January 1995

Linearly-Constrained Entropy Maximization Problem with Quadratic Costs and Its Applications to Transportation Planning Problems

Shu-Cherng Fang
North Carolina State University at Raleigh

Jacob Tsao
University of California - Berkeley, jacob.tsao@sjsu.edu

Linearly-Constrained Entropy Maximization Problem with Quadratic Cost and Its Applications to Transportation Planning Problems[1]

S. C. FANG

North Carolina State University, Operations Research Program and Industrial Engineering Department, Raleigh, North Carolina 27695

H.-S. J. TSAO

University of California, Institute of Transportation Studies, Berkeley, California 94720

Many transportation problems can be formulated as a linearly-constrained convex programming problem whose objective function consists of entropy functions and other cost-related terms. In this paper, we propose an unconstrained convex programming dual approach to solving these problems. In particular, we focus on a class of linearly-constrained entropy maximization problem with quadratic cost, study its Lagrangian dual, and provide a globally convergent algorithm with a quadratic rate of convergence. The theory and algorithm can be readily applied to the trip distribution problem with quadratic cost and many other entropy-based formulations, including the conventional trip distribution problem with linear cost, the entropy-based modal split model, and the decomposed problems of the combined problem of trip distribution and assignment. The efficiency and the robustness of this approach are confirmed by our computational experience.

Many transportation problems can be formulated as a linearly-constrained convex programming problem whose objective function consists of entropy functions and other cost-related terms.[3, 6, 9, 15, 16, 21, 22, 23] To study these transportation planning problems, we first consider the following mathematical optimization model:

Program P_μ:

$$\text{Minimize} \quad f_\mu(\mathbf{x}) \equiv \mu \sum_{j=1}^{n} x_j \ln x_j + \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{D}\mathbf{x}$$

$$\text{subject to:} \quad \mathbf{Ax} = \mathbf{b}, \qquad (1.a)$$

$$\mathbf{x} \geqslant 0, \qquad (1.b)$$

where $\mu > 0$, $\mathbf{x} \in R^n$, $\mathbf{c} \in R^n$, $\mathbf{b} \in R^m$, $\mathbf{D}$ is an $n \times n$ diagonal matrix with diagonal elements $d_j \geqslant 0$, for $j = 1, \ldots, n$, $\mathbf{A}$ is an $m \times n$ $(m \leqslant n)$ matrix, and 0 is the n-dimensional zero vector.

Note that when $\mu = 1$, $\mathbf{c} = 0$, $\mathbf{D} = 0$, and $\mathbf{A}$ denotes a trip matrix, Program P_μ represents the basic entropy-based trip distribution model; when only $\mathbf{D} = \mathbf{0}$, it becomes the trip distribution model with linear cost. Furthermore, when $\mathbf{c} = 0$, $\mathbf{D} = 0$, and $\mathbf{A}$ represents the modal trip matrix and associated cost-weighing constraints, Program P_μ reduces to an entropy-based modal split model.[16] Therefore Program P_μ not only encompasses many existing transportation planning models but also extends them.

Tomlin[22] pointed out that the linear cost (e.g., travel time) per unit flow from an origin to a destination in a trip distribution problem is an oversimplification. Assuming a convex quadratic cost that

[1] Accepted by Mark S. Daskin.

0041-1655/95/2904-0353 $01.25

reflects the cost increase due to congestion in Program P_μ should be a better approximation of the actual cost. The diagonal matrix **D** in the quadratic term of the objective function accommodates the possibility of difference in quadratic approximations for the n origin/destination pairs. Strongly motivated by the trip distribution problem with quadratic cost, we shall focus on studying the more general setting described by Program P_μ.

Although the optimization model P_μ is more realistic and desirable, the task of designing an efficient algorithm is challenging. While relatively little theory has been known about the general model, various algorithms have been proposed for solving an important special case of this model, namely entropy maximization with linear cost subject to linear constraints. Better known algorithms include MART (multiplicative algebraic reconstruction technique),[14, 18] Bregman's method,[17] Jacobi method, Gauss-Seidel method,[5] and Newton's method.[6, 8] None of them possesses the property of being globally convergent with a quadratic rate of convergence. In fact, for some of these methods,[5] even the number of iterations required to reach the optimum could be very sensitive to the individuality and size of the problem. This presents a potential problem for solving real-life large-scale applications.

In this paper, we take a dual approach and design an efficient and robust computational algorithm for solving not only the entropy maximization problem with linear constraints but also the more general model of Program P_μ. With this approach, the solution methods for several important classes of models can be treated in a *unified* manner.

The rest of the paper is organized as follows. In Section 1, we discuss some basic ideas which motivated model P_μ, via a specific example of trip distribution problem with quadratic cost. In Section 2, we derive the Lagrangian dual D_μ for Program P_μ, which turns out to be a convex optimization problem without any explicit constraints. Moreover, under some typical and mild conditions, it is shown that no duality gap exists between P_μ and D_μ. We also provide a simple conversion formula for obtaining a primal optimal solution from a corresponding dual optimal solution. Section 3 briefly introduces the curved search method and customizes it to obtain a quadratically convergent global algorithm for solving Program D_μ and hence Program P_μ. In Section 4, the dual approach developed in Section 2 is specialized for the specific problem of trip distribution with quadratic cost. Some computational results on the trip distribution problem are reported in Section 5. Section 6 concludes this paper.

1. A LINEARLY-CONSTRAINED ENTROPY MAXIMIZATION MODEL WITH QUADRATIC COST

FOR SIMPLICITY, we assume that all mathematical programs stated in this section have a unique optimal solution that satisfies the Kuhn-Tucker condition. To motivate the study of model P_μ, we consider a trip distribution problem with a quadratic cost structure. Suppose that there are K origins and L destinations. Let x_{ij} denote the number of trips from origin i to destination j. The problem can be represented by:

Program Q:

$$\text{Minimize} \quad \sum_{i=1}^{K}\sum_{j=1}^{L} x_{ij} ln x_{ij} + \alpha \sum_{i=1}^{K}\sum_{j=1}^{L} c_{ij}\, x_{ij} + \frac{\beta}{2}\sum_{i=1}^{K}\sum_{j=1}^{L} d_{ij} x_{ij}^2$$

subject to:

$$\sum_{j=1}^{L} x_{ij} = O_i, \quad i = 1,2,\ldots,K, \tag{2}$$

$$\sum_{i=1}^{K} x_{ij} = D_j, \quad j = 1,2,\ldots,L, \tag{3}$$

$$x_{ij} \geqslant 0, \quad i = 1,2,\ldots,K, \quad \text{and} \quad j = 1,2,\ldots,L. \tag{4}$$

Note that, without the quadratic cost terms, Program Q is equivalent to the well-known gravity model. In this special case, the optimal solution has the following form:

$$x_{ij} = cr_i e^{-\alpha c_{ij}} s_j, \tag{5}$$

where $r_i = e^{\lambda_i}$, $s_j = e^{\mu_j}$, $c = e^{-1}$, and λ_i and μ_j are the Lagrange multipliers associated with the constraints. The exponential term is a decreasing function of c_{ij}, which may represent distance, cost or some notion of generalized cost. This exponential term is referred to as the "exponential deterrence" in the literature, which is different from the "quadratic deterrence" in the Newtonian gravity law.

