Estimation et prédiction en temps-réel de tables origine-destination

Michel Bierlaire

Technical report RO-040504
4 mai 2004

1 Introduction

Le problème d'estimation de tables[1] origine-destination (OD) à partir de données de comptages est de première importance pour un grand nombre d'applications impliquant la modélisation d'un système de transport. En effet, ces tables appréhendent statistiquement la demande, qui conditionne le fonctionnement de l'ensemble du système.

Ce papier à pour but de faire une synthèse des caractéristiques de ce problème, de le décrire sous plusieurs aspects (statique, dynamique, temps-réel) et de présenter des résultats récents liés à sa résolution et proposés par l'auteur.

2 Affectation

L'estimation de tables OD est le problème inverse du problème d'affectation de trafic. Nous commençons donc par formaliser celui-ci.

Le problème d'affectation statique du trafic (AST) consiste à identifier des flots sur les arcs d'un réseau à partir d'une table OD. Plusieurs ap-

[1] En pratique, elles sont souvent appelées *matrices origine-destination*. Dans ce papier, elles sont représentées mathématiquement par un vecteur de $\mathbb{R}^m$, et la dénomination de *matrice origine-destination* risque de générer des confusions.

proches ont été proposées dans la littérature, basées sur des hypothèses spécifiques liées au choix d'itinéraire et à l'impact de la congestion sur la demande. Nous référons le lecteur à Sheffi (1985) et Patriksson (1994) pour une description détaillée du problème AST.

Soit $G = (V, E)$ un graphe orienté représentant un réseau de transport. V est l'ensemble des sommets, et E est l'ensemble des n arcs. Soit $\Omega \subseteq V \times V$ l'ensemble des m paires origine-destination dans le réseau. Si $q \in \mathbb{R}^m$ représente les flots OD pour chaque paire OD dans Ω, le vecteur $v = A(q) \in \mathbb{R}^n$ représente les flots sur chaque arc de E.

La fonction d'affectation $A(q)$ peut être décomposée de la manière suivante. Pour tout $i \in \Omega$, soit $\mathcal{P}_i$ l'ensemble des p_i chemins reliant la paire OD i. Nous faisons ici l'hypothèse que le réseau est connexe et que $\mathcal{P}_i \neq \emptyset, \forall i$. Ainsi, $p = \sum_{i \in \Omega} p_i$ représente le nombre total de chemins dans le réseau. Soit la matrice $L \in \mathbb{R}^{n \times p}$ définie comme suit.

$$L_{k\ell} = \begin{cases} 1 & \text{si l'arc } k \text{ appartient au chemin } \ell \\ 0 & \text{sinon.} \end{cases} \tag{1}$$

Il s'agit de la matrice d'incidence arc-chemin, qui dépend uniquement de la topologie du réseau. Soit également la matrice de choix de route $C \in \mathbb{R}^{p \times m}$, où $C_{\ell i}$ est la proportion de déplacements de la paire OD i utilisant le chemin ℓ. Pour que C soit cohérente avec les modèles de choix de route (voir par exemple Ben-Akiva et Bierlaire, 1999 et Ben-Akiva et Bierlaire, 2003), nous supposons que

$$\sum_{\ell \in \mathcal{P}_i} C_{\ell i} = 1 \quad \forall i \in \Omega, \tag{2}$$

et

$$C_{\ell i} = 0 \text{ si } \ell \notin \mathcal{P}_i. \tag{3}$$

Si $\mathcal{I}(k)$ est l'unique paire OD reliée par le chemin k, il est clair que $C_{kj} = 0$ si $j \neq \mathcal{I}(k)$.

Nous pouvons dès lors définir la *matrice d'affectation* $A \in \mathbb{R}^{n \times m}$

$$A = LC, \tag{4}$$

et écrire le modèle d'affectation :

$$A(q) = Aq = LCq. \tag{5}$$

Notons que dans les réseaux congestionnés, la matrice C, et par conséquent la matrice A, dépendent des conditions de trafic. Malheureusement, le fait de tenir compte de cette dépendance dans le modèle complique fortement celui-ci, et il n'est pas rare de fixer C de manière exogène. Noter aussi que le nombre de chemins p est souvent très grand. Dès lors, les matrices L et C ne sont pas directement manipulables. La taille de la matrice A, quant à elle, ne dépend pas de p.

