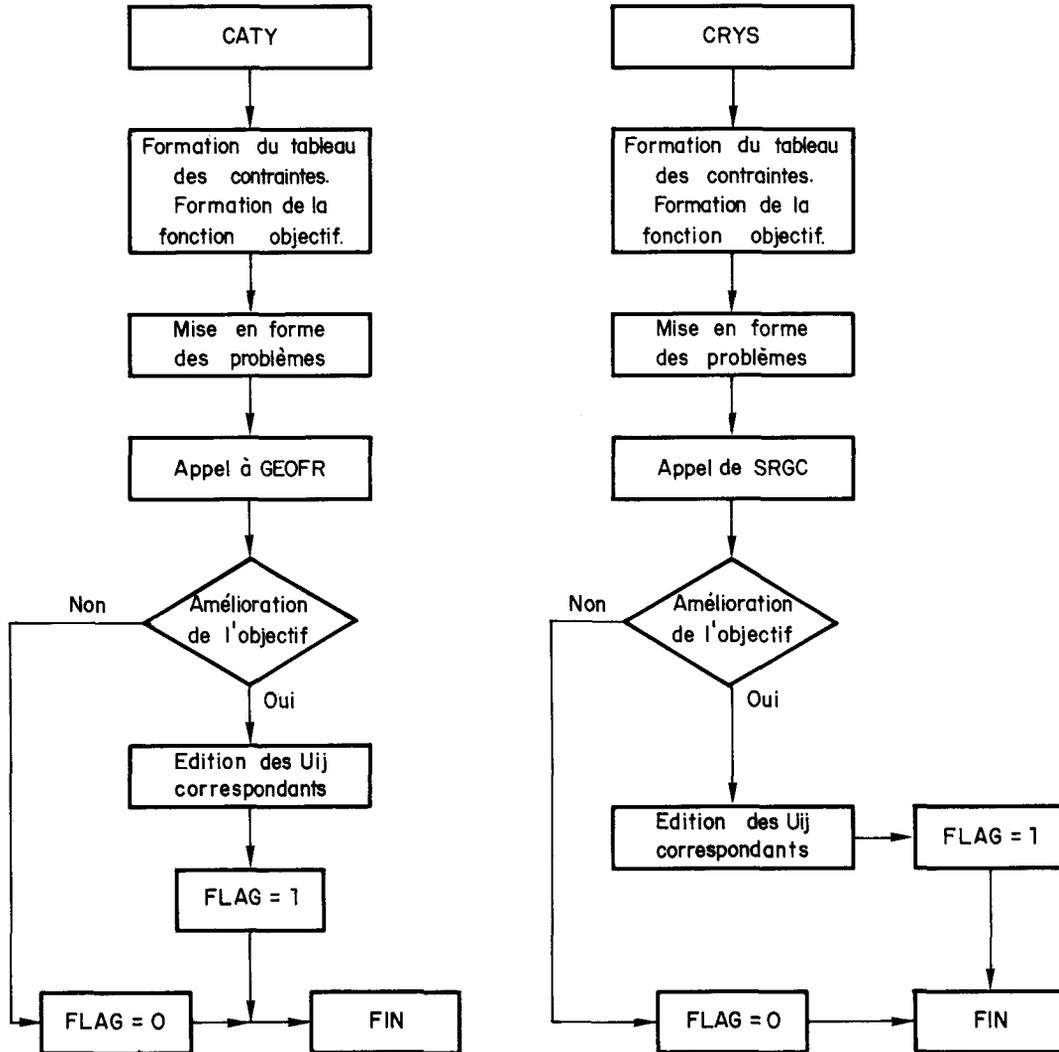
CAS = 2 Solution réalisable trouvée.

CAS = 3 Solution partielle modifiée.

