Méthodes de pénalités logarithmiques en optimisation combinatoire

Bernard Rapacchi

THESE

présentée à

l' Université Scientifique et Médicale de Grenoble

pour obtenir le grade de
DOCTEUR INGENIEUR
«recherche opérationnelle»

par

RAPACCHI Bernard

METHODES DE PENALITES LOGARITHMIQUES EN OPTIMISATION COMBINATOIRE.

Thèse soutenue le 12 janvier 1982 devant la commission d'examen.

M. SAKAROVITCH	Président
C. BENZAKEN J. FONLUPT F. ROBERT J.P. UHRY	Examinateurs

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

année scolaire 1980-1981

Président de l'Université : M. J.J. PAYAN

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS DE 1ère CLASSE

Mlle	AGNIUS DELORD Claudine	Biophysique
	ALARY Josette	Chimie analytique
MM.	AMBLARD Pierre	Clinique dermatologie
	AMBROISE THOMAS Pierre	Parasitologie
	ARNAUD Paul	Chimie
	ARVIEU Robert	Physique nucléaire
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale A
	BEAUDOING André	Clinique pédiatrie et puériculture
	BELORISKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
Mme	BERIEL Hélène	Pharmacodynamie
M.	BERNARD Alain	Mathématiques pures
Mme	BERTRANDIAS Françoise	Mathématiques pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques pures
	BEZES Henri	Clinique chirurgicale & traumatologie
	BILLET Jean	Géographie
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET EYMARD Joseph	Clinique Hépato-gastro-entérologie
Mme	BONNIER Jane-Marie	Chimie générale
MM.	BOUCHERLE André	Chimie et toxicologie
	BOUCHET Yves	Anatomie
	BOUCHEZ Robert	Physique nucléaire
	BRAVARD Yves	Géographie

.../...

MM.	BUTEL Jean	Orthopédie
	CABANEL Guy	Clinique rhumatologie et hydrologie
	CARLIER Georges	Biologie végétale
	CAU Gabriel	Médecine légale et toxicologie
	CAUQUIS Georges	Chimie organique
	CHARACHON Robert	Clinique O.R.L.
	CHATEAU Robert	Clinique neurologique
	CHIBON Pierre	Biologie animale
	COEUR André	Chimie analytique et bromotologique
	COUDERC Pierre	Anatomie pathologique
	CRABBE Pierre	C.E.R.M.O.
	DAUMAS Max	Géographie
	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DELOBEL Claude	M.I.A.G.
	DELORMAS Pierre	Pneumo-phtisiologique
	DENIS Bernard	Clinique cardiologique
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Electrochimie
	DODU Jacques	Mécanique appliquée IUT 1
	DOLIQUE Jean-Michel	Physique des plasmas
	DUCROS Pierre	Cristallographie
	FONTAINE Jean-Marc	Mathématiques pures
	GAGNAIRE Didier	Chimie physique
	GASTINEL Noël	Analyse numérique
	GAVEND Jean-Michel	Pharmacologie
	GEINDRE Michel	Electro-radiologie
	GERBER Robert	Mathématiques pures
	GERMAIN Jean-Pierre	Mécanique
	GIRAUD Pierre	Géologie
	JANIN Bernard	Géographie
	JEANNIN Charles	Pharmacie galénique
	JOLY Jean-René	Mathématiques pures
	KAHANE André	Physique
	KAHANE Josette	Physique
	KLEIN Joseph	Mathématiques pures
	KOSZUL Jean-Louis	Mathématiques pures
	LACAZE Albert	Hermodynamique
	LACHARME Jean	Biologie cellulaire
	LAJZEROWICZ Joseph	Physique

.../...

Mme	LAJZEROWICZ Jeannine	Physique
MM.	LATREILLE René	Chirurgie thoracique
	LATURAZE Jean	Biochimie pharmaceutiques
	LAURENT Pierre	Mathématiques appliquées
	LE NOC Pierre	Bactériologie virologie
	LLIBOUTRY Louis	Géophysique
	LOISEAUX Jean-Marie	Sciences nucléaires
	LOUP Jean	Géographie
	LUU DUC Cuong	Chimie générale et minérale
	MALINAS Yves	Clinique obstétricale
Mlle	MARIOTTE Anne-Marie	Pharmacognostie
MM.	MAYNARD Roger	Physique du solide
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et pétrographie
	MICOUD Max	Clinique maladies infectieuses
	MOURIQUAND Claude	Histologie
	NEGRE Robert	Mécanique IUT 1
	MOZIERES Philippe	Spectrométrie physique
	OMONT Alain	Astrophysique
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY PEYROULA Jean-Claude	Physique
	PERRET Jean	Sémeiologie médicale (neurologie)
	PERRIER Guy	Géophysique
	PIERRARD Jean-Marie	Mécanique
	RACHAIL Michel	Clinique médicale B
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
Mme	RENAUDET Jacqueline	Bactériologie
M.	REVOL Michel	Urologie
Mme	RINAUDO Marguerite	Chimie CERMAV
MM.	DE ROUGEMONT Jacques	Neuro-chirurgie
	SARRAZIN Roger	Clinique chirurgicale B
Mme	SEIGLE MURANDI Françoise	Botanique et crytogamie
MM.	SENGEL Philippe	Biologie animale
	SIBILLE Robert	Construction mécanique IUT 1
	SOUTIF Michel	Physique
	TANCHE Maurice	Physiologie
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique nucléaire

.../...

MM.	VAN CUTSEM Bernard	Mathématiques appliquées
	VAUQUOIS Bernard	Mathématiques appliquées
	VERAIN Alice	Pharmacie galénique
	VERAIN André	Biophysique
	VIGNAIS Pierre	Biochimie médicale

PROFESSEURS DE 2ème CLASSE

MM.	ARNAUD Yves	Chimie IUT 1
	AURIAULT Jean-Louis	Mécanique IUT 1
	BEGUIN Claude	Chimie organique
	BOITET Christian	Mathématiques appliquées
	BOUTHINON Michel	E.E.A. IUT 1
	BRUGEL Lucien	Energétique IUT 1
	BUISSON Roger	Physique IUT 1
	CASTAING Bernard	Physique
	CHARDON Michel	Géographie
	CHEHIKIAN Alain	E.E.A. IUT 1
	COHEN Henri	Mathématiques pures
	COHENADDAD Jean-Pierre	Physique
	COLIN DE VERDIERE Yves	Mathématiques pures
	CONTE René	Physique IUT 1
	CYROT Michel	Physique du solide
	DEPASSEL Roger	Mécanique des fluides
	DOUCE Roland	Physiologie végétale
	DUFRESNOY Alain	Mathématiques pures
	GASPARD François	Physique
	GAUTRON René	Chimie
	GIDON Maurice	Géologie
	GIGNOUX Claude	Sciences nucléaires
	GLENAT René	Chimie organique
	GOSSE Jean-Pierre	E.E.A. IUT 1
	GROS Yves	Physique IUT 1
	GUITTON Jacques	Chimie
	HACQUES Gérard	Mathématiques appliquées
	HERBIN Jacky	Géographie
	HICTER Pierre	Chimie
	IDELMAN Simon	Physiologie animale
	JOSELEAU Jean-Paul	Biochimie
	JULLIEN Pierre	Mathématiques appliquées
	KERCKOVE Claude	Géologie

.../...

MM.	KRAKOWIACK Sacha	Mathématiques appliquées
	KUHN Gérard	Physique IUT 1
	KUPKA Yvon	Mathématiques pures
	LUNA Domingo	Mathématiques pures
	MACHE Régis	Physiologie végétale
	MARECHAL Jean	Mécanique
	MICHOULIER Jean	Physique IUT 1
Mme	MINIER Colette	Physique IUT 1
MM.	NEMOZ Alain	Thermodynamique
	NOUGARET Marcel	Automatique IUT 1
	OUDET Bruno	Mathématiques appliquées
	PEFFEN René	Métallurgie IUT 1
	PELMONT Jean	Biochimie
	PERRAUD Robert	Chimie IUT 1
	PERRIAUX Jean-Jacques	Géologie minéralogie
	PERRIN Claude	Sciences nucléaires
	PFISTER Jean-Claude	Physique du solide
	PIERRE Jean-Louis	Chimie organique
Mlle	PIERY Yvette	Physiologie animale
MM.	RAYNAUD Hervé	Mathématiques appliquées
	RICHARD Lucien	Biologie végétale
	ROBERT Gilles	Mathématiques pures
	ROBERT Jean-Bernard	Chimie physique
	ROSSI André	Physiologie végétale
	SAKAROVITCH Michel	Mathématiques appliquées
	SARROT REYNAUD Jean	Géologie
	SAXOD Raymond	Biologie animale
Mme	SOUTIF Jeanne	Physique
MM.	STUTZ Pierre	Mécanique
	VIALON Pierre	Géologie
	VIDAL Michel	Chimie organique
	VIVIAN Robert	Géographie

CHARGES D'ENSEIGNEMENT PHARMACIE

MM.	ROCHAS Jacques	Hygiène et hydrologie
	DEMENGE Pierre	Pharmacodynamie

PROFESSEURS SANS CHAIRE (médecine)

M.	BARGE Michel	Neuro-chirurgie

.../...

MM.	BOST Michel	Pédiatrie
	BOUCHARLAT Jacques	Psychiatrie
	CHAMBAZ Edmond	Biochimie (hormonologie)
	CHAMPETIER Jean	Anatomie
	COLOMB Maurice	Biochimie
	COULOMB Max	Radiologie
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	GROULADE Joseph	Biochimie A
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Gérontologie
	JALBERT Pierre	Histologie
	MAGNIN Robert	Hygiène
	PHELIP Xavier	Rhumatologie
	REYMOND Jean-Charles	Chirurgie générale
	STIEGLITZ Paul	Anesthésiologie
	VROUSOS Constantin	Radiothérapie

MAITRES DE CONFERENCES AGREGES (médecine)

MM.	BACHELOT Yvan	Endocrinologie
	BENABID Alim Louis	Médecine et chirurgie
	BERNARD Pierre	Gynécologie obstétrique
	CONTAMIN Charles	Chirurgie thoracique
	CORDONNIER Daniel	Néphrologie
	CROUZET Guy	Radiologie
	DEBRU Jean-Luc	Médecine interne
	DYON Jean-François	Chirurgie infantile
	FAURE Claude	Anatomie et organogènèse
	FAURE Gilbert	Urologie
	FLOYRAC Roger	Biophysique
	FOURNET Jacques	Hépato-gastro-entérologie
	GAUTIER Robert	Chirurgie générale
	GIRARDET Pierre	Anesthésiologie
	GUIDICELLI Henri	Chirurgie générale
	GUIGNIER Michel	Thérapeutique (réanimation)
	JUNIEN-LAVILLAUROY Claude	Clinique O.R.L.
	KOLODIE Lucien	Hématologie biologique
	MALLION Jean-Michel	Médecine du travail
	MASSOT Christian	Médecine interne
	MOUILLON Michel	Ophtalmologie

.../...

MM.	PARAMELLE Bernard	Pneumologie
	RACINET Claude	Gynécologie-Obstétrique
	RAMBAUD Pierre	Pédiatrie
	RAPHAEL Bernard	Stomatologie
	SCHAEFER René	Cancérologie
	SEIGNEURIN Jean-Marie	Bactériologie-virologie
	SOTTO Jean-Jacques	Hématologie
	STOEBNER Pierre	Anatomie-pathologique

AVANT-PROPOS

Il est d'usage en cette page, en une vingtaine de lignes (trop c'est obséquieux, moins c'est trop sec) de présenter ses remerciements.

Située entre la liste du personnel de l'Université (qui date souvent) et une dédicace qui rassemble toujours sa famille et son travail dans le début d'un triptyque trop connu, elle est certainement la seule page que tout le monde lit.

Ce n'est là que le début du rituel de la thèse : remerciements, dédicace, texte, références. Puis le jour de la soutenance : "Monsieur ... va nous présenter une thèse qui s'intitule ...", ensuite les questions, la sortie du jury, le retour du jury (tout le monde se lève), "Je suis heureux ..." (?). Et enfin le pot tant attendu.

Ces remerciements, de plus, ne sont souvent que la compilation d'extraits de thèses précédentes. Tu prends les mêmes formules et tu mets les bons noms aux bons endroits.

Toutefois, je n'ose pas prétendre que cette thèse soit le travail d'une seule personne.

Les personnes qui souffrent le plus dans l'élaboration de ce travail sont celles qui déchiffrent les hiéroglyphes manuscrits pour la composition du texte :
Mmes HOTTELLIER et NEUMANN, et Mme BLANC du C.I.C.G. .
J'ajouterai Gérard BESSON pour son aide matérielle précieuse, et le service de reprographie, en particulier, Claude ANGUILLE, pour son efficacité.

Le résultat d'une thèse est souvent le produit de toute une équipe plutôt que d'un seul auteur. Cette équipe est celle de Recherche Opérationnelle de l'I.M.A.G. . En particulier Jean FONLUPT, Michel SAKAROVITCH et Jean-Pierre UHRY sans lesquels cette équipe n'aurait pas la réputation qu'elle a aujourd'hui.

Je n'oublie pas qu'avant la recherche, il faut avoir eu un enseignement complet. C'est pourquoi j'ai voulu associer ici Claude BENZAKEN et François ROBERT.

Enfin, il faut penser à l'ami, le confesseur de ces deux années, Christophe LACOTE, dont le soutien dans ce travail a été plus que précieux.

Une dernière pensée à M. John NAPIER sans lequel ce travail ne pourrait être.

TABLE DES MATIERES

La fonction entropie a été déjà largement utilisée : en économie, en théorie de l'information, en modélisation de régulation de trafic. Elle est surtout connue dans le problème d'équilibre chimique où celui-ci est atteint pour le minimum de cette fonction. Bregman et Romanovsky ont étudié l'utilisation de cette fonction en Programmation Linéaire en 1975, en l'incorporant à la fonction objectif. S. Erlander, tout récemment (1981) a rajouté au système de contraintes linéaires, une contrainte de type entropie.

Si cette étude se décompose en deux parties, le lien essentiel entre les deux, est l'utilisation de cette fonction entropie.

Dans une première partie, nous étudions, à la manière de Bregman-Romanousky, des problèmes de Programmation Linéaire en ajoutant à la fonction objective une pénalité de type entropie dans le cas très particulier où la matrice des contraintes est en (-1,0,1) et vérifie certaines propriétés d'intégrité.

Le résultat principal de la première partie est que dans le cas de ces problèmes très spécifiques, mais nombreux en Optimisation Combinatoire, la résolution du Programme convexe ainsi associé peut produire des algorithmes polynomiaux. Dans le premier chapitre, nous donnons les notations utilisées et exposons certains résultats préliminaires sur les polyèdres associés à ces problèmes.

Dans les deux chapitres suivants, nous proposons deux méthodes pour résoudre le problème de Programmation Linéaire dans le cas où la matrice des contraintes est en (0, 1) et le polyèdre décrit par ces contraintes a tous ses sommets entiers.

Dans le dernier chapitre de la première partie, nous étudions les cas des matrices totalement unimodulaires. La méthode de "scaling" étudiée pour obtenir un algorithme polynomial dans la longueur des données est celle présentée par Maurras, Truemper, Akgul.

La deuxième partie est axée sur l'étude de la fonction entropie pour une modélisation de trafic. Cette étude a d'abord été entreprise par J. FONLUPT [11]. Nous la complétons dans certains cas :

- si le réseau a des contraintes de capacités supérieures ;
- dans le cas de multiflot ;
- si on a plusieurs sources et plusieurs puits pour un même produit ;
- si le réseau a des contraintes de capacités inférieures.

En dernier lieu, nous présentons et discutons quelques résultats obtenus par ces méthodes sur micro-ordinateur PASCALINE ou sur "gros" ordinateur HB-68.

PARTIE I :

PROBLEMES D'OPTIMISATION COMBINATOIRE

Toutes les méthodes connues pour résoudre certains problèmes d'Optimisation Combinatoire tels que le problème du voyageur de commerce, ou de résolution d'équations linéaires en variables entières exigent des temps de calculs qui croissent exponentiellement avec la taille des données. D'autres problèmes tels que la recherche du plus court chemin dans un graphe, l'arbre de poids maxima ou le couplage de poids maximum peuvent être résolus par des algorithmes polynomiaux utilisant des méthodes classiques de combinatoire (algorithme de pivot, de marquage, de chaînes alternées). Une autre classe de problèmes d'Optimisation Combinatoire contient ceux pour lesquels on ne connaît pas de tels algorithmes combinatoires polynomiaux pour les résoudre. Cependant, ces problèmes sont de la classe des problèmes résolubles au temps polynomial par rapport à la taille des données. Cette notion de "bon algorithme" a été introduite pour la première fois par J. EDMONDS [6].

Pour le problème de programmation linéaire , l'algorithme de Khachijan permet de produire un algorithme polynomial de résolution. Cet algorithme utilise des méthodes de programmation convexe (méthodes des ellipsoïdes) . Par contre , les méthodes combinatoires , plus particulièrement celles dérivées du simplex qui sont d'un point de vue pratique très efficaces dans la majorité des cas , ne sont pas d'un point de vue théorique de bons algorithmes.

Le but essentiel ici est de montrer que, pour des problèmes de Programmation Linéaire que nous étudions, on peut trouver d'autres méthodes qui fournissent des algorithmes polynomiaux. Dans l'algorithme de Khachian et celui que nous proposons, le nombre d'itérations est a priori infini. Mais nous montrons que, par des propriétés similaires à celles utilisées par Khachijan, on peut stopper l'algorithme après un nombre polynomial d'itérations.

Chapitre I :
PRESENTATION. NOTATIONS

I.A. PRESENTATION DU PROBLEME :

Dans toute cette première partie, nous désignerons par $\mathcal{M}(m,n)$ l'ensemble des matrices réelles à m lignes et n colonnes.
A désigne une matrice de $\mathcal{M}(m,n)$ dont les éléments valent 0,1 ou -1 ; b est un vecteur de $\mathbb{R}^m$ et c un vecteur de $\mathbb{R}^n$; b et c seront tous les deux à composantes entières. Dans l'espace vectoriel euclidien $\mathbb{R}^n$, soient x et y deux vecteurs de $\mathbb{R}^n$, nous désignons par $\langle x,y \rangle$ le produit scalaire de x par y. On définit le polyèdre $\mathcal{P}$ par le système d'inégalités :

$$\mathcal{P} \begin{cases} X \in \mathbb{R}^n, \ X \geq 0 \\ AX \leq b \end{cases}$$

Nous ferons dans toute cette partie l'hypothèse essentielle suivante :

Hypothèse H1 : le polyèdre P a tous ses points extrêmes à coordonnées entières.

Le but essentiel de cette première partie est de nous intéresser au problème de Programmation Linéaire PL :

$$PL : \begin{cases} X \in \mathcal{P} \\ \text{Max } \langle c,X \rangle \end{cases}$$

Ce problème plus restreint que le cas général du problème de programmation linéaire est d'une grande importance en Optimisation Combinatoire. En effet, en admettant certaines suppositions sur les vecteurs b et c, nous pouvons inclure ici l'étude de certaines matrices, matrices totalement unimodulaires, parfaites, équilibrées.

La plupart des problèmes d'Optimisation Combinatoire qui se résolvent en temps polynomial peuvent se formuler de terme de telles matrices en (0, 1) : indépendants de poids maximum d'un matroïde, d'intersection de matroïdes, couplages. Nous allons d'abord étudier des algorithmes polynomiaux sur le nombre de colonnes et sur le nombre de lignes de la matrice. Malheureusement, ces problèmes en matrices de(-1,0,1) peuvent avoir un très grand nombre de lignes par rapport au nombre de colonnes. Par exemple, pour le problème du stable de cardinalité maximum dans un graphe parfait, chaque ligne est associée à un vecteur représentatif d'une clique maximale ; on peut donc avoir un nombre m de contraintes qui croît exponentiellement avec n. Grötschel, Lovasz, Schrijver ont résolu le problème en temps polynomial en utilisant un dérivé de la méthode des ellipsoïdes sur un problème annexe et des résultats récents sur les graphes parfaits [16]. De même une variante de l'algorithme proposé, que nous développerons au paragraphe II.A.4., permet dans de nombreux cas de résoudre nos problèmes en temps polynomial.

I.B. NOTATIONS

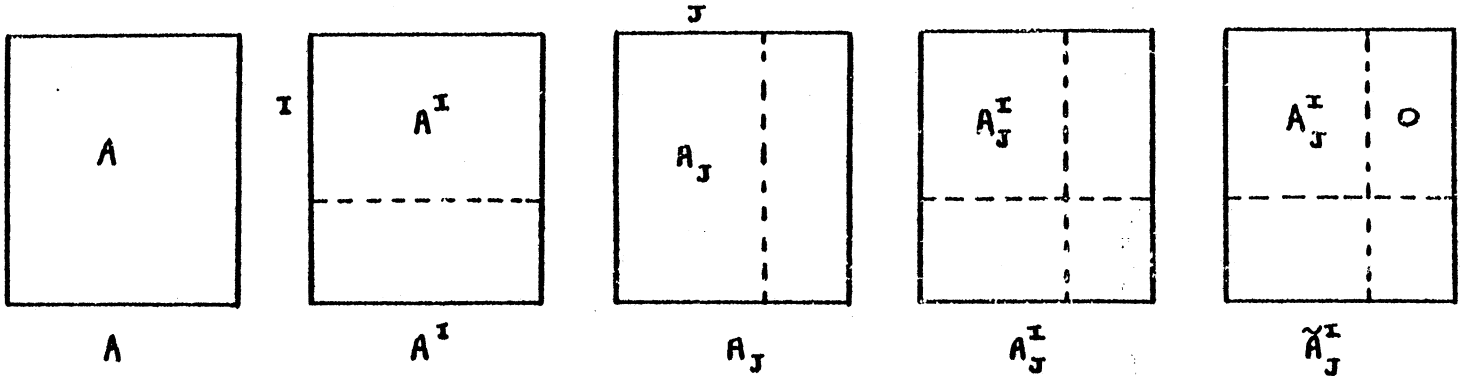
Nous noterons :

- pour $j \in \mathcal{L} = \{1, 2, \ldots, m\}$
  a_j : le j-ième vecteur ligne de A
  $L_j = \{i \in \{1, \ldots, n\} / a_{ji} = 1\}$
- pour $i \in \mathcal{C} = \{1, 2, \ldots, n\}$
  a^i : le i-ième vecteur colonne de A
  $C_i = \{j \in \mathcal{L} / a_{ji} = 1\}$
- pour $J \subset \mathcal{L}$ et $I \subset \mathcal{C}$, A_J désignera la sous-matrice de A dont les lignes sont les a_j avec j appartenant à J ; A^I désignera la sous-matrice de A dont les colonnes sont des a^i avec i élément de I ; et A^I_J désignera la sous-matrice de A dont les éléments sont les a_{ji} avec j élément de J et i élément de I.

