

ANASS NAGIH

GÉRARD PLATEAU

## **Problèmes fractionnaires : tour d'horizon sur les applications et méthodes de résolution**

*RAIRO. Recherche opérationnelle*, tome 33, n° 4 (1999),  
p. 383-419

[http://www.numdam.org/item?id=RO\\_1999\\_\\_33\\_4\\_383\\_0](http://www.numdam.org/item?id=RO_1999__33_4_383_0)

© AFCET, 1999, tous droits réservés.

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

NUMDAM

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

## PROBLÈMES FRACTIONNAIRES : TOUR D'HORIZON SUR LES APPLICATIONS ET MÉTHODES DE RÉOLUTION (\*)

par Anass NAGIH <sup>(1)</sup> et Gérard PLATEAU <sup>(1)</sup>

Résumé. – *Les programmes fractionnaires consistent à optimiser un objectif mis sous la forme d'un rapport de deux fonctions linéaires ou non, soumis à un ensemble de contraintes. Différentes versions de ce modèle, linéaires ou non linéaires, en nombres entiers ou en continu, ont une multitude d'applications que ce soit en optimisation combinatoire, en programmation stochastique, en bases de données ou en économie. Trois grandes stratégies de résolution d'un programme fractionnaire émergent dans la littérature : la résolution directe, la résolution par paramétrisation et la résolution d'un problème équivalent à objectif simplifié. Un catalogue de ces modèles et applications mettant en jeu des programmes fractionnaires précède une synthèse des méthodes de résolution incluant en particulier celles dédiées aux programmes hyperboliques (optimisation d'un rapport de deux fonctions affines soumis à des contraintes linéaires) en variables 0-1.*

Mots clés : Programmation fractionnaire, optimisation en nombres entiers, modélisation, applications.

Abstract. – *Fractional programming consists in optimizing a ratio of two functions subject to some constraints. Different versions of this model, linear or nonlinear, have applications in various fields like combinatorial optimization, stochastic programming, data bases, and economy. Three resolution methods are presented: direct solution, parametric approach and solution of an equivalent problem.*

Keywords: Fractional programming, integer programming, modelisation, applications.

### NOTATIONS

$[\lambda]$  : partie entière d'un réel  $\lambda$

$Co(E)$  : enveloppe convexe d'un ensemble  $E$

$|E|$  : cardinal d'un ensemble  $E$ .

Étant donné  $u$  et  $v$  deux vecteurs de  $\mathbb{R}^n$  :

$uv = \sum_{j=1}^n u_j v_j$ , produit scalaire de  $u$  et de  $v$

$\|u\| = \sqrt{\sum_{j=1}^n u_j^2}$ , norme euclidienne du vecteur  $u$

$u_J$  : sous-vecteur de  $u$  composé des éléments  $u_j$ ,  $j \in J \subseteq \{1, \dots, n\}$ .

---

(\*) Reçu en juin 1996.

(<sup>1</sup>) LIPN, UPRES-A 7030 du CNRS, Université Paris 13, Institut Galilée, avenue Jean-Baptiste Clément, 93430 Villetaneuse, France.

Étant donné  $A$  une matrice de format  $m \times n$  :

$a_{ij}^j$  : élément de la ligne  $i$  et la colonne  $j$  de  $A$ ,  $1 \leq i \leq m$  et  $1 \leq j \leq n$

$A^j$  : colonne  $j$  de  $A$ ,  $1 \leq j \leq n$

$A_i$  : ligne  $i$  de  $A$ ,  $1 \leq i \leq m$ .

Étant donné un problème d'optimisation  $(P)$  :

$v(P)$  : valeur optimale de  $(P)$

$F(P)$  : ensemble des solutions réalisables de  $(P)$ .

## 1. INTRODUCTION

Étant donné  $f$ ,  $h$ , et  $g_i, i = 1, \dots, m$ , des fonctions réelles définies sur  $\mathbb{R}^n$ , avec  $h$  ne s'annulant pas sur un sous-ensemble  $X$  de  $\mathbb{R}^n$ , le problème de programmation fractionnaire consiste à déterminer un élément  $x^*$  de  $X$  optimisant la fonction  $f(x)/h(x)$  sur un domaine défini par le système de contraintes  $g(x) \leq 0$  avec  $x$  dans l'ensemble  $X$ . Il a donc la forme suivante :

$$(P) \quad \begin{cases} \text{maximiser} & \frac{f(x)}{h(x)} \\ \text{sous les contraintes} & g(x) \leq 0 \\ & x \in X \end{cases}$$

avec comme hypothèses classiques :

- $F(P) \neq \emptyset$ ,
- les fonctions  $f$ ,  $h$  et  $g$  sont continues sur  $\mathbb{R}^n$ ,
- $\forall x \in X : h(x) > 0$ ,
- $\exists x \in F(P) : f(x) > 0$ .

Lorsque la fonction  $f$  est concave et les fonctions  $h$  et  $g$  sont convexes,  $(P)$  est désigné par programme fractionnaire concave-convexe.  $(P)$  est dit fractionnaire linéaire, ou encore hyperbolique, lorsque  $f$ ,  $h$  et  $g$  sont des fonctions linéaires ou affines de la variable  $x$ . Il se modélise alors comme suit :

$$\begin{cases} \max & \frac{c_0 + cx}{d_0 + dx} \\ \text{s.c.} & Ax \leq b \\ & x \in X \end{cases}$$

où  $c_0$  et  $d_0$  sont des réels,  $c$  et  $d$  sont des vecteurs de  $\mathbb{R}^n$ ,  $A$  est une matrice réelle de format  $m \times n$  et  $b$  est un vecteur de  $\mathbb{R}^m$ .

Les programmes fractionnaires linéaires ou non linéaires, en continu, en nombres entiers ou en variables 0–1 se rencontrent dans plusieurs domaines tels que les bases de données, l'optimisation combinatoire, la programmation stochastique et l'économie. Plusieurs applications des programmes fractionnaires ont déjà été décrites dans la littérature [14, 64, 65, 67]. Les sections 2.3, 2.4 et 2.5 en rappellent quelques-unes en variables continues. Les applications les plus récentes et proches de l'optimisation combinatoire sont détaillées en sections 2.1 et 2.2.

La forme particulière des programmes fractionnaires a fait que de nombreux auteurs ont entrepris d'élaborer des méthodes de résolution spécifiques qui se sont avérées plus efficaces que l'application directe de méthodes générales de programmation non linéaire.

Dans la littérature émergent trois grandes stratégies de résolution d'un programme fractionnaire.

- *La résolution directe* (Sect. 3) : le programme est traité sous sa forme originale. Elle a été utilisée pour les programmes hyperboliques en variables bivalentes (0–1).
- *La résolution par paramétrisation* (Sect. 4) : à l'inverse de la résolution directe, on construit un problème à objectif simplifié, combinaison linéaire du numérateur et du dénominateur par l'intermédiaire d'un paramètre, tout en gardant inchangé l'ensemble des contraintes. Une séquence de résolutions de ce type de problème fournit une solution du programme fractionnaire. Cette méthode a été utilisée pour les différents programmes fractionnaires linéaires et non linéaires, en variables continues comme en variables discrètes, sur des domaines bornés.
- *La résolution d'un programme équivalent* (Sect. 5) : un changement de variables permet lui aussi de simplifier l'objectif, mais en transportant la difficulté sur l'ensemble des contraintes. Par exemple, un programme fractionnaire concave-convexe est transformé en un programme concave, et un programme hyperbolique en un programme à objectif linéaire, avec des contraintes additionnelles.

Pour terminer, la section 6 présente les différentes approches permettant de définir un dual d'un programme fractionnaire.

## 2. APPLICATIONS

Les programmes fractionnaires linéaires ou non linéaires, en continu, en nombres entiers ou en variables 0–1 se rencontrent dans plusieurs

domaines tels que les bases de données, l'optimisation combinatoire, la programmation stochastique et l'économie. Plusieurs applications des programmes fractionnaires ont déjà été décrites dans la littérature [14, 64, 65, 67]. Les sections 2.3, 2.4 et 2.5 en rappellent quelques-unes en variables continues. Les applications les plus récentes et proches de l'optimisation combinatoire sont détaillées en sections 2.1 et 2.2.

## 2.1. Bases de données

L'optimisation de requêtes en recherche documentaire (Hansen *et al.* [30]) est une application informatique qui débouche sur un programme hyperbolique en variables 0-1.

Étant donné un ensemble de documents avec attributs (mots-clés, mots d'un titre, noms d'auteurs, etc.), le problème envisagé a trait à un système automatique de récupération de documents en réponse à une requête. Heine [32] a montré que l'efficacité d'un tel système dépend de la forme de la requête et que l'écriture « optimale » de cette requête par un utilisateur est loin d'être évidente. Le but est donc d'établir une interface qui aide l'utilisateur à trouver une forme satisfaisante de requête. Le critère d'optimalité choisi est celui de Van Rijsbergen [68] qui, grâce à une étude mathématique basée sur la théorie de la mesure, a élaboré un critère d'évaluation de l'efficacité d'un système de recherche documentaire comme suit :

pour une requête donnée, en notant  $Da$  l'ensemble des documents adéquats et  $Dr$  l'ensemble des documents récupérés, l'efficacité du système est fonction de deux paramètres fondamentaux :

- la précision, c'est-à-dire la probabilité pour que le document récupéré soit adéquat :

$$Pa = \frac{|Da \cap Dr|}{|Dr|},$$

- la récupération, c'est-à-dire la probabilité pour que le document adéquat soit récupéré :

$$Pr = \frac{|Da \cap Dr|}{|Da|}.$$

Le critère à minimiser de Van Rijsbergen s'écrit :

$$\begin{aligned} VR(\alpha) &= 1 - \frac{1}{\frac{\alpha}{Pa} + \frac{1-\alpha}{Pr}} \\ &= \frac{\alpha |Dr| + (1-\alpha) |Da| - |Da \cap Dr|}{\alpha |Dr| + (1-\alpha) |Da|} \end{aligned}$$

où  $\alpha \in [0, 1]$  est un paramètre caractérisant la préférence pour la précision ( $\alpha$  proche de un) ou la récupération ( $\alpha$  proche de zéro).

Par la suite on suppose que toute requête est mise sous forme normale disjonctive. En désignant par  $(\text{clé})_j$ ,  $j = 1, \dots, n$ , toutes les conjonctions logiques élémentaires possibles, en notant par  $Nr_j$  le nombre de documents récupérés correspondants et par  $Na_j$  celui des documents adéquats parmi les  $Nr_j$  récupérés, et en définissant enfin les variables bivalentes  $x_j$ ,  $j = 1, \dots, n$ , comme suit :

$$x_j = \begin{cases} 1 & \text{si } (\text{clé})_j \in \text{requête,} \\ 0 & \text{sinon,} \end{cases}$$

on obtient :

$$|Da| = \sum_{j=1}^n Na_j, \quad |Dr| = \sum_{j=1}^n Nr_j x_j, \quad |Da \cap Dr| = \sum_{j=1}^n Na_j x_j.$$

Avec ces notations, le critère à minimiser  $VR(\alpha)$  s'écrit :

$$VR(\alpha) = \frac{\sum_{j=1}^n (1 - \alpha) Na_j + \sum_{j=1}^n (\alpha Nr_j - Na_j) x_j}{\sum_{j=1}^n (1 - \alpha) Na_j + \sum_{j=1}^n \alpha Nr_j x_j}$$

et ainsi le problème d'optimisation de requêtes a la forme générale d'un programme hyperbolique sans contraintes en variables 0-1 :

$$\begin{cases} \min & \frac{c_0 + \sum_{j=1}^n c_j x_j}{d_0 + \sum_{j=1}^n d_j x_j} \\ \text{s.c.} & x_j \in \{0, 1\} \quad j = 1, \dots, n. \end{cases}$$

Une limitation  $\overline{Nr}$  sur le nombre global de documents à récupérer nécessiterait l'introduction d'une contrainte de type sac à dos :

$$\sum_{j=1}^n Nr_j x_j \leq \overline{Nr}.$$

## 2.2. Génération de colonnes

Dans le but d'étendre la programmation linéaire généralisée au cas des programmes mixtes de grandes tailles, Hansen *et al.* [29] ont proposé des procédures arborescentes utilisant l'algorithme dual du Simplexe, et dans lesquelles la génération de colonnes est réalisée par résolution de programmes fractionnaires en nombres entiers avec contraintes.

