THE A-FOREST ITERATION METHOD FOR THE STOCHASTIC GENERALIZED TRANSPORTATION PROBLEM

Liqun Qi

July 1984
WP-84-58

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

THE A-FOREST ITERATION METHOD FOR THE STOCHASTIC GENERALIZED TRANSPORTATION PROBLEM*

Liqun Qi

International Institute for Applied Systems Analysis,
A-2361 Laxenburg, Austria

* This research was supported by the National Science Foundation under grant No. MCS-8200632.

ACKNOWLEDGEMENT

The author's special thanks are due to Professor S.M. Robinson of the University of Wisconsin-Madison for his suggestions and comments.

ABSTRACT

The stochastic generalized transportation problem (SGTP) has an optimal solution: each of the connected subgraphs of its graph is either a tree or a one-loop tree. We call such a graph an A-forest. We propose here a finitely convergent method, the A-forest iteration method, to solve the SGTP. It iterates from one base A-forest triple to another base A-Forest triple. The iteration techniques constitute some modifications of those for the first iteration method for solving the stochastic transportation problem (STP), which was given in [16]. Sensitivity Analysis and numerical examples are also given.

1. Introduction

In 1955 and 1956, some of the earliest papers on the stochastic programming problem, under the name: linear programming under uncertainty were published. Ferguson and Dantzig [7] [8] presented an allocation problem of aircraft to routes. In the modern terminology, this is a stochastic linear problem with simple recourse [27]. The demands are stochastic. If we fix the demands, then we have a **weighted distribution problem,** or a **generalized transportation problem** (GTP) [5]. Therefore, an appropriate mathematical name of their problem is **stochastic generalized transportation problem** (SGTP). In their papers, the distribution functions of the stochastic demands are discrete. They proposed a special version of the simplex method to solve them. This material, with minor changes, forms as the last chapter of Dantzig's celebrated book [5].

In 1960, Elmaghraby [6] studied this problem with continuously distributed random demands. He showed that Ferguson and Dantzig's method will cause error in this case. He presented an iteration method for solving it. He claimed that his method is finitely convergent. However, in each iteration step, a SYSTEM of nonlinear simultaneous equations must be solved. This is not so easy and the solutions of the system need not be unique. He pointed out that at any iteration, several or even infinitely many tableaux may ensue.

Since then, in almost twenty years, several papers appeared which discuss a relatively simple case: the **stochastic transportation problem** (STP), where all transformation coefficients are 1's. [3] [23] [28] [29] [30] [31]. The SGTP was discussed in few papers [1]. Some authors proposed the use of some general convex programming algorithms to solve the STP or SGTP. These methods are not finitely convergent in general. Some of them , like [29] [30] gave some easily calculated prior bounds for the STP.

In [16], we have presented a finitely convergent method, namely the forest iteration method, to solve the STP. This method is based on the network structure of the problem. At each step, the nonlinear problem consists of solving a small number of one-dimensional monotone equations. Modifications of this method can be found in [17].

The difference between the SGTP and the STP is that the graph of a "basic" optimal solution of the SGTP is not necessarily a collection of trees, i.e., a forest, but a collection of trees and one-loop trees, which we call A-trees. We call such a graph an A-forest. However, it is still possible to develop an iteration method for solving the SGTP, which is based on A-forests and has the same characteristics as the forest iteration method for the STP.

In Section 2, we state the formulation of the SGTP as well as its optimality conditions. In Section 3, we define A-forests and discuss their properties. In Section 4, we discuss the minimization problem on an A-forest when we discard the nonnegativity restrictions. In Sections 5 and 6, we discuss the iteration techniques: cutting, connecting and pivoting. Compared with [16], the technique of connecting is improved so that it is not necessary to calculate the flow changes in connecting. There are also some other improvements for those techniques. In

Section 7, we give the A-forest iteration algorithm and its convergence theorem. In Section 8, we give sensitivity analyses. In Section 9, we use Elmaghraby's allocation problem as our numerical example. Elmaghraby used the solution, when the demands were considered fixed at its expected values, as the starting point of his method in his example. We see that the optimal A-forest is already in hand in this example and few calculations of the method described in Section 4 will give an optimal solution. This shows that in many cases the required iteration number will be very small if we start from a good approximate solution. In Section 10, we use this example to illustrate sensitivity analyses and the iteration techniques.

Our method can be extended to the generalized network flow problem case [5] [13] [24]. This extension work is similar to what we have done for the STP in [16]. We shall not repeat it here.