$\tilde{A}_J^I$ sera une matrice de $\mathcal{M}(|J|,n)$ telle que :

$\tilde{a}_{ji} = a_{ji}$ si $j \in J$ et $i \in I$

$\tilde{a}_{ji} = 0$ si $j \in J$ et $i \notin I$



- pour J inclus dans $\mathcal{L}$ et I inclus dans $\mathcal{C}$, $\bar{J}$ désignera le complémentaire de J dans $\mathcal{L}$. Soit p = card (I), et x vecteur de $\mathbb{R}^p$, $\hat{x}$ désignera le vecteur de $\mathbb{R}^n$ tel que :

$\hat{x}_i = x_i$ si $i \in I$

$\hat{x}_i = 0$ si $i \notin I$

De plus, on notera : $b_J = (b_j)_{j \in J}$

- on posera :

$$M = \operatorname{Max}_{j \in \mathcal{L}} b_j$$

$$M_i = \operatorname{Min}_{j \in C_i} b_j$$

Nous supposerons sans perte de généralité que les coordonnées b_j de b sont toutes strictement positives dans le cas où A est une (0-1) - matrice.

- on désignera par J_n le vecteur de $\mathbb{R}^n$ dont toutes les composantes valent 1.

I.C. PRELIMINAIRES :

Comme nous l'avons signalé un résultat essentiel qui fait que la méthode des ellipsoïdes est polynomiale est le suivant :

Théorème I.1. : [18] [15] [3]

Si le système :

$$\sum_{i=1}^{n} a_{ji}\, x_i < b_j \quad \text{pour } i = 1,2,\ldots,n$$

On a au moins une solution, alors le volume de cet ensemble est au moins $2^{-(n+1)D}$ où D est le nombre de digits qu'il faut pour écrire en machine les $m(n+1)$ entiers a_{ji} et b_j.

Nous allons ici introduire quelques résultats du même type d'approximation.

Théorème I.2.

Soit A une (0-1)-matrice vérifiant l'hypothèse H1. Soit J inclus dans $\mathcal{J}$ et $\beta = 1/2M(n+1)$.

S'il existe un vecteur X de $\mathbb{R}^n$ tel que :

$$\begin{cases} b_j(1-\beta) \le A_J X \le (1+\beta)\, b_J \\ A_{\bar{J}} X \le (1+\beta)\, b_{\bar{J}} \\ X \ge 0 \end{cases}$$

alors il existe un vecteur Y de $\mathbb{R}^n$

$$\begin{cases} A_J Y = b_J \\ A_{\bar{J}} Y \le b_{\bar{J}} \\ Y \ge 0 \end{cases}$$

Démonstration :

1. Posons $X' = \frac{X}{1+\beta}$

Nous avons pour tout j élément de $\mathcal{L}$:

$$\forall j \in \mathcal{L} : \langle a_j, X' \rangle \leq b_j$$

Nous avons d'autre part ; pour tout j élément de J :

$$b_j - \frac{2\beta}{1+\beta} b_j \leq \langle a_j, X' \rangle$$

Le vecteur X' vérifie donc :

$$\begin{cases} b_J \left(1 - \frac{2\beta}{1+\beta}\right) \leq A_J X' \leq b_J \\ A_{\bar{J}} X' \leq b_{\bar{J}} \\ X' \geq 0 \end{cases}$$

2. X' appartient donc à $\mathcal{P}$ et par le théorème de Carathedory [19], [24] , il existe n+1 points $X_1, X_2, \ldots, X_{n+1}$ de $\mathbb{R}^n$ tels que :

- les X_i sont des points extrêmes de $\mathcal{P}$
- $X' = \sum_{i=1}^{n+1} \alpha_i X_i \qquad \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_{n+1} \geq 0$

$$\sum_{i=1}^{n+1} \alpha_i = 1$$

On a donc $\alpha_1 \geq \frac{1}{n+1}$ et d'après l'hypothèse H1 les X_i sont à coordonnées entières. Nous allons montrer que X_1 est tel que :

$$A_J X_1 = b_J$$

Supposons le contraire. Il existe donc $j \in J$ tel que :

$$\langle a_j, X_1 \rangle \leq b_j - 1$$

or,

$$\langle a_j, X' \rangle = \sum_{i=1}^{n+1} \alpha_i \langle a_j, x_i \rangle \leq \sum_{i=2}^{n+1} \alpha_i b_j + \alpha_1 b_j - \alpha_1$$

et

$$\langle a_j, X' \rangle \leq b_j - \alpha_1 \leq b_j - \frac{1}{n+1}$$

On aurait donc :

$$\frac{1}{n+1} \leq \frac{2\beta M}{1+\beta} = \frac{1}{(n+1)\left(1+\frac{1}{2M(n+1)}\right)}$$

$$1 + \frac{1}{2M(n+1)} \leq 1$$

Ceci est impossible et conduit à l'absurdité de notre hypothèse.

Dans le cas plus particulier des matrices totalement unimodulaires, nous pouvons donner un résultat analogue.

Proposition I.3.

Soit A une matrice totalement unimodulaire : c'est-à-dire $\forall J \subset \mathcal{P}$, $\forall I \subset \mathcal{B}$ tels que $|I| = |J|$ alors

$$\det (A_J^I) = 0, +1 \text{ ou } -1.$$

Si il existe un vecteur X de $\mathbb{R}^n$ tel que :

(1) $X \geq 0$; $Ax = b + \varepsilon$ avec $\sum_{j=1}^{m} |\varepsilon_j| < 1$

alors il existe un vecteur Y de $\mathbb{R}^n$ tel que :

(2) $Y \geq 0 \quad AY = b$

Démonstration :

Puisque X vérifie les conditions, il existe une sous-matrice régulière A_1 de A, et une solution de base X_1 telles que :

$$X_1 \geq 0 \qquad X_1 = A_1^{-1}(b+\varepsilon) = A_1^{-1}b + A_1^{-1}\varepsilon .$$

Soit $Y_1 = A_1^{-1}b$; on posera $Y = \tilde{Y}_1$
nous avons $AY = A_1Y_1 = b$
et pour les composantes de Y_1 :

$$X_{1i} = (A_1^{-1}b)_i + (A_1^{-1}\varepsilon)_i$$

et $$|(A^{-1}\varepsilon)_i| \leq \sum_{j=1}^{m} |\varepsilon_j| < 1$$

Puisque $X_1 \geq 0$ et $A_1^{-1}b$ est à coordonnées entières.

On a $(A_1^{-1}b)_i \geq 0$

Ceci implique donc que Y vérifie les conditions (2-), et termine la démonstration.

Chapitre II :

UNE METHODE DIRECTE
POUR LE PROBLEME DE PROGRAMMATION LINEAIRE

II.A. PROBLEME D'EXISTENCE DE SOLUTIONS :

II.A.1. Présentation du problème principal

Dans ce chapitre, notre problème principal (PP) sera défini de la manière suivante :

Soit A une (0-1) - matrice vérifiant H1

Soit $J \subseteq \mathcal{L}$, et $\mathcal{P}_J$ le polyèdre défini par :

Problème (PP) :

$$\mathcal{P}_J \quad \begin{cases} A_J\, X = b_J \\ A_{\bar{J}}\, X \leq b_{\bar{J}} \\ X \geq 0 \end{cases}$$

- question 1 : $\mathcal{P}_J$ est-il vide ou non ?
- question 2 : si $\mathcal{P}_J$ n'est pas vide, trouver $X \in \mathcal{P}_J$

En utilisant le théorème I.2. et en posant $\beta = 1/2\ (n+1)M$, nous pouvons affirmer que la question 1 est équivalente au problème :

Soit $\mathcal{P}_1$ le polyèdre défini par :

$$\begin{cases} b_J\,(1-\beta) \leq A_J\, X \leq b_J(1+\beta) \\ A_{\bar{J}}\, X \leq b_{\bar{J}}(1+\beta) \\ X \geq 0 \end{cases}$$

Question 1' : $\mathcal{P}_1$ est-il vide ou non ?

Remarque : Pour tout X élément de $\mathcal{P}_J$ et pour tout élément de $\mathcal{E}$ on a :

$$0 \leq X_i \leq M_i \leq M$$

II.A.2. Introduction de la fonction entropie

Pour résoudre le problème précédent, nous considérons le programme convexe suivant :

$$(P_C) : z = \underset{X \in \mathcal{P}_J}{\text{Min}} \left\{ \sum_{i=1}^{n} x_i \left(\log \frac{x_i}{M} - 1 \right) \right\}$$

Notons que si $\mathcal{P}_J$ n'est pas vide, z sera négatif ou nul, inversement si $\mathcal{P}_J$ est vide alors $z = +\infty$ [19] [24].

Le problème dual de $(\mathcal{P}_C)$ peut s'écrire :

$$(D_C) : \underset{\substack{\alpha \geq 0, \mu_{\overline{J}} \leq 0 \\ \mu_J}}{\text{Max}} \left\{ \underset{X \geq 0}{\text{Min}} \left\{ \sum_{i=1}^{n} x_i \left(\log \frac{x_i}{M} - 1 \right) - \sum_{j=1}^{m} \mu_j \left(\sum_{i \in L_j} x_i - b_j \right) - \sum_{i=1}^{n} \alpha_i x_i \right\} \right\}$$

Soit f la fonction définie pour X positif :

$$f(X) = \sum_{i=1}^{n} x_i \log \frac{x_i}{M}$$

Des résultats connus pour le problème de l'équilibre chimique montrent que le problème dual (D_C) peut s'écrire :

$$\omega = \underset{\mu_{\overline{J}} \leq 0, \mu_J}{\text{Max}} \left\{ \underset{X \geq 0}{\text{Min}} \left\{ \sum_{i=1}^{n} x_i \left(\text{Log} \frac{x_i}{M} - 1 \right) - \sum_{j=1}^{m} \mu_j \left(\sum_{i \in L_j} x_i - b_j \right) \right\} \right\}$$

(En utilisant les propriétés du sous-gradient de f [11] [1] [24] quand α et μ sont fixés, à l'optimum par rapport à X, on aura $x_i > 0$, pour tout indice i. Ainsi, on est sûr qu'à l'optimum α sera nul).

Nous définirons les fonctions g et h pour μ tel que $\mu_{\bar{J}} \leq 0$ et X tel que $X \geq 0$ par :

$$g(\mu,X) = \sum_{i=1}^{n} x_i(\log \frac{x_i}{M} - 1) - \sum_{j=1}^{m} \mu_j \left(\sum_{i \in L_j} x_i - b_j \right)$$

et

$$h(\mu) = \operatorname{Min}_{X \geq 0} g(\mu, X)$$

Théorème II.1. :

Soit $\bar{X}$ le vecteur qui réalise le minimum de $g(\mu,X)$ le vecteur μ étant fixé tel que $\mu_{\bar{J}} \leq 0$. Posons $\bar{X} = (\bar{x}_i, i \in \mathcal{G})$ et $\lambda_j = e^{\mu j}$ pour j élément de $\mathcal{L}$; on a :

$$(1)\ \bar{x}_i = M \prod_{j \in C_i} \lambda_j \ , \ \sum_{j \in C_i} \mu_j = \log \frac{\bar{x}_i}{M}$$

$$(2)\ h(\mu) = \sum_{j=1}^{m} \mu_j b_j - \sum_{i=1}^{n} \bar{x}_i$$

Démonstration :

On a : $\dfrac{\partial g(\mu,X)}{\partial x_i} = \operatorname{Log} \dfrac{x_i}{M} - \sum_{j \in C_i} \mu_j$

or à l'optimum $\dfrac{\partial g(\mu,\bar{X})}{\partial x_i} = 0$, on en déduit :

$$\bar{x}_i = M\, e^{\sum_{j \in C_i} \mu_j} = M \prod_{j \in C_i} \lambda_j \quad \text{et} \quad \sum_{j \in C_i} \mu_j = \operatorname{Log} \frac{\bar{x}_i}{M}$$

De plus, on a :

$$h(\mu) = g(\mu,\bar{X}) = \sum_{i=1}^{n} \bar{x}_i \left(\sum_{j \in C_i} \mu_j - 1 \right) - \sum_{j=1}^{m} \mu_j \left(\sum_{i \in L_j} \bar{x}_i - b_j \right)$$

et $h(\mu) = \sum_{j=1}^{m} \mu_j b_j - \sum_{i=1}^{n} \bar{x}_i$

Corollaire :

S'il existe μ tel que $\mu_{\overline{J}} \leq 0$ et $h(\mu) > 0$ alors $\mathcal{P}_J$ est vide.

Démonstration :

Si $\mathcal{P}_J$ n'est pas vide, on a pour tout μ tel que $\mu_{\overline{J}} \leq 0$
$h(\mu) \leq \omega = z \leq 0$ (Dualité faible). Ce qui démontre le corollaire.

II.A.3. Description de l'Algorithme de résolution du dual

Le vecteur μ étant donné tel que $\mu_J \leq 0$ si $\overline{X}$ appartient à $\mathcal{P}_J$ le problème est résolu, celui-ci n'est pas vide, sinon il existe une ligne d'indice j_o telle que :

$$\frac{\partial h}{\partial \mu_{j_o}} (\mu) = b_{j_o} - \sum_{i \in L_{j_o}} \overline{x_i} = -\varepsilon_{j_o} \quad \text{avec} \quad \begin{array}{l} |\varepsilon_{j_o}| > 0 \text{ si } j_o \in J \\ \varepsilon_{j_o} > 0 \text{ si } j_o \notin J \end{array}$$

Nous allons améliorer notre solution μ dans la direction de la coordonnée μ_{j_o} : nous cherchons donc μ' tel que :

$$\frac{\partial h}{\partial \mu_{j_o}} (\mu') = 0 \quad \text{donc} \quad \sum_{i \in L_{j_o}} \overline{x'_i} = b_{j_o}$$

nous poserons $\lambda'_{j_o} = \alpha \lambda_{j_o}$, λ'_{j_o} sera tel que :

$$\lambda'_{j_o} \cdot \sum_{i \in L_{j_o}} \frac{\overline{x}'_i}{\lambda'_{j_o}} = b_{j_o} \quad \text{et} \quad \overline{x}'_i = \alpha \overline{x_i} \text{ si } i \in L_{j_o}$$

donc on aura $$\lambda'_{j_o} = \frac{b_{j_o}}{b_{j_o} + \varepsilon_{j_o}} \lambda_{j_o}$$

Ainsi la modification que nous ferons est telle que :

$$\begin{cases} \mu'_{j_o} = \mu_{j_o} + \log\left(\dfrac{b_{j_o}}{b_{j_o}+\varepsilon_{j_o}}\right) \\ \mu'_j = \mu_j \quad \text{si} \quad j \neq j_o \end{cases}$$

et par rapport au vecteur $\bar{X}$ nous aurons :

$$\overline{x'_i} = \frac{b_{j_o}}{b_{j_o}+\varepsilon_{j_o}}\,\overline{x_i} \quad \text{si} \quad i \in L_{j_o}$$

$$\overline{x'_i} = \overline{x_i} \quad \text{si} \quad i \notin L_{j_o}$$

On peut remarquer que si j_o n'appartient pas à J, on aura $\mu'_{j_o} \leq \mu_{j_o}$ et on gardera ainsi la condition $\mu'_{j_o} \leq 0$; et de plus nous avons :

$$h(\mu') = \sum_{j=1}^{m} \mu'_j b_j - \sum_{i=1}^{n} \overline{x'_i} = \sum_{j=1}^{m} \mu_j b_j - b_{j_o} \log\left(1 + \frac{\varepsilon_{j_o}}{b_{j_o}}\right)$$

$$- \sum_{i \in L_{j_o}} \overline{x'_i} - \sum_{i \notin L_{j_o}} \overline{x'_i}$$

$$h(\mu') = \sum_{j=1}^{m} \mu_j b_j - b_{j_o} \log\left(1 + \frac{\varepsilon_{j_o}}{b_{j_o}}\right) - \frac{b_{j_o}}{b_{j_o}+\varepsilon_{j_o}} \sum_{i \in L_{j_o}} \overline{x_i} - \sum_{i \notin L_{j_o}} \overline{x_i}$$

et

$$h(\mu') = h(\mu) + \varepsilon_{j_o} - b_{j_o} \log\left(1 + \frac{\varepsilon_{j_o}}{b_{j_o}}\right)$$

Nous représenterons le vecteur $\lambda = (\lambda_j \; ; \; j \in \mathcal{J})$ et nous poserons $k(\lambda) = h(\mu)$. Nous pouvons décrire l'algorithme.

Algorithme A

Entrée : Soit A une (0-1) - matrice et $b > 0$ vérifiant l'hypothèse H1 ; $J \subseteq \mathcal{L}$: on donne le poly§dre $\mathcal{P}_J$.

Sortie : "Le polyèdre $\mathcal{P}_J$ est vide" ou X appartenant à $\mathcal{P}_J$

Procédure :

Phase (0) : Poser :

$$x_i := M \text{ pour } i = 1, \ldots, n$$
$$k(\lambda) := -nM$$

Phase (1) : Si $k(\lambda) > 0$, terminer : "$\mathcal{P}_J$ est vide".

Phase (2) : chercher s'il existe une contrainte d'indice (j_o) telle que :

$$b_{j_o} - \sum_{i \in L_{j_o}} x_i = -\varepsilon_{j_o} \quad \text{avec} \quad \begin{cases} |\varepsilon_{j_o}| > 0 \text{ si } j_o \in J \\ \varepsilon_{j_o} > 0 \text{ si } j_o \notin J \end{cases}$$

- s'il n'en existe pas : terminer : $X \in \mathcal{P}_J$, $\mathcal{P}_J$ n'est pas vide.

Phase (3) : pour tout $i \in L_{j_o}$ poser :

$$x_i := \frac{b_{j_o}}{b_{j_o} + \varepsilon_{j_o}} x_i$$

Phase (4) : - Poser $k(\lambda) := k(\lambda) + \varepsilon_{j_o} - b_{j_o} \log \left(1 + \frac{\varepsilon_{j_o}}{b_{j_o}}\right)$

- aller en (1).

En toute généralité, cet algorithme converge en un nombre infini d'itérations.

Lemme 1 :

La fonction $a(x) = x - b \log\left(1 + \frac{x}{b}\right) - \frac{x^2}{4b}$ est positive pour $|x|$ plus petit que b, si b est positif.

Démonstration :

Nous avons $a'(x) = 1 - \frac{b}{b+x} - \frac{x}{2b} = \frac{x(b-x)}{2b(b+x)}$.

Nous pouvons décrire le tableau de variations de la fonction a.

x	$-b$		0		b		$+\infty$
a'	‖	−	0	+	0	−	
a	$+\infty$	↘	0	↗	$b(\frac{3}{4} - \log 2)$	↘	$-\infty$

Ainsi le lemme est démontré.

Lemme 2 :

La fonction $c(x) = x - b \log\left(1 + \frac{x}{b}\right)$ est croissante pour x positif.

Démonstration :

Nous pouvons décrire les variations de la fonction :

x	$-b$		0		$+\infty$
c'	‖	−	0	+	
c	$+\infty$	↘	0	↗	$+\infty$

II.A.4. Algorithme de résolution :

Nous désignerons par A_1, l'algorithme déduit de A en remplaçant la phase (2) par :

Phase (2) : chercher s'il existe une contrainte d'indice (j_o) telle que :

$$b_{j_o} - \sum_{i \in L_{j_o}} x_i = -\varepsilon_{j_o} \text{ avec } \begin{cases} |\varepsilon_{j_o}| > \beta b_{j_o} \text{ si } j_o \in J \\ \varepsilon_{j_o} > \beta b_{j_o} \text{ si } j_o \notin J \end{cases}$$

où $\beta = 1/2n\,(M+1)$ (cf. § I-C).

Utilisant les deux lemmes précédents et en oubliant le calcul des logarithmes à chaque itération, on en déduit le théorème suivant :

Théorème II.2. :

L'algorithme A_1 est polynomial en n, m et M. Si de plus, il existe un algorithme polynomial en n qui répond à la question du pas (2) de A, alors l'algorithme est polynomial en n et M.

Démonstration :

Partant de $\lambda = \mathbb{1}_m$ on a $k(\lambda) = -nM$. De plus, le fait qu'au pas (2), si la contrainte (j_o) est trouvée, on ait :

$$k(\lambda) - k(\lambda') = \varepsilon_{j_o} - b_{j_o} \log\left(1 + \frac{\varepsilon_{j_o}}{b_{j_o}}\right)$$

assure une augmentation de la fonction k d'au moins :

$$\frac{\beta^2 b_{j_o}}{4} \geq 1/16\, M^2\, (n+1)^2$$

Ainsi en, au plus, $N = 16n\,(n+1)^2\,M^3$ itérations, l'algorithme A_1 s'arrêtera.

Calculons le nombre d'opérations élémentaires (additions, soustractions, multiplications, divisions) que l'on fait à chaque itération.

Pour chaque ligne, on fait au plus n additions et il y a m lignes, et on fait au plus n modifications des x_i.

Ainsi l'algorithme A_1 est polynomial de l'ordre de $\sigma(n^4 m M^3)$.

Remarque 1 :

Notons que cet algorithme en toute généralité n'est pas polynomial dans la longueur des données, puisque le nombre d'itérations fait intervenir le facteur M et non $\log_2 M$ qui représente celle-ci. Mais pour la plupart des problèmes combinatoires, M est borné par une fonction polynôme en n :

ex : si $0 \leq x_i \leq 1$ alors $M \leq n$

Remarque 2 :

Connaissant le nombre maximum d'itérations N, on peut modifier l'algorithme A de telle façon qu'on ne calcule pas la fonction $k(\lambda)$, mais si au bout de N itérations, on n'a pas conclu que $\mathcal{P}_i$ n'était pas vide, alors on peut affirmer que $\mathcal{P}_J$ est vide.

Nous pouvons ainsi décrire l'algorithme A' :

Algorithme A'

Entrée : A une (0-1) - matrice, m x n, et $b > 0$ vérifiant l'hypothèse H1, $J \subseteq \mathcal{L}$; on donne le polyèdre $\mathcal{P}_J$.

Sortie : "$\mathcal{P}_J$ est vide" ou X appartenant à $\mathcal{P}_i$.

Complexité : $\sigma(n^4 M^3 T)$ où T est la complexité d'un algorithme Z qui répond à la question "$X \in \mathcal{P}_i$?", où $X \in \mathbb{R}^n$

Procédure :

Phase (0) : Poser $x_i := M \quad i = 1, \ldots, n$; $N := 1$
Phase (1) : Si $N > 16\, n\, (n+1)^2\, M^3$: terminer : "$\mathcal{P}_J$ est vide".
Phase (2) : Appliquer Z pour savoir : $X \in \mathcal{P}_i$?
Si $X \in \mathcal{P}_i$: terminer
Si $X \notin \mathcal{P}_i$: soit (j_o) un indice de ligne "violée" par X.

Phase (3) : - pour tout $i \in L_{j_o}$ poser :

$$x_i := \frac{b_{j_o}}{b_{j_o} + \varepsilon_{j_o}} x_i$$

- N := N+1 ; aller en (1).

Remarque 3 :

Cet algorithme particulièrement simple à mettre en oeuvre, est déjà connu, sous une forme apparentée, pour certains problèmes comme les modèles d'ajustement de matrices origines-destinations en théorie du trafic dans l'utilisation des méthodes gravitaires. (méthode de Fratar).

Remarque 4 :

Le vecteur X étant donné, la question de savoir "$X \in \mathcal{P}_i$?" s'appelle le problème de séparation faible [17]. Ainsi plusieurs résultats obtenus dans [17] peuvent se retrouver ici grâce à cette procédure.

