

PHILIPPE HERVÉ

## Les procédures arborescentes d'optimisation

*Revue française d'informatique et de recherche opérationnelle*,  
tome 2, n° V3 (1968), p. 69-79

<[http://www.numdam.org/item?id=RO\\_1968\\_\\_2\\_3\\_69\\_0](http://www.numdam.org/item?id=RO_1968__2_3_69_0)>

© AFCET, 1968, tous droits réservés.

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

NUMDAM

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

## LES PROCEDURES ARBORESCENTES D'OPTIMISATION (1)

par Philippe HERVÉ (2)

---

*Résumé. — Diverses procédures, telles que les méthodes « branch and bound », SEP, programmation dynamique, etc. ont été proposées dans les dernières années. Ce sont des illustrations particulières d'une méthode générale pour trouver des solutions optimales ou sous-optimales dans des situations dont l'aspect discret ou combinatoire implique l'absence des propriétés usuelles de convexité. Dans ces situations, les algorithmes classiques (où l'on construit une suite convergeant vers la solution cherchée) ne sont plus applicables. Une approche générale à ce problème est proposée dans cet article.*

### 1. INTRODUCTION

Dans beaucoup de problèmes de Recherche Opérationnelle dont on a clairement défini l'ensemble des contraintes, et, en conséquence l'ensemble des solutions admissibles, on est amené à déterminer, en utilisant au mieux les moyens de calcul dont on dispose, une solution qui possède certaines qualités qui la rendent préférable aux autres.

En général, on recherche une solution à laquelle est associée une bonne valeur d'une certaine fonction économique, et qu'on peut néanmoins obtenir en un temps de calcul acceptable. A défaut de connaître une relation précise entre les deux critères antagonistes : valeur de la fonction économique et temps de calcul nécessaire pour l'obtention de cette solution, on est amené, dans la plupart des méthodes dites « heuristiques », à rechercher une solution dont on connaît une borne supérieure, fixée au départ, de la différence entre sa valeur économique et celle d'une solution optimale du problème.

Le problème à résoudre s'énonce ainsi :

« Ayant associé à tout élément  $x$  de l'ensemble  $E$  des solutions une

---

(1) Présenté au Congrès Européen de l'IMS-TIMS-ES-IASPS, Amsterdam, 2-7 septembre 1968.

(2) Société de Gestion SHELL, Paris.

valeur économique scalaire  $f(x)$ , déterminer une solution  $x_0$   $\varepsilon$ -minimale, c'est-à-dire telle que :

$$\forall x \in E : f(x_0) \leq f(x) + \varepsilon$$

$\varepsilon$  étant un seuil de précision positif ou nul donné. »

Les procédures arborescentes d'optimisation s'appliquent à une classe très générale de problèmes dans lesquels l'aspect combinatoire écarte toute possibilité d'utiliser un algorithme séquentiel classique, comme en programmation linéaire ou non linéaire continue par exemple, c'est-à-dire de construire une suite de solutions convergeant vers la solution désirée selon une certaine métrique. Les procédures arborescentes consistent, grosso-modo, à examiner tour à tour des sous-exemples de  $E$  de plus en plus restreints, jusqu'à la mise en évidence d'une solution  $\varepsilon$ -minimale dans l'un d'eux. Cela revient, en d'autres termes, à accumuler pas à pas une information globale mais partielle sur  $E$ , et à s'arrêter quand on juge cette information suffisante.

Nous présentons ici une formalisation de l'ensemble de ces procédures dont certaines sont connues sous des appellations variées : méthodes « branch and bound », procédures SEP, backtracking, programmation dynamique, etc...

Après un exposé axiomatique, nous illustrerons les principaux concepts introduits par des exemples pris parmi des méthodes connues.