2. The Stochastic Generalized Transportation Problem

The formulation of the SGTP is as follows [1] [5] [6] [7] [8]:

$$\min_{x,w} \sum_{(i,j)\in S} c_{ij}x_{ij} + \sum_{j=1}^{n} \phi_j(w_j)$$

$$\text{s.t.} \sum_{j\,(i,j)\in S} x_{ij} \le a_i, \quad i = 1,\dots,m, \tag{2.1}$$

$$\sum_{i\,(i,j)\in S} r_{ij}x_{ij} = w_j, \quad j = 1,\dots,n,$$

$$x_{ij} \ge 0, \quad \text{all } (i,j)\in S,$$

where

$$\phi_j(w_j) = q_j^+ \int_{\xi_j < w_j} (w_j - \xi_j)\,dF_j(\xi_j) + q_j^- \int_{\xi_j > w_j} (\xi_j - w_j)\,dF_j(\xi_j)$$

S : the set of available cells.

a_i: the available amount of resource i, $a_i > 0$.

c_{ij}: the cost of manufacturing product j using one unit of source i, $c_{ij} \ge 0$.

x_{ij}: the quantity of resource i devoted to product j in specific unit time.

r_{ij}: the productivity per unit of resource i when producing item j.

w_j: the amount of product j produced in specific unit time.

ξ_j: the observed value of ξ_j.

ξ_j: the random variable of demand for product j.

F_j: the marginal distribution function of bold ξ_j, which is known.

q_j^+ : the salvage cost per unit of excess inventory of item j.

q_j^- : the penalty cost per unit of inventory shortage of item j.

We assume $q_j = q_j^+ + q_j^- \geq 0$. In [1], the transformation coefficients are in the constraints containing a_i. It is the same thing essentially since we can put $\tilde{x}_{ij} = r_{ij} x_{ij}$ to transfer this formulation to that formulation. If we have $r_{ij} = 1$, for all i and j, then we get the formulation of the STP.

According to [25], we know that ϕ_j is convex and continuous. We can add m slack variables $x_{1,n+1}, \ldots, x_{m,n+1}$ to (2.1) to change it to the following form:

$$\min_{x,w} \sum_{(i,j)\in S} c_{ij} x_{ij} + \sum_{j=1}^{n} \phi_j(w_j)$$

$$\text{s.t.} \sum_{j \; (i,j)\in \bar{S}} x_{ij} = a_i, \quad i = 1,\ldots,m,$$

$$\sum_{i \; (i,j)\in S} r_{ij} x_{ij} = w_j, \quad j = 1,\ldots,n, \qquad (2.2)$$

$$x_{ij} \geq 0, \quad \text{all } (i,j) \in \bar{S},$$

where we extend S to $\bar{S}$ to include all slack variables.

In many practical situations, all r_{ij}'s are nonnegative. We denote such problems as the SGTP*. Sometimes, we also call a variable x_{ij} a slack variable if $r_{ij} = 0$. To simplify our discussion, we suppose all $r_{ij} \neq 0$.

According to convex programming theory [18] [19], (x, w) is an optimal solution if and only if there exist $u \in \mathbf{R}^m$, $v \in \mathbf{R}^n$ such that:

$$
\begin{aligned}
&\sum_{j:\,(i,j)\in\bar{S}} x_{ij} = a_i, \quad i = 1,\dots,m,\\
&\sum_{i:\,(i,j)\in S} r_{ij} x_{ij} = w_j, \quad j = 1,\dots,n,\\
&x_{ij} \geq 0, \quad \text{all } (i,j)\in\bar{S},\\
&u_i + r_{ij} v_j \leq c_{ij}, \quad (i,j)\in S,\\
&u_i \leq 0, \quad i = 1,\dots,m,\\
&x_{ij}(c_{ij} - u_i - r_{ij} v_j) = 0, \quad (i,j)\in S,\\
&u_i x_{i,n+1} = 0, \quad i = 1,\dots,m,\\
&-v_j \in \partial\phi_j(w_j), \quad j = 1,\dots,n.
\end{aligned}
\tag{2.3}
$$

If we fix w at (2.2) and only minimize on x, we get a generalized transportation problem (GTP), or sometimes called weighted distribution problem. We denote it by T(w). According to linear programming theory, x is an optimal solution of T(w) if and only if there exist $u \in \mathbf{R}^m$, $v \in \mathbf{R}^n$ such that:

$$
\begin{aligned}
\sum_{\substack{j \\ (i,j)\in\bar{S}}} x_{ij} &= a_i, \quad i = 1,\ldots,m, \\
\sum_{\substack{i \\ (i,j)\in S}} r_{ij} x_{ij} &= w_j, \quad j = 1,\ldots,n, \\
x_{ij} &\geq 0, \quad \text{all } (i,j) \in \bar{S}, \\
u_i + r_{ij} v_j &\leq c_{ij}, \quad (i,j) \in S, \qquad\qquad (2.4) \\
u_i &\leq 0, \quad i = 1,\ldots,m, \\
x_{ij}(c_{ij} - u_i - r_{ij} v_j) &= 0, \quad (i,j) \in S, \\
u_i x_{i,n+1} &= 0, \quad i = 1,\ldots,m, \\
-v_j &\in \partial\phi_j(w_j), \quad j = 1,\ldots,n.
\end{aligned}
$$

We see that the only difference between (2.3) and (2.4) is that (2.3) has one more condition, the last condition.

In [5], Dantzig gave a nice method to solve the GTP. Consider solutions on an $m \times (n+1)$ tableau. What we are interested is [5] [13]:

Theorem 2.1 If a GTP is solvable, then it has an optimal basic graph, each of whose connected subgraphs is either a tree with exactly one slack variable, or a one-loop tree without slack variables. □

We give such a graph a name: an A-forest. We discuss such A-forests in detail in the next section.

