Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 98-18

RECURSIVE ESTIMATION BASED ON THE EQUALITY-CONSTRAINED OPTIMIZATION FOR INTERSECTION ORIGIN-DESTINATION MATRICES

Baibing Li and Bart De Moor[2]

March 1998

[1]This paper is also available by anonymous ftp from ftp.esat.kuleuven.ac.be as file rep9818.ps.Z in the directory pub/SISTA/baibing/reports.

[2]ESAT/SISTA, K.U.Leuven, Kardinaal Mercierlaan 94, B-3001 Heverlee (Leuven), Belgium, tel.+32-16-32.18.55, fax +32-16-32.19.70, email: baibing.li@esat.kuleuven.ac.be.

RECURSIVE ESTIMATION BASED ON THE EQUALITY-CONSTRAINED OPTIMIZATION FOR INTERSECTION ORIGIN-DESTINATION MATRICES

Baibing Li and Bart De Moor

ESAT-SISTA, Dept. of Electrical Engineering, Katholieke Universiteit Leuven
Kardinaal Mercierlaan 94, 3001 Leuven, Belgium

Abstract—A fast constrained recursive identification (CRI) algorithm is proposed to estimate intersection origin-destination (O-D) matrices dynamically. The basic idea of the CRI algorithm is to estimate intersection O-D matrices based on equality-constrained optimization and then to adjust them by Bell's correction (1991) for inequality constraints. Numerical results show that the accuracy of estimates by the CRI algorithm is fairly good—the solutions obtained by the CRI are optimal in majority of the cases, while the computational efforts are very limited—increment mainly lies on the evaluation of an inverse for an $m\times m$ matrix (m=4 for a typical intersection) compared with the ordinary recursive least squares method. These results mean that a properly designed recursive algorithm can indeed avoid iterative procedure in each time step to obtain highly accurate on-line estimates for intersection O-D matrices. Therefore, the CRI algorithm with its reasonable balance between accuracy and computational simplicity is very suitable for practical use.

1. INTRODUCTION

Two types of data, origin-destination (O-D) patterns and magnitudes of flow volumes, are fundamental requirements of management and control for many traffic systems. Therefore, in recent decades, a lot of methods on estimations of O-D matrices have been proposed in literature for various structures of traffic systems and different information available for making inference.

For intersection O-D matrix problems, Cremer & Keller (1987) and Nihan & Davis (1987) presented recursive identification methods to reduce the demands on computation and memory. This kind of recursive methods is very useful in practice. It can not only provide the information for planning and design, but also be suitable for algorithms embedding in an adaptive intersection controller for setting signal timings.

From mathematical point of view, the problem of estimating an intersection O-D matrix is equivalent to solving an optimization problem with both equality and inequality constraints. To simplify this constrained optimization problem to obtain a practical solution of recursive estimation, Cremer & Keller (1987) and Nihan & Davis (1987) considered an estimation-correction two step procedure: first, ignoring both equality and inequality constraints on the split parameters and deriving a recursive solution of an unconstrained optimization; second, using appropriate correction methods to enforce these equality and inequality constraints. Compared with the original problem, this procedure is much simpler and faster while it induces inaccuracy in the solution obtained. For a balance between accuracy and computational simplicity for on-line estimations of O-D matrices, it seems that Cremer & Keller (1987) and Nihan & Davis (1987) paid more attention to the latter.

Another end for the balance between accuracy and computational simplicity for dynamic estimations of O-D matrices came from Sherali et al. (1997). They seemed to prefer much more in the former to the latter. They proposed an iterative procedure for the equality- and inequality-constrained optimization to obtain an optimal solution of the split parameters. This solution undoubtedly can improve accuracy of estimators substantially. As the price to pay, this iterative procedure lacks an explicit formula for analysis and causes considerable complication in computation. Moreover, since the estimate is no longer a recursive one, the demands for computation and memory will increase as time and observations.

Considering that the price in computation and memory is so high, Sherali at al (1997) suggested some practical versions of the algorithm as compromises between accuracy and computational efforts: the first one, termed as SAH (I) in this paper, is a modified version of the above theoretical one in which maximum iteration number is limited to 7 and the algorithm is terminated when objective improvement<1% over each of the 4 latest iterations; the second one termed as SAH (II) is based on SAH (I) and adopts rolling horizon framework to decrease demand on memory and computation. Obviously, both SAH (I) and (II) algorithms have less computational efforts and then are more practical than the theoretical one although the optimality of the theoretical algorithm is undermined.