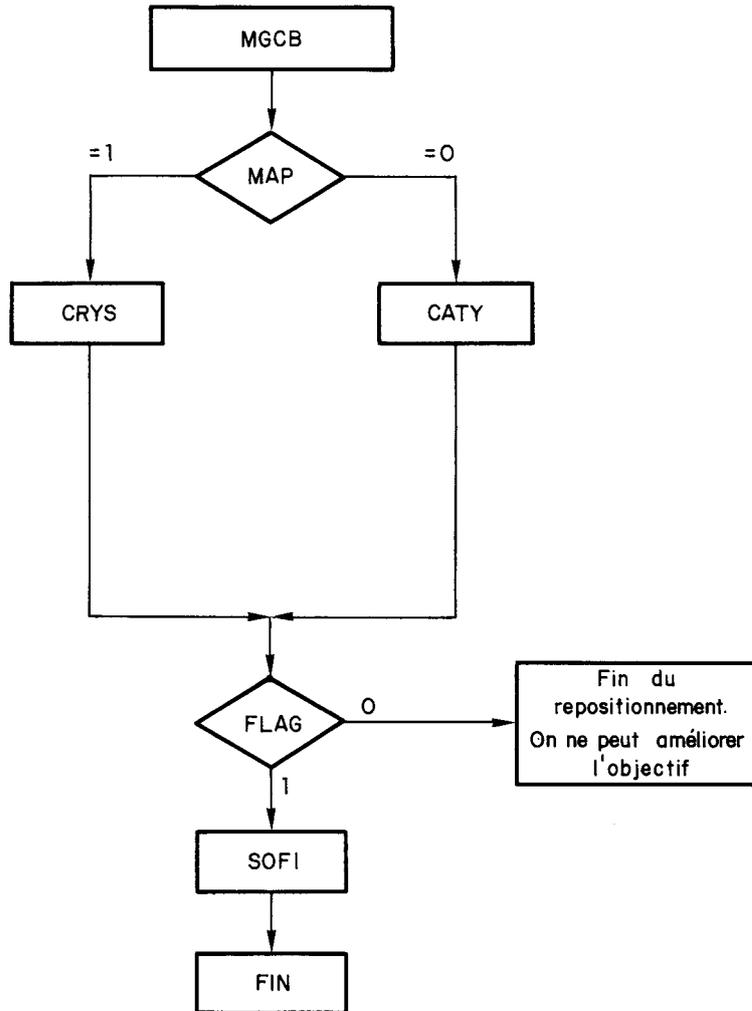
CAS = 4 Solution partielle non modifiée.



SCHEMA DE PRINCIPE DE RESOLUTION DES PROGRAMMES LINEAIRES  
A VARIABLES BOOLEENNES OU A VARIABLES CONTINUES .



