

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

Séminaire Mathématique de Béjaïa (LaMOS)

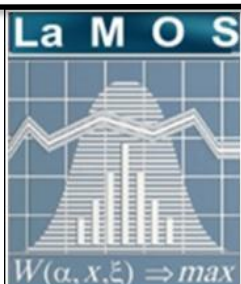
Revue Spécialisée de Mathématiques et Applications

*Compte rendu des séances
2008-2009*



Ger Koole, Vrij Universiteit d'Amsterdam

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 6
Béjaïa 2008**

ISSN : 1112-9433

SÉMINAIRE MATHÉMATIQUE DE BÉJAÏA

Compte rendu des séances 2008-2009.

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, 2009.

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

Table des matières

partie I Méthodes Cybernétiques et Optimisation

1 Résolution d'un problème de programmation bi-niveaux linéaire par la méthode DC	
<i>Aicha ANZI et Mohammed Said RADJEF</i>	3
2 Étude d'un Problème de Négociation sur le Marché à Terme	
<i>Salima KENDI, Fodil LAIB and Mohammed Said RADJEF</i>	9
3 Élément maximal d'une correspondance et ses applications en théorie des jeux	
<i>Karima FAHEM</i>	17

partie II Systèmes avec Rappel et Réseaux

4 Routage efficace en énergie pour les réseaux de capteurs	
<i>Samira Yessad</i>	23
5 Nouvelle technique de plongement de graphes dans l'hypercube	
<i>Kamel KABYL and Abdelhafid BERRACHEDI</i>	29
6 Intégration de mécanismes d'équilibrage de charge dans les réseaux de capteurs sans fil	
<i>Fouzi SEMCHEDINE</i>	33
7 Implémentation asynchrone améliorée d'un détecteur de défaillances	
<i>Nadjette REBOUH</i>	35

partie III Processus Aléatoires et Applications

8 On the numerical methods in Markov chain modeling
Karim ABBAS 41

9 Estimation du taux de rappels dans un système M/M/1 avec rappels
Louiza Berdjoudj 47

partie IV Contrôle Statistique de la Qualité

partie V Index des Auteurs

Index 55

Index 55

Première partie

Méthodes Cybernétiques et Optimisation

Résolution d'un problème de programmation bi-niveaux linéaire par la méthode DC

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. Un programme bi-niveaux linéaire (*PBL*) possède la forme suivante :

$$\left\{ \begin{array}{l} \max_x F(x, y) = c_1^t x + d_1^t y, \\ \text{s.c. } A_1 x + B_1 y \leq b_1, \\ x \geq 0; \\ \max_y f(x, y) = c_2^t x + d_2^t y, \\ \text{s.c. } A_2 x + B_2 y \leq b_2, \\ y \geq 0, \end{array} \right. \quad (1.1)$$

où $x, c_1, c_2 \in \mathbb{R}^{n_1}$; $y, d_1, d_2 \in \mathbb{R}^{n_2}$; $b_1 \in \mathbb{R}^{m_1}$; $b_2 \in \mathbb{R}^{m_2}$; $A_1 \in \mathbb{R}^{m_1 \times n_1}$; $B_1 \in \mathbb{R}^{m_1 \times n_2}$; $A_2 \in \mathbb{R}^{m_2 \times n_1}$ et $B_2 \in \mathbb{R}^{m_2 \times n_2}$.

Les définitions suivantes sont liées au problème (1.1) [4] :

a. *Ensemble des contraintes du problème*

$$S = \{(x, y) : A_1 x + B_1 y \leq b_1, A_2 x + B_2 y \leq b_2, x \geq 0, y \geq 0\}.$$

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

$$S(x) = \{y \in \mathbb{R}^{n_2} : B_2 y \leq b_2 - A_2 x, y \geq 0\}.$$

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

$$P(X) = \{x \in \mathbb{R}^{n_1} : \exists y \in \mathbb{R}^{n_2}, A_1 x + B_1 y \leq b_1, A_2 x + B_2 y \leq b_2, x \geq 0, y \geq 0\}.$$

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

$$R(x) = \{y \in \mathbb{R}^{n_2} : y = \arg \max[f(x, \hat{y}) : \hat{y} \in S(x)]\}.$$

e. Région induite

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

La région induite représente l'ensemble sur lequel le Leader optimise sa fonction objectif.

Bien que les fonctions objectif et les contraintes des deux niveaux sont linéaires, les problèmes de programmation bi-niveaux linéaires sont NP-difficiles et non convexes.

Dans ce travail nous traitons ces problèmes par une technique d'optimisation non convexe basée sur la programmation DC et l'algorithme DCA. Cette technique traite les problèmes d'optimisation où la fonction objectif peut être représentée sous la forme de différence de deux fonctions convexes

$$f(x) = g(x) - h(x)$$

où g et h sont appelées les composantes DC de f et $g - h$ la décomposition DC de f . Un programme DC s'écrit donc

$$\alpha = \inf\{f(x) = g(x) - h(x) : x \in X\}, \quad (1.2)$$

1.2 DCA pour la résolution de (PBL)

1.2.1 Reformulation via une pénalité exacte

Soit le problème (1.1). En remplaçant le problème du deuxième niveau par ses conditions d'optimalité de Karush-Kuhn-Tucker on obtient :

$$\left\{ \begin{array}{l} \max_{x,y} F(x, y) = c_1^t x + d_1^t y \\ A_1 x + B_1 y \leq b_1 \\ A_2 x + B_2 y + w = b_2 \\ B_2^t u - v = d_2 \\ v^t y + u^t w = 0 \\ x \geq 0, y \geq 0, u \geq 0, v \geq 0, w \geq 0, \end{array} \right. \quad (1.3)$$

En passant par les transformations suivantes

$$z = (x \ y \ e \ w \ v \ u)^t \in \mathbb{R}^n, \quad c = (-c_1 \ -d_1 \ 0 \ 0 \ 0 \ 0)^t \in \mathbb{R}^n,$$

$$A = \begin{pmatrix} A_1 & B_1 & I_{m_1} & 0 & 0 & 0 \\ A_2 & B_2 & 0 & I_{m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -I_{n_2} & B_2^t \end{pmatrix} \in R^{m \times n}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ d_2 \end{pmatrix} \in \mathbb{R}^m,$$

$$E_u = (0 \ 0 \ 0 \ 0 \ 0 \ I_{m_2}), \quad E_v = (0 \ 0 \ 0 \ 0 \ I_{n_2} \ 0),$$

$$E_w = (0 \ 0 \ 0 \ I_{m_2} \ 0 \ 0), \quad E_y = (0 \ I_{n_2} \ 0 \ 0 \ 0 \ 0),$$

où

I_k est une $k \times k$ matrice identité ; 0 est la matrice nulle avec la dimension appropriée pour chaque cas, avec $n = n_1 + 2n_2 + m_1 + 2m_2$; $m = m_1 + m_2 + n_2$.

En utilisant ces notations, nous aurons :

$$u^t w = (E_u z)^t (E_w z) = z^t (E_u^t E_w) z = z^t D^1 z, \text{ et } v^t y = (E_v z)^t (E_y z) = z^t (E_v^t E_y) z = z^t D^2 z,$$

ce qui donne : $u^t w + v^t y = z^t D^1 z + z^t D^2 z = z^t D z$ avec $D^1 + D^2 = D$.

Notons que les éléments $d_{ij} (i = \overline{1, n}, j = \overline{1, n})$ de la matrice D sont tous non négatifs. En posant $Dz = q(z)$, on obtient le problème :

$$\min \{ F(z) = c^t z, \quad Az = b, \quad z^t q(z) = 0, \quad z \geq 0 \}. \tag{1.4}$$

avec $q(z) \geq 0, \forall z \geq 0$.

Considérons l'ensemble convexe $\mathcal{Z} = \{z \in \mathbb{R}^n : Az = b, z \geq 0\}$, et soit la fonction $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ définie par $\Psi(z) = \sum_{i=1}^n \min\{q_i(z), z_i\}$.

Ψ est une fonction concave, finie et non négative sur \mathcal{Z} . Nous avons

$$\{z \in \mathcal{Z}, z^t q(z) = 0\} = \{z \in \mathcal{Z}, \Psi(z) \leq 0\}.$$

Le problème $P(z)$ peut être réécrit sous la forme

$$\alpha = \min\{F(z) : z \in \mathcal{Z}, \Psi(z) \leq 0\}. \tag{1.5}$$

D'après ([1], *théorème 1*), si \mathcal{Z} est non vide et borné, alors il existe une constante $k_0 \geq 0$, telle que pour tout $k \geq k_0$ le problème (1.5) est équivalent au problème pénalisé

$$\alpha(k) = \min\{F(z) + k\Psi(z) : z \in \mathcal{Z}\}. \tag{1.6}$$

1.2.2 Décomposition DC de (1.6)

Soit $\chi_{\mathcal{Z}}$ la fonction indicatrice de l'ensemble \mathcal{Z} . Soient g et h deux fonctions définies par

$$g(z) = \chi_{\mathcal{Z}}(z) \text{ et } h(z) = -F(z) - k\Psi(z). \tag{1.7}$$

Donc g et h sont convexes et le problème (1.6) est un programme DC de la forme

$$\min\{g(z) - h(z) : z \in \mathbb{R}^n\}. \tag{1.8}$$

L'application de DCA pour le problème (1.8) revient à calculer les deux séquences $\{t^i\}$ et $\{z^i\}$ définies par

$$t^i \in \partial h(z^i) \text{ et } z^{i+1} \in \partial g^*(t^i).$$

En utilisant les règle de calcul en analyse convexe, on obtient :

- $t^i = -c + k\theta^i$, avec

$$\theta^i = - \sum_{j=1}^n \begin{cases} D_j^t, & \text{si } z_j^i > D_j z^i, \\ e_j, & \text{si } z_j^i < D_j z^i, \\ \gamma e_j + (1 - \gamma) D_j^t, & \text{si } z_j^i = D_j z^i, \end{cases} \quad (1.9)$$

où D_j est la j -ème ligne de D , e_j est le j -ème vecteur unitaire de \mathbb{R}^n et $\gamma \in [0, 1]$.