Soit donc la forme standard d'un programme linéaire mixte à  $m$  contraintes dont le nombre  $n$  de variables est démesurément grand (par exemple la matrice de contraintes ne peut être connue explicitement) :

$$\left\{ \begin{array}{l} \min \quad \sum_{j=1}^n c_j x_j \\ \text{s.c.} \quad \sum_{j=1}^n A^j x_j = b \\ x_j \geq 0, \quad j \in J = \{1, \dots, n\} \\ x_j \text{ entier, } \forall j \in J_E \subset J; J_E \neq \emptyset. \end{array} \right.$$

Dans une méthode classique de Branch-and-Bound, considérons la solution optimale  $\bar{x}$  du programme linéaire associé au problème courant,  $B$  la matrice de base,  $J_B$  l'ensemble des indices des variables en base et  $J_N = J_{\text{sup}} \cup J_{\text{inf}}$  celui des indices des variables hors-base ( $J_{\text{sup}}$  (resp.  $J_{\text{inf}}$ ) est l'ensemble des indices des variables hors-base à leur borne supérieure (resp. inférieure)).

S'il existe  $i \in J_B \cap J_E$  tel que  $\bar{x}_i \notin \mathbb{N}$ , ce problème courant est séparé en deux sous-problèmes par ajout de la contrainte  $x_i \leq \lfloor \bar{x}_i \rfloor$  d'une part, et de la contrainte  $x_i \geq \lfloor \bar{x}_i \rfloor + 1$  d'autre part.

Pour le sous-problème incluant par exemple la contrainte  $x_i \geq \lfloor \bar{x}_i \rfloor + 1$ , la détermination de la variable  $x_k$  entrant dans la base, consiste à déterminer une colonne  $A^k$  telle que

$$\frac{\pi A^k - c_k}{\beta A^k} = \min \left\{ \begin{array}{l} \min \left\{ \frac{\pi A^j - c_j}{\beta A^j} \mid \beta A^j < 0, j \in J_{\text{inf}} \right\} ; \\ \min \left\{ \frac{\pi A^j - c_j}{\beta A^j} \mid \beta A^j > 0, j \in J_{\text{sup}} \right\} \end{array} \right\}$$

où  $\pi$  est le multiplicateur du simplexe (c'est-à-dire  $c_B B^{-1}$ ) et  $\beta$  est la ligne  $i$  de  $B^{-1}$  (itération courante de l'algorithme dual du simplexe).

Par exemple, considérons le problème de découpe unidimensionnelle qui consiste à minimiser la chute. On suppose que les pièces à découper ont une longueur commune  $L$ , que les demandes des clients correspondent à  $b_i$  pièces de longueur  $l_i$ ,  $i = 1, \dots, m$ . Les colonnes de la matrice des contraintes  $A$  représentent les  $n$  possibilités de découpes satisfaisantes. Plus précisément, pour tout  $j \in \{1, \dots, n\}$  et tout  $i \in \{1, \dots, m\}$ ,  $a_i^j$  représente le nombre de découpes de longueur  $l_i$ . La variable de décision  $x_j$  représente le nombre d'utilisations du type de découpe représenté par  $a_i^j$ , et tous les coûts  $c_j$  valent

1,  $j = 1, \dots, n$ . Pour ce problème de découpe, il faut donc résoudre deux programmes hyperboliques en variables entières dont le premier est du type :

$$\left\{ \begin{array}{l} \min \quad \frac{\sum_{i=1}^m \pi_i y_i - 1}{\sum_{i=1}^m \beta_i y_i} \\ \text{s.c.} \quad \sum_{i=1}^m l_i y_i \leq L \\ \sum_{i=1}^m \beta_i y_i < 0 \\ y \in \mathbb{N}^m, y \notin \{A^j \mid j \in J_{\text{sup}}\} \text{ (solutions interdites)} \end{array} \right.$$

et dont la solution optimale  $y^*$  permet de spécifier la colonne  $A^k$ .

### 2.3. Paramétrisation d'un programme linéaire

Considérons un programme linéaire mis sous forme standard :

$$(P) \quad \left\{ \begin{array}{l} \min \quad \sum_{j=1}^n c_j x_j \\ \text{s.c.} \quad \sum_{j=1}^n A^j x_j = b \\ x_j \geq 0, \quad j = 1, \dots, n. \end{array} \right.$$

Lorsqu'un seul coefficient  $c_p$  de l'objectif,  $p \in \{1, \dots, n\}$ , varie (les autres étant tous constants) la valeur optimale de  $(P)$  est une fonction de  $c_p$  concave linéaire par morceaux. Étudier l'effet d'une variation du coefficient  $c_p$  sur la valeur optimale  $v(P)$  de  $(P)$  peut se ramener à un programme fractionnaire [14, 71].

Par exemple, en supposant que  $v(P)$  est finie, une forme d'analyse de sensibilité peut être la suivante :

quelle est la valeur maximale de  $c_p$  telle que la valeur associée du problème ne dépasse pas  $\theta v(P)$ , avec  $\theta \in [0, 1]$  donné ?



En posant  $y = c_p x_p$ , en supposant la valeur optimale de  $x_p$  non nulle (*i.e.* positive), la question se formule comme suit :

quelle est la valeur du programme hyperbolique en variables continues suivant ?

$$\left\{ \begin{array}{l} \max \quad \frac{y}{x_p} \\ \text{s.c.} \quad \sum_{j=1}^n A^j x_j = b \\ \quad \quad \sum_{j \neq p} c_j x_j + y \leq \theta v(P) \\ \quad \quad y \geq 0, x_j \geq 0, \quad j = 1, \dots, n. \end{array} \right.$$

## 2.4. Programmation stochastique

Considérons le programme linéaire :

$$\left\{ \begin{array}{ll} \max & cx \\ \text{s.c.} & Ax = b \\ & x \geq 0 \end{array} \right.$$

et supposons que les composantes de l'objectif  $c$  sont non constantes, indépendantes et varient suivant une loi de probabilité donnée.

La programmation stochastique se propose de maximiser la probabilité pour que la valeur de l'objectif soit supérieure à une valeur donnée  $k$ , c'est-à-dire :

$$(PS_k) \quad \max_{\omega \in \Omega} \mathcal{P}\{w : (\exists x \geq 0), Ax = b, c(\omega)x \geq k\}$$

où la notation  $c(\omega)$  signifie que  $c$  est une variable aléatoire définie sur un espace de probabilité de mesure  $\mathcal{P}$ .

Lorsque les composantes  $c_j$  considérées comme des variables aléatoires, suivent une loi de Gauss ayant  $m$  comme moyenne et  $V$  comme covariance, Bereanu [7] a montré que le programme  $(PS_k)$  est équivalent au programme déterministe suivant :

$$\left\{ \begin{array}{ll} \max & \frac{m^T x - k}{(x^T V x)^{\frac{1}{2}}} \\ \text{s.c.} & Ax = b \\ & x \geq 0 \end{array} \right.$$

qui est un programme fractionnaire linéaire-convexe en variables continues.

## 2.5. Économie

Le domaine de l'économie offre un large éventail d'applications. En effet, la mesure de l'efficacité des systèmes étudiés s'exprime sous forme de rapports entre les critères techniques et/ou économiques. Par exemple :

- rendement/risque : Ziemba [75], Kallberg-Ziemba [39] (modèle concave);
- rendement/investissement : Heinen [33], Mjelde [49, 50] (modèle concave quadratique avec contraintes linéaires);
- coût/temps : Derman [15], Klein [40] (modèle linéaire);
- productivité : Gutenberg [27] (modèle linéaire).

## 3. RÉOLUTION DIRECTE

Cette section traite des méthodes qui résolvent le programme fractionnaire sous sa forme originale, c'est-à-dire sans modifier ni l'objectif ni l'ensemble des contraintes. Cette approche est utilisée pour résoudre les programmes hyperboliques en variables 0-1.

Le premier paragraphe est consacré au programme hyperbolique sans contraintes, le cas avec contraintes fait l'objet du second paragraphe.

### 3.1. Programme hyperbolique en variables 0-1 sans contraintes

À notre connaissance, les premiers auteurs à avoir étudié ce problème sont Hammer et Rudeanu dans leur livre consacré aux méthodes booléennes en recherche opérationnelle [28]. Sous sa forme générale ce problème est NP-Complet (voir par exemple [30]), mais lorsque toutes les données sont supposées positives, des algorithmes exacts de complexité quadratique voire linéaire ont été proposés par Hammer et Rudeanu [28], Robillard [61] et Hansen *et al.* [30]. Ils sont présentés ci-dessous.

On considère donc le programme hyperbolique sans contraintes :

$$(P) \quad \begin{cases} \max & \frac{c_0 + \sum_{j=1}^n c_j x_j}{d_0 + \sum_{j=1}^n d_j x_j} \\ \text{s.c.} & x_j \in \{0, 1\} \quad j = 1, \dots, n \end{cases}$$

dans lequel on suppose que :

- $d_0 + \sum_{j=1}^n d_j x_j > 0$ ,  $\forall x_j \in \{0, 1\} \quad j = 1, \dots, n$ ,
- $c_j > 0$  et  $d_j > 0$ ,  $j = 1, \dots, n$ .

Notons que la deuxième hypothèse n'est pas restrictive puisqu'elle peut être engendrée par les transformations suivantes effectuées sous la première hypothèse (Hansen *et al.* [30]) :

- (i) on obtient  $d_j \geq 0, j = 1, \dots, n$  en remplaçant  $x_j$  par son complémentaire  $1 - x_j$ , pour chaque  $j$  tel que  $d_j < 0$  (il est à noter que la première hypothèse implique que  $d_0 > 0$ ),
- (ii) la deuxième hypothèse est alors obtenue en deux étapes :  
 étape 1 : fixer  $x_j^*$  à 0 pour tout  $j$  tel que  $c_j \leq 0$ , ( $x^*$  étant la solution optimale de (P))  
 étape 2 : fixer  $x_j^*$  à 1 pour tout  $j$  tel que  $d_j = 0$ .

Tous les algorithmes décrits s'appuient sur les deux propriétés fondamentales suivantes :

PROPRIÉTÉ 3.1

$$\min \left\{ \frac{c_j}{d_j}, \frac{c_k}{d_k} \right\} \leq \frac{c_j + c_k}{d_j + d_k} \leq \max \left\{ \frac{c_j}{d_j}, \frac{c_k}{d_k} \right\} \quad \forall j, k \in \{1, \dots, n\}.$$

PROPRIÉTÉ 3.2 (élimination de variables)

$$x_j^* = 0 \text{ pour tout } j \in J = \{1, \dots, n\} \text{ tel que } \frac{c_j}{d_j} \leq \frac{c_0}{d_0}.$$

*Preuve :* Soit  $J_0 = \left\{ j \in J \mid \frac{c_j}{d_j} \leq \frac{c_0}{d_0} \right\}$ , le résultat se déduit directement de la propriété 3.1 et de la relation

$$\frac{c_0 + \sum_{j \in J} c_j x_j}{d_0 + \sum_{j \in J} d_j x_j} \leq \frac{c_0 + \sum_{j \in J-J_0} c_j x_j}{d_0 + \sum_{j \in J-J_0} d_j x_j}, \quad \forall x \in \{0, 1\}^n. \quad \square$$

*Remarque 3.1 :* Cette élimination de variables est un prétraitement commun à tous les algorithmes. Il est donc dorénavant supposé que chacun d'eux résout un programme dont les données sont telles que  $\frac{c_j}{d_j} > \frac{c_0}{d_0}$  pour tout  $j \in J = \{1, \dots, n\}$ , c'est-à-dire que  $J_0 = \emptyset$ .