Denote the objective function of (2.2) by $cx + \phi(w)$.

3. A-Forests

Suppose we have an m × (n+1) transportation tableau T. The n+1 column is for slack variables. Suppose we have coefficients c_{ij}'s and r_{ij}'s given by (2.1) associated with T.

Definition 3.1 Suppose we have a set of cells on T, which forms a loop and which has no slack variables. Then the coefficients of 1's and r_{ij}'s in (2.1) forms a square matrix. If this matrix is nonsingular, we call this cycle a **proper loop;** otherwise, we call it a **false loop.** □

Note that there is no proper loop in the standard transportation problem.

Definition 3.2 On T, a graph is called a one-loop tree, or an **A-tree,** if it is connected and has exactly one loop, which is a proper loop, and if it has no slack variables. A graph is called an **A-forest,** if each of its connected subgraphs is a tree or an A-tree and its row indices run throughout {1,...,m}. □

In Fig.1, we see an example of an A-forest with an A-tree component and a tree component.

According to Theorem 3.2 of [16], the number of cells of a forest is no more than m+n. This is also true for an A-forest even though it has loops.

Theorem 3.3 The number of cells of an A-forest is no more than m+ n, and no less than m.

Proof Since the row index set of an A-forest runs throughout {1,...,m}, we have the second conclusion. For any k-component A-forest (k>1), if it has a loop, we can break this loop by deleting a cell on the loop and we can insert a cell in the column of the deleted cell and in a row containing a cell of another component. The resulting graph is a k-1 component A-forest, the number of whose cells is the same and the number of whose loop is reduced. In this way, we can break all the loops of an A-forest without changing the number of the cells. Now we have a forest with the same number of cells as the original A-forest. According to Theorem 3.1 of [16], we have our conclusion. □

Applying this to (2.2), we have

Definition 3.4 Let $x \in \mathbf{R}^{m\times(n+1)}$, and Gr x, the graph of x, be the graph associated with the set $\{ (i,j) \mid x_{ij} \neq 0 \}$. If (x,w) is a feasible point of (2.2) and f = Gr x is an A-forest, then we call (x,w) an **A-forest point** of (2.2) and (x,w; f) an **A-forest triple** of (2.2). If (x^*,w^*) is an optimal solution of (2.2) and $f^* =$ Gr x^* is an A-forest, then we call f^* an **optimal**

A-forest of (2.2), and $(x^*, w^*; f^*)$ an **optimal A-forest triple** of (2.2).
□

The following lemma can be proved by Theorem 2.1 and Definition 3.1:

Lemma 3.5 If a GTP is solvable, then it has an optimal basic graph, which is an A-forest.

Now we have

Theorem 3.6 The SGTP (2.2) has an optimal A-forest triple.

Proof Replace w_j's in the objective function by combinations of x_{ij}'s. We see that (2.2) becomes an optimization problem in x. Clearly, it is feasible and the feasible set is compact. Since the objective function is continuous, we know that it attains a minimum. Suppose $(\bar{x}, w^*)$ is an optimal solution of (2.2). Then $T(w^*)$ is feasible and bounded. According to the theory of linear programming and Lemma 3.5, $T(w^*)$ has a basic optimal solution x^* such that $f^* = \mathrm{Gr}\ x^*$ is an A-forest. Comparing objective values, we see that (x^*, w^*) is also an optimal solution of (2.2). Therefore, $(x^*, w^*; f^*)$ is an optimal A-forest triple of (2.2). □

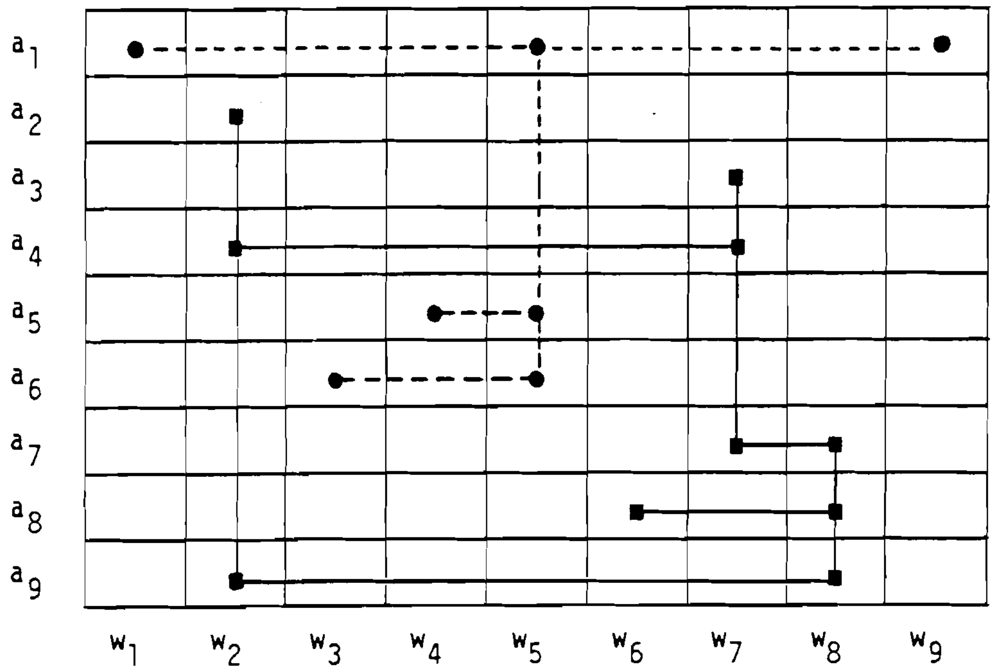

Figure 1. A-forest

(The loop is a proper loop here.)

