

$$W(\alpha, x, \xi) \Rightarrow \max$$

Séminaire Mathématique de Béjaïa (LaMOS)

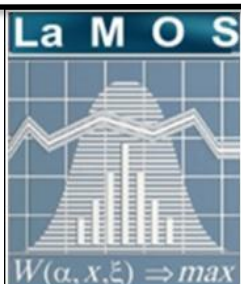
Revue Spécialisée de Mathématiques et Applications

*Compte rendu des séances
2009-2010*



Dr RABTA Boualem (Wirtschaftsuniversität Wien - Autriche)

Sous la direction de Pr. Djamil Aïssani et Dr. Kamal Kabyl



Publication et Edition

LaMOS Research Unit
University of Béjaïa

<http://www.lamos.org>

Volume 9
Béjaïa 2010

ISSN : 1112-9433

SÉMINAIRE MATHÉMATIQUE DE BÉJAÏA

Compte rendu des séances 2009-2010.

Séminaire Mathématique de Béjaïa.

Responsable : *M^r* KABYL KAMAL.

Adresse : Laboratoire de Modélisation et d'Optimisation des Systèmes.

Université de Béjaïa, Route Targa Ouzamour, 06000 (Algérie).

Tel/Fax : (213) 34 21 51 88.

E-mail : lamos _ Bejaia@hotmail.com

©Publications du Laboratoire LAMOS, 2010.

Tous droits de traduction, de reproduction
et d'adaptation réservés pour tous les pays.

Table des matières

partie I Méthodes Cybernétiques et Optimisation

1 Programmation bi-niveaux : application dans le domaine de transport <i>Aicha ANZI et Mohammed Said RADJEF</i>	3
2 Sur la théorie des jeux évolutionnaire Multicritère <i>Fatiha BARACHE</i>	7
3 L'Equilibre Social Idéal dans un Jeu Multicritère non Coopératif avec Contraintes <i>Karima FAHEM et Mohammed Said RADJEF</i>	13

partie II Systèmes avec Rappel et Réseaux

4 Plongement et placement de certaines classes d'arbres dans l'hypercube <i>Kamal KABYL et Abdelahafid BERRACHEDI</i>	19
5 Performance and Optimization of P2P Networks for IP Multimedia applications <i>Mourad AMAD et Djamil AÏSSANI</i>	23
6 Optimisation de consommation d'énergie dans les réseaux de capteurs sans fil <i>Samra BOULFEKHAR</i>	27

partie III Processus Aléatoires et Applications

7 Comparabilité et monotonie : Méthodologie pour l'analyse du système M/G/1 avec rappels et feedback	
---	--

Mohamed BOUALEM 33

partie IV Contrôle Statistique de la Qualité

8 Some statistical aspects to take into account when studying strong stability of stochastic models

Aïcha BARECHE 41

partie V Index des Auteurs

Index 51

Index 51

Première partie

Méthodes Cybernétiques et Optimisation

Programmation bi-niveaux : application dans le domaine de transport

Aicha ANZI et Mohammed Said RADJEF

Laboratoire de Modélisation et d'Optimisation des Systèmes (LAMOS)
Université de Béjaïa, Béjaïa 06000, Algérie
Tél. (213) 34 21 51 88

1.1 Introduction

Le problème de programmation bi-niveaux est un problème d'optimisation hiérarchique, avec deux niveaux de décision. Le premier niveau est appelé Leader et le deuxième Suiveur. En d'autres termes, un programme bi-niveau est un programme mathématique standard avec y est contraint d'être solution optimale pour le programme

$$\max\{f(x, y) : g(x, y) \leq 0\} \quad (1.1)$$

Ce qui donne le programme bi-niveau suivant :

$$\begin{aligned} \max_x F(x, y) \\ \text{s.c. } G(x, y) \leq 0 \\ \max_y f(x, y) \\ \text{s.c. } g(x, y) \leq 0, \end{aligned} \quad (1.2)$$

En se basant sur le programme bi-niveau (1.2), on donne les notations et les définitions suivantes :

a. *Ensemble des contraintes du problème*

$$S = \{(x, y) : G(x, y) \leq 0, g(x, y) \leq 0\}.$$

b. *Ensemble réalisable du Suiveur pour un x fixé*

$$S(x) = \{y : g(x, y) \leq 0\}.$$

c. *Projection de S sur l'espace du Leader*

$$P(X) = \{x : \exists y, (x, y) \in S\}.$$

d. *Ensemble des réactions rationnelles du Suiveur pour $x \in P(X)$*

$$R(x) = \{y : y = \arg \max[f(x, \hat{y}) : \hat{y} \in S(x)]\}.$$

e. *Région induite*

$$RI = \{(x, y) \in S, y \in R(x)\}.$$

La région induite RI représente l'ensemble réalisable du problème (1.2) sur lequel le Leader optimise sa fonction de gain. Cette région est non convexe et souvent discontinue, ce qui rend les problèmes bi-niveaux non convexes et difficiles à résoudre. Ces caractéristiques restent vraies même dans le cas où toutes les fonctions dans (1.2) sont linéaires et dans le cas d'absence des contraintes $G(x, y) \leq 0$.

1.2 Domaines d'application de la programmation bi-niveau

Les applications de la programmation bi-niveau sont nombreuses et variées, par exemple dans le domaine de production, dans le secteur de l'énergie, en finances, en contrôle de pollution, etc.

Dans le domaine de transport, aussi, la programmation bi-niveau a été appliquée pour résoudre les problèmes de tarification optimale, par exemple, le problème de tarification routière (détermination des péages).

1.2.1 Tarification optimale dans un système de péages routiers

Définition du péage : c'est un droit que l'on paie pour emprunter une autoroute, un pont, etc.

Position du problème

Le problème est de déterminer de façon optimale les péages sur un tronçon d'un réseau routier en tenant compte des réactions des usagers du réseau. L'objectif de cette opération peut être la réduction de la congestion ou la maximisation d'un revenu. Nous avons, donc, deux niveaux de décision : le premier niveau ou Leader représenté par le gestionnaire du réseau, et le deuxième niveau ou Suiveur représenté par les usagers.

Dans le cas de péage pour la maximisation du revenu, nous avons le scénario suivant :

- Le gestionnaire du système d'autoroutes est autorisé à mettre en œuvre un système de **péage** sur un ensemble (ou un sous ensemble) d'un réseau routier.
- L'objectif du gestionnaire du réseau est de maximiser son revenu.
- Les usagers souhaitent minimiser leurs coûts de voyage.
- Un plan de péage optimal est tel que les coûts de péage ne sont pas trop élevés, sinon les usagers se détourneront de l'utilisation des arcs à péage.
- Une fois le gestionnaire du réseau a fixé les péages, la réaction des usagers consiste à choisir leurs itinéraires de manière à minimiser le coût total d'un voyage (distance, temps, etc) plus le péage.

Formulation mathématique du problème de péage

Désignons par \mathcal{A} l'ensemble des arcs (routes) du réseau et par $\bar{\mathcal{A}}$ le sous ensemble des arcs à péages. Le problème du Leader est donné par le programme mathématique :

$$\max_{T,x} \sum_{a \in \bar{\mathcal{A}}} T_a x_a \tag{1.3}$$

$$l_a \leq T_a \leq u_a, \forall a \in \bar{\mathcal{A}},$$

où

- T_a et x_a désignent le péage et le flot sur l'arc a respectivement, et
- l_a (respectivement u_a) est la borne inférieure (respectivement supérieure) sur le péage.

Les usagers cherchent à minimiser les coûts totaux de leurs voyages. Le modèle mathématique est donné par le programme linéaire suivant :

$$\begin{aligned} \min_{f,x} \quad & \sum_{a \in \mathcal{A}} c_a x_a + \sum_{a \in \bar{\mathcal{A}}} T_a x_a \\ & \sum_{p \in P_{rs}} f_p^{rs} = d_{rs}, \forall (r,s) \in \Theta, \\ & x_a = \sum_{(r,s) \in \Theta} \sum_{p \in P_{rs}} \delta_{a,p}^{rs} f_p^{rs}, \forall a \in \mathcal{A}, \\ & f_p^{rs} \geq 0 \forall p \in P_{rs}, \forall (r,s) \in \Theta. \end{aligned} \tag{1.4}$$

- La fonction objectif est la somme des coûts qui résultent des péages T_a ($a \in \bar{\mathcal{A}}$) et les autres coûts (durée, distance, etc.) qui sont agrégés dans le vecteur c_a .
- La première contrainte exprime la satisfaction de la demande sur le voyage, d_{rs} , d'une origine r vers la destination s .

- La deuxième contrainte exprime le fait que le flot sur un arc a est égal à la somme des flots sur tous les chemins utilisant cet arc, avec :

$$\delta_{a,p}^{rs} = \begin{cases} 1 & \text{si le chemin } p \in P_{rs} \text{ passe par l'arc } a, \\ 0 & \text{sinon} \end{cases} \quad (1.5)$$

Donc il existe une relation entre les problèmes (1.3) et (1.4)

- Puisqu'ils utilisent le même ensemble de variables, à savoir, les vecteurs de péage T_a et les flots x_a .
- De même la fonction objectif du gestionnaire ne peut être évaluée tant que les flots ne sont pas connus (ces derniers ne sont pas contrôlés directement par le gestionnaire, mais solution du programme correspondant au problème des usagers).

Ce qui nous donne la formulation bi-niveau :

$$\begin{aligned} & \max_{T,x} \sum_{a \in \bar{\mathcal{A}}} T_a x_a \\ & \quad l_a \leq T_a \leq u_a, \quad \forall a \in \bar{\mathcal{A}}, \\ & \quad \min_{f,x} \sum_{a \in \mathcal{A}} c_a x_a + \sum_{a \in \bar{\mathcal{A}}} T_a x_a \\ & \quad \sum_{p \in P_{rs}} f_p^{rs} = d_{rs}, \quad \forall (r,s) \in \Theta, \\ & \quad x_a = \sum_{(r,s) \in \Theta} \sum_{p \in P_{rs}} \delta_{a,p}^{rs} f_p^{rs}, \quad \forall a \in \mathcal{A}, \\ & \quad f_p^{rs} \geq 0 \quad \forall p \in P_{rs}, \quad \forall (r,s) \in \Theta. \end{aligned}$$

1.2.2 Autres applications

Les applications de la programmation bi-niveau dans le domaine de transport ne se limitent pas à cet exemple.

- Le problème de tarification dans le transport public.
- Le problème de tarification pour le transport de marchandises.
- Le problème d'évaluation des prix des billets dans le transport aérien.