*Algorithme de Hammer et Rudeanu [28]*

Les auteurs proposent un algorithme (Fig. 1) de complexité temporelle quadratique basé sur la propriété suivante :

PROPRIÉTÉ 3.3 [28]

$$x_k^* = 1 \text{ pour tout } k \in \{1, \dots, n\} \text{ tel que } \frac{c_k}{d_k} = \max \left\{ \frac{c_j}{d_j} \mid j \in \{1, \dots, n\} \right\}.$$

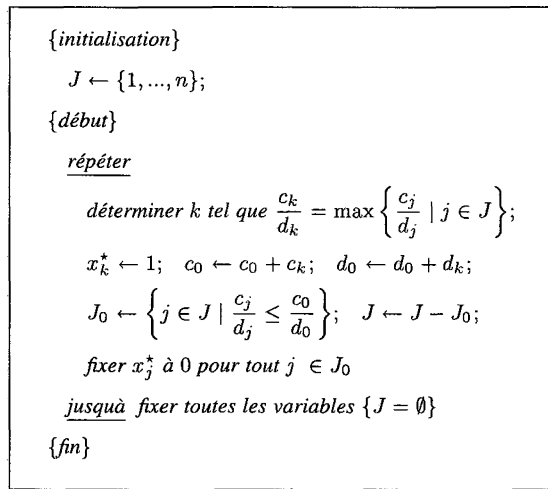


Figure 1. - Algorithme de Hammer et Rudeanu.

*Preuve :* Le résultat se déduit directement de la relation

$$\frac{c_0 + \sum_{j \in J - \{k\}} c_j x_j}{d_0 + \sum_{j \in J - \{k\}} d_j x_j} \leq \frac{c_0 + c_k + \sum_{j \in J - \{k\}} c_j x_j}{d_0 + d_k + \sum_{j \in J - \{k\}} d_j x_j},$$

$$\forall x_j \in \{0, 1\}, j \in J - \{k\}.$$

□

*Algorithme de Robillard [61]*

Cet algorithme (Fig. 2) est également de complexité quadratique. Il repose sur la caractérisation suivante d'une solution optimale.

PROPRIÉTÉ 3.4 [61] : En supposant que  $\frac{c_0}{d_0} < \frac{c_1}{d_1} \leq \frac{c_2}{d_2} \leq \dots \leq \frac{c_n}{d_n}$ , soit  $k$  un entier dans  $\{1, \dots, n\}$  tel que :

$$\frac{c_0 + \sum_{j=k}^n c_j}{d_0 + \sum_{j=k}^n d_j} < \frac{c_i}{d_i} \quad \text{pour } i \geq k$$

et

$$\frac{c_0 + \sum_{j=k}^n c_j}{d_0 + \sum_{j=k}^n d_j} \geq \frac{c_i}{d_i} \quad \text{pour } i < k$$

alors  $x^*$  définie par :

$$\begin{aligned} x_j^* &= 1 & \text{si } j \geq k \\ x_j^* &= 0 & \text{si } j < k \end{aligned}$$

```

{initialisation}
   $J \leftarrow \{1, \dots, n\}; \quad J_0 \leftarrow \emptyset; \quad J_1 \leftarrow \emptyset;$ 
{début}
  pour  $j$  de 1 à  $n$  faire
    si  $\frac{c_j}{d_j} > \frac{c_0}{d_0}$  alors
       $J_1 \leftarrow J_1 \cup \{j\}; \quad c_0 \leftarrow c_0 + c_j; \quad d_0 \leftarrow d_0 + d_j;$ 
    répéter
       $J_0 \leftarrow \left\{ i \in J_1 \mid \frac{c_i}{d_i} \leq \frac{c_0}{d_0} \right\}; \quad J_1 \leftarrow J_1 - J_0;$ 
       $c_0 \leftarrow c_0 - \sum_{i \in J_0} c_i; \quad d_0 \leftarrow d_0 - \sum_{i \in J_0} d_i;$ 
    jusqu'à  $J_0 = \emptyset$ 
  finsi
finpour
  pour  $j$  de 1 à  $n$  faire
    si  $j \in J_1$  alors  $x_j^* \leftarrow 1$ 
    sinon  $x_j^* \leftarrow 0$ 
  finsi
finpour
{fin}

```

Figure 2. – Algorithme de Robillard.

est une solution optimale de (P) dont la valeur optimale est donnée par

$$\frac{c_0 + \sum_{j=k}^n c_j}{d_0 + \sum_{j=k}^n d_j}.$$

*Remarque 3.2 :* La solution optimale n'étant pas unique, l'algorithme fournit celle ayant le plus de composantes nulles.

*Algorithme de Hansen-Poggi-Ribeiro* [30]

Les relations de la propriété 3.4 liant la valeur et la solution optimales d'un programme hyperbolique sans contraintes peuvent être réécrites sous

la forme suivante :

PROPRIÉTÉ 3.5 : Si on note  $\lambda^*$  la valeur optimale de (P), alors toute solution optimale  $x^*$  de (P) est telle que pour tout  $j$  dans  $\{1, \dots, n\}$  :

$$\begin{aligned} x_j^* &= 1 && \text{si } \frac{c_j}{d_j} > \lambda^* \\ x_j^* &= 0 && \text{si } \frac{c_j}{d_j} < \lambda^* \\ x_j^* &= 0 \text{ ou } 1 && \text{si } \frac{c_j}{d_j} = \lambda^*. \end{aligned}$$

En se basant sur cette caractérisation et en procédant par recherche dichotomique, Hansen *et al.* [30] proposent un algorithme (Fig. 3) de complexité linéaire qui fixe à chaque itération au moins la moitié des variables non encore fixées.

Remarque 3.3 : La recherche de l'élément médian dans l'algorithme de la figure 3 d'une liste peut se faire par l'algorithme de Blum *et al.* [9] qui est de complexité linéaire.

### 3.2. Problème hyperbolique en variables 0-1 avec contraintes

Une des premières méthodes de résolution [22] concerne le programme hyperbolique avec contraintes suivant

$$(P) \quad \begin{cases} \max & \frac{c_0 + cx}{d_0 + dx} \\ \text{s.c.} & Ax \leq b \\ & x \in \{0, 1\}^n \end{cases}$$

pour lequel est développée une méthode de type énumération implicite utilisant les travaux de Balas [5], Geoffrion [18] et Florian et Robillard [17]. Il est à noter que cette dernière méthode exploite la notion de contraintes agrégées (« surrogate ») due à Glover [20]. La contrainte agrégée est obtenue en combinant linéairement le système de contraintes  $b - Ax \geq 0$  avec  $(c_0 + cx) - \underline{v}(d_0 + dx) > 0$ , où  $\underline{v}$  est un minorant de la valeur de (P).

```

{initialisation}
 $\tilde{c} \leftarrow c_0 ; \quad \tilde{d} \leftarrow d_0 ; \quad J \leftarrow \{1, \dots, n\} ;$ 

{début}

 $fin \leftarrow faux ;$ 

répéter

    rechercher l'élément médian  $\frac{c_k}{d_k}$  de la liste  $\left\{ \frac{c_j}{d_j} , j \in J \right\} ;$ 

     $J_0 \leftarrow \left\{ j \in J \mid \frac{c_j}{d_j} < \frac{c_k}{d_k} \right\} ; \quad J_1 \leftarrow \left\{ j \in J \mid \frac{c_j}{d_j} > \frac{c_k}{d_k} \right\} ;$ 

     $J_2 \leftarrow \left\{ j \in J \mid \frac{c_j}{d_j} = \frac{c_k}{d_k} \right\} ;$ 

     $\tilde{c} \leftarrow c_0 + \sum_{j \in J_1 \cup J_2} c_j ; \quad \tilde{d} \leftarrow d_0 + \sum_{j \in J_1 \cup J_2} d_j ; \quad \lambda \leftarrow \frac{\tilde{c}}{\tilde{d}} ;$ 

    si  $\lambda > \frac{c_k}{d_k}$  alors

         $x_j^* \leftarrow 0 \quad \forall j \in J_0 \cup J_2 ; \quad J \leftarrow J_1$ 

    sinon

         $x_j^* \leftarrow 1 \quad \forall j \in J_1 \cup J_2 ; \quad fin \leftarrow vrai ;$ 

        si  $J_0 \neq \emptyset$  alors

            déterminer le plus grand élément  $\frac{c_k}{d_k}$  de la liste  $\left\{ \frac{c_j}{d_j} , j \in J_0 \right\} ;$ 

            si  $\lambda < \frac{c_k}{d_k}$  alors

                 $J \leftarrow J_0 ; \quad c_0 \leftarrow \tilde{c} ; \quad d_0 \leftarrow \tilde{d} ; \quad fin \leftarrow faux$ 

            sinon

                 $x_j^* \leftarrow 0 \quad \forall j \in J_0$ 

            finsi

        finsi

    finsi

    jusqu'à fin

     $\lambda^* \leftarrow \lambda$ 

{fin}

```

Figure 3. – Algorithme de Hansen-Poggi-Ribeiro.

Les autres algorithmes proposés dans la littérature sont liés aux structures particulières des systèmes de contraintes [61, 62]. Nous allons traiter en détails le modèle particulier étudié par Saïpe [62], dont l'unique contrainte

impose un nombre  $N \in \{1, \dots, n\}$  de variables à 1 :

$$\left\{ \begin{array}{l} \max \quad r(x) = \frac{\sum_{j=1}^n c_j x_j}{\sum_{j=1}^n d_j x_j} \\ \text{s.c.} \quad \sum_{j=1}^n x_j = N \\ \quad \quad x_j \in \{0, 1\}, \quad j = 1, \dots, n \end{array} \right.$$

où  $c_0 = d_0 = 0$ ,  $d_j > 0$ ,  $c_j \geq 0 \quad \forall j \in \{1, \dots, n\}$ .

*Interprétation géométrique [62]*

En notant que  $\frac{c_j}{d_j}$  est la pente de la droite joignant les points  $(0, 0)$  et  $(d_j, c_j)$  de l'orthant positif, et que la somme des deux vecteurs de coordonnées  $(d_j, c_j)$  et  $(d_k, c_k)$  donne le vecteur de coordonnées  $(d_j + d_k, c_j + c_k)$  de pente  $\frac{c_j + c_k}{d_j + d_k}$ , le programme peut se formuler comme suit :

Étant donnés  $n$  vecteurs dans  $\mathbb{R}^+ \times \mathbb{R}^+$ , trouver  $N$  vecteurs dont la somme vectorielle soit de pente maximale.

Il faut noter que la solution consistant à retenir les  $N$  vecteurs de plus grandes pentes n'est pas nécessairement optimale comme le montre le contre-exemple suivant :

*Exemple 3.1*

$$\left\{ \begin{array}{l} \max \quad \frac{x_1 + 3x_2 + 2x_3}{2x_1 + 5x_2 + x_3} \\ \text{s.c.} \quad x_1 + x_2 + x_3 = 2 \\ \quad \quad x_1, x_2, x_3 \in \{0, 1\}. \end{array} \right.$$

En considérant la solution réalisable  $(0, 1, 1)$  correspondant aux deux plus grands rapports, la valeur obtenue est  $\frac{5}{6}$ . Alors que la valeur optimale 1 correspond à la solution  $(1, 0, 1)$  qui est obtenue en combinant les deux rapports extrêmes. Une illustration géométrique est donnée par la figure 4 dans laquelle chaque vecteur  $\vec{u}_j$  a pour coordonnées  $(d_j, c_j)$ .