4.4. Minimization on an A-Forest

As in [16], we use M_s and N_s to denote the row index set and the column index set (not including n+1) of a graph s. If we have a cell (i,j) such that $i \in M_s$ and $j \in N_s$, then we say (i,j) is in the **area** of s. If $i \in M_s$, $j \in N_t$, and s and t are unconnected, then we say that (i,j) is in the **joint area** of s and t.

Suppose f is an A-forest. Consider to solving

$$\begin{aligned} \min_{x,w} \quad & \sum_{(i,j)\in f} c_{ij}x_{ij} + \sum_{j\in N_f} \phi_j(w_j) \\ \text{s.t.} \quad & \sum_{j,\ (i,j)\in f} x_{ij} = a_i, \quad i=1,\ldots,m, \\ & \sum_{i,\ (i,j)\in f} r_{ij}x_{ij} = w_j, \quad j \in N_f. \end{aligned} \tag{4.1}$$

If f is an optimal A-forest, an optimal solution of (4.1) will be a part of an optimal solution of (2.2), with other parts being zeros. Without confusion, when we talk about an optimal solution (x,w) of (4.1) in the following, it implies that we talk about an $m \times (n+1) + n$ vector such that its corresponding part is an optimal solution of (4.1) and other parts are zeros.

As in Theorem 4.1 of [16], if f is a k-component A-forest, $k > 1$, then (4.1) can be separated into k minimization problems:

$$\min_{x,w} \sum_{(i,j)\in t} c_{ij}x_{ij} + \sum_{j\in N_t} \phi_j(w_j)$$

$$\text{s.t.} \sum_{(i,j)\in t} {}_j \; x_{ij} = a_i, \quad i \in M_t,$$

$$\sum_{(i,j)\in t} {}_i \; r_{ij}x_{ij} = w_j, \quad j \in N_t. \qquad (4.2)$$

where t's are component trees or component A-trees of f.

We discuss (4.2) in three different cases.

A. Minimization on a tree with a slack variable $x_{h,n+1}$.

The tree in Fig. 1 is an example. The necessary and sufficient conditions of optimal solutions on such a tree t are:

$$\sum_{(i,j)\in t} {}_j \; x_{ij} = a_i, \quad i \in M_t,$$

$$\sum_{(i,j)\in t} {}_i \; r_{ij}x_{ij} = w_j, \quad j \in N_t,$$

$$u_i + r_{ij}v_j = c_{ij}, \quad (i,j)\in t, \qquad (4.3)$$

$$u_h = 0,$$

$$-v_j \in \partial\phi_j(w_j), \quad j \in N_t.$$

Similarly to [16], the third and the fourth expressions of (4.3) form a triangular linear system of u_i's and v_j's; the first and the second expressions of (4.3) form a triangular linear system of x_{ij}'s. We can use the former triangular linear system to get u_i's and v_j's, use the fifth expression of (4.3) to determine w_j's, and use the latter triangular linear system to determine x_{ij}'s.

B. Minimization on a tree without slack variables.

The necessary and sufficient conditions of optimal solutions on such a tree t are:

$$\begin{aligned}
\sum_{j,\ (i,j)\epsilon t} x_{ij} &= a_i, \quad i \epsilon \mathcal{M}_t, \\
\sum_{i,\ (i,j)\epsilon t} r_{ij} x_{ij} &= w_j, \quad j \epsilon \mathcal{N}_t, \\
u_i + r_{ij} v_j &= c_{ij}, \quad (i,j) \epsilon t, \\
- v_j &\epsilon\, \partial \phi_j(w_j), \quad j \epsilon \mathcal{N}_t.
\end{aligned} \tag{4.4}$$

Again, this is similar to [16] and we still have triangularity. Treat any v_j as a parameter d. Then we can solve the third expression of (4.4) to get other u_i's and v_j's in term of this parameter d. We see that the first and the second expressions of (4.4) form a linear system such that the number of variables x_{ij}'s is less than the number of equations by 1. Therefore, in nondegenerate case, we can cancel x_{ij}'s to get a linear equation of w_j's:

$$\sum_{j \in N_t} e_j w_j = e_0, \tag{4.5}$$

for some coefficients e_j and e_0. By the fourth expression of (4.4), we know w_j is a nonincreasing function of v_j, therefore also of d, for $j \in N_t$. Therefore, (4.5) gives us a nonlinear equation of d. Solving this nonlinear equation, we get d. In terms of d, we can get u_i's and v_j's. By the fourth expression of (4.4), we can get w_j's. By the first and the second expressions of (4.4), we can now get x_{ij}'s. In degenerate case, as long as (4.2) is feasible, we still can get an expression (4.5) which is consistent with the coefficient matrix of the first and the second sets of (4.4). In the case of the SGTP*, we can see that when treating a v_k as a parameter d, other v_j's can be expressed by a linear expression of this parameter with positive coefficient: i.e.,

$$v_j = \alpha_j d + \beta_j, \quad \alpha_j > 0, \quad \text{for } j \in N_t.$$

And it is not too difficult to prove that all e_j's have the same signs in the SGTP* case. Therefore, (4.5) gives us a monotone equation of d in this case.