- z^{i+1} est solution du problème

$$\min\{-\langle z, t^i \rangle : z \in \mathcal{Z}\}, \quad (1.10)$$

DCA pour (1.6)

- 1 : Soit z^0 point initial et $\epsilon > 0$. Poser $i = 0$, $k \in \mathbb{R}_+$, $\gamma \in [0, 1]$ et $\lambda > 0$.
- 2 : Calculer $t^i \in \partial h(z^i)$ via (1.9).
- 3 : Calculer $z^{i+1} \in \partial g^*(t^i)$ en résolvant (1.10).
- 4 : **Si** $y^{i+1} \in \arg \max \{f(x^{i+1}, y) : B_2 y \leq b_2 - A_2 x^{i+1}, y \geq 0\}$, **alors** aller à 5 ; **sinon** aller à 7.
- 5 : Calculer (v^*, u^*) , solution du problème $\min\{u^t(b_2 - A_2 x^{i+1}) : B_2^t u - v = d_2, u \geq 0, v \geq 0\}$, donc $z^{i+1} = (x^{i+1}, y^{i+1}, e^{i+1}, w^{i+1}, v^*, u^*)$.
- 6 : **Si** $\|z^{i+1} - z^i\| / (\|z^i\| + 1) \leq \epsilon$, **alors** stop z^{i+1} est solution optimale de (1.1) ; **sinon** aller à 7.
- 7 : Poser $z^i = z^{i+1}$, $i = i + 1$, $k = k + \lambda$ et aller à 2.

1.2.3 Remarques

i) Le problème (1.6), avec la décomposition (1.7), est un programme DC polyédral, puisque $g(z) = \chi_{\mathcal{Z}}(z)$ est polyédrale [7]. Donc DCA appliqué à (1.6) possède une convergence finie [2][3].

ii) Dans l'étape 4 on teste la faisabilité de la solution (x, y) pour le (PBL). La vérification de ce test implique que $y \in R(x)$. Et puisque $(x, y) \in S$, nous aurons alors $(x, y) \in RI$ ce qui implique que (x, y) est une solution réalisable du (PBL).

iii) L'algorithme DCA calcule la solution optimale $z^* = (x^*, y^*, e^*, w^*, v^*, u^*)$ du problème (1.6). Si on définit l'ensemble Z_r comme

$$Z_r = \{z = (x, y, w, e, v, u) \in \mathbb{R}^n : z \in \mathcal{Z}, (x, y) \in RI\},$$

alors, d'après *ii*), $z^* \in Z_r$. Une telle solution n'est pas forcément une solution du problème (1.1) à cause de la présence des composantes duales (v^*, u^*) . Le théorème suivant est inspiré de [5] et nous donne une condition pour qu'une solution z^* soit solution de (1.1).

Théorème 1.1 *Soit $k \in \mathbb{R}_+$ et soit l'ensemble $\bar{Z} \subset \mathcal{Z}$ défini par*

$$\bar{Z} = \{z \in \mathbb{R}^n : z \in Z_r, \exists I \subset \{1, \dots, n\} / z_i = 0, \forall i \in I; q_i(z) = 0, \forall i \notin I\},$$

Une solution optimale z^ du problème (1.6) est solution de (1.1), si $z^* \in \bar{Z}$.*

1.3 Résultats numériques

Pour évaluer les performances de notre algorithme nous l'avons implémenté sous MATLAB et testé sur un ensemble de 15 problèmes dont la solution optimale est connue. Le point initial a été calculé en résolvant le problème

$$0 = \min\{\Psi(z) : \bar{A}z = \bar{b}, z \geq 0\} \tag{1.11}$$

où

$$\bar{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ A_2 & B_2 & 0 & I_{m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -I_{n_2} & B_2^t \end{pmatrix} \quad \text{et} \quad \bar{b} = \begin{pmatrix} 0 \\ b_2 \\ d_2 \end{pmatrix}.$$

Nous avons pris $\gamma = 0.5$ et $\epsilon \leq 10^{-3}$.

Dans le but de faire une comparaison, nous avons implémenté l'algorithme développé par Y. Lv et al. [6] pour les problèmes de programmation bi-niveaux linéaire sans contraintes du Leader.

Les résultats numériques ont montré que notre algorithme est rapide, le temps d'exécution est petit et la procédure du calcul du point de départ est efficace car, avec un bon choix du paramètre de pénalité, il converge souvent vers la solution globale.

Concernant la comparaison avec l'autre algorithme, nous avons remarqué que les résultats sont comparables. Cependant, nous avons constaté quelques points de différence concernant essentiellement la sensibilité des deux algorithmes aux paramètres d'entrée.

1.4 Conclusion

Nous avons présenté un algorithme d'optimisation DC pour la résolution du problème de programmation bi-niveaux linéaire, où le problème du Suiveur est remplacé par les conditions d'optimalité KKT associées. Le problème est ensuite reformulé comme un programme

DC à l'aide d'une pénalité exacte et une décomposition DC adéquate. L'algorithme proposé est simple et rapide, puisque seuls des programmes linéaires sont résolus à chaque itération. Les résultats numériques montrent, qu'avec un bon choix du paramètre de pénalité, qui dépend du problème testé, l'algorithme converge souvent vers la solution globale.

Références

1. L.T.H. An and P.D. Tao : A continuous approach for globally solving linearly constrained quadratic zero-one programming problems. *Optimization*, 50 : 93–120, (2001)
2. L.T.H. An and P.D. Tao : Convex analysis approach to dc programming : theory and applications. *Acta Mathematica Vietnamica*, 22 : 289-355, (1997)
3. L.T.H. An and P.D. Tao : The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. *Annals Oper. Res.*, 133 :23–46, (2005)
4. J.F. Bard : Practical bilevel optimization : algorithms and applications. Kluwer academic publishers, Dordrecht (1998)
5. S. Dempe and A.G. Mersha. Linear bilevel programming with upper level constraints depending on the lower level solution. *App. Math. and Comp.*, 180 : 247-254 (2006)
6. Y. Lv, T. Hu, G. Wang and Z. Wan. A penalty function method based on Kuhn–Tucker condition for solving linear bilevel programming. *App. Math. and Comp.*, (2006)
7. R.T. Rockafellar : Convex analysis. Princeton, USA (1970)

Étude d'un Problème de Négociation sur le Marché à Terme

Salima KENDI, Fodil LAIB and Mohammed Said RADJEF

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

Université de Béjaïa, Béjaïa 06000, Algérie

Tél. (213) 34 21 51 88

Résumé L'étude du mécanisme des marchés à terme nécessite le recours aux outils de la modélisation dynamique des systèmes. Dans ce travail, nous proposons une approche de modélisation qui consiste à décomposer le système en deux réseaux de neurones récurrents. Le premier est désigné à générer les prix à terme et le second est désigné à générer les quantités à terme. L'apport de ce travail est la formation de prix à terme à travers les réseaux de neurones en se basant sur les variations de l'offre et de la demande et non sur l'hypothèse que l'histoire se répète en examinant les prix passés pour déterminer la direction du marché à terme comme c'est le cas de la plupart des méthodes d'analyse de marché.

Mots clés : Marchés à terme, Prix à Terme, Modélisation Dynamique, Réseaux de Neurones Récurrents, Apprentissage.

Introduction

La majorité des crashes financiers et des bulles spéculatives est due au comportement irrationnel de l'homme (comportement moutonnier, la panique, le désir du gain et la peur de perdre). Il est probable qu'en remplaçant l'homme par un automate dans la fixation des prix, il y'aurait moins de crashes financiers car les automates poursuivent des stratégies rationnelles tenant compte de la situation réelle du marché.

La capacité des réseaux de neurones de découvrir des rapports non-linéaires dans des données d'entrée les rend plus efficaces pour modéliser les systèmes dynamiques non-linéaires tels que la bourse [5]. Il y a une littérature étendue sur les applications financières des réseaux de neurones (Trippi & Turban, 1993 [3]; Refenes, 1994 [4]; ...).

2.1 Formulation mathématique du mécanisme du marché à terme

Soit $\mathcal{N} = \{1, \dots, n\}$ un ensemble de traders dans un marché à terme. Ces derniers estiment les niveaux de l'offre et de la demande à travers les informations provenant de différentes sources. Les traders envoient leurs ordres \mathbf{u}_i , $i \in \mathcal{N}$, sur la plate-forme

du marché. Les ordres de ventes sont mis dans la liste des ordres de vente : LSO¹ et ceux d'achat dans la liste des ordres d'achat : LBO². Le meilleur ordre de vente (celui du prix le plus bas) est toujours à la tête de LSO et le meilleur ordre d'achat (celui du prix le plus haut) est toujours à la tête de LBO. Supposons que les sessions quotidiennes du marché des contrats à terme, depuis le premier jour de transactions jusqu'au jour de l'expiration, se déroulent dans un intervalle de temps $[0, T]$ réparti en un ensemble de périodes $\mathbb{T} = \{t_0, \dots, t_m\}$, tels que

$$t_0 = 0, t_m = T, t_j = t_{j-1} + h, \quad j = 1, \dots, m,$$

où h est le pas de la discrétisation : $h = \frac{T}{m}$ [2].

2.1.1 Critères de performance économique

Pour mesurer l'efficacité des résultats générés par les réseaux proposés, les critères suivants sont utilisés [1] :

$$z_1(\gamma, \mathbf{S}, \mathbf{D}) = \frac{1}{m} \sum_{j=1}^m \mathbf{1}_{[\text{sign}(G(t_j) - G(t_{j-1})) = -\text{sign}(p(t_j) - p(t_{j-1}))]}, \quad (2.1)$$

$$z_2(\gamma, \mathbf{S}, \mathbf{D}) = \frac{1}{m-1} \sum_{j=1}^{m-1} \mathbf{1}_{[\text{sign}(G(t_{j+1}) - G(t_j)) = \text{sign}(p(t_j) - p(t_{j-1}))]}, \quad (2.2)$$

$$z_3(\gamma, \mathbf{S}, \mathbf{D}) = \frac{1}{m} \sum_{j=1}^m \mathbf{1}_{[\text{sign}(G(t_j)) = -\text{sign}(p(t_j) - p^*(t_{j-1}))]}, \quad (2.3)$$

$$z_4(\gamma, \mathbf{S}, \mathbf{D}) = \frac{1}{m-h} \sum_{k=1}^{m-h} \mathbf{1}_{[|\sigma_G(t_k, t_{k+h}) - \sigma_p(t_k, t_{k+h})| \leq \epsilon]}, \quad (2.4)$$

$$z_5(\gamma, \mathbf{S}, \mathbf{D}) = \frac{1}{m} \sum_{j=1}^m \mathbf{1}_{[q(t_j) > 0]}, \quad (2.5)$$

$$z_6(\gamma, \mathbf{S}, \mathbf{D}) = \frac{1}{m-h} \sum_{k=1}^{m-h} \mathbf{1}_{[\sigma_p(t_k, t_{k+h}) \leq \epsilon]}, \quad (2.6)$$

où, $G(t_j) = S(t_j) - D(t_j)$ représente la différence entre l'offre $S(t_j)$ et la demande $D(t_j)$ à l'instant t_j , $\mathbf{1}_{[\cdot]}$ est la fonction condition, $0 < \epsilon \ll 1$ et z_i est un ratio prenant ses valeurs dans $[0, 1]$ et mesurant l'efficacité des réseaux de neurones proposés selon l'hypothèse i

1. List of Selling Orders.

2. List of Buying Orders.

($i = 1, \dots, 6$).