For researches on dynamically estimating O-D matrices, the last approach should be mentioned here is the one proposed by Bell (1991b). Bell (1991b) suggested an algorithm based on the inequality-constrained optimization to estimate O-D matrices while the equality constraints were dropped. Like the SAH (I) and (II), it is an iterative procedure. According to Sherali et al (1997), the computational time required by this method is very high, thereby making this approach unsuitable for real-time implementation. However, it should be noted that the two models with traverse time proposed by Bell (1991b) are very appropriate for dealing with such O-D matrices problems in which the travel time delay in a certain traffic system is relatively large compared with the sampling interval, such as a section of freeway. Moreover, Bell's algorithm is very powerful to deal with inequality constraints. We shall adopt Bell's approach to adjust estimates in this paper.

In this paper, as a new balance between accuracy and computational simplicity, we consider a recursive identification algorithm based on the equality-constrained optimization. It is a compromise between the Cremer-Keller-Nihan-Davis's (CKND) approach mainly based on the unconstrained optimization, and Sherali-Arora-Hobeika's theoretical algorithm based on both the equality- and inequality-constrained optimization. Moreover, it can be conceptually regarded as a complement of Bell's approach which is based on inequality-constrained optimization.

The basic idea of this approach originates from the fact that situations are quite different for equality constraints and inequality constraints. Comparatively, equality constraints are easier to deal with in an analytical and recursive way. On the other hand, non-negative inequality constraints are often automatically satisfied for large sample sizes. All of these suggest that a recursive estimate based on the equality-constrained optimization (rather than CKND's unconstrained one) can be easier to be implemented in algorithms and meanwhile causes less loss in accuracy if the non-negative inequality constraints are enforced (if necessary) after the equality-constrained optimization.

Another important aspect of the CRI algorithm proposed here is that it adopts specific correction method for inequality constraints, termed Bell's correction. This correction method originates from Bell's algorithm (1991) solving inequality-constrained optimization. Since Bell's correction exploits the inherent structure of the problem, the estimates obtained by the CRI algorithm are often optimal although they are not necessary so in theory.

We shall show that the CRI can be implemented in a non-iterative way in each time step by only slightly modifying the CKND's recursive algorithm. Thus, its price in computation is quite low, much less than SAH (I) and (II). We shall also show by numerical results that the achievement in accuracy is satisfactorily good—it has almost the same accuracy as the optimal solution based on both equality- and inequality-constrained optimization and then can largely improve the accuracy of the estimates by CKND's algorithm.

2. PROBLEM FORMULATION

Consider an intersection with m entering legs and n exiting legs. The time period of interest is divided into N intervals. Let

$y_j(t)$ (j=1,...,n) denote the exiting count at leg j during time interval t;

$y(t)=[y_1(t),......,y_n(t)]^T$;

$q_i(t)$ (i=1,...,m) denote the entering count at leg i during time interval t;

$q(t)=[q_1(t),......,q_m(t)]^T$;

J_i={j: exit j is permissible for users entering at entrance i} (i=1,...,m);

I_j={i: entrance i admits users that can take exit j} (j=1,...,n);

b_{ij} (i=1,...,m; j=1,...,n) denote the probability a vehicle entering at $i \in I_j$ exiting at j (j=1,...,n);

$b_j=[b_{ij}]\ \forall\ i \in I_j$ and $Q_j=[q_i]\ \forall\ i \in I_j$ as $|I_j|\times 1$ vectors such that i is in an increasing order for each j=1,...,n;

$b=[b_{1,}{}^T......,b_n{}^T]^T{}_{p\times 1} \mathrel{\hat{=}} [b^{(i)}]_{p\times 1}$,

i.e., the superscript (i) is numbered such that the ith element of b is denoted as $b^{(i)}$, i=1,...,p, where $p=|I_j|\times|J_i|$. z^T denotes the transpose of the vector z.

The b satisfies conservation equations and non-negative constraints:

$$Rb=\mathbf{1}_{m\times 1} \tag{1a}$$

$$b\geq 0 \tag{1b}$$

where R is the coefficient matrix of the conversation equations satisfying $R=diag\{\mathbf{1}^T,...,\mathbf{1}^T\}L$ for a given permutation matrix L and the jth $\mathbf{1}$ of R is an $|I_j|\times 1$ vector of ones.