L'algorithme proposé par Saïpe est basé sur la propriété 3.3 qui reste valide pour le type de contraintes traitées. Grâce à ce résultat, le programme



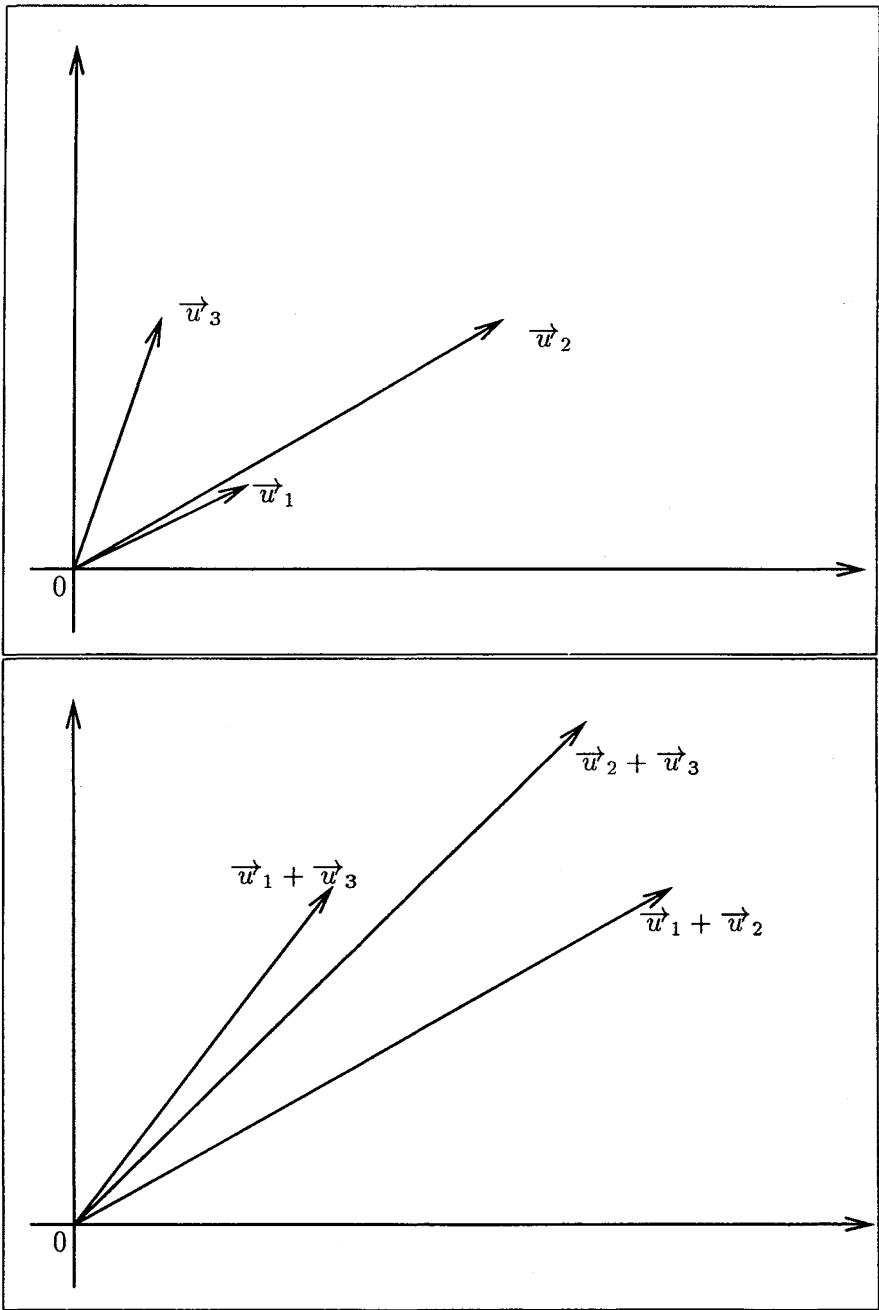


Figure 4. - Illustration géométrique de l'exemple 3.1.

initial se transforme en un programme de même type mais d'ordre  $N-1$  :

$$\begin{cases} \max & r(x_1, \dots, x_{k-1}, 1, x_{k+1}, \dots, x_n) = \frac{c_k + \sum_{j \neq k} c_j x_j}{d_k + \sum_{j \neq k} d_j x_j} \\ \text{s.c.} & \sum_{j \neq k} x_j = N - 1 \\ & x_j \in \{0, 1\}, \quad j \neq k \end{cases}$$

soit encore,

$$\begin{cases} \max & r(x_1, \dots, x_{k-1}, 1, x_{k+1}, \dots, x_n) = \frac{\sum_{j \neq k} \left( c_j + \frac{c_k}{N-1} \right) x_j}{\sum_{j \neq k} \left( d_j + \frac{d_k}{N-1} \right) x_j} \\ \text{s.c.} & \sum_{j \neq k} x_j = N - 1 \\ & x_j \in \{0, 1\}, \quad j \neq k. \end{cases}$$

Il suffit alors d'itérer ce processus  $N-1$  fois sur chaque modèle engendré par la fixation à 1 de la variable satisfaisant la condition de la propriété 3.3. On détermine ainsi les  $N$  composantes de  $x^*$  valant 1, les autres valant toutes 0.

#### 4. RÉOLUTION PAR PARAMÉTRISATION

Afin de simplifier l'objectif du programme mathématique, un paramètre est introduit permettant par exemple de ramener un programme hyperbolique en un programme (paramétré) linéaire, ou bien un programme fractionnaire quadratique en un programme (paramétré) quadratique, tout en gardant l'ensemble des contraintes inchangé. Ainsi le programme obtenu peut être résolu « paramétriquement » : une séquence de résolutions de tels programmes à objectif simplifié engendre une suite de solutions convergeant vers une solution optimale du programme fractionnaire initial.

Proposée initialement en 1956 par Isbell et Marlow [36] pour des programmes hyperboliques, ce n'est qu'à partir de 1967 que cette approche a connu un grand élan avec Dinkelbach [16] qui l'a généralisée aux programmes fractionnaires non linéaires.

Un rappel du programme paramétré et de ses principales propriétés est suivi des différents algorithmes construits autour de cette approche.

Afin d'assurer la convergence de la procédure, l'hypothèse de compacité du domaine défini par les contraintes est adoptée tout au long de cette section.

Soit le programme fractionnaire

$$(P) \quad \begin{cases} \max & \frac{f(x)}{h(x)} \\ \text{s.c.} & g(x) \leq 0 \\ & x \in X. \end{cases}$$

Le programme paramétré associé consiste à simplifier l'objectif en combinant linéairement le numérateur et le dénominateur par l'intermédiaire d'un paramètre  $\lambda \in \mathbb{R}$ . Il est donc défini comme suit

$$(Q(\lambda)) \quad \begin{cases} \max & f(x) - \lambda h(x) \\ \text{s.c.} & g(x) \leq 0 \\ & x \in X \end{cases}$$

pour tout  $\lambda \in \mathbb{R}$ .

Dans le cas d'un programme hyperbolique

$$\begin{cases} \max & \frac{c_0 + cx}{d_0 + dx} \\ \text{s.c.} & Ax \leq b \\ & x \in X \end{cases}$$

le programme paramétré est un programme linéaire dont l'objectif est fonction de  $\lambda \in \mathbb{R}$  :

$$\begin{cases} \max & (c_0 - \lambda d_0) + (c - \lambda d)x \\ \text{s.c.} & Ax \leq b \\ & x \in X. \end{cases}$$

En notant  $\lambda^*$  la valeur optimale de (P), le résultat fondamental liant le programme fractionnaire au programme paramétré associé est donné par :

**PROPOSITION 4.1** (Dinkelbach [16]) : *Toute solution optimale de  $Q(\lambda^*)$  est une solution optimale de (P).*

En tant que fonction de la variable  $\lambda$ , la valeur optimale  $v(\lambda)$  du programme paramétré vérifie quelques propriétés [16] que nous résumons ci-après :

**PROPOSITION 4.2** : *La fonction  $\lambda \mapsto v(\lambda)$  est continue, strictement décroissante, convexe.*

$v(0) > 0$  et  $v(\lambda)$  tend vers  $-\infty$  quand  $\lambda$  tend vers  $+\infty$ . Si de plus  $f$  et  $h$  sont affines, alors  $v$  est linéaire par morceaux.

En particulier, l'équation  $v(\lambda) = 0$  admet une solution unique  $\lambda^*$ , plus précisément :

PROPOSITION 4.3 :

$$(a) \quad v(\lambda) = 0 \iff \lambda = \lambda^*$$

$$(b) \quad v(\lambda) > 0 \iff \lambda < \lambda^*$$

$$(c) \quad v(\lambda) < 0 \iff \lambda > \lambda^*.$$

Ainsi la résolution de (P) revient à trouver la racine de l'équation non linéaire à une seule variable :  $v(\lambda) = 0$ .

Un catalogue des algorithmes de la littérature est présenté ci-dessous. Il inclut des algorithmes de résolution itératifs (Newton, interpolation linéaire) et dichotomiques (recherche dichotomique, recherche dichotomique modifiée, interpolation convexe) ainsi qu'un algorithme  $\varepsilon$ -approchant.

#### *Méthode de Newton* [16, 36]

L'application de la méthode de Newton (Fig. 5) aux programmes fractionnaires linéaires a été d'abord proposée par Isbell et Marlow [36] puis généralisée au cas non linéaire par Dinkelbach [16].

Partant d'un  $\lambda^0$  vérifiant  $v(\lambda^0) > 0$  (c'est-à-dire  $\lambda^0 < \lambda^*$ ), cette méthode connue désormais sous le nom de méthode de Dinkelbach, construit une suite croissante de réels  $(\lambda^p)_{p \in \mathbb{N}}$ , convergeant vers  $\lambda^*$ , chaque élément  $\lambda^p$  de la suite étant égal à  $f(x^{p-1})/h(x^{p-1})$  où  $x^{p-1}$  est une solution optimale de  $(Q(\lambda^{p-1}))$ .

PROPOSITION 4.4 : Soit  $\lambda^l \in \mathbb{R}$  vérifiant  $v(\lambda^l) > 0$  et  $x^l$  une solution optimale de  $Q(\lambda^l)$ , alors on a

$$\lambda^l \leq \frac{f(x^l)}{h(x^l)} \leq \lambda^*.$$

En effet, la droite d'équation  $T(\lambda) = f(x^l) - \lambda h(x^l)$ ,  $\lambda \in \mathbb{R}$ , est une demi-tangente à droite à la courbe représentant  $v$  au point de coordonnées  $(\lambda^l, v(\lambda^l))$  (un sous-gradient est donné par  $-h(x^l)$ ) et coupe l'axe des  $\lambda$  (i.e.  $T(\lambda)$  s'annule) en  $\lambda = f(x^l)/h(x^l)$  (voir Fig. 6). Grâce à la décroissance et la convexité de  $v$  (Prop. 4.2), on a donc  $\lambda^l \leq \frac{f(x^l)}{h(x^l)} \leq \lambda^*$ .

```

{initialisation}
 $\lambda \leftarrow \lambda^l$  ; {avec  $v(\lambda^l) > 0$  i.e.  $\lambda^l < \lambda^*$ }
{début}
  répéter
    déterminer  $x_\lambda$  solution optimale de  $Q(\lambda)$  ;
     $\lambda \leftarrow \frac{f(x_\lambda)}{h(x_\lambda)}$ 
  jusqu'à  $v(\lambda) = 0$ 
{fin}

```

Figure 5. - Algorithme de Newton.

