

Enseignement d'approfondissement  
Optimisation de la tarification  
des services voix d'un opérateur  
de télécommunication

Khaled SELLAMI et Lê Nguyễn HOANG  
Encadré par Moustapha BOUHTOU, Matthieu CHARDY et Frédéric BONNANS

Décembre 2009

# Introduction

Au cours de son stage de recherche à Orange, Khalid Mehl, X 2006, encadré par Moustapha Bouhtou et Matthieu Chardy, chercheurs à Orange Labs, a modélisé le marché des services voix d'un opérateur voix [3]. Il en est arrivé à déterminer une expression du profit en fonction du prix à optimiser sous contraintes. Cependant, cette expression est non-convexe, et les contraintes également. Il s'agit donc d'un problème difficile.

L'objet de cet Enseignement d'Approfondissement est la résolution de ce problème d'optimisation. Khalid Mehl a utilisé des solveurs classiques basés entre autres sur la méthode des points intérieurs ou sur la méthode du gradient réduit. Cependant, la non-convexité du problème conduit à des extrema locaux et non globaux.

En utilisant une approximation polynomiale du modèle par le "Finite-Mixture Logit Model", puis en le réécrivant pour appliquer la méthode des moments proposée par J. B. Lasserre dans [2], il en est arrivé à trouver des solutions qui, dans 73% des cas, se comporte mieux que la méthode du recuit simulé.

Frédéric Bonnans, professeur en mathématiques appliquées à l'Ecole Polytechnique, membre du CMAP et de l'INRIA, a alors suggéré une résolution par la méthode des intervalles, explicitée par Eldon Hansen et William Walster dans [1], et utilisée par Frédéric Messine dans [4]. Nous avons donc essayé de résoudre le problème d'optimisation par cette méthode. Celui-ci étant difficile, nous nous sommes restreints au problème sans contrainte.

Après avoir rappelé les grands principes de cette méthode et ajouté des outils qui nous seront utiles, nous avons rappelé le problème d'optimisation en étudiant comment le réécrire pour appliquer l'optimisation par intervalles. Enfin, nous présenterons les résultats de cette méthode.

# Table des matières

<b>1</b>	<b>Optimisation par l'analyse des intervalles</b>	<b>5</b>
1.1	L'analyse des intervalles . . . . .	5
1.1.1	Introduction à l'analyse des intervalles . . . . .	5
1.1.2	Opérateur moyenne sans contrainte . . . . .	6
1.1.3	Opérateurs moyenne avec contraintes . . . . .	8
1.2	Implémentation . . . . .	9
1.2.1	Implémentation de l'arithmétique des intervalles . . . . .	9
1.2.2	La méthode <b>largesse</b> . . . . .	10
1.2.3	Architecture algorithmique . . . . .	10
1.2.4	Les bases de la méthode <b>optimisation</b> . . . . .	11
1.2.5	Ajouts à la procédure <b>optimisation</b> . . . . .	11
1.3	Application de la <i>Hull consistency</i> . . . . .	12
1.3.1	Principe . . . . .	12
1.3.2	Cas multidimensionnel . . . . .	13
1.3.3	Extension de la <i>Hull Consistency</i> . . . . .	13
1.3.4	Implémentation de la <i>Hull consistency</i> . . . . .	14
<b>2</b>	<b>Problème d'optimisation</b>	<b>15</b>
2.1	Ecriture du problème . . . . .	15
2.1.1	Rappel du modèle . . . . .	15
2.1.2	Expression du profit . . . . .	16
2.1.3	Quelques notations . . . . .	17
2.2	Calcul différentiel . . . . .	17
2.2.1	Différentielles des nombres de consommateurs des offres . . . . .	17
2.2.2	Différentielles des coûts . . . . .	18
2.2.3	Différentielles du profit . . . . .	18
2.2.4	Remarques et commentaires . . . . .	19
2.3	Calcul par intervalles . . . . .	20
2.3.1	Expressions efficaces des nombres de consommateurs . . . . .	20
2.3.2	Calcul par intervalles de différentes variables . . . . .	20
2.3.3	Calcul des termes du profit . . . . .	21
2.3.4	Calcul de $p_{o'} \partial_{o'} n_{o'}$ . . . . .	21
2.3.5	Calcul des termes de la différentielle . . . . .	22
2.4	Les fonctions de réduction . . . . .	23
2.4.1	Problème du revenu de détail . . . . .	23
2.4.2	Inversion de $p$ . . . . .	24
2.4.3	Inversion de $n_o$ . . . . .	24

2.4.4	Inversion de $\partial_o n_o$ . . . . .	24
2.4.5	Réduction par $\partial_o n_o$ . . . . .	24
<b>3</b>	<b>Résultats et commentaires</b> . . . . .	<b>27</b>
3.1	Tests de notre algorithme . . . . .	27
3.1.1	Les exemples de Khalid Mehl . . . . .	27
3.1.2	Un exemple en dimension $n$ . . . . .	28
3.2	Estimation de l'efficacité . . . . .	29
3.2.1	Choix et estimation des données du problème . . . . .	29
3.2.2	Calculs de moyenne . . . . .	30
3.3	Calcul numérique . . . . .	30
3.3.1	Les . . . . .	30

# Chapitre 1

## Optimisation par l'analyse des intervalles

Dans leur ouvrage [1] d'Eldon Hansen et G. William Walser, les auteurs décrivent une série de procédures permettant de calculer des optima globaux, pour des fonctions étant de préférence dérivables presque partout. Cette méthode a été reprise et appliquée par Frédéric Messine dans [4]. Dans cet Enseignement d'Approfondissement, nous allons implémenter certaines de ces procédures pour déterminer une optimisation algorithmique de notre problème.

### 1.1 L'analyse des intervalles

Les algorithmes décrits dans les ouvrages cités font intervenir différentes techniques se fondant sur l'analyse des intervalles. Nous allons d'abord tracer les contours de cette analyse, avant d'énoncer quelques propositions qui serviront à la résolution de notre problème d'optimisation.

#### 1.1.1 Introduction à l'analyse des intervalles

Pour avoir une définition claire et concise des intervalles, nous allons introduire l'ensemble  $\mathbb{R}^*$  des bornes des intervalles.

**Définition 1** *On étend l'ensemble des réels à ses bords, en définissant  $\mathbb{R}^* = \mathbb{R} \cup \{-\infty, +\infty\}$ , muni de la relation d'ordre canonique, et des opérations internes usuellement définissables.*

Ces opérations n'étant pas tout à fait définies, il ne s'agit pas vraiment de lois de compositions internes. Cependant, elles peuvent être tout à fait définies pour l'ensemble des intervalles.

**Définition 2** *On définit l'ensemble des intervalles  $\mathbb{IIR} = \{[a, b]; a, b \in \mathbb{R}^*, a \leq b\}$ , muni des lois de composition internes classiques, les formes indéfinies renvoyant l'élément  $\mathbb{R}^* = [-\infty, +\infty]$ . On munit également cet ensemble de la relation  $X \leq Y$  (resp.  $X < Y$ ) si  $\forall x \in X, \forall y \in Y, x \leq y$  (resp.  $x < y$ ). On notera aussi  $\mathbb{IIR}_+$  (resp.  $\mathbb{IIR}_{++}$ ) l'ensemble des intervalles positifs (resp. strictement positifs), c'est-à-dire vérifiant  $X \geq 0$  (resp.  $X > 0$ ).*

L'extension des lois de composition internes classiques se fait en prenant comme résultat le plus petit intervalle contenant tous les produits réalisables avec chacune des bornes des éléments à composer. Les formes indéfinies correspondent à  $0 \times \infty$  et  $\infty - \infty$ .

L'intérêt de ces intervalles est de pouvoir écrire des fonctions de  $\mathbb{IIR}^n$  dans  $\mathbb{IIR}$  correspondant "bien" à des fonctions à valeurs réelles usuelles. On introduit donc la définition suivante, et on énonce la proposition évidente qui suit.

**Définition 3** Une fonction  $f^I : \mathbb{IIR}^n \rightarrow \mathbb{IIR}$  est une *extension* d'une application  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , si  $\forall X \in \mathbb{IIR}^n$ ,  $f^I(X) \supset f(X) = \{f(x); x \in X\}$

**Proposition 1** Toute composition de lois de compositions internes usuelles (addition, soustraction, multiplication, division) et d'application monotone (dont le résultat est le plus petit intervalle contenant l'image des bornes) est l'expression d'une extension.

Cependant, on aimerait avoir mieux qu'une simple inclusion. D'où la définition suivante.

**Définition 4**  $f^I$  est une extension *efficace* (ou *expression efficace*) de  $f$ , si  $\forall X$ ,  $f^I(X) = f(X)$ .

La difficulté ne réside donc pas dans la recherche d'extension, mais plutôt dans celle d'expressions efficaces. Dans de nombreux cas, on se contentera d'applications assez efficaces, c'est-à-dire pour lesquelles l'écart entre  $f^I(X)$  et  $f(X)$  n'est pas trop grand. On cherchera à éliminer des cas tels que  $X - X$ , que l'on souhaiterait pouvoir dire qu'il est égal à  $\{0\}$ , mais dont la largeur est en fait le double de celle de  $X$ . Pour étudier l'efficacité d'extensions, on dispose des définitions et de la proposition suivantes.

**Définition 5** Deux variables réelles  $x$  et  $y$  sont dites *indépendantes* si on peut prendre n'importe quelle valeur pour  $y$  à  $x$  fixé. Deux variables intervallistes  $X$  et  $Y$  sont *indépendants* si le choix d'un  $y$  dans  $Y$  peut toujours se faire, quelque soit le choix de  $x$  dans  $X$ .

**Proposition 2** Si une expression n'est composée que de lois de compositions internes usuelles et d'application monotone, et fait apparaître au plus une fois chacune des variables, et si ces variables sont indépendantes, alors il s'agit d'une expression efficace.

Cette proposition peut être démontrée par induction structurelle. Mais cette preuve est assez longue. C'est pourquoi, on ne l'explicitera pas. Rajoutons cette définition.

**Définition 6** On notera  $X \ominus Y$  la solution  $Z$  de l'équation intervalliste  $Y + Z = X$ . Cette expression n'est définie que si la largeur de  $X$  est supérieure à celle de  $Y$ , et elle prend alors la valeur  $[\min X - \max Y, \max X - \min Y]$ . Elle correspond à l'extension efficace de  $x - y$  si  $x = y + z$  où  $z$  est indépendant de  $y$ .

Pour une meilleure efficacité dans les calculs à venir, on va maintenant prolonger l'étude faite par Hansen à des cas que l'on va rencontrer.

### 1.1.2 Opérateur moyenne sans contrainte

Dans l'expression du profit, interviennent de nombreuses moyennes pondérées, comme nous le verrons dans le chapitre suivant. Par conséquent, définissons l'opérateur moyenne.

**Définition 7** Soit  $\mathcal{M} : \mathbb{IIR}^K \times \mathbb{IIR}_+^K \rightarrow \mathbb{IIR}$ , défini par  $\mathcal{M}(A, X) = (\sum A_k X_k) / (\sum X_k)$  calculé de façon efficace, c'est-à-dire  $\mathcal{M}(A, X) = \{(\sum a_k x_k) / (\sum x_k); a \in A, x \in X\}$  l'opérateur moyenne, qu'on notera parfois  $\mathcal{M}_K$ ,  $\mathcal{M}_{k \in K}$ , voire  $\mathcal{M}_k$  pour éviter toute ambiguïté. Le vecteur  $A$  est le vecteur des valeurs, et le vecteur  $X$  celui des poids.

Rappelons l'existence de l'injection canonique de  $\mathbb{R}$  dans  $\mathbb{IR}$ , qui permet de définir  $\mathcal{M}$  dans le cas où  $A$  et/ou  $X$  sont à valeurs réelles. Par ailleurs, on admettra que  $X \geq 0$  et  $X \neq 0$ , c'est-à-dire  $\forall k, \min X_k \geq 0$  et  $\exists k_0, \min_{k_0} X_{k_0} > 0$ . Enfin si  $a \in \mathbb{R}^K$ , et  $\exists k \in K, x_k = +\infty$ , on définira  $\mathcal{M}(a, x)$  comme étant le plus petit intervalle contenant tous les  $a_k$  pour  $k$  vérifiant  $x_k = +\infty$ .

**Définition 8** On définit  $x^{max}(i, t)$  (resp.  $x^{min}(i, t)$ ) le vecteur de  $X \in \mathbb{IR}^n$  pour lequel on a

1.  $\forall j < i, x_j^{max} = \min X_j$  (resp.  $x_j^{min} = \max X_j$ ), ie toute petite coordonnée est minimisée.
2.  $\forall j > i, x_j^{max} = \max X_j$  (resp.  $x_j^{min} = \min X_j$ ), ie toute grande coordonnée est maximisée.
3.  $x_i^{max} = t$  (resp.  $x_i^{min} = t$ ), ie la coordonnée de séparation prend la valeur  $t$ .

On ajoutera de plus les définitions de  $x^{max}(i) = x^{max}(i, \max X_i)$  et  $x^{min}(i) = x^{min}(i, \min X_i)$ , ainsi que  $x^{max} = x^{max}(n)$  et  $x^{min} = x^{min}(n)$

On a alors le lemme suivant, qui va nous permettre d'en déduire l'algorithme de calcul efficace de l'opérateur moyenne.