Le problème d'affectation dynamique du trafic (ADT) généralise le problème statique, en intégrant une dimension temporelle. Nous considérons donc un horizon temporel $[0, T]$ discrétisé en H intervalles de temps. Les matrices d'incidence, de choix de route et d'affectation sont généralisées comme suit. Soient deux intervalles de temps h_0 et h tels que h est postérieur à h_0 ($h \geq h_0$). La matrice $L_h^{h_0} \in \mathbb{R}^{n \times p}$ est telle que son élément (k, ℓ) vaut 1 si l'arc k est atteint atteint durant l'intervalle h en suivant le chemin ℓ, et en démarrant durant l'intervalle h_0, et 0 dans le cas contraire. La matrice $C^{h_0} \in \mathbb{R}^{p \times m}$ est la matrice de choix de route, pour les voyageurs démarrant dans l'intervalle h_0. Elle est définie exactement comme dans le cas statique. Enfin, la matrice d'affectation $A_h^{h_0} \in \mathbb{R}^{n \times m}$ est définie par

$$A_h^{h_0} = L_h^{h_0} C^{h_0}. \tag{6}$$

Le modèle d'affectation dynamique du trafic s'écrit donc

$$v_h = \sum_{h_0 \leq h} A_h^{h_0} q_{h_0} = \sum_{h_0 \leq h} L_h^{h_0} C^{h_0} q_{h_0}. \tag{7}$$

3 Estimation de tables OD

Le problème d'estimation de tables OD statiques (ETODS) vise à identifier une table OD qui reflète au mieux un ensemble de comptages. Nous noterons $\widehat{E}$ l'ensemble des r arcs pour lesquels des données de comptage sont disponibles. Etant donné un vecteur de flots observés $\widehat{v} \in \mathbb{R}^r$, le problème ETODS vise à identifier une table OD q telle que $v_{\widehat{E}}(q)$ soit le plus proche possible de $\widehat{v}$, où le vecteur $v_{\widehat{E}}(q) \in \mathbb{R}^r$ contient les éléments de v correspondant aux arcs de $\widehat{E}$.

La notion de "proximité" dans cette définition peut être modélisée de plusieurs manières. Des modèles basés sur les moindres carrés (Cascetta, 1984, Ashok et Ben-Akiva, 1993, Bierlaire et Toint, 1995), l'entropie (Van Zuylen et Willumsen, 1980, Bell, 1984) et le maximum de vraisemblance (Spiess, 1987) ont notamment été proposés dans la litérature.

Si $\widehat{A}$ est la matrice $r \times m$ composée des lignes de A correspondant à $\widehat{E}$, les flots observés $\widehat{v}$ sont donc obtenus par

$$\widehat{v} = \widehat{A}q + \varepsilon_A \tag{8}$$

où ε est une variable aléatoire appréhendant les erreurs de mesures. Nous supposons qu'elle suit une loi normale multivariée

$$\varepsilon_A \sim N(0, W_A) \tag{9}$$

où $W_A \in \mathbb{R}^{n \times n}$ est la matrice de variance-covariance.

Le problème ETODS peut donc être écrit

$$q^* = \text{argmin}_q \, d(\widehat{A}q, \widehat{v}) \tag{10}$$

où d est une fonction de distance. En général, un nombre infini de tables q sont solutions du problème (10). Dès lors, il est nécessaire de faire intervenir une information a priori q_0, telle que

$$q = q_0 + \varepsilon_I \tag{11}$$

avec $\varepsilon_I \sim N(0, W_I)$. L'importance de cette information a priori varie en fonction de la topologie du réseau, du nombre de paires OD, et de la densité des arcs pour lesquels des données sont disponibles. Bierlaire (2002) a proposé une mesure de cette importance appelée *Total Demand Scale*.

Dans le cas des moindres carrés, (10) devient alors

$$q^* = \text{argmin}_q \left\| \begin{pmatrix} W_A^{-1}\widehat{A} \\ W_I^{-1} I \end{pmatrix} q - \begin{pmatrix} W_A^{-1}\widehat{v} \\ W_I^{-1} q_0 \end{pmatrix} \right\|_2^2$$

Si l'on tient compte explicitement de l'influence des conditions de trafic sur la matrice de choix de route, ce problème est difficile à résoudre.