## 2. CADRE FORMEL DE LA PROCEDURE

### 2.1. Graphe dans l'ensemble des parties de $E$

On se donne un ensemble  $D$ , dit « espace des solutions », contenant  $E$ , et une application multivoque  $\Gamma$  de l'ensemble  $\mathcal{F}(D)$  des parties de  $D$  dans lui-même, satisfaisant aux conditions suivantes :

- 1)  $\hat{\Gamma}D$ , fermeture transitive <sup>(1)</sup> de  $D$ , est finie ; posons  $\hat{\Gamma}D = \mathcal{D}$ .
- 2) Pour tout élément  $X$  de  $\mathcal{D}$ , on a les conditions :

$$\Gamma X \neq \emptyset \Rightarrow \begin{cases} \forall Y \in \Gamma X, \forall Z \in \Gamma X : Y \neq Z \Rightarrow Y \not\subset Z & (1) \\ \bigcup_{Y \in \Gamma X} Y = X & (2) \end{cases}$$

Appelons  $\mathcal{E}$  la famille :

$$\mathcal{E} = \{ \sigma_X / \sigma_X = X \cap E, \quad X \in \mathcal{D} \}$$

et  $\gamma$  l'application multivoque dans  $\mathcal{E}$  définie par :

$$\gamma \sigma_X = \{ \sigma_Y / Y \in \Gamma X \}$$

---

(1) Rappelons que la fermeture transitive  $\hat{\Gamma}$  d'une application multivoque  $\Gamma$  de  $D$  dans lui-même est :

$$\hat{\Gamma} D = \{ D \} \cup \Gamma D \cup \Gamma^2 D \cup \dots \cup \Gamma^n D \cup \dots$$

Le graphe  $\mathcal{G} = (\mathcal{E}, \gamma)$  est un graphe orienté sans circuits qui admet  $E$  comme sommet source et les sommets  $\sigma_X$  tels que  $\Gamma X = \emptyset$  comme sommets terminaux.

Dans la suite nous appellerons sommets (sous-entendu de  $\mathcal{G}$ ) les éléments de  $\mathcal{E}$ .

REMARQUES : Il peut arriver que certains sommets  $\sigma_X$  soient vides alors que  $X$  ne l'est pas ; de même on peut avoir  $\sigma_X = \sigma_Y$ , alors que que  $X \neq Y$ .

## 2.2. Information en un sommet de $\mathcal{G}$

A toute étape de la procédure, l'information sur l'ensemble des solutions contenues dans un sommet  $\sigma$  se divise en deux types :

- l'information certaine,
- l'information supplémentaire.

C'est la première que nous décrirons dans ce paragraphe, la seconde n'étant pas formalisée et dépendant largement du type de problème traité.

### 2.2.1. Evaluation en un sommet

A chaque étape, à chaque sommet  $\sigma$  est attaché un scalaire  $v(\sigma)$  qui constitue une évaluation par défaut des valeurs économiques des solutions minimales de  $\sigma$ . Au début, quand on ne sait rien sur  $\sigma$ , on pose :

$$v(\sigma) = -\infty$$

Si au contraire, on apprend à une certaine étape que  $\sigma$  est vide, on pose :

$$v(\sigma) = +\infty$$

### 2.2.2. Examen d'un sommet

L'examen d'un sommet est une procédure qui permet d'en améliorer sa connaissance, c'est-à-dire, entre autres, d'augmenter l'évaluation  $v(\sigma)$  de ce sommet. Cet examen doit satisfaire la condition suivante (3) :

L'examen d'un sommet terminal de  $\mathcal{G}$  fournit une solution minimale de ce sommet [et en conséquence, une évaluation  $v(\sigma)$  sans défaut de ce sommet].

### 2.2.3. Etats d'un sommet

Outre l'évaluation  $v(\sigma)$ , l'information certaine en un sommet est constituée par les valeurs des trois états à deux valeurs possibles chacun, définis comme il suit :

Par convention, on écrira  $\sigma(L)$  pour indiquer que le sommet  $\sigma$  est dans l'état  $L$  ; on emploiera de plus, pour définir un état  $L$  les signes de la logique des propositions (1).