**Lemme 1** Supposons les  $(a_k^{max})_{k \in K}$  ordonnés par ordre croissant, ie  $\forall k, a_k^{max} \leq a_{k+1}^{max}$ . Alors il existe  $i_{max} \in K$  tel que  $\max \mathcal{M}(A, X) = \mathcal{M}(a^{max}, x^{max}(i_{max}))$

On a un lemme similaire en remplaçant chacune des apparitions du mot "max" par "min".

### Démonstration du lemme 1

D'abord, remarquons que  $\mathcal{M}$  est une fonction croissante des  $a_k$ , et le maximum sera donc nécessairement atteint en  $a^{max}$ . De plus,  $\mathcal{M}(a^{max}, x^{max}(i, t))$  est une fonction croissante de  $t$  si et seulement si pour un quelconque  $t_0 \geq 0$ , on a  $\mathcal{M}(a^{max}, x^{max}(i, t_0)) < a_i^{max}$ . En effet, en augmentant  $t$ , la moyenne se rapproche de  $a_i^{max}$ , sans le dépasser.

Supposons pour l'instant  $X$  borné, alors le maximum est atteint sur ce compact en un élément  $x$ . Si la moyenne est alors à gauche (resp. à droite) d'un  $a_i^{max}$ , la monotonie vue précédemment implique que  $x_i$  est à son maximum (resp. minimum). En prenant  $i$  tel que  $a_i^{max}$  le premier à droite de la moyenne, on voit que le maximum est atteint en  $x^{max}(i)$ .

Si maintenant  $X$  n'est pas borné, considérons  $i$  la plus grande valeur pour laquelle  $X_i$  n'est pas borné. Le maximum ne peut pas être strictement inférieure à  $a_i^{max}$ , car la moyenne égale à  $a_i^{max}$  en  $x^{max}(i)$ . S'il est strictement supérieur à  $a_i^{max}$ , la monotonie vue précédemment nécessite que pour tout  $j \leq i, x_j$  est minimum. On se ramène alors au cas précédent, et on en déduit l'existence d'un maximum atteint en  $x^{max}(i')$  pour  $i' > i$ . CQFD.

**Proposition 3** L'opérateur moyenne peut être calculé efficacement à l'aide d'un algorithme dont la complexité temporelle est quadratique.

Explicitons un tel algorithme de  $\mathcal{M}(A, X)$ . Commençons par la borne supérieur. On commence par trier les  $a^{max}$ , qu'on notera simplement  $a$ . Supposons-les triés. On cherche maintenant à déterminer  $i_{max}$ . On le choisit pour commencer au milieu de  $K$ , par exemple  $\lfloor K/2 \rfloor$ . Ensuite,

1. Si  $a_{i_{max}} < \mathcal{M}(a, x^{max}(i_{max}))$ , on incrémente  $i_{max}$  et on reboucle.
2. Si  $\mathcal{M}(a, x^{max}(i_{max})) < a_{i_{max}-1}$  et si  $X_{i_{max}-1}$  n'est pas borné,  $i_{max} - 1$  convient.
3. Si  $\mathcal{M}(a, x^{max}(i_{max})) < a_{i_{max}-1}$ , on décrémente  $i_{max}$  et on reboucle.
4. Si  $a_{i_{max}-1} \leq \mathcal{M}(a, x^{max}(i_{max})) \leq a_{i_{max}}$ ,  $i_{max}$  convient et  $\max \mathcal{M}(a, X) = \mathcal{M}(a, x^{max}(i_{max}))$ .

On détermine  $i_{min}$  et  $\min \mathcal{M}(A, X)$  de façon similaire.

Il y a au plus  $K/2$  itérations, avec un calcul de moyenne de complexité linéaire, d'où la complexité quadratique. Il en est de même pour le tri de  $a^{max}$ . Avec un tri-fusion, et avec un calcul plus précis du nombre d'itération, on doit pouvoir montrer que la complexité est en fait meilleure.

### 1.1.3 Opérateurs moyenne avec contraintes

Nous verrons des cas où on peut ajouter une contrainte aux poids des opérateurs moyenne. On introduit donc la définition suivante.

**Définition 9** Définissons  $\mathcal{M}(A, X | \sum X = x_{total}) = \{(\sum a_k x_k) / (\sum x_k); \sum x_k = x_{total}\}$  et  $\mathcal{M}(A, X | \sum X \in X_{total}) = \{(\sum a_k x_k) / (\sum x_k); \sum x_k \in X_{total}\}$  les opérateurs moyenne sous contrainte d'égalité et d'inclusion sur les poids, calculés de manière efficace.

De la même manière que dans la partie précédente avec l'opérateur moyenne sans contrainte, nous allons chercher des lemmes qui serviront à l'écriture d'un algorithme de calcul efficace.

**Lemme 2** Supposons les  $a^{max}$  triés. Alors il existe  $i_{max}$  tel que le maximum de la moyenne avec contrainte d'égalité est atteint en  $x^{max}(i_{max}, t)$  où  $t = x_{total} - \sum_{j \neq i_{max}} x_j^{max}(i_{max}) \in X_{i_{max}}$ .

#### Démonstration du lemme 2

L'espace des variables admissibles est un compact puisque la contrainte d'égalité borne cet ensemble fermé. Le maximum est donc atteint en un vecteur  $x$ . Si deux coordonnées ne sont pas à des bornes de leurs intervalles de définition, on peut rajouter  $\epsilon$  à l'une et retirer à l'autre tout en restant dans l'espace des variables admissibles.

Si les valeurs  $a^{max}$  en ces coordonnées sont différentes, alors la fonction de  $\epsilon$  ainsi obtenue est strictement monotone et on peut faire mieux que la moyenne en  $\epsilon = 0$ .

Si ces valeurs sont égales, on peut prendre le plus grand  $\epsilon$  conduisant à un vecteur toujours admissible, l'existence d'un tel  $\epsilon$  venant de la positivité de  $X$ . Un tel  $\epsilon$  amène l'une des coordonnées à sa borne, et on diminue alors strictement le nombre de coordonnées non saturées.

On en déduit le lemme 2.

On a un résultat similaire pour le minimum. Pour déterminer l'algorithme de calcul efficace de la moyenne sous contrainte d'égalité, il nous suffit maintenant d'utiliser le lemme suivant.

**Lemme 3** Soit  $Z(X) = \{(i, -t) \in K \times \mathbb{R}_-; t \in X_i\}$ , muni de l'ordre lexical. Alors l'application  $z : Z(X) \rightarrow \mathbb{R}, (i, -t) \mapsto t + \sum_{j \neq i} x_j^{max}$  est décroissante.

La démonstration de ce lemme est facile et ne sera pas explicitée. La solution du problème de maximalisation de l'opérateur de moyenne avec contrainte d'égalité est alors atteinte en  $x^{max}(i_{max}, t)$ , où  $(i_{max}, t)$  est une solution de l'équation  $z(i_{max}, t) = x_{total}$ . Une méthode dichotomique de recherche du  $i_{max}$  peut être utilisée en comparant les  $z(i, \max X_i)$ ,  $i_{max}$  vérifiant  $z(i_{max} - 1, \max X_{i_{max}-1}) \leq x_{total} \leq z(i_{max}, \max X_{i_{max}})$ . Par la suite, on peut déterminer  $t = x_{total} - \sum_{j \neq i_{max}} x_j^{max}(i_{max})$ . La complexité est celle du tri, car le reste est linéaire.



Venons en maintenant à la construction de l'algorithme efficace de l'opérateur moyenne avec contrainte d'inclusion. Pour cela, nous allons utiliser le lemme suivant, quelque peu similaire au lemme 2.

**Lemme 4** *Supposons les  $a^{max}$  triés. Si le maximum sans contrainte n'est pas un point admissible, alors il existe  $i_{max}$  tel que le maximum de la moyenne avec contrainte d'inclusion est atteint en  $x^{max}(i_{max}, t)$  où  $t = x_{total} - \sum_{j \neq i_{max}} x_j^{max}(i_{max}) \in X_{i_{max}}$  et  $x_{total}$  est une borne de  $X_{total}$ .*

#### Démonstration du lemme 4

Supposons que le maximum sans contrainte n'est pas un point admissible, et que le maximum avec contrainte est atteint en un point où la somme des coordonnées est à l'intérieur de  $X_{total}$ .

Si le nombre de coordonnées non saturées est nul, les remarques de monotonie montrent que ce point est nécessairement de la forme  $x^{max}(i)$ . Comme il ne s'agit pas du maximum sans contrainte, on a soit  $\mathcal{M}(a, x^{max}(i)) < a_{i-1}$ , soit  $a_i < \mathcal{M}(a, x^{max}(i))$ . Dans le premier cas, on a intérêt à augmenter le poids de  $a_{i-1}$ , dans le second, à diminuer celui de  $a_i$ , et on augmente strictement la moyenne, ce qui est absurde.

Supposons que la coordonnée  $i$  est non saturée. Si  $a_i$  est égale à la moyenne, alors, soit on peut la saturer en conservant la moyenne et en se ramenant ainsi au cas précédent, soit on peut saturer la contrainte d'inclusion. Sinon,  $\mathcal{M}(x^{max}(i, t + \epsilon))$  est strictement monotone, et n'atteint pas son maximum en  $\epsilon = 0$ , ce qui est absurde.

Pour déterminer une expression efficace de la moyenne avec contrainte d'inclusion, on teste donc d'abord si le maximum sans contrainte est admissible. Si c'est le cas, il convient. Sinon on résout le problème avec contrainte d'égalité, en prenant comme valeur de la somme les bornes de  $X_{total}$ .

Là encore, on traite le problème du minimum de façon similaire. Nous avons ainsi obtenu des algorithmes de calcul efficace qui vont nous permettre de résoudre le problème d'optimisation avec une meilleure performance. Voyons maintenant comment implémenter nos algorithmes.

## 1.2 Implémentation

Nous allons implémenter notre résolution algorithmique sur Java. Ce langage de programmation va nous permettre de définir des classes correspondant notamment aux ensembles  $\mathbb{R}^* = [-\infty, +\infty]$ ,  $\mathbb{IIR} = \{[a, b]; a \in \mathbb{R} \cup \{-\infty\}, b \in \mathbb{R} \cup \{+\infty\}, a \leq b\}$  ou encore  $\mathbb{IIR}^n$ .

### 1.2.1 Implémentation de l'arithmétique des intervalles

Nous définissons la classe `RStar` dont les attributs seront `String inf` et `double valeur`. Ainsi  $+\infty$  sera défini par `inf = "+"`, et  $-\infty$  sera défini par `inf = "-"`. Un réel sera défini par `inf = "real"`, et par sa  $+\infty$  sera défini par `valeur`. Ce champ sera défini en `private`, c'est-à-dire qu'il ne sera pas directement accessible. Pour y avoir accès, il faut utiliser la méthode `real()`, qui renverra une exception si `inf` n'est pas égal à `"real"`. On définit aisément les opérations arithmétiques classiques de cette classe, et renverra des exceptions pour les opérations non définies telles que  $+\infty - \infty$  ou  $0 \times \pm\infty$ .

Nous définissons aussi la classe `Interval` défini par `boolean vide`, `RStar min` et `RStar max`. Les deux derniers sont encore en `private`, pour éviter des manipulations ambiguës. Nous définissons l'addition comme  $X + Y = [\min X + \min Y, \max X + \max Y]$ , et la multiplication comme

le plus petit intervalles contenant les valeurs  $(\min X)(\min Y)$ ,  $(\min X)(\max Y)$ ,  $(\max X)(\min Y)$  et  $(\max X)(\max Y)$ . S'il y a une exception levée, on renverra l'intervalle  $[-\infty, +\infty]$ . Rajoutons la définition de la différence exclusive  $X \ominus Y$  par  $[\min X - \max Y, \max X - \min Y]$ , défini uniquement si la largeur de  $X$  est supérieure à celle de  $Y$ .

On définira également les classes `RVect` (resp. `IRVect`) correspondant à des tableaux `RStar` (resp. `Interval`) qui permettront d'avoir une meilleure lisibilité des calculs. On définira également les opérations usuelles sur ces éléments.

### 1.2.2 La méthode largesse

Dans le cadre de l'utilisation de la méthode des intervalles, il est primordial d'avoir des fonctions intervallistes qui représentent de véritables extensions des fonctions réelles conformément à la définition 3. Or, une simple définition de l'addition comme vue précédemment n'assure pas l'extensivité des fonctions. En effet, les approximations dues à la représentation finie des réels conduiraient à des erreurs.

Rappelons la représentation en Java des `double` utilisés pour représenter les réels. En gros, il s'agit de réels de la forme d'un décimal (un nombre de 52 chiffres en base 2) compris entre 1 et 2, multiplié par une puissance de 2. Ainsi, la précision sur la valeur d'un `double`  $x$  est de  $x \times 2^{-52} \approx 10^{-15}$ .

Dans la classe `RStar`, nous rajoutons les méthodes `largesseInf` et `largesseSup` qui, appliqué à un réel, renvoient respectivement le plus grand réel représentable par un `double` strictement inférieur à l'argument et le plus petit strictement supérieur. Pour un réel  $x$ , il s'agit donc de renvoyer  $x(1 - 2^{-52})$  et  $x(1 + 2^{-52})$ .

On rajoute également la méthode `largesse` de la classe `Interval` qui renvoie un intervalle élargi des deux côtés, en appliquant `largesseInf` et `largesseSup` aux bornes inférieures et supérieures de l'intervalle. Cette méthode permet de bien définir la somme ou la multiplication d'un intervalle avec un `RStar` ou un `double`, et d'autres opérations, conformément à l'extension des fonctions. Ainsi, par exemple, on posera  $\exp(X) = \text{largesse}([\exp(\min X), \exp(\max X)])$ .