Remarque 5 :

Malheureusement, l'algorithme A' fait intervenir au pas (2) le facteur m puisque l'algorithme Z le plus simple que l'on puisse imaginer pour répondre à la question "$X \in \mathcal{P}_i$?" est l'exploration systématique de tout le système de contraintes. Or ce facteur m peut croître exponentiellement avec n : ainsi dans l'exemple du stable dans un graphe parfait où chaque contrainte représente une clique maximale, on voudrait trouver un algorithme qui soit polynomial en n, où n représente le nombre de sommets du graphe. Mais comme ce facteur m n'intervient qu'ici pour le décompte des opérations élémentaires, l'existence d'un algorithme Z, qui peut être classique ou de type oracle, polynomial en n résolvant la question de séparation faible, assure que l'algorithme A' est polynomial en n et M. Cette relation n'est pas sans rappeler le résultat obtenu par Grötschel, Lovasz, Schrijver [17] de l'équivalence en terme de polynomialité entre le problème d'optimisation et le problème de séparation. Par exemple, dans le cas des

graphes parfaits, si on sait résoudre le problème du stable de cardinalité maximum en temps polynomial, et le problème de séparation aussi en temps polynomial alors on sait résoudre le problème du stable de poids maximum en temps polynomial.

II.A.5. Problèmes de précision

Il est bien évident que pour les applications sur ordinateurs, les calculs ne peuvent se faire que sur des valeurs discrètes, on ne peut suivre la théorie qui travaille sur des réels, ou qui calcule des logarithmes et des exponentielles.

Pour le problème des fonctions transcendantes, on sait comment l'éviter grâce à la remarque 2 citée plus haut. Mais il est utile également de remarquer qu'alors que dans les méthodes de Maurras ou Khachijan, il est nécessaire d'utiliser des expressions réelles (du moins dans la théorie, puisqu'on calcule des racines carrées) dans l'algorithme que nous venons de décrire, on utilise uniquement des expressions entières et rationnelles ; nous pourrions donc faire nos calculs en ne conservant que les expressions des numérateurs et dénominateurs de chaque entité, et en faisant les divisions nécessaires pour que ceux-ci ne croissent pas trop.

Toutefois pour éviter ces inconvénients, nous allons étudier ce qu'il se passe si on utilise les formules brutalement.

Propriétés II.3. ;

Pendant tout l'algorithme A nous avons :

$$\forall i \in \mathcal{C} \ ; \ x_i \in]0,M] \qquad (1)$$

$$\text{et } \sum_{j \in C_i} \mu_j \leq 0 \qquad (2)$$

$$\forall j \notin J \ ; \ \mu_j \leq 0 \qquad (3)$$

Démonstration :

Au début de l'algorithme ces propositions sont vraies. Supposons qu'elles soient vraies en cours et montrons qu'elles sont toujours vérifiées à l'itération suivante.

Pour i n'appartenant pas à L_{j_o}, x_i reste inchangé. Pour i appartenant à L_{j_o}, $x_i' = \frac{b_{j_o}}{b_{j_o}+\varepsilon_{j_o}} x_i$ et reste donc positif strictement.

De plus, comme : $\sum_{i\in L_{j_o}} x_i' = b_{j_o}$

on a : $x_i' \leq b_{j_o} \leq M$

Comme nous avons : $\sum_{j\in C_i} \mu_j = \text{Log}\ \frac{x_i}{M} \leq 0$ puisque $x_i \leq M$ et pour j n'appartenant pas à J, nous avons :

$$\mu_j' \leq \mu_j \leq 0$$

ce qui termine la démonstration des trois propriétés.

Théorème II.4.

Avec une précision de l'ordre de $1/12n^3\ M^4$ l'algorithme A reste polynomial avec le même ordre du nombre d'itérations :

Démonstration :

Nous allons supposer qu'il a p positions sur lesquelles on travaille dans le segment [0,1].
Dans l'algorithme A', quand on veut calculer :

$$\alpha = \frac{\varepsilon b_{j_o}}{b_{j_o}+\varepsilon_{j_o}}$$

On prend en fait une position α' telle que :

$$\alpha - \frac{1}{2p} \leq \alpha' \leq \alpha + \frac{1}{2p}$$

On ne calcule pas le vecteur λ' voulu mais un autre $\bar{\lambda}'$ avec :

$$k(\bar{\lambda}') = \sum_{j=1}^{m} \mu_j b_j + b_{j_o} \log \alpha' - \alpha' \sum_{i \in L_{j_o}} x_i - \left(\sum_{i=1}^{n} x_i - \sum_{i \in L_{j_o}} x_i \right)$$

$$k(\bar{\lambda}') = k(\lambda) + b_{j_o} \log \alpha' - (\alpha'-1)\ (b_{j_o}+\varepsilon_{j_o})$$

et donc :

$$k(\bar{\lambda}') \geq k(\lambda) + b_{j_o} \log(\alpha-\tfrac{1}{2p}) - (\alpha-1+\tfrac{1}{2p})(b_{j_o}+\varepsilon_{j_o})$$

et

$$k(\bar{\lambda}') \geq k(\lambda') + b_{j_o} \left\{ \log\ (1 - \frac{1}{2\alpha p}) - \frac{1}{2\alpha p} \right\}$$

or la fonction $f(u) = \log(1-u) + 2u$ est positive pour u compris entre 0 et 1. On en déduit :

$$k(\bar{\lambda}') \geq k(\lambda') - \frac{3}{2} \frac{b_{j_o}}{\alpha p}$$

et

$$k(\bar{\lambda}') \geq k(\lambda) + \frac{1}{16(n+1)^2 M^2} - \frac{3}{2} \frac{b_{j_o}+\varepsilon_{j_o}}{p}$$

or $b_{j_o}+\varepsilon_{j_o} = \sum_{i \in L_{j_o}} x_i \leq nM$

Si on prend $p = 48\ n(n+1)^2\ M^3$

$$k(\bar{\lambda}') \geq k(\lambda) + \frac{1}{48(n+1)^2 M^2}$$

En utilisant la même démonstration que précédemment, on en déduit que l'algorithme A' reste polynomial dans le même ordre $O(n^4 T M^3)$, où T est la complexité d'un algorithme Z pour répondre au problème de la séparation faible.

II.B. RECHERCHE D'UNE SOLUTION REALISABLE DU PROBLEME (PP)

Nous étudierons ici seulement le cas où le nombre de contraintes m reste polynomial en le nombre de variables n.

Pour répondre à la deuxième question du problème (PP) ; si $\mathcal{P}_J$ n'est pas vide, soit X' le vecteur de $\mathcal{P}_i$ trouvé par l'algorithme A', d'après le théorème I.2., nous savons qu'il faut trouver (n+1) points extrêmes P_e^i de $\mathcal{P}$ tels que :

$$X = \frac{X'}{1+\beta} = \sum_{i=1}^{n+1} \alpha_i \, P_e^i$$

avec $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_{n+1} \geq 0$ et $\sum_{i=1}^{n+1} \alpha_i = 1$

II.B.1. Description de l'algorithme B

Nous allons décrire un algorithme polynomial en n et m qui partant d'un point X non nul de $\mathcal{P}$ trouve un point extrême P_X de $\mathcal{P}$. Notons que cette procédure a été introduite pour la première fois par Maurras [28] et décrite dans [3].

Algorithme B (variable X ; résultat P_X) ;

Phase (1) : Nous noterons

$$\mathcal{C}_p = \{i \in \mathcal{C} / X_i > o\}$$
$$\mathcal{L}_e = \{j \in \mathcal{L} / \langle a_j, X\rangle = b_j\}$$

$$A_e = (a_{ji})_{j\in\mathcal{L}_e, i\in\mathcal{C}_p} \qquad b_e = (b_j)_{j\in\mathcal{L}_e}$$

$$A_i = (a_{ji})_{j\notin\mathcal{L}_e, i\in\mathcal{C}_p} \qquad b_i = (b_j)_{j\notin\mathcal{L}_e}$$

Le vecteur X_p sera le vecteur de $\mathbb{R}^{|\mathcal{C}_p|}$ défini par :

$$X_p = (X_i)_{i\in\mathcal{C}_p}$$

X_p est tel que :

$$\begin{cases} A_e\, X_p = b_e \\ A_i\, X_p < b_i \\ X_p > 0 \end{cases}$$

Phase (2) : Si $\mathcal{L}_e = \emptyset$ soit

$Y := \alpha X_p$ avec

$$\alpha = \text{Min} \left\{ \frac{b_j}{\langle a_j, X_p \rangle} \;/\; j \in \mathcal{L} \text{ et } \langle a_j, X_p \rangle \neq 0 \right\}$$

on pose : $X_p := Y$ et aller en (1).

Phase (3) :

On résoud le système linéaire :

$$A_e X = b_e$$

On peut le faire par la méthode de Gauss. Dans cette méthode classique, on trouve une base (un ensemble de colonnes I tel que A^I soit régulière) telle que A^I soit carrée de taille $|\mathcal{L}_e|$. La solution qu'on prend est :

$$x^I = (A^I)^{-1}\, b_e,\; x^{\bar{I}} = 0 ;$$

Si $A^I = A_e$, l'algorithme est terminé, on a une solution de base réalisable.

Notons cependant que cette méthode n'est pas polynomiale comme le montre MAURRAS [28] [3]. J. EDMONDS en suggèrant une amélioration algorithmique a pu rendre cette méthode polynomiale [6].

Phase (4) :

<u>1er cas</u> : Si X'_p est réalisable, alors X'_p est un point extrême de $\mathcal{P}$. On pose :

$$P_X = X'_p$$

terminer.

2e cas : Il existe un ensemble $\mathcal{C}_n$ d'indices de $\mathcal{C}_p$ tel que :

$$X'_{pi} < 0 \qquad \forall i \in \mathcal{C}_n$$

et (ou) il existe un ensemble $\mathcal{L}_n$ d'indices de $\mathcal{L}_i$ tel que :

$$\langle a_j, X'_p \rangle > b_j \qquad \forall j \in \mathcal{L}_n$$

Posons : $Y := \alpha X'_p + (1-\alpha) X_p$
avec α tel que : $0 \leq \alpha \leq 1$ et

$$\alpha = \text{Min} \left\{ \text{Min} \left\{ \frac{X_{pi}}{X_{pi}-X'_{pi}} / i \in \mathcal{C}_n \right\}, \text{Min} \left\{ \frac{b_j - \langle a_j, X_p \rangle}{\langle a_j, X'_p - X_p \rangle} / j \in \mathcal{L}_n \right\} \right\}$$

(On pourra remarquer qu'on ne divise jamais par 0).
Ainsi Y sera tel que :

$$\begin{cases} A_e \; Y = b_e \\ A_i \; Y \leq b_i \\ \quad\; Y \geq 0 \end{cases}$$

Poser X := Y et aller à la phase (1).

Théorème II.5.

L'algorithme B est polynomial en n et m.

Démonstration :

A chaque fois que l'algorithme passe dans la phase (4) la valeur α est choisie de telle façon que :

- soit le cardinal de $\mathcal{C}_p$ diminue d'une unité ;
- soit le cardinal de $\mathcal{L}_e^p$ augmente d'une unité.

Puisqu'au départ $|\mathcal{C}_p| \leq n$
et si $|\mathcal{L}_e| = n$ on est sûr que la solution trouvée à la phase (3) appartiendra à $\mathcal{P}_J$; on peut conclure qu'en, au plus, 2n passages dans la phase (4), l'algorithme est terminé.

Remarque 1 :

Dans le cas où on veut faire disparaître le facteur m dans le calcul de la complexité de l'algorithme, on peut montrer qu'avec l'existence de l'algorithme Z répondant au problème de la séparation, on a un algorithme polynomial en n et T, où T est la complexité de Z. Nous ne le ferons pas ici pour tenter de garder une certaine clarté dans l'exposé.

Connaissant cet algorithme, comment à partir du vecteur X' appartenant à $\mathcal{P}_i$, répondre à la deuxième question du problème (PP)?.

III.B.2. Description de l'algorithme de résolution :

A partir de l'algorithme B, dont on peut constater que le point extrême rendu vérifie les mêmes conditions d'égalité que le point donné en entrée, il est facile d'imaginer un algorithme que nous appellerons algorithme C, qui en n+1 appels à l'algorithme B, construit une combinaison linéaire de points extrêmes de $\mathcal{P}$, et qui est égale au point de départ intérieur à $\mathcal{P}$.

II.C. LE PROBLEME DE PROGRAMMATION LINEAIRE

Les données répondant aux hypothèses faites au chapitre I, on veut résoudre le problème de Programmation Linéaire suivant :

$$(PL) : \begin{cases} AX \leq b \\ X \geq 0 \\ \text{Max } \langle c, X \rangle \end{cases}$$

II.C.1. Problème de cardinalité maximum

Supposons ici que le vecteur c est tel que :

$c_i \in \{0,1\}$ pour tout $i \in \mathcal{C}$.

Le problème de résoudre PL dans le cas de cardinalité maximum est certainement le plus fréquent en Optimisation Combinatoire.

Nous appliquerons l'algorithme suivant :

Algorithme D :

Phase (0) : Poser $u := nM$ où $M = \text{Max} \mid b_j \mid$

Phase (1) : Appliquer l'algorithme A avec :

$$\mathcal{P}_J = \begin{cases} \langle c, X \rangle = u \\ AX \leq b \\ X \geq 0 \end{cases}$$

Phase (2) : Si $\mathcal{P}_J = \emptyset$ poser $u := u-1$ aller en (1)
Sinon aller en (3).

Phase (3) : Appliquer l'algorithme C pour trouver un vecteur $\overline{X}$ solution optimale de base du programme linéaire PL.

Terminer.

Théorème II.7. :

L'algorithme D est polynomial en n, m et M. Si les algorithmes A et C sont polynomiaux en n alors D est polynomial en n et M.

Démonstration :

Nous savons que pour tout $X \in \mathcal{P}$ $x_i \leq M$; d'où $0 \leq \langle c, X \rangle \leq \sum_{i=1}^{n} x_i \leq nM$

A chaque itération, n décroit de une unité, en au plus nM itérations l'algorithme D se termine. La valeur m n'intervient dans les calculs que dans les algorithmes A et C, les théorèmes II.4., II.5. et II.6. entraînent donc ce théorème.

II.C.2. Idées d'algorithmes pour le problème de poids maximum

Si les C_i ne sont pas à valeurs dans {0,1}, on ne peut plus se servir de la procédure utilisée plus haut. Toutefois, on pourrait montrer qu'en décomposant chaque C_i en unaire et dupliquant les variables x_i et autant de fois qu'il faut le système, que cette procédure est polynomiale en $M' = \max |c_i|$.

D'autre part, la façon dont on a posé le problème initial nous permet d'imaginer facilement un algorithme Primal-Dual pour résoudre le problème de poids maximum.

Nous n'irons pas plus loin dans l'argumentation de ces algorithmes.

Chapitre III . Une méthode de pénalité pour une classe de programmes linéaires .

Dans le cas où les données vérifient les hypothèses du chapitre I , on s'intéresse dans ce chapitre , toujours au problème de Programmation linéaire

$$(PL) \quad \begin{cases} A\,x \leq b & x \in R^n \\ x \geq 0 \\ \text{Max} \ \langle c,x \rangle = z_0 \end{cases}$$

où A est une matrice en (0,1) vérifiant l'hypothèse H1 et b et c sont entiers ce qui implique entre autres que z_0 est entier .

Nous allons introduire une pénalité dans la fonction objective de forme entropie avec un paramètre qu'on choisira assez petit pour que la fonction objective primale reste prépondérante vis-à-vis de cette pénalité .

III,A. Programme convexe associé au Programme linéaire initial:

III,A,1. Introduction :

Nous désignerons par (P_p) le programme convexe suivant :

$$\begin{cases} A\,x \leq b \\ x \geq 0 \\ \text{Min} \left\{ p \left(\sum_{i=1}^{n} x_i \text{ Log } x_i/M - \sum_{i=1}^{n} x_i \right) - \langle c,x \rangle \right\} \end{cases}$$

où $M = \text{Max}_j b_j$ et p un paramètre réel positif strictement . Nous noterons , pour $x \in \mathcal{P}$:

$$l_p(x) = p\left(\sum_{i=1}^{n} x_i \text{ Log } x_i/M - \sum_{i=1}^{n} x_i \right) - \langle c,x \rangle$$

III,A,2. Présentation de la méthode de résolution:

L'algorithme de résolution que nous proposons dans ce chapitre est de type dual .

Nous étudierons le dual du programme (P_p) et chercherons à minimiser la fonction objectif du dual , h(y) où y désigne le vecteur variable dual de (P_p).

A une itération (i) , on aura une solution réalisable y^i du dual du programme (P_p) ; on montrera que y^i est aussi solution réalisable du dual de (PL). D'autre part , on pourra associer à y^i un vecteur x^i de l'espace primal R^n .

Si (y^i,x^i) vérifient les conditions d'écarts complémentaires associées à (PL) , x^i est solution de celui-ci . On ne pourra , sans doute , jamais vérifier ces conditions directement , mais grâce au théorème I.2 du chapitre I. on pourra affirmer que si les conditions d'écarts complémentaires sont vérifiées à une certaine précision par le couple (y^i,x^i) , il existe un vecteur x primal réalisable tel que (y^i,x) vérifie celles-ci et donc x est solution de (PL). Dans le cas contraire , en cherchant à améliorer le dual ; on obtiendra une nouvelle solution y^{i+1} et on sera assuré que l'accroissement $h(y^{i+1}) - h(y^i)$ sera suffisamment important pour garantir sous certaines hypothèses supplémentaires sur b et c un algorithme polynomial par rapport à n et m .

Théorème III.1.:

La fonction $l_p(x)$ pour $x \in \mathcal{P}$ vérifie les propriétés suivantes :

(a) $l_p(x) \leq - \langle c,x \rangle$

(b) $l_p(x)$ est convexe .

Le problème dual (D_p) du problème (P_p) s'écrit :

$$(D_p) : \max_{y\geq 0, q\geq 0} \min_{x} \left\{ l_p(x) + \sum_{j=1}^{m} y_j \Big(\sum_{i\in L_j} x_i - b_j \Big) + \sum_{i=1}^{m} q_i x_i \right\}$$

Les propriétés de la fonction $l_p(x)$ [1] [12] font que, à l'optimum on aura $\bar{x}_i > 0$ pour tout i de $\mathcal{C}$ et donc $\bar{q}_i = 0$ pour tout $i \in \mathcal{C}$. (D_p) peut donc s'écrire :

$$(D_p) : \max_{y\geq 0} \min_{x\geq 0} \left\{ l_p(x) + \sum_{i=1}^{m} y_j \Big(\sum_{i\in L_j} x_i - b_j \Big) \right\}$$

III.B. Etude du problème dual (D_p) :

Les conditions de Kuhn et Tucker [24][15] nous assurent que $\bar{x}$ et $\bar{y}$ sont solutions optima de (P_p) et (D_p) si :

$$\begin{cases} \bar{x} \geq 0 \ , \ \bar{y} \geq 0 & (1) \\ \bar{y}_j \left(\sum_{i \in L_j} \bar{x}_i - b_j \right) = 0 \quad \forall j \in \mathcal{L} & (2) \\ \frac{\partial l_p}{\partial x_i}(\bar{x}) + \sum_{j \in C_i} \bar{y}_j = 0 \quad \forall i \in \mathcal{C} & (3) \end{cases}$$

Les conditions (3) ne font qu'exprimer que si $y \geq 0$ est fixé, le minimum de la fonction à optimiser en $x \geq 0$ est vérifié pour $\bar{x}$ tel que :

$$\frac{\partial l_p}{\partial x_i}(\bar{x}) + \sum_{j \in C_i} y_j = 0 \quad \forall i \in \mathcal{C}$$

Soit pour $y \geq 0$ et $x \geq 0$:

$$g(y,x) = l_p(x) + \sum_{j=1}^{m} y_j \left(\sum_{i \in L_j} x_i - b_j \right)$$

$$h(y) = \min_{x \geq 0} g(y,x)$$

Théorème III.2 :

Soit $y \geq 0$ et $\bar{x} \geq 0$ tel que $h(y) = g(y,\bar{x})$, on a :

$$\forall i \in \mathcal{C} \quad \bar{x}_i = M \exp(c^i/p) \prod_{j \in C_i} t_j \quad \text{avec } t_j = \exp(-y_j/p) \qquad \forall j \in \mathcal{L}$$

$$h(y) = - \sum_{j=1}^{m} y_j b_j - p \sum_{i=1}^{n} \bar{x}_i$$

Démonstration :

$\bar{x}$ est tel que $\frac{\partial g}{\partial x_i}(y,\bar{x}) = 0$ donc

$p \left(\text{Log } \bar{x}_i/M + 1 - 1 \right) - c^i + \sum_{j \in C_i} y_j = 0$ ceci implique que

$$\bar{x}_i = M \exp\left((c^i - \sum_{j \in C_i} y_j)/p \right) = M \exp(c^i/p) \sum_{j \in C_i} t_j$$

D'autre part :

$$h(y) = \sum_{i=1}^{n} \bar{x}_i \left(-p + c^i - \sum_{j \in C_i} y_j + p\right) - p \sum_{i=1}^{n} \bar{x}_i - \sum_{i=1}^{n} c^i \bar{x}_i + \sum_{j=1}^{m} y_j \left(\sum_{i \in L_j} \bar{x}_i - b_j \right)$$

$$h(y) = - \langle y, b \rangle - p \langle 1, \bar{x} \rangle$$

Théorème III.3:

Soit $y \geq 0$;

(1) $\forall i \in \mathcal{C}$ $\bar{x}_i \leq M \iff$ y est dual réalisable pour (DL) le problème dual linéaire de (PL) .

(2) $h(y) \leq -z_0$.

(3) si y est dual réalisable alors :

$$- \langle y, b \rangle - p \, n \, M \leq h(y) \leq - \langle y, b \rangle \ .$$

Démonstration :

(1) évident .

(2) Soit y un vecteur réalisable du dual linéaire de (PL) , on a :

$$\bar{x}_i = M \exp\left((c^i - \langle a^i, y \rangle)/p \right)$$

et comme y est dual réalisable :

$$c^i - \langle a^i, y \rangle \leq 0 \qquad \text{et donc}$$

$$0 \leq \bar{x}_i \leq M \quad \text{pour tout } i \in \mathcal{C}$$

et $\langle 1, \bar{x} \rangle \leq n M$ d'où $h(y) = -\langle y, b \rangle - p \langle 1, \bar{x} \rangle \geq - \langle y, b \rangle - p \, n \, M$

(3) Pour tout $y \geq 0$ et $x \in \mathcal{P}$, on a :

$$\langle y, \ (A x - b) \rangle \leq 0$$

donc pour tout $x \in \mathcal{P}$: $l_p(x) \geq g(y,x)$ et

$$h(y) = \min_{x \geq 0} g(y,x) \leq \min_{x \in \mathcal{P}} g(y,x) \leq \min_{x \in \mathcal{P}} l_p(x)$$

Soit $\bar{x}$ solution optimale du problème primal (PL) :

$$h(y) \leq \min_{x \in \mathcal{P}} l_p(x) \leq l_p(\bar{x}) \leq - z_0$$

D'où $\quad y \geq 0 \Rightarrow h(y) \leq - z_0$

III.D. Théorème et algorithme d'approximation :

III.D.1. Théorème d'approximation :

De la même manière que dans les chapitres précédents nous allons décrire un théorème d'approximation .