As an example, consider a typical four-way intersection prohibiting U-turn, $n=m=4$. The conversation equations are

$$\sum_{j=1}^{4} b_{ij} = 1 \quad \text{and} \quad b_{ii}=0 \quad \text{for } i=1,\dots,4. \tag{2}$$

Then, by the above notations, we have

$b_1=[b_{21},b_{31},b_{41}]^T$, $b_2=[b_{12},b_{32},b_{42}]^T$, $b_3=[b_{13},b_{23},b_{43}]^T$, $b_4=[b_{14},b_{24},b_{34}]^T$,

$Q_1=[q_2,q_3,q_4]^T$, $Q_2=[q_1,q_3,q_4]^T$, $Q_3=[q_1,q_2,q_4]^T$, $Q_4=[q_1,q_2,q_3]^T$,

$b=[b_{21},b_{31},b_{41},b_{12},b_{32},b_{42},b_{13},b_{23},b_{43},b_{14},b_{24},b_{34}]^T=[b^{(1)},\dots,b^{(12)}]^T$.

Hence, the conversation equations (2) can be rewritten as (1a) in which

$$R=\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$L=[e_4,e_7,e_{10},e_1,e_8,e_{11},\ e_2,e_5,e_{12},\ e_3,e_6,e_9]$,

where e_j is the jth column of the 12×12 identity matrix.

In this paper, like Cremer & Keller (1987) and Nihan & Davis (1987), we omit traverse time a vehicle passes an intersection in the traffic flow model and deal with it as errors. This is reasonable for intersections due to the traverse time is relatively small compared with sampling time intervals. For other approaches to deal with traverse time, see Bell (1991b) and Sherali et al. (1997).

Then we have linear model (see also, Cremer & Keller, 1987; Nihan & Davis, 1987):

$$y(t) = Q(t)b + e(t) \tag{3}$$

where $e(t)$ is a random vector with zero mean and $Q(t) = diag\{Q_1(t),\cdots,Q_n(t)\}$.

For the observations $y(t)$ and $q(t)$ available at each time interval t in practice, the problem of recursive estimation of b is to find $\hat{b}(t)$ with recursive characteristic such that it can (approximately) solve following optimization problem

$$min\ J(b)= \sum_{s=1}^{t}\sum_{j=1}^{n}\{y_j(s)-Q_j(s)b_j\}^2 \qquad t=1,\dots\dots,N \tag{4}$$

subject to (1).

3. The CRI Algorithm

For the optimization problem (4) and (1), consider following Lagrangian equation

$$L(b;\lambda,\mu)=J(b)+2\lambda^T(Rb-\mathbf{1})-2\mu^T b$$

The Kuhn-Tucher conditions give

$$\hat{b}=\hat{b}_{LS}-(X^TX)^{-1}R^T\lambda+(X^TX)^{-1}\mu \tag{5}$$

with $X^TX=diag\{\sum_{s=1}^{t}Q_1(s)^TQ_1(s),\cdots,\sum_{s=1}^{t}Q_n(s)^TQ_n(s)\}$. Denote

$$X_j^TX_j=\sum_{s=1}^{t}Q_j(s)^TQ_j(s)$$

Then $(X^TX)^{-1}=diag\{(X_1^TX_1)^{-1},\ldots,(X_n^TX_n)^{-1}\}$.

Inserting (5) into the equality constraints (1a), $Rb=\mathbf{1}$, we have

$$\lambda=W[R\hat{b}_{LS}-\mathbf{1}+R(X^TX)^{-1}\mu]$$

with $W=[R(X^TX)^{-1}R^T]^{-1}$. Then, substituting the above equation into (5), we obtain