Notons qu'un élément de `RStar`  $x$  représenté dans le cadre de l'arithmétique des intervalles doit en fait être un intervalle contenant la valeur exacte de  $x$ . Le plus petit de ces intervalles représentable par notre classe `Interval` est alors l'intervalle  $[x.\text{largesseInf}(), x.\text{largesseSup}()]$ .

### 1.2.3 Architecture algorithmique

La classe abstract `Optimisation` regroupera toutes les optimisations intervallistes que l'on fera. N'y sont définies que des méthodes `abstract resolution` qui renvoient l'entière procédure de résolution. Elles s'appliquent à des intervalles ou des vecteurs d'intervalles de la forme  $[a, b]^n$ , où  $a, b \in \mathbb{R}^*$ .

Nous allons maintenant définir les deux classes `abstract OptimisationReelle` et `abstract OptimisationVectorielle` qui héritent de la classe `Optimisation`. La première de ces classes, la classe `OptimisationVectorielle` possède en plus un champ correspondant à la dimension. Ces classes possèdent des méthodes `simplification`, `HullConsistency` et `optimisation`, les

deux premières étant utilisées dans la troisième pour calculer le maximum global.

Ces classes possèdent également des méthodes abstraites **f** et **h** qui permettent de calculer l'image d'un point ou d'un intervalle par  $f$ , et d'utiliser la *Hull Consistency* pour réduire les intervalles de recherche du maximum. Pour trouver le maximum d'une fonction, il faut maintenant créer une classe héritant de l'une des deux et y définir les méthodes **f** et **h** correspondant aux méthodes abstraites ci-dessus.

#### 1.2.4 Les bases de la méthode optimisation

Décrivons maintenant rapidement les méthodes *optimisation*. On se donne donc une fonction **f** et une méthode **HullConsistency**. On définit aussi une liste d'intervalles **l**, qui contient initialement uniquement l'intervalle  $X$  sur lequel on veut maximiser la fonction. On pose aussi **minorantMax** comme l'image du milieu de l'intervalle considéré par  $f$ . Tant que cette liste ne contient pas un unique intervalle  $X$  dont la longueur est inférieure à  $\epsilon \max_{x \in X} |x| = \epsilon_X > 0$ , on retire son premier élément  $X_1$ . Si le maximum de l'image de  $X_1$  par la fonction intervalliste **f** est inférieur au **minorantMax**, on le supprime. Sinon, on lui applique la méthode **HullConsistency**.

1. Si un intervalle  $X$  de longueur inférieure à  $\epsilon_X$  est renvoyé, c'est que celui-ci contient un extremum local bien localisé. Avant de le rajouter en fin de liste, on actualise le **minorantMax**, en calculant  $f$  au milieu de cet intervalle et on reboucle.
2. Si un intervalle  $X$  de longueur inférieure à  $\epsilon_X$  est renvoyé, c'est que la *Hull Consistency* a quelque peu stagné et atteint un intervalle  $X'_1$ . On sépare alors  $X'_1$  en divisant l'une de ses coordonnées prises au hasard en deux parties égales. On obtient ainsi deux intervalles de longueur strictement inférieure, que l'on rajoute en fin de liste. On actualise **minorantMax** comme précédemment et on reboucle.
3. Enfin, si cette méthode lève une exception, c'est qu'il n'y a pas d'extrema dans l'intervalle en question (voir 2.2.1), et on supprime donc cet intervalle. On reboucle.

Si la liste contient un unique intervalle de longueur inférieure à  $\epsilon_X$ , c'est qu'il contient l'unique maximum de la fonction sur  $X$ , et qu'il permet de très bien le localiser. Si la liste est vide à un moment, c'est que le maximum est atteint sur les bords. On renverra alors une erreur. Remarquons qu'en activant l'une des contraintes, on fixe une variable et on pourrait se ramener à un problème d'optimisation de dimension inférieure. On peut donc résoudre le problème en résolvant  $2n$  problèmes de dimension inférieure, si  $n$  est la dimension.

Il s'agit de l'algorithme suggérée par Hansen et Walston. Cependant, on a rencontré deux types de problèmes, alors qu'on l'a testé pour notre problème 3 (voir partie 3.1.2). Nous avons donc rajouté quelques procédures.

#### 1.2.5 Ajouts à la procédure optimisation

En effet, d'une part le nombre d'éléments de la liste risque d'être explosé. S'il est vrai, que, dans le pire des cas, l'algorithme termine malgré tout, la largeur de chacun des éléments de la liste diminuant strictement, la longueur de cette liste peut exploser et ne jamais diminuer théoriquement, conduisant à l'impossibilité de localiser le maximum. Afin de soulever ce problème, on ajoute le critère d'arrêt suivant : si, à un instant donné, la longueur de la liste est supérieure à 10 000, une exception "**explosion de la liste**" est levée et le programme s'arrête sur un échec.

D'autre part, on s'est rendu compte qu'il arrivait que le nombre d'éléments stagne, sans qu'aucun de ces intervalles ne soit ni séparé ni éliminé. On a donc rajouté une variable `iterSansAvancee` que l'on incrémente quand un intervalle n'est ni séparé ni éliminé et que l'on annule dans le cas contraire. Quand cette variable atteint dix fois la valeur de la longueur de la liste, c'est que dix *Hull Consistency* sur chacun de ces intervalles n'a pas permis de résoudre le problème. On renverra alors cette liste. La résolution pourra afficher la réunion de ces intervalles.

Remarquons que le livre [1] suggère d'autres méthodes de réduction des intervalles de recherche, qui pourraient être utilisées à la place ou à la suite de la *Hull Consistency*, telle que la méthode de Newton ou la *Box Consistency*. Cependant ces deux autres procédures nécessitent le calcul des dérivées d'ordre supérieur, que l'on n'a pas cherché à calculer.

### 1.3 Application de la *Hull consistency*

Comme nous n'utiliserons pas les méthodes de Newton ou de *Box Consistency*, la *Hull Consistency* sera notre principale méthode de réduction des intervalles. Il est alors crucial de bien l'utiliser. Nous allons commencer à rappeler le principe de cette méthode, et expliquer son implémentation. Son application à notre problème sera explicitée dans le second chapitre.

#### 1.3.1 Principe

La *Hull consistency* est une méthode de résolution d'équations introduites par Hanson et Waslter dans leur livre. Elle se fonde sur la remarque suivante. Supposons que l'on veuille résoudre l'équation  $f(x) = 0$  pour  $x \in X$ . Alors, cela revient à résoudre  $h(x) = x$  où l'on a posé  $h(x) = f(x) + x$  par exemple, ou plutôt, et c'est bien le piège de l'algorithmique des intervalles, où on a réussi à "bien" écrire  $f(x) = h(x) - x$ . Notons que  $h(x)$  n'a pas à être forcément écrit ainsi, il faut juste exprimer de "bonne" manière une équation du type  $h(x) = x$ , équivalente à  $f(x) = 0$ . En fait, il suffit d'avoir l'implication  $f(x) = 0 \Rightarrow h(x) = x$ .

**Définition 10** Une fonction  $h$  telle que  $f(x) = 0 \Rightarrow h(x) = x$  est appelée fonction de réduction.

On a alors l'importante proposition suivante sur laquelle repose la *Hull Consistency*.

**Proposition 4** Soit  $h$  une fonction de réduction. Alors  $f^{-1}(0) \cap X = f^{-1}(0) \cap (X \cap h(X))$ .

#### Démonstration de la proposition 4

En effet, si  $x^* \in X$  est solution alors  $x^* = h(x^*) \in h(X)$ , et toute solution de l'équation est donc dans  $X \cap h(X)$ ,  $h(X)$  étant l'opération sur les intervalles. Dans de nombreux cas  $X \cap h(X)$  est strictement inclus dans  $X$ , ce qui permet d'avancer dans la localisation des solutions. Ajoutons que le théorème de Brouwer nous indique que si  $h(X) \subset X$ , alors il existe une solution.

Remarquons que si on arrive à écrire  $f(x) = a(x) - g(x)$  où  $g$  est inversible, alors  $h = g^{-1} \circ a$ . On remarquera alors qu'une condition nécessaire et suffisante de la réduction effective en passant par cette fonction  $h$  sur un intervalle  $X$  s'écrit  $a(X) \subset g(X)$ .

A partir de là, et tant qu'il semble que l'on réduit bien l'intervalle de recherche de solution, nous poserons  $X_1 = X \cap h(X)$ , et  $X_{n+1} = X_n \cap h(X_n)$ . Cependant, il arrivera de ne pas beaucoup avancer, par exemple lorsque la différence de longueur entre  $X_n$  et  $X_{n+1}$  n'est pas suffisamment grande. On utilisera alors une procédure de séparation de l'intervalle.

Remarquons que l'on peut avoir une famille de fonctions  $(h_i)$  de réduction pour résoudre une équation de type  $f = 0$ . Ainsi, si  $\forall i, \forall x, f(x) = 0 \Rightarrow h_i(x) = x$ , alors on peut réduire l'intervalle  $X$  en  $(\cap_i h_i(X)) \cap X$ .

### 1.3.2 Cas multidimensionnel

On a maintenant une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  à annuler. On cherche toujours à appliquer la *Hull Consistency*, afin de résoudre l'équation  $f(x) = 0$ . De la même manière, on introduit une famille de fonctions de réductions  $(h_j^i)_{1 \leq i \leq n, j \in J(i)}$ , qui doivent vérifier les implications  $\forall i, \forall j \in J(i), f(x) = 0 \Rightarrow x_i = h_j^i(x)$ .

En utilisant la *Hull Consistency*, on peut alors réduire l'intervalle de recherche de solution  $X = \prod_i X_i$  en l'intervalle  $(\prod_i (\cap_j h_j^i(X))) \cap X$ . Bien entendu, il sera souhaitable que pour tout  $i, J(i)$  soit non vide afin que la *Hull Consistency* réduisent toutes les coordonnées.

### 1.3.3 Extension de la *Hull Consistency*

Dans cette section, nous allons nous intéresser à des cas multidimensionnels où les fonctions de réduction ne peuvent pas être écrites explicitement. Ces cas, que nous rencontrerons dans notre projet, sont des cas que nous avons étudiés par nous-même.

Remarquons que les  $(h_j^i) : \mathbb{I}\mathbb{R}^n \rightarrow \mathbb{I}\mathbb{R}$ , telles que  $\forall i, j, \forall x \in X, f(x) = 0 \Rightarrow x_i \in h_j^i(X)$ , sont des fonctions de réduction. Ainsi, supposons que,  $\forall x \in X, f(x) = 0 \Rightarrow g(x) = g(x_i, x_{-i}) \in a(X)$ , où  $x_{-i} \in X_{-i} = \prod_{j \neq i} X_j$ . Alors  $g_{X_{-i}}^{-1}(a(X) \cap g(X)) = \{x_i \in X_i \mid \exists x_{-i} \in X_{-i}, g(x_i, x_{-i}) \in a(X) \cap g(X)\}$  vérifie l'implication précédente, et convient en tant que fonction de réduction. Remarquons que, lorsque le calcul de  $g$  est efficace par rapport à  $X_{-i}$ , une définition équivalente est l'équivalence suivante  $x_i \in g_{X_{-i}}^{-1}(a(X) \cap g(X)) \Leftrightarrow g(x_i, X_{-i}) \cap a(X) \neq \emptyset$ .

L'égalité entre l'ensemble théorique recherché et son calcul numérique étant dure à obtenir, on peut se contenter d'une inclusion. La difficulté peut alors résider dans le calcul numérique. Le cas simple est alors celui où  $g$  ne dépend pas de  $x_{-i}$  et où  $g$  est alors une fonction facilement inversible de  $x_i$ . Cependant, si  $g$  est strictement monotone, on peut là encore très bien s'en sortir grâce à le lemme suivant.

**Lemme 5** *Supposons que la fonction  $g$  soit strictement croissante et continue. Alors, on a  $g_{X_{-i}}^{-1}(Y \cap g(X)) = [g_{\max X_{-i}}^{-1}(\min Y \cap g(X)), g_{\min X_{-i}}^{-1}(\max Y \cap g(X))]$ , que l'on peut approcher extensivement par dichotomie.*

#### Démonstration du lemme 5

On remarque que si  $g$  est strictement croissante, alors son calcul est efficace. De plus, on a l'équivalence des propositions suivantes :

1.  $g(x_i, X_{-i}) \cap Y = \emptyset$ .
2.  $\exists x_{-i} \in X_{-i}, g(x_i, x_{-i}) \in Y$ .
3.  $\exists x_{-i} \in X_{-i}, \min Y \cap g(X) \leq g(x_i, x_{-i}) \leq \max Y \cap g(X)$ .
4.  $\min Y \cap g(X) \leq g(x_i, \max X_{-i})$  et  $g(x_i, \min X_{-i}) \leq \max Y \cap g(X)$ .
5.  $x_i \in [g_{\max X_{-i}}^{-1}(\min Y \cap g(X)), g_{\min X_{-i}}^{-1}(\max Y \cap g(X))]$ .

La seule implication non triviale est (4)  $\Rightarrow$  (3). Pour démontrer cela, on suppose (4) et on pose  $x_{-i}(\lambda) = \lambda \min X_{-i} + (1 - \lambda) \max X_{-i}$  et  $\varphi(\lambda) = g(x_i, x_{-i}(\lambda))$ . Alors  $\forall \lambda \in [0, 1], x_{-i}(\lambda) \in X_{-i}$ ,  $\varphi$  est croissante, vérifie  $\min Y \cap g(X) \leq \varphi(1)$  et  $\varphi(0) \leq \max Y \cap g(X)$ . Si  $\min Y \cap g(X) \leq \varphi(0)$ ,  $x_{-i}(0)$  convient. Sinon, par continuité,  $\exists \lambda \in [0, 1], \varphi(\lambda) = \min Y \cap g(X)$ , et  $x_{-i}(\lambda)$  convient.