Dès lors, il est souvent préconisé de résoudre le problème itérativement, en enchaînant estimation de la table OD et affectation du trafic. C'est notamment l'approche suggérée par le logiciel SATURN (Van Vliet, 1982). Des techniques plus récentes, basées sur la programmation bi-niveaux, incluent explicitement les conditions d'équilibre de trafic dans le modèle (Barceló et Casas, 1999). Dans ce papier, nous privilégions la première approche. En effet, dans un cadre temps réel, non seulement la solution de modèles complexes demande trop d'efforts pour être obtenue, mais on peut déléguer la majorité du travail de calibration à des méthodes off-line, et se consacrer on-line uniquement à l'évaluation de déviations par rapport à une situation de référence. Dès lors, les erreurs résultant de l'hypothèse d'un modèle de choix de route fixe sont en général moins importantes.

Pour refléter cela dans le cadre dynamique, nous allons décomposer la table OD de l'intervalle h en une partie de référence, ou *table historique*, et une déviation, c'est-à-dire

$$q_h = q_h^H + \tilde{q}_h. \tag{12}$$

De plus, l'information a priori sera fournie ici par un modèle de prédiction auto-regressif

$$\tilde{q}_h = \sum_{k=h-q'}^{h-1} F_h^k \tilde{q}_k + \omega_h, \quad \omega_h \sim N(0, W_I) \tag{13}$$

où $F_h^k \in \mathbb{R}^{m \times m}$ représente l'influence de la table au temps k sur la table au temps h. Seuls les q' intervalles de temps précédents sont pris en compte dans ce processus. Le modèle (13) doit être calibré sur base de nombreuses données historiques, afin de bien appréhender la dynamique du système. Dès lors, il détermine totalement le processus de prédiction des tables OD, en combinant les tables historiques q_H avec les déviations prédites. L'estimation des matrices s'avère plus compliquée.

En injectant (12) dans (7), nous obtenons

$$\hat{v}_h - \sum_{k=h-p'}^{h} \hat{A}_h^k q_k^H = \sum_{k=h-p'}^{h} \hat{A}_h^k \tilde{q}_k + \varepsilon_h, \quad \varepsilon_h \sim N(0, W_{Ah}) \tag{14}$$

où p' correspond au nombre d'intervalles de temps nécessaire au plus long trajet dans le réseau. L'équation (13) est appelée *équation de transition* et (14) *équation de mesure*. Cette modélisation a été proposée par Ashok et Ben-Akiva (1993). Pour simplifier les notations, nous écrivons (14)

$$\tilde{v}_h = \sum_{k=h-p'}^{h} \widehat{A}_h^k \tilde{q}_k + \varepsilon_h, \quad \varepsilon_h \sim N(0, W_{Ah}) \tag{15}$$

En rassemblant les équations de mesure et de transition pour les intervalles de temps de 1 à h, nous obtenons le problème suivant

$$\begin{pmatrix} I & 0 & \cdots & & & & \cdots & 0 \\ -F_2^1 & I & 0 & \cdots & & & \cdots & 0 \\ -F_3^1 & -F_3^2 & I & 0 & \cdots & & \cdots & 0 \\ & & & & \vdots & & \vdots & \\ 0 & 0 & 0 & \cdots & -F_h^{h-q'} & \cdots & -F_h^{h-1} & I \\ \widehat{A}_1^1 & 0 & \cdots & & & & \cdots & 0 \\ \widehat{A}_2^1 & \widehat{A}_2^2 & \cdots & & & & \cdots & 0 \\ \widehat{A}_3^1 & \widehat{A}_3^2 & \widehat{A}_3^3 & \cdots & & & \cdots & 0 \\ & & & \vdots & & & \vdots & \\ 0 & 0 & 0 & \cdots & \cdots & \widehat{A}_h^{h-p'} & \cdots & \widehat{A}_h^h \end{pmatrix} \begin{pmatrix} \tilde{q}_1 \\ \vdots \\ \tilde{q}_h \end{pmatrix} \sim \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \tilde{v}_1 \\ \tilde{v}_2 \\ \tilde{v}_3 \\ \vdots \\ \tilde{v}_h \end{pmatrix}$$

qui, dans le cadre des moindres carrés, s'écrit

$$\min_q \|W^{-1}\tilde{C}q - W^{-1}\tilde{d}\|_2^2 = \min_q \|Cq - d\|_2^2 \tag{16}$$

avec

$$W = \begin{pmatrix} W_{I1} & 0 & \cdots & & \cdots & 0 \\ 0 & \ddots & & & & \\ \vdots & & W_{Ih} & & & \\ & & & W_{A1} & & \\ \vdots & & & & \ddots & \\ 0 & & & & & W_{Ah} \end{pmatrix}$$