Théorème III.4. :

Soit $y \geq 0$ un vecteur dual réalisable avec :

$$\frac{\delta h}{\delta y_j}(y) = \langle a_j, \bar{x} \rangle - b_j = \varepsilon_j$$

tel qu'il existe $s_1 > 0$ tel que

$$|\varepsilon_j| \geq s_1 b_j \Rightarrow \varepsilon_j \leq - s_1 b_j \text{ et } y_j = 0$$

alors

$$\langle y,b \rangle - 2 s_1 \langle y,b \rangle - p n M \leq z_0 \leq \langle y,b \rangle$$

Démonstration :

Soit $\bar{y}$ une solution optimale du dual linéaire (DL) ; comme la fonction h est concave [24] [15] , on a :

$$h(\bar{y}) \leq h(y) + \langle \bar{y} - y , \delta h(y) \rangle$$

où $\delta h(y) = \left(\frac{\delta h}{\delta y_j}(y) ; j = 1,...m \right)$ est le vecteur gradient de la fonction h au point y . On peut écrire :

$$h(\bar{y}) \leq h(y) - \langle y , \delta h(y) \rangle + \langle \bar{y} , \delta h(y) \rangle$$

a. Regardons le terme $- \langle y, \delta h(y) \rangle$

1. si $\left| \frac{\delta h}{\delta y_j}(y) \right| \leq s_1 b_j$ alors $- y_j \frac{\delta h}{\delta y_j}(y) \leq s_1 y_j b_j$

2. si $\frac{\delta h}{\delta y_j}(y) < -s_1 b_j$ alors $y_j = 0$ donc
$- \langle y, \delta h(y) \rangle \leq s_1 \langle y,b \rangle$

b. Regardons le terme $+ \langle \bar{y}, \delta h(y) \rangle$

1. $\left| \frac{\delta h}{\delta y_j}(y) \right| \leq s_1 b_j \Rightarrow \bar{y}_j \frac{\delta h}{\delta y_j}(y) \leq s_1 \bar{y}_j b_j$

2. $\frac{\delta h}{\delta y_j}(y) \leq -s_1 b_j \Rightarrow \bar{y}_j \frac{\delta h}{\delta y_j}(y) \leq 0$

On peut écrire : $\langle \bar{y} \ , \ \delta h(y) \rangle \leq s_1 \langle \bar{y} , b \rangle = s_1 \ z_o$

On en déduit :

$$- z_o - p \ n \ M \leq h(\bar{y}) \leq h(y) + s_1 \langle y,b \rangle + s_1 \ z_o$$

D'où:

$$- h(y) - s_1 \langle y,b \rangle - p \ n \ M \leq (\ 1 + s_1 \ ) \ z_o$$

$$- \frac{h(y)}{1 + s_1} - \frac{s_1}{1 + s_1} \langle y,b \rangle - \frac{p \ n \ M}{1 + s_1} \leq z_o$$

$$- h(y) + \frac{s_1}{1 + s_1} (\ h(y) - \langle y,b \rangle \ ) - \frac{p \ n \ M}{1 + s_1} \leq z_o$$

or $h(y) = - \langle y,b \rangle - p \ \langle \bar{x},1 \rangle$ et d'après le théorème précedent :

$$\langle y,b \rangle - \frac{2s_1}{1 + s_1} \langle y,b \rangle - p \ n \ M \leq z_o \leq \langle y,b \rangle$$

comme $s_1 > 0$ on a le théorème .

III.D.2. Changement de solution duale :

Supposons que le vecteur $y \geq 0$ dual réalisable ne vérifie pas les conditions du théorème précédent où $s_1 > 0$ est donné ; deux cas sont alors possibles :

1°_cas_ Le vecteur $\bar{x}$ n'est pas primal réalisable à la précision demandée il existe donc une contrainte d'indice $j_o \in \mathcal{L}$ telle que :

$$\langle a_{j_o} , \bar{x} \rangle - b_{j_o} = \varepsilon_{j_o} \quad \text{avec} \quad \varepsilon_{j_o} > s_1 \ b_{j_o}$$

on va poser :

$$t'_{j_o} = \frac{b_{j_o}}{b_{j_o} + \varepsilon_{j_o}} \ t_{j_o} \quad \text{ou}$$

$$\begin{cases} y'_{j_o} = y_{j_o} + p \ \text{Log} \ (\ 1 + \dfrac{\varepsilon_{j_o}}{b_{j_o}} \ ) > y_{j_o} \\ y'_j = y_j \qquad \text{si } j \neq j_o \end{cases}$$

Soit $y' = (y'_j \ , \ j = 1,...m \ )$. Il est facile de voir que $y' \geq 0$ puisque $y'_{j_o} > y_{j_o}$, et d'autre part :

$$\begin{cases} \bar{x}'_i = \dfrac{b_{j_o}}{b_{j_o} + \varepsilon_{j_o}} \bar{x}_i & \text{pour } i \in L_{j_o} \\ \bar{x}'_i = \bar{x}_i & \text{pour } i \notin L_{j_o} \end{cases}$$

Puisque y est dual réalisable , on a $\bar{x}_i \leq M$ et donc $\bar{x}'_i \leq M$ pour tout i de $\mathcal{C}$ et ainsi on conclut que y' est aussi dual réalisable . D'autre part , on a

$$h(y') = h(y) + p \left(\varepsilon_{j_o} - b_{j_o} \operatorname{Log} \left(1 + \frac{\varepsilon_{j_o}}{b_{j_o}} \right) \right)$$

2°_cas_ Le vecteur $\bar{x}$ est primal réalisable mais ne vérifie pas avec y les écarts complémentaires à la précision demandée . Il existe $j_o \in \mathcal{L}$ tel que:

$$\langle a_j , \bar{x} \rangle - b_{j_o} = \varepsilon_{j_o} \quad \text{avec } \varepsilon_{j_o} < -\varepsilon_1 b_{j_o} \quad \text{et } y_{j_o} \neq 0$$

(a). On a $y'_{j_o} = y_{j_o} + p \operatorname{Log} \left(1 + \dfrac{\varepsilon_{j_o}}{b_{j_o}} \right) \geq 0$

Il est facile de montrer que y' est toujours dual réalisable et que l'augmentation de la fonction objective reste la même que dans le cas précédent.

(b) Si $y_{j_o} + p \operatorname{Log} \left(1 + \dfrac{\varepsilon_{j_o}}{b_{j_o}} \right) < 0$

c'est-à-dire $t_{j_o} > \dfrac{b_{j_o} + \varepsilon_{j_o}}{b_{j_o}}$ avec $t_{j_o} = \exp(-y_{j_o} / p)$

on posera alors $t'_{j_o} = 1$ soit $y'_{j_o} = 0$.

On aura dans ce cas :

$$\begin{cases} \bar{x}'_i = \bar{x}_i / t_{j_o} & \text{si } i \in L_{j_o} \\ \bar{x}'_i = \bar{x}_i & \text{sinon} \end{cases}$$

Comme $\sum_{i \in L_{j_o}} \bar{x}'_i = \left(\sum_{i \in L_{j_o}} \bar{x}_i \right) / t_{j_o} = \dfrac{b_{j_o} + \varepsilon_{j_o}}{t_{j_o}} \leq b_{j_o} \leq M$, on aura $\bar{x}'_i \leq M$ pour $i \in L_{j_o}$ et ainsi on conclut que y' sera dual réalisable . De plus:

$$h(y') = - \langle y, b \rangle + y_{j_o} b_{j_o} - p \left(\left(\sum_{i \in L_{j_o}} \bar{x}_i \right) / t_{j_o} + \sum_{i \notin L_{j_o}} \bar{x}_i \right)$$

$$h(y') = h(y) + y_{j_o} b_{j_o} - p \left(b_{j_o} + \varepsilon_{j_o} \right)\left(t_{j_o}^{-1} - 1 \right)$$

or $\left(b_{j_o} + \varepsilon_{j_o} \right) / t_{j_o} < b_{j_o}$ et donc

$$h(y') \geq h(y) + y_{j_o} b_{j_o} + p \left(b_{j_o} + \varepsilon_{j_o} \right) - p\, b_{j_o}$$

et finalement on en déduit la formule bien connue :

$$h(y') \geq h(y) + y_{j_o} b_{j_o} + \varepsilon_{j_o} p$$

Nous pouvons maintenantintroduire l'algorithme suivant :

Algorithme F :

Entrée : Le programme linéaire suivant :

$$(PL)\begin{cases} A\,x \leq b \\ x \geq 0 \\ \max \ \langle c, x \rangle \end{cases} \quad \text{où } A, b, c \text{ vérifient l'hypothèse H1.}$$

s_1 et p tous positifs strictement paramétres réels .

Soit $M = \text{Max } b_j$ et $M' = \text{Max } c^i$

Sortie : Un vecteur $\bar{x} \geq 0$ et un vecteur $y \geq 0$ dual réalisable pour (PL)

tels que :

$$A\,\bar{x} \leq \left(1 + s_1 \right) b$$

$\forall j \in \mathcal{L} \ \langle a_j, \bar{x} \rangle - b_j < - s_1 b_j \Rightarrow y_j = 0$; $\forall i \in \mathcal{C} \ \bar{x}_i = M \exp\left(\left(c^i - \langle a^i, y \rangle \right)/p \right)$

Complèxité : $o(n^2 M M' m / p s_1^2)$;

Procédure :

Phase (0) : - Chercher un ensemble $J_o \subseteq \mathcal{L}$ avec $|J_o| = k \leq n$ tel que :

$$\forall i \in \mathcal{C}, \exists j \in J_o \ / \ i \in L_j$$

- Poser $\begin{cases} y_j = M' = \text{Max} \left(c^i / i \in \mathcal{C} \right) & \text{si } j \in J_o \\ y_j = 0 \text{ sinon ;} \end{cases}$

- Poser $t_j = \exp\left(- y_j / p \right)$ pour tout $j \in J_o$

$$\bar{x}_i = M \exp\left(c^i / p \right) \prod_{j \in J_o \cap C_i} t_j$$

Phase (1) : - Chercher s'il existe un indice j_0 tel que :

$$\langle a_{j_0}, \bar{x} \rangle - b_{j_0} = \varepsilon_{j_0} \text{ avec } (\varepsilon_{j_0} > s_1 b_{j_0}) \text{ ou } (\varepsilon_{j_0} < - s_1 b_{j_0} \text{ et } t_{j_0} < 1)$$

- S'il n'en existe pas aller en (4) sinon aller en (2) .

Phase (2) : - Faire les changements :

Forme 1 : si $\varepsilon_{j_0} > s_1 b_{j_0}$

- Poser $t'_{j_0} := b_{j_0} t_{j_0} / (b_{j_0} + \varepsilon_{j_0})$; $J_0 := J_0 \cup \{ j_0 \}$;

Forme 2 a : si $\varepsilon_{j_0} < - s_1 b_{j_0}$ et $t_{j_0} < (b_{j_0} + \varepsilon_{j_0}) / b_{j_0}$

- Poser $t'_{j_0} := b_{j_0} t_{j_0} / (b_{j_0} + \varepsilon_{j_0})$;

Forme 2 b : si $\varepsilon_{j_0} < - s_1 b_{j_0}$ et $t_{j_0} \geq (b_{j_0} + \varepsilon_{j_0}) / b_{j_0}$

- Poser $t'_{j_0} := .1$; $J_0 := J_0 \setminus \{ j_0 \}$;

Phase (3) : - Poser $\bar{x}_i := (t'_{j_0} / t_{j_0}) \bar{x}_i$ pour $i \in L_{j_0}$

$t_{j_0} := t'_{j_0}$;

- Aller en (1) .

Phase (4) : - Poser $y_j := - p \text{ Log } t_j$ pour $j \in J_0$

$y_j := 0$ pour $j \notin J_0$

- Terminer .

Démonstration de la validité :

Nous désignerons par N le nombre de passages dans la phase (2) , N se décompose en :

N_1 = nombre de changements de la forme 1

N_2 = nombre de changements de la forme 2 a

N_3 = nombre de changements de la forme 2 b .

On a bien évidemment $N = N_1 + N_2 + N_3$. D'autre part , pour faire un changement de la forme 2 il faut $t_{j_0} < 1$ donc il faut soit qu'on ait posé

$t_{j_o} < 1$ et $j_o \in J_o$ à la phase (2) , soit qu'on ait posé $t_{j_o} = 1$ au départ et que l'on ait augmenté celui-ci par la suite ; on en déduit :

$$N_2 + N_3 \leq N_1 + |J_o| \leq N_1 + n$$

D'où $$N \leq 2 N_1 + n$$

Soit y_d le vecteur initial choisi dans la phase (0) . y_d est dual réalisable et ainsi par récurrence le vecteur y sera toujours dual réalisable . De plus, on a :

$$h(y) = -\langle y, b \rangle - p \langle \bar{x}, 1 \rangle \geq - n M M' - p n M$$

On constate également qu'à chaque passage dans la forme 1 la fonction h(y) augmente d'au moins $(p\ s_1^2\ b_{j_o})/4$ soit d'au moins $p\ s_1^2 / 4$.

Comme $h(y) \leq l_p(0) = 0$, on conclut que :

$$p\ s_1^4\ N_1 / 4 \leq n M (M' + p)$$

$$N_1 \leq \frac{4 n M (M' + p)}{p\ s_1^2}$$

D'autre part , à chaque itération , si j_o a été trouvé , on fait n additions au plus pour calculer la somme des $\bar{x}_i$ et on fait n changements de ces variables. Si on connait un algorithme qui trouve une contrainte violée à plus de $s_1\ b_{j_o}$ c'est-à-dire qui soit de la forme 1 , et soit T une borne de l'effort de cet algorithme pour répondre à la question ; puisque Card $\{ y_j / y_j > 0 \} \leq N_1 + n$, et est donc polynomial en n , M , M' , on peut calculer la complèxité de l'algorithme F en étant de l'ordre de :

$$o (n M M' T / p\ s_1^2)$$

Remarque 1 :

Si on connait un algorithme tel que le facteur m n'intervient pas dans la valeur T alors l'algorithme F sera polynomial sans le facteur m . Sinon , l'al-

gorithme le plus simple est de regarder contrainte par contrainte et sur chacune de ces contraintes il y a n additions à faire donc T = n m ce qui donne la complèxité de l'algorithme telle qu'elle a été donnée plus haut .

Remarque_2_:

La différence essentielle entre les deux algorithmes A (présenté au chapitre précédent) et F est que A ne travaille que sur des rationnels , celui-ci se rapproche beaucoup de l'Analyse Numérique car il fait appel à des fonctions transcendantales telles que les exponentielles et logarithmes . Toutefois , celles-ci ne sont appelées que dans les phases initiales et finales pour pouvoir reconnaître le vecteur y .

III.E. Algorithme de résolution :

Nous pouvons alors décrire l'algorithme suivant :

Algorithme G

Entrée_: Le programme linéaire :

$$(PL)\begin{cases} A\,x \leq b \\ x \geq 0 \\ \text{Max } \langle c,x\rangle \end{cases}$$

où A , b , c vérifient les hypothèses du chapitre précédent .

Sortie_: Une solution optimale entière x^* de (PL) .

Complèxité_:_ $o\left(n^5\, m\, M^4\, M' \text{ Log } (n\, M) \right)$.

Procédure :

Phase (0) : - Appliquer l' algorithme F à (PL) avec les valeurs des paramètres

$$s_1 = \left(4\,(n+1)\,M \right)^{-1}$$

$$p = \left(n\,M \text{ Log}(4\,n\,(n+1)\,M^2) \right)^{-1}$$

- Soit $\bar{x}$ et y les deux vecteurs obtenus :

Phase (1) : - Poser $J := \{ j \in \mathcal{J} / \; y_j > 0 \}$

$$I := \{ i \in \mathcal{C} / \; \langle a^i, y\rangle - c^i > 1 / n\,M \}$$

- Soit $\mathcal{P}_J^I$ le polyèdre défini par :

$$\begin{cases} A_J^I x = b_J \\ A_{\bar{J}}^I x \leq b_{\bar{J}} \\ x \geq 0 \end{cases}$$

Phase (2) : - Appliquer l'algorithme C avec pour entrée $x' = \bar{x}_I$ et soit x_1 la solution retournée ;

- Poser $x^* := \tilde{x}_1$, x^* est optimal . Terminer .

Démonstration de la validité :

En prenant I et p comme indiqué , nous allons montrer que pour tout i n'étant pas dans I on a :

$$\bar{x}_i \leq 1 / 4n (n+1) M$$

En effet :

$$\bar{x}_i = M \exp ((c^i - \langle a^i, y \rangle)/p)$$

$$(c^i - \langle a^i, y \rangle)/p = nM \text{ Log } (4n (n+1) M^2) (c^i - \langle a^i, y \rangle)$$

$$(c^i - \langle a^i, y \rangle)/p \leq - \text{Log } (4n (n+1) M^2)$$

Et ainsi : $\forall i \notin I ; \bar{x}_i \leq 1 / 4n (n+1) M$

D'autre part puisque $\bar{x}$ est tel que :

$$\begin{cases} A \bar{x} \leq b (1 + s_1) = b (1 + \frac{1}{4^{---}(n+1)^{-}M}) \\ \bar{x} \geq 0 \end{cases}$$

on a , $x' = \bar{x}_I = (\bar{x}_i , i \in I)$, tel que

$$A^I x' \leq b (1 + \frac{1}{4^{---}(n+1)^{-}M})$$

Posons $\beta = 1/2(n+1)M$, parallélement aux notations prises dans le théorème I.2 on a donc :

$$\begin{cases} A^I x' \leq b (1 + \beta) \\ x' \geq 0 \end{cases}$$

D'un autre côté , pour tout j de J , on peut écrire :

$$\sum_{i \in L_j} \bar{x}_i \geq b_j \left(1 - \beta/2 \right)$$

et : $$\sum_{i \in L_j \cap I} \bar{x}_i = \sum_{i \in L_j} x'_i \geq b_j \left(1 - \beta/2 \right) - \sum_{i \notin I} \bar{x}_i \geq b_j \left(1 - \beta/2 \right) - \beta/2$$

et comme $b_j \geq 1$, on a :

$$\forall j \in J \; ; \quad \sum_{i \in L_j} x'_i \geq b_j \left(1 - \beta \right)$$

et finalement x' est tel :

$$\begin{cases} b_J \left(1 - \beta \right) \leq A_J^I x' \leq b_J \left(1 + \beta \right) \\ A_{\bar{J}}^I x' \leq b_{\bar{J}} \left(1 + \beta \right) \\ x' \geq 0 \end{cases}$$

En appliquant l'algorithme C , on va trouver un vecteur x_1 à composantes entières tel que :

$$\begin{cases} A_J^I x_1 = b_J \\ A_{\bar{J}}^I x_1 \leq b_{\bar{J}} \\ x_1 \geq 0 \end{cases}$$

Soit $x^* := \tilde{x}_1 = (x_i^*)_{i \in \mathcal{C}}$ tel que :

$$\begin{cases} x_i^* = x_{1i} & \text{si } i \in I \\ x_i^* = 0 & \text{sinon} \end{cases}$$

On a donc deux vecteurs x^* et y tels que :

$$\begin{cases} A x^* \leq b \\ x^* \geq 0 \end{cases} \qquad \begin{cases} y A \geq c \\ y \geq 0 \end{cases} \qquad \begin{array}{l} y_j > 0 \Rightarrow \langle a_j, x^* \rangle - b_j = 0 \\ \langle a^i, y \rangle - c^i > 1/nM \Rightarrow x_i^* = 0 \end{array}$$

Si on calcule :

$$\langle y, b \rangle - \langle c, x^* \rangle = \langle y, Ax^* - b \rangle + \langle yA - c, x^* \rangle = \sum_{i \in I} x_i^* \left(\sum_{j \in C_i} y_j - c^i \right)$$

$$\langle y,b\rangle - \langle c,x^*\rangle < \Big(\sum_{i\in I} x_i^* \Big) \frac{1}{n^2 M}$$

or $x_i^* \leq M$ pour tout i de I et : $\langle y,b\rangle - \langle c,x^*\rangle < 1$

Comme x^* est entier , $\langle c,x^*\rangle$ aussi et comme:

$$\langle y,b\rangle \geq \min_{y\in\mathcal{D}} \langle y,b\rangle = \max_{x\in\mathcal{P}} \langle c,x\rangle = z_o$$

$$z_o \geq \langle c,x^*\rangle > z_o - 1$$

on peut écrire : $z_o = \langle c,x^*\rangle$

Avec les paramètres choisis , la complèxité de l'algorithme F est de l'ordre $o(n^5 m M^4 M' \text{ Log } (n M))$. Comme l'algorithme C est en $o(n^4 m)$ c'est F qui donne la complèxité de l'algorithme G .

Chapitre IV : Algorithmes polynomiaux pour les matrices totalement unimodulaires

IV.A. Présentation du problème:

Définition:

A est une matrice totalement unimodulaire (matrice t.u.) si:

$$\forall I \subset \mathcal{C},\ \forall J \subset \mathcal{L} \text{ avec } |I| = |J| \text{ alors}$$
$$\det\left(A_J^I\right) \in \{-1, 0, +1\}$$

Pour des propriétés équivalentes à cette définition on pourra se reporter à [25]. Cette définition implique trivialement que tous les éléments de A appartiennent à $\{-1, 0, +1\}$.

Nous nous interessons ici au problème :

$$(PL) : \begin{cases} Ax = b \\ x \geq 0 \\ \max \langle c , x \rangle \end{cases} \qquad (1)$$

où A est une m x n matrice t.u. , $b \geq 0$ et $c \leq 0$ sont entiers et non-nuls Nous supposerons de plus que le systeme (1) a une solution réalisable pour n'importe quel membre de droite entier positif. Il est facile de voir que n'importe quel programme linéaire où la matrice des contraintes est totalement unimodulaire et les données entierespeut se mettre sous cette forme , en effet:

- si une variable x_i est non-astreinte on pourra poser
$$x_i = x_i^1 - x_i^2 \quad x_i^1 , x_i^2 \geq 0$$
- si $b_j \leq 0$ ou $c_i \geq 0$, on change le signe de la ligne ou de la colonne ou on change la variable x_i en la bornant.
- si les contraintes s'écrivent
$$\begin{cases} A x \leq b \\ x \geq 0 \end{cases}$$
on se raméne à
$$\begin{cases} A x + U z = b \\ x , z \geq 0 \end{cases}$$
où U est la matrice identité .

Il est facile de constater qu'on n' enléve pas le caractère unimodulaire de la matrice des contraintes.

Pour que l'algorithme que nous produisons ici soit polynomial dans la longueur des données , nous utilisons une méthode de scaling qui a été décrite par J.F. MAURRAS , K. TRUEMPER et M.AKGUL [22];toutefois l'algorithme de base sera aussi inspiré des méthodes décrites précédemment en utilisant la fonction entropie.

IV.B. Algorithme de base:

IV.B.1 Introduction à l'algorithme:

Soit $\mathcal{P}$ le polyèdre défini par

$$\mathcal{P} \quad \begin{cases} A\,x = u_{j_0} \\ x \geq 0 \end{cases}$$

où A est une matrice t.u. en (-1, 0 , +1) et u_{j_0} est le j_0-ième vecteur de base canonique de R^m , $u_{j_0} = (0,\ldots,0,1,0,\ldots,0)$.

On pose le problème : - $\mathcal{P}$ est-il vide ?

- si $\mathcal{P}$ n'est pas vide , trouver $X \in \mathcal{P}$.