With the quadratic terms, the model is no longer a classic gravity model. Instead, its optimal solution, as can be seen later, satisfies the following equation:

$$x_{ij} = cr_i e^{-\alpha c_{ij} - \beta d_{ij} x_{ij}} s_j. \tag{6}$$

This differs from Equation (5) of the gravity model in that x_{ij} appears in the exponent in the right-hand-side expression and the optimal solution is no longer explicitly solvable as a function of the program parameters. When all βd_{ij}'s are very small, Program Q produces approximately the gravity model. However, the quadratic terms have the following "self-deterrence" or "self-regulation" effect on the optimal solution. By increasing the value of one particular βd_{ij} but keeping all other parameters the same, the corresponding optimal trip number x_{ij} decreases. Moreover, for any particular positive value of βd_{ij}, the larger the optimal x_{ij} of the gravity model is, the larger the decrease to the optimal x_{ij} of the new model is. This characteristic can be used to model deterrence due to congestion in the stage of trip distribution. Moreover, solving the combined trip distribution and assignment problem often involves a problem decomposition where the trip distribution is a component which is required to be solved repeatedly. This self-regulation feature of Program Q could benefit the convergence rate due to its ability to reflect, at the stage of distribution, the congestion cost obtained in the assignment stage. This possibility is being studied by the authors. For ease of discussion, we will refer to this new model as the "self-deterrent gravity model" in the rest of this paper. We keep the notion of gravity in the name because the optimal solution x_{ij} is still a decreasing function of c_{ij}.

We now offer an alternative motivation for studying Program Q. It is well-known that the gravity model can be derived via several other approaches. In particular, the optimal solution of the following mathematical program is of the gravity form.

$$\text{Program R:} \quad \text{Minimize} \quad \sum_{i=1}^{K}\sum_{j=1}^{L} x_{ij} ln x_{ij}$$

$$\begin{aligned} \text{subject to:} \quad & \sum_{j=1}^{L} x_{ij} = O_i, \quad i = 1,2,\ldots,K, \\ & \sum_{i=1}^{K} x_{ij} = D_j, \quad j = 1,2,\ldots,L, \\ & \sum_{i=1}^{K}\sum_{j=1}^{L} c_{ij}x_{ij} = C, \\ & x_{ij} \geqslant 0, \quad i = 1,2,\ldots,K, \\ & \text{and} \quad j = 1,2,\ldots,L. \end{aligned}$$

Similarly, the self-deterrent gravity model defined by Equation (6) can be derived as the optimal solution of the following mathematical program:

$$\text{Program S:} \quad \text{Minimize} \quad \sum_{i=1}^{K}\sum_{j=1}^{L} x_{ij} ln x_{ij}$$

$$\begin{aligned} \text{subject to:} \quad & \sum_{j=1}^{L} x_{ij} = O_i, \quad i = 1,2,\ldots,K, \\ & \sum_{i=1}^{K} x_{ij} = D_j, \quad j = 1,2,\ldots,L, \\ & \sum_{i=1}^{K}\sum_{j=1}^{L} c_{ij}x_{ij} = C, \\ & \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{L} d_{ij}x_{ij}^2 = D, \\ & x_{ij} \geqslant 0, \quad i = 1,2,\ldots,K, \\ & \text{and} \quad j = 1,2,\ldots,L. \end{aligned}$$

It is easy to verify that when the quadratic constraint in Program S is replaced by a slightly generalized constraint

$$\frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{L} d_{ij}x_{ij}^2 + \sum_{i=1}^{K}\sum_{j=1}^{L} f_{ij}x_{ij} = D, \tag{7}$$

the self-deterrent gravity model has the following form:

$$x_{ij} = cr_i e^{-(\alpha c_{ij} + \beta f_{ij}) - \beta d_{ij}x_{ij}} s_j. \tag{8}$$

Note that when the parameter D is not known with confidence or the equality constraint is too strong, we may consider a corresponding inequality constraint. Also note that, as pointed out by WILSON[24] about parameter α, the parameter β of the self-deterrent gravity model does not have to be known exactly at the beginning. It could be found in the calibration process.

A particular application of Program S is as follows. In formulating and solving trip distribution problems, the planner often has information about the past trip distributions. This information often leads to certain "target values" for the current planning cycle. There are different ways to model such target values. One way is to use the target values as the "prior" and formulate a cross-entropy minimization problem, i.e., minimization of Kullback's discrimination information measure. Another way is to impose a constraint on a weighted sum of the squared deviation of the distribution x_{ij}

from the target values p_{ij}, i.e.,

$$\sum_{i=1}^{K}\sum_{j=1}^{L} w_{ij}(x_{ij}-p_{ij})^2 = T, \qquad (9.\text{a})$$

or, equivalently,

$$\frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{L} d_{ij}x_{ij}^2 + \sum_{i=1}^{K}\sum_{j=1}^{L} f_{ij}x_{ij} = D, \qquad (9.\text{b})$$

where $d_{ij} = 2w_{ij}$, $f_{ij} = -2w_{ij}p_{ij}$ and $D = T - \sum_{i=1}^{K}\sum_{j=1}^{L} w_{ij}p_{ij}^2$. $\sum_{i=1}^{K}\sum_{j=1}^{L} w_{ij}(x_{ij}-p_{ij})^2$ has been used as a measure for the "distance" between x_{ij} and p_{ij} [e.g., 4].

Gravity models with additional constraints other than the production, attraction, and cost have been proposed and studied.[10] So far all those additional constraints have been considered in linear form. The proposed quadratic cost term (or the quadratic constraint) provides a new dimension of possible models for transportation planning. The additional parameter β of the self-deterrent gravity model provides one more degree of freedom in model identification and calibration. In addition, as will be shown later in this paper, the new model can be solved efficiently, in both theory and practice.

2. AN UNCONSTRAINED DUAL APPROACH

We first provide the condition under which Program P_μ achieves a finite minimum at a unique point $\mathbf{x}^*(\mu) \geq 0$, for each $\mu > 0$.

LEMMA 1. *If Program P_μ has a feasible solution, then it achieves a finite minimum at a unique point $\mathbf{x}^*(\mu) \in R^n$ for each $\mu > 0$.*

Proof. With the convention of $0ln0 = 0$, the objective function $f_\mu(\mathbf{x})$ is continuous on its entire domain $\{\mathbf{x}|\mathbf{x} \geq 0\}$. For each $j = 1, \ldots, n$, if $x_j = 0$, $f_{\mu,j}(x_j) \equiv \mu x_j ln x_j + c_j x_j + \frac{1}{2}d_j x_j^2 = 0$. It is obvious that $f_{\mu,j}(x_j) \to \infty$ as $x_j \to \infty$. Therefore, $f_\mu(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$ in the effective domain $\{\mathbf{x}|\mathbf{x} \geq 0\}$ of f_μ and hence all level sets $\{\mathbf{x}|f_\mu(\mathbf{x}) \leq \alpha\}$, $\alpha \in R$, of f_μ are bounded. The continuity of the objective function implies that these level sets are also closed. Therefore, all level sets are compact. Clearly, the intersection of any of these level sets and the feasible region $\Omega \equiv \{\mathbf{x}|\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ is also compact. By the assumed feasibility of P_μ, there exists one level set L_μ^0 such that $L_\mu^0 \cap \Omega$ is non-empty. Since this intersection is compact and $f_\mu(\mathbf{x})$ is continuous on it, by the well-known fact that any continuous function on a compact set always attains its optimum (Theorem of Weierstrass), Program P_μ attains its finite minimum. Finally, the fact that $f_{\mu,j}(x_j)$ is strictly convex over the domain $[0, \infty)$ for each j, f_μ is strictly convex. This and the convexity of Ω imply that P_μ achieves the finite minimum at a unique point $\mathbf{x}^*(\mu) \geq 0$, for each $\mu > 0$. □

Like all interior-point methods, we assume throughout this paper that Program P_μ has an interior feasible solution $\mathbf{x} > 0$. We now derive the Lagrangian dual D_μ of P_μ. By defining the Lagrangian

$$L(\mathbf{x},\mathbf{w}) = \mu\sum_{j=1}^{n} x_j ln x_j + \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\sum_{j=1}^{n} d_j x_j^2 - \sum_{i=1}^{m} w_i\left(\sum_{j=1}^{n} a_{ij}x_j - b_i\right),$$

where $w_i \in R$, $i = 1, 2, \ldots, m$, the Lagrangian dual program becomes

$$\text{Program } D_\mu^1: \quad \max_{\mathbf{w}\in R^m} \min_{\mathbf{x}\geq 0} L(\mathbf{x},\mathbf{w}).$$