Références

1. J.F. Bard : Practical bilevel optimization : algorithms and applications. Kluwer academic publishers, Dordrecht (1998)
2. B. Colson, P. Marcotte and G. Savard : Bilevel programming : A survey. 4OR A Quarterly, J. Oper. Res., (2007)
3. S. Dempe : Foundations of Bilevel Programming. Kluwer academic publishers, Dordrecht (2000)
4. J.B.E. Etoa : Optimisation hiérarchique : théorie, algorithmes et applications. Publibook, France, (2008)

Sur la théorie des jeux évolutionnaire Multicritère

Fatiha BARACHE

Laboratoire de Modélisation et d'Optimisation des Systèmes (LAMOS)
Université de Béjaïa, Béjaïa 06000, Algérie
Tél. (213) 34 21 51 88

2.1 Introduction

La théorie des jeux évolutionnaire (TJE) s'est développée à la suite des travaux du biologiste John Maynard Smith [1]. La TJE a connu un très grand intérêt ses dernières années notamment dans les sciences économiques et sociales, son champs d'application s'élargie aux réseaux de télécommunication, aux réseaux de transport, etc. La théorie des jeux classique impose un degré de rationalité très élevé sur les individus. Par contre, la théorie des jeux évolutionnaire permet de prédire et de décrire les choix des différents individus lorsqu'on est en face d'une hypothèse de rationalité plus faible [4].

Le concept d'équilibre, décrit dans les TJE, est le concept de stratégie évolutionnairement stable (ESS) qui est un raffinement de l'équilibre de Nash [4]. Pour qu'une stratégie soit un équilibre de Nash, il faut qu'un seul joueur ne puisse pas tirer profit d'un changement de stratégie. Dans l'équilibre de Nash, si plusieurs joueurs changent leurs décisions, il se peut bien qu'ils en tirent un gain. Or, la notion de ESS est plus robuste en permettant à toute une fraction de la population de changer leurs décisions [1,3].

L'approche qui consiste à considérer un seul critère ou objectif qu'un individu souhaite maximiser, n'est souvent pas suffisante pour décrire les besoins et le comportement des individus. Kiran Somasundaram [2] étend la notion de stabilité évolutionnaire aux jeux avec fonctions de paiement vectorielles [5]. Il espère que cette notion de stabilité évolutionnaire multicritère, permettra de modéliser une classe plus large des problèmes d'interaction. Comme dans les jeux à un seul objectif, il se focalise sur les interactions par paires symétriques dans une large population. On ne compromet pas les interactions entre plus que deux individus à un instant.

2.2 Notation

On considère les mêmes hypothèses qu'on a posé dans la section précédente. On a le jeu

$$\langle I, S, U \rangle. \quad (2.1)$$

– I : population d'individus (joueurs). S : l'ensemble des issues du jeu en stratégies pures. Δ : l'ensemble des stratégies mixtes, tel que $\Delta = \{x \in \mathbb{R}_+^n, \sum_{i=1}^n x_i = 1\}$.

– U : on associe à chaque joueur une sequence de matrices des gains. Supposons qu'il y a l critères (objectifs) en jeux. Pour le joueur I on lui associe les matrices des gains suivantes A_1, A_2, \dots, A_l avec

A_i : la matrice des gains du joueur I en considérant le critère i .

$B_i = A_i'$: la matrice des gains du joueur II en considérant le critère i .

La fonction de gain quand le joueur I joue la stratégie $x \in \Delta$ et le joueur II joue la stratégie $y \in \Delta$ est donnée par

$$U(x, y) = (u^1, u^2, \dots, u^l)(x, y) = (u^1(x, y), u^2(x, y), \dots, u^l(x, y)) = (x' A_1 y, x' A_2 y, \dots, x' A_l y)$$

Remarque 2.1

- Les équilibres Pareto Nash symétrique sont notés par Δ^{PNE} .
- L'ensemble des meilleures réponses Pareto pour une stratégie y est noté par $\beta^{P^*}(y)$.

Les différentes relations de classement \succ de \mathbb{R}^l , que nous considérons, sont indiquées dans le tableau suivant.

Notation	Définition	Nom
$x \gg y$	$x_i > y_i, \quad i = 1, 2, \dots, l$	Strict component-wise order
$x > y$	$x_i \geq y_i, \quad i = 1, 2, \dots, l$ et $x \neq y$	Component-wise order
$x >_{\text{lex}} y$	$x_j > y_j, \quad j = \min\{i, x_i \neq y_i\} \quad j \leq l$	Lexicographic component-wise order
$x_\lambda > y_\lambda$	$\lambda'x > \lambda'y, \quad \lambda \geq 0$	λ -scalarized order

2.3 Stabilité évolutionnaire

Définition 2.1 Une stratégie $x \in \Delta$ est dite évolutionnairement stable (en respectant la relation de classement ou d'ordre \succ), si $\forall y \in \Delta, y \neq x, \exists \varepsilon_y$ tel que $\forall \varepsilon \in (0, \varepsilon_y)$, on a $U(x, \varepsilon y + (1 - \varepsilon)x) \succ U(y, \varepsilon y + (1 - \varepsilon)x)$.

Remarque 2.2 Les différentes relations de classements donnent lieu à plusieurs définitions de la stabilité évolutionnaire dans le cadre multi-objectifs.

2.3.1 Stabilité évolutionnaire fortement idéale

Quand on choisit la relation d'ordre "strict component-wise", nous obtenons la stabilité évolutionnaire idéalement forte. La séquence de composants de jeux symétriques est donnée par les matrices de paiement $A_m, B_m = A'_m, 1 \leq m \leq l$ qui correspondent aux matrices dans le jeu multi-objectifs original. On note $G_m^c, 1 \leq m \leq l$: un jeu symétrique en considérant la $m^{\text{ème}}$ composante. $\langle I, S, u^m \rangle, \beta^{m*}(y)$: les meilleures réponses correspondantes pour le jeu G_m^c . $\Delta^{NE(G_m^c)}$: l'ensemble des équilibres de Nash symétrique pour le jeu G_m^c . $\Delta^{ESS(G_m^c)}$: l'ensemble des équilibres évolutionnairement stables pour le jeu G_m^c . Δ^{SIESS} : l'ensemble des stratégies évolutionnairement stables fortement idéales.

Définition 2.2 Une stratégie $x \in \Delta$ est dite évolutionnairement stable (en respectant la relation de classement ou d'ordre \succ), si $\forall y \in \Delta, y \neq x, \exists \varepsilon_y$ tel que $\forall \varepsilon \in (0, \varepsilon_y)$, on a $U(x, \varepsilon y + (1 - \varepsilon)x) \gg U(y, \varepsilon y + (1 - \varepsilon)x)$.

Proposition 1 $\Delta^{SIESS} = \bigcap_m \Delta^{ESS(G_m^c)}$.

2.3.2 Stabilité évolutionnaire idéale

Quand on choisit la relation d'ordre "component-wise", nous obtenons la stabilité évolutionnaire idéale. On considère $G_m^c, 1 \leq m \leq l$. On note Δ^{IESS} : l'ensemble des stratégies évolutionnairement stables idéales.

Définition 2.3 Une stratégie $x \in \Delta$ est dite évolutionnairement stable (en respectant la relation de classement ou d'ordre \succ), si $\forall y \in \Delta, y \neq x, \exists \varepsilon_y$ tel que $\forall \varepsilon \in (0, \varepsilon_y)$, on a $U(x, \varepsilon y + (1 - \varepsilon)x) \geq U(y, \varepsilon y + (1 - \varepsilon)x)$.

Proposition 2 $\Delta^{IESS} = \{x \in \bigcap_m \Delta^{NE(G_m^c)} \mid U(x, y) > U(y, y), \forall y \in \bigcap_m \beta^{*m}(x), y \neq x\}$.

2.3.3 Stabilité évolutionnaire lexicographique

Quand on choisit la relation d'ordre "Lexicographic component-wise", nous obtenons la stabilité évolutionnaire lexicographique. On considère $G_m^c, 1 \leq m \leq l$. On note Δ^{LESS} : l'ensemble des stratégies évolutionnairement stables lexicographiques.

Définition 2.4 Une stratégie $x \in \Delta$ est dite évolutionnairement stable (en respectant la relation de classement ou d'ordre \succ), si $\forall y \in \Delta, y \neq x, \exists \varepsilon_y$ tel que $\forall \varepsilon \in (0, \varepsilon_y)$, on a $U(x, \varepsilon y + (1 - \varepsilon)x) >_{lex} U(y, \varepsilon y + (1 - \varepsilon)x)$.

Proposition 3 $\Delta^{LESS} = \{x \in \bigcap_m^j \Delta^{NE(G_m^c)} \mid u^1(x, y) \geq u^1(y, y), \dots, u^{j-1}(x, y) \geq u^{j-1}(y, y), u^j(x, y) > u^j(y, y), \forall y \in \bigcap_m^j \beta^{*m}(x), y \neq x\}$.

2.3.4 Stabilité évolutionnaire Biaisée

Quand on choisit la relation d'ordre " λ -scalarized order " , nous obtenons la stabilité évolutionnaire Biaisée. On considère le jeu scalarisé $G(\lambda)$. On note Δ^{BESS} : l'ensemble des stratégies évolutionnairement stables biaisées. $\lambda'U(x, y)$: la fonction de gain dans le jeu $G(\lambda)$, $x \in \Delta, y \in \Delta$.

Proposition 4 $\Delta^{BESS} = \{x \in \Delta^{PNE(G)} \mid y \neq x, y \in \beta^{*P}(x),$
 $\Rightarrow x \text{ domine } y \text{ comme réponse à la stratégie opposant } y\}$.

2.4 Replicateur Dynamique

On considère que les individus sont programmés à jouer un jeu symétrique à deux joueurs multicritères avec une fonction de gain U . $p_i(t)$ le nombre d'individus qui sont programmés à jouer la stratégie pures $s_i \in X$ à l'instant t . $p(t) = \sum_i^n p_i(t) > 0$ la population totale. $x(t) = (x_1(t), \dots, x_k(t))$ est l'état de la population. $x_i(t)$ est la proportion des individus qui sont programmés à jouer la stratégie pure s_i à l'instant t : $x_i(t) = p_i(t)/p(t)$. $x(t) \in \Delta$. $U(e^i, x)$: est le gain espéré quand la stratégie pure s_i est jouée contre le profil de stratégie x . $U(x, x)$: est le gain espéré quand le profil de stratégie x est joué dans la population (le gain moyen dans la population avec la stratégie x). $U(x, x) = \sum_{i=1}^n x_i U(e^i, x)$. \dot{x}_i : est la proportion d'individus qui joueront la stratégie pure s_i à la prochaine génération.