(1)  $\bar{A}$  = négation de  $A$  ;  $A \wedge B$  =  $A$  et  $B$  ;  $A \vee B$  =  $A$  ou  $B$ .

a) *État « marqué »* ( $M$ )

Par définition on a  $\sigma(M)$  si le sommet  $\sigma$  a été examiné. Autrement dit, marquage et examen sont synonymes.

La transition :  $\sigma(\bar{M}) \rightarrow \sigma(M)$  est irréversible.

b) *État « suffisamment connu »* ( $S$ )

A une certaine étape, on a  $\sigma(S)$  si,  $x_0$  étant la plus petite solution de  $E$  connue à cette étape, on a :

$$f(x_0) \leq v(\sigma) + \varepsilon$$

La valeur de  $f(x_0)$  ne pouvant aller qu'en décroissant au cours de la procédure, il en résulte que la transition  $\sigma(\bar{S}) \rightarrow \sigma(S)$  est irréversible.

c) *État « fermé »* ( $F$ )

Un sommet  $\sigma$  est fermé si tous les sommets de  $\gamma\sigma$  sont marqués ou suffisamment connus

$$\sigma(F) \Leftrightarrow \forall \tau \in \gamma\sigma : \tau(M \vee S)$$

On voit que la transition :  $\sigma(\bar{F}) \rightarrow \sigma(F)$  est irréversible.

### 2.3. Description et justification de la procédure

La procédure consiste à marquer tour à tour certains sommets de  $\mathfrak{C}$  choisis comme il suit

A. A la première étape, marquer le sommet  $E$ .

B. A une étape quelconque :

1. Choisir un sommet  $\sigma$  tel que  $\sigma(M \wedge \bar{F})$ .
2. Dans l'ensemble  $\gamma\sigma$ , choisir un sommet  $\tau$  tel que  $\tau(\bar{M} \wedge \bar{S})$ .
3. Marquer  $\tau$ .
4. Dédire de ce marquage les modifications éventuelles sur l'information en chacun des sommets de  $\mathfrak{C}$ .

C. Arrêter la procédure quand on ne peut plus la continuer.

Montrons que cette procédure conduit bien à la détermination d'une solution  $\varepsilon$ -minimale. Pour cela, énonçons d'abord trois propriétés des états des sommets :

1°

$$\sigma(S) \Leftrightarrow \forall \tau \in \gamma\sigma : \tau(S) \quad (4)$$

Cette propriété est évidente.

2°

$$\sigma(S) \Rightarrow \sigma(F) \quad (5)$$

En effet, d'après (4)

$$\sigma(S) \Rightarrow \forall \tau \in \gamma\sigma : \tau(S) \Rightarrow \forall \tau \in \gamma\sigma : \tau(M \vee S) \Rightarrow \sigma(F)$$

3° Si  $\sigma_x$  est un sommet terminal de  $\mathcal{G}$ , c'est-à-dire, si  $\Gamma X = \emptyset$ , alors

$$\sigma(M) \Rightarrow \sigma \tag{6}$$

Ceci est une conséquence de la condition (3) (cf. § 2.2.2.).

Démontrons maintenant la validité de la procédure :

a) La procédure est finie puisque l'ensemble des sommets de  $\mathcal{G}$  est fini.

b) Quand la procédure est achevée, tout sommet marqué est fermé. Supposons en effet qu'il existe encore un sommet  $\sigma_0(M \wedge \bar{S})$ . Si on ne peut plus continuer la procédure c'est que tous les sommets marqués sont fermés. Donc on a  $\sigma_0(F)$ , or :

$$\sigma_0(F) \Rightarrow \forall \sigma_1 \in \gamma \sigma_0 : \sigma_1(M \vee S)$$

et en vertu de (4) :

$$\sigma_0(\bar{S}) \Rightarrow \exists \sigma_1 \in \gamma \sigma_0 : \sigma_1(\bar{S})$$