La solution est données par les équations normales

$$q^* = (C^T C)^{-1} C^T d$$

avec

$$\mathrm{Var}(q^*) = (C^T C)^{-1}$$

Clairement, cette modélisation ne permet pas de résoudre efficacement le problème. En effet, la taille de la matrice C est $(hm+hn)\times hm$, ce qui non seulement peut être très élevé lorsque l'on considère des réseaux réels pour lesquels m est grand (~ 10000), mais surtout la taille du problème augmente avec le temps, vu qu'elle dépend de l'intervalle de temps courant h. De plus, les matrices $\tilde{C}$ et W sont très creuses, en ce sens qu'elles contiennent une grande majorité d'éléments nuls, ce qui mérite d'être exploité. Nous décrivons maintenant deux méthodes permettant de résoudre efficacement un tel problème, dans un contexte temps-réel.

4 Le filtre de Kalman

La méthode du filtre de Kalman (Kalman, 1960) est une méthode de résolution incrémentale pour le problème des moindres carrés linéaires. Dans un contexte en temps-réel, elle est conçue pour mettre à jour au fur et à mesure la solution du moindres carrés dès que de nouvelles données sont disponibles. Pour décrire cette méthode, considérons un problème de moindres carrés contenant deux bloc de données :

$$\min_x \left\| \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} x - \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \right\|_2^2 = \min_x \|C_1 x - d_1\|_2^2 + \|C_2 x - d_2\|_2^2 \qquad (17)$$

Nous résolvons d'abord le problème correspondant au premier bloc, c'est-à-dire

$$\min_x \|C_1 x - d_1\|_2^2$$

dont l'équation normale est

$$C_1^T C_1 x_1^* = C_1^T d_1, \qquad (18)$$

et donc la solution est

$$x_1^* = (C_1^T C_1)^{-1} C_1^T d_1. \qquad (19)$$

L'équation normale de (17) s'écrit

$$\begin{aligned}(C_1^T C_1 + C_2^T C_2)x_2^* &= C_1^T d_1 + C_2^T d_2 \\ &= C_1^T C_1 x_1^* + C_2^T d_2 \\ &= C_1^T C_1 x_1^* + C_2^T d_2 + C_2^T C_2 x_1^* - C_2^T C_2 x_1^* \\ &= (C_1^T C_1 + C_2^T C_2)x_1^* + C_2^T (d_2 - C_2 x_1^*).\end{aligned} \tag{20}$$

où la deuxième équation dérive directement de (18). Ainsi, nous obtenons

$$x_2^* = x_1^* + (C_1^T C_1 + C_2^T C_2)^{-1} C_2^T (d_2 - C_2 x_1^*), \tag{21}$$

qui décrit bien x_2^* comme une mise à jour de x_1^*. D'une manière générale, le filtre de Kalman s'écrit pour $h = 2, 3, \ldots,$

$$\begin{aligned}H_h &= H_{h-1} + C_h^T C_h \\ x_h^* &= x_{h-1}^* + H_h^{-1} C_h^T (d_h - C_h x_{h-1}^*)\end{aligned} \tag{22}$$

avec $H_0 = 0$ et $x_0 = 0$.

Cette résolution incrémentale est avantageuse, en ce sens qu'elle évite de maintenir les données relatives aux intervalles 1 à $i-2$. Non seulement la solution est mise à jour, mais la variance des estimateurs H_i^{-1} est propagée d'un intervalle de temps à l'autre. De plus, la taille du problème à résoudre ne dépend pas de h.

Elle comporte cependant quelques désavantages. D'une part, la structure creuse de (16) est complètement ignorée par la méthode, et la matrice H_h est en général pleine, même si (16) est très creux. D'autre part, l'incorporation de contraintes de bornes dans le problème, permettant notamment d'éviter d'obtenir des flots OD négatifs, s'avère non triviale. Enfin, le nombre d'opérations à effectuer à chaque intervalle de temps reste constant, et cela que la solution précédente soit une bonne approximation ou non de la solution finale.