La performance moyenne est

$$\bar{z}(\gamma, \mathbf{S}, \mathbf{D}) = \sum_{k=1}^6 w_k z_k, \quad (2.7)$$

où w_1, \dots, w_6 sont les poids associés aux critères ci-dessus, avec $0 \leq w_k \leq 1, k = 1, \dots, 6$

et $\sum_{k=1}^6 w_k = 1$.

2.1.2 Réseaux de neurones

Le modèle mathématique d'un neurone artificiel est composé d'un intégrateur effectuant la somme pondérée de ses R entrées. Le résultat n de cette somme est ensuite transformé par une fonction de transfert f qui produit la sortie a du neurone. Les R entrées du neurone correspondent au vecteur $p = [p_1 \ p_2 \ \dots \ p_R]^T$, alors que $w = [w_{1,1} \ w_{1,2} \ \dots \ w_{1,R}]^T$ représente le vecteur des poids ou "*poids synaptiques*". La sortie n de l'intégrateur est donnée par l'équation suivante :

$$n = \sum_{j=1}^R w_{1,j} p_j - b, \quad (2.8)$$

b est appelé "*biais*" ou "*seuil d'activation*" du neurone. Le résultat n s'appelle "*niveau d'activation*" du neurone.

Un réseau de neurones est un ensemble de neurones interconnectés suivant une topologie de connexion. Il existe deux types de réseaux de neurones : les réseaux *statiques* (ou non bouclés) et les réseaux *dynamiques* (récurrents ou bouclés). Le graphe de ces derniers contient au moins un cycle, d'où le terme bouclage.

2.2 Modélisation du mécanisme de la plate-forme du marché à terme par les réseaux de neurones

Dans ce travail, nous proposons une approche de modélisation et de résolution du phénomène de la génération des prix et des quantités à terme pour un producteur et un consommateur. Cette approche consiste à décomposer le système en deux réseaux de neurones récurrents capables de réagir aux variations de l'offre et de la demande dans la fixation des prix et des quantités à terme. Le premier réseau est désigné à générer les prix à terme et le second est désigné à générer les quantités³ à terme.

3. On note par quantité le nombre de contrats échangés lors d'une transaction donnée.

2.2.1 Présentation des réseaux générateurs des prix et des quantités à terme

Pour la génération des prix à terme, nous avons construit un réseau de neurones récurrent (RP) avec le toolbox "*Neural Network*" de MATLAB. Le réseau est constitué de quatre sous-réseaux MLP (SRp(1)-SRp(4)). Le réseau retourne les prix transactionnels en fonction des prévisions de l'offre et de la demande et du prix nominal.

Le réseau de neurones construit pour générer les quantités à terme (RQ) est un réseau bouclé constitué de cinq sous-réseaux MLP (SRq(1)-SRq(5)). Le réseau retourne les quantités transactionnelles en fonction des prévisions ajustées de l'offre et de la demande et des positions ouvertes du producteur et consommateur respectivement.

2.3 Génération des données d'apprentissage

Nous supposons que : le nombre de périodes $m = 100$, l'offre suit une loi normale de moyenne $\mu_S = 5000$ et d'écart type $\sigma_S = 0.5$ et la demande suit la même loi de moyenne $\mu_D = 5000$ et d'écart type $\sigma_D = 0.3$.

Dans notre travail, nous avons d'abord cherché des prix et des quantités de transactions (en fonction de l'offre et de la demande) maximisant la somme pondérée (2.7). Puis, nous les avons utilisés comme des sorties désirées pour les deux réseaux de neurones proposés.

L'algorithme d'apprentissage utilisé est le "*TRAINS*" (*Sequential order incremental training w/learning functions*), le critère d'erreur utilisé est l'erreur quadratique moyenne : "*MSE*" et le taux d'apprentissage choisi est celui par défaut : 0.1.

La courbe des prix générés par le réseau prix est représentée sur la figure ???. Nous constatons bien que les variations des prix sont inversement proportionnelles aux variations des écarts entre les prévisions de l'offre et celles de la demande.

2.4 Comparaison des sorties du réseau générateur de prix aux sorties désirées d'apprentissage

À la fin de l'apprentissage, nous avons obtenu le graphe de la figure 2.1.

Nous constatons que les deux courbes se rapprochent, l'erreur quadratique moyenne est égale à 0.0348.

Soient les notations suivantes :

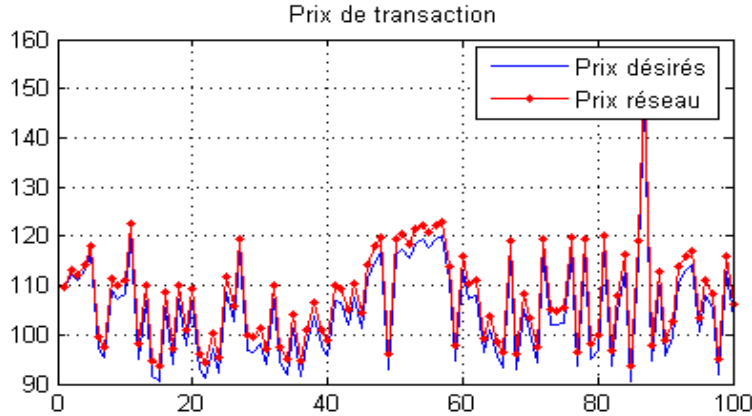


FIGURE 2.1. Sorties du réseau prix et sorties désirées

- Z_g : performances des résultats obtenus à travers les algorithmes génétiques en maximisant la somme pondérée (2.7) des critères de performances suggérés dans les formules (2.1)-(2.6).
- Z_{rg} : performances des résultats obtenus par les réseaux de neurones dont les sorties désirées sont celles obtenues à travers les algorithmes génétiques en maximisant la somme pondérée (2.7).

	z_1	z_2	z_3	z_4	z_5	z_6	\bar{z}
Z_g	0.9495	0.7245	1.0000	0	1.0101	1.0000	0.7807
Z_{rg}	0.9495	0.7245	0.8889	0	1.0101	1.0000	0.7622
Z_g	0.9192	0.7449	1.0000	0	1.0101	0.9888	0.7772
Z_{rg}	0.9192	0.7449	0.9394	0	1.0101	0.9888	0.7671
Z_g	0.8788	0.7449	1.0000	0	1.0101	1.0000	0.7723
Z_{rg}	0.8788	0.7449	1.0000	0	1.0101	1.0000	0.7723
Z_g	0.9394	0.7551	1.0000	0	1.0101	1.0000	0.7841
Z_{rg}	0.9394	0.7551	0.9596	0	1.0101	1.0000	0.7774

TABLE 2.1. Performances obtenues par les sorties des réseaux proposés et les sorties désirées

2.5 Analyse des résultats du tableau

Nous remarquons que les performances les plus élevées sont obtenues directement à travers les algorithmes génétiques en maximisant la somme pondérée (2.7). Mais ce qui nous intéresse est un modèle fixe capable d'apprendre le mécanisme du marché à terme, c'est-à-dire : le modèle d'un réseau de neurones. Nous constatons que les performances des

résultats des réseaux de neurones sont légèrement plus basses que les précédentes ; cela s'explique car l'apprentissage peut être bon mais pas parfait.

Conclusions

Après avoir réalisé notre étude et analysé les résultats, nous en tirons certaines conclusions :

le mécanisme de la plate-forme du marché à terme peut être modélisé par les réseaux de neurones, ce qui donne la possibilité d'avoir un modèle capable d'apprendre son comportement (détermination des prix et des quantités transactionnels) ; le choix du modèle de réseaux de neurones influe considérablement sur les performances du modèle et, par conséquent, sur les performances des résultats ; la méthode que nous avons adoptée en maximisant d'abord les critères de performances suggérés en utilisant les algorithmes génétiques a donné de bons résultats pour les modèles de réseaux de neurones.

Perspectives

Nous suggérons comme perspectives :

la détermination des prix et des quantités à terme à travers une méthode analytique, en appliquant les jeux dynamiques avec indétermination ; l'intégration des spéculateurs afin de refléter le fonctionnement réel d'un marché à terme ; l'architecture des réseaux de neurones que nous avons proposée reste une parmi une infinité d'architectures qui peuvent être proposées pour les mêmes réseaux ; afin de pouvoir utiliser cette approche dans des cas réels, nous devons considérer le cas général où il existe n traders, et chaque trader peut avoir sa propre stratégie de négociation. Ceci fait l'objet d'une thèse de doctorat en cours d'achèvement et d'un projet de recherche.

Références

1. M.S. RADJEF and F. LAIB. Optimal Strategies for Automated Traders in a Producer-Consumer Futures Market, Internal Report, LAMOS, May 08 2008.
2. M.S. RADJEF and F. LAIB. On the Mechanism of the Futures Market : a Formulation and Some Analytical Properties. To be presented at the 13th International Symposium on Dynamic Games and Applications, Wroclaw (Poland), 30th June-3rd July, 2008
3. R. TRIPPI & E. TURBAN. Neural Networks in Finance and Investment : Using Artificial Intelligence to Improve Real-world Performance. Chicago : Probus, 1993.
4. A. N. REFENES. Neural Networks in the Capital Markets. Chichester : Wiley, 1995.

5. R. LAWRENCE. Using neural networks to forecast stock market prices. Department of Computer Science, University of Manitoba, December 12 1997.
6. R. J. VAN EYDEN. The Application of Neural Networks in the Forecasting of Share Prices. Finance and Technology Publishing, 1996.
7. J. CORDIER. Les marchés à terme. Paris, 1984.

Élément maximal d'une correspondance et ses applications en théorie des jeux

Karima FAHEM

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

3.1 Introduction

Le progrès de la théorie des jeux et de l'équilibre économique va de paire avec celui de l'analyse non linéaire, chacun interagissant sur l'autre, les développements mathématiques trouvant des applications dans les sciences économiques, celle-ci motivant de nouvelles recherches, jetant de nouveaux défis aux mathématiciens.

Bien que le problème d'équilibre économique a été posé en 1874 par Léon Walras, il a fallu attendre la naissance de l'analyse non linéaire en 1910 avec le théorème de point fixe de Brouwer qui a permis à John Von Neumann dans les années vingt et trente de construire les fondations mathématiques de la théorie des jeux pour qu'ensuite suivent les travaux de Nash, Debreux Aumann, Shapley,...etc dans les années 50 pour démontrer l'existence des équilibres économiques.

En 1961, Ky Fan démontra un lemme qui donna naissance à un autre concept d'analyse non linéaire appelé élément maximal pour une correspondance. Développé et appliqué par Sonnenschein (1971), Shafer (1974),...pour établir l'existence d'un équilibre dans un jeu généralisé.

3.2 Notations et définitions

On suppose qu'une relation de préférence qu'elle soit faible ou stricte est définie sur un espace topologique Z .