En conclusion :

$$\sigma_0(M \wedge \bar{S}) \Rightarrow \exists \sigma_1 \in \gamma \sigma_0 : \sigma_1(M \wedge \bar{S})$$

De proche en proche il existe dans  $\mathcal{G}$  un chemin orienté partant de  $\sigma_0$  et dont tous les sommets sont dans l'état  $M \wedge \bar{S}$ , ce qui est impossible puisque l'extrémité de ce chemin, qui est un sommet terminal de  $\mathcal{G}$  ne peut être  $M \wedge \bar{S}$ , en vertu de la propriété (6).

Donc, quand la procédure est achevée, tous les sommets marqués de  $\mathcal{G}$  sont dans l'état  $S$ , et en particulier le sommet initial  $E$ . Si  $x_0$  est la plus petite solution rencontrée au cours de la procédure, il vient :

$$f(x_0) \leq v(E) + \varepsilon \Rightarrow \forall x \in E : f(x_0) \leq f(x) + \varepsilon$$

et  $x_0$  est bien une solution  $\varepsilon$ -minimale de  $E$ .

Si on n'a rencontré aucune solution au cours de la procédure, c'est que  $E$  est vide.

REMARQUE 1 : Cette procédure est dite arborescente parce que, si à chaque étape on marque l'arc de  $\mathcal{G}$  reliant le sommet  $\sigma(M \wedge \bar{F})$  au sommet  $\tau(M \wedge S)$  choisi, le sous-graphe partiel de  $\mathcal{G}$  ainsi marqué est, à chaque étape, une arborescence de racine  $E$ .

REMARQUE 2 : Si on représente par les sommets d'un cube les différentes

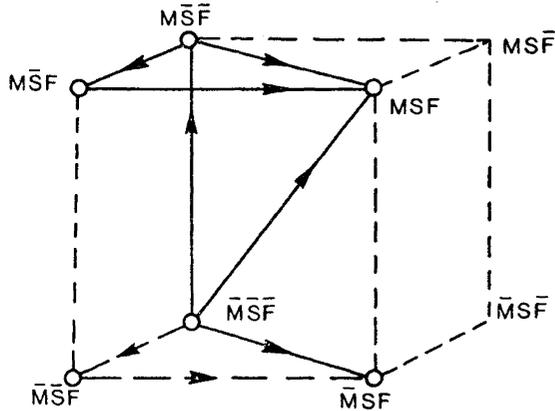


Figure 1

valeurs possibles de l'ensemble des trois états d'un sommet, on peut figurer par des flèches les transitions que peuvent subir ces états au cours de la procédure.

Sur la figure :

— sont marqués d'un point les seuls sommets du cube représentant des états possibles,

— les arcs en tiret représentent les transitions qui ne peuvent exister si le graphe  $\mathcal{G}$  est une arborescence.

### 3. DOMAINE D'APPLICATION ET ADAPTATION DE LA PROCEDURE

Il est difficile de cerner de façon exhaustive l'ensemble des problèmes concernés par cette procédure. Citons cependant :

- les problèmes de recouvrement minimum sur un graphe,
- les recherches d'ensembles intérieurement stables maximaux et extérieurement stables minimaux,
- les problèmes du type « voyageur de commerce », et plus généralement les problèmes de tournées.
- les problèmes d'ordonnement à contraintes disjonctives,
- les programmes linéaires en variables entières ou mixtes,
- les problèmes de programmation dynamique.

Dans le cadre de la procédure décrite plus haut, il existe une grande marge de liberté pour définir avec précision une méthode de résolution, selon le problème à résoudre et les moyens de calcul dont on dispose.