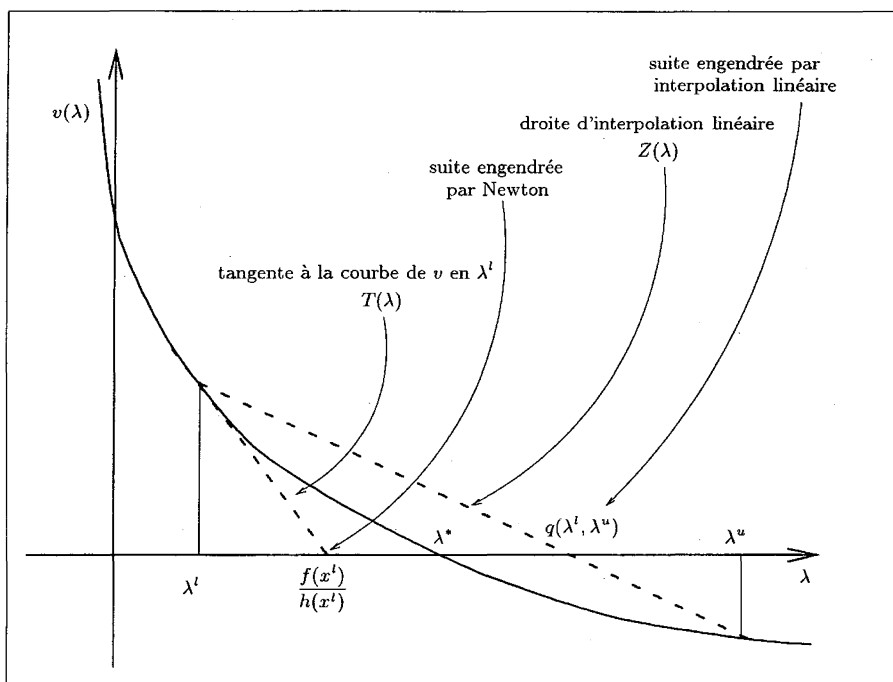


Figure 6. - Suite des solutions engendrées par les algorithmes de Newton (Fig. 5) et d'interpolation linéaire (Fig. 7).

La convergence de cette méthode est superlinéaire, c'est-à-dire  $\frac{\lambda^{p+1} - \lambda^*}{\lambda^p - \lambda^*} \rightarrow 0$ , quand  $p \rightarrow +\infty$ , où  $\lambda^p$  est la valeur de  $\lambda$  à l'itération  $p$  de l'algorithme [34].

Cette méthode est à la base des algorithmes proposés par Florian et Robillard [17] et Radzik [60] pour résoudre le programme hyperbolique en variables 0–1, ceux élaborés par Grunspan et Thomas [25] et Anzai [3] pour le programme hyperbolique en nombres entiers et enfin, par Michelon et Maculan [47] pour la résolution du programme hyperbolique en variables mixtes (entières-continues).

### Méthode d'interpolation linéaire [63]

La méthode classique de Newton génère une suite de réels convergeant vers  $\lambda^*$  par valeurs inférieures. Schaible [63] propose une méthode (Fig. 7) basée sur la notion d'interpolation linéaire qui engendre elle une suite de réels  $(\lambda^p)_{p \in \mathbb{N}}$  convergeant vers  $\lambda^*$  par valeurs supérieures. Ces valeurs sont les abscisses des intersections d'un ensemble de cordes avec l'axe des  $\lambda$ .

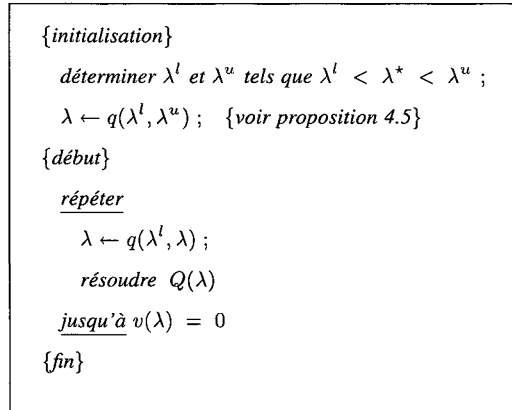


Figure 7. – Algorithme d'interpolation linéaire.

Plus précisément, soit  $\lambda^l$  et  $\lambda^u \in \mathbb{R}$  tels que  $v(\lambda^l) > 0$  et  $v(\lambda^u) < 0$ , et  $x^l$  et  $x^u$  les solutions optimales respectives de  $Q(\lambda^l)$  et  $Q(\lambda^u)$ . La droite d'équation

$$Z(\lambda) = v(\lambda^l) \frac{\lambda - \lambda^u}{\lambda^l - \lambda^u} - v(\lambda^u) \frac{\lambda - \lambda^l}{\lambda^l - \lambda^u}$$

joignant les points  $(\lambda^l, v(\lambda^l))$  et  $(\lambda^u, v(\lambda^u))$  coupe l'axe des  $\lambda$  en

$$q(\lambda^l, \lambda^u) = \frac{\lambda^u v(\lambda^l) - \lambda^l v(\lambda^u)}{v(\lambda^l) - v(\lambda^u)}.$$

PROPOSITION 4.5 :

$$\lambda^* \leq q(\lambda^l, \lambda^u).$$

```

{initialisation}
    déterminer  $\lambda^l$  et  $\lambda^u$  tels que  $\lambda^l < \lambda^* < \lambda^u$ ;
{début}
    répéter
         $\lambda \leftarrow \frac{\lambda^l + \lambda^u}{2}$  ;
        résoudre  $Q(\lambda)$  ;
        si  $v(\lambda) > 0$  alors  $\lambda^l \leftarrow \lambda$ 
        sinon
            si  $v(\lambda) < 0$  alors  $\lambda^u \leftarrow \lambda$ 
        finsi
    finsi
    jusqu'à  $v(\lambda) = 0$ 
{fin}

```

Figure 8. – Algorithme de recherche dichotomique.

### Méthode de recherche dichotomique [34]

On considère un intervalle  $[\lambda^l, \lambda^u]$  contenant  $\lambda^*$  (il suffit pour cela que  $v(\lambda^l) > 0$  et  $v(\lambda^u) < 0$ ) pour le réduire par dichotomie tout en gardant la propriété  $\lambda^* \in [\lambda^l, \lambda^u]$ .

L'arrêt se fait lorsque la longueur de l'intervalle est suffisamment petite.

Ainsi, en posant  $\lambda = \frac{\lambda^l + \lambda^u}{2}$ , l'intervalle  $[\lambda, \lambda^u]$  (respectivement  $[\lambda^l, \lambda]$ ) si  $v(\lambda) > 0$  (resp.  $v(\lambda) < 0$ ) est considéré à l'itération suivante (Fig. 8).

La convergence est linéaire et la longueur de l'intervalle après  $p$  itérations est  $\frac{\lambda^u - \lambda^l}{2^p}$ .

### Méthode de recherche dichotomique modifiée [34]

La première amélioration, conséquence directe des propositions 4.4 et 4.5, permet d'envisager avant dichotomie un intervalle inclus dans  $[\lambda^l, \lambda^u]$  :

**PROPOSITION 4.6 :** Soit  $\lambda^l$  et  $\lambda^u \in \mathbb{R}$  vérifiant  $v(\lambda^l) > 0$  et  $v(\lambda^u) < 0$ . Étant donné  $x^l$  une solution optimale de  $Q(\lambda^l)$ , et  $q(\lambda^l, \lambda^u) = \frac{\lambda^u v(\lambda^l) - \lambda^l v(\lambda^u)}{v(\lambda^l) - v(\lambda^u)}$ , alors

$$\lambda^l \leq \frac{f(x^l)}{h(x^l)} \leq \lambda^* \leq q(\lambda^l, \lambda^u) \leq \lambda^u.$$

Ainsi, à chaque itération, l'intervalle  $\left[ \frac{f(x^l)}{h(x^l)}, q(\lambda^l, \lambda^u) \right]$  (voir Fig. 6) est utilisé à la place de  $[\lambda^l, \lambda^u]$ .

Au lieu d'envisager systématiquement le milieu de l'intervalle de confiance, la deuxième amélioration consiste en l'utilisation d'une fonction de poids plus générale  $\alpha(p)$  croissante et vérifiant  $\alpha(p) \geq 2$  et  $\lim_{p \rightarrow +\infty} \alpha(p) = +\infty$  (par exemple  $\alpha(p) = p + 1$  ou  $2^p$ ) qui permet d'élarguer plus de la moitié de cet intervalle (Fig. 9).

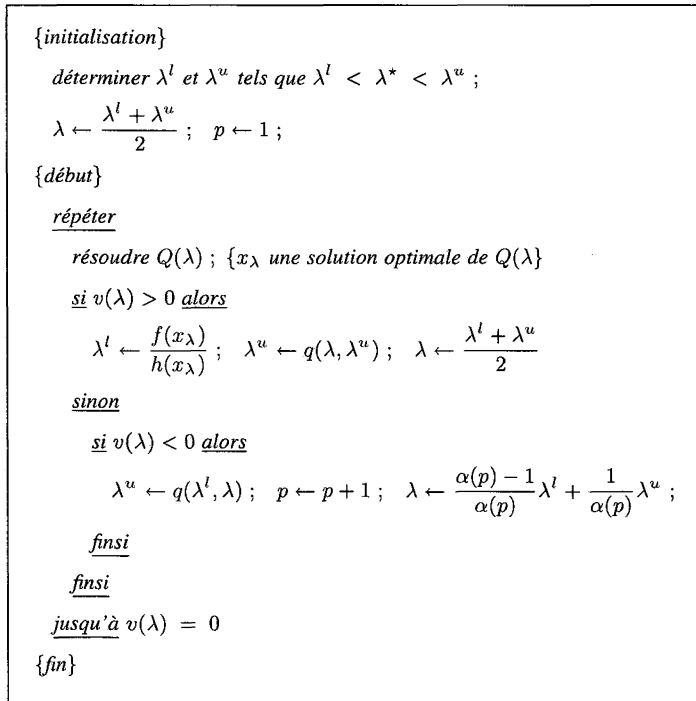


Figure 9. – Algorithme de recherche dichotomique modifiée.



*Méthode d'interpolation convexe [34]*

Les différentes méthodes présentées ci-dessus sont basées sur des interpolations linéaires, alors que la fonction  $v$  est convexe. Afin d'exploiter cette propriété, une fonction d'interpolation convexe  $\hat{v}$  entre les points  $(\lambda^l, v(\lambda^l))$  et  $(\lambda^u, v(\lambda^u))$  est calculée avec les sous-gradients  $-h(x^l)$  et  $-h(x^u)$ . Sa racine  $\hat{\lambda}$  considérée à l'itération suivante (Fig. 10) vérifie la propriété suivante :

PROPOSITION 4.7 : Soit  $\lambda^l$  et  $\lambda^u \in \mathbb{R}$  vérifiant  $v(\lambda^l) > 0$  et  $v(\lambda^u) < 0$ ,  $x^l$  une solution optimale de  $Q(\lambda^l)$ ,  $x^u$  une solution de  $Q(\lambda^u)$  et  $q(\lambda^l, \lambda^u) = \frac{\lambda^u v(\lambda^l) - \lambda^l v(\lambda^u)}{v(\lambda^l) - v(\lambda^u)}$ . Alors on a

$$\lambda^l \leq \min \left\{ \frac{f(x^l)}{h(x^l)}, \frac{f(x^u)}{h(x^u)} \right\} \leq \max \left\{ \frac{f(x^l)}{h(x^l)}, \frac{f(x^u)}{h(x^u)} \right\} \\ \leq \hat{\lambda} \leq q(\lambda^l, \lambda^u) \leq \lambda^u.$$

Une étude expérimentale [34] comparant ces différentes méthodes sur le problème du sac à dos fractionnaire en variables continues bornées :

$$\begin{cases} \max & \frac{c_0 + cx}{d_0 + dx} \\ \text{s.c.} & ax = b \\ & 0 \leq x_j \leq 1 \quad j = 1, \dots, n \end{cases}$$

```

{initialisation}
    déterminer  $\lambda^l$  et  $\lambda^u$  tels que  $\lambda^l < \lambda^* < \lambda^u$  ;
{début}
    répéter
        calculer  $\hat{v}(\lambda)$  et sa racine  $\hat{\lambda}$  ;
        résoudre  $Q(\hat{\lambda})$  ;
        si  $v(\hat{\lambda}) > 0$  alors  $\lambda^l \leftarrow \hat{\lambda}$ 
        sinon
            si  $v(\hat{\lambda}) < 0$  alors  $\lambda^u \leftarrow \hat{\lambda}$ 
        finsi
    finsi
    jusqu'à  $v(\hat{\lambda}) = 0$ 
{fin}

```

Figure 10. – Algorithme d'interpolation convexe.

montre qu'en général, la méthode d'interpolation convexe est la plus rapide. Cependant, la convergence superlinéaire n'est pas établie théoriquement. D'autre part, il est montré que les méthodes de Newton et de recherche dichotomique modifiée dominent les méthodes d'interpolation linéaire et de recherche dichotomique dont les performances sont très instables.