On dénote par \succ une relation de préférence stricte.

et par

- $U_s(x) = \{y \in Z : y \succ x\}$ section supérieure stricte de x ,
- $L_s(x) = \{y \in Z : x \succ y\}$ section inférieure stricte de x .

Définition 3.1 Une relation binaire stricte \succ admet un **élément maximal** sur B s'il existe un élément $x^* \in B$ tel que

$$\nexists x \in B : x \succ x^* \quad \text{ce qui est équivalent à } B \cap U_s(x^*) = \emptyset.$$

Une relation binaire U (stricte ou faible) peut être représenté par une correspondance (une fonction multivoque) $U : Z \longrightarrow Z$ de la manière suivante $\forall x \in Z \quad U(x) = \{y \in Z : y \succeq x \text{ (ou } y \succ x)\} \subset Z$.

$$\begin{aligned} U(x) &= \{y \in Z : y \succeq x \text{ (ou } y \succ x)\} \text{ section supérieure de } x, \\ U^{-1}(x) &= \{y \in Z : x \in U(y)\} = \{y \in Z : x \succeq y \text{ (ou } x \succ y)\}, \text{ section inférieure de } x \\ Gr(U) &= \{(x, y) \in Z \times Z : y \in U(x)\} \text{ le graphe de } U. \end{aligned}$$

Définition 3.2 Une correspondance $C : Z \longrightarrow 2^Z$ admet un **élément maximal** s'il existe $x^* \in Z$ tel que $C(x^*) = \emptyset$.

3.3 Théorèmes d'existence

Soit $U : K \longrightarrow 2^K$ une correspondance représentant une relation de préférence.

Théorème 3.1 (Fan (1961)) Supposons que K est un sous ensemble compact et convexe de \mathbb{R}^m et C vérifie les conditions suivantes :

1. pour tout $x \in K$, $x \notin U(x)$,
2. pour tout $x \in K$, $U(x)$ est convexe,
3. le graphe de U est ouvert dans $K \times K$.

Alors l'ensemble des éléments maximaux est non vide et compact.

3.4 Élément maximal pour une famille de correspondances

Soit I un ensemble d'indices, pour tout $i \in I$ soit $U_i : Z \longrightarrow 2^{Z_i}$

Définition 3.3 $x^* \in Z$ est un **élément maximal** pour la famille $\{U_i\}_{i \in I}$ si

$$U_i(x^*) = \emptyset, \quad \forall i \in I.$$

Théorème 3.2 [Deguire et al (1999)] Soient $\{X_i\}_{i \in I}$ une famille de sous-ensembles non vides, convexes où tout X_i est dans un espace vectoriel topologique de Hausdorff E_i et I un ensemble quelconque d'indices. Soit $X = \prod_{i \in I} X_i$ et pour tout $i \in I$, soit $A_i : X \longrightarrow 2^{X_i}$, une correspondance telle que

- a) A_i est à valeurs convexes ;
- b) $y_i \notin A_i(y)$, pour tout $y \in X$;
- c) $A_i^{-1}(y_i) = \{x \in X : y_i \in A_i(x)\}$ est ouvert dans X , pour tout $y_i \in X_i$.
- d) $\bigcup_{i \in I} \{x \in X : A_i(x) \neq \phi\} = \bigcup_{i \in I} \text{int}\{x \in X : A_i(x) \neq \phi\}$.
- e) Supposons qu'il existe un sous-ensemble non vide compact K de X et des sous-ensembles non vides convexes et compacts C_i de X_i pour tout $i \in I$, tels que, pour tout $x \in X \setminus K$, il existe $i \in I$ tel que $A_i(x) \cap C_i \neq \phi$.

Alors il existe $x \in K$ tel que $A_i(x) = \phi$ pour tout $i \in I$.

3.5 Application en théorie des jeux

Considérons le jeu multicitère sous sa forme normale

$$G = (I, X_i, f_i) \tag{3.1}$$

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 $r(i) \in \mathbb{N}$.

Définition 3.4 (Shapley (1959)) une issue $x^* = (x_1^*, x_2^*, \dots, x_n^*) \in X$ est dite :

- Équilibre Slater-Nash pour le jeu (3.1), si pour tout joueur $i \in I$, il n'existe pas $x_i \in X_i$ tel que :

$$f_i(x_i, x_{-i}^*) > f_i(x_i^*, x_{-i}^*) \tag{3.2}$$

- Équilibre Pareto-Nash pour le jeu (3.1), si pour tout joueur $i \in I$, il n'existe pas $x_i \in X_i$ tel que :

$$f_i(x_i, x_{-i}^*) \geq f_i(x_i^*, x_{-i}^*). \tag{3.3}$$

Définition 3.5 (Voorneveld 2000) On dit que $x \in X$ est un équilibre idéal de Nash pour le jeu (3.1), si pour tout joueur $i \in I$ et tout $y_i \in X_i$,

$$f_i(x) = f_i(x_i, x_{-i}) \geq f_i(y_i, x_{-i}).$$

A l'instar du point fixe, on utilise l'élément maximal en théorie des jeux pour démontrer l'existence d'équilibres. On détermine une correspondance ou une famille de correspondances pour laquelle l'élément maximal est l'équilibre étudié. On verra dans ce qui suit deux exemples, il s'agit de l'équilibre Slater Nash et l'équilibre idéal.

3.5.1 Existence d'un équilibre Slater Nash

Pour tout $i \in I$, soit la correspondance $A_i : X \longrightarrow 2^{X_i}$, par

$$A_i(x) = \{y_i \in X_i : f_i(x) < f_i(y_i, x_{-i})\}, \quad (3.4)$$

Proposition 1. *Une issue $x^* \in X$ est un équilibre Pareto faible de Nash du jeu multicritère (3.1) si et seulement si $x^* \in X$ est un élément maximal de la famille de correspondances définies par (3.4).*

En utilisant un théorème 3.2, Ansari et al [1] ont établi des conditions suffisantes de son existence.

3.5.2 Existence d'un équilibre idéal de Nash

Définition 3.6 (Voorneveld (2000)) *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 :*

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

Soit Λ une collection représentative pour le jeu (3.1), alors d'après Voorneveld [3], on a $x \in X$ est un équilibre idéal de Nash si et seulement si pour tout $\lambda \in \Lambda$ et $i \in I$ on a $\langle \lambda_i, f_i(x) \rangle - \langle \lambda_i, f_i(y_i, x_{-i}) \rangle \geq 0 \quad \forall y_i \in X_i$. Soit pour tout $i \in I$, la correspondance $A_i : X \longrightarrow 2^{X_i}$, définie par

$$A_i(x) = \{y_i \in X_i : \exists \lambda \in \Lambda \text{ tel que } \langle \lambda_i, f_i(x) \rangle - \langle \lambda_i, f_i(y_i, x_{-i}) \rangle < 0\}. \quad (3.5)$$

Proposition 2. *Une décision $x^* \in X$ est un équilibre idéal de Nash du jeu multicritère (3.1) si et seulement si $x^* \in X$ est un élément maximal de la famille de correspondances définies par (3.5).*

En utilisant le théorème 3.2, Radjef et Fahem [2] ont établi des conditions suffisantes d'existence de cet équilibre.

Références

1. Ansari Q. H and Schaible S and Yao J. C, *The System of Generalized Vector Equilibrium Problems with Applications*, Journal of Global Optimization 22 (2002) 3-16.
2. Radjef M.S and Fahem K, *A note on ideal Nash equilibrium in multicriteria games*. Appl. Math. Lett. 2, 1105-1111 (2008).
3. 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

Routage efficace en énergie pour les réseaux de capteurs

Samira Yessad

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

Resumé de l'exposé

Un réseau de capteurs sans fil est constitué d'un grand nombre de capteurs interconnectés via un lien sans fil (radio, infrarouge, optique) ; chaque capteur a pour fonctions de capturer, de mémoriser, de traiter et de communiquer les données acquises vers une station de base. Ce type de réseau a beaucoup d'avantages qui ont fait d'eux des réseaux très utilisés dans des domaines différents, comme pour la surveillance de l'environnement, dans des champs de bataille, dans des océans, et plein d'autres domaines d'applications. Cependant, la faible capacité énergétique habituellement assignée à chaque nœud, et l'impossibilité d'une intervention humaine, a poussé les utilisateurs à s'intéresser à une propriété cruciale qui est la durée de vie minimale. Cette durée dépend fortement de pas mal de paramètres qui peuvent être considérés soit dans les protocoles MAC ou les protocoles de routage. Pour cela, plusieurs protocoles de routage, adressant les problèmes de limitations liées à la consommation des ressources énergétiques, ont été proposés dans la littérature. Ils sont, en général, basés sur la recherche d'une route multi-sauts optimale en considérant l'énergie résiduelle des nœuds intermédiaires.

En effet, un protocole de routage doit impérativement utiliser le minimum de ressources énergétiques et par la suite prolonger la vie du réseau pour éviter son partitionnement. Ainsi, l'idéal serait d'économiser l'énergie de tous les nœuds capteurs d'une manière équitable pour qu'ils fonctionnent tous ensemble pour une longue durée et puis tomber énergétiquement tous ensemble en même temps. Ceci ne peut pas être réalisé par les protocoles de routage à un chemin optimal, puisque les mêmes nœuds sont utilisés pour le routage ce qui va épuiser l'énergie de ces nœuds plus que d'autres.

Le but de notre exposé est de présenter un protocole de routage multi-chemin (EAR pour

Energy Aware Routing protocol) proposé par Rahul et al le critiquer et de présenter un nouveau protocole de routage multi-chemin pour les réseaux de capteurs (FEAR pour Fair Energy Aware Routing protocol), que nous avons proposé pour améliorer EAR.

Notre protocole repose sur deux idées. La première idée étant celle proposée par Rahul et al dans le protocole EAR, qui consiste à sauvegarder plusieurs chemins avec un coût minimal et non pas un seul chemin qui a le coût minimal. En plus, à chaque chemin est assigné une probabilité qui permet de choisir le chemin à utiliser à chaque envoi de données.

EAR contient trois phases,

- Une phase d’initialisation : dans laquelle les chemins sont découverts et les tables de routage sont établies en utilisant une inondation locale d’un message contenant une variable coût initialisée à 0 et mise à jour avec le coût d’énergie total calculé au niveau des nœuds. Ainsi les routes avec des coûts minimaux sont sauvegardées dans la table de routage et des probabilités leurs sont assignées.
- Une phase de communication des données : dans laquelle les nœuds sources de données et les nœuds intermédiaires choisissent aléatoirement un voisin de la table de routage pour router les données en utilisant les probabilités.
- Phase de maintenance des routes : l’inondation locale est effectuée périodiquement pour mettre à jour la table des routes

La deuxième idée, qui est notre nouvelle idée, consiste à ajouter un paramètre pour le calcul de la probabilité, il s’agit du nombre de routes optimales auxquelles le nœud voisin appartient.