5 LSQR

Pour pallier à ces désavantages, Bierlaire et Crittin (2004) ont proposé une nouvelle méthode, où la résolution de (16) est assurée par l'algorithme

LSQR de Paige et Saunders (1982). LSQR est un algorithme itératif pour la résolution de problèmes de moindres carrés creux de grande taille. Il est analytiquement équivalent à l'algorithme des gradients conjugués (Hestenes et Stiefel, 1952), et converge donc (théoriquement) en un nombre d'itérations inférieur ou égal à la taille du problème. Une des caractéristiques principales de l'algorithme est que la matrice C de (16) ne doit pas être explicitement stockée. L'algorithme utilise uniquement des produits matrice-vecteur Cx et $C^T y$. Ainsi, n'importe quelle implémentation de l'opérateur linéaire et de son adjoint est suffisante. De plus, une version de cet algorithme lorsque des contraintes de bornes sont imposées aux variables a été proposée par Bierlaire et al. (1991).

Afin d'éviter l'explosion du problème (16) lorsque h augmente, Bierlaire et Crittin (2004) ont proposé de tronquer le problème, et de n'y incorporer que les r' derniers intervalles de temps. Cette approximation permet de contrôler le compromis entre précision et taille du modèle.

La méthode s'avère particulièrement efficace pour les problèmes creux. Bierlaire et Crittin (2004) comparent le nombre théorique de flops (floating point operations) pour les deux algorithmes. Par exemple, la figure 1 compare le nombre de flops de chaque méthode pour différentes taille du problème. Le réseau est tel que le nombre d'arcs est dix fois inférieur au nombre de paires OD. Les matrices de variance-covariance sont diagonales, ainsi que les matrices de transition de (13). De plus, $p' = r' = 10$ et $q' = 9$. La densité[2] des matrices d'affectation est de 5%.

Des comparaisons du nombre effectif de flops sur des exemples numériques ont également été effectuées. D'abord, un petit réseau dont on contrôle tous les paramètres a été analysé, afin d'étudier l'impact de la simplification introduite par le problème tronqué. Comme la "vraie" matrice est connue dans ce cas d'école, il est possible de comparer l'erreur de la solution produite par l'algorithme de Kalman à l'erreur produite par LSQR (voir Figure 2 pour un scénario avec $r' = 3$). Il apparaît que l'impact de l'approximation est mineur, mais surtout que l'erreur commise reste relativement stable avec le temps.

Enfin, une comparaison des flops effectifs sur des données réelles (un

[2] Pourcentage d'éléments non nuls

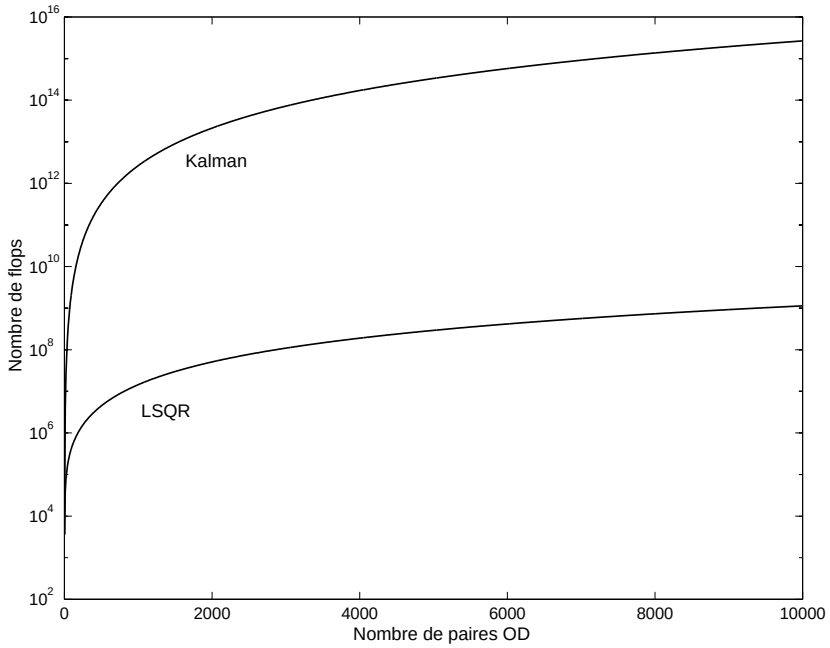

FIG. 1 – Nombre de flops en fonction de la taille du problème

réseau à Irvine, Ca., comportant 618 arcs, 296 noeuds et 627 paires OD) montre le gain substantiel d'efficacité lorsque l'algorithme LSQR est utilisé, alors qu'il produit des solutions très semblables à celles produites par le filtre de Kalman (Figure 3). De plus amples détails sur l'analyse de cette méthode sont présentés par Crittin (2003) et Bierlaire et Crittin (2004).