### 1.3.4 Implémentation de la *Hull consistency*

La famille des fonctions de réduction  $h$  sera codée par une méthode `h`, qui, s'appliquant à un vecteur `IRVect`, renvoie un tableau de liste d'intervalles, c'est-à-dire un élément de la classe `LinkedList<Interval>[]`. Pour un vecteur  $X$  donné, on calcule donc les  $h_j^i(X)$  un à un. Ensuite, on les intersecte en utilisant le calcul par intervalle, et on déduit alors un élément  $X_1$  de `IRVect` égal à  $(X_1)_i = X_i \cap (\cap_{j \in J(i)} h_j^i(X))$ .

La méthode `HullConsistency` prendra un `IRVect` en argument, ainsi qu'un double `epsilon` strictement positif, de l'ordre de  $10^{-6}$ . Elle renvoie un autre `IRVect`. Elle utilise la méthode `h` pour trouver les  $h_j^i(X)$  et  $X_1$ . Ensuite, elle calcule la largeur de  $X_1$ , égale à la somme des largeurs des coordonnées de  $X_1$ , et la largeur de  $X$ . Trois cas se présentent.

1. si la largeur de  $X_1$  est inférieure à  $\epsilon_X$ , on renvoie  $X_1$ , dont on imaginera qu'il possède un zéro de  $f$  bien localisé.
2. si la somme de la largeur de  $X_1$  et de  $1000 \times \epsilon_X$  est inférieure à la largeur de  $X$ , c'est qu'on a bien avancé. Alors, on recommence en lançant récursivement la `HullConsistency` à  $X_1$  et  $\epsilon_X$ .
3. sinon, la somme de la largeur de  $X_1$  et de  $\epsilon_X$  est supérieure à la largeur de  $X$ , et on stagne. On renvoie alors  $X_1$ , qui sera ensuite séparé.

Il reste alors à déterminer les fonctions  $h_j^i$  correspondant à ce problème. Pour cela, il nous faut étudier notre problème un peu plus en détail pour vérifier quelles fonctions permettent une bonne réduction des intervalles, et rendent donc la *Hull Consistency* efficace.

# Chapitre 2

## Problème d'optimisation

Le problème d'optimisation porte sur la fixation des prix optimaux. La fonction objectif et ses contraintes ont été explicitées grâce à la modélisation de Khalid MEHL au cours de son stage de recherche chez Orange Labs, encadré par Matthieu CHARDY et Mustapha BOUHTOU. Ce dernier a utilisé des méthodes d'approximations polynomiales, en passant par la méthode moment pour trouver les solutions, qui dans plusieurs cas permet d'améliorer celle de solveurs standards, lesquels ne renvoyant que des optima locaux. En utilisant l'analyse des intervalles expliquées dans [?], nous allons chercher à trouver de meilleures solutions au problème.

### 2.1 Ecriture du problème

Nous rappelons que les variables d'optimisation sont les prix des offres. Il s'agit alors de maximiser le profit en fonction de ces prix. Le problème posé par Khalid MEHL introduit des contraintes d'inégalité, que l'on va rappeler. Cependant, nous nous attaquerons, en tout cas dans un premier temps, au problème sans contrainte.

#### 2.1.1 Rappel du modèle

Considérons un opérateur téléphonique, celui ci doit maximiser son revenu qui s'écrit de la façon suivante :  $Revenu = R_{detail} + R_{interco} - C^*$ .  $R_{detail}$  est le revenu qui provient de la vente des minutes de communication aux particuliers.  $R_{interco}$  est le revenu qui est lié aux interconnexions entre les autres opérateur : en effet, lorsque le client de l'opérateur A souhaite appeler un client de l'opérateur B, c'est B qui se charge de la terminaison de cet appel et est payé par A pour fournir ce service. Ainsi  $R_{interco}$  est la différence entre ce que l'opérateur reçoit de la part de la concurrence et ce qu'il leur paie.  $C^*$  représente le coût du réseau optimal qui répond à la demande.

Considérons  $O$  l'ensemble des offres  $o$  de notre opérateur et  $O_c$  celui des concurrents. On note  $n_o$  le nombre de clients qui souscrivent à l'offre  $o$ , et  $p_o$  son prix. On a  $R_{detail} = \sum_{o \in O} p_o n_o$ , où  $n_o$ , le nombre de clients pour l'offre  $o$ , est une fonction de leur appréciation pour cette offre.

Pour modéliser cette appréciation, nous pouvons décomposer l'ensemble des consommateurs  $D$  en différents segments, chaque segment regroupe des consommateurs qui pensent de la même manière selon (tranche d'âge, classe sociale, domaine de travail...).

Un segment  $d$  sera caractérisé par

1.  $N_d$  le nombre de ses consommateurs.
2.  $\alpha_d$  la sensibilité vis-à-vis au prix.
3.  $u_{d,o}$  l'appréciation de l'offre  $o$  sans prise en compte du prix.
4.  $\epsilon_d$  une variable aléatoire qui suit la loi de Gambel.

Ainsi nous pouvons montrer que le nombre de client choisissant l'offre  $o$  s'écrit

$$n_o = \sum_{d \in D} \left[ N_d \exp(u_{d,o} - \alpha_d p_o) / \left( \sum_{o \in O} \exp(u_{d,o} - \alpha_d p_o) \right) \right]$$

Tout opérateur téléphonique  $S$  est formé de plusieurs services. Par exemple, chez Orange, on trouve Orange mobile, RTC (fixe classique) et VOIP (fixe utilisant internet). Par ailleurs, toute offre rassemble un certain nombre de minutes de chaque service. Pour décrire la relation entre offres et services, on note :

1.  $m_{s,o}$  le nombre de minutes du service  $s$  incluses dans l'offre  $o$ .
2.  $n_s^{min} = \sum_{o \in O} m_{s,o} n_o$  le nombre de minutes fournies par  $s$ .
3.  $b_{s,o}$  la variable booléenne valant 1 si et seulement si  $o$  inclut  $s$ .
4.  $n_s^{cl} = \sum_{o \in O} b_{s,o} n_o$  le nombre de client associés au service  $s$

Le revenu  $R_{interco}$  d'un opérateur  $S$  est la différence entre le coût des appels entrants (que la concurrence rembourse). Faute de temps, nous n'avons pas expliqué ici son expression.

### 2.1.2 Expression du profit

L'écriture du problème d'optimisation est assez compliquée. Elle a été explicitée entièrement à la page 28 du rapport de Khalid MEHL. Nous allons utiliser des notations pour en avoir une forme plus condensée, qui sera utile au calcul par intervalles. Les coûts sont des fonctions affines qui s'écrivent de la manière suivante, pour  $\alpha \in (\mathbb{R}^{S \cup S_c})^2$  et  $\beta \in \mathbb{R}$ .

$$c(\alpha, \beta) = \beta + \frac{\sum_{s,s'} \alpha_{s,s'} n_s^{min} n_{s'}^{cl}}{\sum_{s''} n_{s''}^{cl}}$$

Ainsi,  $\ddot{c} = c(\ddot{\alpha}, \ddot{\beta})$ ,  $\dot{c} = c(\dot{\alpha}, \dot{\beta})$ ,  $\bar{c} = c(\bar{\alpha}, \bar{\beta})$ ,  $c^S = c(\alpha^S, \beta^S)$ ,  $c^{\dot{s}} = c(\alpha^{\dot{s}}, \beta^{\dot{s}})$  et  $c^{\ddot{s}} = c(\alpha^{\ddot{s}}, \beta^{\ddot{s}})$ .

On a les relations  $(\bar{\alpha}, \bar{\beta}) = \sum_{s \in S} (\alpha^{\dot{s}} + \alpha^{\ddot{s}}, \beta^{\dot{s}} + \beta^{\ddot{s}})$ ,  $(\ddot{\alpha}, \ddot{\beta}) = \sum_{s \in S} (\alpha^{\dot{s}}, \beta^{\dot{s}})$  et  $(\dot{\alpha}, \dot{\beta}) = \sum_{s \in S_c} (\alpha^{\ddot{s}}, \beta^{\ddot{s}})$ .

Les coûts s'interprètent comme des coûts d'imputation des opérateurs services  $s$  recevant un appel ( $c^{\dot{s}}$ ) ou en émettant un ( $c^{\ddot{s}}$ ).  $c^S$  est le coût total imputé à l'opérateur  $S$ . On remarque par ailleurs que le revenu d'interconnexion peut se décomposer en la différence du transfert d'argent de  $S_c$  vers  $S$  et du transfert inverse. Ainsi,  $R_{interco} = R_{S_c \rightarrow S} - R_{S \rightarrow S_c}$ .

On peut écrire  $R_{S_c \rightarrow S} = \sum_{s \in S_c, s' \in S} t_{s \rightarrow s'} f_{s \rightarrow s'}$  et  $R_{S \rightarrow S_c} = \sum_{s \in S, s' \in S_c} t_{s \rightarrow s'} f_{s \rightarrow s'}$ . Or on a  $t_{s \rightarrow s'} = \frac{c^{\dot{s}'}}{\bar{c}} \frac{c^S}{\sum_{s'' \in S \cup S_c} f_{s'' \rightarrow s'}}$ , et  $f_{s \rightarrow s'} = n_s^{min} \frac{n_{s'}^{cl}}{N^{cl}}$  où  $N^{cl} = \sum_{s \in S \cup S_c} n_s^{cl}$ . On en déduit alors



$\sum_{s \in \text{SUS}_c} f_{s \rightarrow s'} = \frac{n_s^{cl}}{N^{cl}} \sum_{o \in \text{OUO}_c} M_o n_o$  où  $M_o = \sum_{s \in \text{SUS}_c} m_{s,o}$ . Il vient  $R_{S_c \rightarrow S} = \frac{\ddot{c} \sum_{o \in O_c} M_o n_o}{\bar{c} \sum_{o \in \text{OUO}_c} M_o n_o} c^S$  et  $R_{S \rightarrow S_c} = \frac{\dot{c} \sum_{o \in O} M_o n_o}{\bar{c} \sum_{o \in \text{OUO}_c} M_o n_o} c^S$ . Ainsi,  $R_{interco} = \frac{\ddot{c}(\sum_{O_c} M_o n_o) - \dot{c}(\sum_O M_o n_o)}{\bar{c} \sum_{\text{OUO}_c} M_o n_o} c^S$ . Rajoutons le seuil de rentabilité de l'offre  $o$  défini par  $r_o = \frac{c^S}{\bar{c}} \sum_{s \in S} \frac{m_{s,o} c^s}{n_s^{min}}$ .

Notons  $a \cdot b = \sum_{o \in \text{OUO}_c} a_o b_o$ ,  $(a \cdot b)_O = \sum_{o \in O} a_o b_o$  et  $(a \cdot b)_{O_c} = \sum_{o \in O_c} a_o b_o$ . On définit aussi  $\Lambda = (M \cdot n)_O / (M \cdot n)$  et  $\bar{\Lambda} = 1 - \Lambda$  les proportions de communications de notre opérateur et de notre concurrent. Sous contraintes  $\forall o \in O, p_o \geq r_o$ , que l'on tendra à oublier, le profit  $\Pi$  s'écrit :

$$\Pi = (p \cdot n)_O - c^S + \frac{\bar{\Lambda} \ddot{c} - \Lambda \dot{c}}{\bar{c}} c^S$$

### 2.1.3 Quelques notations

Afin d'appliquer des outils d'optimisation efficaces, nous allons calculer la différentielle du profit. Pour simplifier le calcul, nous allons introduire d'autres notations. Définissons donc les variables suivantes.

$$\Delta_{d,o} = \exp(u_{d,o} - \alpha_d p_o) \quad ; \quad \Delta_O(d) = \sum_{o \in O} \Delta_{d,o} \quad ; \quad \Delta_{total}(d) = \sum_{o \in \text{OUO}_c} \Delta_{d,o}$$

$$N_{total} = \sum_{o \in \text{OUO}_c} n_o = \sum_{d \in D} N_d \quad ; \quad B_o = \sum_{s \in \text{SUS}_c} b_{s,o} \quad ; \quad M_o = \sum_{s \in \text{SUS}_c} m_{s,o}$$

$$\gamma_{o,o'} = \frac{1}{B_{o'}} \sum_{s,s' \in \text{SUS}_c} \alpha_{s,s'} m_{s,o} b_{s',o'}$$

On remarque que l'on peut alors écrire  $c(\alpha, \beta) = \beta + \sum_{o,o' \in \text{OUO}_c} \gamma_{o,o'} n_o \frac{B_{o'} n_{o'}}{N^{cl}}$ .

## 2.2 Calcul différentiel

Dans l'algorithme d'optimisation que nous allons utiliser, nous allons chercher à éliminer le plus vite possible de grands intervalles. Pour cela, on va chercher à supprimer ceux où la différentielle est non nulle. Pour faire cela, il nous faut "bien" calculer les dérivées partielles, dans l'optique d'un calcul par intervalle efficace (voir 1.3).