$$\hat{b}=\hat{b}_{LS}-(X^TX)^{-1}R^TW[R\hat{b}_{LS}-\mathbf{1}]+H\mu \tag{6}$$

with $H=(X^TX)^{-1}-(X^TX)^{-1}R^TWR(X^TX)^{-1}=[h_{ij}]_{p\times p}$

Hence, from (6), the estimate $\hat{b}$ consists of three parts: the first one $\hat{b}_{LS}$ is the least square solution which can be estimated recursively by the recursive formula for $(X^TX)^{-1}$; The second one $(X^TX)^{-1}R^TW[R\hat{b}_{LS}-\mathbf{1}]$ is the equality-constraint correction for the least square solution $\hat{b}_{LS}$. Note that the main computational increment for the equality-constraint correction is to evaluate the $m\times m$ matrix $W=[R(X^TX)^{-1}R^T]^{-1}$. The first two terms actually constitute the solution of equality-constrained optimization problem (4) and (1a); Finally, the third term $H\mu$ is related to the inequality-constraints. Bell (1991a) proposed an algorithm to solve an inequality-constrained optimization problem for estimating O-D matrices. In this paper, we use the first-step of iteration in Bell's algorithm as a correction for inequality constraints, that is, we seek an one-step iteration solution $\mu \geq 0$ according to Bell's algorithm (1991a) to further correct the solution of equality-constrained optimization problem (4) and (1a). When the resulting estimate $\hat{b} \geq 0$ and $\mu_i \hat{b}^{(i)} = 0$ for $\forall i$, it is just the optimal solution by Bell's algorithm. Otherwise, we truncate and normalize its components for enforcing the inequality constraints (1b). Numerical results show that the Bell's correction is very powerful such that, for majority of cases, $\hat{b}$ is optimal solutions after executing the Bell's correction. Hence, truncating and normalizing are often unnecessary.

Algorithm:

Given: Structure matrix for equality-constraints R;

Traffic counter $y_j(t)$ ($j=1,\ldots,n$) and $q_i(t)$ ($i=1,\ldots,m$) at each time interval t.

Step 1. Constituting $Q_j(t)=[q_i(t)]$ for $\forall\ i \in I_j$ as $|I_j| \times 1$ vectors for each ($j=1,\ldots,n$);

Step 2. Computing recursive least squares estimate $\tilde{b}(t) = [\tilde{b}_1^T(t), \ldots\ldots, \tilde{b}_n^T(t)]^T_{p \times 1}$

$$\tilde{b}_j(t) = \tilde{b}_j(t-1) + K_j(t)[y_j(t) - Q_j(t)\tilde{b}_j(t-1)]$$

$$S_j(t) = Q_j^T(t) P_j(t-1) Q_j(t) + 1$$

$$K_j(t) = P_j(t-1) Q_j(t) S_j^{-1}(t)$$

$$P_j(t) = P_j(t-1) - K_j(t) S_j(t) K_j^T(t) \quad (=(X_j^T X_j)^{-1})$$

Step 3. Computing equality-constrained least squares estimate

$$\bar{b}(t) = \tilde{b}(t) - PR^T W[R\tilde{b}(t) - \mathbf{1}]$$

$$P = diag\{P_1(t), \ldots, P_n(t)\} \quad (=(X^T X)^{-1})$$

$$W = [RPR^T]^{-1} \tag{7}$$

Step 4. Computing Bell's correction for inequality-constraints

4.1. Computing H=P-PR^TWRP=$[h_{ij}]_{p\times p}$ and D=$diag\{1/h_{ii}\}$ (see Remark (ii) below)

4.2. Computing μ^*=$T(-D\bar{b}(t))$

where $T(v)$ is a vector function of a vector v truncating negative elements of v to zeros.

4.3. Computing

$$b'(t) = \bar{b}(t) + H\mu^* \tag{8}$$

Step 5. Enforcing inequality constraints (1b) if necessary.

If NOT $b'(t) \geq 0$, then truncating and normalizing, i.e., firstly, truncating those parts of elements of $b'(t)$ which are greater than 1 or less than 0 (denote the truncated one as $b''(t)$); and secondly, each elements of $b''(t)$ which corresponds to ith entrance is divided the ith element of $Rb''(t)$.

Denote the resulting estimate as $\hat{b}(t)$ and use it as the estimate at tth time interval.

END.

Remarks: (i) Initial estimates for $\tilde{b}_j(0)$ and $P_j(0)$ can be taken in any way as in the oridinary recursive least squares.

(ii) Unlike its counterpart in Bell (1991), the matrix H is semi-definitive. However, for the principal diagonal element h_{ii} of the matrix H, we have h_{ii}>0 ($\forall$ j), which is ensured by Lemma given in APPENDIX as long as $|J_i|$>1 $\forall i$ (i.e., each entrance corresponds at least two permissible exits; otherwise, the slit parameter is 1 and then no need to estimate it).