Setting the partial derivative of $L(\mathbf{x},\mathbf{w})$ with respect to x_j to zero gives us

$$\mu + \mu ln x_j + c_j + d_j x_j - \sum_{i=1}^{m} a_{ij}w_i = 0. \qquad (10)$$

LEMMA 2. *Equation* (10) *defines an implicit function $x_j \equiv h_j(\mathbf{w}) > 0$ for all $\mathbf{w} \in R^m$. Moreover, h_j is continuously differentiable.*

Proof. For $x_j > 0$ and $\mathbf{w} \in R^m$, define

$$F_j(x_j,\mathbf{w}) \equiv d_j x_j + \mu ln x_j - \sum_{i=1}^{m} a_{ij}w_i + c_j + \mu. \qquad (11)$$

Apparently, F_j is continuously differentiable over its effective domain. Since, in addition, $\partial F_j/\partial x_j \neq 0$ throughout its domain, by the Implicit Function Theorem [1, p. 147], Equation (11) indeed defines an implicit function $x_j \equiv h_j(\mathbf{w})$ for all $\mathbf{w} \in R^m$. Moreover, by the same theorem, the implicit function is continuously differentiable over its entire domain. □

The strict convexity of $f_\mu(\mathbf{x})$ and the linearity of the equality constraints (1.a) of Program P_μ imply that, given any $\mathbf{w} \in R^m$, $L(\mathbf{x},\mathbf{w})$ is strictly convex in $\mathbf{x}$. Therefore, $h_j(\mathbf{w}) > 0$ indeed defines the minimal solution of $\min_{\mathbf{x}\geq 0} L(\mathbf{x},\mathbf{w})$ for each $\mathbf{w}$. Consequently, the dual Program D_μ^1 is equivalent to

$$\text{Program } D_\mu^2: \quad \max_{\mathbf{w}\in R^m} L(\mathbf{x},\mathbf{w}), \quad \text{where } x_j \equiv h_j(\mathbf{w}).$$

Multiplying both sides of Equation (10) by x_j and then summing both sides over j give

$$\mu \sum_{j=1}^{n} x_j ln x_j + \sum_{j=1}^{n} c_j x_j + \frac{1}{2} \sum_{j=1}^{n} d_j x_j^2 - \sum_{i=1}^{m} \sum_{j=1}^{n} w_i a_{ij} x_j = -\frac{1}{2} \sum_{j=1}^{n} d_j x_j^2 - \mu \sum_{j=1}^{n} x_j.$$

Therefore, Program D_μ^2 is equivalent to

Program D_μ:

$$\max_{\mathbf{w} \in R^m} d_\mu(\mathbf{w}) \equiv -\frac{1}{2} \sum_{j=1}^{n} d_j h_j^2(\mathbf{w}) - \mu \sum_{j=1}^{n} h_j(\mathbf{w}) + \sum_{i=1}^{m} b_i w_i,$$

where, for any $\mathbf{w} \in R^m$, $h_j(\mathbf{w}) > 0$ is defined as the unique solution of the following equation:

$$d_j h_j(\mathbf{w}) + \mu ln h_j(\mathbf{w}) = \sum_{i=1}^{m} a_{ij} w_i - c_j - \mu. \quad (12)$$

Note that, treating $h_j(\mathbf{w})$ as a variable, Equation (12) is a single-variable equation which could be solved by a straightforward one-dimensional search. Also note that, in the absence of the quadratic term in the primal objective function, $h_j(\mathbf{w})$ can be solved explicitly and be expressed as a gravity model in the form of Equation (5).

We devote the rest of this section to establishing duality theorems.

THEOREM 1. (*Weak Duality Theorem*) *Suppose Program P_μ has a feasible solution. Then, $Min(P_\mu) \geqslant Sup(D_\mu)$.*

Proof. This theorem follows immediately from the Lagrangian duality theory. □

THEOREM 2. *Given that $\mathbf{w}^* \in R^m$ and $\mathbf{x}^* \in R^n$ such that $\mathbf{Ax}^* = \mathbf{b}$ and $\mathbf{x}^* > 0$, if*

$$d_j x_j^* + \mu ln x_j^* = \sum_{i=1}^{m} a_{ij} w_i^* - c_j - \mu, \quad (12')$$

then $\mathbf{x}^*$ is an optimal solution to Program P_μ and $\mathbf{w}^*$ is an optimal solution to Program D_μ. Moreover, $Min(P_\mu) = Max(D_\mu)$.

Proof. Multiplying both sides of Equation (12′) by x_j^* and then summing over j establish the equality of the two objective function values. Given the feasibility of $\mathbf{w}^*$ and $\mathbf{x}^*$, this theorem follows from Theorem 1. □

LEMMA 3. *The dual objective function $d_\mu(\mathbf{w})$ is continuously twice differentiable.*

Proof. By Lemma 2, $h_j(\mathbf{w})$ is continuously differentiable. Therefore, $d_\mu(\mathbf{w})$ is also continuously differentiable and the k-th element of its gradient vector is given by

$$\frac{\partial d_\mu(\mathbf{w})}{\partial w_k} = -\frac{1}{2} \sum_{j=1}^{n} 2 d_j h_j(\mathbf{w}) \frac{\partial h_j(\mathbf{w})}{\partial w_k} - \mu \sum_{j=1}^{n} \frac{\partial h_j(\mathbf{w})}{\partial w_k} + b_k, \quad (13)$$

where $\partial h_j(\mathbf{w}) / \partial w_k$ can be obtained as follows. First denote $h_j(\mathbf{w})$ by x_j and define

$$F_j(x_j, \mathbf{w}) \equiv d_j x_j + \mu ln x_j - \sum_{i=1}^{m} a_{ij} w_i + c_j + \mu. \quad (14)$$

Then, by Equation (12) and the chain rule, we have

$$\frac{\partial F_j}{\partial x_j} \frac{\partial x_j}{\partial w_k} + \frac{\partial F_j}{\partial w_k} = 0, \quad (15)$$

or equivalently,

$$\left(d_j + \frac{\mu}{x_j} \right) \frac{\partial h_j(\mathbf{w})}{\partial w_k} - a_{kj} = 0. \quad (16)$$

In other words,

$$\frac{\partial h_j(\mathbf{w})}{\partial w_k} = \frac{a_{kj}}{d_j + \frac{\mu}{x_j}} = \frac{a_{kj} h_j(\mathbf{w})}{d_j h_j(\mathbf{w}) + \mu}. \quad (17)$$

Plugging (17) into equation (13), we finally have

$$\begin{aligned} \frac{\partial d_\mu(\mathbf{w})}{\partial w_k} &= -\frac{1}{2} \sum_{j=1}^{n} 2 d_j h_j(\mathbf{w}) \frac{a_{kj} h_j(\mathbf{w})}{d_j h_j(\mathbf{w}) + \mu} - \mu \sum_{j=1}^{n} \frac{a_{kj} h_j(\mathbf{w})}{d_j h_j(\mathbf{w}) + \mu} + b_k \\ &= -\sum_{j=1}^{n} a_{kj} \left(\frac{1}{d_j h_j(\mathbf{w}) + \mu} \right) \times \left[h_j(\mathbf{w}) \left(d_j h_j(\mathbf{w}) + \mu \right) \right] + b_k \\ &= -\sum_{j=1}^{n} a_{kj} h_j(\mathbf{w}) + b_k. \end{aligned} \quad (18)$$

Since $h_j(\mathbf{w})$ is continuously differentiable, $d_\mu(\mathbf{w})$ is continuously twice differentiable. □

