## **OPERATIONS RESEARCH CENTER**

working paper



### MASSACHUSETTS INSTITUTE OF TECHNOLOGY

# Equilibration Operators for the Solution of Constrained Matrix Problems

by

A. Nagurney and A. Robinson

OR 196-89

June 1989

Equilibration Operators for the Solution of Constrained Matrix Problems

> by Anna Nagurney and Alan Robinson

School of Management University of Massachusetts Amherst, Massachusetts 01003

June 1989

•

#### ABSTRACT

In this paper two equilibration algorithms are introduced for the general quadratic constrained matrix problem, whose object is to compute the best possible estimate of an unknown matrix with known bounds on individual entries, row and column totals, and totals of subsets of individual entries. The problem has been widely studied due its frequent appearance as a "core" problem in numerous application areas of operations research. A considerable amount of work has been done on this problem with the quadratic objective function restricted to be diagonal; here we allow any positive definite quadratic form as minimand. We also provide computational comparisons of the two algorithms with each other, and with our implementation of an algorithm of Bachem and Korte.

1

#### 1. INTRODUCTION

This paper discusses the problem of computing the best possible estimate X of an unknown matrix with known bounds on individual entries, row and column totals, and totals of subsets of individual entries. The matrix X might be required to be a functional form of another known matrix, or to be the minimum "distance" from a given matrix  $X^0$ . Bacharach [1] gave the problem its name (i.e. "Constrained Matrix"); before and since it has been widely studied due to its frequent appearance as a "core" problem in numerous application areas of operations research. These include the estimation of origin-destination flows in traffic analysis (Carey and Revelli [7], Carey, Hendrickson and Siddharthan [6]), the estimation of input-output tables (Bachem and Korte [3]), social-accounting matrices (Van der Ploeg [43,44], Morrison and Thuman [31], Harrigan and Buchanan [21], Byron [5], Friedlander [19], Zenios, Drud and Mulvey [47]), of contingency tables in statistics (Deming and Stephan [14]), the projection of traffic within telecommunication networks (Kruithoff [27]), the treatment of census data (Erickson [15]), and the analysis of political voting patterns (Johnston, Hay and Taylor [25]). Recently, there has also been considerable interest in these methods in image resconstruction in electron microscopy and diagnostic radiology (Herman and Lent [22], Herman, Lent, and Rowland [23], Lent and Censor [28]).

Deming and Stephan [14] treated as a constrained matrix problem the statistical problem of estimating an unknown contingency table X as an "iteratively proportioned" (i.e. row and column scaled) version of an initial matrix  $X^0$ , with nonnegative individual entries and given marginals. They gave, however, no existence, uniqueness, nor convergence proofs for their method. These proofs followed later (Gorman [20], Bingen [4], Bacharach [1]) after Sir Richard Stone [42] independently discovered and resurrected the method, which has become widely known as the RAS method, after the functional form it deals with.

Other researchers have formulated constrained matrix problems as mathematical programming problems, with an objective function that forces "conservatism" on the

process of rationalizing X from the initial estimate  $X^0$ . The intellectual foundation for the approach is threefold. Firstly, if viewed from the perspective of mathematical statistics, the quadratic penalty function gives as solution the minimum variance unbiased linear estimate of the matrix X (Byron [5], Van der Ploeg [43,44], Carey and Revelli [7]). Secondly, if viewed from the perspective of information theory, the entropy function gives rise to the estimate of X which minimizes the "information added" to  $X^0$  needed to conform to the contraints (Wilson [46], Snickars and Weibull [41], Erickson [15], Erlander, Jornsten and Lundgren [16]). Lastly, it has been shown (Bacharach [1]) that a particular functional form, the result of application of the RAS method, is equivalent to constrained entropy minimization.

In this paper we consider the general quadratic constrained matrix problem. We allow any positive definite quadratic form as objective function, allow for row and column totals to be specified, allow bounds on individual entries, and allow for constraints on totals of subsets of individual entries. Byron [5] and Van der Ploeg [43,44] considered general penalty matrices and general equality constraints on variables, but did not allow for variable bounding. Ohuchi and Kaji [36,37] considered a diagonal quadratic form, and constraints of the transshipment type. Morrison and Thuman [31] considered several nonlinear objective functions subject to equality constraints, but also did not allow for bounding of variables. Cottle, Duvall, and Zikan [8] developed a specialized decomposition scheme for the case where the quadratic matrix was the identity, and did allow bounds on individual entries. A computational scheme for a more general version (although a diagonal matrix is still assumed) is given Harrigan and Buchanan [21] who used the algorithm of Bachem and Korte [2] to compute the solution of interval-constrained input-output problems.

Our computational procedure is motivated, in part, by the problem at hand; the "equilibration" of matrices (cf. Van der Sluis [45]). We propose a decomposition scheme which resolves the main problem into a series of equilibrium subproblems of three types, which we shall call the row, column, and cut-set problems. We introduce equilibration operators for each of these problem types, and embed these in the iterative price

decentralization scheme of Pang [38]. Equilibration operators were first introduced by Dafermos and Sparrow [13] for the traffic assignment problem. The theory has since been extended to the framework of the spatial price equilibrium problem (where demands and supplies are *elastic*) by Nagurney [33,34] and Dafermos and Nagurney [12]. Nagurney [35] has established that the spatial price equilibrium problem is isomorphic to the constrained matrix problem with a diagonal quadratic form and unknown row and column totals. For analytical results about equilibration operators, the interested reader might wish to consult Eydeland and Nagurney [17].