C. Minimization on an A-tree.

The A-tree in Fig. 1 is an example. The necessary and sufficient conditions of optimal solutions on such an A-tree t are the same as (4.4) in appearance, though in fact there is a loop, which means that there is one more equation in the first and the second expressions of (4.4) and that there is one more equation in the third expression too. According to our assumption on proper loops, we know that the third expression of (4.4) forms a nonsingular linear system, which is near-triangular [5], and the first and the second expressions of (4.1) form a nonsingular near-triangular linear system too. Therefore, we can solve the third expression of (4.4) to get u_i's and v_j's. A nice method to solve such a system is given in [5], i.e., treating any u_i or v_j as a parameter, we shall get other u_i's and v_j's in terms of this parameter. There is exactly one u_i or one v_j which has two linear expressions in term of this parameter. By equating these two expressions, we get this parameter, therefore, other u_i's and v_j's. From the fourth expression of (4.4), we get w_j's. Applying the same method to the first and the second expressions of (4.4), we get x_{ij}'s.

Proof According to (5.3), we have

$$r = 1 - \min_{(i,j)\in I} \frac{x_{ij}}{x_{ij} - \hat{x}_{ij}}.$$

Since $J = \emptyset$, we know $r < 1$. Therefore, (5.6) holds. According to (5.2), (5.4) and the convexity of the objective function, we know (5.5) holds. □

Theorem 5.4 Suppose we have an A-forest triple $(x, w; f)$, which is not a base A-forest triple. By repeating the cutting technique at most n times, we obtain a base A-forest triple $(x', w'; f')$ such that (5.5) holds.

Proof By (5.6), the number of cells of the A-forest is strictly decreasing in each cutting if it is still not a base A-forest triple. According to Theorem 3.3, we have our conclusion. □

If f is not a base A-forest triple, then we can get an optimal solution $(\hat{x}, \hat{w})$ of (4.1) such that $I := \{ (i,j) \mid \hat{x}_{ij} < 0 \} \neq \emptyset$ and

$$c\hat{x} + \phi(\hat{w}) < cx + \phi(w) \qquad (5.2)$$

Now take

$$r := \min \{ r' \mid 0 \leq r \leq 1, (1-r')\hat{x} + r' x \geq 0 \} \qquad (5.3)$$

and

$$x' := (1-r)\hat{x} + rx \qquad (5.4)$$

Let w' correspond to x', $f' = \mathrm{Gr}\ x'$. Then $(x', w'; f')$ is an A-forest triple of (2.2), and (5.1) holds according to the convexity of the objective function and (5.2). In general, we don't know whether strict inequality holds in (5.1) or not since r may be 1. However, we have

Theorem 5.3 In the above case, if $J := \{ (i,j) \in I \mid x_{ij} = 0 \} = \emptyset$, in particular if $(x, w; f)$ is an A-forest triple, then

$$cx' + \phi(w') < cx + \phi(w) \qquad (5.5)$$

and

$$f' \subseteq f,\ f' \neq f. \qquad (5.6)$$

5. Base A-Forest Triples and Cutting

In the forest iteration method for solving the STP, the concept of the base forest triple plays a fundamental role. Here we have

Definition 5.1 If (x,w; f) is an A-forest triple of (2.2) and the corresponding part of (x,w) is an optimal solution of (4.1) associated with f, then f is called a **base A-forest** of (2.2) and (x,w; f) is called a **base A-forest triple** of (2.2). □

From an A-forest triple, we need a method to get a base A-forest triple with a lower objective value. The technique is again called cutting. We now need to extend the concept of A-forest triple.

Definition 5.2 If (x,w) is a feasible point of (2.2) and $f \supseteq \mathrm{Gr}\ x$ is an A-forest, then we call (x,w; f) a **generalized A-forest triple** of (2.2). □