THEOREM 3. *Program D_μ has a concave objective function $d_\mu(\mathbf{w})$. If the constraint matrix $\mathbf{A}$ in Program P_μ has full row-rank, then $d_\mu(\mathbf{w})$ is strictly concave.*

Proof. By Lemma 3, $d_\mu(\mathbf{w})$ is continuously twice differentiable. The (k_1, k_2)-th element of the Hessian Matrix is simply

$$\begin{aligned}\frac{\partial^2 d_\mu(\mathbf{w})}{\partial w_{k_1} \partial w_{k_2}} &= \frac{\partial}{\partial w_{k_2}}\left[-\sum_{j=1}^{n} a_{k_1 j} h_j(\mathbf{w}) + b_{k_1}\right] \\ &= \sum_{j=1}^{n} -a_{k_1 j}\left[\frac{\partial}{\partial w_{k_2}} h_j(\mathbf{w})\right] \\ &= \sum_{j=1}^{n} -a_{k_1 j}\frac{a_{k_2 j} h_j(\mathbf{w})}{d_j h_j(\mathbf{w}) + \mu} \\ &= \sum_{j=1}^{n}\left(-\frac{h_j(\mathbf{w})}{d_j h_j(\mathbf{w}) + \mu}\right) a_{k_1 j} a_{k_2 j}.\end{aligned} \tag{19}$$

Since $h_j(\mathbf{w}) > 0$, $d_j \geqslant 0$, and $\mu > 0$, we have $-h_j(\mathbf{w})/(d_j h_j(\mathbf{w}) + \mu) < 0$. Consequently, the Hessian matrix can be written as $\mathbf{A}\mathbf{D}_r(\mathbf{w})\mathbf{A}^T$, where $\mathbf{D}_r(\mathbf{w})$ is an $n \times n$ diagonal matrix with negative diagonal elements $r_j(\mathbf{w}) = -h_j(\mathbf{w})/(d_j h_j(\mathbf{w}) + \mu)$. (Since $\mu > 0$, $d_j \geqslant 0$ and $h_j(\mathbf{w}) > 0$, $r_j(\mathbf{w})$ is well defined for all $\mathbf{w} \in R^m$.) Therefore, $d_\mu(\mathbf{w})$ is concave. (In fact, concavity follows immediately from the Lagrangian duality theory.) By matrix theory, the Hessian matrix must be nonsingular and negative definite as long as $\mathbf{A}$ has full row-rank. Therefore, $d_\mu(\mathbf{w})$ is strictly concave if $\mathbf{A}$ has full row-rank. □

THEOREM 4. *Given $\mu > 0$, if Program P_μ has an interior feasible solution then Program D_μ attains a finite maximum and Min($\mathbf{P}_\mu$) = Max(d_μ). If, in addition, its constraint matrix* $\mathbf{A}$ *has full row-rank, then Program D_μ has a unique optimal solution $\mathbf{w}^*(\mu) \in R^m$. In either case, formula (12′) provides a dual-to-primal conversion that defines the optimal solution $\mathbf{x}^*(\mu)$ of Program P_μ.*

Proof. By Lemma 1, Program P_μ attains its minimum at a unique optimal solution. By the Interior-Point Assumption and the Lagrangian duality theory [e.g., 20, p. 154], there is no duality gap between Programs P_μ and D_μ and there exists a $\mathbf{w}$ at which Program D_μ attains its finite maximum. Since $d_\mu(\mathbf{w})$ is differentiable over R^m, the first order optimality conditions hold at any optimal solution $\mathbf{w}^*(\mu)$. By setting $\nabla d_\mu(\mathbf{w}^*(\mu)) = 0$, the conditions are

$$\sum_{j=1}^{n} a_{kj} h_j(\mathbf{w}^*(\mu)) = b_k, \qquad \text{for } k = 1, 2, \ldots, m. \tag{20}$$

By defining $x_j^*(\mu) = h_j(\mathbf{w}^*(\mu)) > 0$ according to (12′), Equation (20) becomes $\mathbf{A}\mathbf{x}^*(\mu) = \mathbf{b}$. The optimality of $x_j^*(\mu)$ follows from Theorem 2. The matrix $\mathbf{A}$ having full row-rank implies the strict concavity of d_μ and hence the uniqueness of the optimal solution $\mathbf{w}^*(\mu)$. □

Based on this strong duality theorem, we can solve the constrained Program P_μ by first solving the unconstrained Program D_μ and then obtaining a primal optimal solution via the conversion equation. This unique setting allows us to exploit various unconstrained convex optimization methods for solving the general model. Different methods of course lead to different performance. Later in this paper, we will customize the "curved-search algorithm",[2] a global algorithm with a quadratic rate of convergence, for solving the Lagrangian dual of the trip distribution problem with quadratic cost, which is a special case of Program P_μ.

3. A QUADRATICALLY CONVERGENT GLOBAL ALGORITHM

IN THIS SECTION, we outline a computational procedure for solving P_μ, briefly describe the curved-search method, and prove the convergence properties.

The computational procedure is as follows:

Step 1: Select an arbitrary initial dual solution $\mathbf{w}_0$.
Step 2: Determine $h_j(\mathbf{w})$ according to equation (12) and use unconstrained convex optimization techniques to find an optimal solution $\mathbf{w}^*(\mu)$ of Program D_μ.
Step 3: Compute the optimal solution $\mathbf{x}^*(\mu)$ of Program P_μ according to equation (12′).

It is clear that Step 2 accounts for the major computation effort. In this paper, we report our computational experience with the "curved-search method."[2] Instead of solving Program D_μ directly, we solve the following equivalent convex minimization problem:

$$\text{Program } \mathrm{D}'_\mu: \quad \min_{\mathbf{w} \in R^m} -d_\mu(\mathbf{w}).$$

Most classical iterative methods improve a current solution by moving along a straight line. The basic idea of the curved-search method is to improve a current solution by moving along a quadratic curve which is determined by minimizing a certain model of the objective function subject to suitable constraints. BEN-TAL ET AL.[2] showed that the quadratic curve used in their method turns out to be a nonlinear combination of the "signed" Newton direction and the steepest descent direction.

More precisely, for an unconstrained convex minimization problem with a twice continuously differentiable objective function f, their curved-search method moves from one solution $\mathbf{w}_k$ to the next solution $\mathbf{w}_{k+1}$ along the quadratic curve (in variable t)

$$\mathbf{q}_k(t) = \mathbf{w}_k + t\mathbf{d}_k + \frac{1}{2}t^2\mathbf{z}_k \quad (\text{for } t \geqslant 0),$$

where

$$\mathbf{d}_k = -\beta_k \frac{|\nabla f(\mathbf{w}_k)|^2}{\nabla f^T(\mathbf{w}_k)\left[\nabla^2 f(\mathbf{w}_k)\right]^{-1}\nabla f(\mathbf{w}_k)} \times \left[\nabla^2 f(\mathbf{w}_k)\right]^{-1}\nabla f(\mathbf{w}_k) \quad \text{and}$$

$$\mathbf{z}_k = -\alpha_k |\nabla f(\mathbf{w}_k)| \nabla f(\mathbf{w}_k).$$

By searching for an appropriate step-length t_k such that

$$t_k \in \arg\min_{t>0} f\left(\mathbf{w}_k + t\mathbf{d}_k + \frac{1}{2}t^2\mathbf{z}_k\right), \tag{21}$$

a new solution $\mathbf{w}_{k+1}$ is defined by

$$\mathbf{w}_{k+1} = \mathbf{w}_k + t_k\mathbf{d}_k + \frac{1}{2}t_k^2\mathbf{z}_k. \tag{22}$$

Given a small $\varepsilon \geqslant 0$, the optimality is considered reached if $|\nabla f(\mathbf{w}_{k+1})| \leqslant \varepsilon$. Notice that α_k and β_k are adjustable positive parameters associated with the constraints with which the above-mentioned model of the objective function is minimized. Therefore, Equations (21) and (22) actually define a family of curved-search algorithms. The performance of this family of algorithms can be fine-tuned by choosing appropriate values of α_k and β_k. Also notice that when $\alpha_k \equiv 0$, the curved-search algorithm becomes the (signed) Newton method. When $\beta_k \equiv 0$, it becomes the steepest descent algorithm.