## OPTIMISATION SUR LE DOMAINE LINEARISE



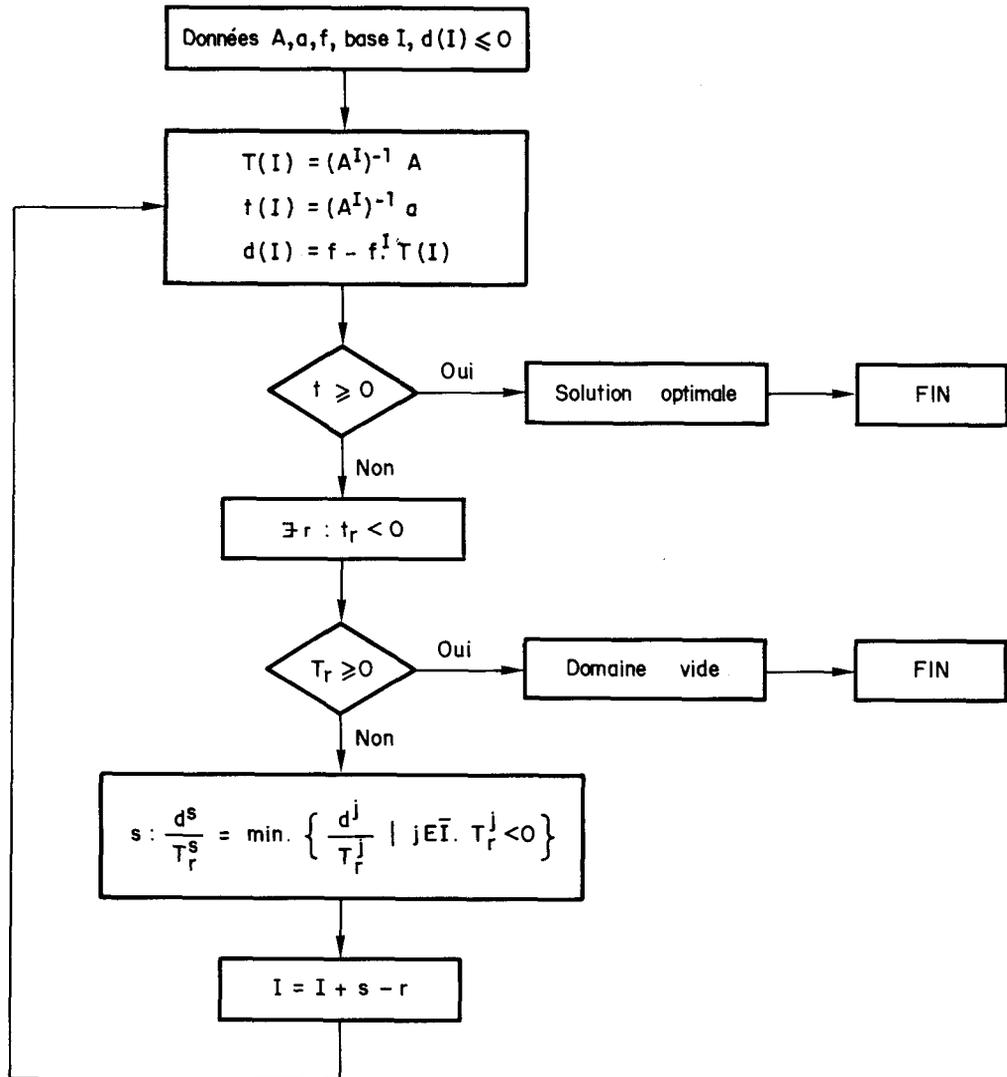
MAP = 1 si on résoud le programme linéaire continu.

= 0 si non.

FLAG = 0 si on ne peut améliorer l'objectif linéarisé.

= 1 si non.