The paper is organized as follows: in Section 2 we present the formulation of the problem. In Section 3 we give the algorithms, including the outer loop and the alternative equilibration operators. In Section 4 we present the computational experiments illustrating the relative performance of the two algorithms and an algorithm of Bachem and Korte [2]. The test runs covered a wide range of problem sizes and densities. We then summarize and conclude.

#### 2. FORMULATION OF THE PROBLEM

We now give the formulation of the constrained matrix problem considered in this paper. The constraints are of the transportation type with an arbitrary set of side constraints. We denote the given  $m \times n$  matrix by  $X^0 = (x_{ij}^0)$ , and the matrix estimate by  $X = (x_{ij})$ . Let  $s_i$  denote the given row i total,  $d_j$  the given column j total, and  $t_h$ the right hand side of the  $h^{th}$  additional constraint. Let  $u_{ij}$ ,  $l_{ij}$  denote respectively the upper and lower bounds of the variable  $x_{ij}$ . Let the  $mn \times mn$  matrix  $Q = (Q_{ijkl})$  denote the imposed weight for the mixed variable term  $(x_{ij} - x_{ij}^0)(x_{kl} - x_{kl}^0)$  and assume the matrix Q to be strictly positive definite. Note that while Q is an  $mn \times mn$  matrix, we shall continue to use double subscripting to refer to its individual components.

Then our problem may be written as follows:

Minimize 
$$1/2 \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{n} Q_{ijkl}(x_{ij} - x_{ij}^{0})(x_{kl} - x_{kl}^{0})$$
 (1)

subject to: 
$$\sum_{j=1}^{n} x_{ij} = s_i, \quad i = 1, ..., m$$
 (2)

$$\sum_{i=1}^{m} x_{ij} = d_j, \qquad j = 1, \ldots, n \tag{3}$$

$$\sum_{ij} a_{ij} x_{ij} = t_h, \qquad h = 1, \dots, p \tag{4}$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{for all } i,j \tag{5}$$

where the minimand represents the weighted squared sums of the deviations.

Deming and Stefan [14] considered (1) with Q diagonal,  $(Q_{ijij}) = 1/x_{ij}^0$  subject to constraints (2) and (3), whereas Friedlander [19] considered the case where Q = I. Bachem and Korte [2] treated (1) for a general diagonal matrix Q with all the constraints (2)-(5). Cottle et al. [8], on the other hand, studied Friedlander's problem with the additional constraints (5). Ohuchi and Kaji [36,37] also studied the Bachem and Korte problem with upper and lower bounds. For a discussion of applications to transportation with constraints (2),(3), and (5), see Florian [18]. In a more general setting, Morrison and Thumann [31] studied the constrained matrix problem with constraints (2)-(5), but retained the requirement that Q be diagonal. Harrigan and Buchanan [21] formulated the problem in the framework of input-output estimation, with interval constraints, rather than equalities in (2)-(4) and used an expanded diagonal objective function. For an overview of input/output matrices and applications, see Polenske [39] and Miller and Blair [30]. Van der Ploeg [43,44] studied problem (1) with the equality constraints (2)-(4) and applied it to social/national accounts. The objective function (1) permits the utilization of mixed-variable weight terms and so extends the modelling capabilities of the constrained matrix problem. An example of a possible fully dense Q matrix is the inverse of the variance-covariance matrix (cf. Mosteller and Tukey [32]). For other applications where mixed variable weight terms might be desirable, the interested reader should consult Judge and Yancey [26], or Harrigan and Buchanan [21].

Our method uses Lagrangean relaxation to resolve the above problem into three categories of subproblems that are easy to solve: a row subproblem, a column subproblem, and a cut-set subproblem. Each subproblem in turn is viewed as an equilibrium problem and is efficiently solved by specially designed equilibration operators. As noted by previous authors, this general approach may be thought of as a block cyclic ascent method applied to the dual problem, or a modified Gauss-Seidel scheme to solve the Kuhn-Tucker conditions. Q may be any strictly positive definite matrix.

5

#### **3. THE ALGORITHM**

We begin this section with a discussion of the outer loop of the algorithm. In Sections 3.2 and 3.3 we discuss two alternative equilibration operators to solve the row subproblems, and in Section 3.4 we summarize the solution procedure for the row subproblems. In Section 3.5 we discuss the solution of the column and cut-set subproblems.

#### 3.1 The Outer Loop

For simplicity of exposition, we shorthand the constraints (2)-(4) to  $A_1x = b_1$ ,  $A_2x = b_2$  and  $A_3x = b_3$ . x denotes the vectorization of the matrix X; i.e.  $x = (x_{11}, ..., x_{1n}, x_{21}, ..., x_{2n}, ..., x_{m1}, ..., x_{mn})$ . We shall continue, however, to use double subscripting to refer to *individual* components of x. Let  $y_1 \in R^m, y_2 \in R^n, y_3 \in R^p$  be the dual vectors corresponding to the row, column and cutset constraints respectively. Let  $S = \{x \mid x \text{ satisfies (5)}\}$ . The outer loop is as follows:

#### Algorithm: Outer Loop

Step 1 (Initialization) Let  $y^0 = (y_1^0, y_2^0, y_3^0)$  be an arbitrary non-negative vector. Let k = 0.