*Méthode de Meggido [46]*

Sachant que pour  $\lambda$  donné  $c_0 - \lambda d_0$  est une constante, le programme paramétré associé à un programme hyperbolique peut aussi être défini comme suit :

$$(Q'(\lambda)) \quad \begin{cases} \max & (c - \lambda d)x \\ \text{s.c.} & Ax \leq b \\ & x \in X. \end{cases}$$

Sa valeur notée  $v'(\lambda)$ , exclut désormais la constante  $c_0 - \lambda d_0$ . Ainsi, la proposition 4.3 devient

PROPOSITION 4.8 :

- (a)  $v'(\lambda) = \lambda d_0 - c_0 \iff \lambda = \lambda^*$ ,
- (b)  $v'(\lambda) > \lambda d_0 - c_0 \iff \lambda < \lambda^*$ ,
- (c)  $v'(\lambda) < \lambda d_0 - c_0 \iff \lambda > \lambda^*$ .

La méthode de Meggido n'est autre qu'une recherche dichotomique sur l'ensemble des abscisses des points de cassure de l'enveloppe supérieure de la collection des droites  $c_j - \lambda d_j$ ,  $j = 1, \dots, n$  (voir Fig. 11). Elle est basée sur le raisonnement suivant : étant donné un intervalle  $[\lambda^l, \lambda^u]$  contenant la valeur  $\lambda^*$  du problème fractionnaire ( $P$ ) et un point de cassure d'abscisse  $\lambda^1$ , si  $v'(\lambda^1)$  est strictement supérieure à  $\lambda^1 d_0 - c_0$ , alors d'après la proposition 4.8,  $\lambda^1$  est strictement inférieure à  $\lambda^*$ . On réduit donc l'intervalle de confiance en donnant à  $\lambda^l$  la valeur  $\lambda^1$ . À l'inverse, si  $v'(\lambda^1)$  est strictement inférieure à  $\lambda^1 d_0 - c_0$ , c'est  $\lambda^u$  qui prend la valeur  $\lambda^1$ .

Le résultat fondamental de complexité est le suivant : si pour un  $\lambda$  donné, le programme paramétré peut être résolu par un algorithme de complexité polynômiale, alors le programme fractionnaire peut être aussi résolu en un temps polynômial [46].

Cette approche a été adaptée par Hashizume *et al.* [31] pour construire une valeur  $\varepsilon$ -approchée de  $\lambda^*$ .

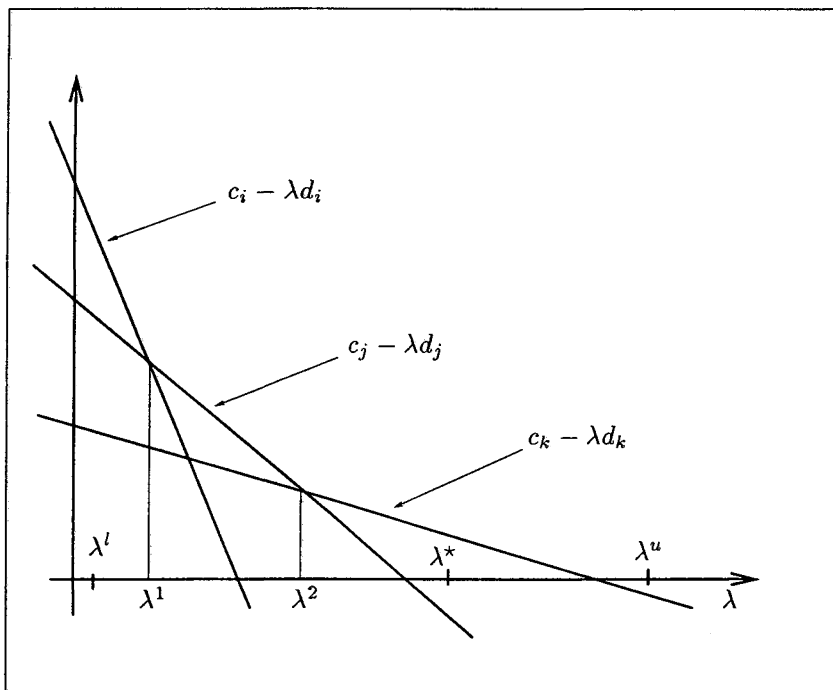


Figure 11. – Suite des valeurs engendrées par la méthode de Meggido.

Pour  $0 < \varepsilon < 1$  donné, on rappelle qu'un algorithme  $\varepsilon$ -approchant fournit une valeur  $v'_\varepsilon(\lambda)$  qui est telle que

$$v'_\varepsilon(\lambda) \leq v'(\lambda) \quad \text{et} \quad \frac{v'(\lambda) - v'_\varepsilon(\lambda)}{v'(\lambda)} \leq \varepsilon$$

ou encore

$$(1 - \varepsilon)v'(\lambda) \leq v'_\varepsilon(\lambda) \leq v(\lambda) \quad \text{et} \quad v'_\varepsilon(\lambda) \leq v'(\lambda) \leq \frac{1}{1 - \varepsilon}v'_\varepsilon(\lambda).$$

En combinant ces inégalités avec la proposition 4.8 on obtient :

PROPOSITION 4.9 [31]

(A)  $v'_\varepsilon(\lambda) > \lambda d_0 - c_0 \implies \lambda < \lambda^*,$

(B)  $v'_\varepsilon(\lambda) < (1 - \varepsilon)\lambda d_0 - c_0 \implies \lambda > \lambda^*,$

(C)  $(1 - \varepsilon)\lambda d_0 - c_0 \leq v'_\varepsilon(\lambda) < \lambda d_0 - c_0 \implies \lambda \text{ est une } \varepsilon\text{-approximation de } \lambda^*.$

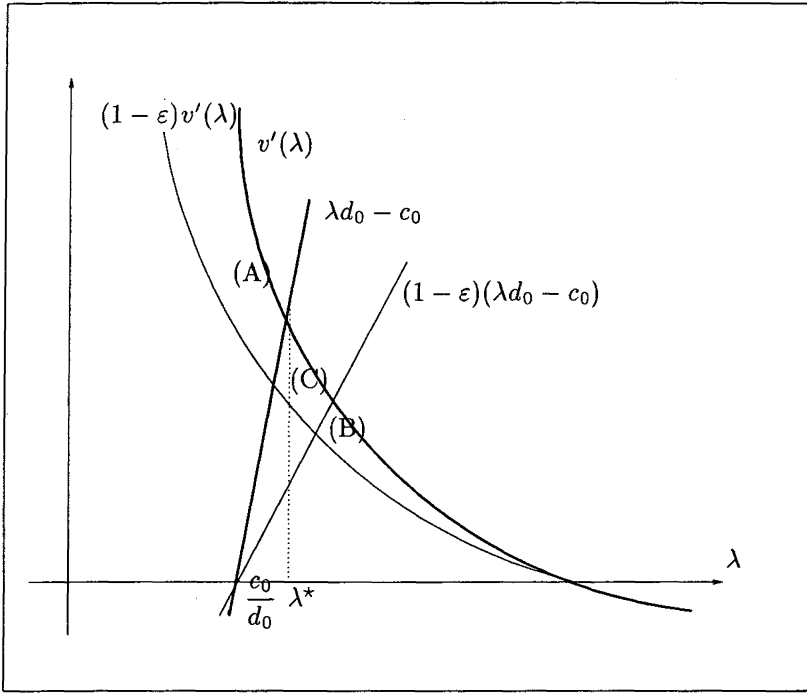


Figure 12.

Ainsi le but de l'algorithme est de trouver  $\lambda$  satisfaisant la condition (C) de la proposition 4.9 (Fig. 12).

L'adaptation de ce principe au problème spécifique du sac à dos hyperbolique en variables 0-1 permet de plus d'envisager des algorithmes  $\varepsilon$ -approchants polynômiaux ou totalement polynômiaux de la littérature pour le problème de sac à dos linéaire en variables 0-1. Plus précisément, Hashizume *et al.* [31] utilisent un algorithme  $\varepsilon$ -approchant de complexité  $O(n^4/\varepsilon)$  pour le problème de sac à dos linéaire en variables 0-1 [42, 58] et aboutissent à un temps global de résolution du programme hyperbolique en variables 0-1 en  $O\left(\frac{n^4 \log \log \bar{c}}{\varepsilon} + n^5 \left(\log n + \frac{\log(1/\varepsilon)}{\varepsilon}\right)\right)$  où  $\bar{c} = \max \{c_i, d_i : i \in \{1, \dots, n\}\}$ .

### Méthode de type primal

En notant que la connaissance d'une solution réalisable  $\underline{x}_\lambda$  de  $Q(\lambda)$  vérifiant  $f(\underline{x}_\lambda) - \lambda h(\underline{x}_\lambda) > 0$  suffit à conclure que  $\lambda < \lambda^*$ , plusieurs

auteurs ont privilégié l'utilisation d'heuristiques constructives. Ce principe peut être illustré comme suit : tout algorithme itératif de montée engendrant une suite de solutions peut être interrompu dès qu'une solution  $\underline{x}_\lambda$  vérifie  $f(\underline{x}_\lambda) - \lambda h(\underline{x}_\lambda) > 0$ , sinon la résolution de  $Q(\lambda)$  doit être exacte. En notant dans cette dernière alternative  $x_\lambda$  une solution optimale de  $Q(\lambda)$ , le processus est itéré en donnant à  $\lambda$  la valeur  $f(\underline{x}_\lambda)/h(\underline{x}_\lambda)$  dans le premier cas ou la valeur  $f(x_\lambda)/h(x_\lambda)$  dans le second cas.

Dans le cas d'un programme linéaire, Martos [43, 44] exploite cette propriété en combinant la méthode de Newton et l'algorithme primal du Simplexe.

Cette approche a été étendue au cas des programmes fractionnaires en nombres entiers par Granot et Granot [23] et Ishii *et al.* [37].

## 5. RÉOLUTION D'UN PROBLÈME ÉQUIVALENT À OBJECTIF NON FRACTIONNAIRE

La transformation du programme fractionnaire en un programme équivalent à objectif non fractionnaire est obtenu par un changement de variables. À l'inverse de l'approche paramétrée, ce changement de variables induit l'ajout d'une contrainte et d'une variable. Plus précisément, cette transformation, proposée par Charnes et Cooper [13] pour le programme hyperbolique en variables continues

$$(P) \quad \begin{cases} \max & \frac{c_0 + cx}{d_0 + dx} \\ \text{s.c.} & Ax \leq b \\ & x \geq 0 \end{cases}$$

s'effectue en introduisant deux nouvelles variables

$$y = \left( \frac{1}{d_0 + dx} \right) x \quad \text{et} \quad t = \frac{1}{d_0 + dx}$$

pour aboutir à un programme linéaire équivalent

$$(PE) \quad \begin{cases} \max & cy + c_0 t \\ \text{s.c.} & Ay - bt \leq 0 \\ & dy + d_0 t = 1 \\ & y, t \geq 0. \end{cases}$$

Cette notion d'équivalence est précisée ci-dessous :

**PROPOSITION 5.1** (Charnes-Copper [13]) : Si  $(y^*, t^*)$  est une solution optimale de (PE), alors  $t^* > 0$  et  $x^* = \frac{y^*}{t^*}$  est une solution optimale de (P).