Note that, in applying this curved search algorithm to solving Program D'_μ, the presence of the implicit functions h_j, $j = 1, 2, \ldots, n$, actually does not require any additional care except the need for a simple line search routine to solve for $x_j = h_j(\mathbf{w})$. Since the dual objective function, the gradient vector and the Hessian matrix can all be expressed in terms of h_j's, calling the simple line search routine prior to their calculation suffices.

We now establish the property of global convergence and the quadratic rate of local convergence.

THEOREM 5. *Suppose that the constraint matrix* $\mathbf{A}$ *has full row-rank and Program* P_μ *is strictly feasible. Then, the curved-search algorithm either stops after finitely many steps or generates an infinite sequence* $\{\mathbf{w}_k \in R^m | k = 1, 2, \ldots\}$ *such that*

(i) $d_\mu(\mathbf{w}_{k+1}) > d_\mu(\mathbf{w}_k)$ *for all* k;

(ii) $\{\mathbf{w}_k \in R^m | k \in N\}$ *has at least a cluster point* $\mathbf{w}^c$ *in the level set* $L_0 \equiv \{\mathbf{w} \in R^m | d_\mu(\mathbf{w}) \geqslant d_\mu(\mathbf{w}_0)\}$;

(iii) *for each such cluster point* $\mathbf{w}^c$, $|\nabla d_\mu(\mathbf{w}^c)| \leqslant \varepsilon$.

Proof. Lemma 3 establishes that d_μ is twice continuously differentiable. If we can further show that the level set L_0 is compact, then Theorem 5 is a direct consequence of Theorem 3.1 of [2]. As shown in the proof for Theorem 4, the level set of the dual objective associated with the maximum contains only the optimal solution $\mathbf{w}^*(\mu)$. Since $-d_\mu$ is a convex function, Corollary 8.7.1 of [19] directly implies that all level sets are bounded. The continuity of d_μ guarantees the closedness of the level sets. Therefore, all level sets, including L_0, are compact. □

Note that, under the assumptions of Theorem 5, since D_μ has a unique optimal solution $\mathbf{w}^*(\mu)$ satisfying the first order optimal condition, the sequence generated by the algorithm, with $\varepsilon = 0$, converges to the unique cluster point $\mathbf{w}^c = \mathbf{w}^*(\mu)$. Moreover, this infinite sequence has a quadratic rate of convergence.

THEOREM 6. *Under the assumptions of Theorem* 5, *suppose that the sequence* $\{\mathbf{w}_k \in R^m | k \in N\}$ *generated by the curved-search algorithm converges to* $\mathbf{w}^*(\mu)$. *The rate of convergence is quadratic.*

Proof. We establish the quadratic rate of convergence by showing that the following four conditions of Theorem 4.4 of [2] are satisfied: (i) $\{\mathbf{w}_k \in R^m | k \in N\}$ converges to $\mathbf{w}^*(\mu)$, (ii) $\nabla d_\mu(\mathbf{w}^*(\mu)) = 0$, (iii) $\nabla^2 d_\mu(\mathbf{w}^*(\mu))$ is negative definite and (iv) $\mathbf{H}(\cdot) \equiv \nabla^2 d_\mu(\cdot)$ is locally Lipschitz-continuous in a neighborhood of $\mathbf{w}^*$.

The first three conditions have already been established. To prove condition (iv), recall that $\mathbf{H} = \mathbf{A}\mathbf{D}_r(\mathbf{w})\mathbf{A}^T$, where $\mathbf{D}_r(\mathbf{w})$ is a diagonal matrix with $r_j(\mathbf{w})$ as its j-th diagonal element. Since $d_j h_j(\mathbf{w}) + \mu > 0$ and $h_j(\mathbf{w})$ is differentiable, $r_j(\mathbf{w})$, $j = 1, 2, \ldots, n$, is also differentiable. Therefore, every element of $\mathbf{H}$ is a finite combination of differentiable functions. Since differentiability implies Lipschitz continuity, condition (iv) is also satisfied and the proof is complete. □

4. THE TRIP DISTRIBUTION PROBLEM WITH QUADRATIC COST

IN THIS SECTION, we apply the theory developed in Section 2 to the specific problem of trip distribution with quadratic cost defined in Section 1, i.e., Program Q. To put the problem in the form of Program

P_μ defined in Introduction, rewrite Program Q as follows.

Program P'_μ:

$$\text{Minimize} \quad f_\mu(\mathbf{x}) \equiv \mu \sum_{i=1}^{K}\sum_{j=1}^{L} x_{ij} \ln x_{ij} + \sum_{i=1}^{K}\sum_{j=1}^{L} c_{ij}x_{ij} + \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{L} d_{ij}x_{ij}^2$$

$$\text{subject to:} \quad \sum_{j=1}^{L} x_{ij} = O_i, \quad i = 1,2,\dots,K, \tag{23}$$

$$\sum_{i=1}^{K} x_{ij} = D_j, \quad j = 1,2,\dots,L, \tag{24}$$

$$x_{ij} \geqslant 0, \quad i = 1,2,\dots,K, \text{ and } j = 1,2,\dots,L. \tag{25}$$

Note that $\sum_{i=1}^{K} O_i = \sum_{j=1}^{L} D_j$ and one of the $K+L$ equality constraints is redundant. Therefore, for obtaining a full-rank constraint matrix, we delete the last equality constraint. To make the format of Program P'_μ consistent with that of Program P_μ, we relabel the variables from 1 through $K \times L$ row-wise, i.e.,

$$x_k \equiv x_{[k/L]+1,k-([k/L]\times L)}, \; k = 1,2,\dots,K\times L, \tag{26}$$

where $[\cdot]$ is the integer truncation symbol. Moreover, the constraint matrix A can be partitioned as:

$$A = \begin{bmatrix} A^1 \\ A^2 \end{bmatrix}, \tag{27}$$

where $A^1 \equiv [a_{ij}^1]$, $A^2 \equiv [a_{ij}^2]$, and

$$a_{ij}^1 = 1, \quad \text{if } (i-1)\times L < j \leqslant i \times L, \tag{28}$$
$$0 \text{ otherwise;}$$
$$a_{ij}^2 = 1, \quad \text{if } j = (k-1)\times L + i$$
$$\text{for some } k = 1,2,\dots,K, \tag{29}$$
$$0 \text{ otherwise.}$$

The k-th element of the gradient, after some algebraic manipulation, becomes:

$$g_k = -\sum_{j=(k-1)\times L+1}^{L} h_j(\mathbf{w}) + b_k, \quad \text{if } k = 1,2,\dots,K; \tag{30}$$
$$-\sum_{j=1}^{K} h_{(k-K)+L\times(j-1)}(\mathbf{w}) + b_k, \quad \text{if } k = K+1, K+2,\dots,K+L-1.$$

The Hessian matrix of the dual objective function can be partitioned into a 2×2 block symmetric matrix, in which the upper-left (1,1) block and the lower-right (2,2) block are diagonal. The k_1-th, $k_1 = 1,2,\dots,K$, diagonal element of the Hessian matrix, i.e., the k_1-th diagonal element of the (1,1) diagonal block, is

$$H_{k_1,k_1} = -\sum_{j=(k_1-1)\times L+1}^{k_1\times L} \frac{h_j(\mathbf{w})}{d_j h_j(\mathbf{w}) + \mu}. \tag{31}$$