Step 2 To obtain  $y^{k+1} = (y_1^{k+1}, y_2^{k+1}, y_3^{k+1})$  solve three quadratic problems  $(QP_{\alpha})$  consecutively for  $\alpha = 1, 2, 3$ :

Minimize 
$$1/2x^T Q x + \left(c^T - \sum_{\beta < \alpha} (y_{\beta}^{k+1})^T A_{\beta} - \sum_{\beta > \alpha} (y_{\beta}^k)^T A_{\beta}\right) x$$
  
Subject to:  $A_{\alpha} x = b_{\alpha}$  and  $x \in S$ . (6)

Step 3 If the convergence criterion is not satisfied, return to step 2, otherwise stop.

Pang [38] proposed the above scheme, and proved its convergence for any strictly convex quadratic programming problem that is feasible. Cottle et al. [8] prove convergence of their algorithm under a strong consistency assumption which is a Slater-type constraint qualification. Problems  $(QP_{\alpha})$  for  $\alpha = 1, 2, 3$  are referred to henceforth as the row equilibration subproblems, the column equilibration subproblems, and the cut-set equilibration subproblems respectively. Figure 1 gives a pictorial overview of the decomposition scheme.

:



\*

#### **3.2** Explanation of First Method for Row Equilibration

We now present the equilibration operators for the solution of each of these subproblems  $QP_{\alpha}$ . As the column and cut-set equilibration operators are straightforward modifications of the row equilibration operator, we present only the row operator here and indicate later in Section 3.5 the changes needed to solve subproblems  $QP_2$  and  $QP_3$ .

At iteration k + 1 of the outer loop, the row equilibration subproblem which must be solved is

Minimize 
$$1/2x^TQx + (c^T - y_2^kA_2 - y_3^kA_3)x$$
  
Subject to: 
$$A_1x = b_1, \quad x \in S$$
 (7)

Each of the two equilibration operators described below solves (7). Both are based conceptually in the theory of equilibration operators for the traffic assignment problem introduced by Dafermos and Sparrow [13], and later generalized by Dafermos [9]. It is to be noted, however, that these early operators did not allow for bounded variables.

We first let  $\tilde{c} = (\tilde{c}_{ij}) = (c^T - y_2^k A_2 - y_3^k A_3)$  in the objective function of (7). Our subproblem then becomes:

Minimize 
$$1/2x^TQx + \tilde{c}^Tx$$
  
subject to:  $\sum_{j=1}^n x_{ij} = s_i$   $i = 1, \dots, m$  (8)  
 $l_{ij} \leq x_{ij} \leq u_{ij}$   $i = 1, \dots, m; j = 1, \dots, n$ 

If we now define  $C_{ij}(x) = (Qx)_{ij} + \tilde{c}_{ij}$  then the equilibrium conditions for the problem (8) are given in the following Theorem:

**Theorem 1.** A vector x satisfying the constraints of (8) is a solution of (8) iff it has the following property: For any row i = 1, ..., m, the columns j = 1, ..., n can be so numbered that

:

$$C_{ij_{1}} \leq C_{ij_{2}} \leq \ldots \leq C_{ij_{l}} \leq C_{ij_{l+1}} = \ldots = C_{ij_{s}} = \lambda_{i} \leq C_{ij_{s+1}} \leq \ldots \leq C_{ij_{n}}$$
(9)  
where  $x_{ij} = u_{ij}, \quad j = j_{1}, \ldots, j_{l}$   
 $l_{ij} < x_{ij} < u_{ij}, \quad j = j_{l+1}, \ldots, j_{s}$   
 $x_{ij} = l_{ij}, \quad j = j_{s+1}, \ldots, j_{n}$ 

**Proof:** The property stated above is equivalent to the Kuhn-Tucker conditions.

Both schemes for solving (8) exploit the special structure of these equilibrium conditions. The first scheme attacks the conditions directly, adjusting matrix entries towards the Kuhn-Tucker conditions until these are satisfied. The second scheme uses some recent results in the theory of variational inequalities to solve (8) via a series of subproblems where the quadratic term is diagonalized, and the linear term is iteratively updated until convergence occurs.

#### 3.2.1 The Row Equilibration Operator $R^1$

Conditions (9) characterize the optimal solution as an equilibrium point in the following sense: all variables  $x_{ij}$  at lower bound have high marginal cost  $C_{ij}(x)$ , those at upper bound have low marginal cost, and those in the interior of their bounds must all have the same marginal cost  $\Delta$ . This scheme updates a vector  $x^k$  to  $x^{k+1}$  for each row *i* in turn iteratively adjusting a pair of variables  $x_{ir}$  and  $x_{iq}$  which are most "out of equilibrium." These are altered, while all other variables are held constant, until one of them hits a bound, or until a minimum is reached. Formally; for any  $x^k$  and row *i*,  $R_i^1 x^k$  is defined by the following procedure:

#### Equilibration Operator $R^1$

Step 1. Find:

$$egin{array}{rl} r &= ext{column index j} & ext{s.t. } C_{ij} = \max\{C_{ij}|x_{ij}^k>0, \ j=1,\ldots,n\} \ q &= ext{column index j} & ext{s.t. } C_{ij} = \min\{C_{ij}|x_{ij}^k 
eq u_{ij}, j=1,\ldots,n\} \end{array}$$

Step 2. Solve:

This is easily solved. Let

$$\Delta = \min\left(\frac{C_{ir} - C_{iq}}{Q_{ir} + Q_{iq} - 2Q_{rq}}, x_{ir}^{k}, u_{iq} - x_{iq}^{k}\right)$$
(10)

Then  $R_i x^k = x^{k+1} - \Delta \mathbf{e}_{ir} + \Delta \mathbf{e}_{iq}$ , where  $\mathbf{e}_{jk} \in R^{mn}$  is the  $(jk)^{th}$  unit vector.