La dynamique de la population correspondante est :

$$\dot{x}_i = x_i[U(e^i, x_{-i}) - U(x, x)] \quad (2.2)$$

On suppose que les fonctions gains son biaisées par λ qui reflète l'importance de chaque fonction. Le gain moyen devient : $\lambda'U(x, x)$. β : le taux de naissance (taux de croissance) dans la population. δ : le taux de mort dans la population. La dynamique de la population devient : $\dot{p}_i = [\beta + \lambda'U(e^i, x) - \delta]p_i$. La dynamique correspondante à x_i est :

$$\dot{x}_i = [\lambda'U(e^i, x) - \lambda'U(x, x)]x_i \quad \dots \quad BRE(\lambda) \quad (2.3)$$

$BRE(\lambda)$: le replicateur dynamique biaisé.

Remarque 2.3 Cette dynamique correspond aux jeux où tous les joueurs choisissent les mêmes poids pour les différents objectifs de leurs vecteurs fonction gain.

2.4.1 Equilibre de Nash Pareto

L'ensemble des équilibres de Nash Pareto pour un vecteur de fonction de gain du jeu symétrique à deux joueurs est donné par :

Définition 2.5 $\Delta^{PNE} = \{x \in \Delta \mid x = \arg \max_{z \in \Delta}^P U(z, x)\}.$

On note $\mathcal{U}(z, x)$ l'espace de décisions pour le joueur I quand le joueur II joue la stratégie $x \in \Delta$. Dans un programme linéaire multicritères, \mathcal{U} représente un polyèdre convexe avec des sommets dans $U(e^i, x), i = 1 \dots n$.

Lemme 2.1.

$\Delta^{PNE} = \{x \in \Delta \mid \lambda U(e^i, x) - \lambda U(x, x) \in C(x) \text{ et } U(e^i, x) \in \text{face Pareto Dominante de } \mathcal{U}\}.$

2.4.2 Etats stationnaires

Les états stationnaires donnés par la dynamique $BRE(\lambda)$ sont comme suit :

Définition 2.6 $\Delta^\circ(\lambda) = \{x \in \Delta \mid \lambda U(e^i, x) = \lambda U(x, x)\}.$

On note l'ensemble des états stationnaires intérieurs de $BRE(\lambda)$ par $\Delta^{\circ\circ}(\lambda) = \Delta^\circ(\lambda) \cap \text{int}(\Delta).$

Proposition 5 $\{e^1, e^2, \dots, e^n\} \cup \Delta^{PNE}(\lambda) \subset \Delta^\circ(\lambda).$

$$\Delta^{\circ\circ}(\lambda) = \Delta^{PNE}(\lambda) \cap \text{int}(\Delta).$$

Proposition 6 Si $x \in \Delta^{BESS}$ alors $BRE(\lambda)$ est asymptotiquement stable.

2.5 References

1. Price G.R. Smith M. *The logic of animal conflict*. Nature. 246 15 :18, 1973.
2. Somasundaram K. *Evolutionary Stability in Multi-Objective Games*. The Institute for systems research. 2009.
3. Taylor P.D. Joncker L.B. *Evolutionary stable strategies and game dynamics*. Mathematical biosciences, Elsevier. 40 : 145-156, 1978.
4. Weibull J.W. *Evolutionary Games Theory*. MIT Press, 1997.
5. Zenely M. *Games with multiple payoffs*. International Journal of Game Theory. 4(4) : 179-191, 1975.

L'Equilibre Social Idéal dans un Jeu Multicritère non Coopératif avec Contraintes

Karima FAHEM et Mohammed Said RADJEF

Laboratoire de Modélisation et d'Optimisation des Systèmes (LAMOS)
 Université de Béjaïa, Béjaïa 06000, Algérie
 Tél. (213) 34 21 51 88

3.1 Introduction

L'objectif de ce travail est d'introduire le concept d'équilibre idéal dans les jeux multicritères avec contraintes. Ce concept a été défini pour les jeux multicritères par Voorneveld *et al* en 2000. Quelques propriétés seront données ainsi qu'une caractérisation de l'équilibre, ce qui nous permettra par la suite d'étudier son existence.

Mots clé : Multicriteria games, Ideal equilibria, Constrained games, social equilibria.

3.2 Notations et préliminaires

Pour tout $m \in \mathbb{N}$, on note par \mathbb{R}_+^m l'ordhand non négatif de \mathbb{R}^m et par $int\mathbb{R}_+^m$ son intérieur, ainsi $\mathbb{R}_+^m = \{x = (x_1, \dots, x_m) \in \mathbb{R}^m, x_i \geq 0, \forall i \in \{1, \dots, m\}\}$, $int\mathbb{R}_+^m = \{x = (x_1, \dots, x_m) \in \mathbb{R}^m, x_i > 0, \forall i \in \{1, \dots, m\}\}$. Notons par Δ_m et $int\Delta_m$ le simplexe unité de \mathbb{R}^m et son intérieur respectivement $\Delta_m = \{\mu = (\mu_1, \dots, \mu_m) \in \mathbb{R}_+^m : \sum_{i=1}^m \mu_i = 1\}$, $int\Delta_m = \{\mu = (\mu_1, \dots, \mu_m) \in int\mathbb{R}_+^m : \sum_{i=1}^m \mu_i = 1\}$. Pour un ensemble B , on note 2^B l'ensemble de toutes les parties de B et coB son enveloppe convexe. Soit $I = \{1, \dots, n\}$ un ensemble d'indices. Pour tout $i \in I$, soit X_i un ensemble d'un espace vectoriel topologique, $X = \prod_{i \in I} X_i$ et $X_{-i} = \prod_{j \in I, j \neq i} X_j$. Pour tout $x \in X$, $x_i \in X_i$ est la i^{eme} coordonnée de x ou la projection de x sur X_i , $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in X_{-i}$ est la projection de x sur X_{-i} et on écrit $x = (x_i, x_{-i})$.

Soient $a, b \in \mathbb{R}^m$, alors on note : $a = b$ si $a_j = b_j$ pour tout $j \in \{1, \dots, m\}$;

Pour tout $x, y \in \mathbb{R}^m$, on désigne par $\langle x, y \rangle = \sum_{k=1}^m x_k y_k$ le produit scalaire des vecteurs

x et y ; on note e_k le k^{eme} vecteur de la base canonique de \mathbb{R}^m et $e(m) = (1, \dots, 1) \in \mathbb{R}^m$ le vecteur unité. Considérons le jeu multicritère avec contraintes sous sa forme normale

$$G = (I, X_i, C_i, f_i) \quad (3.1)$$

où $I = \{1, \dots, n\}$ est l'ensemble des joueurs, et pour tout joueur $i \in I$, X_i est l'ensemble de ses stratégies pures, $f_i = (f_{i1}, f_{i2}, \dots, f_{ir(i)}) : X \rightarrow \mathbb{R}^{r(i)}$ sa fonction multicritère avec $r(i)$ critères et $r(i) \in \mathbb{N}$, $C_i : X_{-i} \rightarrow 2^{X_i}$ sa correspondance des contraintes qui restreint les stratégies du joueur i au sous ensemble $C_i(x_{-i}) \subset X_i$ lorsque le reste des joueurs choisissent leurs stratégies dans X_{-i} .

3.3 Concepts d'équilibres

Ding [2], en 2000, a introduit l'équilibre Pareto fort et Pareto faible dans les jeux multi-objectifs avec contraintes et étudia leurs existences dans un H -espace moyennant des vecteurs poids, voir aussi [1, 3].

Définition 3.1 [2] *Une multistratégie \bar{x} est dite équilibre Pareto efficace (resp. Pareto faiblement efficace) pour le jeu multicritère avec contraintes (3.1) si*

- a) *Pour tout $i \in I$, $\bar{x}_i \in C_i(\bar{x}_{-i})$; et*
- b) *Il n'existe pas de stratégie $y_i \in C_i(\bar{x}_{-i})$ telle que $f_i(y_i, \bar{x}_{-i}) \geq f_i(\bar{x})$ (resp. $f_i(y_i, \bar{x}_{-i}) > f_i(\bar{x})$).*

Dans ce présent papier, nous introduisons le concept d'équilibre idéal pour les jeux multicritères avec contraintes. Rappelons que cet équilibre a été défini dans les jeux multicritères non coopératifs sous forme normale par Voorneveld *et al* [6], dont l'existence a été établie par Radjef et Fahem [4]. C'est un équilibre qui maximise tous les critères de tous les joueurs simultanément.

Définition 3.2 *Une multistratégie \bar{x} est dite équilibre social idéal pour le jeu (3.1) si :*

1. *Pour tout $i \in I$, $\bar{x}_i \in C_i(\bar{x}_{-i})$ et*
2. *$f_i(\bar{x}) \geq f_i(y_i, \bar{x}_{-i}) \quad \forall y_i \in C_i(\bar{x}_{-i})$.*

3.4 Caractérisation de l'équilibre social idéal

Dans cette section, nous suivons le travail de Voorneveld *et al* pour caractériser l'équilibre social idéal. Soit

$$J_\lambda = (I, (X_i)_{i \in I}, (g_i)_{i \in I}) \quad (3.2)$$

le jeu pondéré associé au jeu multicritère défini par (3.1), où chaque joueur a le même ensemble de stratégies que dans G et $g_i : X \rightarrow \mathbb{R}$ est définie par

$$g_i(x) = \sum_{k=1}^{r(i)} \lambda_{ik} f_{ik}(x) = \langle \lambda_i, f_i(x) \rangle, \quad \lambda_i \in \Delta_{r(i)}.$$

Définition 3.3 [6] Une collection $\Lambda \subset \prod_{i \in I} \Delta_{r(i)}$ de vecteurs poids est dite représentative du jeu (3.1) si pour toute organisation $i \in I$ et chacun de ses membres $k \in \{1, 2, \dots, r(i)\}$, il existe un vecteur poids qui attribue le poids unité à ce membre de l'organisation, autrement dit :

$$\forall i \in I, \forall k \in \{1, 2, \dots, r(i)\}, \exists \lambda = (\lambda_j)_{j \in I} \in \Lambda, \text{ avec } \lambda_i = e_k \in \Delta_{r(i)}.$$

Le théorème suivant nous donne une caractérisation de l'équilibre social idéal.

Théorème 3.1 Soient J_m le jeu multicritère (3.1), une multistratégie $x \in X$ et Λ une collection de vecteurs poids représentative pour J_m .