### 2.2.1 Différentielles des nombres de consommateurs des offres

Pour différencier les nombres de consommateurs, nous allons calculer leurs développements limités à l'ordre 1. On suppose ici  $o \in O \cup O_c$ . Bien entendu, pour  $o' \in O_c$ ,  $dp_{o'} = 0$ .

$$n_o(p + dp) = \sum_{d \in D} \left( N_d \frac{\Delta_{d,o}}{\Delta_{total}(d)} \frac{1 - \alpha_d dp_o}{1 - \sum_{o' \in O} \frac{\alpha_d \Delta_{d,o'}}{\Delta_{total}(d)} dp_{o'}} \right) + o(dp)$$

$$n_o(p + dp) = \sum_{d \in D} \left( N_d \frac{\Delta_{d,o}}{\Delta_{total}(d)} (1 - \alpha_d dp_o) \left( 1 + \sum_{o' \in O} \frac{\alpha_d \Delta_{d,o'}}{\Delta_{total}(d)} dp_{o'} \right) \right) + o(dp)$$

$$n_o(p+dp) = n_o(p) + \sum_{d \in D} \frac{\alpha_d N_d \Delta_{d,o}}{\Delta_{total}(d)} \left( \frac{\Delta_{d,o}}{\Delta_{total}(d)} - 1 \right) dp_o + \sum_{o' \in O - \{o\}} \sum_{d \in D} \frac{\alpha_d N_d \Delta_{d,o} \Delta_{d,o'}}{\Delta_{total}(d)^2} dp_o + o(dp)$$

On en déduit les dérivées partielles des  $n_o$  :

$$\begin{aligned} \partial_{o'} n_o &= \sum_{d \in D} \frac{\alpha_d N_d \Delta_{d,o} \Delta_{d,o'}}{\Delta_{total}(d)^2} \text{ pour } o' \neq o. \\ \partial_o n_o &= \sum_{d \in D} \frac{\alpha_d N_d \Delta_{d,o}}{\Delta_{total}(d)} \left( \frac{\Delta_{d,o}}{\Delta_{total}(d)} - 1 \right) = \sum_{d \in D} \alpha_d N_d \left[ \left( \frac{\Delta_{d,o}}{\Delta_{total}(d)} - 1/2 \right)^2 - 1/4 \right] \end{aligned}$$

Cette écriture permet un calcul par intervalles plus précis. Pour améliorer la précision du calcul par intervalles, on remarquera aussi que l'on peut écrire de manière différente le rapport suivant :  $\frac{\Delta_{d,o'}}{\Delta_{total}(d)} = \frac{1}{1 + \frac{1}{\Delta_{d,o'}} (\Delta_{total}(d) \ominus \Delta_{d,o'})}$ .

## 2.2.2 Différentielles des coûts

Les expressions de  $\ddot{c}, \bar{c}, c^{\bar{s}}, c^S$  se ressemblent. On va donc poser  $c$  se définissant de la même manière avec des  $\beta$  et  $\alpha$ . On obtient

$$\begin{aligned} c(n+dn) &= \beta + \frac{\sum \alpha_{s,s'} n_s^{min} n_{s'}^{cl} (1 + \sum (\frac{m_{s,o}}{n_s^{min}} + \frac{b_{s',o}}{n_{s'}^{cl}}) dn_o)}{N^{cl} (1 + \sum \frac{B_o}{N^{cl}} dn_o)} + o(dn) \\ c(n+dn) &= c(n) + \sum_{o \in O} \frac{1}{N^{cl}} \left[ \sum_{s,s' \in S \cup S_c} \alpha_{s,s'} n_s^{min} n_{s'}^{cl} \left( \frac{m_{s,o}}{n_s^{min}} + \frac{b_{s',o}}{n_{s'}^{cl}} - \frac{B_o}{N^{cl}} \right) \right] dn_o + o(dn) \\ c(n+dn) &= c(n) + \sum_{o \in O} \left[ \sum_{s,s'} \frac{\alpha_{s,s'} m_{s,o} b_{s',o} n_{o'}}{N^{cl}} + \sum_{s,s'} \frac{\alpha_{s,s'} n_s^{min}}{(N^{cl})^2} (N^{cl} b_{s',o} - n_{s'}^{cl} B_o) \right] dn_o + o(dn). \end{aligned}$$

Or on peut remarquer que  $N^{cl} b_{s',o} - n_{s'}^{cl} B_o = \sum_{s'',o''} b_{s'',o''} n_{o''} b_{s',o} - \sum_{s'',o''} b_{s',o''} n_{o''} b_{s'',o} = 0$ .

Ainsi, si  $\gamma_{o,o'} = \frac{1}{B_{o'}} \sum_{s,s' \in S \cup S_c} \alpha_{s,s'} m_{s,o} b_{s',o'}$ , alors  $\frac{\partial c}{\partial n_o} = \frac{(\gamma_o \cdot (Bn))}{N^{cl}}$ .

Remarquons que  $\gamma$  est une matrice réelle et non une matrice intervalle. On n'aura donc pas à chercher de calcul efficace. De plus, on remarque que la dérivée partielle du coût  $c$  par rapport au nombre de clients ayant choisis l'offre  $o$  est bien positive. De plus, si le coût  $c$  n'est pas affecté par l'offre  $o$ , typiquement de la même manière que  $c^S$  n'est pas affecté par les offres  $o \in O_c$ , alors la dérivée du coût est bien nulle.

## 2.2.3 Différentielles du profit

Revenons en maintenant à l'expression du profit  $\Pi$  et différencions-le. Commençons en différenciant terme par terme. On rappelle que  $d(ab) = adb + bda$ , et  $d(a/b) = a/b(da/a - db/b)$ . On obtient ainsi les différentielles suivantes.

$$\begin{aligned} d\left(\sum_{o \in O} p_o n_o\right) &= \sum_{o \in O} (n_o + \sum_{o' \in O} p_{o'} \partial_o n_{o'}) dp_o + o(dp) \text{ et } dc = \sum_{o \in O} \left( \sum_{o' \in O} \frac{\partial c}{\partial n_{o'}} \partial_o n_{o'} \right) dp_o. \\ \text{Ainsi, } \partial_o(p \cdot n)_O &= n_o + (p \cdot \partial_o n)_O \text{ et } \partial_o c = \sum_{o' \in O \cup O_c} \frac{(\theta_{o'} \cdot (Bn)) - c B_{o'}}{N^{cl}} \partial_o n_{o'}. \end{aligned}$$

Pour simplifier l'écriture des calculs, on pose  $\Lambda = \frac{(M \cdot n)_O}{(M \cdot n)}$  et  $\bar{\Lambda} = 1 - \Lambda$ . Ainsi,

$$\begin{aligned} dR_{S_c \rightarrow S} &= \bar{\Lambda} \frac{c^S}{\bar{c}} d\ddot{c} + \bar{\Lambda} \frac{\ddot{c}}{\bar{c}} dc^S - \bar{\Lambda} \frac{c^S \ddot{c}}{\bar{c}^2} d\bar{c} + \frac{c^S \ddot{c} (M \cdot dn)_{O_c}}{\bar{c} (M \cdot n)} - \bar{\Lambda} \frac{c^S \ddot{c} (M \cdot dn)}{\bar{c} (M \cdot n)}. \\ dR_{S \rightarrow S_c} &= \Lambda \frac{c^S}{\bar{c}} d\dot{c} + \Lambda \frac{\dot{c}}{\bar{c}} dc^S - \Lambda \frac{c^S \dot{c}}{\bar{c}^2} d\bar{c} + \frac{c^S \dot{c} (M \cdot dn)_O}{\bar{c} (M \cdot n)} - \Lambda \frac{c^S \dot{c} (M \cdot dn)}{\bar{c} (M \cdot n)}. \end{aligned}$$

On a alors  $dR_{interco} = dR_{S_c \rightarrow S} - dR_{S \rightarrow S_c}$ . On peut alors écrire

$$dR_{interco} = \bar{\Lambda} \frac{c^S}{\bar{c}} d\ddot{c} - \Lambda \frac{c^S}{\bar{c}} d\dot{c} + \frac{\ddot{\bar{\Lambda}} - \dot{c}\Lambda}{\bar{c}} c^S \left( \frac{dc^S}{c^S} - \frac{d\bar{c}}{\bar{c}} - \frac{(M \cdot dn)}{(M \cdot n)} \right) + \frac{\ddot{c}(M \cdot dn)_{O_c} - \dot{c}(M \cdot dn)_O}{\bar{c}(M \cdot n)} c^S$$

Finalement, on peut écrire l'expression de la différentielle du profit en fonction de  $dp$ .

$$\begin{aligned} \partial_o \Pi &= n_o + (p \cdot \partial_o n)_O \sum_{o' \in O \cup O_c} + \left( \bar{\Lambda} \frac{c^S}{\bar{c}} \frac{\partial \ddot{c}}{\partial n_{o'}} - \Lambda \frac{c^S}{\bar{c}} \frac{\partial \dot{c}}{\partial n_{o'}} - \frac{\ddot{\bar{\Lambda}} - \dot{c}\Lambda}{\bar{c}^2} c^S \frac{\partial \bar{c}}{\partial n_{o'}} + \frac{\ddot{\bar{\Lambda}} - \dot{c}\Lambda}{\bar{c}} \frac{\partial c^S}{\partial n_{o'}} + \right. \\ &\quad \left. \frac{\ddot{c}M_{o'} 1_{o' \in O_c} - \dot{c}M_{o'} 1_{o' \in O}}{\bar{c}(M \cdot n)} c^S - \frac{\ddot{\bar{\Lambda}} - \dot{c}\Lambda}{\bar{c}} \frac{M_{o'}}{(M \cdot n)} c^S - \frac{\partial c^S}{\partial n_{o'}} \right) \partial_o n_{o'} \end{aligned}$$

$$\partial_o \Pi = n_o + (p \cdot \partial_o n)_O + \sum_{o' \in O \cup O_c} \left[ \frac{(\Gamma_{o'} \cdot (Bn))}{N^{cl}} + \frac{\dot{c} + \ddot{c}}{\bar{c}} \frac{M_{o'}}{(M \cdot n)} (\Lambda - 1_{o' \in O}) \right] \partial_o n_{o'}$$

où on a défini la matrice d'intervalles  $\Gamma = \frac{c^S}{\bar{c}} \bar{\Lambda} (\ddot{\gamma} - \frac{\ddot{c}}{\bar{c}} \dot{\gamma}) - \frac{c^S}{\bar{c}} \Lambda (\dot{\gamma} - \frac{\dot{c}}{\bar{c}} \ddot{\gamma}) + \frac{\ddot{\bar{\Lambda}} - \dot{c}\Lambda}{\bar{c}} \gamma^S - \gamma^S$ , ce qui s'écrit aussi  $\Gamma = \frac{\bar{\Lambda} \ddot{c} - \Lambda \dot{c}}{\bar{c}} (\gamma^S - \frac{c^S}{\bar{c}} \ddot{\gamma}) + \frac{\bar{\Lambda} \ddot{\gamma} - \Lambda \dot{\gamma}}{\bar{c}} c^S - \gamma^S$ .

De plus, si on pose  $\kappa_o = \frac{\partial}{\partial n_o} (R_{interco} - c^S) = \frac{(\Gamma_o \cdot (Bn))}{N^{cl}} + \frac{\dot{c} + \ddot{c}}{\bar{c}} \frac{M_o}{(M \cdot n)} c^S (\Lambda - 1_{o \in O})$ , alors l'expression de la dérivée partielle du profit par rapport au prix  $p_o$  d'une offre  $o$  s'écrit alors

$$\partial_o \Pi = n_o + (p \cdot \partial_o n)_O + (\kappa \cdot \partial_o n)$$

#### 2.2.4 Remarques et commentaires

On peut noter que  $\kappa_o = \frac{\partial R_{interco}}{\partial n_o} - \frac{\partial c^S}{\partial n_o}$ . S'il est clair que  $\frac{\partial c^S}{\partial n_o} \geq 0$ , puisque l'on a considéré des fonctions de coûts croissantes du flux de communication dans les services de  $S$ , lequel est croissant du nombre de clients choisissant l'offre  $o$  de l'opérateur  $S$ .

En revanche, on peut supposer  $\frac{\partial R_{interco}}{\partial n_o}$  négatif si  $o \in O$  et positif sinon. En effet, quand le nombre de clients choisissant une offre  $o$  de l'opérateur  $S$  augmente, la proportion de l'ensemble des services de  $S$  utilisée par des clients de concurrents de  $S$  diminue, et les transferts d'interconnexion sont alors en défaveur de  $S$ .

## 2.3 Calcul par intervalles

Dans cette partie, nous allons essayer de déterminer des expressions aussi efficaces que possibles. Rappelons que les seules véritables variables qui permettent de maximiser le profit sont les prix  $p_o$ .

### 2.3.1 Expressions efficaces des nombres de consommateurs

Commençons donc par le plus facile. Les nombres de clients par offre et les nombres de clients par service peuvent être calculés assez facilement de manière efficace.

$$n_o = \sum_{d \in D} \frac{N_d}{1 + (\Delta_{total}(d) \ominus \Delta_{d,o})/\Delta_{d,o}}$$

$$n_s^{cl} = \sum_{d \in D} \frac{N_d}{1 + [\Delta_{total}(d) \ominus (\sum b_{s,o} \Delta_{d,o})]/(\sum b_{s,o} \Delta_{d,o})}$$

Remarquons que ces deux écritures sont efficaces. En effet, comme les fonctions  $\Delta_{d,o}$  sont des fonctions décroissantes de  $p_o$ . Pour chaque terme  $\frac{N_d}{1 + (\Delta_{total}(d) \ominus \Delta_{d,o})/\Delta_{d,o}}$ , le minimum est atteint pour  $p_o$  minimum et pour tous les autres  $p_{o'}$  maximum, et l'inverse pour le maximum. Il en est donc de même pour les bornes de  $n_o$ . Ainsi l'expression est efficace. Il en est de même pour  $n_s^{cl}$ .