We apply  $R_i^1$  until the equilibrium conditions given in Theorem 1 hold for row i. We then proceed to  $R_{i+1}^1$  to equilibrate the  $(i+1)^{st}$  row. We continue this procedure until all rows are in equilibrium. The operator  $R^1$  is defined as the composition

$$(R_1^1 \circ R_2^1 \cdots R_m^1) \circ (R_1^1 \circ R_2^1 \cdots R_m^1) \cdots \circ (R_1^1 \circ R_2^1 \cdots R_m^1)$$

Convergence follows from an adaptation of the proof of convergence for the equilibration operator without upper bounds in Dafermos [9].

In the special case where Q is a diagonal matrix, then all rows can be equilibrated in parallel, for each row subproblem is independent of all other row subproblems. Cottle et al. [8] also make this observation.

#### 3.3 Explanation of Second Method for Row Equilibration

We now present a second operator that may be used to solve (8). It is based on a projection method for the solution of variational inequalities first proposed by Dafermos [11]. This method calls the Row Equilibration Operator  $R^2$  as a subroutine at each iteration. Denote by

$$K = \{x \mid \Sigma_j x_{ij} = s_i, \ l_{ij} \leq x_{ij} \leq u_{ij} \ orall i, j\}$$

the (convex) feasible region to problem (RE). Since a strictly convex function has only one local minimizer over a bounded, closed convex region, the optimal solution satisfies the variational inequality

$$abla f(x^*)^T(x-x^*) \geq 0, \qquad orall \ x \in K$$

The reader may verify that this is equivalent to the fixed point problem

Find 
$$x^* \in K$$
 s.t.  $x^* = P_K^G[x^* - G^{-1}\nabla f(x^*)]$ 

where  $P_K^G(x)$  denotes the projection of x onto K with respect to the inner product norm  $\langle x, y \rangle_G = \langle x, Gy \rangle_2$ , where G is any symmetric positive definite matrix. Our projection algorithm is thus

$$x^{k+1} = P_K^G[x^k - G^{-1} \nabla f(x^k)]$$

We choose G to be diagonal (Q). So we need to solve

÷

$$x^{k+1} = \min_{x \in K} \|x - x^k + G^{-1} 
abla f(x^k)\|_G^2$$

to find  $x^k$ . The reader can verify that this gives rise to the iterative scheme

$$x^{k+1} = \min_{x \in K} 1/2x^T \tilde{Q}x + (\tilde{Q}x^k + c - Qx^k)^T x$$
(+)

where  $\tilde{Q} = \text{diagonal}(Q)$ . For proof that (†) is indeed a contraction map, and the associated fixed-point is unique, the reader should consult Dafermos [11]. The projection method is as follows:

#### Algorithm: Projection Method

Step 1. Start with any feasible flow  $x^0$ .

Step 2. Given  $x^k$ , find  $x^{k+1}$  by solving the quadratic program:

Minimize 
$$1/2 \quad x^T \tilde{Q} x + \tilde{\tilde{c}}^T x$$
  
subject to:  $\sum_{j=1}^n x_{ij} = s_i$  (11)  
 $l_{ij} \le x_{ij} \le u_{ij}$ 

where  $\tilde{Q}$  denotes diagonal(Q), and  $\tilde{\tilde{c}}^{k} = \tilde{Q}x^{k} + c - Qx^{k}$ . **Step 3.** If a convergence criterion is satisfied, then stop, otherwise set k = k + 1, update  $\tilde{\tilde{c}}^{k+1}$  and repeat step 2.

### **3.3.1** The Row Equilibration Operator $R^2$

;

We now give a specialized, efficient and exact equilibration operator  $R_2^i$  to solve (11). As for  $R_1^i$ , this is motivated by the Kuhn-Tucker conditions for the problem (11). The scheme is based on the computation of the Lagrange multiplier  $\lambda_i$  where  $C_{ij_o}$  of (9) takes now the special form:

$$C_{ij_o} = w_{ij_o i j_o} x_{i j_o} + \tilde{\tilde{c}}_{i j_o}$$

since  $\bar{Q}$  is diagonal. This scheme provides the exact solution for subproblem (7).  $R_i^2$  is motivated as follows: If we know  $\lambda_i$  in (9) with the  $C_{ij}$ 's given by (9), then

$$\begin{aligned} x_{ij} &= \left(\frac{\lambda_i - \tilde{c}_{ij}}{w_{ijij}}\right), & \text{for } j_{l+1} \leq j \leq j_s \\ x_{ij} &= 0, & \text{for } j_{s+1} \leq j \leq j_n \\ x_{ij} &= u_{ij}, & \text{for } j_1 \leq j \leq j_l \end{aligned}$$
(12)

where 
$$\lambda_i = rac{s_i - \sum_{j=j_1}^{j_l} u_{ij} + \sum_{j=j_{l+1}}^{j_s} rac{\tilde{z}_{ij}}{w_{ijij}}}{\sum_{j=j_l}^{j_s} rac{1}{w_{iiii}}}$$

c

The statement of the operator is as follows (for a special case where  $w_{ijij} = 1$  see Cottle, et al. [8]; for the unbounded case, in disjoint networks, see Dafermos and Sparrow [13]).