(iii) For the CRI algorithm, the most important aspect in computation is its characteristics of non-iteration in each time interval and recurrence between time intervals. Both of them ensure computational efficiency of the CRI.

(iv) Compared with the CKND's algorithm, the main computational increment of the CRI is need to compute an inverse of a $m\times m$ matrix in (7). Since m is usually quite small for intersection O-D matrices (for a typical four-way intersection, m=4), this does not cause too much difficulty. Moreover, it is noticeable that both the equality and inequality corrections in **Step 3** and **4** share the inverse of matrix RPR^T in (7). Hence, the computational increament for Bell's correction is very limited.

(v) Compared with the general equality- and inequality-constrained optimization in theoretical sense, the CRI method induces inaccuracy since it is only based on an equality-constrained optimization. However, numerical results given below show that the CRI's accuracy is very satisfactory—the solutions of equality-constrained optimization with Bell's correction are optimal for majority of cases.

(vi) Finally, a note for truncation functions. Nihan and Davis proposed a truncation method (see, Nihan and Davis, 1987, Eq. (23)) to seek a scalar γ such that

$$0 \le b_{new} = b_{old} + \gamma[\text{correction term}] \le 1$$

The requirement for this method is b_{old} satisfying the inequality $0 \le b_{old} \le 1$. This is the reason that we use a direct truncation function in **Step 5** rather than using Nihan and Davis'. Based on the same reason, we use the projection by Sherali and Shetty (1980) to project the least squares solution (used as an initial estimate) in CKND's algorithm in section 4.2.

4.NUMERICAL COMPARISON

In this section, the CRI algorithm proposed here for dynamically estimating intersection O-D matrices will be compared with other competing algorithms by simulation. All runs are made with Matlab 5.0 on Sun Ultra 1 Creator workstation at 200 MHz.

The length of whole simulation period is set as M+N in which traffic counts obtained during the first M time intervals are used for obtaining initial estimates and latter N time intervals for comparison. To decrease random fluctuations, 10 separate simulation experiments are conducted for each of following two tests.

For each simulation experiment, define the criterion for comparison on accuracy as the mean absolute value distance (*MAVD*) over the latter *N* periods:

$$MAVD = \sum_{t=M+1}^{N+M} [\sum_{j=1}^{p} \left| \hat{b}^{(j)}(t) - b^{(j)} \right| / p] / N$$

where $b=[b^{(j)}]_{p\times 1}$ is the "true" split parameter vector and $\hat{b}(t) = [\hat{b}^{(j)}]_{p\times 1}$ is the estimate of the "true" parameter vector by each algorithm compared here. Moreover, define the criterion for comparison on computational efforts as the mean CPU time (*ms*) per time interval (MCPUT) over the latter *N* periods:

$$MCPUT = \sum_{t=M+1}^{N+M} [CPU \text{ time for estimation at } t\text{th time interval}] / N$$

4.1. Evaluation of the optimality of the CRI algorithm

The purpose of this study is to compare the accuracy and computational efforts of the CRI algorithm with a constrained optimization algorithm, sequential quadratic programming (SQP) algorithm given by Matlab 5.0, Optimization Toolbox.

Consider a typical four-way intersection prohibiting U-turn, $n=m=4$. The "true" parameter matrix *B* is given by (see Nihan & Davis, 1989):

$$B = \begin{bmatrix} 0 & 0.230 & 0.414 & 0.356 \\ 0.051 & 0 & 0.149 & 0.800 \\ 0.352 & 0.358 & 0 & 0.290 \\ 0.083 & 0.843 & 0.074 & 0 \end{bmatrix} \quad (9)$$