Ainsi, Comme EAR, notre protocole comporte trois phases :

- Une phase d’initialisation : Un message contenant une variable coût initialisée à 0 est diffusé par la station de base puis chaque nœud recevant ce message le rediffuse à son tour vers ses voisins.

En recevant le message, Le nœud calcule la métrique d’énergie pour le nœud voisin qui a envoyé le message et l’additionne au coût total du chemin. Ainsi, si un nœud i envoie le message à un nœud j , ce dernier calcule la métrique d’énergie du nœud i (C_{ij}) en utilisant la formule suivante :

$$C_{ij} = e_{ij}^\alpha R_i^\beta$$

ou e_{ij} est l’énergie nécessaire pour l’envoi et la réception sur le lien entre le nœud i et le nœud j et R_i est l’énergie résiduelle du nœud i normalisé à son énergie initiale. Les

facteurs du poids α et β peuvent être choisis pour trouver le chemin qui consomme le minimum d'énergie ou le chemin qui contient les nœuds avec le maximum d'énergie résiduelle ou bien trouver le chemin qui combine les deux.

Une fois C_{ij} est calculé, le nœud j l'additionne avec la valeur de la variable coût ($cout_i$) envoyée par le nœud i pour calculer le coût du chemin vers la station de base à travers le voisin i ($cout_{ij}$) :

$$cout_{ij} = cout_i + C_{ij}$$

Les routes avec des coûts minimaux sont sauvegardées dans la table de routage puis un message (FTM pour Forwarding Table Message) contenant ces routes est diffusé localement par le nœud. Chaque nœud recevant le message FTM répond par un message (NFTM pour Number Forwarding Table Message) contenant le nombre de nœuds voisins qui compte ce nœud dans leurs table de routage.

Une fois un nœud reçoit ce dernier message, il peut assigner les probabilités de choix des routes en fonction du coût de la route et le nombre de nœuds qui utilisent cette même route, et ce en utilisant la formule suivante :

$$P_{ij} = \frac{\frac{1}{C_{ij} \times N}}{\sum_{k \in FT_j} \frac{1}{C_{jk} \times N}}$$

Cette dernière prendra en considération le coût du chemin comme dans EAR et le nombre de nœuds qui utilisent le même nœud.

- Une phase de communication des données : dans laquelle les nœuds sources de données et les nœuds intermédiaires choisissent aléatoirement un voisin de la table de routage pour router les données en utilisant les probabilités.
- Phase de maintenance des routes : l'inondation locale est effectuée périodiquement pour mettre à jour la table des routes

n'ayant pas eu encore les résultats de simulation pour comparer EAR et FEAR, nous présentons un exemple pour illustrer l'apport de FEAR par rapport à EAR pour réaliser plus d'équité entre les nœuds capteurs et ainsi maximiser la durée de vie du réseau.

Dans l'exemple simple de la figure 4.1 deux sources (S_1 et S_2) envoient des données à la station de base (SB). Supposons que les nœuds de cet exemple utilisent le protocole EAR pour le routage, le nœud S_1 peut envoyer ses données à travers les deux routes (D,E,F,SB) avec une probabilité égal à 0.5 et (A,B,C,SB) avec une probabilité égale à 0.5. Le nœud S_2 aussi peut envoyer ses données à travers deux routes, la route à travers les nœuds

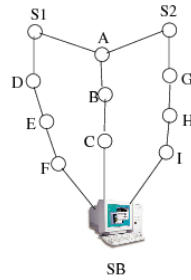


FIGURE 4.1. Exemple illustratif

(A,B,C,SB) qui est commune entre les deux nœuds S_1 et S_2 avec une probabilité égale à 0.5 et la route à travers les nœuds (G,H,I,SB) avec une probabilité égale à 0.5.

Supposons que les nœuds S_1 et S_2 vont envoyer six paquets de données chacun, alors chacun envoie trois de ces paquets par la route commune (à travers les nœuds A,B,C,SB) et les trois autres paquets par l'autre route.

En supposant que l'énergie initiale de tous les nœuds était la même, nous pouvons remarquer que malgré que le protocole EAR a pu réaliser une certaine équité entre les nœuds du réseau en utilisant plusieurs chemins de routage au lieu d'un seul, mais il y a la route commune entre les deux nœuds sources qui est utilisée plus que les deux autres (elle est utilisée pour six communications et les deux autres pour trois communications seulement). Ceci peut épuiser les nœuds de cette route plus que les autres.

Supposons maintenant que les nœuds de cet exemple utilisent le protocole FEAR pour le routage. Alors la probabilité des routes (D,E,F,SB) et (G,H,I,SB) devient $2/3$ et celle de la route commune devient $1/3$. Ainsi, avec ces probabilités, chacun des deux nœuds source va envoyer deux paquets à travers la route commune et les quatre restantes par l'autre route. Alors, avec FEAR on peut avoir une grande équité entre les nœuds capteurs et ainsi la durée de vie du réseau sera maximisée.

En guise de conclusion, nous avons essayé, à travers notre exposé, de présenter en général les réseaux de capteurs sans fil en mettant le point sur leur propriété cruciale qui est la durée de vie, et les solutions proposées au niveau des protocoles de routage. Parmi les protocoles de routage des réseaux de capteurs, nous avons présenté EAR, qui est un protocole multi-chemin proposé par Rahul et al pour maximiser la durée de vie des RCSFs que nous avons critiqué puis amélioré.

Alors, nous avons proposé un nouveau protocole de routage (FEAR), qui réduit la probabilité de l'utilisation d'un nœud qui est sollicité par plusieurs voisins dans le but de maximiser la durée de vie du réseau.

En guise de perspective, nous espérons comparer EAR et FEAR par simulation et prouver qu'avec FEAR le réseau peut fonctionner pour une longue durée.

Nouvelle technique de plongement de graphes dans l'hypercube

Kamel KABYL and Abdelhafid BERRACHEDI

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

Résumé L'hypercube étant une structure dont la topologie est utilisée en informatique (parallélisme, réseaux), il est fondamentale de déterminer quels sont les graphes et particulièrement les arbres qui sont plongeables dans l'hypercube c'est-à-dire qui sont sous-graphes de l'hypercube. Ce problème très étudié est toujours ouvert et on ne connaît que des résultats partiels pour certaines familles d'arbres. Dans ce travail nous avons introduit une application qui permet de déterminer la dimension minimale de certaines classes d'arbres.

Mots-clés : Hypercube, Plongement, Graphes, Arbres, Isomorphisme

5.1 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 . D'une manière générale, l'étude d'un plongement de graphe G dans l'hypercube revient à voir si G est isomorphe à un sous graphe de Q_n . Arfati, Papadimitriou et Papageorgiou [1] ont montré le résultat suivant : Le problème de décider si un graphe G est plongeable dans Q_n est $N.P$ -complet. Wagner et Corneil [2] ont montré que ce problème reste $N.P$ complet même dans le cas où G est un arbre. Un hypercube de dimension n , noté Q_n , est le graphe dont l'ensemble de sommets sont les n -uplets binaires et deux sommets sont adjacents si et seulement s'ils diffèrent en une seule coordonnée.

Un graphe biparti $G(X \cup Y; E)$ est dit équilibré si $Card(X) = Card(Y)$. L'hypercube Q_n est un graphe biparti équilibré, n -régulier ayant 2^n sommets et $n.2^{n-1}$ arêtes.

Comme conditions nécessaires de plongement de graphe dans Q_n on a : pour qu'un graphe G soit plongeable dans Q_n il faut que : $|V(G)| \leq 2^n$, G est biparti et le degré maximum de (G) , $\Delta(G) \leq n$. Si de plus $|V(G)| = 2^n$ alors G doit être équilibré. Toutes ces conditions sont nécessaires pour un graphe G plongeable dans Q_n , mais ne sont pas suffisantes.

La Cn -valuation 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 sont marquées par K . I. Havel et Moravek [5] ont montré qu'un graphe G est plongeable dans Q_n si et seulement s'il existe une Cn -valuation de G .

5.2 Nouvelle technique du plongement

5.2.1 Opération γ

Soit ζ l'ensemble des arbres binaires, on définit l'opération γ dans ζ de la manière suivante :

$\forall T_1, T_2 \in \zeta, \gamma(T_1, T_2) = T$ avec T est un arbre obtenu en reliant un sommet de T_1 à un sommet de T_2 par une arête.

Theorem 1. *Tout arbre T obtenu par l'application γ est de dimension cubique inférieure ou égale à $Max(dim(T_1), dim(T_2)) + 1$.*

Remark 1. Si on ne connaît pas $dim(T_1)$, et $dim(T_2)$ on fait de la même manière pour T_1 et T_2 c'est-à-dire : $T_1 = \gamma(T'_1, T'_2)$ avec $dim(T_1) \leq Max(dim(T'_1), dim(T'_2)) + 1$ même chose pour T_2 et ainsi de suite.

Démonstration. Supposons qu'on connaît $dim(T_1)$ et $dim(T_2)$ et que $Max(dim(T_1), dim(T_2)) = dim(T_1) = n$, donc $|T_1| \leq 2^{dim(T_1)}$ et que $|T_2| \leq 2^{dim(T_2)}$, alors si on marque l'arête qui joint T_1 à T_2 par $n+1$ on obtient une $C_n + 1$ valuation de T et comme $|T| \leq |T_1| + |T_2| \leq 2^n + 1$, d'où $dim(T) \leq n + 1$, dans le cas où on ne connaît pas $dim(T_1)$ et $dim(T_2)$ on utilise la remarque précédente.

5.2.2 Exemples

1. Considérons les arbres suivants :

a) L'arbre T_1 est donné par la figure suivante :

b) L'arbre T_2 est donné par la figure suivante :

Par l'opération γ , si on relie le sommet u au sommet v par une arête, on obtient l'arbre T de la figure suivante :

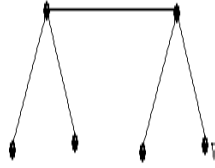


FIGURE 5.1.

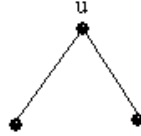


FIGURE 5.2.

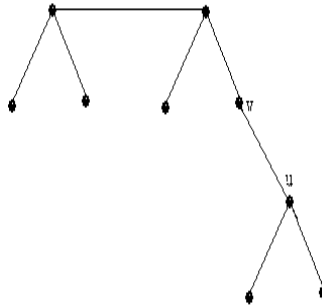


FIGURE 5.3.

Il est claire que $\dim(T_1)=3$ et $\dim(T_2)=2$, alors d'après le thérème précédant on trouve $\dim(T) \leq \max(3;2)+1$ c'est á dire $\dim(T) \leq 4$, et ce résultat on peut le verifier facilement sur le l'arbre T , en marquant l'arête reliant u à v par 4, et comme le nombre de sommet de T est égale à 9 qui est supérieur á 2^3 alors T ne peut pas être plongeable dans Q_3 , D'où $\text{Dim}(T)= 4$.