#### Row Equilibration Operator $R^2$

Step 1. Sort the components  $\tilde{\tilde{c}}_{is}, \ldots, \tilde{\tilde{c}}_{in}$  in ascending order and relabel  $x_{i1}, \ldots, x_{in}$  and  $\tilde{\tilde{c}}$  accordingly. Henceforth, we assume that  $\tilde{\tilde{c}}_{i1} \leq \tilde{\tilde{c}}_{i2} \leq \ldots \leq \tilde{\tilde{c}}_{in}$ . Define  $\mathcal{M} = \{1, \ldots, m+1\}$  and  $\mathcal{H} = \emptyset$ . Let  $\mathcal{K} = \mathcal{M}/\mathcal{H} = \{j_i, \ldots, j_{\hat{K}+1}\}$  where  $j_1 < \ldots < j_{\hat{K}} < j_{\hat{K}+1} = m+1$ . Let  $\bar{\ell} = 1$ .

Step 2. Define

$$\lambda_{\overline{l}} \equiv \frac{s_i - \sum_{h \in \mathcal{H}} u_{ih} + \sum_{\overline{k}=1}^{\overline{l}} \frac{\tilde{c}_{ij_{\overline{k}}}}{w_{ij_{\overline{k}}}ij_{\overline{k}}}}{\sum_{\overline{k}=1}^{\overline{l}} \frac{1}{w_{ij_{\overline{k}}}ij_{\overline{k}}}}$$

**Step 3.** If  $\lambda_{\overline{l}} \in [\tilde{\tilde{c}}_{ij_{\overline{l}}}, \tilde{\tilde{c}}_{ij_{\overline{l+1}}}]$ , then for  $\overline{k} = 1, \ldots, \hat{K}$ , let

$$\boldsymbol{x}_{ij_{\overline{k}}} = \begin{cases} \frac{\lambda_{\overline{\ell}} - \tilde{c}_{ij_{\overline{k}}}}{w_{ij_{\overline{k}}}ij_{\overline{k}}}, & \overline{k} = 1, \dots, \overline{\ell} \\ 0 & \overline{k} = \overline{\ell} + 1, \dots, \hat{K} \end{cases}$$
(14)

If for  $\overline{k} = 1, \ldots, \overline{\ell}$ ,  $x_{ij_{\overline{k}}} > u_{ij_{\overline{k}}}$ , redefine  $x_{ij_{\overline{k}}} = u_{ij_{\overline{k}}}$ , and transfer  $j_{\overline{k}}$  from  $\mathcal{K}$  to  $\mathcal{H}$ . Let  $\overline{\ell} = 1$  and go to Step 2. Otherwise row *i* is equilibrated so **stop**. **Step 4.** If  $\lambda_{\overline{l}} \notin [\tilde{c}_{ij_{\overline{l}}}, \tilde{c}_{ij_{\overline{l}}+1}]$ , let  $\overline{\ell} = \overline{\ell} + 1$ , and go to Step 2.

 $R^2$  is then defined by the composition  $R_1^2 \circ \ldots \circ R_m^2$ . Problem  $(RE_{diag}^i)$  is solved exactly through a single pass. This operator can be implemented on all rows *i* in parallel. Since  $R^2$  is in essence a sort and search algorithm its computational requirements will depend on the sorting routine used. It is important that any implementation of these operators

be done with great care; for some suggestions the reader might wish to consult Eydeland and Nagurney [17].

#### 3.4 Summary of Row Equilibration

To summarize, one can solve the relaxed problem (8) in two ways:

1. Repeated application of the operator  $R^1$  until the equilibrium conditions are satisfied.

2. Solution of a series of diagonal approximations to (8) using the exact operator  $R^2$  until the equilibrium conditions are met.

#### **3.5.** Column and cut-set operators

After the row equilibration (and consequent updating of  $y_1^k$  to  $y_1^{k+1}$  is complete), we then proceed to the column equilibration (to update  $y_2^k$  to  $y_2^{k+1}$ ) and then to the cut-set equilibration (to update  $y_3^k$  to  $y_3^{k+1}$ ). Analogous operators  $T^1, T^2$  and  $C^1, C^2$  exist for the cut-set and column equilibrium problems respectively. It is to be noted that these operators can be combined in any way to form the outer loop, i.e.  $R^1C^1T^1$ ,  $R^2C^1T^2$ ,  $R^1C^2T^2$  and  $R^2C^2T^2$  are all valid outer loop schemes.

#### 4. COMPUTATIONAL EXPERIMENTS

In this section we report on our computational experience with a wide variety of randomly generated constrained matrix problems using the decomposition schemes with embedded equilibration operators described in Section 3. We include comparisons of both types of equilibration with the algorithm given in Bachem and Korte [2]. All computer programs were coded in Fortran, and were compiled under VS FORTRAN at optimization level 3, running under VM/XA 5.5 on the IBM 3090-600E at the Cornell National Supercomputing Facility at Cornell University. The CPU times reported are exclusive of input and output times, but include initialization times.

The matrix Q in (1) was generated to be symmetric and strictly diagonally dominant, which ensured strict positive definiteness. In particular, each element of Q was generated in two stages. First, for each pair (k, l) a random number  $u \in U[0, 1]$  is generated. If  $u \in [0, density]$ , then  $Q_{kl}$  is selected to be *non-zero*, otherwise  $Q_{kl}$  is set to zero. If  $Q_{kl}$ is to be non-zero, then its value is generated in the second stage in such a manner that the resulting Q matrix will be strictly diagonally dominant. In the above-diagonal part of Q,  $Q_{kl}$  is generated randomly row by row in the range  $[.01, Q_{kl}^{max}]$ , where

$$Q_{kl}^{max} = .5 \times min[Q_{kk} - \sum_{j < l: j \neq k} Q_{kj}, Q_{ll} - \sum_{j < l} Q_{jl}]$$

.  $Q_{lk}$  in the lower triangular part is obtained from  $Q_{kl}$  in the upper triangular part.