### 2.3.2 Calcul par intervalles de différentes variables

Venons en maintenant au calcul de  $n_s^{min}$  et  $N^{cl}$ . Celui-ci est un peu plus difficile, et l'on va devoir trouver d'autres solutions pour les trouver efficacement.

$$n_s^{min} = \sum_{d \in D} N_d \frac{\sum m_{s,o} \Delta_{d,o}}{\sum \Delta_{d,o}} = \sum_{d \in D} N_d \mathcal{M}_O(m_s, \Delta_d)$$

$$N^{cl} = \sum_{d \in D} N_d \frac{\sum B_o \Delta_{d,o}}{\sum \Delta_{d,o}} = \sum_{d \in D} N_d \mathcal{M}_O(B, \Delta_d)$$

Le calcul efficace des  $n_s^{min}$  et  $N^{cl}$  est plus difficile. On remarquera que pour le calcul de  $\mathcal{M}_O(m_s, \Delta_d)$  comme pour celui de  $\mathcal{M}_O(B, \Delta_d)$ , quelque soit le  $d$  correspondant, correspond à prendre les  $p_o$  correspondant aux grands  $m_{s,o}$  le plus petit possible, et vice-et-versa.

Passons maintenant au calcul des coûts et des rapports de coûts. On rappelle l'expression d'un coût  $c$  avec pour paramètres  $\alpha$  et  $\beta$ .

$$c = \beta + \frac{1}{N^{cl}} \sum_{s,s' \in S \cup S_c} \alpha_{s,s'} n_s^{min} n_{s'}^{cl} = \beta + \frac{1}{\sum B_o n_o} \sum_{s,s',o,o'} \alpha_{s,s'} m_{s,o} n_o b_{s',o'} n_{o'}$$

$$c = \beta + \sum_o n_o \frac{\sum_{o'} [(\sum_{s,s'} \alpha_{s,s'} m_{s,o} b_{s',o'})/B_{o'}] B_{o'} n_{o'}}{\sum_{o'} B_{o'} n_{o'}} = \beta + \frac{N_{total}}{\sum_o n_o} \sum_o n_o \mathcal{M}_{o'}((\gamma_{o,o'}), (Bn)_{o'})$$

$$c = \beta + N_{total} \mathcal{M}_o(\mathcal{M}_{o'}(\gamma_{o,o'}, (Bn)_{o'} | \sum (Bn)_{o'} \in N_{total} [\min B, \max B]), n_o | \sum n_o = N_{total})$$

où  $\gamma_{o,o'} = (\sum_{s,s'} \alpha_{s,s'} m_{s,o} b_{s',o'})/B_{o'}$ , et où  $(Bn)_{o'} = B_{o'} n_{o'}$ .

### 2.3.3 Calcul des termes du profit

Essayons maintenant de calculer les recettes dues à la vente de détail.

$$p \cdot n = \sum_{o \in O} p_o n_o = \sum_{o,d} N_d(p_o \Delta_{d,o} / \sum_{o' \in O \cup O_c} \Delta_{d,o'}) = \sum_{o,d} N_d(p_o \Delta_{d,o} / (\Delta_{total}(d) \ominus \Delta_{d,o} + \Delta_{d,o}))$$

On voit apparaître des termes de la forme  $\varphi_1(x, y) = \mu x e^{-\lambda x} / (y + \mu e^{-\lambda x})$  où  $x$  représente des variables  $p_o$  et  $y$  est une fonction décroissante des autres  $p_{o'}$ . Pour maximiser (resp. minimiser) une telle expression, il faut prendre les  $p_{o'}$  maximum (resp. minimum), ce qui fixe  $y$ , et on se retrouve face à une fonction  $\varphi_1(x)$  non monotone à une variable.

**Lemme 6** *Le minimum de  $\varphi_1$  sur  $X$  est atteint sur l'un des 2 bords. Le maximum est atteint en un point approchable par dichotomie (donc en un temps logarithmique de la précision de calcul).*

#### Démonstration du lemme 5

En calculant la dérivée, on se rend compte que la croissance de  $h$  au voisinage de  $x$  équivaut à la condition  $\mu e^{-\lambda x} \geq y(\lambda x - 1)$ . Remarquons que  $\mu e^{-\lambda x}$  est croissant et  $y(\lambda x - 1)$  décroît strictement. Au vu des valeurs aux bornes, on en déduit  $\exists! x^* \in \mathbb{R}_{++}, \mu e^{-\lambda x^*} \geq y(\lambda x^* - 1)$ . On en déduit alors le lemme, le maximum étant atteint en le point de  $X$  le plus proche de  $x^*$ .

Si cette méthode calcule quasiment efficacement chacun des termes  $(p_o \Delta_{d,o} / \sum \Delta_{d,o'})$ , elle ne correspond pas un calcul efficace de  $p \cdot n$ , notamment de son minimum, puisque pour minimiser un de ces termes pour  $p_o$ , on prend des  $p_{o'}$  minimum, mais pour de tels  $p_{o'}$ , les termes  $(p_{o'} \Delta_{d,o'} / \Delta_{d,o''})$  ne sont a priori pas minimiser. Cette dépendance induit ici une inefficacité.

### 2.3.4 Calcul de $p_{o'} \partial_o n_{o'}$

Pour calculer la différentielle, il nous faut trouver une expression assez efficace de  $p \cdot \partial_o n$ , c'est-à-dire des  $p_{o'} \partial_o n_{o'}$  et  $p_o \partial_o n_o$ . On rappelle que l'on a  $\partial_o n_{o'} = \sum_{d \in D} \frac{\alpha_d N_d \Delta_{d,o} \Delta_{d,o'}}{\Delta_{total}(d)^2}$  et que

$$\partial_o n_o = \sum_{d \in D} \frac{\alpha_d N_d \Delta_{d,o}}{\Delta_{total}(d)} \left( \frac{\Delta_{d,o}}{\Delta_{total}(d)} - 1 \right). \text{ On commencera par } p_{o'} \partial_o n_{o'}.$$

$p_o$  et  $\Delta_{d,o'}$  apparaissent sous la forme d'une somme de  $\varphi_2(x, y, z) = \frac{x e^{-\lambda x} y}{(e^{-\lambda x} + y + z)^2}$  pour différent intervalles de définition  $X \times Y \times Z$  correspondant à  $\Delta_{d,o}$  et  $\Delta_{d,o'}$  et  $\sum \Delta_{d,o''}$  pour  $o'' \notin \{o, o'\}$ . Comme tous les termes sont positifs,  $\varphi_2$  est clairement une fonction décroissante de  $z$ , qu'il faut prendre minimum (resp. maximum) si on veut maximiser (resp. minimiser)  $\varphi_2$ . On supposera donc maintenant  $z$  fixé.

**Lemme 7** *Soit  $\varphi \in C^1(X \times Y, \mathbb{R})$ . On suppose que, à  $x$  fixé,  $\varphi$  est une fonction concave de  $y$  et vice-versa. Alors le minimum est atteint sur l'un des 4 coins du carré  $X \times Y$ . De plus, le maximum se trouve, soit au point de différentielle nulle ou se trouve en au plus 4 valeurs approchables par dichotomie (donc en temps logarithmique).*

#### Démonstration du lemme 6

Si le minimum était atteint en un point pour lequel l'une des coordonnées est non saturée, la concavité montre que l'on pourrait faire au moins aussi bien en saturant cette coordonnée d'un côté ou de l'autre. Cela prouve la première partie du lemme.

Supposons maintenant que le maximum ne soit pas atteint en un point de différentielle nulle. C'est que l'une des contraintes est active. Si  $x$  est saturé (2 cas), la fonction  $\varphi$  ne dépendant maintenant plus que de  $y$  est concave et atteint son maximum en le point le plus proche du point annulant  $\partial_2\varphi$ , lequel est calculable par dichotomie car  $\partial_2\varphi$  est décroissante. Sinon, c'est que  $y$  est saturé, et on a 2 autres cas, eux aussi abordables par dichotomie. CQFD.

Or on a  $\partial_2\varphi_2 \geq 0 \Leftrightarrow y \leq z + e^{-\lambda x}$  et  $\partial_1\varphi_2 \geq 0 \Leftrightarrow \psi_2(x, y) = (1 + \lambda x)e^{-\lambda x} + (1 - \lambda x)(y + z) \geq 0$ . Ainsi,  $\varphi_2$  vérifie les hypothèses du lemme 6. De plus, en un point de différentielle nulle, on a  $y = z + e^{-\lambda x}$  et  $\psi_2(x, y) = 0$ , d'où  $\psi_2(x, z + e^{-\lambda x}) = (1 + \lambda x)e^{-\lambda x} + (1 - \lambda x)(2z + e^{-\lambda x}) = 2e^{-\lambda x} + 2z(1 - \lambda x)$ , qui est une fonction décroissante de  $x$ , que l'on peut donc résoudre par dichotomie. Le maximum et le minimum de  $\varphi_2$  sont donc calculables en temps logarithmique.

Venons en à  $p_o\partial_o n_o$ , dont l'expression fait surgir  $\varphi_3(x, z) = \frac{xe^{-x}}{z + e^{-x}} \left( \frac{e^{-x}}{z + e^{-x}} - 1 \right)$ , égal à  $\varphi_3(x, z) = -z \frac{xe^{-x}}{(z + e^{-x})^2}$ . Remarquons que  $\varphi_3$  est une fonction de  $z$  d'une forme quasi-similaire à la manière dont  $\varphi_2$  est une fonction de  $y$ . On en déduit qu'elle atteint son minimum en  $z^*(x)$  le point de  $Z$  le plus proche de  $e^{-x}$  à  $x$  fixé. Son maximum est atteint pour  $z$  sur les bords. Alors  $\partial_1\varphi_3 \geq 0 \Leftrightarrow -2xe^{-x}(z + e^{-x})e^{-x} - (1 - x)e^{-x}(z + e^{-x})^2 \geq 0 \Leftrightarrow -2xe^{-x} - (1 - x)(z + e^{-x}) \geq 0 \Leftrightarrow \psi_3(x, z) = -(x + 1)e^{-x} + (x - 1)z \geq 0$  qui est une fonction croissante de  $x$  pour  $z > 0$  (il suffit de dériver pour le vérifier). Une recherche dichotomique permet de trouver le point de  $x^*(z)$  le point de  $X$  le plus proche du point  $x$  annulant la dérivée de  $\varphi_3$  par rapport à  $x$  à  $z$  fixé.

Pour un point de différentielle nulle, on a  $z = e^{-x}$  et  $\varphi_3(x, e^{-x}) = -x/4$ , dont la dérivée par rapport à  $x$  ne s'annule jamais. Donc si  $X \times Z$  est strictement inclus dans  $\mathbb{IIR}_+^2$ , il n'y a pas sur cet intervalle de point de différentielle nulle.

Récapitulons. Le minimum est atteint pour  $x$  sur un bord de  $X$  et  $z = z^*(x)$ . Le maximum est atteint pour  $z$  sur un bord de  $Z$  et  $x = x^*(z)$ . Cela nous fait 4 cas à tester.

Remarquons que l'image de la somme des  $\varphi_2$  ou des  $\varphi_3$  fait intervenir des termes dépendants. Le calcul du revenu de détail n'est donc pas efficace, mais on s'en contentera.

### 2.3.5 Calcul des termes de la différentielle

Afin d'appliquer les algorithmes d'optimisation décrits dans le deuxième chapitre, il nous faut calculer tous les termes de la différentielles. Commençons par calculer les dérivées partielles des nombres de clients par offre.

Pour  $o' \neq o$ , on a  $\partial_o n_{o'} = \sum_{d \in D} \frac{\alpha_d N_d \Delta_{d,o} \Delta_{d,o'}}{\Delta_{total}(d)^2}$ , qui est une fonction décroissante des  $\Delta_{d,o''}$  pour  $o'' \notin \{o, o'\}$ , et donc croissante des  $p_{o''}$ . Remarquons par ailleurs que  $\Delta_{d,o}$  et  $\Delta_{d,o'}$  apparaissent sous la forme  $\varphi_4(x_1, x_2, y) = x_1 x_2 / (y + x_1 + x_2)^2$  à  $d$  fixé, qui est positif, nul en 0, et croissant par rapport à  $x_1$  (resp.  $x_2$ ) lorsque  $x_1 \leq y + x_2$  (resp.  $x_1 \leq y + x_2$ ). Ainsi, si par exemple  $x_1^{max} \leq x_2^{max}$ , le maximum sera atteint quand  $x_1$  prend sa valeur maximale, et quand  $x_2 = \min(x_2^{max}, y + x_1^{max})$ . Quant au minimum, il sera atteint sur l'un des coins du carré de définition de  $(x_1, x_2)$ .

$$\text{On a } \partial_o n_o = \sum_{d \in D} \frac{\alpha_d N_d \Delta_{d,o}}{\Delta_{total}(d)} \left( \frac{\Delta_{d,o}}{\Delta_{total}(d)} - 1 \right) = - \sum_{d \in D} \alpha_d N_d \left[ \frac{1}{4} - \left( \frac{1}{2} - \frac{1}{1 + (\Delta_{total} \ominus \Delta_{d,o}) / \Delta_{d,o}} \right)^2 \right]$$

qui est une écriture efficace de  $\partial_o n_o$ , car c'est une expression monotone par rapport aux  $p_o$ , à condition que pour tout  $d, \Delta_{d,o}$  ne peut être supérieur à  $\Delta_{total}(d) \ominus \Delta_{d,o}$ , ceci impliquant alors l'inégalité  $\left( \frac{1}{1 + (\Delta_{total} \ominus \Delta_{d,o}) / \Delta_{d,o}} \right)^2 \leq \frac{1}{2}$ . Sinon, il s'agit tout de même d'une écriture assez efficace dans la mesure où, à  $d$  fixé, l'expression est efficace. Cependant, si  $\nu = (\Delta_{total} \ominus \Delta_{d,o}) / \Delta_{d,o}$  est très petit ou très grand, les approximations de calcul dues aux troncatures des double, associées à nos largesse, conduit à de grosses erreurs. On remarque alors que  $\frac{1}{4} - \left( \frac{1}{2} - \frac{1}{1 + \nu} \right)^2 = \frac{\nu}{(1 + \nu)^2} = \frac{1/\nu}{(1 + 1/\nu)^2}$ . La première convient mieux pour des valeurs de  $\nu$  moyennes, pas trop éloignées de 1, la seconde pour des petites valeurs de  $\nu$  et la dernière pour de grandes valeurs. On calculera cette expression comme l'intersection des trois écritures.