Set M=20 and N=200. Entering counts $q_i(t)$ are generated as the outcome of a Poisson random variable with mean 100, and the turning movement flows $x_{ij}(t)$ (i, j=1,...,4), i.e., the number of vehicles entering at i exiting at j during time interval t, are generated as the outcome of a multinomial random vector with $M(q_i(t),\beta_i)$ (i=1,...,4), where β_i is the ith row (crossing out its zero-element due to $x_{ii}(t)$=0, i=1,...,4) of B in (9). Considering the errors caused by omitted time lag a vehicle traversing from entrance to exit, and other errors such as the ones caused by weather changes or congestion in (3), we generate the exiting counts $y_j(t)$ as the sum of all traffic flows exiting at jth leg during tth time interval, plus a normal perturbation with standard deviation equal to 10% of the expectation sum, i.e.,

$$y(t) = z(t) + [Vu], \quad t=1,...N$$

where $z(t)=[z_1(t),..., z_4(t)]^T$ and $z_j(t)=\sum_{i=1}^{4} x_{ij}(t)$. u is the standard normal random variable. $V=0.1E\{z(t)\}$. $E\{\}$ denote expectation of a random variable. $[s]$ is the rounding off function of s.

To evaluate the optimality of the CRI algorithm, we compute the percentage of optimal solutions obtained by the CRI. If during time interval t, the estimate by the CRI algorithm is optimal by Kuhn-Tucher conditions for the constrained optimization problem (4) and (1), we let $opt(t)=1$; else 0. Then the percentage of optimal solutions (*POS*) for an experiment is defined as:

$$POS = \sum_{t=M+1}^{N+M} opt(t) / N$$

The numerical results of the CRI and SQP algorithms are listed in Table 1.

Table 1. Comparison of the CRI and SQP algorithms

Experiment no.	SQP		CRI		
	MAVD	*MCPUT*	*MAVD*	*MCPUT*	*POS*
1	0.0341	5,204.90	0.0342	5.35	99.5
2	0.0521	5,659.50	0.0516	5.30	99.5
3	0.0589	5,610.35	0.0600	5.35	95.5
4	0.0573	5,289.05	0.0589	5.50	73
5	0.0353	4,510.50	0.0353	5.30	99.5
6	0.0536	4,637.15	0.0536	5.25	99.5
7	0.0432	4,699.85	0.0432	5.30	96
8	0.0435	5.447.15	0.0440	5.30	99
9	0.0490	4,952.40	0.0490	5.30	97
10	0.0439	4,464.70	0.0435	5.55	74
Average	0.0471	5,047.56	0.0473	5.35	93.25

Obviously, from Table1, the accuracy of the CRI is fairly good. The *MAVD*s of the CRI are quite close to the ones of the SQP in each experiment. The *POS* indicates that on average, there are 93.25 % of estimates reaching optimal during the simulation period in each experiment. Meanwhile, the computational efforts are reduced very largely (only about 0.11% of the CPU time of the SQP algorithm). This suggests that a properly designed fast algorithm is necessary and possible to estimate intersection O-D matrices on-line.

Figure 1 gives typical plots of estimates obtained by the SQP and CRI algorithms. It can be seen that they are identical for most of the cases just as *POS* indicates.

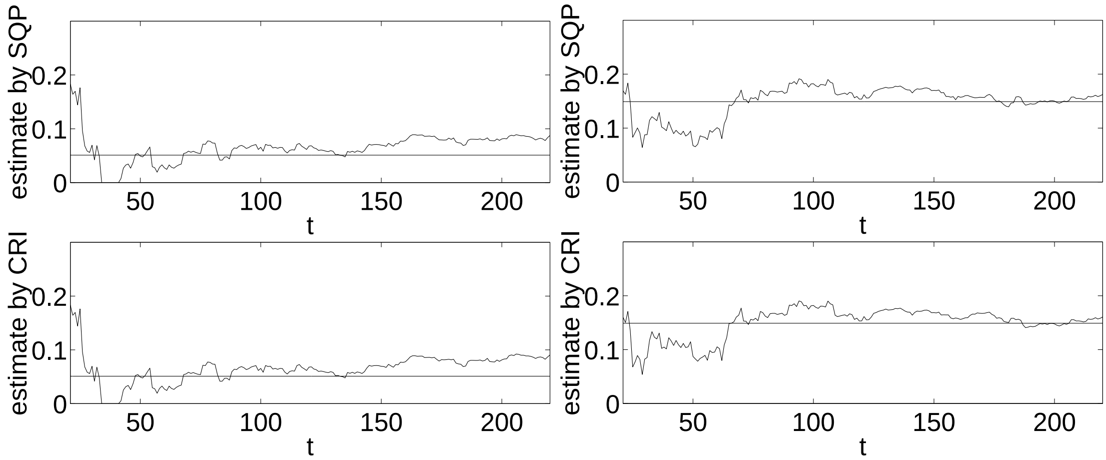

Figure 1(a). Estimates of b_{21}=0.051 by the SQP and CRI algorithms in an experiment