6 Conclusion

Dans ce papier, nous avons décrit deux aspects importants liés à l'estimation et à la prédiction des tables origine-destination. D'une part, la modélisation du problème comme le problème inverse de l'affectation de trafic a été développée. L'introduction d'une information a priori et de la dimension temporelle a permis de décrire le problème comme un très grand problème de moindres carrés. D'autre part, deux algorithmes pour la résolution de ce problème de moindres carrés ont été présentés. L'algo-

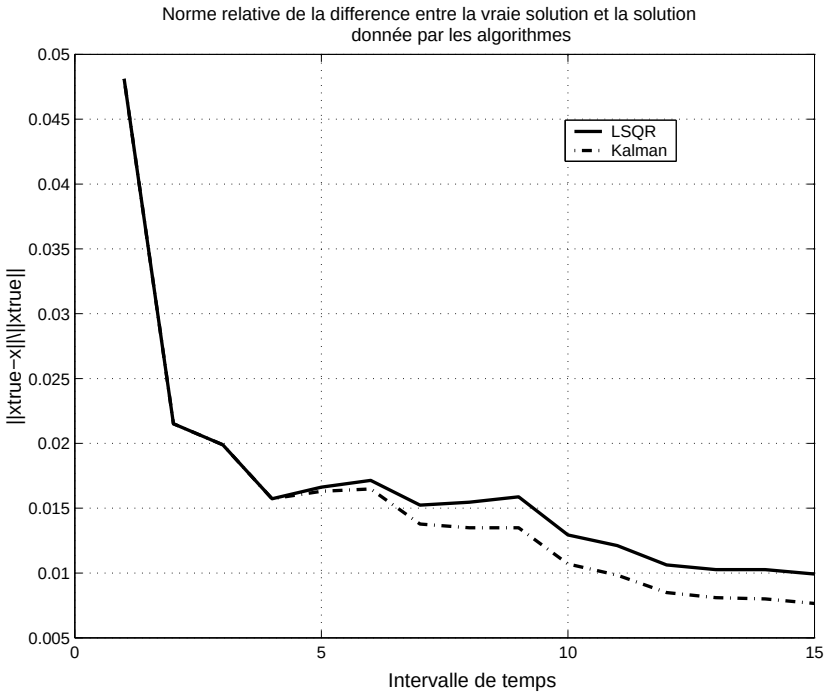

FIG. 2 – Evolution de l'erreur

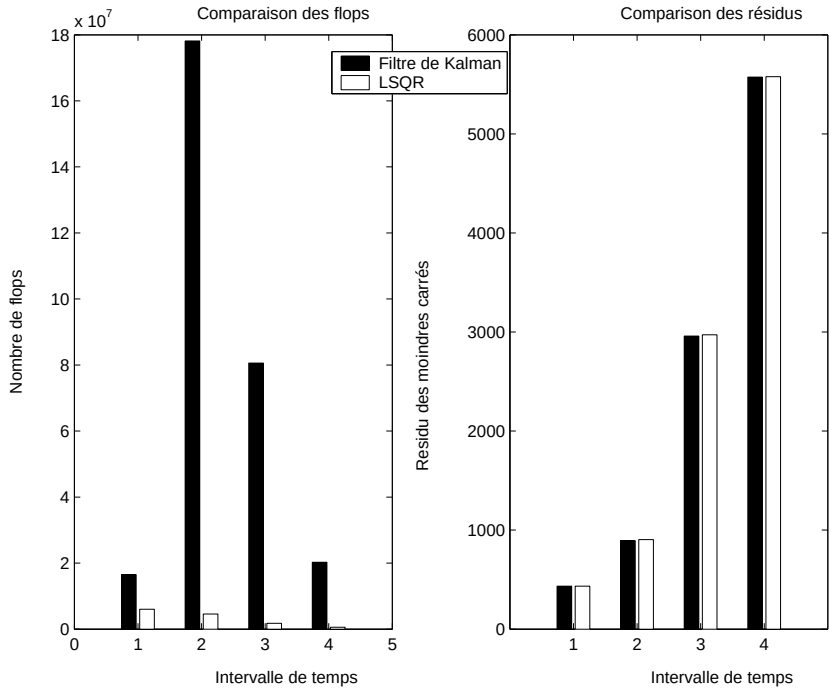

FIG. 3 – Comparaison des flops entre Kalman et LSQR

rithme du filtre de Kalman, utilisé notamment par Ashok et Ben-Akiva (1993), s'avère bien adapté au contexte temps-réel, mais ne permet pas d'exploiter la structure creuse du problème. Bierlaire et Crittin (2004) ont montré que l'algorithme LSQR pouvait être utilisé pour résoudre les problèmes de grandes tailles de manière efficace en temps réel.

Références

Ashok, K. et Ben-Akiva, M. (1993). Dynamic origin-destination matrix estimation and prediction for real-time traffic management systems, *in* C. Daganzo (ed.), *Transportation and Traffic Theory*, Elsevier Science Publishing Company Inc. Proceedings of the 12th ISTTT.

Barceló, J. et Casas, J. (1999). The use of neural networks for short-term prediction of traffic demand, *in* A. Ceder (ed.), *Transportation and Traffic Theory. Proceedings of the 14th ISTTT*, Pergamon, pp. 419–443.

Bell, M. G. H. (1984). Log-linear models for the estimation of origin-destination matrices from traffic counts : an approximation, *in* J. Volmuller et R. Hamerslag (eds), *Proceedings ninth international symposium on transportation and traffic theory*, VNU Science Press, Utrecht, Netherlands.

Ben-Akiva, M. et Bierlaire, M. (1999). Discrete choice methods and their applications to short-term travel decisions, *in* R. Hall (ed.), *Handbook of Transportation Science*, Kluwer, pp. 5–34.