Cutting Suppose we have a generalized A-forest triple (x,w; f) of (2.2). We solve (4.1) on f. If f is a base A-forest, then we can get a nonnegative optimal solution (x′,w′) of (4.1) such that (x′,w′; f) is a base A-forest triple and

$$cx' + \phi(w') \leq cx + \phi(w) \tag{5.1}$$

6. Connecting and Pivoting

If $(x, w; f)$ is a base A-forest triple, then there are $u \in R^m$, $v \in R^n$ such that:

$$\sum_{j \atop (i,j)\in\bar{S}} x_{ij} = a_i, \quad i = 1,\ldots,m,$$

$$\sum_{i \atop (i,j)\in S} r_{ij} x_{ij} = w_j, \quad j = 1,\ldots,n,$$

$$x_{ij} \geq 0, \quad \text{all } (i,j) \in \bar{S},$$

$$u_i + r_{ij} v_j = c_{ij}, \quad (i,j) \in f, \tag{6.1}$$

$$u_i = 0, \quad \text{for } (i,n+1) \in f,$$

$$x_{ij} = 0, \quad \text{all } (i,j) \in \bar{S} - f,$$

$$-v_j \in \partial\phi_j(w_j), \quad j = 1,\ldots,n.$$

Comparing (6.1) with (2.3), we see that if

$$u_i + r_{ij} v_j \leq c_{ij}, \quad (i,j) \in S - f, \tag{6.2}$$

and

$$u_i \leq 0, \quad \text{for } (i,n+1) \in \bar{S} - f, \tag{6.3}$$

then $(x, w; f)$ is an optimal A-forest triple of (2.2). If (6.2) or (6.3) is not satisfied, we need to apply connecting or pivoting. For connecting, we follow an idea similar to Chapter 3: connecting without changing flows.

A. Connecting If in above there is a cell (h, k) violating (6.2), (h, k) is in the joint area of two distinct components of f, and not both of these two components are A-trees, then we simply let $\bar{f} = f \cup \{(h, k)\}$. Or if in above f has no slack variable and there is a cell $(h, n+1)$ violating (6.3), we let $\bar{f} = f \cup \{(h, n+1)\}$. Then, we get a generalized A-forest triple $(x, w; \bar{f})$.

Suppose F is continuous. Similarly to [16], we know ϕ is differentiable and we have

Theorem 6.1 (Connecting) In the above, if we apply cutting to $(x, w; f)$ once, we shall get an A-forest triple $(x', w'; f')$ such that

$$cx' + \phi(w') < cx + \phi(w) \tag{6.4}$$

and

$$(h, k) \subseteq f'. \tag{6.5}$$

where $k = n+1$ in the second case.

Proof Suppose h is in the row index set of a component of f. Then there is a cell $(h, p) \in f$. Let δ be a small positive number and

$$\bar{x}_{hk}(\delta) = \delta, \qquad \bar{w}_k(\delta) = w_k + r_{hk}\delta,$$

$$\bar{x}_{hp}(\delta) = x_{hp} - \delta, \qquad \bar{w}_p(\delta) = w_p - r_{hp}\delta,$$

$$\bar{x}_{ij}(\delta) = x_{ij}, \qquad \bar{w}_j(\delta) = w_j,$$

for other (i,j)'s and j's. For δ small enough, since $x_{hp} > 0$, we know that $(\bar{x}(\delta), \bar{w}(\delta); \bar{f})$ is an A-forest triple, where $\bar{f} = \text{Gr}\ \bar{x}(\delta) = f \cup \{(h,k)\}$. Now, we prove that for δ small enough,

$$c\bar{x}(\delta) + \phi(\bar{w}(\delta)) < cx + \phi(w). \tag{6.6}$$

As in [16], we see that

$$\begin{aligned} e(\delta) &:= (c\bar{x}(\delta) + \phi(\bar{w}(\delta))) - (cx + \phi(w)) \\ &= (c_{hk} - c_{hp})\delta + \phi_k(w_k + r_{hk}\delta) - \phi_k(w_k) \\ &+ \phi_p(w_p - r_{hp}\delta) - \phi_p(w_p) \end{aligned}$$

is a differentiable convex function of δ and

$$\begin{aligned} e'(0) &:= c_{hk} - c_{hp} + r_{hk}\phi'_k(w_k) - r_{hp}\phi'_p(w_p) \\ &= c_{hk} - u_h - r_{hp}v_p + r_{hk}\phi'_k(w_k) - r_{hp}\phi'_p(w_p) \\ &= c_{hk} - u_h - r_{hk}v_k + r_{hk}(v_k + \phi'_k(w_k)) - r_{hp}(v_p + \phi'_p(w_p)). \end{aligned}$$

The second equality of $e(0)$ is due to $(h,p) \in f$ and (6.1). According to (6.1),

$$-v_k = \phi'_k(w_k), \qquad -v_p = \phi'_p(w_p).$$

Therefore,

$$e'(0) = c_{hk} - u_h - r_{hk} v_k < 0 .$$

This proves that (6.6) holds for δ small enough.