2. Prenons $T_1 = R_n$ et $T_2 = R_n$ et T l'arbre obtenu par l'application γ en reliant un sommet pendent de T_1 à un sommet pendent de T_2 par une arête. D'après le théorème précédent $\dim (T)$ est inférieure ou égale á $\max(\dim(T_1), (T_2))+1 = n+3$. comme l'arbre T a $|T_1|+|T_2| = |R_n|+|R_n| = 2^{n+2} + 2$ sommets donc ne peut pas être plongeable dans Q_{n+2} car le nombre de sommets de T est supérieur au nombre de sommet de Q_{n+2} d'où $\dim(T)= n+3$. Pour $n=2$ l'arbre T est montré dans la figure suivante :

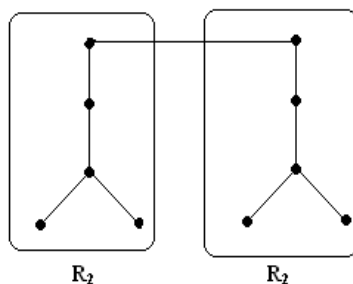


FIGURE 5.4.

5.3 Conclusion

Dans ce travail, nous nous sommes intéressés au plongement d'arbres dans l'hypercube. Beaucoup de chercheurs se sont intéressés à ce problème, leurs travaux ont permis de caractériser quelques classes d'arbres.

Tous les arbres sont plongables dans l'hypercube. Le problème consiste à trouver la plus petite dimension de l'hypercube dans lequel un arbre donné y est plongable, on parle alors d'hypercube optimal. Pour ce faire la notion de la C_n -valuation est utilisée. Ce problème a fait l'objet de plusieurs études ce qui a permis de trouver certains résultats pour quelques classes d'arbres.

Dans un deuxième temps, nous avons présenté une nouvelle technique pour déterminer les dimensions de certaines classes d'arbres.

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. Corneil, D.G. and Wagner, A. : Embeding trees in a hypercube is NP- complet. siam j. comput **19** (1990),570-590
3. Havel, I. : On hamiltonian circuits and spanning trees of hypercubes. Cas prest. Mat **109** (1984) 135-152.
4. Havel, I. and Liebel, P. : One legged caterpillars spans hypercubes. Journal of graph theory. **10** (1986) 69-77
5. Havel, I. and Moravek, J. : B-valuation of graphs . Czech- Math .jour ., **22** (1972),338-351.
6. Firsov, V. : On isometric embeddings of graph into a boolean cube. cyber - netics 1, (1965) 112-113.

Intégration de mécanismes d'équilibrage de charge dans les réseaux de capteurs sans fil

Fouzi SEMCHEDINE

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

Le développement des réseaux sans fil et des réseaux mobiles ouvre une nouvelle ère dans le domaine des télécommunications. La communication sans fil est devenue un des domaines les plus vibrants dans la mesure où elle offre des solutions ouvertes pour fournir des services essentiels là où l'installation d'infrastructure filaire n'est pas possible. Cependant, pour rendre cette communication plus attractive, flexible et avec moins d'infrastructures, une nouvelle génération de réseaux appelée réseaux de capteurs sans fil est apparue. Ces capteurs intègrent des dispositifs de captage et de communication sans fil dans un seul circuit, à dimension réduite, et avec un coût raisonnable. Un des problèmes majeurs dans ce type de réseaux est la consommation d'énergie. Lors d'une communication, l'émission et la réception d'un paquet sont des processus coûteux en énergie. Des protocoles de routage ont été proposés pour remédier à cet inconvénient.

Dans cet exposé, nous proposons une version améliorée du protocole DD appelée : DD-LB (Directed Diffusion with Load Balancing mechanism). Dans DD-LB, nous intégrons un mécanisme d'équilibrage de charge (utilisé pour l'affectation des tâches dans les systèmes de serveurs Web distribués) dans le protocole DD afin d'équilibrer la charge d'énergie des capteurs et augmenter, ainsi, la durée de vie du réseau. Les résultats de simulation de DD et DD-LB ont montré que DD-LB peut augmenter la durée de vie du réseau par rapport à DD, en équilibrant la charge d'énergie des capteurs.

Le protocole Directed Diffusion est considéré comme un protocole de référence dans le domaine de routage centré-données. Il suppose que les informations générées par les capteurs sont décrites par des paires (attribut, valeur). Par exemple, dans un champ de bataille, ces attributs servent à décrire le type de phénomène capté, sa position géographique, sa durée, etc. La station de base réclame les données requises par l'envoi d'un intérêt sous forme d'une requête basé également sur le schéma de paires (attribut, valeur) vers les capteurs concernés. Durant le processus de dissémination d'intérêt, des gradients

sont établis, afin de tracer les routes, pouvant être employées ultérieurement, pour la collecte des événements captés. L'une de ces routes est par la suite renforcée pour être utilisée. La dissémination de l'intérêt par la station de base à travers tous les capteurs du réseau provoque une consommation d'énergie inutile quand plusieurs capteurs, concernés par cet intérêt, génèrent les mêmes informations. De même, l'envoi du premier paquet, de la même information, par les capteurs vers la station de base consomme de l'énergie inutilement.

Directed Diffusion avec équilibrage de charge (DD-LB) est un protocole de routage des réseaux de capteurs sans fil que nous avons proposé et qui est une amélioration du protocole Directed Diffusion en lui introduisant la notion d'équilibrage de charge pour remédier à ses inconvénients. L'équilibrage de charge est une technique fiable et puissante qui a fait ses preuves dans plusieurs domaines, en particulier dans les systèmes de serveurs distribués. Dans ces systèmes, plusieurs serveurs homogènes sont installés pour répondre aux requêtes des clients. Un répartiteur, jouant le rôle d'un intermédiaire entre les clients et l'ensemble des serveurs, est mis en place afin de répartir la charge des requêtes sur les serveurs et améliorer ainsi les performances.

Le principe de DD-LB consiste à introduire cette technique dans le protocole DD afin de d'équilibrer la consommation d'énergie des capteurs. L'idée est de partager le réseau en zones où chaque zone regroupe les capteurs qui peuvent capter les mêmes informations. Dans chaque zone, un capteur est choisi, parmi les capteurs qui y appartiennent, comme répartiteur où son rôle est de choisir, selon une technique d'équilibrage de charge, le capteur qui va envoyer l'information à la station de base ; ce qui permet d'équilibrer la charge des capteurs de la zone et par conséquent celle du réseau. La technique d'équilibrage de charge base sur le paramètre d'énergie du capteur, d'où le capteur, qui a plus d'énergie, est choisi pour envoyer l'information. La dissémination de l'intérêt, dans le réseau, se fait à travers les capteurs répartiteurs.

Cette technique a comme avantage d'économiser l'énergie consommée lors de la dissémination à travers tous les nœuds du réseau. Lors de l'envoi de la donnée par le capteur source vers la station de base, cette dernière sera router par les capteurs répartiteurs. Les capteurs répartiteurs de chaque zone changent à chaque intervalle de temps ou à chaque nombre donné d'événement.

Implémentation asynchrone améliorée d'un détecteur de défaillances

Nadjette REBOUH

Laboratoire de Modélisation et d'Optimisation des Systèmes LAMOS
Université de Béjaïa 06000, Algérie.
email : nadjette_rebouh34@yahoo.fr

L'évolution des moyens de communication et de l'Internet conduit l'informatique répartie à s'imposer sur la scène technologique mondiale (commerce électronique, transactions financières, etc.). Cela est naturel car les organisations, les personnes et les données sont dispersées ; il est donc logique qu'un système informatique s'adapte à cette réalité en autorisant la distribution de l'état de l'application sur une collection de machines indépendantes reliées par un réseau de communication.

La maîtrise de cette distribution ne se fait malheureusement pas sans mal. Les matériels ne sont pas fiables, entraînant ainsi la défaillance de certaines entités. Au niveau du réseau, les équipements d'interconnexion ne le sont pas non plus et certaines liaisons peuvent se trouver surchargées, d'où une impossibilité de communication entre les entités ou pire, une perception erronée par une entité de l'état d'une autre. On constate donc que l'environnement d'une application répartie est hostile et pour le moins incertain.

Cependant un des importants avantages de la distribution est la tolérance aux fautes partielles qui peut être obtenue par la réplication des composants critiques du système. Bien que la réplication fournit plus de fiabilité et de disponibilité, elle peut conduire à une incohérence du système ainsi un problème de coordination entre les entités répliquées se pose. Ce dernier peut être résolu par l'utilisation des problèmes d'accord. Un tel accord est nécessaire pour garantir la cohérence du système.

Une bonne partie des problèmes d'accord peut s'exprimer à partir du problème du consensus. Ce problème exige que tous les sites corrects (c'est-à-dire, ceux qui ne subissent pas de défaillance) choisissent une valeur (terminaison). Cette valeur doit d'une part être la même pour tous les sites corrects (accord) et d'autres part avoir été proposée par au moins un processus participant au consensus (validité).

Le problème du consensus est plus au moins difficile à résoudre selon le modèle considéré. En dépit de sa simplicité dans un modèle synchrone en présence de défaillances, il se

complexifie dans des modèles partiellement synchrones [2],[1],[4] jusqu'à devenir impossible dans un modèle asynchrone. Cette impossibilité provient de l'impossibilité de distinguer de manière déterministe un processus très lent d'un autre défaillant dans un tel modèle [3].

L'affaiblissement du problème à résoudre et le renforcement du système asynchrone sont deux voies proposées pour contourner ce résultat négatif de Fisher, Lynch et Paterson [3]. La première approche, pour le problème du consensus, consiste en un affaiblissement d'une de ses propriétés. Par exemple affaiblir sa propriété de terminaison en une terminaison avec une probabilité égale à 1 [7]. L'autre approche consiste en l'ajout d'hypothèses de synchronie (les modèles partiellement synchrones introduits dans [2], [1], [4]) ou l'augmentation du modèle asynchrone par des oracles qui incorporent des hypothèses temporelles. Les détecteurs de défaillances non fiables introduits par Chandra et Toueg [4] sont définis formellement par deux propriétés abstraites de complétude et de précision.

Trois familles d'implémentation des détecteurs de défaillances non fiables ont été proposées dans la littérature, une première basée sur le temps physique, une deuxième qui n'utilise pas le temps physique et est basée sur une propriété comportementale sur le flux de messages échangés introduite par Mostefaoui et al. [5] et une dernière famille qui est une hybridation des deux premières [6].

Plusieurs travaux de recherche sur les performances de détecteurs de défaillances n'utilisant pas le temps physique ont plus porté sur une estimation fondée sur des calculs probabilistes que sur une simulation réelle. Ainsi, Mostefaoui et al. ont proposé une étude analytique pour calculer la probabilité de satisfaction de la propriété comportementale pour le cas d'un nombre maximum de processus pouvant subir une défaillance égale à 1.