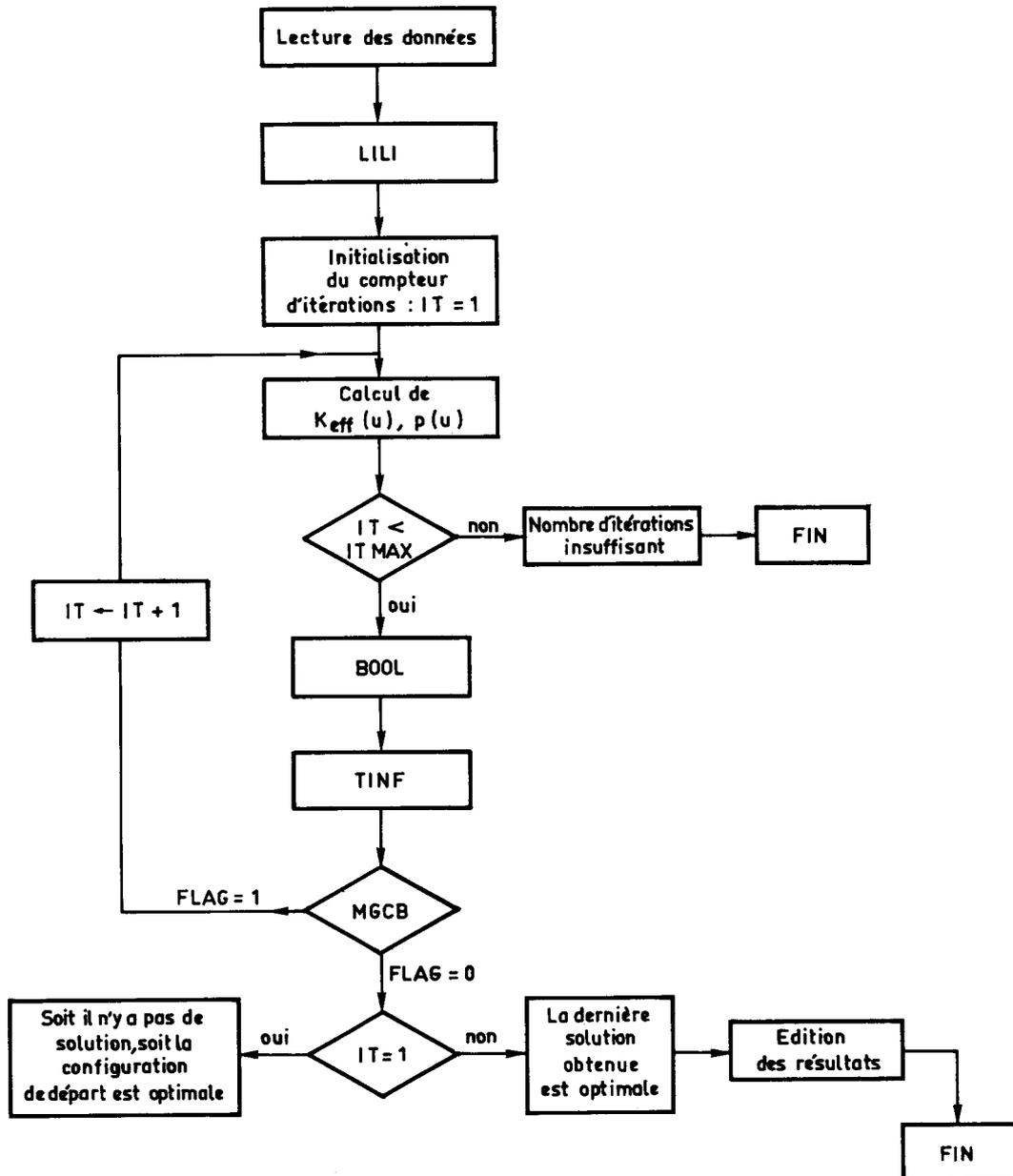


Organigramme de la méthode duale - simpliciale.