Cette liberté porte sur :

- le choix de l'ensemble  $D$  et de l'application  $\Gamma$ ,
- le mode d'examen des sommets de  $\mathcal{G}$ ,
- le choix, à chaque étape, d'un sommet  $\sigma(M \wedge \bar{F})$
- le choix, à chaque étape, d'un sommet  $\tau(\bar{M} \wedge \bar{S})$

Examinons tour à tour quelques-uns de ces points.

#### 3.1. Ensemble $D$ et application $\Gamma$

Dans la plupart des méthodes,  $D$  est un espace dans lequel on peut aisément repérer les éléments, sans répétition ni omission, à l'aide de paramètres indépendants.

**EXEMPLE :** Dans le problème du voyageur de commerce, on cherche, sur un graphe non orienté à arêtes valuées dans  $\mathbf{R}$  un circuit de longueur minimale dans l'ensemble  $E$  des circuits, passant une fois et une seule par tout sommet. Si le graphe possède  $n$  sommets, on peut plonger  $E$  dans l'espace  $D$  des ensembles de  $n$  arêtes distinctes du graphe.

EXEMPLE : Dans le programme linéaire à variables entières bivalentes :

$$\begin{aligned} & \min \{ Cx / x \in E \} \\ E = \{ x / Ax \geq b, \quad x_j = 0 \text{ ou } 1 \quad \forall j \in J \} \end{aligned}$$

on peut plonger  $E$  dans l'ensemble  $D$  des sommets de l'hypercube unité de dimension  $|J|$ .

$$D = \{ x / x_j = 0 \text{ ou } 1, \quad \forall j \in J \}$$

Dans la plupart des méthodes, un élément  $X$  de  $\mathfrak{D}$  est une partie de  $D$  définie en fixant les valeurs de certains paramètres repérant les éléments de  $D$ , ou bien en restreignant le domaine de variation de certains de ces paramètres.

EXEMPLE : Dans la méthode de Dakin [5] pour résoudre les programmes linéaires à variables mixtes :

$$E = \{ (x, y) / Ax + By = c; \quad x \geq 0; \quad 0 \leq y \leq L; \\ y \text{ a des composantes entières} \}$$

Dans ce cas :

$$D = \{ (x, y) / x \geq 0; \quad 0 \leq y \leq L; \quad y \text{ a des composantes entières} \}$$

Un élément  $X$  de  $\mathfrak{D}$  est défini par la donnée de deux vecteurs  $a$  et  $b$ , à composantes entières et indicées sur le même ensemble que  $y$ , et tels que  $a \leq b$ .

Par définition on a :

$$X(a, b) = \{ (x, y) / x \geq 0; \quad a \leq y \leq b; \quad y \text{ a des composantes entières} \}$$

L'application  $\Gamma X(a, b)$  est une dichotomie de  $X(a, b)$  obtenue de la façon suivante :

— On choisit selon un certain procédé lié au résultat de l'examen de  $X(a, b)$  un indice  $j^*$  d'une composante de  $y$  et deux entiers consécutifs  $\alpha$  et  $\beta$  tels que :

$$a_{j^*} \leq \alpha < \beta \leq b_{j^*}$$

— Soit alors  $a'$  le vecteur obtenu en substituant  $\beta$  à  $a_{j^*}$  dans  $a$  et  $b'$  le vecteur obtenu en substituant  $\alpha$  à  $b_{j^*}$  dans  $b$ .

Par définition :

$$\Gamma X(a, b) = \{ X(a', b), \quad X(a, b') \}$$

### 3.2. Examen des sommets de $\mathfrak{C}$

A chaque sommet  $\sigma$  de  $\mathfrak{C}$  est associé le problème : trouver un élément minimal de  $\sigma$ , au sens de la fonction économique  $f$ . A défaut de résoudre ce problème, on le remplace par un problème moins contraint. Plus

précisément, on plonge  $\sigma$  dans un ensemble  $\sigma'$  plus général dans lequel on sait déterminer directement par un algorithme efficace, un élément minimal dont la valeur va donc constituer une évaluation *par défaut* des solutions de  $\sigma$ .