Les assertions suivantes sont équivalentes :

- a) $x \in X^{IS}(G)$
- c) $x \in \bigcap_{\lambda \in \prod_{i \in I} \Delta_{r(i)}} X^S(J_\lambda)$
- d) $x \in \bigcap_{\lambda \in \Lambda} X^S(J_\lambda)$

Où $X^{IS}(G)$ est l'ensemble des équilibres sociaux idéaux du jeu G et $X^S(J_\lambda)$ est l'ensemble des équilibres sociaux du jeu scalarisé J_λ

3.5 Existence de l'équilibre social idéal

Posons dans le jeu (3.1), $C_i : X \rightarrow 2^{X_i}$ et définissons les correspondances $A_i : X \rightarrow 2^{X_i}$ par

$$A_i(x) = \{x_i \in X_i, \exists \lambda \in \Lambda : \langle \lambda_i, f_i(x) \rangle - \langle \lambda_i, f_i(y_i, x_{-i}) \rangle < 0\}.$$

On a la proposition suivante :

Proposition 7 $\bar{x} \in X$ est un équilibre social idéal si et ssi

$$\begin{cases} \bar{x}_i \in C_i(\bar{x}) \quad \forall i \in I, \text{ et;} \\ C_i(\bar{x}) \cap A_i(\bar{x}) = \emptyset \quad \forall i \in I. \end{cases}$$

Pour l'existence d'un équilibre social idéal nous obtenons le théorème suivant

Théorème 3.2 *Supposons que pour tout $i \in I$, les conditions suivantes sont vérifiées :*

1. X_i est un sous ensemble non vide convexe et compact de \mathbb{R}^{l_i} ;
2. La correspondance C_i est continue ; et
3. Pour tout $x \in X$, $C_i(x)$ est non vide et convexe ;
4. Les fonctions f_i sont continues ; et
5. La fonction $y_i \mapsto f_i(y_i, x_{-i})$ est $\mathbb{R}_+^{r(i)}$ -quasi-concave-like.

Alors le jeu multicritère (3.1) admet un équilibre social idéal.

Références

1. Ansari Q.H and Zubair Khan, *On existence of Pareto equilibria for constrained multiobjective games*, Southeast Asian Bulletin of Mathematics 273-982 (2004)
2. Ding X.P, *Existence of Pareto equilibria for constrained multiobjective games in H-spaces* Comp. Math. Appl. 39, 125-134 (2000).
3. Ding X.P, Park J.Y and Jung I.H, *Pareto equilibria for constrained multiobjective games in L-convex spaces* Comp. Math. Appl. 46, 1589-1599 (2003).
4. Radjef M.S and Fahem K, *A note on ideal Nash equilibrium in multicriteria games*. Appl. Math. Lett. 2, 1105-1111 (2008).
5. Shafer W and Sonnenschein H, *Equilibrium in Abstract economies without ordered preferences*. Journal of Mathematical Economics 2, 345-348 (1975).
6. Voorneveld M and Grahn S and Dufwenberg M, *Ideal Equilibria in Non cooperative Multicriteria Games*, Mathematical Methods of Operations Research 52, 65-77 (2000).

Deuxième partie

Systemes avec Rappel et Réseaux

Plongement et placement de certaines classes d'arbres dans l'hypercube

Kamal KABYL et Abdelahafid BERRACHEDI

Laboratoire de Modélisation et d'Optimisation des Systèmes (LAMOS)
Université de Béjaïa, Béjaïa 06000, Algérie
Tél. (213) 34 21 51 88

4.1 Introduction

Un plongement de $G(V, E)$ dans l'hypercube est défini par la donnée d'une application injective φ de l'ensemble des sommets de G dans l'ensemble des sommets de Q_n , et d'une application P_φ de l'ensemble des arêtes de G dans l'ensemble des arêtes de Q_n , qui associe à chaque arête uv de G une arête $\varphi(u) \varphi(v)$ dans Q_n . Une classe importante, à étudier, est celle des arbres dans l'hypercube. Cette importance résulte de l'utilisation de ces arbres dans plusieurs domaines, à savoir : informatique, sciences sociales, recherche opérationnelle, optimisation combinatoire, théorie des réseaux électriques, etc. Un graphe $G = (V, E)$ est dit cubique s'il est plongeable dans Q_n pour un certain n .

On définit dans ce papier des nouvelles classes pour lesquelles la dimension cubique est déterminée. On a donné, aussi, le nombre maximum de copies de certaines topologies qu'on peut placer dans un hypercube de dimension donnée.

Un hypercube de dimension n , noté Q_n , est le graphe dont l'ensemble de ses sommets est le n -uplets binaires et deux sommets sont adjacents, si et seulement, s'ils, diffèrent en une seule coordonnée.

La Cn -évaluation, aux cas des arbres, est donnée comme suit : Un arbre T est Cn -valué si les arêtes de T sont marquées par les entiers de l'ensemble $\{1, 2, 3, \dots, n\}$ de sorte que pour toute chaîne P de T , il existe un entier $K \in \{1, 2, 3, \dots, n\}$ pour lequel un nombre impair d'arêtes de P est marquées par K . I. Havel et Moravek [7] ont montré qu'un graphe G est plongeable dans Q_n , si et seulement, s'il existe une Cn -évaluation de G .

4.2 Plongement et placement de certaines classes d'arbres

4.2.1 La classe AD_n

Pour $n \geq 1$, l'arbre AD_n est obtenu à partir de l'arbre binaire D_n en reliant un seul sommet de degré 1 à un nouveau sommet. AD_n possède donc 2^{n+1} sommets.

AD_3 est montré sur la figure suivante :

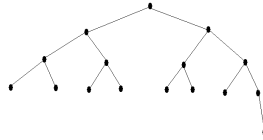


FIGURE 4.1.

Théorème 4.1 *Pour tout $n \geq 3$, $\dim(AD_n) = n + 1$.*

4.2.2 La classe $A\hat{D}_n$

Pour $n \geq 1$ l'arbre, $A\hat{D}_n$ est obtenu à partir de deux copies disjointes de AD_n , en reliant les racines par une arête. $A\hat{D}_2$ est donné par le graphe suivant :

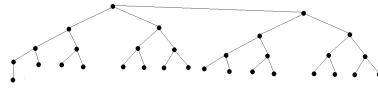


FIGURE 4.2.

Théorème 4.2 *L'arbre $A\hat{D}_n$ est plongable dans Q_{n+2} et $\dim(A\hat{D}_n) = n + 2$*

On peut parler d'un autre plongement, concernant ce type d'arbre, qui nécessite de trouver combien d'arbre de même topologie qu'on puisse plonger dans un hypercube de dimension donnée.

On peut faire une généralisation comme suit :

pour $n=1$, l'arbre $A\hat{D}_1$ est montré sur la figure suivante :

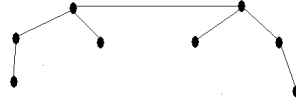


FIGURE 4.3.

Pour $k \geq 2$, l'arbre $A^k \widehat{D}_1$ est obtenu en reliant deux copies de $A^{k-1} \widehat{D}_1$ et un sommet de degré 3 de $A^{k-1} \widehat{D}_1$ à un sommet de degré 3 de $A^{k-1} \widehat{D}_1$ par une arête.

Proposition 1. *Le nombre maximum de copies de l'arbre $A \widehat{D}_1$, qu'on puisse placer dans un hypercube de dimension n est 2^{n-3} .*

4.2.3 La classe AB_n

Pour $n \geq 1$, l'arbre AB_n est obtenu de la manière suivante : AB_1 est le graphe de la figure suivante :

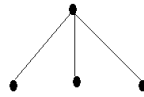


FIGURE 4.4.

Pour $n \geq 2$, AB_n est obtenu en reliant deux copies disjointes de AB_{n-1} , tel qu'un sommet de degré $n + 1$ de la première copie est relié par une arête à un sommet de degré $n + 1$ de la deuxième copie.

AB_2 est montré sur la figure suivante :



FIGURE 4.5.

On peut parler d'un autre plongement, concernant ce type d'arbre, qui nécessite de trouver combien d'arbre de même topologie qu'on puisse plonger dans un hypercube de dimension donnée.

Proposition 2. *Le nombre maximum de copies de AB_1 , qu'on puisse placer dans un hypercube de dimension n , est 2^{n-2} .*

Pour la démonstration, il suffit d'utiliser la récurrence sur n .

Références

1. Arfati, J. Papadimitriou, C.H. and Papageorgiou, P. : The complexity of cubical graphs. proceedings of 11 th international Kolloquium on automata , languages and programming. (1984) 51-57.
2. Berrachedi, A. : Sur la dimension cubique de quelques classes d'arbres. Actes du Colloque Cosi'04, Colloque sur L'optimisation et les Systèmes d'Information. Université de Tizi-Ouzou.
3. Bezrukov, S. and Monien, B. Unger, W. and Wechsung, G. : Embedding ladders and caterpillars into hypercube. discrete applied mathematics , **83** (1992) 21-29.
4. Corneil, D.G. and Wagner, A. : Embedding trees in a hypercube is NP- complet. siam j. comput **19** (1990),570-590
5. Havel, I. : On hamiltonian circuits and spanning trees of hypercubes. Cas prest. Mat **109** (1984) 135-152.
6. Havel, I. and Liebel, P. : One legged caterpillars spans hypercubes. Journal of graph theory. **10** (1986) 69-77
7. Havel, I. and Moravek, J. : B -valuation of graphs . Czech- Math .jour ., **22** (1972),338-351.
8. Firsov, V. : On isometric embeddings of graph into a boolean cube. cyber - netics 1, (1965) 112-113.
9. Harary, F. Lewinter, M. and Widolski, W. : On two legged caterpillars which span a hypercube. Congr. Numer. **66** (1988) 103-108.
10. Kabyl, k. : Dimension cubique de deux nouvelles classes d'arbres. Actes du Colloque Cosi'05, Colloque sur L'optimisation et les Systèmes d'Information. Université de Béjaia.
11. kobeissi, M. and Mollard, M . : Spanning graphs of hypercubes starlike and double starlike trees. Accepté à discrete Math.
12. Labord, J.M. and Rao hebbbar, S.P. : Another characterization of hypercube . discrete Math., **39** , (1982) 161-166.
13. Nebesky, L. : Embedding m -quasistars into n -cubes. C zechoslovak mathematical, journal, praha,38 (113),1988.
14. Nekri, M. and Berrachedi, A. : Two new classes of Trees Embeddable into hypercubes. RAIRO Oper. Res., **38**, (2004) 295-303.

Performance and Optimization of P2P Networks for IP Multimedia applications

Mourad AMAD et Djamil AÏSSANI

Laboratoire de Modélisation et d'Optimisation des Systèmes (LAMOS)

Université de Béjaïa, Béjaïa 06000, Algérie

Tél. (213) 34 21 51 88

Résumé In this paper, we propose a novel P2P architecture for improving dependability, adaptability, and scalability, whereas maintaining low complexity and rapid convergence for lookup algorithm. The new architecture is termed HPM, because it is organized as a set of hierarchical rings, which connect neighboring nodes in terms of both physical and logical locations, facilitating inter and intra routing mechanisms. In particular, we show that lookup cost of HPM is $O(\sum_{i=1}^4 \log_2(n_i))$, where n_i represents the number of nodes on ring level i .