On rappelle que  $\Lambda = \frac{(M \cdot n)_O}{(M \cdot n)} = \frac{1}{1 + (M \cdot n)_{O_c} / (M \cdot n)_O}$ , sous la contrainte  $\sum n = N_{total}$ . On peut calculer cette valeur de manière efficace par rapport aux  $n_o$  grâce au lemme suivant.

**Lemme 8** *Supposons  $M_1 \geq M_2 \geq \dots \geq M_O$  et  $M_{O+1} \leq M_{O+2} \leq \dots \leq M_{O+O_c}$ . Alors le maximum de  $(M \cdot n)_{O_c} / (M \cdot n)_O$  est atteint en  $n^{max}(i_{max}, t)$  tel que  $t = N_{total} - \sum_{j \neq i} n_j^{max}(i_{max})$ .*

Dans l'expression des dérivées partielles, on remarque qu'apparaissent des différences de la forme  $A\Lambda - B(1 - \Lambda)$ . Si  $A$  et  $B$  sont des intervalles positifs, alors cette expression est monotone en chacun des termes et est donc une expression efficace.

On peut alors trouver la matrice d'intervalles  $\Gamma = \frac{\bar{\Lambda}\bar{c} - \Lambda\dot{c}}{\bar{c}} (\gamma^S - \frac{c^S}{\bar{c}} \bar{\gamma}) + \frac{\bar{\Lambda}\ddot{\gamma} - \Lambda\dot{\gamma}}{\bar{c}} c^S - \gamma^S$ . On a alors l'expression du vecteur  $\kappa_o = \mathcal{M}(\Gamma_o, Bn) + \frac{\dot{c} + \bar{c}}{\bar{c}} \frac{M_o}{N_{total} \mathcal{M}(M, n)} c^S (\Lambda - 1_{o \in O})$ , où les moyennes  $\mathcal{M}(\Gamma_o, nB)$  et  $\mathcal{M}(M, n)$  sont calculé sous contrainte  $\sum n_o B_o \in N_{total} [\min B, \max B]$  et  $\sum n_o = N_{total}$ .

## 2.4 Les fonctions de réduction

Nous allons maintenant voir différentes manières d'utiliser la *Hull Consistency* pour notre modèle. Nous rappelons que  $\partial_o \Pi = n_o + (p \cdot \partial_o n)_O + (\kappa \cdot \partial_o n)$ . Il faut maintenant réécrire cette expression sous une forme  $\partial_o \Pi = 0 \Rightarrow g(p_o', p_o'') \in a(P)$ . Dans cette partie, on posera  $L_o = (p \cdot \partial_o n)_O$  et  $L'_o = (p \cdot \partial_o n)_{O - \{o\}}$ , ainsi que  $K_o = (\kappa \cdot \partial_o n)$  et  $K'_o = (\kappa \cdot \partial_o n)_{O \cup O_c - \{o\}}$ .

### 2.4.1 Problème du revenu de détail

En ce qui concerne le problème du revenu de détail, où on néglige le revenu d'interconnexion, il s'agit de maximiser  $\Pi = (p \cdot n)_O - c^S$ , sous contrainte  $p_o \geq r_o$ . La dérivée partielle du profit par rapport au prix  $p_o$  de l'offre  $o$  s'écrit  $\partial_o \Pi = n_o + (p \cdot \partial_o n)_O + (\kappa^{PRD} \cdot \partial_o n)$ , où on a posé  $\kappa_o^{PRD} = -\frac{\partial c^S}{\partial n_o} = -\mathcal{M}(\gamma_o^S, nB)$ , dont on peut remarquer qu'il est nul lorsque  $o \notin O$ , et strictement négatif sinon. On a alors  $\partial_o \Pi = n_o + (p \cdot \partial_o n)_O + (\kappa^{PRD} \cdot \partial_o n)_O$ .

### 2.4.2 Inversion de $p$

L'expression de la dérivée partielle du profit fait apparaître des termes  $p_o$  explicitement. Ainsi,

$$\partial_o \Pi = 0 \iff (p \cdot \partial_o n)_O = -n_o - (\kappa \cdot \partial_o n) \iff p_{o'} = -\left(n_o + K_o + (p \cdot \partial_o n)'_O\right) / \partial_o n_{o'}$$

Cette équation est valable pour tout couple  $(o, o')$ , ce qui nous fait  $|O|^2$  équations.

### 2.4.3 Inversion de $n_o$

De façon plus immédiate, en posant  $g(p_o, p_{-o}) = n_o$ , on obtient  $|O|$  équations en remarquant

$$\partial_o \Pi = 0 \iff n_o = -(p \cdot \partial_o n)_O - K_o$$

où on écrit toujours  $K_o = \partial_o(R_{interco} - c^S) = (\kappa \cdot \partial_o n)$ . On rajoute que  $n_o$  est une expression strictement monotone de  $p_o$  et  $p_{-o}$ , ce qui permet de retrouver  $p_o$  rapidement par dichotomie si on a réussi à réduire  $n_o$ .

### 2.4.4 Inversion de $\partial_o n_o$

On rappelle que  $L'_o = (p \cdot \partial_o n)_{O-\{o\}}$  et  $K'_o = K_o \ominus (\kappa_o \partial_o n_o) = (\kappa \cdot \partial_o n)'$ . L'intérêt de séparer ces termes est  $\partial_o n_{o'} > 0 \iff o' \neq o$ . Remarquons que l'on a maintenant

$$\partial_o \Pi = 0 \iff -\partial_o n_o = \frac{n_o + (p \cdot \partial_o n)'_O + K_o}{p_o} \iff -\partial_o n_o = \frac{n_o + (p \cdot \partial_o n)_O + K'_o}{\kappa_o}$$

On a ici une implication de la forme  $p \in P$  et  $df(p) = 0 \Rightarrow g(p_o, P_{-o}) \cap a(P) \neq \emptyset$ , où on a  $a(P) = \frac{n_o + (p \cdot \partial_o n)'_O + K_o}{p_o} \cap \frac{n_o + (p \cdot \partial_o n)_O + K'_o}{\kappa_o}$  et  $g(p_o, P_{-o}) = -\partial_o n_o$ . Rappelons que  $-\partial_o n_o = \sum \alpha_d N_d \left[ \frac{1}{4} - \left( \frac{1}{2} - \frac{1}{1 + \Delta_{d,-o}/\Delta_{d,o}} \right)^2 \right]$ , et que cette expression correspondant au  $g$  que l'on recherche n'est ni monotone ni même concave si  $|D| \geq 2$ . Cependant, on peut s'en plus ou moins bien s'en sortir comme le montre la partie suivante.

### 2.4.5 Réduction par $\partial_o n_o$

Si  $\min a(P) \cap g(P)$  est vide, c'est qu'il n'y a pas de maximum dans l'intervalle recherché. Supposons maintenant que  $\min a(P) \cap g(P) > \min g(P)$ , c'est-à-dire qu'on peut réduire l'intervalle de recherche de  $g$ . Remarquons que  $\forall P'_{-o} > 0$ , la fonction  $p_o \mapsto \min g(p_o, P'_{-o})$  est croissante (resp. décroissante) lorsque nous avons les conditions équivalences suivantes :

1.  $\forall d, \Delta_{d,o}(p_o) \geq \max \Delta_{d,-o}(P_{-o})$  (resp.  $\forall d, \Delta_{d,o}(p_o) \leq \min \Delta_{d,-o}(P_{-o})$ )
2.  $\min_{d \in D} \Delta_{d,o}(p_o) \geq \max_{d \in D} \max \Delta_{d,-o}(P_{-o})$  (resp.  $\max_{d \in D} \Delta_{d,o}(p_o) \geq \min_{d \in D} \min \Delta_{d,-o}(P_{-o})$ )
3.  $p_o \leq p_o^-(P_{-o})$  (resp.  $p_o \geq p_o^+(P_{-o})$ )

Où on aura posé les expressions  $p_o^-(P_{-o}) = \left( \min_{d \in D} \Delta_{d,o} \right)^{-1} \left( \max_{d \in D} \max \Delta_{d,-o}(P_{-o}) \right)$ , ainsi que  $p_o^+(P_{-o}) = \left( \max_{d \in D} \Delta_{d,o} \right)^{-1} \left( \min_{d \in D} \min \Delta_{d,-o}(P_{-o}) \right)$ , que l'on peut approcher par dichotomie, quand on remarque la monotonie de  $\min_{d \in D} \Delta_{d,o}$  et  $\max_{d \in D} \Delta_{d,o}$ .



Ainsi, si  $\min P_o < p_o^-(P_{-o})$  et  $g(\min P_o, \max P_{-o}) \leq \min a(P) \cap g(P) \leq g(p_o^-(P_{-o}), \max P_{-o})$ , alors il est alors possible de calculer par dichotomie l'élément  $p_o^1 \in [\min P_o, p_o^-(P_{-o})]$  tel que l'on ait  $p_o^1 = g_{\max P_{-o}}^{-1}(\min a(P) \cap g(P))$ . On calcule ensuite  $p_o^2$  de façon similaire pour  $p_o$  grand et  $p_{-o} = \min P_{-o}$ . Alors l'application  $P \mapsto [p_o^1, p_o^2]$  est une fonction de réduction sur  $p_o$ .

Cette méthode dérivée de la *Hull Consistency* fournit au plus  $|O|$  équations de réduction implicites.



# Chapitre 3

## Résultats et commentaires

Nous avons implémenté notre algorithme d'optimisation par analyse des intervalles, ainsi que notre problème d'optimisation du revenu global des services voix d'un opérateur par le système de tarification. Dans cette dernière partie, nous allons présenter les résultats de notre étude.

### 3.1 Tests de notre algorithme

Avant d'appliquer notre algorithme d'optimisation, nous l'avons testé sur quelques exemples. Nous allons également voir à travers ces exemples la performance de notre algorithme et sa capacité à fournir une grande précision sur les résultats. Bien entendu, dans ces exemples simples, une utilisation de la méthode de Newton aurait pu garantir une complexité quadratique. Cependant, nous en resterons à la *Hull Consistency* uniquement.

#### 3.1.1 Les exemples de Khalid Mehl

Dans son rapport [3], Khalid Mehl donne deux exemples de problème d'optimisation correspondant à des cas simplistes du problème générale. En effet, on considère des systèmes à un seul concurrent possédant une offre, et on propose successivement une et deux offres, pour un seul service. Rappelons que Khalid Mehl a utilisé une méthode d'approximation des exponentielles par des polynômes, afin d'utiliser la méthode des moments. Les résultats de cette approximation sur ses exemples étaient peu concluants.

Voici le premier problème.

**Problème 1**  $\max_{x \in \mathbb{R}} f_1(x) = \frac{xe^{-2x}}{(e^{-2x} + e^{-2})}$  sans contrainte.

Remarquons que la dérivée vérifie  $f_1'(x) = 0 \Leftrightarrow (1 - 2x)e^{-2x}(e^{-2x} + e^{-2}) + 2xe^{-2x}e^{-2x} = 0 \Leftrightarrow e^{-2x} + (1 - 2x)e^{-2} = 0 \Leftrightarrow x = h_1(x) = (1 + e^{-2(x-1)})/2$ . Nous avons appliqué notre optimisation à  $f_1$ , avec pour  $h_1$  pour fonction de réduction, et avec une erreur admissible  $\epsilon$  de  $10^{-8}$ . On obtient alors quasi-instantanément une localisation du point réalisant le maximum dans l'intervalle  $X^* = [0.9999999999999989, 1.0000000000000001]$ . La valeur de ce maximum est alors dans  $f(X^*) = [0.4999999999999989, 0.5000000000000001]$ . Or le vrai maximum est atteint en 1 et vaut 0.5. On est pas mal!

**Problème 2**  $\max_{x,y \in \mathbb{R}_+} f_2(x,y) = \frac{xe^{-x} + ye^{-y}}{e^{-x} + e^{-y} + 1}$  sans contrainte.

On a  $\partial_1 f_2 = 0 \Leftrightarrow (1-x)e^{-x}(e^{-x} + e^{-y} + 1) + (xe^{-x} + ye^{-y})e^{-x} = 0$ , ce qui équivaut à

1.  $x = h_1^x(x,y) = 1 + e^{-x} + (1+y-x)e^{-y}$
2.  $y = h_2^y(x,y) = e^{-y}(x - xe^{-x} + 1) + x - 1$
3.  $x = h_3^x(x,y) = e^{-y}(e^{-x} - x - 1) + y + 1$ .