Si $\mathcal{P}$ n'est pas vide alors il existe une solution de base soit $x^* = Bu_{j_0} \geq 0$ puisque B est en (-1, 0 , 1) on aura x^* entier et $x^* \in \{0,1\}$. Notre problème est donc équivalent au problème :

$$\mathcal{P}' \quad \begin{cases} A\,x = u_{j_0} \\ 0 \leq x \leq 1 \end{cases}$$

- $\mathcal{P}'$ est-il vide ?

-si $\mathcal{P}'$ n'est pas vide on doit trouver $x \in \mathcal{P}'$.

Pour cela on pose le problème convexe:

$$(P_c) : \operatorname*{Min}_{x \in \mathcal{P}''} \left\{ \sum_{i=1}^{2n} x_i \operatorname{Log} x_i - \sum_{i=1}^{2n} x_i \right\} = z$$

où $\mathcal{P}''$ est le polyèdre défini par les contraintes:

$$\mathcal{P}'' \quad \begin{cases} A\,x = u_{j_0} \\ x_i + x_{i+n} = 1 \qquad \forall i=1,\ldots,n \\ x \geq 0 \end{cases}$$

Si $\mathcal{P}''$ n'est pas vide , z est négatif ou nul. Le dual de (P_c) peut s'écrire (cf. Chapitre II):

$$(D_c): \operatorname{Max}_{y} \operatorname{Min}_{x \geq 0} \left\{ \left(\sum_{i=1}^{2n} x_i \operatorname{Log} x_i - \sum_{i=1}^{2n} x_i \right) - \sum_{j=1}^{m} y_j \langle a_j, x \rangle - \sum_{i=1}^{n} y_{i+n} (x_i + x_{i+n} - 1) + y_{j_o} \right\} = w$$

Soit

Soit $g(y,x)$ la fonction à minimiser et $h(y) = \operatorname{Min}_{x \geq 0} g(y,x)$ $\quad y \in \mathbb{R}^{n+m}$

Remarque:

Dans le cas où $\mathcal{P}$ n'est pas vide on aura pour tout y , $h(y) \leq w = z \leq 0$. Donc si on trouve y tel que $h(y) > 0$,on saura répondre que $\mathcal{P}$ est vide.

Notations:

Dans la suite de ce chapitre , où nous étudions des matrices t.u. en $(-1, 0, 1)$ nous conviendrons des notations suivantes:

- pour $j \in \mathcal{L} = \{1,2,\ldots,m\}$

$$L_j^+ = \{ i \in \mathcal{C} / a_{ji} = +1 \} \qquad L_j^- = \{ i \in \mathcal{C} / a_{ji} = -1 \}$$

- pour $i \in \mathcal{C} = \{1,2,\ldots,n\}$

$$C_i^+ = \{ j \in \mathcal{L} / a_{ji} = +1 \} \qquad C_i^- = \{ j \in \mathcal{L} / a_{ji} = -1 \}$$

Théorème IV.1 :

Soit $\bar{x}$ le vecteur qui réalise le minimum de la fonction $g(y,x)$,le vecteur y étant fixé. Posons $\bar{x} = (\bar{x}_i \ , \ i \in \mathcal{C} \cup \{n+1,\ldots,2n\})$ et $t_j = e^{y_j}$. On a:

$$\forall i \in \mathcal{C} \quad \bar{x}_i = \frac{\prod_{j \in C_i^+} t_j}{\prod_{j \in C_i^-} t_j} t_{m+i} \quad \text{et} \quad \bar{x}_{i+n} = t_{m+i} \qquad (1)$$

$$(2) \qquad h(y) = y_{j_o} + \sum_{i=1}^{n} y_{m+i} - \sum_{i=1}^{2n} \bar{x}_i$$

Démonstration:

On a $\dfrac{\delta g}{\delta x_i}(y,x) = \operatorname{Log} x_i - \sum_{j \in C_i^+} y_j - \sum_{j \in C_i^-} y_j - y_{m+i} \quad \forall i \in \mathcal{C}$

$$\frac{\delta g}{\delta x_{i+n}}(y,x) = \operatorname{Log} x_{m+i} - y_{m+i} \qquad \forall i \in \mathcal{C}$$

or à l'optimum $\dfrac{\delta g}{\delta x_i}(y,\bar{x}) = 0 \quad \forall i \in \mathcal{C} \cup \{n+1,\ldots,2n\}$. On en déduit les égalités (1). D'autre part, en faisant les calculs ,on trouve :

$$h(y) = g(y,\bar{x}) = y_{j_o} + \sum_{i=1}^{n} y_{m+i} - \sum_{i=1}^{2n} \bar{x}_i$$

Remarque :

Soit b le vecteur de $\mathbb{R}^{m+n}$ représentant le second membre du systéme d'égalités définissant $\mathcal{P}''$. Soit $\mathcal{P}''_i$ le polyèdre défini par :

$$\mathcal{P}''_i \quad \begin{cases} (1-\beta)b \leq A\,x \leq b\,(1+\beta) \\ x \geq 0 \end{cases}$$

avec $\beta = 1 / 2\,(n+1)$. Si $\mathcal{P}''_i$ n'est pas vide alors $\mathcal{P}''$ n'est pas vide (cf chapitre I).

IV.B.2. Description de l'algorithme :

Le vecteur y étant fixé , si $\bar{x}$ appartient à $\mathcal{P}''_i$ le problème est résolu $\mathcal{P}''$ n'est pas vide , sinon il existe une ligne d'indice j_1 telle que

$$\frac{\delta h}{\delta y_{j_1}}(y) = b_{j_1} - \sum_{i \in L^+_{j_1}} \bar{x}_i + \sum_{i \in L^-_{j_1}} \bar{x}_i = \varepsilon_{j_1} \quad \text{avec} \quad |\varepsilon_{j_1}| \geq \beta$$

Nous allons améliorer la fonction h(y) dans la direction de y_{j_1}. Nous allons chercher donc y' tel que :

$$\frac{\delta h}{\delta y_{j_1}}(y') = 0$$

1° cas : $j_1 \geq m+1$, dans ce cas :

$$\frac{\delta h}{\delta y_{j_1}}(y) = 1 - \bar{x}_i - \bar{x}_{i+n} = \varepsilon_{j_1}$$

on va poser $t'_{j_1} = \frac{1}{1+\varepsilon_{j_1}}\, t_{j_1}$ (cf chapitre II)

et $h(y') = h(y) + \varepsilon_{j_1} - \mathrm{Log}\,(1+\varepsilon_{j_1})$

2° cas : $j_1 \in \mathcal{L}$; mais $j_1 \neq j_0$

$$\frac{\delta h}{\delta y_{j_1}}(y) = -\sum_{i \in L^+_{j_1}} \bar{x}_i + \sum_{i \in L^-_{j_1}} \bar{x}_i = \varepsilon_{j_1}$$

Soit $X^+ = \sum_{i \in L^+_{j_1}} \bar{x}_i$ et $X^- = \sum_{i \in L^-_{j_1}} \bar{x}_i$. On remarque que $X^+ > 0$ et $X^- > 0$. On va chercher $t'_{j_1} = p\, t_{j_1}$ tel que :

$$t'_{j_1} \frac{X^+}{t_{j_1}} - t_{j_1} \frac{X^-}{t'_{j_1}} = 0 = p\,X^+ - \frac{X^-}{p} \qquad \text{soit } p = \sqrt{\frac{X^-}{X^+}}$$

Donc on pose : $t'_{j_1} = \sqrt{\frac{X^-}{X^+}}\, t_{j_1}$ et $\bar{x}'_i = \sqrt{\frac{X^-}{X^+}}\, \bar{x}_i$ si $i \in L^+_{j_1}$

$$\bar{x}'_i = \sqrt{\frac{X^+}{X^-}}\,\bar{x}_i \qquad \text{si} \quad i \in L^-_{j_1}$$

$$\bar{x}'_i = \bar{x}_i \qquad \text{sinon}$$

D'autre part , on a :

$$h(y') = h(y) + X^+ + X^- - pX^+ - \frac{X^-}{p} = h(y) + X^+ + X^- - 2\sqrt{X^+X^-}$$

$$h(y') = h(y) + \left(\sqrt{X^+} - \sqrt{X^-}\right)^2$$

or : $X^- - X^+ = \left(\sqrt{X^-} - \sqrt{X^+}\right)\left(\sqrt{X^-} + \sqrt{X^+}\right) = \varepsilon_{j_1}$

On déduit l'augmentation de la fonction objectif du dual :

$$h(y') = h(y) + \frac{\varepsilon^2_{j_1}}{\left(\sqrt{X^-} + \sqrt{X^+}\right)^2}$$

<u>3° cas :</u> $j_1 = j_o$

$$\frac{\delta h}{\delta y_{j_o}}(y) = -\sum_{i \in L^+_{j_o}} \bar{x}_i + \sum_{i \in L^-_{j_o}} \bar{x}_i + 1 = \varepsilon_{j_o}$$

On cherche $t'_{j_o} = p\, t_{j_o}$ tel que :

$$p\,X^+ - \frac{X^-}{p} = 1 \quad \Leftrightarrow \quad p^2X^+ + p - X^- = 0$$

puisqu'on prend $p > 0$, on trouve que :

$$p = \frac{1 + \sqrt{1 + 4X^+X^-}}{2X^+}$$

et $h(y') = h(y) + \left(\text{Log } p - pX^+ - \frac{X^-}{p} + X^+ + X^- \right)$.

Nous pouvons alors décrire l'algorithme suivant:

<u>Algorithme II:</u>

<u>Entrée:</u> A matrice totalement unimodulaire, le systéme:

$$\mathcal{P}'' \quad \begin{cases} A\,x_1 = u_{j_o} \\ x_1 + x_2 = 1 \\ x_1 , x_2 \geq 0 \quad x_k \in \mathbb{R}^n \end{cases}$$

Sortie: un vecteur $\bar{x}=(\bar{x}_1,\bar{x}_2)$ tel que :

$$\begin{cases} u_{j_o} - \beta \leq A\,\bar{x}_1 \leq u_{j_o} + \beta \\ 1 - \beta \leq \bar{x}_1 + \bar{x}_2 \leq 1 + \beta \\ \bar{x}_1 \,,\ \bar{x}_2 \geq 0 \end{cases}$$

ou la réponse " $\mathcal{P}$" est vide".

Compléxité: Soit Nmax = $32\ n^4$; l'algorithme est en $o(mn^5)$.

Description de la procédure:

Phase (0) : Poser $\bar{x}_1 := \bar{x}_2 := \mathbb{1}_n$.

Niter:=1.

Phase (1) : Chercher s'il existe une ligne d'indice $j = i + m \geq m$ tel que :

$$\bar{x}_{1i} + \bar{x}_{2i} = 1 + \varepsilon_j \quad \text{avec} \quad |\varepsilon_j| > \beta$$

si oui aller en (2) sinon aller en (3).

Phase (2) : Poser $\bar{x}_{1i} = \dfrac{1}{1+\varepsilon_j}\ \bar{x}_{1i}$;

$$\bar{x}_{2i} = \frac{1}{1+\varepsilon_j}\ \bar{x}_{2i} ;$$

Aller en (7).

Phase (3) : Chercher s'il existe une ligne d'indice $j_1 \leq m$ et $j_1 \neq j_o$ tel que :

$$X^+ - X^- = \sum_{i \in L^+_{j_1}} \bar{x}_i - \sum_{i \in L^-_{j_1}} \bar{x}_i = \varepsilon_{j_1} \quad \text{avec } |\varepsilon_{j_1}| > \beta$$

si oui aller en (4) sinon aller en (5).

Phase (4) : Poser $p = \sqrt{X^-/X^+}$ et $\bar{x}_i = p\,\bar{x}_i$ si $i \in L^+_{j_1}$

$\bar{x}_i = \bar{x}_i / p$ si $i \in L^-_{j_1}$

$\bar{x}_i = \bar{x}_i$ sinon.

Aller en (7).

Phase (5) : Si $\bar{x}$ est tel que :

$$X^+ - X^- = \sum_{i \in L^+_{j_o}} \bar{x}_i - \sum_{i \in L^-_{j_o}} \bar{x}_i = 1 + \varepsilon_{j_o} \quad \text{avec } |\varepsilon_{j_o}| > \beta$$

aller en (6) sinon : Terminer $\bar{x}$ répond à la question.

Phase (6) : Poser: $p = \dfrac{1 + \sqrt{1 + 4X^+X^-}}{2X^+}$

et $\bar{x}_i := p\,\bar{x}_i$ si $i \in L^+_{j_o}$;

$\bar{x}_i := \bar{x}_i/p$ si $i \in L^-_{j_o}$;

$\bar{x}_i := \bar{x}_i$ sinon.

Aller en (7).

Phase (7) : Poser Niter := Niter + 1 ;

Si Niter $\leq$ Nmax aller en (1) sinon Terminer : écrire "$\mathcal{P}$" est vide".

Démonstration de la validité:

On part de y=0 et donc de $h(y) = -2n$. A chaque fois qu'on passe dans la phase (2) , la fonction h augmente d'au moins $\beta^2/4$. A chaque fois qu'on passe dans la phase (4) , on a $\bar{x}_i \leq 1 + \beta$ pour tout $i = 1,\ldots,2n$ (puisque on a refusé le passage dans la phase (3)), donc $X^+ \leq n(1+\beta)$ et $X^- \leq n(1+\beta)$ et ainsi on a $(\sqrt{X^+} + \sqrt{X^-})^2 \leq 4n(1+\beta)$. Donc l'augmentation de la fonction h est d'au moins $\beta^2/4n(1+\beta)$. Soit N_1 le nombre de passages dans la phase (2) , N_2 dans la phase (4) et N_3 dans la phase (6) . On a Niter= $N_1 + N_2 + N_3$. D'autre part chaque fois qu'on passera en (6) on ajuste la contraintre j_o et l'itération suivante on passera soit en (2) , soit en (4). Donc $N_3 \leq N_1 + N_2$ et Niter $\leq 2(N_1 + N_2)$. Comme l'augmentation de la fonction h est d'au moins $1/16n^3$ on est sûr qu'en au plus $32n^4$ itérations on aura soit trouvé $\bar{x}$ vérifiant le système approché , soit $h(y) > 0$ et donc on peut conclure que $\mathcal{P}$" est vide . Le calcul de la compléxité se fait en calculant le nombre d'opérations élémentaires.

I.V.C. Procédure de programmation linéaire:

I.V.C.1. Introduction au problème initial:

Nous nous intéressons ici au problème de programmation linéaire (P).

$$(P)\quad \begin{cases} A\,x = u_{j_o} \\ x \geq 0 \\ \text{Max } \langle c, x \rangle \end{cases}$$ où A est une matrice t.u. et c est entier.

De plus on suppose que le polyèdre des contraintes n'est pas vide . Le problème dual de (P) est :

$$(D)\quad \begin{cases} y\,A \geq c \\ y \text{ non astreint} \\ \min \quad y_{j_0} \end{cases}$$

Soit y un vecteur dual réalisable . On peut décomposer le systéme des contraintes en deux sous-ensembles :

$$\begin{cases} y\,A^I = c^I \\ y\,A^{\bar{I}} > c^{\bar{I}} \end{cases}$$

y est non optimal si et seulement si il existe y_0 tel que:

$$\begin{cases} y_0\,A^I \leq 0 \\ y^o_{j_0} > 0 \end{cases}$$

Puisque A est t.u. on peut affirmer que dans ce cas il existe un vecteur y_0 entier qui répond à la question et donc que :

$$\begin{cases} y_0\,A^I \leq 0 \\ y^o_{j_0} \geq +1 \end{cases}$$

en faisant les modifications préconisées au début de ce chapitre ,on écrit:

il existe x tel que: $B^I x = u_{n'} \quad x \geq 0 \quad n' = |I| + 1$

où la matrice B^I est définie par:

$t_A{}^I$	$-t_A{}^I_{J_0}$	$-U$	0
1	0	0	1

Le nombre de colonnes de la matrice B^I est $\bar{n}$ avec :

$$\bar{n} = 2m - 1 + |I| + 1 \leq n + 2m$$

La matrice $A^I_{j_o}$ correspondant à la matrice A^I_J avec $J = \mathcal{L} - \{j_o\}$. Le nombre de lignes de la matrice B^I est égal à $|I| + 1$. Il est facile de voir que la matrice B^I est totalement unimodulaire ; puisque, ajoutant la contrainte $y_{j_o} \geq 1$ à la matrice A^I la matrice ainsi obtenue est t.u. .

I.V.C.2. Description de l'algorithme :

Algorithme I :

Entrée : Le programme linéaire :

$$(1)\begin{cases} A\,x = u_{j_o} \\ x \geq 0 \\ \text{Max } \langle c, x \rangle \end{cases}$$ où A est une m × n matrice t.u. et c est entier.

Supposition : Le système (1) a une solution réalisable.

Sortie : Un vecteur optimal entier x^* pour (1).

Complèxité : $o(mn^7 c_{max})$ où $c_{max} = \underset{i \in \mathcal{L}}{\text{Max}} |c^i|$

Description de la procédure :

Phase (0) : Prendre une solution duale réalisable y^o par un algorithme "greedy".

Phase (1) : Poser $I = \{ i \in \mathcal{L} / y^o A^i = c^i \}$.

Appliquer l'algorithme II avec comme entrée la matrice B^I telle qu'elle a été définie au paragraphe précédent.

Si II se termine par la réponse d'un vecteur $x' \geq 0$ tel que :

$|B^I x' - u_n| \leq \beta$ et $|x' - 1| \leq \beta$

aller en (2) sinon aller en (3).

Phase (2) : Soit $b' = B^I x'$. Trouver une solution de base x^1_1 et la base correspondante B telle que :

$x^1_1 \geq 0$; $B^I x^1_1 = b'$ (les pivots doivent être soigneusement pris comme indiqués dans [7]).

Soit $y^1 = B^{-1} b$, y^1 définit un vecteur y de base tel que

$$\begin{cases} y A^I \leq 0 \\ y_{j_o} \geq 1 \end{cases}$$

Poser $y^o := y^o - qy$ avec q tel que :

$$q\, y\, A^I \geq 1$$

Aller en (1).

Phase (3) : y^o est solution optimale du dual .

Appliquer l'algorithme H avec en entrée la matrice A^I et u_{j_o} . Soit x'' le vecteur rendu par H.

Soit $b'' = A^I x''$. Trouver une solution de base x_1'' et la base correspondante B'' telles que : $A^I x_1'' = b''$, $x_1'' \geq 0$.

Soit $x = B''^{-1} u_{j_o}$; x est tel que :

$$\begin{cases} A^I x = u_{j_o} \\ x \geq 0 \end{cases}$$

Soit $x^* := \tilde{x}$; x^* est solution optimale de base du système (1).

Démonstration de la validité :

Partant d'une solution duale réalisable , on a au plus $y^o_{j_o} \leq \underset{i}{\text{Max}} |c^i|$. La valeur minimale de l'optimum de la fonction objectif du primal est:

$$\min = -n \underset{i}{\text{Max}} |c^i| = -n\, c_{max} .$$

A la fin de l'algorithme , on a deux vecteurs x^* et y^o tels que :

$$\begin{cases} A\, x^* = u_{j_o} \\ x^* \geq 0 \end{cases} \qquad y^o A \geq c \qquad \langle y^o, a^i \rangle - c^i > 0 \Rightarrow x^*_i = 0$$

En utilisant le théorème des écarts complémentaires , on en déduit que x^* et y^o sont optimales .

-Si H répond que le polyèdre suivant est vide

$$\begin{cases} y\, A^I \leq 0 \\ y_{j_o} > 0 \end{cases}$$

alors en utilisant le Lemme de Farkas il existe un vecteur x tel que :

$$\begin{cases} A^I x = u_{j_o} \\ x \geq 0 \end{cases}$$

ce qui nous certifie que dans la phase (3) , H nous répond que ce polyèdre n'est

pas vide .

- On remarque que le vecteur y solution de base de :

$$\begin{cases} y\, A^I \lesssim 0 \\ y_{j_0} \gtrsim 1 \end{cases}$$

est tel que $y_j \in \{0,1\}$. Donc pour tout i non dans I , on a :

$$\langle a^i, y \rangle \lesssim n$$

Puisque $\langle a^i, y^0 \rangle \gtrsim c^i + 1 \quad \forall i \notin I$, on a $q \gtrsim 1/n$.

On endéduit que la valeur de la fonction objectif du dual décroit de $1/n$ à chaque passage dans la phase (2) . Ce qui nous permet d'affirmer qu'en au plus $n(n+1)\, c_{max}$ itérations l'algorithme I est fini. En combinant ceci à la complèxité de l'algorithme II on en déduit la complèxité de cet algorithme.

IV.D. Algorithme de résolution :

Maintenant nous pouvons résoudre le problème de Programmation Linéaire si la matrice des contraintes est t.u. et les données entieres , ceci en temps polynomial par rapport à la taille des données . La méthode que nous allons décrire est celle employée dans [22], celle-ci est une méthode de "scaling" similaire à celle employée pour la première fois par Edmonds et Karp [8] dans un algorithme polynomial pour un problème de flot dans des réseaux.

La méthode de résolution procéde ainsi , l'algorithme J transforme le problème initial en plusieurs programmes linéaires , chacun ayant en second membre un vecteur de base canonique . L'algorithme K prend un de ces problèmes , en trouve une solution réalisable . Celle-ci est utilisée pour transformer le dual en un problème identique au programme linéaire principal et utilise la même méthode de scaling pour le résoudre .

IV.D.1. Algorithme principal:

Algorithme J

Entrée : Le programme linéaire

$$(1)\begin{cases} A\,x = b \\ x \geq 0 \\ \text{Max } \langle c,x \rangle \end{cases}$$

où A est t.u. ; $b \geq 0$, $c \leq 0$ sont entiers et non-nuls.

Supposition : (1) a une solution réalisable pour tout second membre non-négatif

Sortie : x^* solution optimale entière de (1) .

Complèxité : $o(\, d_2(b) \,.\, T\,)$ où $d_2(z)$ est le nombre de 1 qu'il faut pour écrire le vecteur entier z en représentation binaire et T la complèxité de l'algorithme K décrit plus loin.

Description de la procédure :

Phase (0) : Soit D une matrice en (0,1) avec m lignes où chaque ligne j représente b_j en notation binaire . D a suffisament de colonnes pour que le plus grand des b_j soit écrit .

Poser k := 1 et x^* := 0 ;

Phase (1) :Pour j := 1,...,m : si $D_{jk} = 1$ alors résoudre :

$$(2)\begin{cases} A\,x = u_j \\ x_i \geq 0 \qquad \forall\, i \in I \\ \text{Max } \langle c,x \rangle \end{cases}$$

où $I = \{\, i \;/\; x^*_i = 0 \,\}$; par l'algorithme K , rendant $\hat{x}$, puis poser

$$x^* := x^* + \hat{x} \,.$$

Phase (2) : Poser k := k + 1 .

Si D a au moins k colonnes , poser x^* := 2 x^* et aller en (1) sinon terminer ; x^* résoud (1) .

Démonstration de la validité : cf [22] .

IV.D.2. Algorithme de routine :

Algorithme K:

Entrée : Programme linéaire (2) ;

Suppo ition : (2) a une solution optimale .

Sortie : $\hat{x}$ solution optimale de base pour (2) .

Complèxité : $o(n^8 d_2(-c))$ où la fonction d_2 est celle définie plus haut.

Description de la procédure :

Phase (0) : - Trouver une solution réalisable de base $\tilde{x}$ de système $A x = u_j$, avec $\tilde{x}_i \geq 0$ pour $i \in I$; en utilisant l'algorithme H et la méthode employée dans l'algorithme I à la phase (2).