Parfois, à défaut de savoir déterminer une solution minimale de  $\sigma'$  on sait prouver si  $\sigma'$  est vide ou non.

EXEMPLE : Recherche d'une évaluation par défaut. Dans les méthodes « branch and bound » pour résoudre les programmes linéaires en variables bivalentes, on plonge les ensembles  $\sigma$  qui sont du type :

$$\sigma = \{ x / Ax \leq b ; \quad x_j = 0 \text{ ou } 1 \quad \forall j \in J \}$$

dans l'ensemble plus général :

$$\sigma' = \{ x / Ax \leq b ; \quad 0 \leq x_j \leq 1 \quad \forall j \in J \}$$

La détermination d'un élément minimal de  $\sigma'$  revient alors à résoudre un programme linéaire classique.

EXEMPLE : Critère de vacuité d'un sommet  $\sigma$ . Dans la méthode de Balas [1] pour résoudre les mêmes problèmes que dans l'exemple précédent, une série de tests logiques sur les coefficients du tableau des données fournissent des conditions suffisantes de vacuité de chaque sommet.

### 3.3. Choix, à chaque étape, d'un sommet $\sigma(M \wedge \bar{F})$

Parmi toutes les règles possibles de choix du sommet  $\sigma(M \wedge \bar{F})$  on peut en dégager deux principales :

#### 1) Règle LIFO (Last in first out)

Elle consiste à prendre pour sommet  $\sigma(M \wedge \bar{F})$  le sommet non fermé le plus récemment marqué.

L'application de cette règle revient à marquer des chemins de  $\mathcal{C}$  jusqu'à rencontrer un arrêt, c'est-à-dire jusqu'à ce que le dernier sommet atteint devienne fermé après son marquage. On remonte alors le chemin en sens inverses jusqu'à rencontrer de nouveau un sommet non fermé. On repart de ce dernier sommet pour décrire un nouveau chemin (cette procédure s'appelle aussi « backtracking »).

Cette règle se justifie quand il est particulièrement facile d'enchaîner l'examen d'un sommet  $\sigma$  avec celui de l'un de ses suivants  $\tau \in \gamma\sigma$ .

EXEMPLE : Voir la méthode de Balas [1].

#### 2) Règle du sommet d'évaluation minimale

Elle consiste à choisir systématiquement parmi les sommets  $\sigma(M \wedge \bar{F})$  l'un de ceux qui possède l'évaluation la plus faible.

Quand l'évaluation en chaque sommet est de bonne qualité, on peut en effet penser que la règle ci-dessus est celle qui conduit probablement au plus vite vers la solution cherchée.

Des expériences effectuées par l'auteur sur des programmes linéaires mixtes ont montré cependant que cette idée *a priori* est souvent erronée.

### 3.4. Choix du sommet $\tau(\bar{M} \wedge \bar{S})$ à marquer

Ici on peut distinguer aussi deux façons de procéder. La première façon consiste, une fois choisi le sommet  $\sigma(M \wedge \bar{F})$ , à examiner tour à tour tous les sommets  $\tau \in \gamma\sigma$ . Dans le cas où  $\mathcal{T}$  est une arborescence, cette procédure possède un avantage considérable pour la mise en œuvre sur calculateur, car les seuls sommets marqués non fermés sont alors un sous-ensemble de l'ensemble des points pendants de la sous-arborescence marquée. En conséquence, pour retenir en mémoire cette sous-arborescence, il suffit de conserver simplement la liste de ses sommets pendants.

La deuxième façon de procéder consiste, au contraire, à ne marquer à la fois qu'un seul sommet  $\tau \in \gamma\sigma$  (nous en verrons plus loin un exemple avec la méthode de Land et Doig).