En fait, s'il existe une solution réalisable  $x$  telle que  $\frac{c_0 + cx}{d_0 + dx} > 0$ , la contrainte d'égalité  $dy + d_0 t = 1$  peut être remplacée par la contrainte d'inégalité  $dy + d_0 t \leq 1$ , plus simple à traiter (Schaible [63]).

Cette transformation en un programme linéaire équivalent a pour but d'appliquer les algorithmes standards tels que la méthode du Simplexe (Arsham et Khan [4] et Charnes et Cooper [13]). Pour les programmes fractionnaires en variables entières, Granot et Granot [23] proposent une méthode de génération de coupes (de type Gomory) appliquée au programme linéaire (PE). D'autre part, dans le cas d'un programme hyperbolique à  $n$  variables bivalentes, Williams [73] propose une transformation spécifique en un programme linéaire équivalent en variables mixtes dont la taille croît de  $n+1$  variables continues et de  $3n$  contraintes. Ce nouveau programme est résolu par un algorithme de type Branch -and- Bound.

*Remarque 5.1* (Craven [14], Schaible et Ibaraki [65])

Cette transformation peut être plus généralement effectuée sur un programme fractionnaire concave-convexe

$$(P) \quad \begin{cases} \max & \frac{f(x)}{h(x)} \\ \text{s.c.} & g(x) \leq 0 \\ & x \geq 0 \end{cases}$$

pour aboutir à un programme concave équivalent

$$(PE) \quad \begin{cases} \max & t f\left(\frac{y}{t}\right) \\ \text{s.c.} & g\left(\frac{y}{t}\right) \leq 0 \\ & t h\left(\frac{y}{t}\right) = 1 \\ & y, t \geq 0 \end{cases}$$

obtenu en posant

$$y = \left(\frac{1}{h(x)}\right)x \quad \text{et} \quad t = \frac{1}{h(x)}.$$

*Remarque 5.2 : Dans le cas d'un programme hyperbolique en variables 0-1, cette transformation induit des contraintes de type quadratique. Plus précisément, en notant que  $x_j \in \{0, 1\}$  est équivalent à  $x_j(x_j - 1) = 0$ , la contrainte  $\frac{1}{t} y_j \in \{0, 1\}$  s'écrit sous la forme équivalente  $y_j(y_j - t) = 0$ , soit  $y_j^2 - y_j t = 0$  qui est de type quadratique.*

## 6. PROBLÈMES DUALS

On considère le programme fractionnaire

$$(P) \quad \begin{cases} \max & \frac{f(x)}{h(x)} \\ \text{s.c.} & g(x) \leq 0 \\ & x \in X \end{cases}$$

avec comme hypothèses classiques :

- $F(P) \neq \emptyset$ ,
- les fonctions  $f$ ,  $h$  et  $g$  sont continues sur  $\mathbb{R}^n$ ,
- $\forall x \in X : h(x) > 0$ ,
- $\exists x \in F(P) : f(x) > 0$ .

La dualisation directe des contraintes de (P) peut engendrer un saut de dualité démesuré et même une valeur infinie pour le dual comme le montre l'exemple de Schaible [63] :

$$\max \left\{ \frac{1}{x} \mid x > 0, x \geq 1 \right\} = 1$$

alors que

$$\forall u \geq 0, \quad \sup_{x > 0} \left\{ \frac{1}{x} + u(x - 1) \right\} = +\infty.$$

Afin de remédier à cet inconvénient, plusieurs approches sont suggérées [1, 6, 8, 14, 21, 63, 65, 72], celles basées sur le programme équivalent, ou sur le programme paramétré, ou bien celles utilisant directement le programme d'origine après une réécriture fractionnaire des contraintes.

### 6.1. Dual utilisant le problème équivalent

Schaible [63] propose une approche basée sur le programme équivalent (voir Sect. 5). On sait que le programme fractionnaire concave-convexe (P)

peut être transformé par un changement de variables, en un programme concave équivalent (PE). Ainsi, le programme fractionnaire équivalent au dual de ce dernier permet de définir un dual de (P).

Afin d'illustrer cette dernière notion, considérons le programme hyperbolique :

$$(P) \quad \begin{cases} \max & \frac{c_0 + cx}{d_0 + dx} \\ \text{s.c.} & Ax = b \\ & x \geq 0 \end{cases}$$

dont le programme équivalent

$$(PE) \quad \begin{cases} \max & cy + c_0t \\ \text{s.c.} & Ay - bt = 0 \\ & dy + d_0t = 1 \\ & y, t \geq 0 \end{cases}$$

admet pour dual

$$(DE) \quad \begin{cases} \min & r \\ \text{s.c.} & A^T s + d^T r \geq c \\ & -bs + d_0r \geq c_0. \end{cases}$$

Ainsi, le programme fractionnaire équivalent à (DE) permet de définir un dual de (P) :

$$(D) \quad \begin{cases} \min & \frac{r}{by - d_0r + 1} \\ \text{s.c.} & (A - cb)y + (d + d_0c)r \geq c \\ & y \geq 0, r \geq 0. \end{cases}$$

## 6.2. Dual utilisant le problème paramétré

La notion de programme paramétré (voir Sect. 4), permet également de définir une notion de dualité. Bitran et Magnanti [8] l'ont utilisée pour étendre quelques résultats de la dualité classique au cas des fonctions fractionnaires non-différentiables. Bector [6] retrouve ces théorèmes par l'approche suivante.



### 6.3. Duals lagrangiens

Une réécriture fractionnaire des contraintes, proposée initialement par Gold'stein [21] et étudiée ensuite par Bector [6] et Weir et Mond [72], permet d'envisager un modèle homogène équivalent au modèle initial (invariance à la fois du domaine et de la valeur).

#### *Relaxation lagrangienne*

Le dénominateur de l'objectif étant supposé strictement positif pour tout  $x$  de  $X$ , on peut envisager le programme équivalent suivant :

$$(P1) \quad \begin{cases} \max & \frac{f(x)}{h(x)} \\ \text{s.c.} & \left( \frac{1}{h(x)} \right) g(x) \leq 0 \\ & x \in X \end{cases}$$

dont le dual lagrangien est défini par :

$$\min_{u \in \mathbb{R}_+^m} \max_{x \in X} \left\{ \frac{f(x) - ug(x)}{h(x)} \right\}.$$

En supposant que toutes les fonctions sont différentiables et en considérant le cas où  $f$  est concave,  $h$  linéaire et  $g$  convexe, Bector [6] prouve différents théorèmes de dualité qui restent encore valides si  $h$  est convexe [63].

En somme, de nombreuses notions de dualités ont été proposées. La plupart de ces études sont restées à un niveau théorique et n'ont pas connu de développements algorithmiques.

En utilisant la même transformation, nous proposons [52] un schéma de décomposition lagrangienne pour les programmes hyperboliques en variables 0–1 et des résultats de dominance de cette dernière sur la relaxation lagrangienne. Nous étendons ainsi aux programmes hyperboliques en variables 0–1, des résultats classiques obtenus dans le cas linéaire par Guignard et Kim [26] et dans le cas quadratique convexe par Michelon et Maculan [48].

Considérons le problème hyperbolique en variables 0–1 suivant

$$(P) \quad \begin{cases} \max & \frac{c_0 + cx}{d_0 + dx} \\ \text{s.c.} & Ax \leq b \\ & x \in X = \{0, 1\}^n. \end{cases}$$

Le problème dual obtenu en dualisant les contraintes  $Ax \leq b$  sous forme fractionnaire est donné par :

$$(DR) \quad \min_{u \in \mathbb{R}_+^m} \max_{x \in X} \frac{(c_0 + ub) + (c - uA)x}{d_0 + dx}.$$

### Décomposition lagrangienne

L'introduction d'une variable de copie permet de découpler l'objectif fractionnaire des contraintes. Plus précisément, on considère le problème  $(P)$  dans lequel est introduite une contrainte de copie  $y = x$  :

$$(P') \quad \begin{cases} \max & \frac{c_0 + cx}{d_0 + dx} \\ \text{s.c.} & y = x \\ & Ay \leq b \\ & x \in X, y \in Y \supseteq X \end{cases}$$

équivalent au problème  $(P)$  d'origine.  $Y$  peut par exemple être égal à  $Co(X)$ , ou tout simplement  $X$ .

La contrainte  $y = x$  est dualisée sous la forme fractionnaire

$$\frac{y}{d_0 + dx} = \frac{x}{d_0 + dx}.$$

Le problème dual associé est du type :

$$(DD) \quad \min_{w \in \mathbb{R}^n} \max_{x, y} \left\{ \frac{c_0 + (c - w)x + wy}{d_0 + dx} \mid Ay \leq b, y \in Y, x \in X \right\}.$$

En nommant  $(K_w)$  le programme linéaire

$$\begin{cases} \max & wy \\ \text{s.c.} & Ay \leq b \\ & y \in Y \end{cases}$$

on obtient une réécriture du problème dual sous la forme

$$(DD) \quad \min_{w \in \mathbb{R}^n} \max_{x \in X} \left\{ \frac{c_0 + v(K_w) + (c - w)x}{d_0 + dx} \right\}.$$

Il faut noter que, contrairement à la décomposition lagrangienne classique, pour  $w \in \mathbb{R}^n$  donné, le calcul de la valeur de la fonction duale nécessite la

résolution successive de deux programmes non indépendants : d'abord un programme linéaire à contraintes d'inégalité qui fournit un ajustement de la constante du numérateur de l'objectif fractionnaire, puis un programme hyperbolique sans contraintes en variables 0-1.

De plus rappelons que la résolution d'un programme hyperbolique sans contraintes en variables 0-1 est de complexité temporelle linéaire [30, 57].

Naturellement, les valeurs  $v(DR)$  et  $v(DD)$  des deux duals fournissent des majorants de la valeur  $v(P)$  du problème primal, et on rappelle que  $v(DD)$  a été prouvée de qualité meilleure que  $v(DR)$  [52].

Notre résultat est ensuite généralisé aux programmes fractionnaires concaves-convexes avec contraintes linéaires [55].

## 7. CONCLUSION

Ce tour d'horizon montre toute la richesse et la variété des travaux consacrés aux programmes fractionnaires qu'ils soient linéaires, non linéaires, en variables continues ou en nombres entiers.

De nombreuses applications, économiques ou algorithmiques dont certaines sont décrites ici, ont été à la source des motivations de nombreuses recherches.

Ce catalogue nous a incités à apporter notre contribution au niveau de deux points clés de la résolution des programmes fractionnaires [51] : la dualité et les heuristiques. Les résultats relatifs à la décomposition lagrangienne résumés en section 6.3 sont détaillés dans [52, 53] et [54] pour les programmes hyperboliques ainsi que dans [55] et [56] pour les programmes fractionnaires concaves-convexes.