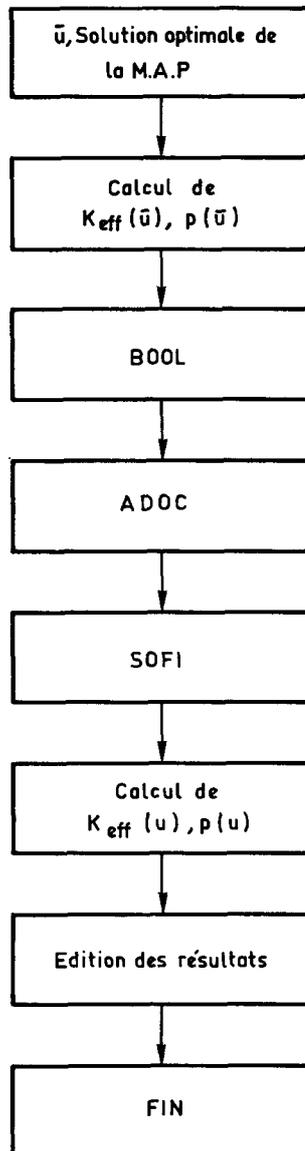
DONNEES INITIALES -

- Configuration et géométrie du coeur
- Nombre d'assemblages à déplacer
- Pour chaque assemblage, son taux d'irradiation (burn-up)
- Nombre d'assemblages où la puissance est contrôlée
- Numéro et nombre d'assemblage dont la puissance est fixée
- $|\Delta K_{\infty}|^{\max}$  (ou  $|\Delta \tau|^{\max}$ )
- Nombre maximum d'itérations autorisées (ITMAX)
- TYPE d'optimisation souhaitée : MAP = 1 on applique la M.A.P.  $\overline{[9]}$ , MAP = 0, on procède à une recherche directe
- Erreur relative maximum tolérée pour l'optimum d'un programme linéaire booléen
- Type de problème à résoudre. LTYPE = 1 on résoud C1, LTYPE = 2 on résoud C2.

## ORGANIGRAMME GENERAL (Recherche directe ou M.A.P)



## APPROXIMATION DISCRETE DE L'OPTIMUM CONTINU



## BIBLIOGRAPHIE

- [1] E. BALAS  
"An additive algorithm for solving linear programming with zero-one variables",  
Operations Research 13. 517-546 (1965).
- [2] E. BALAS  
"Discrete Programming by the filter method",  
Operations Research, 15. 915-947 (1967).
- [3] C. BERGE  
Graphes et Hypergraphes - Dunod - 1972.
- [4] BUSSAC et REUSS  
Traité de neutronique. Hermann 1978.
- [5] M. DUMAS  
Thèse de Docteur Ingénieur - Orsay 1081.
- [6] M. DUMAS  
Rapport CEN N° 215 "S".
- [7] A. GEOFFRION  
"Integer programming by Implicit Enumeration Method",  
Siam Review Vol. 9. 178-190 (1967).
- [8] A. GEOFFRION  
"An Improved Implicit Enumeration approach for integer Programming",  
Operations Research 17. 437-454 (1969)
- [9] GRIFFITH et STEWARD  
Method of Approximate Programming, Management Science 7, 379 (1961).
- [10] HALING  
"Operating Strategy for maintaining an optimum power design distribution throughout  
life."  
TID 7682. ANS Topical Meeting Nuclear Performances of Power reactors cores. (1963)
- [11] HUARD  
Analyse Numérique linéaire. Publication de L'université des Sciences et Techniques  
de Lille. 1972.
- [12] A. KAVENOKY, D. FOURQUET, D. ROBEAU  
Ariane. Description du système et manuel de référence. Rapport SERMA n° 145 "DR"

- [ 13 ] / A. KAVENOKY, J.J. LAUTARD, A. MANUEL, D. ROBEAU .  
"NEPTUNE". Conférence OCDE, Paris 26-28 Novembre 1979 .
- [ 14 ] / J.J.LAUTARD  
Rapport à paraître .
- [ 15 ] / F. MIGNOT  
Thèse de Doctorat es Sciences Mathématiques. Paris VI. 1975 .
- [ 16 ] / H. MUELLER MERBACH  
"On round-off errors in linear programming",  
Lectures Notes in Operations Research and Mathematical Systems. Springer Verlag. 1970 .
- [ 17 ] / J. PLANCHARD  
"Résolution des équations de la diffusion en régime stationnaire" ,  
Bulletin de la Direction des Etudes et Recherches ,  
EDF. Série C. n° 2 1969. 117-140
- [ 18 ] / D. ROBEAU  
Thèse de 3ème cycle. 1976 ORSAY .
- [ 19 ] / B. ROY  
Algèbre moderne et théorie des graphes. Tomes 1 et 2. Dunod. 1970 .
- [ 20 ] / M. SIMONNARD  
Programmation linéaire. Tome 1 et 2 Dunod 1972 .
- [ 21 ] / J.P. STEEN  
Théorie des graphes. Publications du laboratoire de calcul de l'Université des Sciences  
et Techniques de Lille. 1969 .
- [ 22 ] / H.A. TAHA  
Integer Programming. Academic Press. 1975.

*Manuscrit reçu le 26 février 1982*