Each element of the vector c in (1) was generated in the range [100, 1000]. The upper bounds  $u_{ij}$  were generated uniformly in the range [1, 100]. The lower bounds were set uniformly to zero. The row totals  $s_i$  of each row i were set equal to  $.1 \sum_j u_{ij}$ , and the column totals  $d_j$  were set equal to  $.1 \sum_i u_{ij}$ . In the equilibration codes, we set the initial vector  $y^0$  to be zero in all experiments, and all  $x_{ij}$  were initialized to  $d_j/m$ . The termination criteria were based on the relative residuals  $R(s_i) = (\sum_j x_{ij} - s_i)/s_i$ , and  $R(d_j) = (\sum_i x_{ij} - d_j)/d_j$ . If, following the solution of the row equilibration problems,  $R(d_j) \leq .001$  for all j, then the problem was considered solved; likewise, the problem was also considered solved if an analogous situation occurred after solution of the column equilibration problems. The sorting algorithm used in  $R^2$  and  $C^2$  was Straight Insertion Sort (see Press et al. [40], Cottle et al. [8]), with permutations not saved between iterations. Pointers were used for both the projection and equilibration methods, except for the runs on fully dense problems.

The algorithm of Bachem and Korte [2] (BK) is a derivative of Hildreth's quadratic programming procedure [24], and dualizes all the constraints, including the upper and lower bounds on variables. BK is an iterative algorithm, updating from one iterate to the next by a series of matrix multiplications (involving  $Q^{-1}$ ,  $D^{-1}$ , and the (possibly augmented) constraint matrix of the transportation type) and vector additions. We were unable to find any suitable library routines for the sparse inversion of the type required by BK (we did not wish to compare some appropriate assembly language routines such as those in ESSL with our FORTRAN codes).

The algorithm requires that the mn\*mn matrix Q be inverted, together with another much denser (m+n-1)\*(m+n-1) matrix D. The reader should note that the algorithm given in the paper assumed Q to be positive and diagonal in order to derive some elegant special structure algorithms; this assumption was not necessary for the validity of the algorithm, positive definiteness of Q is all that is required. The authors noted that (like the methods given in this paper) BK could also solve problems with additional constraints.

We report on the performance of two versions of BK. First, the specialized version whose convergence is proven by Theorem 1 in Bachem and Korte (1978), where Q is assumed to be a diagonal matrix. Second, a generalized version where Q is allowed to be any diagonally dominant symmetric matrix. Because Q is strongly diagonally dominant no principal pivoting is needed; both the Q matrix and the D matrix were inverted using a standard LU factorization. Early in the computational experimentation we also implemented a sparse matrix inversion routine, but found that the overhead of pointers was too costly at densities greater than 20 percent or so, and so did not report on this implementation here. For example, we observed an average 4-fold improvement in processing times when no pointers were used in problems with 10 rows and columns with densities greater than 30 percent. In both codes, we required the convergence of successive iterates (i.e.  $|x_{ij}^{i+1} - x_{ij}^{i}| \leq .001$ ) and the convergence of the relative residuals  $R(s_i)$  and  $R(d_j)$  to a tolerance of .001. These latter tolerances are the same as those required for the equilibration methods.

In the first two tables, we give results for diagonal problems, and for problems of increasing density of Q. Table 1 shows the results of the computational runs on problems without upper bounds. Table 2 gives the results of the computational runs on problems with upper bounds. Table 3 gives results for the projection method on larger problems.

Except in the diagonal case, our results would seem to suggest that the two equilibration algorithms presented in this paper are quite efficient in practice. As was to be expected, the CPU time required for BK for a given problem size was largely independent of problem density, since BK is based on matrix inversion. It is to be noted that the additional restriction of upper bounds did not seem to affect solution times for any of the algorithms very much. All runs of the equilibration and projection methods on problems that were not fully dense were done with codes using pointers, even though the overhead of the sparse machinery seems not be advantageous above 30-50 percent density. We included the runs with the sparse code here, so that the reader could see the tradeoff point for him/herself. The CPU time for the fully dense problem is an approximate upper bound on the practical run times for sparser problems, since the latter could easily have been run in dense form with zeroes stored explicitly.

The problem data, like any data, will have its limitations. We do not believe that test runs with a diagonally dominant Q gave unrepresentative results about the relative performance of these algorithms. Structurally, all algorithms tested in this paper did not care about the form of Q, only that it was positive definite. As mentioned earlier, the projection method combined with exact equilibration is, in fact, a parallel algorithm which can be applied to either the general or the diagonal case of the problem. The relative performance of BK and the equilibration algorithms on parallel architectures is an avenue of research that merits further consideration.