Key Words : P2P, Routing Optimization, Physical Proximity

5.1 Introduction

P2P computing refers to a class of systems that employ distributed resources to perform critical functions, such as resources localization, in a decentralized manner. According to different design goals, a number of system variants has been seen in P2P community. For example, P2P system proposed in [2] allows file sharing or computer resources and services by direct exchange between systems, or, P2P is defined as a class of applications that takes advantage of resources-storage, cycle, content, human presence-available in the Internet.

In this paper, we propose a new P2P architecture for resources discovery called *HPM*. HPM can be classified as a structured P2P system. The nodes in HPM architecture are organized as a multi-levels hierarchical set of rings, closed in terms of physical and logical proximities. Each level is composed of several rings. A ring is composed of at most 256 nodes. A node can belong simultaneously to two rings. One of the main benefits of our proposed architecture is the rapid convergence of the lookup process optimizing the cost lookup, while providing an efficient mechanism for fault tolerance and scalability. The routing table is also better optimized compared to the main P2P architecture.

5.2 Related Work

The process of routing in P2P networks operates at application layer. The overlay P2P network may lead to routing inefficiency, as opposed to routing service provided by the

transport layer (IP). Our proposed HPM routing objectives are : minimizing the number of hops and delay, locating nodes that store data or resources in purely decentralized P2P networks, but also, controlling the overhead while considering the physical proximity of nodes. In this context, HPM is considered as a scalable P2P protocol that optimizes resources discovery and localization function in a decentralized manner. It is based on cryptographic hash function for resource identifier, IP addresses and port number for node identifier. HPM architecture belongs to the third generation of P2P systems, which is based on specific topology that consider physical proximity. The following section describes and analyses the HPM architecture.

5.3 HPM : Concept, Principle and Architecture

HPM is organized as a set of hierarchical rings (*see figure 5.1*). Each layer or level i is composed of 2^{i-1} rings that connect neighboring nodes. In HPM, each node n is identified by a unique identifier, which is the i^{th} part of its IP address divided on four equal parts. i represents the level to which the node n belongs. The resources are also identified by a unique identifier. Each resource key is also composed of four parts (a.b.c.d). Each ring has 256 (2^8) nodes. The first level is composed of one ring and contains the nodes with IP addresses that are different in the first part, with no restriction on the other parts. From the example illustrated on figure 5.1, a first node n with IP addresses : 176.x.y.z, belongs to level 1, and gets identifier N176, while the other nodes with IP addresses, such as 176.a.b.c do not belong to the same ring (*level 1*) as node n . Each level connects a maximum of 256^{i-1} rings. One of the main HPM characteristics is that nodes with the network IP address such as : 50.31.60.**123**, 50.31.60.**125** belong to the same ring on the same level (*level 4*). In this way, the physical and logical proximities are somehow taken into account. On each ring of each level, nodes are organized and ordered increasingly based on their identifiers.

Each resource with an "a.b.c.d" type identifier, will be placed and located at the node with IP address w.x.y.z, where w (*respectively x,y,z*) is the lowest value greater or equal to a (*respectively b,c,d*). In figure 5.1, data with key K50.31.240.252 is placed on node N254, with IP address 50.31.240.254 on level 4, because 50 (*respectively 31, 240 and 252*) is the lowest value greater or equal to 50 (*respectively 31, 240 and 254*).

5.3.1 Finger table in HPM

Each node n maintains a routing table of at most m entries, called the finger table. The i^{th} entry in the finger table of node n contains the identifier of the first node s , that

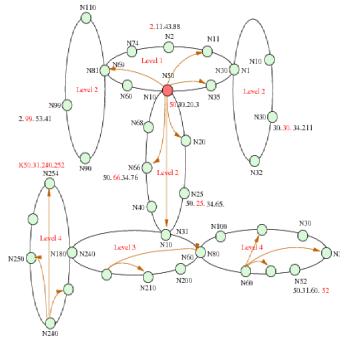


FIGURE 5.1. HPM architecture

succeeds n by at least 2^{i-1} on the identifiers circle. We call node s the i^{th} finger of node n . A finger table entry includes both HPM identifier and IP address of the relevant node. Figure 5.2 shows the finger table of a node with IP address **50.10.20.3**, belonging to two levels, and then has two identifiers : N50 on level 1 and N10 on level 2.

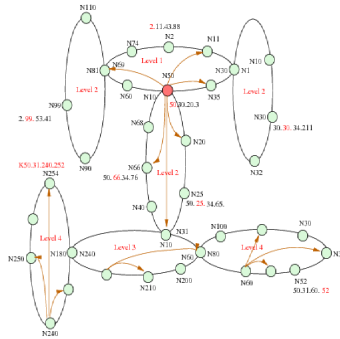


FIGURE 5.2. Finger table in HPM

5.3.2 Lookup process in HPM

For each ring k at level i , we use the i^{th} part of the data key for the lookup process, as in Chord. If the request succeeds on this ring k , the cost lookup is $O(\log_2(n_i))$. In case where the resource does not exist on the active covered ring, the search or localization is done on ring level $i + 1$ or $i - 1$, in a deterministic manner (*greedy routing*), then the lookup process cost is $O(\sum_{i=1}^4 \log_2(n_i))$, where n_i is the number of nodes on the covered ring at level i (*on which the request succeeded*). Algorithm 1 gives a pseudo code for the lookup process in HPM.

Algorithm 1 : Lookup data pseudo code (HPM) **Lookup (Key $c_1c_2c_3c_4$)**

```

1 : Begin
2 : Locate the node  $X$  (in the same ring) of IP Address  $p_1p_2p_3p_4$ , where  $p_i$  is the smallest value greater or equal to  $c_i$ ,  $\forall i \in [1..4]$ .
3 : If  $\exists c_j$  where  $c_j > p_j$  and  $j < i$  Then
4 : go to level  $(i - 1)$  and call lookup (Key  $c_1c_2c_3c_4$ )
      (if  $i > 1$ , otherwise, data does not exist)
5 : Else If the data is present Then
6 : Loading data from resulting node.
7 : Else
8 : go to level  $(i + 1)$  and call Lookup
      (Key  $c_1c_2c_3c_4$ ) (if  $i < 4$ , otherwise, data does not exist)
9 : End.

```

5.4 Conclusion and Future Work

In this paper, we have proposed a new approach aims to efficient resource discovery and location. Our proposed HPM architecture provides this discovery/localization service, based on a complete decentralized architecture, by determining with efficiency the node responsible for storing the requested key's value. The node identifier is built from one part of its IP address, while the resources identifier is generated by a hashing function from the resource name as key. In N-node HPM network, each node maintains routing information for only $2 * O(\log_2(n_i))$. HPM takes into consideration the physical topology and proximity. So, higher the level is, higher the physical nodes proximity is. In terms of perspectives, recent technique for application layer multicast [4] can be implemented using the proposed HPM for real time oriented applications.

Références

1. Detlef Schoder and kai Fischbach, The Peer-to-Peer Paradigm, in Proc of the 38th Hawaii International Conference on System Sciences, IEEE Internet computing, 2005.
2. Peer-to-Peer Working Group, Bidirectional Peer-to-Peer communication with interposing Firewalls and NATs, Peer-to-Peer Working group, White Paper, 2001.
3. Haiying Shena, Cheng-Zhong Xua and Guihai Chenb, Cycloid : A constant-degree and lookup-efficient P2P overlay network, Journal of performance evaluation, 63(3), Elsevier, Pages 195-216, 2006.
4. Mourad Amad and Ahmed Meddahi, A Scalable Approach for Application Layer Multicast in P2P Networks, in Proceedings of Sixth Annual IEEE International Conference, on Pervasive Computing and Communications, Hong Kong, 2008.
5. Hari Balakrishnan, Frans Kaashoek, David Karger, Robert Moris and Ion Stoica, looking up data in P2P systems, Com of the ACM, Vol 46(2), P43-P48, 2003.
6. Karl Aberer, Luc Onana Alima, Ali Ghodsi, Sarunas Girdzijauskas, Seif Haridi and Manfred Hauswirth, The essence of P2P : A reference architecture for overlay networks, in Proceedings the 5th IEEE International conference on Peer-to-Peer computing, 2005.

Optimisation de consommation d'énergie dans les réseaux de capteurs sans fil

Samra BOULFEKHAR

Laboratoire de Modélisation et d'Optimisation des Systèmes (LAMOS)
Université de Béjaïa, Béjaïa 06000, Algérie
Tél. (213) 34 21 51 88

Les réseaux de capteurs sans fil comportent un ensemble de petits dispositifs (appelés aussi capteurs intelligents) ayant des capacités de capture d'information, de calcul et de communications non filaires. La consommation énergétique de ce type de dispositifs est l'un des points critiques dans leur utilisation. En effet, les capteurs ont des ressources réduites et une capacité de transmission limitée. Des techniques optimisant la consommation d'énergie ont fait l'objet de plusieurs travaux de recherche. Ce type de contraintes corrélé avec le problème de déploiement d'un grand nombre de capteurs ont mis en évidence la nécessité d'utiliser des protocoles de routage adressant les problèmes de limitations liées à la consommation des ressources énergétiques [1].

Des topologies de type cluster [2, 3] ont été proposées afin de réduire la consommation d'énergie. Dans une architecture hiérarchique, des nœuds ayant plus d'énergie peuvent être utilisés pour traiter et envoyer des informations, tandis que des nœuds de plus faible énergie peuvent capturer les informations. Cette répartition de tâches conduit à une organisation virtuelle en clusters. Le routage hiérarchique pour ce type d'architecture fait notamment référence aux techniques de choix des cluster heads, de principe d'agrégation (fusion des informations pour réduire le nombre de messages transmis), plutôt qu'aux techniques de routage. Dans ce contexte, un protocole de routage hiérarchique basé sur le clustering optimisant la consommation d'énergie dans les RCSF, EEMCRP (Energy Efficient Multi-hop Clustering Routing Protocol for Wireless Sensor Networks) a été proposé. EEMCRP comprend trois phases, la phase de formation des clusters, phase d'élection des cluster-heads et phase de transmission des données entre les nœuds ordinaires et leurs CHs ainsi que le routage entre les CHs (CH en CH). Ce protocole vise à concrétiser plusieurs objectifs, parmi lesquels nous citons :

- Minimiser l'énergie consommée ainsi que l'équilibrage de charge.
- Eviter le problème des points chauds.

- Traiter le problème de la scalabilité ainsi que le cas des réseaux vastes.
- Eviter le problème des nœuds qui n'appartiennent à aucun cluster.
- Optimiser le nombre des nœuds ordinaires dans chaque cluster, et cela suivant deux paramètres : la surface du réseau et le nombre total des nœuds ordinaires.