Ben-Akiva, M. et Bierlaire, M. (2003). Discrete choice models with applications to departure time and route choice, *in* R. Hall (ed.), *Handbook of Transportation Science, Second Edition*, Kluwer, pp. 7–37.

Bierlaire, M. (2002). The total demand scale : A new measure of quality for static and dynamic origin-destination trip tables., *Transportation Research B* **36**(9) : 837–850.

Bierlaire, M. et Crittin, F. (2004). An efficient algorithm for real-time estimation and prediction of dynamic od tablel, *Operations Research* **52**(1).

Bierlaire, M. et Toint, P. L. (1995). MEUSE : an origin-destination estimator that exploits structure, *Transportation Research B* **29**(1) : 47–60.

Bierlaire, M., Toint, P. L. et Tuyttens, D. (1991). On iterative algorithms for linear least squares problems with bound constraints, *Linear Algebra and its Applications* **143** : 111–143.

Cascetta, E. (1984). Estimation of trip matrices from traffic counts and survey data : a generalised least squares approach estimator, *Transportation Research B* **18**(4/5) : 289–299.

Crittin, F. (2003). *New algorithmic metohds for real-time transportation problems*, Phd thesis # 2877, Ecole Polytechnique Fédérale de Lausanne.

Hestenes, M. R. et Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems, *J. Res. N.B.S.* **49** : 409–436.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *J. of Basic Eng., Trans. ASME, Series D* **82**(1) : 33–45.

Paige, C. C. et Saunders, M. A. (1982). LSQR : an algorithm for sparse linear equations and sparse least squares, *ACM Transactions on Mathematical Software* **8** : 43–71.

Patriksson, M. (1994). *The Traffic Assignment Problem, Models and Methods*, VSP, Utrecht, NL.

Sheffi, Y. (1985). *Urban Transportation Networks*, Prentice-Hall, Englewood Cliffs, USA.

Spiess, H. (1987). A maximum likelihood model for estimating origin-destination matrices, *Transportation Research B* **21**(5) : 395–412.

Van Vliet, D. (1982). SATURN, a modern assignment model, *Traffic Engineering and Control* **23** : 578–581.

Van Zuylen, H. J. et Willumsen, L. G. (1980). The most likely trip matrix estimated from traffic counts, *Transportation Research B* **14** : 281–293.