## RÉFÉRENCES

1. J. ABRHAM et S. LUTHRA, Comparison of duality models in fractional linear programming, *Zeitschrift Oper. Res.*, 1977, 21, p. 125-130.
2. S. C. AGRAWAL et M. CHAND, A note on integer solutions to linear fractional interval programming problems by Branch and Bound technique, *Naval Res. Logist. Quart.*, 1981, 28, n°4, p. 671-677.
3. Y. ANZAI, On integer fractional programming, *J. Oper. Res. Soc. Japan*, 1974, 17, n°1, p. 49-69.
4. H. ARSHAM et A. B. KHAN, A complete algorithm for linear fractional programs, *Comput. Math. Appl.*, 1990, 20, n°7, p. 11-23.
5. E. BALAS, An additive algorithm for solving linear programs with zero-one variables, *Oper. Res.*, 1965, 13, n°4, p. 517-546.

6. C. R. BECTOR, Duality in nonlinear fractional programming, *Zeitschrift Oper. Res.*, 1973, 17, p. 183-193.
7. B. BEREANU, *Decision regions and minimum risk solutions in linear programming*, in: A. Prekopa, Ed., Colloquium on applications of mathematics to economics, Budapest, 1963, Publ. house of the Hungarian academy of sciences, Budapest, 1965, p. 37-42.
8. G. R. BITRAN et T. L. MAGNANTI, Duality and sensitivity analysis for fractional programs, *Oper. Res.*, 1976, 24, n°4, p. 675-699.
9. M. BLUM, R. W. FLOYD, V. PRATT, R. L. RIVEST et R. E. TARJAN, Time bounds for selection, *J. Comput. Syst. Sci.*, 1973, 7, p. 448-461.
10. P. BOUCHER, A. NAGIH et G. PLATEAU, *Méthodes heuristiques pour le problème hyperbolique en variables 0-1*, Optimizations Days, Montréal, Canada, mai 1996.
11. S. CHANDRA et M. CHANDRAMOHAN, A Branch and Bound Method for Integer Nonlinear Fractional Programs, *ZAMM*, 1980, 60, p. 735-737.
12. S. CHANDRA et M. CHANDRAMOHAN, A note on integer linear fractional programming, *Naval Res. Logist. Quart.*, 1980, 27, p. 171-174.
13. A. CHARNES et W. W. COOPER, Programming with linear fractional functionals, *Naval Res. Logist. Quart.*, 1962, 9, p. 181-186.
14. B. D. CRAVEN, *Fractional Programming*, Helderman, Berlin, 1988.
15. C. DERMAN, On sequential decisions and Markov chains, *Management Sci.*, 1962, 9, p. 16-24.
16. W. DINKELBACH, On nonlinear fractional programming, *Management Sci.*, 1967, 13, p. 492-498.
17. M. FLORIAN et P. ROBILLARD, Programmation hyperbolique en variables bivalentes, *Revue Française d'Informatique et de Recherche Opérationnelle*, 1971, 5, n°1, p. 3-9.
18. A. M. GEOFFRION, An improved implicit enumeration approach for integer programming, *Oper. Res.*, 1969, 17, p. 437-454.
19. A. M. GEOFFRION, The Lagrangean Relaxation for Integer Programming, *Math. Programming*, 1974, 2, p. 82-114.
20. F. GLOVER, Surrogate constraints, *Oper. Res.*, 1968, 16, n°4, p. 741-749.
21. L. S. GOLDSTEIN, Dual problems of convex and fractionally-convex programming in functional spaces, *Soviet Math. Dokl.*, 1967, 8, p. 212-216.
22. D. GRANOT et F. GRANOT, On solving fractional (0, 1) programs by implicit enumeration, *INFOR*, 1976, 14, p. 241-249.
23. D. GRANOT et F. GRANOT, On integer and mixed integer fractional programming problems, *Ann. Discrete Math.*, 1977, 1, p. 221-231.
24. M. GRUSPAN, *Fractional programming: A survey*, Technical Report 50, Department of Industrial and Systems Engineering, University of Florida, 1971.
25. M. GRUNSPAN et M. E. THOMAS, Hyperbolic integer programming, *Naval Res. Logist. Quart.*, 1973, 20, n°2, p. 341-356.
26. M. GUIGNARD et S. KIM, Lagrangean decomposition for integer programming: A model yielding stronger Lagrangean bounds, *Math. Programming*, 1987, 32, p. 215-228.
27. E. GUTENBURG, *Einführung in die Betriebswirtschaftlehre*, Gabler Wiesbaden, 1975.
28. P. L. HAMMER et S. RUDEANU, *Boolean methods in operations research and related areas*, Springer, Berlin - New York, 1968.
29. P. HANSEN, M. MINOUX et M. LABBÉ, Extension de la programmation linéaire généralisée au cas des programmes mixtes, *C. R. Acad. Sci. Paris Sér. I Math.*, 1987, 305, p. 569-572.

30. P. HANSEN, M. V. POGGI DE ARAGAO et C. C. RIBEIRO, Hyperbolic 0-1 programming and query information retrieval, *Math. Programming*, 1991, 52, p. 255-263.
31. S. HASHIZUME, M. FUKUSHIMA, N. KATO et T. IBARAKI, Approximation Algorithms for combinatorial fractional programming problem, *Math. Programming*, 1991, 37, p. 255-267.
32. M. H. HEINE, Distance between sets as an objective measure of retrieval effectiveness, *Information Storage and Retrieval*, 1973, 9, p. 181-198.
33. E. HEINEN, *Grundlagen betriebswirtschaftlicher Entscheidungen*, Das Zielsystem der Unternehmung Gabler Wiesbaden, 1971.
34. T. IBARAKI, Parametric approaches to fractional programs, *Math. Programming*, 1983, 26, p. 345-362.
35. O. H. IBARRA et C. E. KIM, Fast approximation algorithms for the knapsack and sum of subsets problems, *J. Assoc. Comput. Machinery*, 1975, 22, n°4, p. 463-468.
36. J. R. ISBELL et W. H. MARLOW, Attribution games, *Naval Res. Logist. Quart.*, 1956, 3, p. 71-94.
37. H. ISHII, T. IBARAKI et H. MINE, A primal cutting plane algorithm for integer fractional programming problems, *J. Oper. Res. Soc. Japan*, 1976, 19, p. 228-244.
38. H. ISHII, T. IBARAKI et H. MINE, Fractional Knapsack problems, *Math. Programming*, 1977, 13, p. 255-271.
39. J. G. KALLBERG et W. T. ZIEMBA, Generalized Concave Functions in Stochastic Programming and Portfolio Theory. in: S. Schaible and W. T. Ziemba, eds., *Generalized Concavity in Optimization and Economics*, Academic Press, New York 1981, p. 719-767.
40. M. KLEIN, Inspection-maintenance-replacement schedule under Markovian deterioration, *Management Sci.*, 1963, 9, p. 25-32.
41. L. S. LASDON, *Optimization Theory for Large Systems*, Collier-MacMillan, 1970.
42. S. MARTELLO et P. TOTH, *Knapsack Problems*, J. Wiley & Sons, 1990.
43. B. MARTOS, Hyperbolic programming, *Naval Res. Logist. Quart.*, 1964, 11, p. 135-155.
44. B. MARTOS, The Direct Power of Adjacent Vertex Programming Methods, *Management Sci.*, 1965, 12, n°3, p. 241-252.
45. B. MARTOS, *Nonlinear programming: Theory and methods*, North-Holland, Amsterdam, 1975.
46. N. MEGGIDO, Combinatorial optimization with rational objective functions, *Math. Oper. Res.*, 1979, 4, n°4, p. 414-424.
47. P. MICHELON et N. MACULAN, An algorithm for the mixed or integer fractional programming, Research Report ES-161-88, University of Rio de Janeiro, 1988.
48. P. MICHELON et N. MACULAN, Lagrangean methods for 0-1 quadratic problems, *Discrete Applied Mathematics*, 1993, 42, p. 257-269.
49. K. M. MJELDE, Allocation of resources according to a fractional objective, *European J. Oper. Res.*, 1978, 2, p. 116-124.
50. K. M. MJELDE, *Methods of the Allocation of Limited ressources*, J. Wiley and Sons, Chichester, 1983.
51. A. NAGIH, *Sur la résolution des programmes fractionnaires en variables 0-1*, Thèse de Doctorat, Université Paris 13, France, juin 1996.
52. A. NAGIH et G. PLATEAU, A Lagrangian Decomposition for 0-1 Hyperbolic Programming Problems, *Int. J. Math. Algorithms*, to appear.
53. A. NAGIH et G. PLATEAU, *Méthodes lagrangiennes pour les problèmes hyperboliques en variables 0-1*, FRANCORO : Rencontres Francophones de Recherche Opérationnelle, Mons, Belgique, juin 1995.

54. A. NAGIH et G. PLATEAU, An exact Method for the 0-1 Fractional Knapsack Problem, INFORMS, New Orleans, USA, 29 octobre - 1<sup>er</sup> novembre 1995.
55. A. NAGIH et G. PLATEAU, *Sur la résolution des programmes fractionnaires en variables 0-1*, CIRO : Conférence Internationale en Recherche Opérationnelle, Marrakech, Maroc, juin 1996.
56. A. NAGIH et G. PLATEAU, *Dualité lagrangienne en programmation fractionnaire concave-convexe en variables entières*, Rapport de Recherche, LIPN 96, Université Paris 13, 1996.
57. A. NAGIH et G. PLATEAU, A Partition Algorithm for 0-1 unconstrained hyperbolic programs, *Investigación Oper.*, 1999, 8, n° 1.
58. C. H. PAPADIMITRIOU et K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
59. J. PONSTEIN, Seven Kinds of convexity, *SIAM Review*, 1967, 9, n° 1.
60. T. RADZIK, Newton's method for fractional combinatorial optimization, *Report STAN-CS-92-1406*, Stanford university, California, 1992.
61. P. ROBILLARD, 0-1 hyperbolic programming, *Naval Res. Logist. Quart.*, 1971, 18, p. 47-58.
62. A. L. SAIPE, Solving a (0, 1) hyperbolic program by branch and bound, *Naval Res. Logist. Quart.*, 1975, 22, p. 397-416.
63. S. SCHAIBLE, Duality in fractional programming: a unified approach, *Oper. Res.*, 1976, 24, n° 3, p. 452-461.
64. S. SCHAIBLE, Fractional programming: Applications and algorithms, *European J. Oper. Res.*, 1981, 17, p. 111-120.
65. S. SCHAIBLE et T. IBARAKI, Fractional programming, *European J. Oper. Res.*, 1983, 12, p. 325-338.
66. I. C. SHARMA et K. SWARUP, On duality in linear fractional functionals programming, *Zeitschrift für Operations Research*, 1972, 16, p. 91-100.
67. I. M. STANCU-MINASIAN, Applications of the fractional programming, *Econom. Comput. Econom. Cybernet. Stud. Res.*, 1980, 14, p. 69-86.
68. C. J. VAN RIJBERGEN, Function of evaluation, *The Journal of Documentation*, 1974, 30, n° 4, p. 365-373.
69. V. VERMA, Constrained Integer Linear Fractional Programming Problem, *Optimization*, 1990, 21, n° 5, p. 749-757.
70. V. VERMA, H. C. BAKHSI et M. C. PURI, Ranking in integer linear fractional programming problems, *Methods and Models of Operations Research*, 1990, 34, p. 325-334.
71. H. M. WAGNER et J. S. C. YUAN, Algorithmic equivalence in linear fractional programming, *Management Sci.*, 1968, 14, p. 301-306.
72. T. WEIR et B. MOND, Duality for Fractional Programming without Constraint Qualification, *Utilitas Mathe.*, 1990, 38, p. 193-197.
73. H. P. WILLIAMS, Experiments in the formulation of integer programming problems, *Math. Programming Studi.*, 1974, 2, p. 180-197.
74. H. WOLF, Parametric Analysis in linear fractional programming, *Oper. Res.*, 1986, 34, n° 6, p. 930-937.
75. W. T. ZIEMBA, C. PARKAN et R. BROOKS-HILL, Calculation of investment portfolios with risk free borrowing and lending, *Management Sci.*, 1974, 21, p. 209-222.