#### *Procédure SEP* [3]

Cette procédure, formalisée par MM. Roy, Bertier et Nghiem, peut se définir ainsi :

C'est l'utilisation simultanée de la règle du sommet d'évaluation minimale et de la première façon de procéder décrite dans ce paragraphe.

## 4. UN EXEMPLE D'APPLICATION DE LA PROCEDURE : LA METHODE DE LAND ET DOIG [8]

Nous avons choisi cet exemple parce qu'il illustre tous les concepts introduits dans ce papier. C'est une méthode pour résoudre les programmes linéaires mixtes.

$$\min \{ ax + by \mid (x, y) \in E \}$$

avec :

$$E = \{ (x, y) \mid 0 \leq x \leq L; \quad y \geq 0; \quad Ax + By = c \quad x_j \text{ entier } \forall j \in J \}$$

1) Ensemble  $D$  :

$$D = \{ (x, y) \mid 0 \leq x \leq L; \quad y \geq 0, \quad x \text{ a des composantes entières} \}$$

2) Ensemble  $\mathfrak{D}$  : Étant donné :

— un sous-ensemble  $K \subset J$ ,

— un vecteur  $\bar{x}_K$  indicé sur  $K$  et à composantes entières, telles que :  $0 \leq \bar{x}_j \leq L_j, \quad \forall j \in K$ .

On leur associe le sous-ensemble de  $D$  :

$$D(\bar{x}_K) = \{ (x, y) \mid (x, y) \in D; \quad x_K = \bar{x}_K \}$$

Il lui correspond, dans  $\mathfrak{E}$ , l'ensemble :

$$E(\bar{x}_K) = D(\bar{x}_K) \cap E$$

3) *Examen d'un sommet*  $E(\bar{x}_K)$

A ce sommet on associe une première évaluation, par défaut  $V(\bar{x}_K)$  obtenue en plongeant  $E(\bar{x}_K)$  dans :

$$E'(\bar{x}_K) = \{ (X, Y) / Y \geq 0 \quad 0 \leq x \leq L \quad Ax + By = c; \quad x_K = \bar{x}_K \}.$$

La détermination d'un élément minimal  $(\xi, \eta)_{\bar{x}_K}$  de  $E(\bar{x}_K)$  s'obtient par simple résolution d'un programme linéaire.

4) *Application*  $\gamma$

En un sommet  $E(\bar{x}_K)$  de  $\mathfrak{G}$  deux cas peuvent se présenter :

— ou bien  $(\xi, \eta)_{\bar{x}_K}$  a toutes les composantes de  $\xi$  entières, dans ce cas  $E(\bar{x}_K)$ , après examen, est dans l'état  $S$  et on pose :

$$\gamma E(\bar{x}_K) = \emptyset$$

— ou bien il existe au moins une composante  $\xi_\alpha$  de  $(\xi, \eta)_{\bar{x}_K}$  qui n'est pas entière, et on pose alors :

$$\gamma E(\bar{x}_K) = \{ E(\bar{x}_K, \bar{x}_\alpha) / \bar{x}_\alpha = 0, 1, 2, \dots, L_\alpha \}$$

*Propriété fondamentale des évaluations*  $V(\bar{x}_K)$

Pour des raisons de convexité évidentes, on voit que :

$$V(\bar{x}_K) \leq V(\bar{x}_K, [\xi_\alpha] + 1) \leq V(\bar{x}_K, [\xi_\alpha] + 2) \leq \dots \leq V(\bar{x}_K, L_\alpha)$$

et

$$V(\bar{x}_K) \leq V(\bar{x}_K, [\xi_\alpha]) \leq V(\bar{x}_K, [\xi_\alpha] - 1) \leq \dots \leq V(\bar{x}_K, 0)$$

De cette propriété, il résulte que  $V(\bar{x}_K, \bar{x}_\alpha)$  constitue une évaluation par défaut des sommets  $E(\bar{x}_K, \bar{x}'_\alpha)$  tels que :

$$\bar{x}'_\alpha \geq \bar{x}_\alpha \quad \text{si} \quad \bar{x}_\alpha > \xi_\alpha$$

et

$$\bar{x}'_\alpha \leq \bar{x}_\alpha \quad \text{si} \quad \bar{x}_\alpha < \xi_\alpha$$

évaluation que l'on peut donc connaître, même si l'on n'a pas marqué le sommet  $E(\bar{x}_K, \bar{x}'_\alpha)$ .