19

;

Dimension			Run Time (CPU sec.)		
of $Q$	# runs	Density	Equilibration	Projection	BK
100 x 100	10	Diagonal	.0238	.0104	.0044
	10	30	.6803	.0948	.7326
	10	50	1.2828	.1602	.7428
	10	70	1.9123	.2303	.7483
	10	100	.6434	.1270	.7725
400 x 400	10	Diagonal	.1376	.0479	.0186
	10	30	29.1673	1.4914	75.4334
	10	50	57.1891	<b>2.46</b> 18	75.5325
	10	70	90.5185	3.6123	75.3804
	10	100	36.6513	1.8373	78.9557
900 x 900	10	Diagonal	.5127	.1972	.0820
	2	30	249.3400	7.6269	1454.6160
	2	50	425.9000	12.8084	1455.1595
	2	70	631.9799	17.6258	1460.7494
	2	100	308.7723	9.5129	1458.3820

#### Table 1: Computational results for problems without upper bounds

¢

×

;

Dimension	on Run Time (CPU sec.		c.)		
of $Q$	# runs	Density	Equilibration	Projection	BK
100 x 100	0 x 100 10 Diagonal		.0307	.0171	.0066
	10	30	.8482	.1281	.7334
	10	50	1.5866	.2063	.7456
	10	70	<b>2.39</b> 40	.3016	.7481
	10	100	.7833	.0726	.7522
400 x 400	10	Diagonal	.1811	.1033	.0270
	10	<b>3</b> 0	35.9120	1.7467	75.2602
	10	50	68.9153	2.7732	75.3745
	10	70	109.4292	4.2474	75.3860
	10	100	44.7596	1.7156	75.5292
900 x 900	10	Diagonal	.5048	.2900	.0736
	2	30	289.2019	7.5886	1462.0900
	2	50	460.6929	11.9050	1456.1765
	2	70	666.5816	15.6453	145 <b>7.3</b> 540
	2	100	353.1348	9.7596	1460.2340

Table 2: Computational results for problems with upper bounds

ſ.

# Table 3: Computational results for the projection methodon fully dense larger problems

Dimension		
of $Q$	# runs	Run Time (CPU sec.)
1600 1600	-	0.0 7017
1000 X 1000	1	26.7217
$2500 \ge 2500$	1	71.4807
<b>3600 x 3600</b>	1	208.3290
4900 x 4900	1	428.8780
6400 x 6400	1	493.5415
8100 x 8100	1	809.3456
10000 x 10000	1	1305.5940
14400 x 14400	1	3000.5200

.

#### ACKNOWLEDGEMENTS

This work was supported in part by a Faculty Research Grant from the University of Massachusetts and in part by NSF Grant RII-880361 under the NSF VPW program while the first author was a visiting faculty member at the Massachusetts Institute of Technology. The cordiality and hospitality of both the Operations Research Center and the Center for Transportation Studies are warmly appreciated. The authors would like to thank Robert Sanz de Santamaria and Dae-shik Kim for assistance with the numerical results. The research was conducted at the Cornell National Supercomputer Facility, a resource of the Center for Theory and Simulation in Science and Engineering at Cornell University, which is funded in part by the National Science Foundation, New York State, and the IBM Corporation.

#### REFERENCES

- [1] Bacharach, M., Biproportional Scaling and Input-Output Change, Cambridge University Press, Cambridge, (1970).
- [2] Bachem, A. and B. Korte, "Algorithm for Quadratic Optimization over Transportation Polytopes," Zeitschrift für Angewandte Mathematik und Mechanik, 58, (1978) T459-T461.
- [3] Bachem, A. and B. Korte, "Minimum Norm Problems over Transportation Polytopes," *Linear Algebra and Its Applications*, **31**, (1980), 102–118.
- [4] Bingen, F., "Simplification de la demonstration d'un theoreme de M.W.M. Gorman," note, Universite Libre de Bruxelles, duplicated, 1965.
- [5] Byron, R.P., "The Estimation of Large Social Accounts Matrices," Journal of the Royal Statistical Society Series A, 141, (1978), 359-369.
- [6] Carey, M., C. Hendrickson, and K. Siddharthan, "A Method for Direct Estimation of Origin/Destination Trip Matrices," *Transportation Science*, 15, No. 1, (1981), 32-49.
- [7] Carey, M. and R. Revelli, "Constrained Estimation of Direct Demand Functions and Trip Matrices," *Transportation Science*, **20**, No. 3, (1986), 143-152.
- [8] Cottle, R.W., S.G. Duvall, and K. Zikan, "A Lagrangean Relaxation Algorithm for the Constrained Matrix Problem," Naval Research Logistics Quarterly, 33, (1986), 55-76.
- [9] Dafermos, S. "An Extended Traffic Assignment Model with Applications to Two-Way Traffic," *Transportation Science*, **5** (1971), 366-389.
- [10] Dafermos, S. "Integrated Equilibrium Flow Models for Transportation Planning," in *Traffic Equilibrium Methods*, Springer Lecture Notes in Economics and Mathematical Systems, No. 118, (1976), M. Florian, editor.

- [11] Dafermos, S. "An Iterative Scheme for Variational Inequalities," Mathematical Programming, 26, (1983), 40-47.
- [12] Dafermos, S. and A. Nagurney, "Supply and Demand Equilibration Algorithms for a Class of Market Equilibrium Problems," *Transportation Science*, 23 (1989), 118-124.
- [13] Dafermos, S. and F.T. Sparrow, "The Traffic Assignment Problem for a General Network," Journal of Research of the National Bureau of Standards, 73b, No. 2, (1969) 91-118.
- [14] Deming, W.E. and F.F. Stephan, "On a Least-Squares Adjustment of a Sampled Frequency Table when the Expected Marginal Totals are Known," Annals of Mathematical Statistics, 11, (1940), 427-444.
- [15] Erickson, J. "On Solving Linearly Constrained Maximum Entropy Problems," Mathematical Programming, 18, (1980), 146-154.
- [16] Erlander, S., K.O. Jornsten, and J.T. Lundgren, "On the Estimation of Trip Matrices in the Case of Missing and Uncertain Data," *Transportation Research*, 19b, No. 2, (1985), 123-141.
- [17] Eydeland, A. and A. Nagurney, "Progressive Equilibration Algorithms: the Case of Linear Transaction Costs," Computer Science in Economics and Management, (1989), in press.
- [18] Florian, M., "Nonlinear Cost Network Models in Transportation Analysis," in Mathematical Programming Study, 26, (1986), 167-196.
- [19] Friedlander, D. "A Technique for Estimating a Contingency Table Given the Marginal Total and Some Supplementary Data," Journal of the Royal Statistical Society, A124, (1961), 412-420.