Détail du protocole

Le protocole EEMCRP est constitué de plusieurs tours d'agglomération (clustering) successifs. Chaque tour comporte deux phase : phase d'élection des responsables de chaque cluster, et phase de communication des données. Par contre la phase de formation des clusters est fait une seule fois au début d'installation de réseau. Dans ce qui suit, nous donnerons le détail de chaque phase à part :

1. Formation des clusters

EEMCRP est un protocole auto-organisé basé sur la clusterisation statique où les clusters sont formé une fois pour toute. Chaque nœud ordinaire est affecté à un cluster par la station de base.

2. Election des clusters-head

Consiste à sélectionner les responsables de chaque cluster durant chaque tour, sachant que le temps est divisé en tours (une tranche de temps). Entre chaque tour et l'autre, on a une petite période de temps t_r (la durée de rafraîchissement).

3. Transmission des données

Comme le protocole basé sur la transmission multi-sauts, alors, on doit spécifier les deux cas de transmission : au milieu d'un cluster et entre les clusters.

Donc, dans cet exposé, on a présenté un nouveau protocole de routage hiérarchique statique à plusieurs sauts nommé EEMCRP. Notre protocole permet d'optimiser la consommation d'énergie des capteurs en distribuant la dissipation d'énergie sur l'ensemble des noeuds capteurs, en regroupant les noeuds adjacents dans le même cluster, en évitant le problème des points chauds ainsi que la scalabilité, en optimisant le nombre des nœuds ordinaires dans chaque cluster.

La suite de nos travaux de recherches sera orientée sur l'implémentation de EEMCRP, puis son évaluation et une comparaison de ses performances avec d'autres protocoles hiérarchiques [4, 5].

Références

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. l. Cayirci. "A survey on sensor networks". IEEE Communications, pp. 393- 422, 2002.
2. A. Abbasi, M. Younis "A survey on clustering algorithms for wireless sensor networks" Computer Communications 30 (2007), pp. 2826-2841,2007.
3. D. J. Dechene, A. El Jardali, M. Luccini, A. Sauer. "A survey of clustering algorithms for wireless sensor networks", Project Report 2006.
4. O. Younis , S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks : A Hybrid, Energy-Efficient Approach ", proceedings of the IEEE Infocom, Mars 2004.
5. W. Heizelman, J. Kulik, and H. Balakrishnan. "Adaptive protocols for information dissemination in wireless sensor networks". Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking,, 2001

Troisième partie

Processus Aléatoires et Applications

Comparabilité et monotonie : Méthodologie pour l'analyse du système $M/G/1$ avec rappels et feedback

Mohamed BOUALEM

Laboratoire de Modélisation et d'Optimisation des Systèmes (LAMOS)
Université de Béjaïa, Béjaïa 06000, Algérie
Tél. (213) 34 21 51 88

7.1 Introduction

La majorité des études sur les systèmes d'attente avec rappels considère le modèle sans feedback. Néanmoins, plusieurs situations réelles peuvent être modélisées comme des systèmes de files d'attente avec rappels et feedback. La notion de Feedback a été initialement introduite par Takacs (1963) pour l'étude de certains systèmes d'attente classiques, et depuis plusieurs papiers sont apparus sur ce sujet en considérant d'autres types de systèmes avec différentes variantes (rappels, vacances, pannes,...).

Le phénomène de feedback dans les systèmes d'attente avec rappels peut apparaître dans plusieurs situations pratiques [1, 2], par exemple, dans les systèmes MATS (Multiple Access Telecommunication Systems) où des messages s'avérant comme erreurs à la destination (messages perdus ou corrompus) sont renvoyés .

En raison de la complexité des modèles d'attente avec rappels, les résultats analytiques sont généralement difficiles à obtenir ou ne sont pas très exploitables du point de vue pratique. Pour résoudre le problème, il existe plusieurs méthodes numériques et d'approximation.

Le but de ce travail est d'appliquer les méthodes de comparaison stochastique pour l'étude du modèle $M/G/1$ avec rappels et feedback.

7.2 Comparaison stochastique

La méthode de comparaison stochastique est un outil mathématique utilisé pour l'étude des performances de certains systèmes modélisés par des chaînes de Markov.

L'idée générale de cette méthode est de borner un système complexe par un nouveau système, plus simple à résoudre et fournissant des bornes qualitatives pour ces mesures de

performances. Cette approche est basée sur la théorie des ordres stochastiques (**ordre** \leq_{st} , **ordre** \leq_v , **ordre** \leq_L). Pour plus de détails, voir [Boualem & Djellab & Aïssani (2009)[3], Muller & Stoyan (2002)[4], Aïssani & Taleb (2010)[5], Schantikumar (1994), Stoyan (1983), Boland & Proshan (1994), Gine et al.(2003) , Mokdad & Castel-Taleb (2008), Theodore & Christian (1999)].

7.3 Description mathématique

On considère un système de files d'attente à un seul serveur où les clients primaires arrivent suivant un flux poissonnien de taux λ . Un client qui arrive et trouve le serveur occupé, quitte l'aire du service pour rejoindre un groupe de clients bloqués appelé *orbite*. Après un certain temps aléatoire, il renouvelle sa tentative d'entrer en service, une fois, deux fois, ..., jusqu'à ce qu'il le trouve disponible. Une fois servi, le client doit décider, soit de rejoindre l'orbite pour un autre service avec une probabilité c ($0 \leq c \leq 1$) où de quitter le système définitivement avec une probabilité complémentaire \bar{c} . Les intervalles de temps inter-rappels suivent une distribution exponentielle de taux θ . Comme cette politique de rappel dépend du nombre de clients dans l'orbite, on l'appelle politique de rappel classique. Les temps de service sont supposés d'une loi arbitraire, de fonction de distribution $B(x)$, de transformée de laplace $\beta(s)$ et de moyenne finie β_1 . Toutes les variables aléatoires introduites sont mutuellement indépendantes.

L'état du système à l'instant t peut être décrit par le processus :

$$X(t) = (C(t), N_o(t), \xi(t), t \geq 0), \quad (7.1)$$

où,

$$C(t) = \begin{cases} 0, & \text{si le serveur est oisif;} \\ 1, & \text{si le serveur est occupé;} \end{cases} \quad (7.2)$$

$N_o(t)$: le nombre de client en orbite à l'instant t .

Si $C(t) = 1$, $\xi(t)$ représente le temps de service écoulé du client en service.

7.3.1 Chaîne de Markov induite

Soit $\{t_n, n \in \mathbb{N}\}$ une suite d'instant de la complétion d'un service.

La suite de variables aléatoire $Y_n = \{q_n = N(t_n^+), n \in \mathbb{N}\}$ forme une chaîne de Markov induite, dont l'équation fondamentale est

$$q_{n+1} = q_n - \delta_{q_n} + v_{n+1} + \eta, \quad (7.3)$$

où,

v_{n+1} : le nombre de clients qui arrivent pendant un temps de service qui se termine à l'instant t_{n+1} .

Sa distribution est donnée par :

$$K_i = P(v_{n+1} = i) = \int_0^\infty \frac{(\lambda x)^i}{i!} e^{-\lambda x} dB(x). \quad (7.4)$$

δ_{q_n} est la variable de Bernoulli définie par :

$$\delta_{q_n} = \begin{cases} 1, & \text{si le } (n+1)^{\text{ème}} \text{ client provient de l'orbite;} \\ 0, & \text{sinon.} \end{cases}$$

La distribution de δ_{q_n} est donnée par :

$$P(\delta_{q_n} = 1/q_n = k) = \frac{k\theta}{\lambda+k\theta} \text{ et } P(\delta_{q_n} = 0/q_n = k) = \frac{\lambda}{\lambda+k\theta}.$$

La variable aléatoire η est définie par

$$\eta = \begin{cases} 1, & \text{si le client servi décide de rejoindre l'orbite;} \\ 0, & \text{si le client servi décide de quitter le système.} \end{cases}$$

En outre, $P[\eta = 1] = c$ et $P[\eta = 0] = \bar{c} (= 1 - c)$.

Théorème 7.1 *La chaîne de Markov induite est ergodique si et seulement si*

$$\rho = \lambda\beta_1 + c < 1. \quad (7.5)$$

7.3.2 Inégalités préliminaires

On considère deux modèles d'attente $M/G/1$ avec rappels et feedback, de paramètres $\lambda^{(1)}, \theta^{(1)}, B^{(1)}(x)$ et $\lambda^{(2)}, \theta^{(2)}, B^{(2)}(x)$ respectivement. Notons par $\{k_n^{(1)}\}, \{k_n^{(2)}\}$ les probabilités du nombre de clients arrivant durant une période de service dans chaque système. Les lemmes suivants donnent les conditions, sur les paramètres des deux systèmes, sous lesquelles ces probabilités sont comparables au sens des ordres \leq_{st} , \leq_v et \leq_L

Lemme 7.1. Si $\lambda^{(1)} \leq \lambda^{(2)}$ et $B^{(1)} \leq_{st} B^{(2)}$ alors $\{k_n^{(1)}\} \leq_{st} \{k_n^{(2)}\}$,

Lemme 7.2. Si $\lambda^{(1)} \leq \lambda^{(2)}$ et $B^{(1)} \leq_v B^{(2)}$ alors $\{k_n^{(1)}\} \leq_v \{k_n^{(2)}\}$.

Lemme 7.3. Si $\lambda^{(1)} \leq \lambda^{(2)}$ et $B^{(1)} \leq_L B^{(2)}$ alors $\{k_n^{(1)}\} \leq_L \{k_n^{(2)}\}$.

7.4 Monotonie de la chaîne de Markov incluse

Pour chaque distribution $\alpha = (\alpha_n)_{n \geq 0}$, on associe à l'opérateur de transition \mathcal{T} de la chaîne de Markov induite une distribution $\mathcal{T}\alpha = \beta = (\beta_m)_{m \geq 0}$ telle que :

$$\beta_m = \sum_{n \geq 0} \alpha_n p_{nm}. \quad (7.6)$$

Les deux théorèmes suivants donnent la condition sous laquelle l'opérateur de transition \mathcal{T} est monotone par rapport aux ordres stochastique \leq_{st} et convexe \leq_v .

Théorème 7.2 *L'opérateur de transition \mathcal{T} est monotone, par rapport à l'ordre stochastique. C'est-à-dire, pour deux distributions quelconques $\alpha^{(1)}$ et $\alpha^{(2)}$, l'inégalité $\alpha^{(1)} \leq_{st} \alpha^{(2)}$ implique la suivante : $\mathcal{T}\alpha^{(1)} \leq_{st} \mathcal{T}\alpha^{(2)}$.*

Théorème 7.3 *L'opérateur \mathcal{T} est monotone, par rapport à \leq_v . i.e, pour deux distributions quelconques $\alpha^{(1)}$ et $\alpha^{(2)}$, l'inégalité $\alpha^{(1)} \leq_v \alpha^{(2)}$ implique la suivante : $\mathcal{T}\alpha^{(1)} \leq_v \mathcal{T}\alpha^{(2)}$.*

7.4.1 Comparabilité des opérateurs de transition

Dans cette section, on considère deux modèles d'attente $M/G/1$ avec rappels et feedback, de paramètres $\lambda^{(i)}$, $\theta^{(i)}$, $c^{(i)}$, $B^{(i)}$. Notons par $\mathcal{T}^{(i)}$ l'opérateur de transition associé à la chaîne de Markov incluse dans le $i^{\text{ième}}$ système, $i = 1, 2$.