Dans cet exposé, j'ai présenté une nouvelle implémentation asynchrone, en se basant sur les résultats de simulation du protocole de Mostefaoui et al. [5]. Ces résultats ont montré les limites de l'évaluation probabiliste déjà faite. Ainsi, ils ont visualisé l'inconvénient majeur de l'implémentation qui est le nombre élevé des fausses suspicions que le détecteur de défaillances peut commettre. J'ai présenté, aussi, une comparaison entre les deux implémentations asynchrones en termes du nombre de fausses suspicions.

Références

1. C. Dwork, N. Lynch and L. Stockmeyer : Consensus in the presence of partial synchrony. *Journal of the ACM*, vol. 35, n° 2, pages 288-323, avril 1988.
2. D. Dolev, C. Dwork and L. Stockmeyer : On the minimal synchronism needed for distributed consensus. *Journal of the ACM*, Vol. 34, No. 1, pages 77-97, January 1987.
3. F. Fisher, N. Lynch and M. Paterson : Impossibility of distributed consensus with one faulty process. *Journal de l'ACM*, pages 374-382, Avril 1985.

4. T. Chandra and S. Toueg : Unreliable failure detectors for reliable distributed systems. Journal de l'ACM, pages 245-267, 1996.
5. A. Mostefaoui, E. Mourgaya and M. Raynal : Asynchronous implementation of failure detectors. Proceedings of the 2003 International Conference on Dependable Systems and Networks(DSN'03), pages 351-360, 2003.
6. A. Mostefaoui, D. Powell and M. Raynal : A hybrid approach for building eventually accurate failure detectors. Proceedings of the 10th IEEE International Symposium (PRDC'2004), pages 57-65, March 2004.
7. M. Ben-Or : Another advantage of free choice :Completely asynchronous agreement protocols. 2nd ACM Symp. on Principles of Distributed Computing (PODC'83), pages 27-30, 1983.

Troisième partie

Processus Aléatoires et Applications

On the numerical methods in Markov chain modeling

Karim ABBAS

Laboratory of Modelisation and Optimization of Systems LAMOS
University of Béjaïa, Algeria.
E-mail : karabbas2003@yahoo.fr

8.1 Introduction

In the context of performance evaluation, numerical analysis methods refer to those methods which work with a Markov chain representation of the system under evaluation and use techniques from the domain of numerical analysis to compute stationary state probabilities or other measures of interest. The use of mathematical models to analyze complex systems has a long history. With the advent of high powered workstations and cheap memory, these applications have greatly expanded. More and more frequently, however, the characteristics of the system to be modeled are such that analytical solutions do not exist or are unknown so that systems engineers turn to computing numerical solutions rather than analytical solutions.

In this talk we would like to make a review of choosen numerical algorithms used for numerical solving of Markov chains.

We are interested in stationary solutions of homogeneous, irreducible Markov chain. Such a chain can be described with an infinitesimal generator matrix Q defined for continuous-time Markov chains (CTMCs) as following :

$$Q = (q_{ij})_{1 \leq i \leq n, 1 \leq j \leq n},$$
$$q_{ij} = \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(\Delta t)}{\Delta t} \text{ for } i \neq j,$$
$$q_{ii} = - \sum_{i \neq j} q_{ij},$$

where $p_{ij}(\Delta t)$ is the probability that if the chain is in the state i it will be in the state j after the time Δt .

For a discrete-time Markov chain (DTMC) we can define the matrix Q as $Q = P - I$, where P is a stochastic matrix of the DTMC and it is defined :

$$P = (p_{ij})_{1 \leq i \leq n, 1 \leq j \leq n},$$

where p_{ij} is the probability that if the chain is in the state i it will be in the state j in the next time moment.

Stationary solutions can be obtained (both for CTMCs and for DTMCs) from a linear system :

$$\pi Q = 0,$$

where $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ is a vector of probabilities of particular states of the Markov chain (so $\pi \geq 0$ and $\sum_{i=1}^n \pi_i = 1$) which are to be found.

The same probability vector can be obtained as an eigenvector of the stochastic matrix P from :

$$\pi P = \pi.$$

For CTMCs we can define P as :

$$P = I + \alpha Q, \tag{8.1}$$

where $0 < \alpha < 1 / (\max_{i=1, \dots, n} |q_{ii}|)$.

For a convenient notation we assume $\pi = x^t$ and our problem is to solve :

$$Q^t x = 0, \tag{8.2}$$

(or as an eigenvector problem : $P^t x = x$) with the constraints :

$$x \geq 0, e^t x = 1. \tag{8.3}$$

Despite its familiar form the equation is rather special. The matrix Q is singular so the equation (8.2) has solutions and it can be proven [?] that - if $\text{rank } Q = n - 1$ (it is true in interesting for us cases) - there exists exactly one solution satisfying (8.3). Moreover, the matrix Q is huge (sometimes millions of states or even more), very sparse and illconditioned. We have to choose a suitable algorithm for solving our problem (depending on our aims to achieve - accuracy, time or size).

There are following approaches to solve the equation (8.2) [?] :

- ⊇ direct methods ;
- ⊇ iterative methods ;
- ⊇ projection methods ;
- ⊇ decompositional methods.

8.2 Direct Methods

Methods which would give us an exact solution in a finite number of steps – if the machine accuracy were infinite – are called direct methods (or sometimes : exact methods). The traits of the direct methods are :

- ✓ constant execution time (or rather a constant number of computation steps for a given matrix size) known in advance ;
- ✓ modification (or rather complete reconstruction) of the given matrix ;
- ✓ the fill-in ;
- ✓ rather good accuracy.

The fill-in is a very troublesome phenomenon. It consists in appearing nonzero elements in the output matrices in places of zero elements in the input matrix. It is very unconvient if we want to store matrices in a compact manner (i.e. without zero entries) – what is very efficient and indeed necessary for such huge and sparse matrices. In compact storage schemes we must implement some routines to insert new nonzero elements – or provide some space for such entries. However, amount of this space must be estimated in advance what is not a trivial problem. Sometimes we simply have not the space needed for the fill-in.

8.3 Iterative Methods

All the iterative methods have the similar scheme. They start with a starting vector $x^{(0)}$ and then they generate a sequence $(x^{(0)}, x^{(1)}, \dots)$ which – hopefully – converges to the solution vector x .

The advantages of the iterative methods :

- ✓ they need no modification of the given matrix (so no fill-in is generated and we do not need any additional space for new elements and we spend no additional time on inserting these elements into a complicated storage structure) ;
- ✓ they need very little additional memory ;
- ✓ they are usually faster than direct methods – especially when we do not need very good accuracy offered by the direct methods ;
- ✓ they are easy to implement efficiently and easy to vectorize and to parallelize.

However, the iterative methods have some disadvantages too. We do not know the time needed to achieve required accuracy. Moreover, sometimes we can have even troubles with

convergency and we can achieve a solution not satisfying us – especially when the required accuracy is high (what can be an issue in our applications).

8.4 Projection Methods

The projection methods consists in approximating the solution vector with a vector from a small-dimension subspace. Such approximations are repeated until our approximation is sufficiently close to the solution – in some sense the projection methods are iterative methods.

The projection methods need more space than iterative methods (because they have to store huge basis vectors of subspaces), but can converge faster than classical iterative methods – although the convergence rate is much better for the matrices ‘more beautiful’ in their structure than the ones arising in solving Markov chains.

8.5 Decompositional Methods

In Markov chain models it is frequently the case that the state space can be meaningfully partitioned into subsets. Perhaps the states of a subset interact only infrequently with the states of other subsets, or perhaps the states possess some property that merits special consideration. In these cases it is possible to partition the matrix Q accordingly and to develop iterative methods based on this partition. In general such decompositional iterative methods require more computation per iteration, but this is offset by a faster rate of convergence.

The IAD - Iterative Aggregation / Disaggregation methods are related to decompositional iterative methods. They are particularly powerful when the Markov chain is NCD - Nearly Completely Decomposable, as the partitions are chosen based on how strongly the states of the Markov chain interact with one another, [?, ?]. The choice of good partitions for both decompositional iterative methods and IAD methods is an active area of current research.

8.6 Conclusion

In this talk we classified traditional methods for solving, $Q^t x = 0$, arising during modeling with Markov chains.

Selection of a suitable solution method is by no means easy. The choice depends on many questions :

- △ the matrix structure and its degree of decomposability (e.g. is it NCD ?) ;
- △ the matrix closeness to a suitable structure (and possibility to convert it) ;
- △ the matrix sparseness ;
- △ the matrix size (and our storage possibilities) ;
- △ time to find the solution ;
- △ desired accuracy ;
- △ the matrix conditioning.

Références

- [Cou77] P.J. Courtois. *Decomposability*. Academic Press, New York, 1977.
- [Mey89] C.D. Meyer. Stochastic complementation, uncoupling Markov chains and the theory of nearly reducible systems. *SIAM Rev.*, 31 :240–272, 1989.
- [Ste94] W. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Chichester, West Sussex, 1994.

Estimation du taux de rappels dans un système M/M/1 avec rappels

Louiza Berdjoudj

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

Résumé On considère l'estimation du taux de rappels dans un système M/M/1 avec rappels en utilisant les estimateurs intégrales. Le problème concerne la Variance de cet estimateur. Pour résoudre ce problème on introduit une nouvelle description markovienne de ce système.

Mots-clés : systèmes avec rappels, taux de rappels, statistique inférentielle.

9.1 Introduction

Les systèmes de files d'attente avec rappels ou avec répétition d'appels (Retrial Queues) sont caractérisés par la propriété suivante : un client arrivant dans le système et qui trouve tous les serveurs occupés, quitte le système définitivement, ou rappelle ultérieurement à des instants aléatoires. Un client qui attend pour rappeler est dit "en orbite". Plusieurs articles ont été publiés dans les années 70-80. Ces systèmes de files d'attente sont largement utilisés dans la modélisation des ordinateurs et les systèmes de télécommunications. Une description complète de situations où les systèmes de files d'attente avec rappels se présentent peut être trouvée dans les monographies de Falin et Templeton (1997) [6] et Artalejo et Gomez (1999) [3]. L'actualité du sujet est également confirmée par l'organisation périodique d'une conférence internationale sur les systèmes avec rappels (International Workshop on Retrial Queues). Mais il y a peu de travaux sur les problèmes statistiques pour les systèmes avec rappels ;

Les taux des arrivées et de service peuvent être estimés sans difficultés. Mais l'estimation du taux des rappels est un problème difficile. Ceci est lié aux circonstances suivantes :

1. On ne peut distinguer entre les arrivées primaires et secondaires
2. Les systèmes avec rappels ne peuvent être observés entièrement (en particulier, en pratique il est vraiment difficile d'observer les clients en orbite).

mais on peut toujours observer :

1. le flot joint des arrivées primaires et des appels répétés au service
2. Holding times.