The k_1-th, $k = K+1, K+2,\dots,K+L-1$, diagonal element of the Hessian matrix, i.e., the $(k_1 - K)$-th diagonal element of the (2,2) diagonal block, is

$$H_{k_1,k_1} = -\sum_{k=1}^{K} \frac{h_{(k-1)\times L+k_1-K}(\mathbf{w})}{d_{(k-1)\times L+k_1-K}h_{(k-1)\times L+k_1-K}(\mathbf{w}) + \mu}. \tag{32}$$

The (k_1,k_2), $k_1 = 1,2,\dots,K$ and $k_2 = K+1, K+2,\dots,K+L-1$, element of the Hessian matrix, i.e., the $(k_1, k_2 - K)$, element of the upper-right (1,2) block, is:

$$H_{k_1,k_2} = -\frac{h_{(k_1-1)\times L+(k_2-K)}(\mathbf{w})}{d_{(k_1-1)\times L+(k_2-K)}h_{(k_1-1)\times L+(k_2-K)}(\mathbf{w}) + \mu}. \tag{33}$$

Note that the special structure of the matrix A associated with the trip distribution problems simplifies the calculation of both the gradient and the Hessian matrix. In the case of gradient, each element of the gradient vector requires a summation of only either L or K terms, instead of $K \times L$ terms required otherwise. As for the Hessian matrix, each of its diagonal element involves the summation of either K or L terms, as opposed to $K \times L$ terms. Moreover, the calculation of any off-diagonal element involves no such summations at all. It turns out that the inversion of the Hessian matrix can also be facilitated by using the block matrix inversion formula.

Given any invertible matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}, \tag{34}$$

its inverse $\mathbf{B}^{-1}$ can be obtained as follows:

$$\mathbf{B}^{-1} = \begin{bmatrix} [\mathbf{B}_{11} - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\mathbf{B}_{21}]^{-1} & -\mathbf{B}_{11}^{-1}\mathbf{B}_{12}[\mathbf{B}_{22} - \mathbf{B}_{21}\mathbf{B}_{11}^{-1}\mathbf{B}_{12}]^{-1} \\ -\mathbf{B}_{22}^{-1}\mathbf{B}_{21}[\mathbf{B}_{11} - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\mathbf{B}_{21}]^{-1} & [\mathbf{B}_{22} - \mathbf{B}_{21}\mathbf{B}_{11}^{-1}\mathbf{B}_{12}]^{-1} \end{bmatrix}. \tag{35}$$

Since the (1,1) and (2,2) blocks of the Hessian matrix are diagonal, the inversion of the Hessian matrix can be reduced to the inversion of one $K \times K$ matrix and one $(L - 1) \times (L - 1)$ matrix plus some elementary matrix multiplications.

5. COMPUTATIONAL EXPERIENCE

WE FIRST BRIEFLY describe how the test problems are randomly generated. The quadratic coefficients are generated randomly by multiplying a random number uniformly distributed in (0,1) by a scaling factor. The positivity guarantees the convexity of the primal objective function. We will report the sensitivity of the algorithm performance with respect to the quadratic scale ranging from 0.01 to 1.0. The linear coefficients are generated similarly. The scaling factor for the linear coefficients ranges from 1.0 to 100.0. The weight of the entropy term is set to three possible values, 5.0, 0.5, or 0.05. The right-hand-side of each of the primal constraint is generated by multiplying a random number uniformly distributed between (0,1) by a demand scale, which ranges from 100.0 to 5000.0.

A major indicator for the robustness of a particular algorithm is its performance as a function of the size of the problem. In what follows, we will use the notation of $\#O \times \#D$ to denote the size of a problem with $\#O$ origins and $\#D$ destinations. To highlight the sensitivity of the algorithm to the size of the problem, we chose to (i) generate the right-hand-sides, for most of the test problems, using a common demand scale, 1000.0, (ii) use a common weight of 0.5 for the entropy term, (iii) use the common linear scale of 10.0. With these parameters fixed and the quadratic scale set at a particular value, we compare the performance of the algorithm with respect to the problem size. We also vary the quadratic scale and compare its performance as a function of this scale. The reason for varying the quadratic scale is to gauge the sensitivity of the performance with respect to the coefficients of the highest-order polynomial terms. Due to the large number of possible combinations of parameter values, we fix the size of 100 origins and 100 destinations as the base size and vary some other parameters for gauging the performance sensitivity. Also, to make sure that this algorithm can consistently solve large-size problems, we ran four sets of $400O \times 400D$ problems, with only the quadratic scale varying and the linear scale, the weight of the entropy term and the demand scale fixed at 10.0, 0.5 and 1000.0 respectively.

The selected demand scale of 1000.0, in our opinion, should be large enough. For example, in a $400O \times 400D$ problem, this scale would amount to on average 80 million trips. We had selected this large number to demonstrate the power of this algorithm. In solving P_{μ}, only the *relative* magnitude, not their absolute magnitude, of the quadratic scale, linear scale and the weight of the entropy term matters. For example, the optimal solution of the problem defined by the three scales set respectively at 1.0, 100.0, and 5.0 is the same as that of the problem defined by 0.1, 10.0, and 0.5. We believe that the (relative) ranges we have chosen indeed cover most practical cases of interest. The relative magnitude selected for studying the sensitivity of the performance with respect to the problem size is 0.1, 10.0, and 0.5. At these scales, none of the three terms in the primal objective dominates any of the other terms.

Our goal here is to demonstrate the property of global convergence, the rate of convergence, and the robustness of this algorithm with respect to varying individuality of the test problems. It is not our intention to provide evidence on the fastest possible implementation of this algorithm. The matrix inversion used in our implementation is done by Gaussian elimination, not by Cholesky factorization or any other more efficient methods which could further improve the performance. Both α_k and β_k are set to 1 in all test cases. The initial solution for the unconstrained dual problem is set at $(0, 0, \ldots, 0)$ for all test problems. We stop the algorithm only when the norm of gradient becomes zero or no better point can be found by the line search. The ε's used are 10^{-5} and 10^{-10} respectively and, for easier comparison, they are set at these same values regardless of problem size. The progression of the objective values does not play any role in the stopping rule. (It is apparent in our test results that the already low number of iterations can be further cut even in half had we chosen to stop the algorithm when the progression does not improve the current objective by more than 0.1%.) For use of this algorithm on real problems, the matrix inversion could be performed using more efficient methods, the progression of the objective values may also be used as a stopping criterion, and the α_k, β_k, and ε's and other parameters needed for numerical search should be "fine tuned" for the specific problem size. Also, the performance could benefit from a more intelligent initial solution.

All tests are done on a SUN SPARCstation 2 and the results are tabulated in Table I. Each horizontal block in the table corresponds to one common seed in random problem generation. This arrangement is to enable the study of the role played by the different parameters, e.g., the size of the quadratic scale. The first column contains the test case number. The next two columns, the number of origins and that of destinations, represent the size of the problem. The fourth and the fifth columns give the quadratic scale and the linear scale. The sixth column gives the demand scale. The seventh column contains the weight of the entropy term. The last three columns provide the performance measures: the number of iterations, and the gross and the per-iteration CPU time consumption. Note that the CPU seconds are provided *only* as a reference.

Here are some observations derived from the test results.

(1) The number of iterations increases very slowly as the size of the problem grows. (See the blocks for $10O \times 10D$ and $50O \times 50D$, the first three test problems in each of the two $100O \times 100D$ blocks, the blocks for $200O \times 200D$ and $300O \times 300D$, and finally the four $400O \times 400D$ blocks.) Aside from the first block of $400O \times 400D$ problems and test case 35, the number of iterations hardly increases with the problem size at all and basically remain at around 10. Let us reiterate that had we chosen to stop the algorithm when the objective function does not improve by more than 0.1%, most of these numbers can even be further cut in half. This shows an amazingly low degree of sensitivity to the problem size.