Les deux théorèmes suivants donnent les conditions de comparabilité de ces opérateurs par rapport aux ordres partiels : \leq_{st} et \leq_v .

Théorème 7.4 *Si $\lambda^{(1)} \leq \lambda^{(2)}$, $\theta^{(1)} \geq \theta^{(2)}$, $c^{(1)} \leq c^{(2)}$, $B^{(1)} \leq_{st} B^{(2)}$ alors $\mathcal{T}^{(1)} \leq_{st} \mathcal{T}^{(2)}$, i.e. pour une distribution quelconque α , on a $\mathcal{T}^{(1)}\alpha \leq_{st} \mathcal{T}^{(2)}\alpha$.*

Théorème 7.5 *Si $\lambda^{(1)} \leq \lambda^{(2)}$, $\theta^{(1)} \geq \theta^{(2)}$, $c^{(1)} \leq c^{(2)}$, $B^{(1)} \leq_v B^{(2)}$ alors $\mathcal{T}^{(1)} \leq_v \mathcal{T}^{(2)}$, i.e. pour une distribution quelconque α , on a $\mathcal{T}^{(1)}\alpha \leq_v \mathcal{T}^{(2)}\alpha$.*

Théorème 7.6 *Si $\lambda^{(1)} \leq \lambda^{(2)}$, $\theta^{(1)} \geq \theta^{(2)}$, $c^{(1)} \leq c^{(2)}$, $B^{(1)} \leq_L B^{(2)}$ alors $\mathcal{T}^{(1)} \leq_L \mathcal{T}^{(2)}$.*

7.4.2 Inégalités stochastiques des distributions stationnaires

Soient $\pi_n^{(1)}$, $\pi_n^{(2)}$ les distributions stationnaires du nombre de clients dans chaque système. Les deux théorèmes suivants donnent les conditions de comparabilité des distributions stationnaires du nombre de clients, pour deux systèmes de files d'attente $M/G/1$ avec rappels et feedback, par rapport aux ordres stochastique et convexe.

Théorème 7.7 *Si $\lambda^{(1)} \leq \lambda^{(2)}$, $\theta^{(1)} \geq \theta^{(2)}$, $c^{(1)} \leq c^{(2)}$, $B^{(1)} \leq_s B^{(2)}$, alors $\{\pi_n^{(1)}\} \leq_s \{\pi_n^{(2)}\}$, où $\leq_s = \leq_{st}$ (ou \leq_v).*

Théorème 7.8 *Si dans le système M/G/1 avec rappels et feedback, la distribution de temps de service est NBUE, alors $\{\pi_n\} \leq_v \{\pi_n^*\}$ où $\{\pi_n^*\}$ est la distribution stationnaire du nombre de clients dans le système M/M/1 avec rappels et feedback.*

Cas particulier

Si $c = (1 - \bar{c}) = 0$, notre modèle devient le M/G/1 avec rappels classique sans feedback. En se basant sur le théorème.7.7, on peut énoncer les deux théorèmes suivants :

Théorème 7.9 *Pour un système M/G/1 avec rappels, la distribution π_n est minorée, par rapport à l'ordre convexe, par une distribution dont la fonction génératrice est donnée par :*

$$\pi_n^*(z) = (1 - \rho) \exp[\rho(z - 1)] \frac{1 - z}{\exp[\rho(z - 1)] - z} \times \exp \left[\frac{\lambda}{\theta} \int_1^z \frac{1 - \exp[\rho(t - 1)]}{\exp[\rho(t - 1)] - t} dt \right].$$

Théorème 7.10 *Si pour le modèle M/G/1 avec rappels, la distribution de temps de service est NBUE (resp. NWUE), alors la distribution stationnaire du nombre de clients dans ce système π_n est inférieure (resp. supérieure), par rapport à l'ordre convexe, à la distribution stationnaire, π_n^* , du nombre de clients dans le système M/M/1 avec rappels.*

Où :

$$\pi_n^* = \frac{(\rho/\theta)^n}{n!} \prod_{i=0}^n (\lambda + i\theta)(1 - \rho)^{\frac{\lambda}{\theta} + 1}. \quad (7.7)$$

7.5 Conclusion

Dans ce travail, on a trouvé des conditions pour lesquelles l'opérateur de transition de la chaîne de Markov induite du modèle M/G/1 avec rappels et feedback est monotone par rapport aux ordres stochastique et convexe.

On a montré aussi que la distribution stationnaire du nombre de clients dans un système M/G/1 avec rappels et feedback, est majorée par la distribution stationnaire du nombre de clients dans un système M/M/1 avec rappels et feedback, si la distribution des temps de service est NBUE.

Pour terminer, nous avons présenté quelques cas particuliers.

Références

1. J.R. Artalejo Accessible bibliography on retrial queues : Progress in 2000-2009 Mathematical and Computer Modelling **51**(9-10) (2010) 1071-1081.

2. J.R. Artalejo, A. Gomez-Corral Retrial queueing system : A computation approach Springer Edition, Berlin, 2008.
3. M. Boualem, N. Djellab , D. Aïssani Stochastic inequalities for $M/G/1$ retrial queues with vacations and constant retrial policy Mathematical and Computer Modelling **50** (2009) 207-212.
4. A. Müller, D. Stoyan Comparison methods for stochastic models and risk John Wiley and Sons, LTD, 2002.
5. S. Taleb, A. Aïssani Unreliable $M/G/1$ retrial queue : monotonicity and comparability Queueing Systems **64** (2010) 227-252.

Quatrième partie

Contrôle Statistique de la Qualité

Some statistical aspects to take into account when studying strong stability of stochastic models

Aicha BARECHE

Laboratoire de Modélisation et d'Optimisation des Systèmes (LAMOS)
Université de Béjaïa, Béjaïa 06000, Algérie
Tél. (213) 34 21 51 88

8.1 Introduction

In [1], we considered the use of kernel density involving some boundary correction techniques for the study of the strong stability of the $M/M/1$ system. In this work, we reconsider these techniques, review some other techniques, give some related fields of such applications, and make a reflexion in the strong stability sense.

8.2 Kernel density estimation

One of the most popular and widely studied class of nonparametric estimators of a density f is the so called kernel class of estimators. If X_1, \dots, X_n is a sample from a random variable X having the probability density function (pdf) f and a distribution function (cdf) F , the Parzen-Rosenblatt kernel estimate [13, 11] of the density $f(x)$ for each point $x \in \mathbb{R}$ is given by

$$f_n(x) = \frac{1}{nh_n} \sum_{j=1}^n K\left(\frac{x - X_j}{h_n}\right), \quad (8.1)$$

where K is a symmetric density function called kernel and h_n is the bandwidth.

8.3 Boundary bias correction

Classical symmetric kernels work well when estimating densities with non-bounded support. However, when these latter are defined on the positive real line $[0, \infty[$, without correction, the kernel estimates suffer from boundary effects since they have a boundary bias ($\mathbb{E}(f_n(0)) = \frac{1}{2}f(0) + o(h_n)$). In fact, using a fixed symmetric kernel is not appropriate for fitting densities with bounded supports as a weight is given outside the support.

Additionally, standard kernel methods yield wiggly estimation in the tail of the distribution (especially for heavy-tailed distributions) since the mitigation of the boundary bias leads to favor a small bandwidth which prevents pooling enough data.

Several techniques have been introduced to get a better estimation either on the border or in the tail. Some of them propose the use of particular kernels or bandwidths :

- Reflection method (mirror image modification) [14, 15] ;
- Boundary kernel method [12] ;
- Transformation method (transformed kernel) [10, 9] ;
- Pseudo-data method [5].

The main criticism addressed to these approaches is that a number of them are quite complicated (since implementation is a nontrivial exercise) and thus difficult to work with, both numerically and analytically. Also, they allow the corrected estimator to become negative. In the last decade, to remedy these problems, other techniques propose the use of estimators based on flexible kernels (asymmetric kernels and smoothed histograms). They are very simple in implementation, free of boundary bias, always non-negative, their support matches the support of the probability density function to be estimated, and their rate of convergence for the mean integrated squared error is $o(n^{-4/5})$. We can cite :

8.3.1 Asymmetric kernel estimators

A simple idea for avoiding boundary effects is using a flexible kernel, which never assigns a weight out of the support of the density function and which corrects automatically and implicitly the boundary effects. The first category of the flexible kernels consists of the asymmetric kernels [3, 4] defined by the form

$$\hat{f}_b(x) = \frac{1}{n} \sum_{i=1}^n K(x, b)(X_i), \quad (8.2)$$

where b is the bandwidth and K is the asymmetric kernel.

Gamma kernels

The asymmetric kernel K can be taken as a Gamma density function K_G with the parameters $(x/b + 1, b)$ [4] given by

$$K_G\left(\frac{x}{b} + 1, b\right)(t) = \frac{t^{x/b} e^{-t/b}}{b^{x/b+1} \Gamma(x/b + 1)}, \quad t \in [0, \infty[, \quad (8.3)$$

where $\Gamma(\cdot)$ is the Gamma function.

Beta kernels

The asymmetric kernel K can be taken as a beta density function K_{β_1} with the parameters $(x/b, (1-x)/b)$ [3] such that

$$K_{\beta_1}(b, x)(t) = \frac{t^{b-1}(1-t)^{x-1}}{B(b, x)}, \quad t \in [0, 1], \quad (8.4)$$

where $B(\cdot)$ is the beta function.

8.3.2 Smoothed histograms

The second category of the flexible kernels is constituted of smoothed histograms [6, 7] defined by the form

$$\hat{f}_k(x) = k \sum_{i=0}^{+\infty} \omega_{i,k} p_{ki}(x), \quad (8.5)$$

where the random weights $\omega_{i,k}$ are given by

$$\omega_{i,k} = F_n\left(\frac{i+1}{k}\right) - F_n\left(\frac{i}{k}\right), \quad (8.6)$$

where F_n is the empirical distribution function, k is the smoothing parameter and $p_{ki}(\cdot)$ can be taken as a Poisson distribution function with parameter kx ,

$$p_{ki}(x) = e^{-kx} \frac{(kx)^i}{i!}, \quad i = 0, 1, \dots \quad (8.7)$$

8.4 Heavy-tailed distributions

A class of distributions that is often used to capture the characteristics of highly-variable stochastic processes (i.e., more variable than the exponential distribution) is the class of heavy-tailed distributions. In the literature, different definitions of heavy-tailed like distributions exist.