Cette remarque permet de modifier à chaque étape, après un marquage, l'évaluation  $v(\sigma)$  de tout sommet de  $\mathfrak{G}$  à l'aide de la règle suivante :

$$v(\sigma) = \min_{\tau \in \gamma\sigma} v(\tau)$$

---

(1)  $[a]$  signifie : partie entière par défaut du nombre réel  $a$ .

Choix d'un sommet  $\sigma(M \wedge \bar{F})$  à chaque étape

Parmi les sommets d'évaluation minimale on choisit le sommet le plus récemment marqué (ceci constitue une application en cascade de la règle de l'évaluation minimale et de la règle LIFO).

Choix d'un sommet  $\tau(\bar{M} \wedge \bar{S})$  à chaque étape

Soit  $\sigma = E(\bar{x}_K)$  le sommet que l'on vient de choisir ; pour choisir  $\tau$  on procède ainsi :

— si aucun sommet  $\tau \in \gamma\sigma$  n'a encore été marqué, on choisit pour  $\tau$  l'un des deux sommets :

$$E(\bar{x}_K, [\xi_\alpha]) \quad \text{ou} \quad E(\bar{x}_K, [\xi_\alpha] + 1)$$

— si certains éléments de  $\gamma\sigma$  ont déjà été marqué, on choisit pour  $\tau$  celui des éléments de  $\gamma\sigma$  non marqué et non suffisamment connu dont l'examen va permettre d'augmenter l'évaluation de  $\sigma$ .

#### BIBLIOGRAPHIE

Nous citons ici quelques-uns parmi les nombreux articles en relation avec les méthodes arborescentes :

1. E. BALAS, « An additive algorithm for solving linear programs with zero one variables », *Operations Research*, vol. 13, n° 4, July-August 1965, pp. 517-546.
2. M. L. BALINSKI, « Integer Programming : Methods, Uses and computation », *Management Sciences*, vol. 12, 1965, pp. 253-313.
3. P. BERTIER et B. ROY, « Une procédure de résolution pour une classe de problèmes pouvant avoir un caractère combinatoire », *ICC Bulletin*, vol. 4, 1965.
4. P. BERTIER, « Procédures pour élaborer des tournées de distribution » (thèse), *METRA*, série spéciale n° 8, 1966.
5. R. J. DAKIN, « A tree search algorithm for mixed integer programming problems », *Computer Journal*, vol. 8, n° 3, October 1965, pp. 250-255.
6. F. GLOVER, « Truncated Enumeration Methods for solving pure and mixed integer linear programs ». Working paper for limited distribution, operations Research Center, University of California, Berkeley.
7. P. HERVÉ, « Résolution des programmes linéaires à variables mixtes par la procédure SEP », *METRA*, vol. VI, n° 1, 1967, pp. 77-91.
8. A. H. LAND and A. G. DOIG, « An automatic method for solving discrete programming problems », *Econometrica*, vol. 28, 1960, pp. 497-520.
9. E. L. LAWLER and D. E. WOOD, « Branch and Bound Methods, A. Survey », *Operations Research*, vol. 14, n° 4, pp. 699-719.
10. B. ROY, P. BERTIER et P. T. NGHIEM, « Programmes linéaires en nombres entiers et procédure SEP », *METRA*, vol. IV, n° 3, 1965.