25

.

- [20] Gorman, W.M. "Estimating trends in Leontieff Matrices,": a note on Mr. Bacharach's paper, Nuffield College, Oxford, duplicated, 1963.
- [21] Harrigan, F. and I. Buchanan, "Quadratic Programming Approach to Input-Output Estimation and Simulation," *Journal of Regional Science*, 24, No. 3, (1984), 339-358.
- [22] Herman, G.T. and A. Lent, "A Family of Iterative Quadratic Optimization Algorithms for Pairs of Inequalities, with Application in Diagnostic Radiology," *Mathematical Programming Study*, 9, (1978), 15-29.
- [23] Herman, G.T., A. Lent, and S.W. Rowland, "ART: Mathematics and Applications," J. of Theoretical Biology, 42, (1973), 1-32.
- [24] Hildreth, C. "A Quadratic Programming Procedure," Naval Research Logistics Quarterly, 4, (1957), 79-85.
- [25] Johnston, R.J., A.M. Hay, and P.J. Taylor, "Estimating the Sources of Spatial Change in Election Results: a Multiproportional Approach," *Environment and Planning A*, 14, (1982), 951-961.
- [26] Judge, G. and T.A. Yancey, Improved Methods of Inference in Econometrics, North-Holland, 1986.
- [27] Kruithoff, J. "Calculation of Telephone Traffic," De Ingenieur, 3, (1937), 21–26.
- [28] Lent, A. and Y. Censor, "Extensions of Hildreth's Row Action Method for Quadratic Programming," SIAM Journal of Control and Optimization, 18, (1980), 444-454.
- [29] McNeil, S. and C. Hendrickson, "A Note on Alternative Matrix Entry Estimation Techniques," *Transportation Research*, 19b, No. 6, 509-519.

:

- [30] Miller, R.E. and P.D. Blair, Input/Output Analysis: Foundations and Extensions, Prentice-Hall, Englewood Cliffs, N.J., 1985.
- [31] Morrison, W. and R.G. Thuman, "A Lagrangean Multiplier Approach to the Solution of a Special Constrained Matrix Problem," Journal of Regional Science. 20, (1980), 279-292.
- [32] Mosteller, F. and J.W. Tukey, Data Analysis and Regression, Addison-Wesley Publishing Co., Inc., 1977.
- [33] Nagurney, A. "Computational Comparisons of Spatial Price Equilibrium Methods," Journal of Regional Science, 27, (1987), 55-76.
- [34] Nagurney, A. "Import and Export Equilibration Algorithms for the Net Import Spatial Price Equilibrium Problem," *Journal of Cost Analysis*, (1989), in press.
- [35] Nagurney A., "An Algorithm for the Solution of a Quadratic Programming Problem with Application to Constrained Matrix and Spatial Price Equilibrium Problems," *Environment and Planning A*, 21 (1989), 99-114.
- [36] Ohuchi, A. and I. Kaji, "An Algorithm for the Hitchcock Transportation Problem with Quadratic Cost Functions," *Journal of the Operations Research Society* of Japan, 24, (1981), 170-181.
- [37] Ohuchi, A. and I. Kaji, "Lagrangean Dual Coordinatewise Maximization for Network Transportation Problems with Quadratic Costs," Networks, 14, (1984), 525-530.
- [38] Pang, J.S., "More Results on the Convergence of Iterative Methods for the Symmetric Linear Complementary Problem," Journal of Optimization Theory and Applications, Vol. 49, No. 1, (1986), 107-134.
- [39] Polenske, K., U.S. Multiregional Input-Output Accounts and Model, Lexington Books, Lexington, MA, 1980.

- [40] Press, W.H., B.F. Flannery, S.A. Teukolsky, and W.T. Vetterling, Numerical Recipes, Cambridge University Press, 1986.
- [41] Snickars, F. and J.W. Weibull, "A Minimum Information Principle, Theory and Practice," *Regional Science and Urban Economics*, 7, (1977) 137-168.
- [42] Stone R., "Multiple classifications in Social Accounting," Bulletin de l'Institut International de Statistique, **39**, (1962) 215-233.
- [43] Van der Ploeg, F., "Reliability and Adjustment of Sequences of Large Economic Accounting Matrices," Journal of the Royal Statistical Society A, 145, (1982), 169-194.
- [44] Van der Ploeg, F., "Balancing Large Systems of National Accounts," Computer Science in Economics and Management 1, (1988), 31-39.
- [45] Van der Sluis, A., "Condition Numbers and Equilibration of Matrices," Numerische Mathematik, 14, (1969), 14-23.
- [46] Wilson, A.G. Entropy in Urban and Regional Modelling, (1970), Pion Ltd., London.
- [47] Zenios, S.A., A. Drud, and J.M. Mulvey, "Balancing Large Social Accounting Matrices with Nonlinear Network Programming," *Networks*, (1989), to appear.

28