(2) For the smaller problems, the number of iterations tends to increase as the magnitude of the quadratic scale goes up. However, this is no longer valid for the larger problems. (Compare the three test problems within each of the blocks pointed out in (1).)

(3) The number of iterations tends to increase as the magnitude of the linear scale goes up. (Compare, for example, the second and the sixth test problems in the second $100O \times 100D$ block, i.e., case 13 and 17.)

(4) The number of iterations tends to increase, but only very slightly, as the weight of the entropy term goes up. (Compare, for example, the fourth and the fifth test problems of the second $100O \times 100D$ block, i.e., cases 15 and 16.)

(5) Also, the number of iterations tends to increase, but very slightly, as the magnitude of the demand scale goes up. (Compare, for example, the second, the eighth and the ninth test problems in the second $100O \times 100D$ block, i.e. cases 19 and 20.)

(6) When we increase the magnitude of more than one parameter, the effect on the number of iterations seems to compound. For example, increasing both the linear scale and the demand scale simultaneously would increase noticeably the number of iterations. (Compare the second and the fourth test problems in the first $100O \times 100D$ block, i.e., cases 8 and 10.) Also, increasing the quadratic scale, the linear scale and the demand scale simultaneously would increase the number of iterations even more. (Compare the second and the fifth test problems in the first $100O \times 100D$ block, i.e., cases 8 and 11.)

(7) Note that some of these test problems may not be realistic and are created simply to test the limit of this algorithm. As mentioned earlier, the performance can be further improved in several ways, some of which depend on the specifics of the practical problems being solved.

For the pure entropy maximization problem with linear constraints, the average number of iterations grows very slowly from 5 for 10×100 problems to 14 for 500×1000 problems.[11] Our experience with two other classes of problems[12, 13] is very similar. For larger problems, the major computational requirements at each iteration come from the inversion of the Hessian matrix. Since the calculation of the Hessian matrix has been significantly simplified, the proportion of computation effort spent on solving the one-dimensional implicit functions is significant but decreases as the problem size increases.

By setting the unit quadratic costs d_{ij} in Program P'_μ to zero, we can use the computational procedure and the curved search algorithm to solve the trip distribution problem with *only* linear cost. A well-known algorithm for solving such a problem is the Bregman's balancing algorithm.[17] In the rest of this section, we compare and discuss the difference in performance between these two methods.

Bregman's balancing method is applicable in a setting more general than entropy optimization with linear constraints. When applied to solving such entropy optimization problems, Bregman's method can be viewed as one which minimizes the dual objective with respect to one variable at a

TABLE I
Trip Distribution with Quadratic Cost: Computational Results

Case #	#O	#D	q s	l s	d s	μ	#(iter)	cpu(s)	cpu/iter
1	10	10	0.01	10.0	1000.0	0.5	10	3.83	0.38
2	10	10	0.1	10.0	1000.0	0.5	10	4.00	0.40
3	10	10	1.0	10.0	1000.0	0.5	20	10.13	0.51
4	50	50	0.01	10.0	1000.0	0.5	8	70.28	8.78
5	50	50	0.1	10.0	1000.0	0.5	8	74.94	9.37
6	50	50	1.0	10.0	1000.0	0.5	17	192.27	11.31
7	100	100	0.01	10.0	1000.0	0.5	7	329.76	47.10
8	100	100	0.1	10.0	1000.0	0.5	8	376.02	47.00
9	100	100	1.0	10.0	1000.0	0.5	10	503.54	50.35
10	100	100	0.1	100.0	5000.0	0.5	22	1981.59	90.07
11	100	100	1.0	100.0	5000.0	0.5	36	3346.44	92.96
12	100	100	0.01	10.0	1000.0	0.5	9	432.81	48.09
13	100	100	0.1	10.0	1000.0	0.5	9	415.67	46.18
14	100	100	1.0	10.0	1000.0	0.5	16	770.62	48.16
15	100	100	0.1	10.0	1000.0	0.05	11	748.14	68.01
16	100	100	0.1	10.0	1000.0	5.0	16	848.80	53.05
17	100	100	0.1	100.0	1000.0	0.5	16	1248.57	78.03
18	100	100	0.1	1.0	1000.0	0.5	7	322.77	46.11
19	100	100	0.1	10.0	5000.0	0.5	10	591.92	59.19
20	100	100	0.1	10.0	100.0	0.5	8	378.48	47.31
21	200	200	0.01	10.0	1000.0	0.5	12	2391.57	199.29
22	200	200	0.1	10.0	1000.0	0.5	10	2211.48	221.15
23	200	200	1.0	10.0	1000.0	0.5	11	3023.25	274.84
24	300	300	0.01	10.0	1000.0	0.5	12	8196.24	683.02
25	300	300	0.1	10.0	1000.0	0.5	10	7242.34	724.23
26	300	300	1.0	10.0	1000.0	0.5	13	9953.42	765.65
27	400	400	0.01	10.0	1000.0	0.5	24	39390.30	1641.26
28	400	400	0.1	10.0	1000.0	0.5	23	37666.60	1637.67
29	400	400	1.0	10.0	1000.0	0.5	28	51081.20	1824.33
30	400	400	0.01	10.0	1000.0	0.5	11	17745.00	1613.18
31	400	400	0.1	10.0	1000.0	0.5	10	16861.70	1686.17
32	400	400	1.0	10.0	1000.0	0.5	12	21771.70	1814.31
33	400	400	0.01	10.0	1000.0	0.5	12	20850.90	1737.58
34	400	400	0.1	10.0	1000.0	0.5	13	21614.30	1662.64
35	400	400	1.0	10.0	1000.0	0.5	24	45410.50	1892.10
36	400	400	0.01	10.0	1000.0	0.5	8	13613.00	1701.63
37	400	400	0.1	10.0	1000.0	0.5	9	15279.60	1697.73
38	400	400	1.0	10.0	1000.0	0.5	10	18138.20	1813.82

time.[17] Therefore, the number of iterations required grows very rapidly with respect to the size of the problem. As pointed out by JORNSTEN and LUNDGREN,[16] Bregman's method is computationally efficient when all constraint coefficients a_{ij} have the value of zero or one. This is because, with this special structure, no line searches are required. However, with the presence of any non-zero-one coefficients a_{ij} in a constraint, the computational effort required for the associated iteration becomes much larger. Compounded with the rapid increase of the number of iterations as the problem size grows, the computational effort required for general problems grows rapidly. Therefore, the number of general constraints must be limited.[16]

Our experience shows that, for the trip distribution problems with *only* linear cost, in general Bregman's method tends to be more efficient than our method. Although the number of Bregman iterations grows very fast with respect to the problem size while its counterpart in our approach barely grows, the small computational effort per iteration compensates for the huge number of iterations. However, one clear advantage of our approach is robustness. We demonstrate it by comparing the performance of the two algorithms against two well-known "benchmark" entropy optimization problem with trip distribution (transportation) constraints. The problems are posed as matrix scaling problems.

The matrix scaling problem is to find two diagonal matrices, $\mathbf{D}_1$ and $\mathbf{D}_2$, for a given square matrix $\mathbf{M}$ such that the scaled matrix $\mathbf{D}_1\mathbf{M}\mathbf{D}_2$ has prescribed row and column sums. This problem is equivalent to solving for a gravity model and hence equivalent to a trip distribution problem with linear cost. Two matrix scaling problems, denoted by P_1 and P_2, respectively, with

$$\mathbf{M}_1 = \begin{bmatrix} 10^4 & 10^2 & 10^2 \\ 10^2 & 1 & 1 \\ 10^2 & 1 & 1 \end{bmatrix}$$

$$\text{and} \quad \mathbf{M}_2 = \begin{bmatrix} 10^2 & 10^2 & 0 \\ 10^2 & 10^4 & 1 \\ 0 & 1 & 10^2 \end{bmatrix}.$$

were studied by ELFVING and others.[5] Note that scaling $\mathbf{M}_1$ is equivalent to a trip distribution problem with $\mu = 1$,

$$\mathbf{c}_1 = \begin{bmatrix} -ln(10^4) \\ -ln(10^2) \\ -ln(10^2) \\ -ln(10^2) \\ -ln(1) \\ -ln(1) \\ -ln(10^2) \\ -ln(1) \\ -ln(1) \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{b}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Since the rank of matrix $\mathbf{A}_1$ is 5 (one of the scaling constraints is redundant), we can eliminate the last constraint to obtain a full-rank matrix. P_2 is defined similarly. Note that there are two zeros in $\mathbf{M}_2$. We replace them by a small number 10^{-10} before applying the two algorithms.