- Changer (2) en le système :

$$(3) \quad \begin{cases} \min \langle -c, x \rangle \\ A x = u_j \\ x_i \geq 0 \quad \forall i \in I \; ; \; x_i \geq -2 \quad \forall i \notin I . \end{cases}$$

Comme $\tilde{x}$ est une $(-1,0,+1)$-solution pour (3), ce dernier problème peut être changé en :

$$(4) \quad \begin{cases} \min \langle -c, x \rangle \\ A x = 0 \\ x \geq d \end{cases}$$

où d est le $(0,-1,-2,-3)$-vecteur défini par :

$$d_i = \begin{cases} 0 - \tilde{x}_i & \text{si } i \in I \\ -2 - \tilde{x}_i & \text{si } i \notin I \end{cases}$$

Le problème dual de (4) est :

$$(5) \quad \begin{cases} \max \langle (0 \; d^t) \; , \begin{bmatrix} y^1 \\ y^2 \end{bmatrix} \rangle \\ \left[A^t / U \right] \begin{bmatrix} y^1 \\ y^2 \end{bmatrix} = -c \\ y^2 \geq 0 \end{cases}$$

Phase (1) : - Nous allons résoudre (5) d'une manière analogue à l'algorithme J. Soit E une $(0,1)$-matrice avec n lignes où la ligne 1 représente $-c^1$ en notation binaire.

- Poser $k := 1$ et $y^* := \begin{bmatrix} y^{1*} \\ y^{2*} \end{bmatrix} := 0$;

Phase (2) : Pour l := 1,...,n : si $E_{kl} = 1$ résoudre :

$$(6) \quad \begin{cases} \max \ \langle (0, d^t) , y \rangle \\ [A^t | U]\, y = u_l \\ y_i^2 \geq 0 \quad i \in L \end{cases}$$

où $L = \{ i / y_i^{2*} = 0 \}$, par l'algorithme I rendant $\hat{y}$; poser $y^* := y^* + \hat{y}$.

Phase (3) : Poser k := k + 1 ; si E a au moins k colonnes poser $y^* := 2y^*$; aller en (2) .

Phase (4) : y^* résoud (5) . Par l'algorithme H puis la méthode de l'algorithme I , phase (2) , trouver une solution de base x^* à :

$$\begin{cases} A\, x = u_j \\ x_i \geq 0 \quad i \in I \\ x_i = 0 \quad \text{si } i / y_i^{2*} > 0 \end{cases}$$

x^* est solution optimale entière pour (2) .

Démonstration de la validité : cf [22].

Théorème I.V. :

L'algorithme J utilisant l'algorithme K comme subroutine pour résoudre (2) a une complèxité de l'ordre de $o(d_2(b)\, d_2(-c)\, n^8)$.

PARTIE II : L'AFFECTATION EXPONENTIELLE

La fonction entropie a déjà été largement utilisée dans les modèles de régulation de trafic [9] [27]. Etant donné un réseau routier et un trafic circulant entre une origine et une destination, nous étudions, dans un cadre beaucoup plus général, une extension d'un modèle d'affectation de trafic introduit par R.B. Dial [5] puis complété par J. Fonlupt [11] [12].
L'originalité de cette méthode et le lien commun avec la première partie sont que nous utiliserons des méthodes de pénalités en ajoutant à la fonction objectif du problème primal une fonction de pénalité de type entropie.

Les méthodes que nous exposons dans cette partie ont l'avantage pratique d'être à la fois simples et efficaces, et ainsi d'être facilement utilisées pour des problèmes d'affectation de trafic.

CHAPITRE I - INTRODUCTION - RAPPELS

I.A. INTRODUCTION - NOTATIONS

Soit un graphe $G = (V,E)$; l'ensemble des sommets est $N = \{v_1,\ldots,v_n\}$ et l'ensemble des arcs $E = \{e_1,\ldots,e_m\}$. Les sommets v_1 et v_n seront appelés respectivement sommet origine et sommet destination de G. A chaque arc e_i est associé un coût $c(e_i)$.

Le graphe G représentera un réseau routier, les sommets v_i de V étant les noeuds du réseau et les arcs e_i de E étant les tronçons du réseau reliant certaines paires de noeuds. Le coût $c(e_i)$ de l'arc e_i représente par exemple le temps nécessaire pour relier le noeud origine $or(e_i)$ au noeud extrémité $ex(e_i)$.

Considérons un trafic d'intensité F se rendant du sommet v_1 au sommet v_n. Une affectation sera une répartition du trafic entre tous les itinéraires reliant v_1 à v_n. L'affectation suivant le plus court chemin serra celle où tout le trafic emprunte un plus court chemin de v_1 à v_n. Dial [5] a étudié un autre type d'affectation dans le cas où le graphe G est sans circuit. J. Fonlupt [11] [12] a complété cette étude de l'affectation exponentielle dans le cas général.

Nous nous proposons ici de prolonger celle-ci dans les cas suivants :

- Le problème comporte des contraintes de capacités supérieures sur chaque arc.

- Chaque noeud est soumis à une condition émission-attraction.
- Le problème comporte des contraintes de capacités inférieures sur chaque arc.

Nous étudions également le problème de l'affectation exponentielle dans le cas du multi-flot.

I.B. RAPPELS SUR L'AFFECTATION EXPONENTIELLE

Rappelons tout d'abord les résultats obtenus dans [5] [12].

I.B.1. Définitions

Un chemin t sera défini par la succession de ces arcs $t = [e_{i_1}, \ldots, e_{i_p}]$. $T(1,n)$ sera l'ensemble des chemins joignant v_1 à v_n. $T(1,n ; e_i)$ sera l'ensemble des chemins joignant v_1 à v_n passant par l'arc e_i. La longueur $c(t)$ d'un chemin sera définie par :

$$c(t) = \sum_{e_i \in t} c(e_i)$$

L'affectation exponentielle A_λ de paramètre $\lambda > 0$ peut se définir de la manière suivante :
Chargeons chaque itinéraire $t \in T(1,n)$ d'un trafic $r(t)$ égal à:

$$r(t) = \mu\, e^{-\frac{c(t)}{\lambda}}$$

La constante μ sera définie de telle façon que la somme de tous les trafics sur les chemins t de $T(1,n)$ soit égale à

l'intensité F du trafic entrant par v_1. L'affectation exponentielle A_λ définit sur chaque arc e de E un trafic noté $x_\lambda(e)$ tel que :

$$x_\lambda(e) = \sum_{t \in T(1,n;e)} r(t)$$

Tout vecteur trafic entre v_1 et v_n, $X = [x(e) ; e \in E]$ satisfait aux contraintes :

$$P : \begin{cases} x \geq 0 \\ y(1) = \sum_{e \in \Omega^+(1)} x(e) = \sum_{e \in \Omega^-(1)} x(e) + F \\ y(i) = \sum_{e \in \Omega^+(i)} x(e) = \sum_{e \in \Omega^-(i)} x(e) \\ y(n) = \sum_{e \in \Omega^-(n)} x(e) = \sum_{e \in \Omega^+(n)} x(e) + F \end{cases}$$

où nous avons pris la notation $\Omega^+(S)$ pour désigner le cocycle sortant de l'ensemble de sommets S et $\Omega^-(S)$ pour le cocycle entrant. Soit P le polyèdre défini par ces contraintes. Un plus court chemin entre v_1 et v_n est une solution de base du programme linéaire.

$$(L_o) \begin{cases} X \in P \\ Z_o = \min Z_o(X) = \min \sum_{e \in E} c(e)\, x(e) \end{cases}$$

Pour que (L_o) ait une solution finie, nous supposerons pour

la suite que le graphe G ne contient pas de circuit de longueur négative.

Pour X élément de P, nous poserons

$$f_i(X) = \sum_{e \in \Omega^-(i)} x(e) \operatorname{Log} \frac{x(e)}{y(i)} \quad \text{pour} \quad 2 < i < n$$

$$f_1(X) = F \operatorname{Log} \frac{F}{y(1)} + \sum_{e \in \Omega^-(1)} x(e) \operatorname{Log} \frac{x(e)}{y(1)}$$

$$f_\lambda(X) = \lambda \sum_{i=1}^{n} f_i(X)$$

Soit (L_λ) le programme convexe :

$$\begin{cases} x \in P \\ z_\lambda = \min z_\lambda(X) = \min \left\{ f_\lambda(X) + \sum_{e \in E} c(e)\, x(e) \right\} \end{cases}$$

I.B.2. Résultats

* Théorème I.1 : [12]

a - Il existe $\lambda_o > 0$ tel que pour tout $\lambda > 0$ et $\lambda < \lambda_o$ le programme (L_λ) a une solution unique X_λ qui est le vecteur trafic de l'affectation exponentielle de paramètre λ.

b - La fonction $\lambda \to (X_\lambda, Z_\lambda)$ est continue pour $0 < \lambda < \lambda_o$. La fonction $\lambda \to Z_\lambda$ est décroissante. Quand λ

tend vers 0, Z_λ tend vers Z_o et X_λ tend vers une solution X_o de (L_o).

$$c - X_\lambda > 0.$$

* Résolution de

Nous poserons : $c_\lambda(e) = \exp(-c(e)/\lambda)$. La matrice $A(\lambda)$ de $M(n,n)$ sera définie par :

- $a_{ii} = 0$

- S'il n'existe aucun arc joignant v_i à v_j : $a_{ij} = 0$.

- S'il existe e tel que $on(e) = v_i$ et $ex(e) = v_j$: $a_{ij} = c_\lambda(e)$.

Posons :

$$u_1^T = [1,0,\ldots,0]$$
$$u_n^T = [0,\ldots,0,1]$$

$$\omega = [\omega(i) \; ; \; i = 1,\ldots,n]$$
$$\omega' = [\omega'(i) \; ; \; i = 1,\ldots,n]$$

* Théorème I.2 : [12]

a - Le système linéaire $[I - A^T(\lambda)]\omega = u_1$ a une solution unique $\omega > 0$.

b - Le système linéaire $[I - A(\lambda)]\,\omega' = u_n$ a une solution

unique $\omega' > 0$.

c - La solution X_λ de L_λ est donnée par :

$$x_\lambda(e) = \frac{\omega(or(e)).\omega'(ex(e))}{\omega(n)}$$

d - La valeur de (L_λ) est $Z_\lambda = -\lambda F \text{ Log } \omega(n)$.

* Théorème I.3

a - L'inverse $B(\lambda) = [b_{ij}(\lambda)]$ de $M(n,n)$ de la matrice $I-A(\lambda)$ existe et $B(\lambda) > 0$ pour tout $\lambda > 0$:

$b_{ij}(\lambda) \neq 0 \longleftrightarrow$ il existe un chemin d'origine v_i et d'extrémité v_j

b - Supposons qu'il existe un chemin d'origine v_i et d'extrémité v_j. Soit A_λ l'affectation exponentielle pour un trafic d'intensité F entre ces deux sommets. A_λ produit sur chaque arc e d'origine v_ℓ et d'extrémité v_k un trafic :

$$x_\lambda(e) = F\, c_\lambda(e)\, \frac{b_{i\ell}(\lambda)\, b_{ik}(\lambda)}{b_{ij}(\lambda)} .$$

CHAPITRE II : CAS DE CONTRAINTES DE CAPACITES SUPERIEURES

II.A. DEFINITION DU PROBLEME

Nous nous intéresserons dans ce chapitre au problème de l'affectation exponentielle dans le cas où on est soumis sur chaque arc e de E à des contraintes de capacités supérieures S(e). C'est un problème pratique important, surtout dans les problèmes de trafic où on est toujours obligé de prendre en compte ces contraintes. Nous verrons à la fin de ce chapitre pourquoi la méthode décrite est efficace

Le vecteur trafic doit donc satisfaire les contraintes suivantes :

$$P_S \begin{cases} X \in P \\ x(e) \leq S(e) \quad \forall\, e \in E \end{cases}$$

Soit P_S le polyèdre décrit par ces contraintes. Nous supposerons de plus que pour toute coupe (V_1, V_2) séparant le sommet origine v_1 ($v_1 \in V_1$) du sommet destination v_n ($v_n \in V_2$), on a :

$$\sum_{e \in \Omega^+(V_1)} S(e) > F$$

Il est facile de constater que ceci entraîne que le polyèdre P_S n'est pas vide.

Soit (L_λ^S) le programme convexe :

$$(L_\lambda^S) : \quad X \in P_S$$

$$Z_\lambda^S = \min Z_\lambda(X) = \min \{ f_\lambda(X) + \sum_{e \in E} c(e)x(e) \}$$

Connaissant les propriétés de la fonction convexe $f_\lambda(X)$ [1] [12] et en utilisant les mêmes méthodes que dans la partie I, on peut montrer que le dual (D_λ^S) de (L_λ^S) peut s'écrire :

$$(D_\lambda^S) : Z_\lambda^S = \max_{\mu \geq 0} \min_{X \in P} \{ f_\lambda(X) + \sum_{e \in E} c(e)x(e) + \sum_{e \in E} (x(e) - S(e)\mu(e)) \}$$

Si on pose, pour μ_0 vecteur de $R^{|E|}$, $\mu_0 \geq 0$.

$$g(\mu_0) = \min_{X \in P} \{ f_\lambda(X) + \sum_{e \in E} (c(e) + \mu(e)x(e) - \sum_{e \in E} (S(e)\mu(e)) \}$$

Quand le vecteur μ_0 est fixé, pour calculer $g(\mu_0)$, nous sommes ramenés à résoudre un problème d'affectation exponentielle (L_{μ_0}) où les coûts sur chaque arc sont modifiés suivant une loi de "pénalité" fonction de μ_0.

* Théorème II.1

a $\forall \lambda \neq 0 \quad \lambda < \lambda_0$ la solution optimale X_λ^S de (L_λ^S) est unique.

b - La fonction $\lambda \to (X_\lambda^S, Z_\lambda^S)$ est continue pour $\lambda > 0$.

c Quand λ tend vers 0, X_λ^S tend vers X_0^S solution du problème de flot de coût minimum pour le polyèdre P_S.

Démonstration : Ces résultats sont des applications de théorèmes classiques d'analyse sur le comportement d'une minimum d'une fonction continue dépendant d'un paramètre λ sur un ensemble compact et du théorème I.1 ; ils se démonstrent de la même façon que celui-ci.

II.B. RESOLUTION DU PROBLEME

II.B.1. Notations

Le vecteur $\mu_o \geq 0$ étant fixé, pour résoudre le problème d'affectation exponentielle (L_{μ_o}), nous reprenons les notations du chapitre précédent en indiquant toutefois les éléments en question dépendant de μ_o, nous avons donc :

$$(I - A_o^T)\ \omega_o = u_1 \quad ; \quad (I - A_o)\ \omega_o' = u_n$$

$$x_\lambda^o(e) = F\ c_\lambda(e)\ \exp(-\mu_o\ (e)/\lambda)\ \frac{\omega_o(or(e)).\omega_o'(ex(e))}{\omega_o(n)}$$

$$g(\mu^o) = -\lambda\ F\ \mathrm{Log}\ \omega_o(n) - \sum_{e \in E} S(e)\ \mu_o(e)$$

où X_λ^o est le vecteur solution de (L_{μ_o}). Les conditions de Kuhn et Tucker [24] nous apprennent que μ_o et X_λ^o sont solutions optimales de (D_λ^S) et (L_λ^S) s'ils vérifient :

$$\begin{cases} \mu_o \quad 0 \quad ; \quad X_\lambda^o \in P_S \\ \mu_o(e)\ (x_\lambda^o(e) - S(e)) = 0 \quad \forall\ e \in E \end{cases}$$

Partant d'un vecteur $\mu_o \geq 0$, on va améliorer la fonction g si μ_o et X_λ^o ne vérifient pas ces conditions d'optimalité, en améliorant dans une direction de coordonnée choisie.

Un arc e_o étant choisi tel que :

$$\mu_o(e_o)\ (x^o_\lambda\ (e_o) - S(e_o)) \neq 0$$

On va chercher à minimiser

$$g(\mu) = g(\mu_o + \beta\ u_{e_o})$$

en conservant les contraintes $\mu \geq 0$ donc $\mu_o(e_o) + \beta \geq 0$.

II.B.2. Résolution de $\min(\mu_o + \beta\ u_{e_o})$

Nous noterons : $E_{ij} = (e_{k\ell})$ la matrice $n \times n$ telle que :

$e_{k\ell} = 0$ si $k \neq i$ ou $\ell \neq j$

$e_{ij} = 1$

La matrice $A(\mu)$ des coûts telle qu'elle a été définie dans le chapitre précédent vérifie :

$$A(\mu) = A(\mu_o) + (e^{-\frac{\beta}{\lambda}} - 1)\ a_{or(e_o)ex(e_o)}\ E_{or(e_o)ex(e_o)}$$

Le problème que nous allons d'abord résoudre est, connaissant l'inverse de $I-A(\mu_o)$, calculer l'inverse de $I-A(\mu)$ où un seul coefficient a été modifié, par rapport à $I-A(\mu_o)$.

Pour la suite nous poserons :

$$i = or(e_o) \quad ; \quad j = ex(e_o)$$

$$\alpha = (e^{-\frac{\beta}{\lambda}} - 1)\ a_{ij} ; \gamma = e^{-\beta/\lambda} - 1$$

$$B(\mu) = [I-A(\mu)]^{-1} \quad ; \quad B = B(\mu^o)$$

$$A(\mu) = A(\mu^o) + \alpha\ E_{ij}$$

Soit la matrice $C = B(I-A(\mu)) = I - \alpha B E_{ij}$.

Nous avons $C^{-1} = I + \frac{\alpha}{1-\alpha b_{ji}} B E_{ij}$, en supposant que C^{-1} existe. En effet :

$$C^{-1} C = I - \alpha B E_{ij} + \frac{\alpha}{1-\alpha b_{ji}} B E_{ij} - \frac{\alpha^2}{1-\alpha b_{ji}} B E_{ij} B E_{ij}.$$

En remarquant que pour toute matrice carrée $A = (a_{ij})$

$$E_{ij}A = \sum_{\ell=1}^{n} a_{j\ell} E_{i\ell} \; ; \; A E_{ij} = \sum_{k=1}^{n} a_{ki} E_{kj}$$

et $E_{ij} A E_{ij} = a_{ji} E_{ij}$

Nous pouvons écrire :

$$C^{-1} C = I - \alpha B E_{ij} + \frac{\alpha}{1-\alpha b_{ji}} B E_{ij} - \frac{\alpha^2 b_{ji}}{1-\alpha b_{ji}} B E_{ij}$$

et ainsi on a bien $C^{-1}C = I$.

D'où

$$B(\mu) = I-A(\mu)^{-1} = C^{-1}B$$

et

$$B(\mu) = B + \frac{\alpha}{1-\alpha b_{ji}} B E_{ij} B$$

Ainsi connaissant la matrice B et le coefficient α il sera très facile de calculer la matrice $B(\mu)$ qui sera utile pour résoudre le problème de l'affectation exponentielle pour le nouveau vecteur μ. De plus, on remarquera les calculs suivants que :

$$B E_{ij} B = B\left(\sum_{\ell=1}^{n} b_{j\ell} E_{i\ell}\right) = \sum_{\ell=1}^{n} b_{j\ell} \sum_{k=1}^{n} b_{ki} E_{k\ell}$$

$$B E_{i} B = \sum_{k=1}^{n} \sum_{\ell=1}^{n} b_{ki} b_{j\ell} E_{k\ell} = (b_{ki} b_{j\ell})_{k,\ell}$$

Nous pouvons alors calculer le coefficient β pour minimiser la fonction $g(\mu_o + \beta\ e_o)$.

Nous remarquons que :

$$\omega_\mu(n) = b_{ln}(\mu) = \omega^o(n) + \frac{\alpha}{1-\alpha b_{ji}}\ b_{li}\ b_{jn}$$

$$b_{li}(\mu) = b_{li}\ \frac{1}{1-\alpha b_{ji}}$$

$$b_{jn}(\mu) = b_{jn}\ \frac{1}{1-\alpha b_{ji}}$$

On peut montrer qu'on déduit facilement

$$\omega_\mu(n) = \omega^o(n)\ \left(1 + \frac{\gamma}{F(1-\alpha b_{ji})}\ x^o(e_o)\right)$$

et $$\omega'_\mu(n) = -\ \frac{a_{ij}\ e^{-\beta/\lambda}}{\lambda}\ b_{li}(\mu)\ b_{jn}(\mu)$$

Si on calcule la dérivée par rapport à β de $g(\mu_o + \beta\ u_{e_o})$, on trouve :

$$\frac{dg}{d\beta}(\mu) = -\ \lambda\ F\ \frac{\omega'_\mu(n)}{\omega_\mu(n)} - S(e_o) = x_\mu(e_o) - S(e_o)$$

Ainsi le maximum sur la direction choisie sera obtenue pour tel que :

Si $x^o(e_o) - S(e_o) > 0$ alors $x_\mu(e_o) = S(e_o)$

Si $x^o(e_o) - S(e) < 0$ alors

- soit $x_\mu(e_o) = S(e_o)$ si $\mu(e_o) \geq 0$
- soit $\mu(e_o) = 0$.

Nous allons voir que le calcul de β peut se faire en résolvant une équation algébrique du second degré. En effet :

$$\omega'_\mu(n) = -\ \frac{\gamma+1}{\lambda F}\ \omega^o(n)\ x^o(e)\ \ \frac{1}{(1-\gamma a_{ij} b_{ji})^2}$$

A l'optimum on a :

$$-\lambda F \omega'_{\mu}(n) = \omega_{\mu}(n) S(e_o)$$

$$x^o(e_o)(\gamma+1) = S(e_o)\{(1-\gamma a_{ij}b_{ji})^2(1+ \frac{\gamma}{(1-\gamma b_{ji}a_{ij})} \frac{x^o(e_o)}{F}\}$$

$$x^o(e_o)(\gamma+1) = S(e_o)\{(1-\gamma a_{ij}b_{ji})^2 + \gamma(1-\gamma a_{ij}b_{ji}) \frac{x^o(e_o)}{F}\}$$

La valeur γ est donc solution de l'équation algébrique du second degré :

$$S(e_o)(a_{ij}^2 b_{ji}^2 - a_{ij}b_{ji} \frac{x^o(e_o)}{F})\gamma^2 + \gamma\{S(e^o)(\frac{x^o(e_o)}{F} - 2a_{ij}b_{ji}) - x^o(e^o)\}$$

$$+ S(e_o) - x^o(e^o) = 0.$$

Connaissant γ on peut en déduire β.

II.B.3. Algorithme de résolution

Nous pouvons présenter l'algorithme suivant :

- Partir d'un ensemble de pénalités μ sur les arcs (par exemple 0).

- Résoudre l'affectation exponentielle en remplaçant les coûts initiaux $c(e)$ par $c(e)+\mu(e)$.

- Si le flot et le système de pénalités vérifient les conditions de Kuhn et Tucker, on est à l'optimum. Sinon :

. S'il existe un arc e_o tel que le flot sur cet arc dépasse sa capacité, alors on augmente la pénalité sur e_o.

. Sinon s'il existe un arc e_o tel que le flot est inférieur à la capacité et la pénalité correspondante non nulle alors on diminue cette pénalité jusquà ce qu'elle soit nulle ou que le flot soit égal à sa capacité et on recommence.