Figure 1(b). Estimates of b_{23}=0.149 by the SQP and CRI algorithms in an experiment

4.2. Comparison for different algorithms

In this sub-section, simulation experiments are conducted for randomly generated split parameter matrices and different lengths of simulation duration. The purpose of this study is to compare performance of the CRI with the ones of following algorithms:

(a) SAH (I). The practical version of SAH's. (see INTRODUCTION for detail);

(b) SAH (II). The rolling-horizon-framework version of SAH (I) in which the data set is updated such that only the latest M=20 traffic counts are retained. See Sherali et al. (1997);

(c) The CKND's algorithm. There is a class algorithms investigated by Cremer & Keller (1987) and Nihan & Davis (1987). We use the recursive formula (7), and normalization-truncation formula (23) and (24) given in Nihan & Davis (1987) for simulation. To eliminate the influence of different choices of initial estimates, $b(0)$ and $P(0)$ are taken as the values of least squares using the first M=20 group traffic counts. If such least squares estimate does not satisfy the constraints (1), project it onto the feasible region by using the algorithm given by Sherali & Shetty (1980). Note that SAH (I) (II) also adopt this algorithm to produce feasible initial solutions. See also, **Remark** (vi) of section 3.

For a typical four-way intersection prohibiting U-turn, n=m=4, the "true" parameter matrix B is generated as the method given by Sherali et al. (1997), that is, for $B=[\beta_{ij}]$ with $\beta_{ii}=0$, $\beta_{ij}\in(0,1)$ ($i\neq j$) is randomly generated on a uniform distribution, and the rows are scaled such that $\sum_{j=1}^{4}\beta_{ij}=1$ for all i. Set M=20. N is set to be 100, 150, 200, 300, 500 respectively to investigate influences of duration length for accuracy and computational efforts for each algorithm. Entering and exiting counts $q_i(t)$ and $y_j(t)$ are generated as in section 4.1.

The summary of the numerical results of 10 simulation experiments are given in Table 2.

Table 2. Average performances of four algorithms over 10 experiments

N	SAH(I)		SAH(II)		CKND		CRI	
	MAVD	*MCPUT*	*MAVD*	*MCPUT*	*MAVD*	*MCPUT*	*MAVD*	*MCPUT*
100	0.0647	190.19	0.0840	98.66	0.1091	4.59	0.0667	5.41
150	0.0534	234.98	0.0827	98.07	0.1124	4.56	0.0534	5.33
200	0.0432	279.63	0.0856	98.34	0.0708	4.54	0.0430	5.31
300	0.0418	370.55	0.0857	97.93	0.0668	4.53	0.0419	5.34
500	0.0340	556.64	0.0830	97.86	0.0578	4.51	0.0341	5.29

It can be seen from the Table 2 that:

(i) The CRI has almost the same accuracy as the SAH (I), both of which are significantly better than others on accuracy.

(ii) The computational effort of the CRI only increases slightly compared with the CKND's algorithm, but is still quite satisfactory.

(iii) From the point of view of accuracy, the SAH (II) is the worst one. The main reason is that relative less information (only M=20 groups of traffic counts) is used to estimate parameters in every time interval. The previous run's solution at each time interval can only provide a good start point for optimization to reduce search time but not improve the quality of estimate itself. This can be also seen from the fact that all of the algorithms compared here have increasing accuracy as N becomes large except for the SAH (II), and the SAH (II) has even worse accuracy than CKND's when $N \geq 200$.

(iv) From the computational point of view, the SAH (I) is the longest one. Except for its iterative characteristics in each time interval, the more computational efforts originate from its non-recurrence. This can be also seen from the fact that the computational efforts (CPU time) of all algorithms compared here are stable for different lengths of simulation duration except for the SAH (I).