Both algorithms stop when the maximum violation of the primal constraints is no larger than a preset threshold. (For the curved-search algorithm, we set the other parameters to the same values used for the earlier tests.) Two selected thresholds are 0.001 and 0.00001. The performance results for the two methods are shown in Table II and Table III, respectively. Note that the performance of the Bregman's method is very sensitive to the individuality of these two problems while that of the curved-search method is unaffected. The robustness of the latter, compared to some other entropy optimization methods, was also shown in [11].

We make the following remarks regarding the two different methods before closing this section. Our method solves a more general optimization problem, which accommodates quadratic cost in the objective function. Its performance seems very insensitive to the individuality of the problem and is not affected as drastically as the Bregman's method when the constraint matrix contains non-zero-one entries. The number of iterations, as a function of problem size, grows very slowly, which makes it especially attractive for the case of large-size general constraint matrix **A**.[11] As mentioned earlier, our algorithm can be fine tuned to improve the performance, e.g., using the factorization techniques to speed up matrix inversion. A comprehensive computational comparison among several major algorithms for the entropy optimization problem with linear constraints is a worthy subject for future study.

TABLE II

Performance of Bregman's Method

	Max Violation = 0 001		Max Violation = 0 00001	
Problem	Iterations	CPU Seconds	Iterations	CPU Seconds
P_1	125	0.03	185	0.04
P_2	1710	0.37	24900	5.32

TABLE III

Performance of Curved-Search Method

	Max Violation = 0.001		Max Violation = 0 00001	
Problem	Iterations	CPU Seconds	Iterations	CPU Seconds
P_1	6	0.08	8	0.1
P_2	4	0.05	6	0.08

6. CONCLUSION

IN THIS PAPER, we present a general entropy optimization model, together with an efficient dual-based algorithm, which not only encompasses many existing transportation planning models but also extends them. In addition to the capability of accommodating quadratic cost in Program P_μ, our approach has several important advantages. First of all, the strong duality theorem holds under very mild conditions and the dual program D_μ is an unconstrained convex program. A primal optimal solution can be easily obtained from the dual optimal solution through a simple conversion equation. Second, the post-optimality analysis, including parametric programming and sensitivity analysis, can benefit from the unconstrained nature of the dual. Third, the robustness and efficiency of the curved-search algorithm make this approach particularly suitable for large-size real-life problems. More sophisticated implementations of the algorithm can further improve the already impressive performance and the high degree of "parallelizability" of most of the computation can be further exploited for higher efficiency.

ACKNOWLEDGMENT

THE AUTHORS would like to thank two anonymous referees and an Associate Editor of Transportation Science for their very helpful comments and suggestions. The work of the first author was partially supported by the North Carolina Supercomputing Center, the 1992 Cray Research Award, and the National Science Council Grant #NSC 81-0415-E-007-10 of the Republic of China.

REFERENCES

1. T. M. APOSTOL, *Mathematical Analysis*, Addison-Wesley, Reading, MA, 1972.
2. A. BEN-TAL, A. MELMAN AND J. ZOWE, "Curved Search Methods for Unconstrained Optimization," *Optimization* **21,** 669–695 (1990).
3. S. BRICE, "Derivation of Nested Transport Models within a Mathematical Programming Framework," *Transportation Research* **23B,** 19–28 (1989).
4. R. W. COTTLE, "Application of a Block Successive Overrelaxation Method to a Class of Constrained Matrix Problems," *Proceedings of the International Congress on Mathematical Programming*, R. W. Cottle, M. L. Kelmanson, and B. Korte (eds.), Rio de Janeiro, Brazil, April, 1981, 89–103.
5. T. ELFVING, "On Some Methods for Entropy Maximization and Matrix Scaling," *Linear Algebra and Its Applications* **34,** 321–339 (1980).
6. J. ERIKSSON, "A Note on Solution of Large Sparse Maximum Entropy Problems with Linear Equality Constraints," *Mathematical Programming* **18,** 146–154 (1980).
7. S. ERLANDER, "Accessibility, Entropy and the Distribution and Assignment of Traffic," *Transportation Research* **11,** 149–153 (1977).
8. S. ERLANDER, "Entropy in Linear Programs," *Mathematical Programming* **21,** 137–151 (1981).
9. S. ERLANDER, "Efficient Population Behavior and the Simultaneous Choices of Origins, Destinations and Routes," *Transportation Research B* **24B,** 363–373 (1990).
10. S. ERLANDER AND N. F. STEWART, *The Gravity Model in Transportation Analysis*, VSP, The Netherlands (1990).
11. S. C. FANG AND H.-S. J. TSAO, "A Quadratically Convergent Global Algorithm for the Linearly-Constrained Minimum Cross-Entropy Problem," North Carolina State University Operations Research Report No. 255, July 1991, *European Journal of Operational Research* **79**, 369–378 (1994).
12. S. C. FANG AND H.-S. J. TSAO, "Linear Programming with Entropic Perturbation," *Zeitschrift fur Operations Research* **37,** 171–186 (1993).
13. S. C. FANG AND H.-S. J. TSAO, "An Unconstrained Convex Programming Approach to Solving Convex Quadratic Programming Problems," *Optimization* **27,** 235–243 (1993).
14. R. GORDON, R. BENDER AND G. T. HERMAN, "Algebraic Reconstruction Techniques (ART) for Three Dimensional Electron Microscopy and X-Ray Photography," *J. Theoretical Biology* **29,** 471–481 (1970).
15. K. O. JORNSTEN, "An Algorithm for the Combined Distribution and Assignment Problem," *Transportation Research* **15B,** 21–33 (1981).
16. K. O. JORNSTEN AND J. T. LUNDGREN, "An Entropy-Based Modal Split Model," *Transportation Research B* **23B,** 345–359 (1989).
17. B. LAMOND AND N. F. STEWART, "Bregman's Balancing Method," *Transportation Research B* **15B,** 239–248 (1981).
18. A. LENT, "Maximum Entropy and MART," in *Image Analysis and Evolution, SPSE Conference Proceedings*, R. Shaw (ed.), Toronto, Canada, 249–257 (1976).
19. R. T. ROCKAFELLAR, Convex Analysis, Princeton University Press, Princeton, NJ (1972).
20. R. T. ROCKAFELLAR, "Lagrange Multipliers in Optimization," *Nonlinear Programming—SIAM-AMS Proceedings Vol. IX*, R. W. Cottle and C. E. Lemke (eds.), American Mathematical Society, Providence, RI (1976).
21. K. N. A. SAFWAT AND T. L. MAGNANTI, "A Combined Trip Generation, Trip Distribution, Modal Split, and Trip Assignment Model," *Transportation Science* **18,** (1988).
22. J. A. TOMLIN, "A Mathematical Programming Model for the Combined Distribution-Assignment of Traffic," *Transportation Science* **5,** 123–140 (1971).
23. J. A TOMLIN AND S. G. TOMLIN, "Traffic Distribution and Entropy," *Nature* **220,** 974–976 (1968).
24. A. G. WILSON, "A Statistical Theory of Spatial Distribution Models," *Transportation Research* **1,** 253–269 (1967).

(Received, February 1993; revisions received: November 1993, September 1994; accepted October 1994)