Definition We refer to a distribution as heavy-tailed if its coefficient of variation is larger than the one of the exponential distribution.

Definition A distribution is heavy-tailed if its *complementary cumulative distribution* (CCDF), often referred to as the *tail*, $F^c(t) = 1 - F(t)$, where $F(t)$ is the CDF, decays slower than exponentially, i.e., there is some $\gamma > 0$ such that

$$\lim_{t \rightarrow \infty} \exp(\gamma t) F^c(t) \rightarrow \infty. \quad (8.8)$$

Definition A distribution has short tail if its CCDF $F^c(t)$, decays exponentially or faster, i.e., there is some $\gamma > 0$ such that

$$\lim_{t \rightarrow \infty} \exp(\gamma t) F^c(t) \rightarrow 0. \quad (8.9)$$

Definition A light-tailed distribution is one in which the extreme portion of the distribution (the part farthest away from the median) spreads out less far relative to the width of the center (middle 50) of the distribution than is the case for the normal distribution.

8.4.1 Kernel density estimation of heavy-tailed distributions

Kernel density estimation of heavy-tailed distributions has been studied by several authors. Some of them proposed estimators based on transformation of the original variable. [2] consider transformations based on the Champernowne generalized distribution. An alternative way to obtain a kernel density estimator for heavy-tailed distributions is using the variable kernel estimator, which consists in selecting a different bandwidth parameter depending on the point where the density is being estimated [8].

8.5 Estimating quantiles and related risk measures

Quantiles often seem to be the natural thing to estimate in many applications when the underlying distribution is heavy-tailed. Furthermore, estimates of quantiles of the loss distribution in actuarial science and financial risk management are a fundamental part of the business. In this context, loss distribution is a positive variable which describes the probability distribution of payment to the insured. Quantile (called extreme Value-At-Risk (VaR) from the actuarial point of view) represents the worst loss ever on a target horizon that cannot be exceeded with a given level of confidence.

8.5.1 Estimating quantiles

Let X be a non-negative random variable admitting a continuous probability density function (pdf) f (called loss distribution in actuarial science and financial risk management), and cumulative distribution function (cdf) F_X . Given a probability level p ($0 < p < 1$), the p th quantile of X based on a random sample X_1, \dots, X_n is defined as

$$Q(X, p) = \inf\{x : F_X(x) \geq p\} = F_X^{-1}(p). \quad (8.10)$$

Nonparametric estimation

For continuous distribution $q(\alpha) = F_X^{-1}(\alpha)$ (the quantile function is simply the inverse of the cumulative distribution function), thus, a natural idea would be to consider $\hat{q}(\alpha) = \hat{F}_X^{-1}(\alpha)$, for some nonparametric estimation of F_X .

The kernel based cumulative distribution function, based on sample X_1, \dots, X_n is

$$\hat{F}_n(x) = \frac{1}{nh} \sum_{i=1}^n \int_{-\infty}^x k\left(\frac{X_i - t}{h}\right) dt = \frac{1}{n} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right), \tag{8.11}$$

where $K(x) = \int_{-\infty}^x k(t)dt$, k being a kernel and h the bandwidth.

Smoothing nonparametric estimators

a) Implicit class Find a smooth estimator for F_X , and then find (numerically) the inverse. The α -quantile is defined as the solution of $F_X \circ q_X(\alpha) = \alpha$.

If \hat{F}_n denotes a continuous estimate of F , then a natural estimate for $q_X(\alpha)$ is $\hat{q}_X(\alpha)$ such that $\hat{F}_X \circ \hat{q}_X(\alpha) = \alpha$, obtained using, for example, Gauss-Newton algorithm.

b) Explicit class Consider a linear combination of order statistics.

$$Q_n(p) = \sum_{i=1}^n \left[\int_{\frac{(i-1)}{n}}^{\frac{i}{n}} k\left(\frac{t-p}{h}\right) dt \right] X_{(i)} = \sum_{i=1}^n \left[K\left(\frac{\frac{i}{n} - p}{h}\right) - K\left(\frac{\frac{i-1}{n} - p}{h}\right) \right] X_{(i)}. \tag{8.12}$$

The idea is to give more weight to order statistics $X_{(i)}$ such that i is closed to pn .

Transforming observations

Given a random variable X , if H is a strictly increasing function, then the p -quantile of $H(X)$ is equal to $H(Q(X, p))$.

Thus, an idea can be to transform initial observations X_1, \dots, X_n into a sample Y_1, \dots, Y_n where $Y_i = H(X_i)$ taking values in $[0, 1]$, and then to use a beta-kernel based estimator. If $H : \mathbb{R} \rightarrow [0, 1]$, then

$$Q_n(X, p) = H^{-1}(Q_n(Y, p)). \tag{8.13}$$

In theory, any transformation $H : \mathbb{R} \rightarrow [0, 1]$ should work. But [2] suggested to set $Y_i = H(X_i)$ where H is the Champernowne generalized distribution which is suitable when modeling insurance claims, i.e. positive variables.

8.5.2 Risk measures

A very general class of risk measures can be defined as follows

$$\mathcal{R}_g(X) = \int_0^1 F_X^{-1}(1-u)dg(u), \quad (8.14)$$

where g is a distortion function, i.e. increasing with $g(0) = 0$ and $g(1) = 1$.

8.6 Conclusion

In this paper, we presented some statistical techniques to take into consideration in the analysis of some stochastic processes. It will be interesting to apply these techniques in the study of strong stability of some stochastic models (especially when a law or its density function is unknown). For example in, queueing systems (like $G/M/1$ or $M/G/1$), reliability models and risk models, since they involve :

- computing a variation distance which depends on the unknown density to estimate ;
- computing a reliability function ($\bar{F} = 1 - F$) which depends on the cdf function F ;
- Computing a risk measure which depends on F^{-1} and often involve heavy-tailed distributions.

Références

1. A. Bareche, and D. Aïssani. Kernel density in the study of the strong stability of the $M/M/1$ queueing system. *Operations Research Letters*, 36 : 535–538, 2008.
2. T. Buch-Larsen, J.P. Nielsen, M. Guillen, and C. Bolance. Kernel density estimation for heavy-tailed distribution using the Champernowne transformation. *Statistics*, 6 :503–518, 2005.
3. S.X. Chen. Beta kernel estimators for density functions. *Computational Statistics and Data Analysis*, 31 :131–145, 1999.
4. S.X. Chen. Probability Density Function Estimation Using Gamma Kernels. *Ann. Inst. Statist. Math.*, 52 :471–480, 2000.
5. A. Cowling, and P. Hall. On pseudodata methods for removing boundary effects in kernel density estimation. *J. Roy. Statist. Soc. Ser.*, B 58 :551–563, 1996.
6. N. Gawronski, and U. Stadtmüller. On Density Estimation by Means of Poisson’s Distribution. *Scand. J. Statist.*, 7 :90–94, 1980.
7. N. Gawronski, and U. Stadtmüller. Smoothing histograms by means of lattice and continuous distributions. *Metrika*, 28 :155–164, 1981.
8. M.C. Jones. Variable kernel density estimation. *Australian Journal of Statistics*, 32 :361–371, 1990.
9. J.S. Marron, and D. Ruppert. Transformations to reduce boundary bias in kernel density estimation. *J. Roy. Statist. Soc. Ser.*, B 56 :653–671, 1994.
10. M.P. Wand, J.S. Marron and D. Ruppert. Transformations in density estimation (with dicussion). *J. Amer. Statist. Assoc.*, 86 :343–361, 1991.
11. E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33 :1065–1076, 1962.

12. J. Rice. Boundary modification for kernel regression. *Commun. Statist. Theory Meth.*, 13 :893–900, 1984.
13. M. Rosenblatt. Remarks on some non-parametric estimates of a density function. *Ann. Math. Statist.*, 27 :832–837, 1956.
14. E.F. Schuster. Incorporating support constraints into nonparametric estimation of densities. *Commun. Statist. Theory Meth.*, 14 :1123–1136, 1985.
15. B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

Cinquième partie

Index des Auteurs

Index

AISSANI Djamil, 23

AMAD Mourad, 23

ANZI Aicha, 3

BARACHE Fatiha, 7

BARECHE Aicha, 41

BERRACHEDI Abdelhafid, 19

BOUALEM Mohamed, 33

BOULFEKHAR Samra, 27

FAHEM Karima, 12

KABYL Kamal, 19

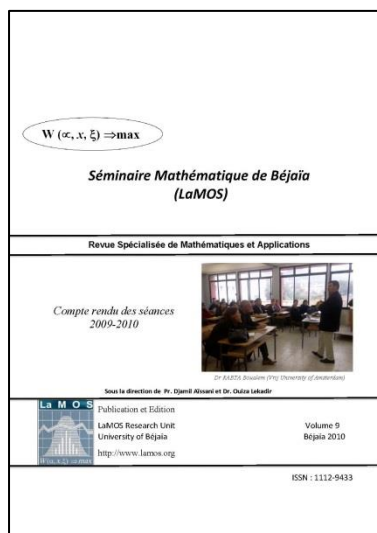
RADJEF Mohammed Said, 3, 12

Lamos Editions, 2009.

L'ouvrage :

Ce volume contient la plupart des exposés faits en 2009/2010 au Séminaire Mathématique de Béjaïa. Ils sont centrés autour des domaines d'intérêt des chercheurs du laboratoire de Recherche LAMOS (Modèles déterministes et stochastiques de la Recherche Opérationnelle, Probabilités et statistiques, Optimisation, ...) avec une attrayante dispersion vers des domaines lointains.

Séminaire Mathématique de Bejaia (LaMOS). ISSN : 1112 – 9433



Le premier exposé à Bejaia a été présenté en 1987 par Rachid Senoussi (alors enseignant à l'Université de Constantine), à l'époque du projet de création de l'Unité de Recherche en Statistiques Appliquées (1985). Le Séminaire a atteint son rythme de croisière en 1991/1992, après la création du Département des Sciences Exactes (22 exposés programmés). Depuis 1995, il est lié à la Post-Graduation "Modélisation Mathématique et Techniques de Décision" (cf. Bulletin de la S.M.A.I. Société Française de Mathématiques Appliquées et Industrielles N°49, 1997, pp. 50).

Parmi les conférenciers de l'année 2009 - 2010: Mourad Amad, Samra Boulfekhar, Aïcha Bareche, Aïcha Anzi, ...

Pour citer ce volume : Revue « *Séminaire Mathématique de Béjaïa (LaMOS)* », Volume 9, 2010, 60 pages. ISSN : 1112 – 9433.

ISSN : 1112 – 9433