5.CONCLUSIONS

Based on the equality-constrained optimization, we propose a fast constrained recursive identification algorithm to estimate intersection O-D matrices. From the algorithm point of view, it is only a slight modification of the CKND's algorithm and then its increment in computation is quite limited. It is a non-iterative procedure in each time interval and retains the recursive characteristics such that its computational and memory demands do not increase as time. Hence, it is very efficient in computational sense compared with the SAH (I) and (II). The numerical results show that the sub-optimal solutions obtained by the CRI are fairly good—they are very close to the optimal solutions obtained by using general constrained optimization algorithm, SQP. This good balance between accuracy and computational efforts suggests that it is very suitable for on-line estimations of intersection O-D matrices.

APPENDIX.

Lemma. Suppose that P is a $p\times p$ non-singular symmetrical matrix, L is a $p\times p$ permutation matrix and $R=diag\{\mathbf{1}^{\mathrm{T}},...,\mathbf{1}^{\mathrm{T}}\}L$, where the length m_l of the lth $\mathbf{1}$ of R satisfies $m_l>1$ and $\sum_{l=1}^{m} m_l=p$. Then,

$$H=P-PR^T[RPR^T]^{-1}RP\geq 0$$

and its principal diagonal element $h_{ii}>0$ ($j=1,...,m$).

Proof: Denote $U=P^{1/2}$ and $A=(RP^{1/2})^T$. Let G be a $p\times(p-m)$ matrix with rank(G)=p-m such that $G^TA=0$. Then $H=U[I-A(A^TA)^{-1}A^T]U=U[G(G^TG)^{-1}G^T]U\geq 0$.

Let u_i denote the ith column of U ($i=1,...,n$). Then $P^{-1/2}u_i=e_i$, where e_i is the ith column of identity matrix I. Hence, we have

$h_{ii}=0 \Leftrightarrow u_i^T[G(G^TG)^{-1}G^T]u_i=0 \Leftrightarrow (G^TG)^{-1/2}G^Tu_i=0 \Leftrightarrow G^Tu_i=0 \Leftrightarrow \exists$ an $m\times 1$ vector a such that $u_i=Aa=(RP^{1/2})^Ta \Leftrightarrow e_i=R^Ta \Leftrightarrow e_i=L^Tdiag\{\mathbf{1},...,\mathbf{1}\}a \Leftrightarrow Le_i=diag\{\mathbf{1},...,\mathbf{1}\}a \Leftrightarrow \exists\ e_k$ (of the kth column of I) such that $e_k=diag\{\mathbf{1},...,\mathbf{1}\}a$.

Thus, the proof is completed by noting that the last equations hold only if the length of kth $\mathbf{1}$ is 1.

Acknowledgments—The first author of this paper is a research fellow (post-doctor) of K.U.Leuven and the second author is a senior Research Associate with the FWO-Flanders. This work was supported by the Fellowship of K.U.Leuven (F97-72), Sustainable Mobility Programme-Project MD/01/24 (1997-2000) (Traffic Congestion Problems in Belgium: Mathematical Models, Analysis, Simulation, Control and Actions), Concerted Research Action GOA-MIPS, the FWO project G.0256.97: Numerical Algorithms for Subspace

System Identification, the FWO Research Communities: ICCOS and Advanced Numerical Methods for Mathematical Modeling, and the Belgian Government: Interuniversity Attraction Poles (IUAP P4-02 and P4-24).

REFERENCES

Bell, M.G.H. (1991a) The estimation of origin-destination matrices by constrained generalized least squares. *Transpn Res.*, Series B25, 13-22.

Bell, M.G.H. (1991b) The real-time estimation of origin-destination flows in the presence of platoon dispersion. *Transpn Res.*, Series B25, 115-125.

Cremer, M. and Keller, H. (1987) A new class of dynamic methods for the identification of origin-destination flows. *Transpn Res.*, Series B21, 117-132.

Nihan, N. and Davis, G. (1987) Recursive estimation of origin-destination matrices from input/output counts. *Transpn Res.*, Series B21, 149-163.

Nihan, N. and Davis, G. (1989) Application of prediction-error minimization and maximum likelihood to estimate intersection O-D matrices from traffic counts. *Transpn Sci.*, Vol.23, 77-90.

Sherali, H.D. and Shetty, C.M. (1980) On the generation of deep disjunctive cutting planes. *Naval Res. Logistics Quart*. 35, 473-492.

Sherali, H.D., Arora, N. and Hobeika, A.G. (1997) Parameter optimization methods for estimating dynamic origin-destination trip-tables. *Transpn Res.,* Series B31, 141-157.