L'avantage pratique de ces méthodes de résolution est qu'elles peuvent aboutir à des algorithmes conversationnels de résolution. Etant donné un vecteur $\mu \geq 0$ de pénalité, on regarde la solution de l'affectation exponentielle correspondante ; parmi les arcs qui sont violés par les contraintes de capacité on peut en choisir un et en augmentant la pénalité correspondante, on contraint les usagers qui l'empruntent à subir un péage dissuasif qui fera baisser le trafic sur cet arc ; si on a un arc qui n'est pas saturé et dont la pénalité n'est pas nulle, en diminuant celle-ci, on va influencer les usagers à emprunter cet arc.

Le fait qu'on puisse ainsi traiter le problème, arc par arc, est dû à ce que, le vecteur μ étant fixé, la solution optimale est unique.

Au contraire, en programmation linéaire si on a un système qui se découpe en deux blocs :

$$\begin{cases} A_1 X = b_1 \\ \qquad\qquad X \geq 0 \\ A_2 X = b_2 \\ \min \ \langle c, X \rangle \end{cases}$$

Si aux contraintes du système $A_2 X = b_2$, on associe un vecteur de pénalité μ et qu'on résoud le programme linéaire

$$\begin{cases} A_1 X = b_1 \qquad X \geq 0 \\ \min \langle c, X \rangle + \langle \mu, b_2 - A_2 X \rangle \end{cases}$$

alors en général le système peut avoir plusieurs solutions, ce qui empêche d'utiliser des techniques similaires.

CHAPITRE III - PROBLEME DU MULTIFLOT

III.A. DEFINITION DU PROBLEME

Nous exposons dans ce chapitre les méthodes de l'affectation exponentielle utilisées pour le problème du multiflot. On sait que celui-ci peut se résoudre comme un programme linéaire avec un nombre important de variables et de contraintes.

En général, il n'existe pas de bonnes méthodes de résolution du multiflot parce que la matrice n'a aucune particularité, et on est obligé d'utiliser des méthodes quelconques qui ne tiennent pas compte de la structure particulière du problème.

Pour le problème du multiflot, si on ajoute la fonction entropie, on se place dans le cadre déjà étudié ; on peut, au contraire, avoir de bonnes méthodes de résolution. En effet, l'aspect particulier du problème se retrouve naturellement dans ces techniques ; et surtout le fait de pouvoir regarder les contraintes une par une va rendre l'algorithme intéressant car simple et de type conversationnel.

On suppose ici qu'il a été sélectionné parmi les sommets du réseau k couples (s_j, p_j) $j = 1,\ldots,k$ et qu'entre chaque source s_j et puits p_j, on fait circuler un flot d'un produit j d'intensité F_j. De plus, on suppose que chaque arc e de E est soumis à une contrainte de capacité supérieure S(e).

Soit $x^j(e)$ représentant la quantité de produit j qui circule sur l'arc e. Les contraintes décrivant un vecteur de multiflot sont données par k sytèmes du type P_j.

$$y_j(s_j) = \sum_{e\in\Omega^+(s_j)} x^j(e) = \sum_{e\in\Omega^-(s_j)} x^j(e) + F_j$$

$$y_j(v_i) = \sum_{e\in\Omega^+(v_i)} x^j(e) = \sum_{e\in\Omega^-(v_i)} x^j(e) \quad \forall\, v_i \in V$$

P_j :

$$y_j(p_j) = \sum_{e\in\Omega^+(v_i)} x^j(e) + F_j = \sum_{e\in\Omega^-(v_i)} x^j(e)$$

$$x^j(e) \geq 0 \quad \forall\, e \in E$$

Les contraintes de capacités sont données par les inégalités

$$x(e) = \sum_{j=1}^{k} x^j(e) \leq S(e) \quad \forall\, e \in E$$

Soit P_M le polyèdre décrit par ces systèmes, nous posons le problème (L_λ^M) :

$$(L_\lambda^M) : \begin{cases} X \in P_M \\ Z_\lambda^M = \min Z_\lambda^M(X) = \min \{ \sum_{j=1}^{k} (\sum_{e\in E} c(e)\, x^j(e) + z_\lambda^j(x^j) \} \end{cases}$$

Théorème III.1

a - La solution optimale X_λ^M de (L_λ^M) est unique, pour $\lambda > 0$.

b - La fonction $\lambda \to (X_\lambda^M, Z_\lambda^M)$ est continue pour $\lambda > 0$.

c - Quand λ tend vers 0, X_λ^M tend vers X_0^M solution optimale du problème du multiflot.

Démonstration : Celle-ci est toujours la même que celle utilisée dans [12].

III.B. RESOLUTION DU PROBLEME (L^M)

Sans nous étendre longuement sur la démonstration qui est la même que celle du problème précédent, énonçons brièvement la méthode qu'on peut utiliser. Introduisons les contraintes de capacités de chaque arc e dans la fonction objective avec une pénalité $\mu(e_o)$; le problème dual est :

$$Z_\lambda^M = \underset{\mu \geq 0}{\text{Max}} \; \underset{\substack{X \in O \\ X \in P_M}}{\text{Min}} \left\{ \sum_{j=1}^{k} \sum_{e \in E} (c(e)+\mu(e))X^j(e) + Z_\lambda^j(X^j) - \sum_{e \in E} S(e)\mu(e) \right\}$$

On peut découper le minimum qui est demandé par rapport à X, en plusieurs minima, chacun par rapport à X^j puisque le problème étant écrit ainsi, les X^j sont indépendants entre eux.

$$Z_\lambda^M = \underset{\mu \geq 0}{\text{Max}} \left\{ \sum_{j=1}^{k} \underset{X^j \in P_j}{\text{Min}} \left\{ \sum_{e \in E} (c(e)+\mu(e))x^j(e) + Z^j(X^j) \right\} - \sum_{e \in E} S(e)\mu(e) \right\}$$

Ainsi, si μ est fixé, on est amené à résoudre les problèmes d'affectations exponentielles pour chaque produit $j = 1,\ldots,k$. Pour connaître donc les vecteurs solutions $X_\lambda^j(\mu)$, il suffit de connaître une seule matrice, la matrice β_μ inverse de $[I-A_\mu(\lambda)]$ (cf. chapitre I). Si μ n'est pas solution optimale, on va augmenter notre fonction dans une direction e_o qui n'est pas un optimum local. Les théorèmes classiques d'Analyse Convexe nous apprennent que, de la même façon, dans le chapitre précédent, le maximum sera ateint pour μ tel que :

si $x^o(e_o) - S(e_o) > 0$ alors $x_\mu(e_o) = S(e_o)$

si $x^o(e_o) - S(e_o) < 0$ alors

- soit $x_\mu(e_o) = S(e_o)$ si $\mu(e_o) \geq 0$
- soit $\mu(e_o) = 0$ sinon.

La différence essentielle pour calculer μ avec le chapitre précédent est qu'on ne peut plus trouver μ en résolvant une équation algébrique du second degré. La méthode employée, dans les essais faits sur Pascaline ou Multics, est une approche de la solution optimale en remarquant que si le flot qui passe par l'arc e_o est supérieur à sa capacité on doit augmenter la pénalité $\mu(e_o)$ et s'il est inférieur on doit au contraire l'abaisser tout en gardant la contrainte de pénalité positive ou nulle

CHAPITRE IV

CAS DE FONCTIONS EMISSIONS-ATTRACTIONS
CAS DE CONTRAINTES DE CAPACITES INFERIEURES

IV.A. PRESENTATION DES PROBLEMES

Nous nous intéressons dans ce chapitre à deux problèmes connus dans les cas d'affectation de trafic, mais nous nous bornerons seulement à décrire succintement les techniques employées.

Le premier cas est le problème du flot avec une hypothèse supplémentaire que sur chaque noeud i du réseau, il existe un flot entrant ou un flot sortant de valeur σ_i. Pour que ceci ait une signification, on suppose que la somme des flots entrant est égale à la somme des flots sortant.

La méthode pour résoudre ce problème est d'introduire une "super-source" S et un "super-puits" P, reliés respectivement à toutes les sources et tous les puits i par un arc de coût nul et de capacité $|\sigma_i|$. Ainsi on est ramené au cas de contraintes supérieures.

Le deuxième cas est le problème de flot à capacités supérieures tel qu'il a été étudié aux chapitres précédents auquel on rajoute des contraintes de capacités inférieures.

La méthode de résolution est simplement de faire un changement d'origine sur le vecteur de flot. On ramène ainsi ce problème à un problème d'émissions-attractions à capacité supérieure tel qu'il a été étudié au chapitre IV auquel on rajoute des contraintes de capacités inférieures sur chaque arc.

CHAPITRE V - QUELQUES RESULTATS

Pour tester, ou plutôt pour avoir une idée de l'efficacité de ces méthodes, un code a été écrit sur deux ordinateurs : PASCALINE et le HB-68 du C.I.C.G., en langage Pascal. Celui-ci résoud les problèmes de multiflot avec capacités supérieures uniquement. Le programme est de type conversationnel et permet à l'utilisateur de choisir le paramètre λ de résolution, de choisir à chaque itération l'arc sur lequel on augmente notre fonction objective ou de laisser le libre choix de celui-ci au programme pendant un certain nombre d'itérations. D'autre part, comme l'algorithme décrit plus est un algorithme se rapprochant des méthodes d'Analyse Numérique, il ne faut pas espérer atteindre la solution optimale des problèmes dans tous les cas.

A cette condition de ne pas chercher précisément cette solution mais de vouloir seulement une idée, relativement nette, de celle-ci, les méthodes de l'affectation exponentielle se traitent relativement mieux que d'autres connues pour le problème du multiflot.

Toutefois, les difficultés, a priori, sont les suivantes :

- Problème de précision (ou calcul des exponentielles très petites).
- Choix de la valeur du paramètre λ.
- Si à partir du résultat donné pour l'affectation exponentielle on veut connaître une solution du problème de coût minimum, il va se poser deux problèmes : l'affectation exponentielle charge tous les itinéraires d'un certain trafic. Ceci n'est pas le cas pour le problème de coût minimum. Il faudra donc éliminer les circuits et les chemins qui ne sont pas dans la solution optimale de coût minimum.

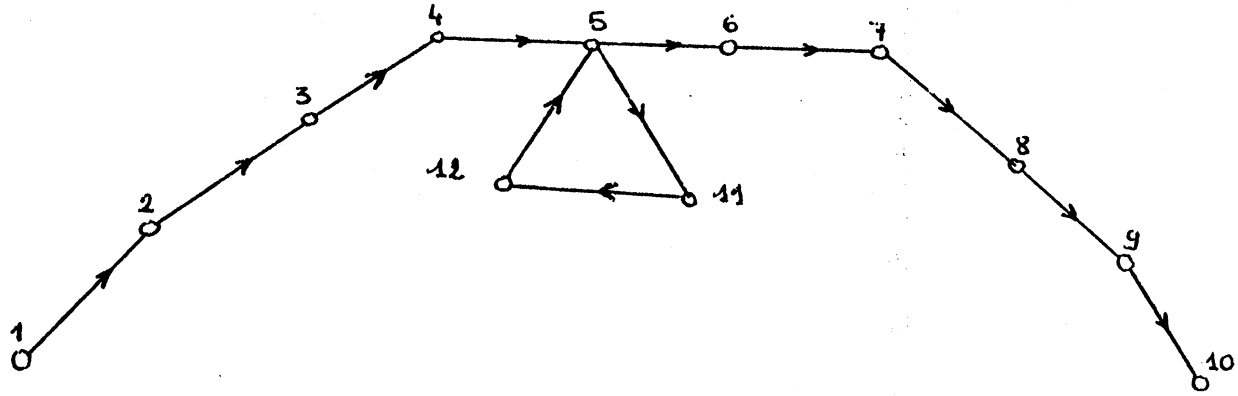
V.1. PROBLEMES DE CIRCUITS

Si on suppose que dans notre graphe il existe un circuit C de longueur ℓ, l'affectation exponentielle de paramètre λ (sans contrainte de capacités) impose plusieurs trafics sur ce circuit. Ainsi il y aura un flot égal à $\mu \exp(-\ell/\lambda)$ que fera une fois le tour, puis $\mu \exp(-2\ell/\lambda)$ qui fera deux fois le tour... .

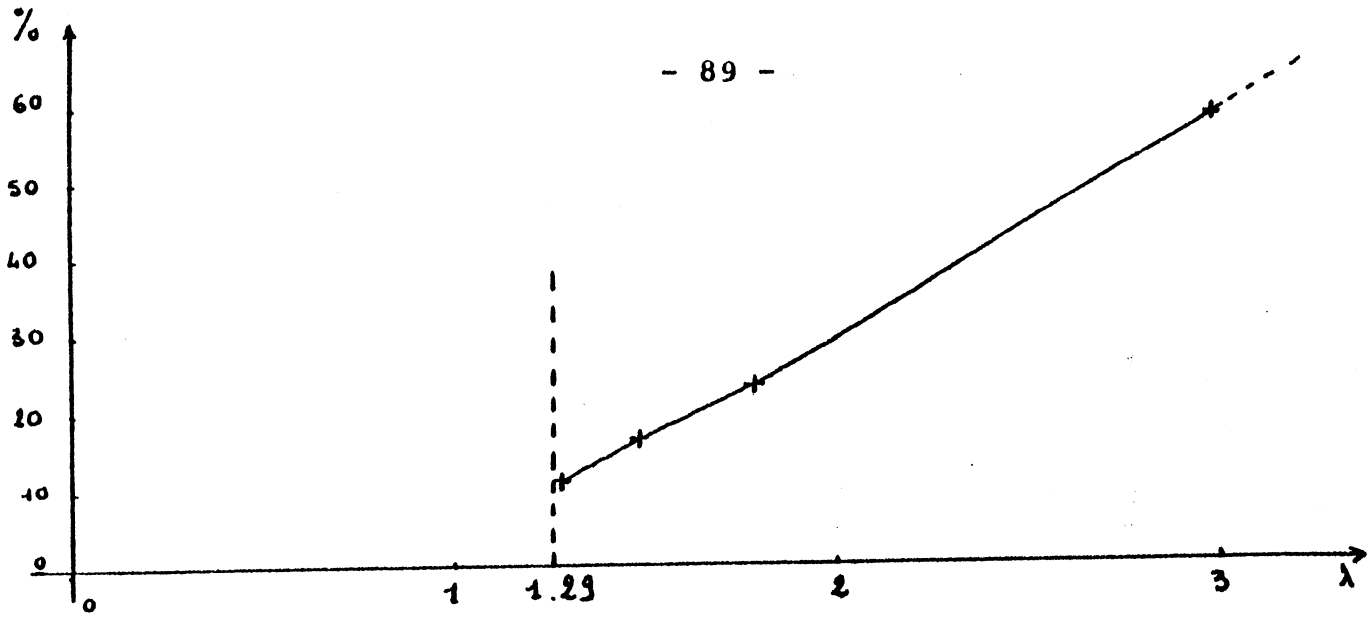
Sur ce circuit circulera donc un flot d'intensité I :

$$I = \frac{\exp(-\ell/\lambda)}{1 - \exp(-\ell/\lambda)}$$

Il apparaît ainsi que si le paramètre λ est trop grand, ce flot apportera une perturbation conséquente dans notre solution optimale. Pour illustrer ceci, nous avons choisi l'exemple suivant :



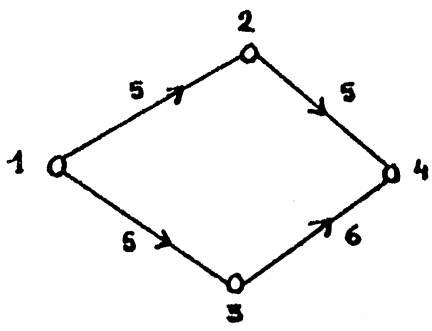
Tous les arcs ont une capacité de 10. Ceux du chemin élémentaire 1 à 10, un coût de 5 et ceux du circuit (5, 11, 12, 5) un coût de 1. On fait circuler entre 1 et 10 un flot d'intensité 9 pour ne pas avoir de problème de capacité. Si nous traçons le graphe du pourcentage de flot circulant dans le circuit en fonction du paramètre λ, nous obtenons :



Nous verrons plus loin pourquoi la courbe s'arrête à 1.29 dans l'axe des abscisses. Notre circuit a une longueur de 6,6% vis à vis du chemin le plus court et comme on peut le constater nous ne pouvons qu'espérer au minimum, un flot de 11% du flot total sur ce circuit. Toutefois, dans des exemples plus académiques, les circuits ont une longueur qui reste en rapport étroit avec les plus courts chemins, et comme nous le verrons dans les exemples suivants, les circuits s'éliminent facilement dans l'affectation exponentielle.

V.2. PROBLEMES DE DIFFERENCES DE CHEMINS

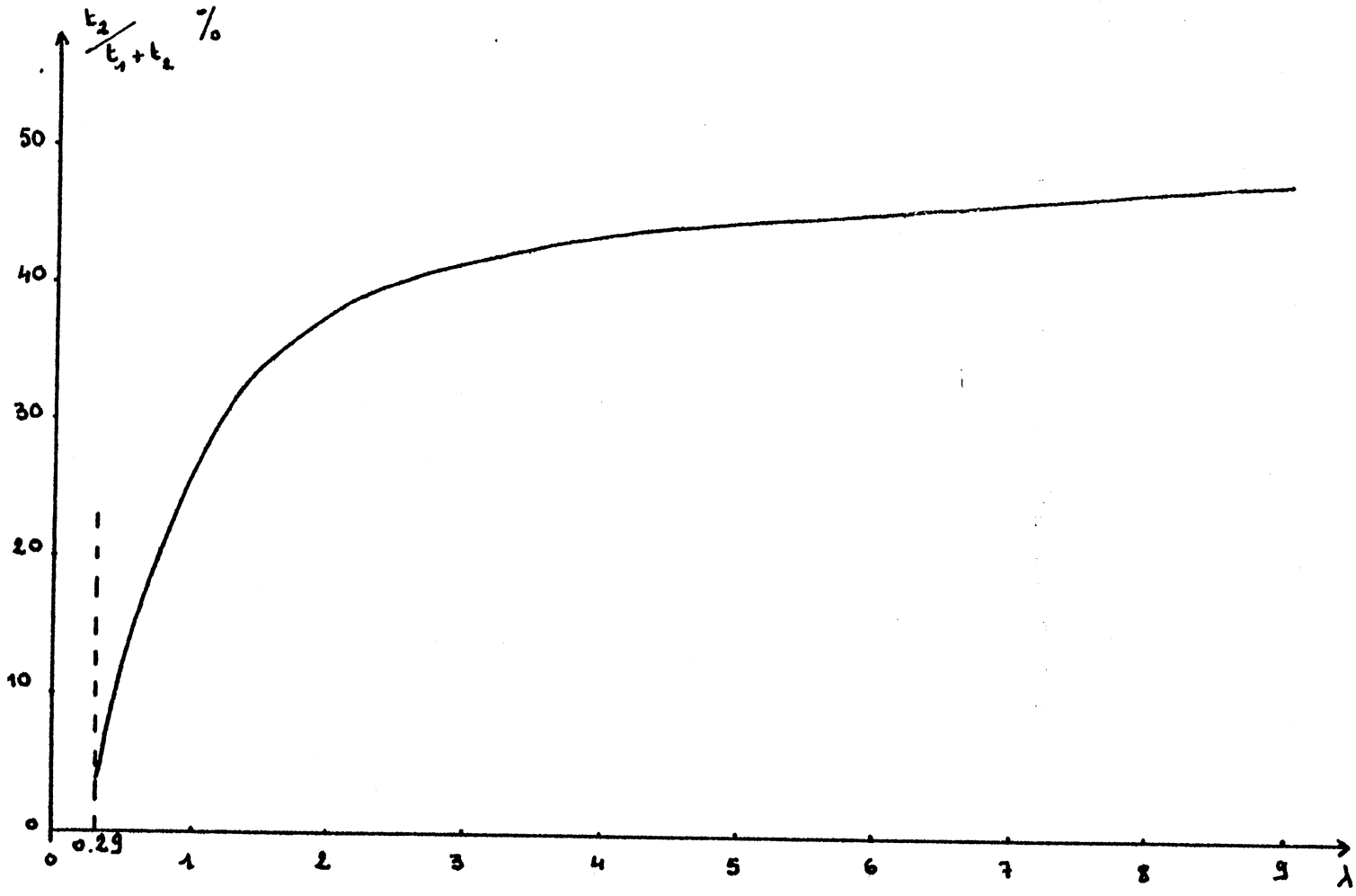
De la même façon, que s'il y a un circuit tel que la différence des chemins avec ou sans emprunter ce circuit soit petite, s'il existe plusieurs chemins qui aient des longueurs peu différentes, l'affectation exponentielle va charger les chemins les plus longs alors qu'à l'optimum ils ne devraient pas l'être. Citons l'exemple suivant, très simple :



$t_1 = (1,2,4)$

$t_2 = (1,3,4)$

Si on représente la fonction du pourcentage de flot passant par le chemin t_2 vis à vis du flot total, en fonction du paramètre λ, nous aurons :



Il suffit de comparer les deux courbes précédentes pour comprendre qu'il sera beaucoup plus difficile de "séparer" les chemins par ordre de grandeur que d'annihiler les effets des circuits.