En remarquant la symétrie entre  $x$  et  $y$ , on voit qu'on peut définir trois fonctions de réduction pour chacune des coordonnées. Là encore, on a lancé notre algorithme d'optimisation avec une précision à  $10^{-8}$ , et on a obtenu  $X^* = [1.4630555133590368, 1.4630555133822338]$  et  $Y^* = [1.4630555133620022, 1.463055513367837]$  au bout d'une fraction de seconde. Et la valeur du maximum est localisée  $f(X^*, Y^*) = [0.4630555133634215, 0.4630555133676762]$ . On peut remarquer que  $X^* \neq Y^*$ , cela étant dû à la méthode de séparation, qui repose sur le choix aléatoire de la coordonnée à séparer.

### 3.1.2 Un exemple en dimension $n$

Les cas précédents étant de faible dimension avec des expressions simples (donc assez efficaces), il est peu surprenant de constater une grande performance de notre algorithme. Traitons maintenant à travers le problème 3, qui généralise le précédent à des dimensions supérieures.

**Problème 3**  $\max_{x \in \mathbb{R}_+^n} f(x) = \frac{\sum x_i e^{-x_i}}{\sum e^{-x_i} + 1}$  sans contrainte.

On a  $\partial_i f = 0 \Leftrightarrow x_i = h_1^i(x) = 1 + \frac{\sum x_j e^{-x_j}}{1 + \sum e^{-x_j}} \Leftrightarrow x_i = h_2^i(x) = 1 + \sum x_j e^{-x_j} + (1-x_i) \sum e^{-x_j}$ .

Remarquons qu'aucune de ces expressions n'est efficace. Nous les avons cependant implémentées ainsi, afin de tester notre algorithme. Celui-ci a alors fonctionné pour une précision de  $10^{-8}$  et des dimensions inférieures ou égal à 12 en un temps raisonnable (de l'ordre de quelques minutes). Au delà, le nombre d'éléments de la liste a tendance à dépasser 10 000. Voici ce qu'affiche notre méthode de résolution pour une optimisation sur  $\mathbb{R}_+ \cup \{+\infty\}$  :

Nous avons trouvé 73 solutions difficiles à départager.

Le maximum est atteint sur leur réunion  $X^*$ , défini par

X1\* = [2.256542638235386, 2.2565426383569207] ;  
 X2\* = [2.256542638233173, 2.2565426383569207] ;  
 X3\* = [2.256542638109426, 2.2565426383569207] ;  
 X4\* = [2.256542638109426, 2.2565426383047362] ;  
 X5\* = [2.256542638109426, 2.2565426382331726] ;  
 X6\* = [2.256542638109426, 2.256542638255615] ;  
 X7\* = [2.256542638109426, 2.256542638327705] ;  
 X8\* = [2.256542638109426, 2.2565426383569207] ;  
 X9\* = [2.256542638234515, 2.2565426383569207] ;  
 X10\* = [2.2565426382763047, 2.2565426383569207] ;  
 X11\* = [2.256542638109426, 2.2565426383569207] ;  
 X12\* = [2.256542638109426, 2.2565426383569207] ;

Et la valeur du maximum est dans l'intervalle  $f(X^*) = [1.2565426381094544, 1.2565426383568923]$

Cela correspond-il à la vraie solution? L'analyse des valeurs aux bornes de  $f$  permet de ce réduire à un compact. Le maximum est donc atteint, en un point de dérivée nulle. Remarquons que les conditions  $\forall x_i = h_1^i(x) = 1 + f(x)$  impliquent que, au maximum, tous les  $x_i$  sont égaux à un réel  $x^*$ , et  $x^*$  vérifie  $x^* = 1 + f(x)$ . Cela correspond à ce que l'on a trouvé.

Notons que l'on en déduit que le maximum s'écrit  $nx^*e^{-x^*}/(ne^{-x^*} + 1)$ , dont la dérivée s'annule si et seulement si  $e^{-x^*} = (x^* - 1)/n$ . La solution de cette équation, dont on peut prouver l'existence et l'unicité par la monotonie et les valeurs aux bornes, est la solution du problème. Graphiquement, on voit que la fonction  $x^* \mapsto (x^* - 1)/n$  s'horizontalise lorsque  $n$  grandit, ce qui augmente la valeur de la solution  $x^*$ . C'est bien ce qu'on a constaté sur les résultats obtenus en augmentant la dimension.

## 3.2 Estimation de l'efficacité

Dans cette partie, nous allons appliquer notre algorithme à notre problème d'optimisation. Nous allons ainsi générer des valeurs pour les données du problème, en expliquant comment on les a choisies pour qu'elles soient raisonnables.

Ensuite, nous allons estimer l'efficacité de nos calculs par intervalle, dans le but de voir si notre algorithme a des chances de réussir. En particulier, la question que l'on se pose porte sur les capacités de notre *Hull Consistency* d'effectivement réduire les intervalles, et de localiser les trop grosses approximations qui empêcheraient un bon fonctionnement de notre algorithme.

### 3.2.1 Choix et estimation des données du problème

On supposera maintenant  $O$ ,  $O_c$ ,  $S$ ,  $S_c$  et  $D$  fixés. On définit aussi  $q$  la probabilité que  $b_{s,o}$  soit égal à 1 lorsque  $(s, o) \in S \times O \cup S_c \times O_c$ , et on la fixe elle aussi.

Pour tester notre problème, nous avons tiré des valeurs aléatoires pour nos données qui nous semblaient raisonnable. Ainsi les  $N_d$  seront choisis uniformément dans  $[1; 9] = 1 + 8[0; 1]$ . On peut les interpréter en millions d'habitants ou en tant que poids de la population  $d$ . Le profit calculé est alors le profit par millions d'habitants.

Par ailleurs, on décide que si  $s$  et  $o$  sont tous les deux mis en place, soit par notre opérateur, soit par les concurrents (ie  $(s, o) \in S \times O \cup S_c \times O_c$ , alors, avec une probabilité  $q$ , on intègre le service  $s$  à l'offre  $o$ , c'est-à-dire que l'on fixe  $b_{s,o}$  à 1. Dans le cas où  $b_{s,o} = 1$ ,  $m_{s,o}$  sera choisi aléatoirement dans  $[75; 125] = 75 + 50[0; 1]$ .

De plus, pour  $o \in O_c$ , on choisira  $p_o$  dans  $\frac{1}{\bar{m}} \sum m_{s,o}[20; 30] = \frac{10}{\bar{m}} \sum m_{s,o}(2 + [0; 1])$  aléatoirement. On prendra aussi pour tout  $o \in O \cup O_c$ ,  $u_{d,o}$  dans  $\frac{1}{\bar{m}} \sum m_{s,o}[1; 2]$ . En effet, il est naturel d'imaginer une certaine corrélation entre ce qui compose l'offre, à savoir les minutes disponibles dans chacun des services, et l'utilité que la classe  $d$  lui attribue, ainsi qu'avec le prix de l'offre.

De plus, les paramètres  $\alpha_{s,s'}$ , les coûts par temps de connexion et par client entre les services  $s$  et  $s'$ , lorsque  $s, s' \in S$ , seront choisis dans  $\frac{\bar{p}}{8qS\bar{m}}[0.75; 1.25] = \frac{\bar{p}}{8qS\bar{m}}(0.75 + 0.5 * [0; 1])$ . On

prendra alors  $\beta^S$  dans  $\bar{p}\bar{N}D/8[0.8; 1.2] = 0.2\bar{p}\bar{N}D(2 + [0; 1])$ . On peut alors prendre les  $\alpha^{\bar{s}}$  et  $\alpha^{\hat{s}}$  dans  $\frac{\bar{p}}{16qS^2\bar{m}}[0.75; 1.25]$ , puisqu'on remarque une homogénéité entre ces termes : s'ils sont tous multipliés par une constante, le profit reste inchangé.

### 3.2.2 Calculs de moyenne

Voyons les ordres de grandeurs des paramètres que l'on va maintenant calculer, pour observer leur cohérence. On a l'expression du coût  $c^S = \beta^S + N_{total}\mathcal{M}(\mathcal{M}(\gamma_o^S, nB), n)$ , où on a  $\gamma_{o,o'}^S = \frac{1}{B_{o'}} \sum \alpha_{s,s'}^S m_{s,o} b_{s',o'}$ , qui est une somme de  $q^2 S^2$  termes de l'ordre de  $\bar{\alpha}^S \bar{m}$ . Or  $B_{o'}$  est de l'ordre de  $qS$ , donc  $\bar{\gamma}^S = qS\bar{\alpha}^S \bar{m} = \bar{p}/8$ . On a alors  $\bar{c}^S = \bar{N}D\bar{p}/4$ , qui est le quart de la recette totale cumulée des opérateurs. C'est un ordre de grandeur cohérent, puisqu'on peut bien imaginer des profits positifs mais tout de même contraints par la fonction de coût.

On remarque que les autres fonctions de coût du système apparaissent toujours sous la forme d'un rapport. Leur ordre de grandeur n'est donc pas pertinente dans l'absolu, seul leur ordre de grandeur relatif compte. On remarque alors que  $\bar{c}$  est de l'ordre de la moitié de  $\bar{c}$ , tandis que  $\dot{c}$  est lui de l'ordre de  $S_c/(2S)$  fois  $\bar{c}$ .

La dérivée du coût  $\frac{\partial c}{\partial n_o} = \mathcal{M}(\gamma, nB)$  est elle de l'ordre de  $\bar{\gamma}$ . Pour  $c = c^S$ , la dérivée est donc de l'ordre de  $\bar{p}/8$ . Remarquons que l'on a toujours pris  $\bar{\beta} = \bar{N}D\bar{\gamma}$  qui est de l'ordre de  $c/2$ , donc la dérivée est de l'ordre de  $\bar{\gamma}$ , qui est de l'ordre de  $c/(2\bar{N}D)$ . En particulier  $\bar{\gamma}/c$  est proche de  $1/(2\bar{N}D)$  qui ne dépend pas du coût.

Ainsi, les coefficients de la matrice  $\Gamma = \frac{\bar{\Lambda}\bar{c} - \Lambda\dot{c}}{\bar{c}}(\gamma^S - \frac{c^S}{\bar{c}}\bar{\gamma}) + \frac{\bar{\Lambda}\bar{\gamma} - \Lambda\dot{\gamma}}{\bar{c}}c^S - \gamma^S$  sont de l'ordre de  $\left[\bar{\Lambda}/(4\bar{N}D) - \Lambda S_c/(4\bar{N}DS)\right]\bar{N}D\bar{p}/4 - \bar{p}/8 = -(1 + \frac{S_c + S}{S}\Lambda)\bar{p}/16$ . Cette expression dépend donc de  $\Lambda$  qui dépend fortement des prix, puisque c'est la proportion d'occupation de notre opérateur par les utilisateurs. Cependant, cette expression est bien négative en générale, pour tout  $\Lambda$ , car les variations du revenu d'interconnexion par variation de répartition des clients dans les offres sont négligeables devant celles du coût imputé à notre opérateur.

On a alors  $\kappa_o = \mathcal{M}(\Gamma_o, Bn) + \frac{\dot{c} + \ddot{c}}{\bar{c}} \frac{M_o}{N_{total}\mathcal{M}(M, n)} c^S (\Lambda - 1_{o \in O})$  dont l'ordre de grandeur s'écrit  $\bar{\Gamma} + \frac{S + S_c}{2S} \frac{\bar{c}^S}{\bar{N}D} (\Lambda - 1_{o \in O}) = -(3 + \frac{3S_c + S}{S + S_c}\Lambda)\bar{p}/16 + \frac{S + S_c}{2S} \frac{\bar{p}}{4} (\Lambda - 1_{o \in O})$ .

## 3.3 Calcul numérique

Dans cette section, nous allons voir ce que le calcul numérique nous a donné. En particulier, nous commenterons les résultats et discuterons les chances de réussite de notre algorithme.

### 3.3.1 Les

## Conclusion

Nous avons réussi à écrire notre algorithme, et, comme nous l'avons vu, nous avons réussi à l'appliquer à des cas simples. Cependant, l'implémentation de notre problème d'optimisation a été difficile, notamment d'un point de vue intervalliste. Les approximations conséquentes de nos formulations extensives n'ont pas permis de localiser de solution par la *Hull Consistency*.

Cependant, les bases sont là, et il est peut-être possible de mieux réussir à localiser la solution par la méthode des intervalles, si l'on affine le calcul par intervalles ou si l'on trouve des fonctions de réduction plus efficaces.

Par ailleurs, comme Khalid MEHL l'a déjà suggéré, il doit être possible d'aller plus loin dans l'étude mathématique du marché des télécommunications en utilisant des propriétés de la théorie des jeux. Cependant, il faudrait pour cela réussir à trouver plus facilement les prix optimaux d'un opérateur  $S$  quand ceux de ses concurrents sont fixés. On obtient ainsi la Best-Reply, et aboutir à des équilibres de Nash.





# Bibliographie

- [1] G. William Walster Eldon Hansen. *Global Optimization using Interval Analysis*. Marcel Dekker, Inc, 2004.
- [2] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimisation*, 2001.
- [3] Khalid Mehl. Tarification pour l'optimisation du revenu global des services voix dun opérateur. Technical report, Ecole Polytechnique et OrangeLabs, 2009.
- [4] Frédéric Messine. *L'Optimisation Globale par Intervalles : de l'Etude Théorique aux Applications*. PhD thesis, ENSEEIHT, 2006.