We now apply cutting on $(x, w; \bar{f})$. If $\bar{f}$ is a base A-forest, then (4.1) has an optimal solution (x', w') such that $x', w'; f' = \bar{f})$ is an A-forest triple with lower objective value and satisfies (6.5). Suppose that $\bar{f}$ is not a base A-forest and that we find an optimal solution $(\hat{x}, \hat{w})$ of (4.1) on $\bar{f}$. Then

$$c\hat{x} + \phi(\hat{w}) < cx + \phi(w). \qquad (6.7)$$

We claim that

$$\hat{x}_{hk} \geq 0. \qquad (6.8)$$

Suppose we get an A-forest triple $(\dot{x}(\delta), \dot{w}(\delta); f(\delta))$ by applying cutting to $(\bar{x}(\delta), \bar{w}(\delta); \bar{f})$. By (5.5) and (6.6), we have

$$c\dot{x}(\delta) + \phi(\dot{w}(\delta)) < c\bar{x}(\delta) + \phi(\bar{w}(\delta)) < cx + \phi(w). \qquad (6.9)$$

If (6.8) does not hold, then when δ is small enough, we have

$$\dot{x}_{hk}(\delta) = 0.$$

This implies that

$$\dot{f}(\delta) \subseteq f.$$

Since (x, w; f) is a base A-forest triple and $(\dot{x}(\delta), \dot{w}(\delta))$ is nonnegative, we know that

$$cx + \phi(w) \leq c\dot{x} + \phi(\dot{w}(\delta)). \qquad (6.10)$$

(6.9) and (6.10) are contradicted to each other. Therefore, (6.8) holds. This proves that $J = \emptyset$ in Theorem 5.3. According to Theorem 5.3, we know that if we apply cutting to (x, w; f), we can get an A-forest triple (x′, w′; f′) such that (6.4) holds. According to (6.8), we know that (6.5) also holds. This completes our proof. □

B. Pivoting If (x, w; f) is a base A-forest triple but (6.2) or (6.3) is violated at a cell (h, k), and if (h, k) is not located in a location described in cutting, then we do pivoting as described on pages 418-419 of [5]: Assume x_{hk} is increased into θ. We have $\Delta w_k = -r_{hk}\theta$, $\Delta a_h = -\theta$, other Δa_i's and Δw_j's are zeros. Then we can solve Δx_{ij} on the one or two components of f, whose area or joint area (h, k) is, in term of θ multiplied by some real numbers. Since x_{ij}'s on f are positive, we can determine θ and an exit cell on f. Then we can make the changes in x.

We have

Theorem 6.2 (Pivoting) In pivoting described above, we get a new A-forest triple such that w is not changed, that the objective value is strictly decreased by a quantity of $(u_h + r_{hk} v_k - c_{hk})\theta$, or $u_{h,n+1}\theta$ correspondingly, where θ is positive.

Proof The key point to prove is that we can get an strict decrease of the objective value. It suffices to prove that the current value of x is not an optimal solution of T(w). Consider the area of the subgraph including (h,k). Consider the minimization problem of fixing x_{ij}'s out of this area and fixing all w_j's in (2.2). This is a linear programming problem. We know that the current value of x is not an optimal solution of this subproblem since (6.2) or (6.3) is violated at (h,k) and since the current basic solution for this subproblem is positive, i.e., nondegenerate. Therefore, the current value of x is not an optimal solution of T(w). The expression of the decrease of the objective value is obtained from the theory of linear programming too. This proves our theorem. □

Example for pivoting Suppose f is the A-forest in Fig. 1 and (h,k) = (7,6). Assume that x_{76} is increased to θ. Then we put $\Delta a_7 = -\theta$, $\Delta w_6 = -r_{76}\theta$, other Δa_i's and other Δw_j's zeros. We have system for Δx_{ij}'s:

$$\Delta x_{42} + \Delta x_{47} = 0,$$

$$\Delta x_{77} + \Delta x_{78} = -\theta,$$

$$\Delta x_{86} + \Delta x_{88} = 0,$$

$$\Delta x_{92} + \Delta x_{98} = 0,$$

$$r_{42}\Delta x_{42} + r_{92}\Delta x_{92} = 0,$$

$$r_{86}\Delta x_{86} = -r_{76}\theta,$$

$$r_{47}\Delta x_{47} + r_{77}\Delta x_{77} = 0,$$

$$r_{78}\Delta x_{78} + r_{88}\Delta x_{88} + r_{98}\Delta x_{98} = 0.$$

After solving Δx_{86} and Δx_{88}, we get a nonlinear system on the loop. We can choose one Δx_{ij}, say Δx_{42} as a parameter. Then it can be used in turn to express other Δx_{ij}'s and back to Δx_{42}. This gives an equation in Δx_{42} alone, and the latter can be numerically evaluated. Substituting Δx_{42} to the expressions of other Δx_{ij}'s, we can evaluate other Δx_{ij}'s numerically. Now all Δx_{ij}'s are expressed by θ homogeneously. Comparing them with current x_{ij}'s, we can determine the value of θ and an exit cell.

7. The A-Forest Iteration Algorithm

Now we can give the algorithm and its convergence theorem.

Algorithm 7.1 (A-Forest Iteration Algorithm)

(1) Starting from an estimate w of the optimal w^*, solve T(w) to get an A-forest triple (x, w; f). A convenient estimate of w is the expected value of the demand.

(2) Apply repeated cuttings to get a base forest triple with a lower objective value.

(3) Check whether (6.2) and (6.3) are satisfied or not. If they are satisfied, an optimal A-forest triple is in hand. Stop. If they are not, do connecting or pivoting. Go to Step 2. □ .

Theorem 7.2 (Convergence Theorem) If F is continuous, then Algorithm 7.1 converges in finitely many steps.