V.3. PROBLEMES DE PRECISION

La première réaction vis à vis de l'affectation exponentielle est la méfiance puisque, par rapport au problème du plus court chemin, on perd l'intégrité des solutions optimales. Mais il faut voir celle-ci comme étant une approche, rapide, de la

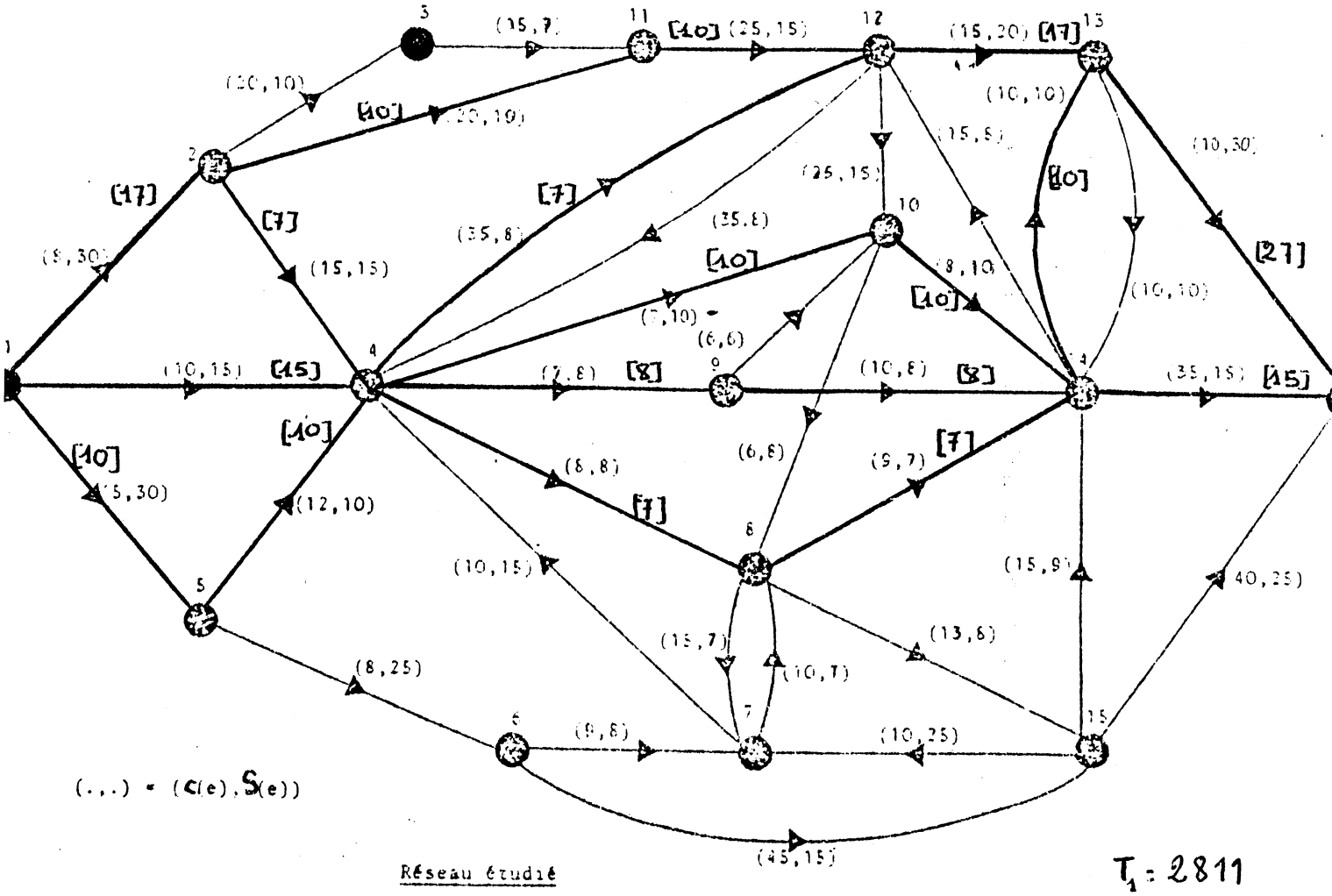
solution optimale. Elle demande la présence d'un cerveau plus évolué qu'un ordinateur pour "sentir" une solution de flot entière, au vu de solutions réelles. De ce fait, on voudrait prendre une valeur du paramètre λ très petite. Malheureusement, on a rapidement des problèmes "d'underflow" (10^{-38} accepté sur MULTICS) et des problèmes de précision (que se passe-t-il quand on inverse une matrice avec des coefficients si petits ?). On peut se définir une nouvelle origine (si x est plus petit que epsilon alors x égal 0) mais cela revient à interdire complètement certains arcs. Ainsi la valeur minimum du paramètre λ dépendra essentiellement du plus fort coût sur l'ensemble des arcs, du nombre de sommets de la longueur totale du chemin le plus court.

V.4. QUELQUES EXEMPLES

V.4.1. Problèmes de flot

L'exemple de base sur lequel nous avons travaillé est présenté dans [30].

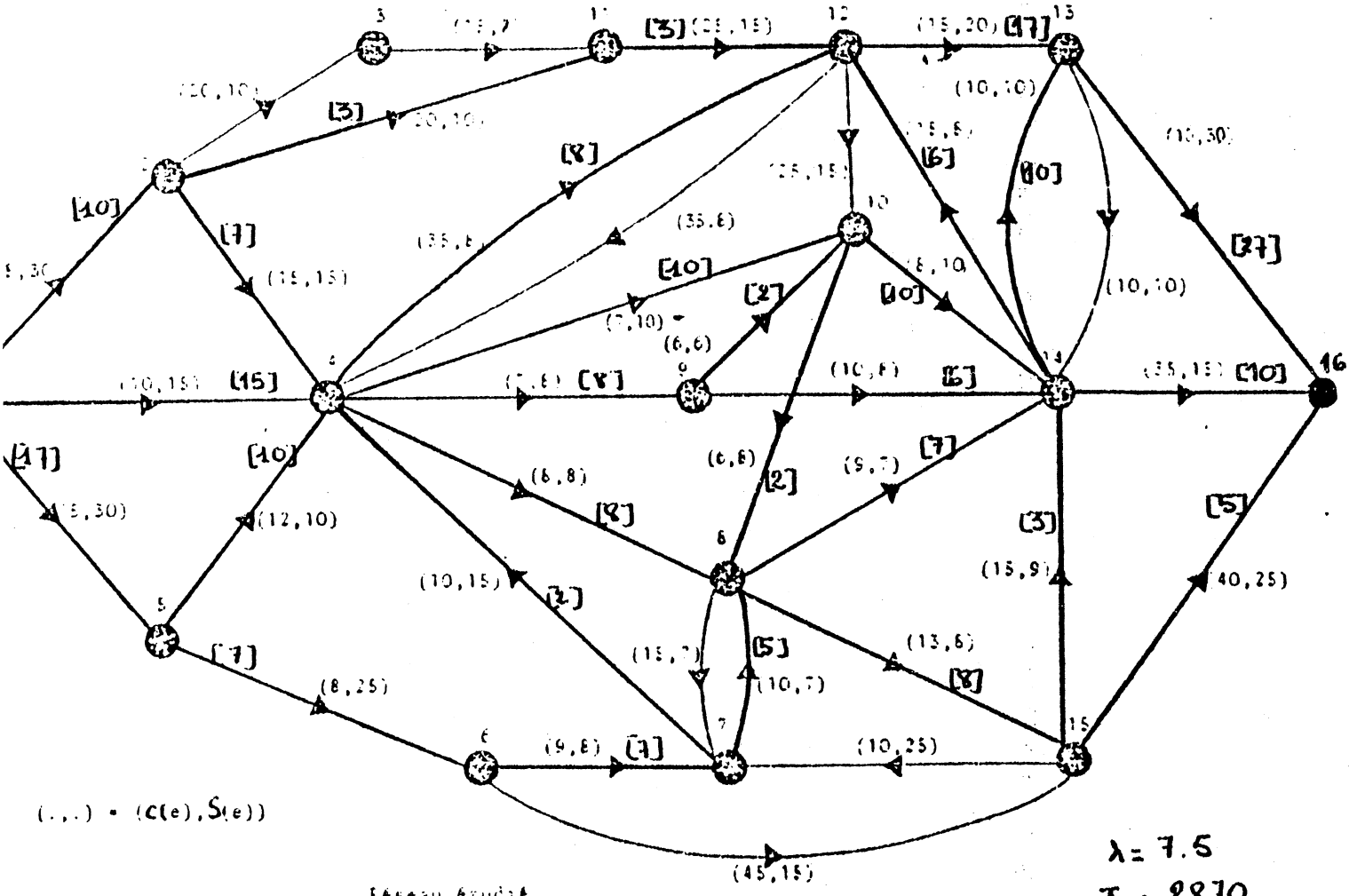
Nous présentons tout d'abord la solution optimale de flot de coût minimum, pour un flot entrant au sommet 1, d'intensité 42 et sortant en 16



Réseau étudié

$T_1 = 2811$

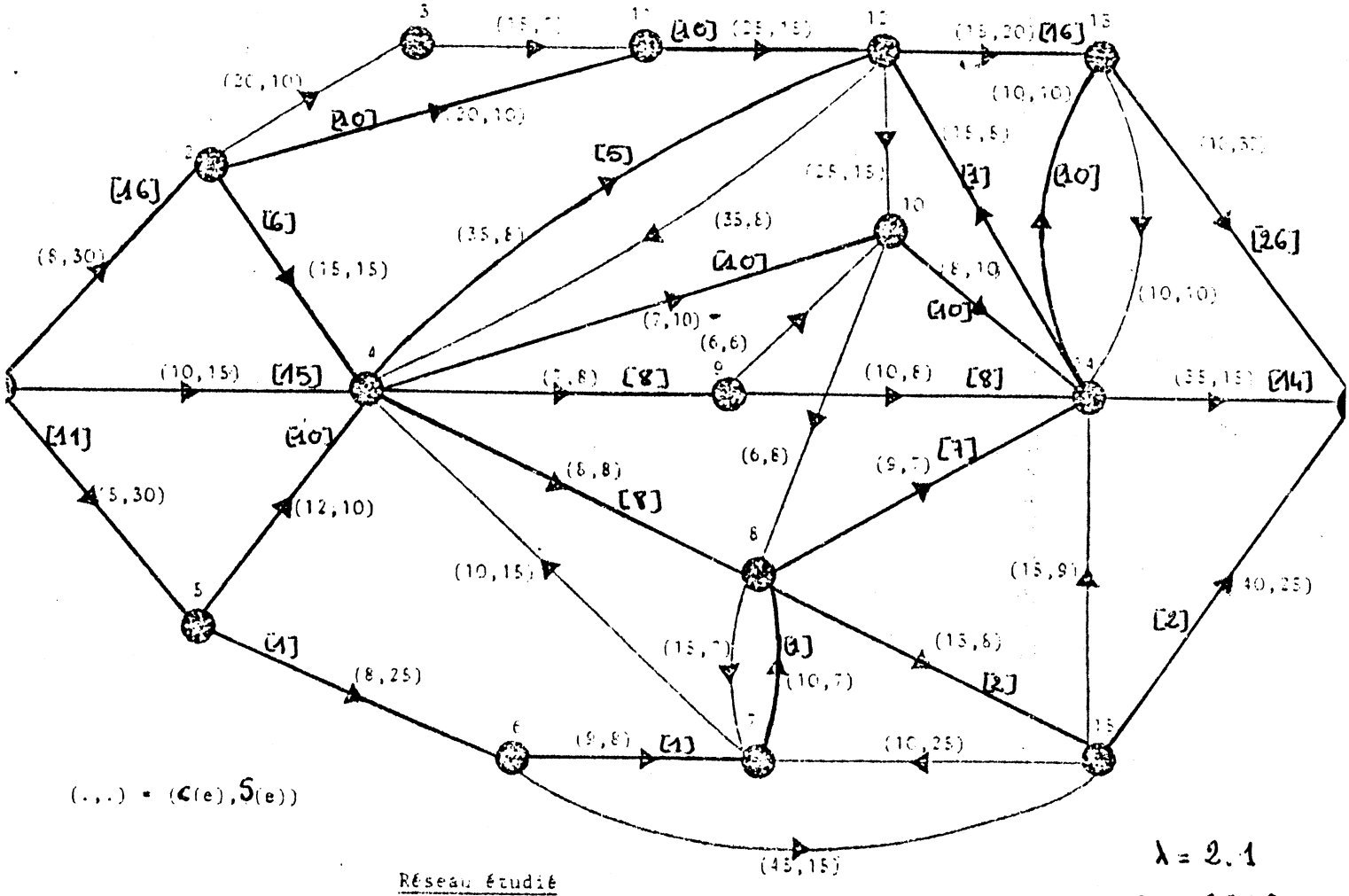
Nous allons d'abord évalué l'importance de la valeur du paramètre λ. Nous donnons seulement les valeurs entières du flot que nous proposons vis à vis des solutions rélles données par le programme.



Réseau étudié

Comme on peut le constater, si nous ne sommes pas très loin en valeur de la solution (à 2,8%) par rapport à la valeur optimum, la solution trouvée ne donne pas une idée de ce qu'est la solution optimale. Mais il est remarquable que pour cette valeur de λ, il n'y a aucun flot qui circule sur un circuit.

Si nous baissons la valeur de λ, nous nous rapprochons beaucoup plus de la solution.



Réseau étudié

La solution fournie se trouve à 0,3% de la solution optimale. D'autre part, il est facile de voir où se situent les "perturbations". Il y a tout d'abord un flot d'intensité 1 qui circule sur le chemin (1,5,6,7,8,15,16) qui a une longueur de 85 alors qu'il pourrait circuler sur le chemin le plus long (1,2,4,12,13, 16) qui a une longueur de 83. Un autre flot d'intensité 1 circule sur le chemin (14,12,13,16) de longueur 40 au lieu d'aller directement sur l'arc (14,16) de longueur 35. Enfin un autre flot d'intensité 1 circule sur le chemin (4,8,15,16) de longueur 61 au lieu d'aller sur (4,12,13,16) de longueur 60. Ces remarques font ressortir les avantages que nous apporte cette méthode. Si, toutefois, nous ne sommes pas à l'optimum, nous n'en sommes pas très loin ; et on fait ainsi ressortir les chemins qui, n'étant pas dans la solution optimale, sont très proches des chemins les meilleurs.

Une autre réflexion que nous apporte cette méthode se situe sur les capacités des arcs. En effet, certains arcs sont tels que les contraintes de capacité qui nous sont données sont inopérantes pour notre solution. D'autres, par contre, se révèlent très importantes. Ainsi, pour toute valeur de λ proche de 2.1, ces arcs auront une pénalité importante. Ainsi dans l'ordre :

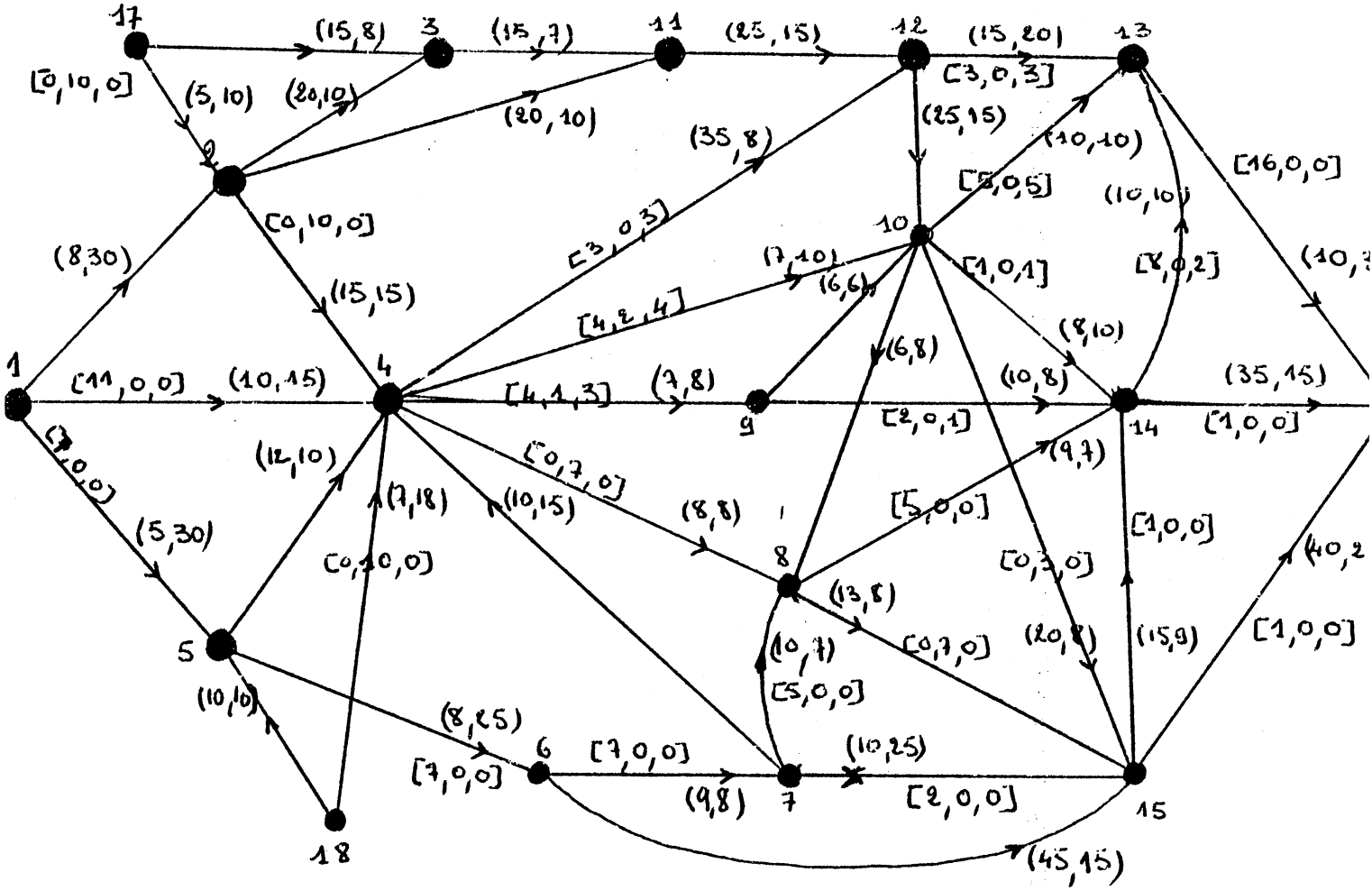
arc :	(14,15)	$\mu \simeq 16$
	(1, 4)	$\mu \simeq 11$
	(10,14)	$\mu \simeq 10$
	(8,14)	$\mu \simeq 8$
	(4, 9)	$\mu \simeq 8$
	(5, 4)	$\mu \simeq 5$
	(4, 8)	$\mu \simeq 1$

Il est remarquable aussi que ce système de pénalités qui sont un coût que l'on fait subir aux usagers qui empruntent ces arcs, définit sur chaque chemin un coût total dont la valeur est comprise entre 80 et 86. Ceci veut dire qu'on cherche à rendre les chemins de même longueur pour que la solution trouvée soit

une solution optimale du problème sans capacité mais avec ces nouveaux coûts $c(e) + \mu(e) = c'(e)$.

V.4.2 - Problèmes de multiflot

Nous présentons tout d'abord un problème de triflot sans circuit.



Un flot d'intensité 18 va du sommet 1 au sommet 16, un flot d'intensité 10 va du sommet 17 au sommet 15, et un flot d'intensité 10 va du sommet 18 au sommet 13.

La solution entière présentée ici a été obtenue pour la valeur de $\lambda = 2.25$, et elle a été obtenue après 39 itérations (une itération étant une amélioration du vecteur de pénalité). Il est remarquable que cette solution, qui n'est pas optimale, présente les mêmes problèmes que ceux rencontrés précédemment, en particulier dans le problème de différenciation des chemins par leurs longueurs et dans la découverte des arcs dont les contraintes de capacité sont génantes.

V.5. CONCLUSION SUR CES METHODES

Les méthodes issues de l'affectation exponentielle ne prétendent pas être des méthodes exactes de résolution de flot ou multiflot. Ce ne sont que des méthodes approchées mais rapides, où le facteur humain, l'interactivité de l'homme et la machine joue un grand rôle, afin d'obtenir une "bonne" idée de la solution optimale. Les problèmes rencontrés sont de plusieurs ordres :

- Différenciation des chemins vis à vis de leur longueur : si la différence est petite, il y aura trop de monde sur le chemin le plus long et si elle est trop grande, les problèmes d'exposant entraîneront une présence quasi-nulle sur des chemins intéressants.

- Problèmes de calculs : on travaille, du moins en théorie, sur des réels, avec des exponentielles.

- Problèmes d'appréciation : il faut, à partir des résultats réels obtenus, estimer des solutions de flots entières.

Toutefois, ces méthodes ont l'avantage d'être rapides par rapport à des méthodes équivalentes pour le multiflot à des problèmes de plus court chemin ou d'affectation de trafic. D'une part, on peut noter que ces méthodes ont un intérêt par elles-même car elles correspondent à des modèles utilisées en pratique (par exemple en théorie du trafic) et qu'elles ont déjà été appliquées au moins dans le cas où le graphe est sans circuit. D'autre part, si on désire obtenir une approximation

d'une solution d'un problème de flot de coût minimum, on a rapidement une bonne idée de celle-ci et en particulier une idée des arcs qui seront saturés dans cette solution.

- REFERENCES -

[1] . J.C. ANGLES D'AURIAC , B. DELAGENIERE , E. DUMAS , M. HUMBERT , B. RAPACCHI : " Etude d'un équilibre chimique " . ENSIMAG (1976-1977)

[2] C. BERGE : " Graphes et hypergraphes " . Dunod (1970) .

[3] V. CHVATAL : " Linear Programming " . W.H. Freeman & Co . (1980) .

[4] G.B. DANTZIG : " Linear programming and extensions " . Princeton Univ. Press . (1963) .

[5] R.B. DIAL : " A probabilistic multipath traffic assignement model which obviates path enumeration " . Transp. Res. Vol 5 (1971) p. 83-111

[6] J. EDMONDS : " Paths , trees and flowers " . Canad. J. Math. 17 (1965) p. 449-467 .

[7] J. EDMONDS : " Systems of distinct representatives and linear algebra". Journal of Research of the National bureau of Standards . B 71B (1967). p 241-245 .

[8] J. EDMONDS , R.M. KARP : " Theorical improvements in algorithmic efficiency for network flow problems " . Journal of the Association for Computing Machinery . 19 (1972) . p. 248-264.

[9] S. ERLANDER. : " Accesibily , entropy and the distribution and assignement of traffic " . Trans. Res. Vol 11 (1977) p. 149-153 .

[10] S. ERLANDER : " Entropy in linear programs " . Math. Prog. 21 (1981) p. 137-151 .

[11] J. FONLUPT : "L'affectation exponentielle et le problème du plus court chemin dans un graphe " . Séminaire d'Analyse Numérique n° 275 Mathématiques Appliquées . USMG .

[12] J. FONLUPT : " L'affectation exponentielle et le problème du plus court chemin dans un graphe " . RAIRO 2 (1981) . 165 - 184 .

[13] J. FONLUPT , B. RAPACCHI : " Programmation logarithmique et bons algorithmes " . Groupe Combinatoire de l'AFCET . (1980) .

[14] J. FONLUPT , B. RAPACCHI : " Uses logarithmic programming in Linear Programming " . Mathematishe Optimierung (1981) .

[15] P. GACS , L. LOVASZ : " Khachijan's algorithm for linear programming" Math. Prog. Studies . A paraître .

[16] M.R. GAREY , D.S. JOHNSON : " Computers and Intractability : a guide to the theory of NP-completeness . Freeman , San Francisco . (1979).

[17] M. GROTSCHEL , L. LOVASZ , A. SCHIJVER : " The Ellipsoïd method and its consequences in combinatorial Optimization " . A paraître dans Combinatorica 2 , (1981) .

[18] L.G. KHACHIJAN : " A polynomial algorithm in linear programming " . Soviet Mathematics Doklady 20 (1979) 191-194 .

[19] P.J. LAURENT : " Approximation et Optimisation " . Hermann (1972).

[20] L. LOVASZ : " A new linear programming algorithm - Better or worse than the simplex method ? "

[21] J.F. MAURRAS : " Bons algorithmes , vieilles idées " . Note EDF-HR 32.0320 (1978) .

[22] J.F. MAURRAS , K. TRUEMPER , M. AKGUL : " Polynomial algorithms for a class of linear programs " . Math. Prog. 21 (1981) n° 2 121-136 .

[23] F. ROBERT : " Matrices non-négatives et normes vectorielles " . Cours polycopié . ENSIMAG . USMG (1973) .

[24] R.T. ROCKAFELLAR : " Convex Analysis " . Princeton Univ. Press (1970).

[25] M. SAKAROVITCH , J.D. MacALLISTER : " Classification de certaines matrices en (0,1) " . Disc. Math. 20 (1977) 133-141.

[26] J.R. WESTLAKE : " A han book of numerical matrix inversion and solution of linear equations " . J. Wiley (1968).

[27] A.G. WILSON : " A statistical theory of spatial distribution models ".

[28] J.F. MAURRAS : Communication orale (1981) .

Dernière page d'une thèse

VU

Grenoble, le 15 Decembre 1981

Le Président de la thèse

Vu, et permis d'imprimer,

Grenoble, le 23.12.81

Le Président de l'Université Scientifique et Médicale