En effet ceci est suffisant pour estimer le taux des rappels.

9.2 Description du modèle

On considère un système de files d'attente à un seul serveur où les clients primaires arrivent selon un processus poissonnien de taux λ . Un client primaire qui trouve le serveur libre est immédiatement servi et quitte le système après complétion du service. Mais si le serveur est occupé le client primaire entre en orbite et forme une source d'appels secondaires. Les inter-rappels sont exponentielles de taux θ , les temps de service ont des exponentiels de taux

μ . On suppose que les instants inter-arrivés, les instants de rappels et les durées de service sont mutuellement indépendantes.

On suppose que $\rho = \frac{\lambda}{\mu} < 1$ (condition du régime stationnaire du système) est vérifiée pour toute la suite. Soit $N(t)$ le nombre de clients dans l'orbite à l'instant t et

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

le processus $(C(t); N(t))$ est markovien d'espace d'états $\{0, 1\} \times N$ et de taux de transitions suivants :

1. Si $m = 0$ alors

$$q_{(m,n)}(i,j) = \begin{cases} \lambda, & \text{si } (i,j) = (1,n); \\ n\theta, & (i,j) = (1,n-1); \\ -(\lambda + n\theta), & (i,j) = (0,n); \\ 0, & \text{ailleurs.} \end{cases}$$

2. si $m = 1$ alors

$$q_{(m,n)}(i,j) = \begin{cases} \lambda, & \text{si } (i,j) = (1,n+1); \\ \mu, & (i,j) = (0,n); \\ -(\lambda + \mu), & (i,j) = (1,n); \\ 0, & \text{ailleurs.} \end{cases}$$

Le processus $(C(t); N(t))$ est le plus simple processus associé avec le système considéré.

Introduisons une nouvelle description markovienne de ce système. Soit $M(t)$ le nombre total des arrivées (primaires et secondaires depuis le dernier instant de départ. Ceci signifie que si $M(t) = 0$ alors le serveur est libre à l'instant t mais si $M(t) = m \geq 1$, alors le serveur est occupé à l'instant t et durant le temps de service écoulé exactement $m - 1$ clients ont rappelé pour le service ; Le processus $(M(t); N(t))$ est Markovien d'espace d'états N^2 . Les taux de transitions de ce processus sont comme suit :

1. Si $m = 0$ alors

$$q_{(m,n)}(i,j) = \begin{cases} \lambda, & \text{si } (i,j) = (1,n); \\ n\theta, & (i,j) = (1,n-1); \\ -(\lambda + n\theta), & (i,j) = (0,n); \\ 0, & \text{ailleurs.} \end{cases}$$

2. si $m \geq 1$ alors

$$q_{(m,n)}(i,j) = \begin{cases} \lambda, & \text{si } (i,j) = (m+1,n+1); \\ n\theta, & \text{si } (i,j) = (m+1,n); \\ \mu, & (i,j) = (0,n); \\ -(\lambda + \mu + n\theta), & (i,j) = (m,n); \\ 0, & \text{ailleurs.} \end{cases}$$

Notons que $C(t) = \delta(M(t))$, où $\delta(n)$ est une fonction indicatrice d'entiers positifs.

9.3 Caractéristiques stationnaires du système

Le processus $(C(t); N(t))$ est étudié en détail dans la théorie des files d'attente (voir Falin). En particulier, sa distribution stationnaire $p_{in} = P\{C(t) = i, N(t) = n\}$, est donnée par :

$$p_{0n} = \frac{\rho^n}{n! \theta^n} \prod_{i=0}^{n-1} (\lambda + i\theta) (1 - \rho)^{\frac{\lambda}{\theta} + 1} \quad (9.1)$$

$$p_{1n} = \frac{\rho^{n+1}}{n! \theta^n} \prod_{i=0}^n (\lambda + i\theta) (1 - \rho)^{\frac{\lambda}{\theta} + 1} \quad (9.2)$$

Les fonctions génératrices partielles correspondantes sont données par :

$$p_0(z) = \sum_{n=0}^{\infty} z^n p_{0n} = (1 - \rho) \left(\frac{1 - \rho}{1 - \rho z} \right)^{\frac{\lambda}{\theta}} \quad (9.3)$$

$$p_1(z) = \sum_{n=0}^{\infty} z^n p_{1n} = \rho \left(\frac{1-\rho}{1-\rho z} \right)^{\frac{\lambda}{\theta}+1} \quad (9.4)$$

La fonction génératrice du nombre de clients dans l'orbite est :

$$p(z) = p_0(z) + p_1(z) = (1 + \rho - \rho z) \left(\frac{1-\rho}{1-\rho z} \right)^{\frac{\lambda}{\theta}+1} \quad (9.5)$$

Ceci implique que les moments factoriels $\Phi_n = E(N(t))_n$ de la taille de la file sont donnés par :

$$\Phi_n = \frac{\rho^{n+1}}{(1-\rho)^n} \frac{(\lambda + \theta) \dots (\lambda + (n-1)\theta)(\mu + n\theta)}{\theta^n} \quad (9.6)$$

Le processus $(M(t), N(t))$ a été introduit pour la première fois par Falin (1997)

9.4 Estimation du taux de rappels

En pratique on ne peut pas observer le processus $N(t)$, mais on peut observer les processus $C(t)$ et $M(t)$. Ainsi on peut calculer les valeurs intégrales moyennes

$$\xi_T = \frac{1}{T} \int_0^T c(t) dt, \quad \eta_T = \frac{1}{T} \int_0^T M(t) dt.$$

Comme le processus $(M(t), N(t))$ est ergodique, avec une probabilité 1 il existe

$$\lim_{T \rightarrow \infty} \xi_T = E(C(t)) = \lambda,$$

$$\lim_{T \rightarrow \infty} \eta_T = E(M(t)) = \frac{\lambda}{1-\lambda} (1 + \lambda\theta),$$

et

$$\lim_{T \rightarrow \infty} \frac{\eta_T(1 - \xi_T) - \xi_T}{\xi_T^2} = \theta.$$

Ainsi, si l'intervalle d'observation $(0, T)$ est assez long on peut utiliser

$$\zeta_T = \frac{\eta_T(1 - \xi_T) - \xi_T}{\xi_T^2}$$

comme un estimateur du taux des rappels. Mais pour utiliser cet estimateur en pratique on doit connaître sa variance. Falin(1997) a considéré le comportement asymptotique de $\mathbf{Var}\zeta_T$ quand $T \rightarrow \infty$.

La variance de ζ_T , été calculé approximativement en utilisant la formule [] :

$$\mathbf{Var}f(\xi, \eta) \simeq \left(\frac{\partial f(x, y)}{\partial \xi} \right)^2 \mathbf{Var}\xi + 2 \frac{\partial f(x, y)}{\partial \xi} \frac{\partial f(x, y)}{\partial \eta} \mathbf{Cov}(\xi, \eta) + \left(\frac{\partial f(x, y)}{\partial \eta} \right)^2 \mathbf{Var}\eta \quad (9.7)$$

où $x = E(\xi)$, $y = E(\eta)$

Dans notre cas cette equation devient :

$$\mathbf{Var}\zeta_T \simeq \frac{(\lambda - (2 - \lambda)M)^2}{\lambda^6} \mathbf{Var}\xi_T + 2 \frac{(1 - \lambda)(\lambda - (2 - \lambda)M)}{\lambda^5} \frac{(1 - \lambda)^2}{\lambda^4} \mathbf{Var}\eta_T,$$

où $M = E(M(t))$.

Ainsi le problème est réduit au calcul de $\mathbf{Var}\xi_T$, $\mathbf{Var}\eta_T$ et $\mathbf{Cov}(\xi_T, \eta_T)$.

Références

1. Artalejo J.R. (1999) A classified bibliography of research on retrial queues :Progress in 1990-1999. In : *Top*, Vol. 7, Nr. 2. 187–211.
2. Artalejo J.R. (1996) Retrial queues with negative arrivals. In : *Proceedings of the International Conference on Stochastic Processes*. Cochín. 159–167.
3. Artalejo J.R., Gomez-corrall A. (1999) On a single server queue with negative arrivals and request repeated *J.Appl. Prob.*,vol. 36.907–918.
4. Baccelli F., Makowski A.M. (1985) Direct Martingale arguments for stability : the M/G/1 case. *Systems Control Letters*, Vol. 6, 181–186.
5. Baccelli F., Makowski A.M. (1989) Dynamic,transient and stationary behaviour of the M/GI/1 queue via martingals.*Annals of Probability*, Vol. 17, Nr. 4, 1691–1699.
6. Falin G.I.,Templeton J.G.C. (1997) *Retrial Queues*. Chapmanand Hall, London.
7. Gelenbe E. (1991) Production form queueing networks with negative and positive customers. *J. Appl.Probab*, Vol.28, 656–663
8. Neuveu J. (1972) *Martingales à temps discret*. Masson et Cie, Paris.
9. Roughan M. (1994) *An application of martingales to queieing theory*. PhD. thesis, Department of Applied Mathematics, University of Adelaide.

Quatrième partie

Contrôle Statistique de la Qualité

Cinquième partie

Index des Auteurs

Index

ABBAS Karim, 41
AINZI Aicha, 3

BERDJOUJ Louiza, 46
BERRACHEDI Abdelhafid, 28

FAHEM Karima, 16

KABYL Kamel, 28
KENDI Salima, 9

LAIB Fodil, 9

RADJEF Mohammed Said, 3
RADJF Mohammed Said, 9
REBOUH Nadjette, 35

SEMCHEDINE Fouzi, 33

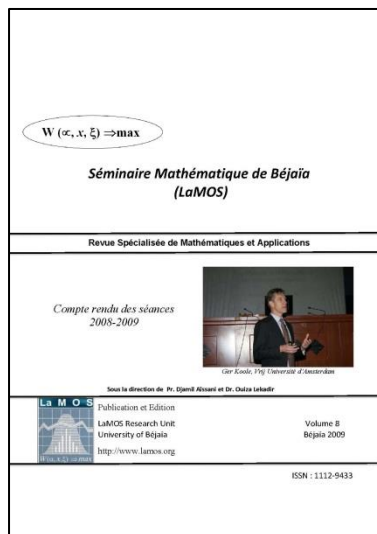
YESSAD Samira, 23

Lamos Editions, 2009.

L'ouvrage :

Ce volume contient la plupart des exposés faits en 2008/2009 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 2008 - 2009: Karima Fahem (Tizi Ouzou), Fodil Laib (Cévilal), Abdelhafid Benrachdi (USTHB), Fouzi Semchedine, ...

Pour citer ce volume : Revue « *Séminaire Mathématique de Béjaïa (LaMOS)* », Volume 6, 2008, 63 pages. ISSN : 1112 – 9433.

ISSN : 1112 – 9433