Proof According to Theorem 6.1 and Theorem 6.2, we have a strict decrease of the objective value from one base A-forest triple to another base A-forest triple. Since the number of base A-forest triples is finite, we get our conclusion. □

8. Sensitivity Analyses

Suppose we have an optimal A-forest triple (x,w; f) with multipliers u and v. We discuss when the optimal A-forest will remain an optimal A-forest under perturbations of the data. We divide the perturbations of data into several cases:

(A). Perturbation of a direct cost coefficient c_{hk}.

This case is the same as the case for the STP. Therefore, we only list the results.

(1). (h,k) is not in f.

If we increase c_{hk}, (x,w; f) is still an optimal A-forest triple since (2.3) still holds. On the other hand, the maximal decrease of c_{hk} such that the optimality is not changed is $c_{hk} - u_h - r_{hk}v_k$. When we decrease c_{hk} greater than $c_{hk} - u_h - r_{hk}v_k$, (x,w; f) loses optimality and iteration techniques described in Sections 5 and 6 are needed to get an optimal A-forest triple.

(2). (h,k) is column non-corner cell of f, i.e., (h,k) is in f and in the h-th row there is no other cell of f, e.g., (2,2) and (3,7) in Fig.1.

The behavior is the converse of (1). If we decrease c_{hk}, (x,w; f) is still an optimal solution. However, the objective value decreases by a quantity equal to the change of $c_{hk}x_{hk}$. The maximal increase of c_{hk} such that the optimality is not changed is min { $c_{hj} - u_h - r_{hk}v_k$ | $j = 1,2,\ldots,n+1$, $j \neq k$ }. Otherwise, iteration techniques are needed to get an optimal A-forest triple.

(3). (h,k) is in f and (2) does not hold.

We should use the method described in Section 4.4 to recalculate u, v, w and x on the component of f where (h,k) is, then check optimality by (2.3). If (2.3) holds with the new u, v and x, then f is still an optimal A-forest with new (x,w). If (2.3) does not hold, then iteration techniques are needed to get an optimal A-forest triple.

(B). Perturbation of a transformation coefficient r_{hk}, where (h,k) is not in f.

We confine the discussion to the SGTP*. Suppose w_k is positive. Then there is a cell $(p,k) \in f$ such that

$$u_h + r_{pk}v_k = c_{pk}.$$

Since u_p is nonpositive, since c_{pk} is nonnegative and since r_{pk} is positive, we know that v_k is nonnegative. If v_k is zero, whatever r_{hk} is, there is no change in optimality. Suppose v_k is positive. If we decrease r_{hk}, (x,w; f) is still an optimal A-forest triple since (2.3) still holds. Similarly, we know that the maximal increase of r_{hk} such that the optimality is not changed is $(c_{hk} - u_h - r_{hk}v_k)/v_k$. Otherwise, iteration techniques are needed to get a new optimal A-forest triple.

(C). Other cases.

There are three other cases: perturbation of a transformation coefficient r_{hk}, where (h,k) is in f; perturbation of a resource a_i; perturbation of a penalty coefficient q_k^+ or q_k^-. In all these cases, we face the same situation of (A)(3). We need to use the method described in Section 4.4 to recalculate u, v, w and x on the component of f, where (h,k) is, then check optimality by (2.3). If (2.3) holds with the new u, v and x, then f is still an optimal A-forest with new (x,w). If (2.3) does not hold, then iteration techniques are needed to get an optimal A-forest triple.

A numerical illustration is given in Section 10.

9. An Example

In [6], Elmaghraby gave a numerical example which is a modification of the problem of allocation of aircraft to routes, presented by Ferguson and Dantzig in [8].

The problem, simply stated, is the following. An airline company operates more than one route, and has available more than one type of aircraft. Each type has its relevant capacity and costs of operation. The demand on each route is known only in the form of a distribution function, and the question asked is: which aircraft should be allocated to which route in order to minimize the expected total cost of operation? This latter involves two kinds of costs: the direct costs connected with running and servicing an aircraft, and the penalty costs incurred whenever a passenger is denied transportation because of lack of seating capacity. However, there is no salvage cost of excess seating capacity, i.e., $q_j^+ = 0$ in (2.1).

In this example, $m = 4$, $n = 5$, $S = \{(1,1), (1,2), (1,3), (1,4), (1,5), (2,2), (2,3), (2,4), (2,5), (3,2), (3,4), (3,5), (4,1), (4,2), (4,3), (4,4), (4,5)\}$. Elmaghraby gave his data with an optimal allocation (x, w), when the demand is considered fixed at its expected value, as follows:

Aircraft i	routes j					Available Aircraft a
	1	2	3	4	5	
1	x 10 r 16 c 18	x 0 r 15 c 21	x 0 r 28 c 18	x 0 r 23 c 16	x 0 r 81 c 10	10
2		x 8 r 10 c 15	x 5 r 14 c 16	x 6 r 15 c 14	x 0 r 57 c 9	19
3		x 7.8 r 5 c 10		x 0 r 7 c 9	x 17.2 r 29 c 6	25
4	x 10 r 9 c 17	x 0 r 11 c 16	x 5 r 22 c 17	x 0 r 17 c 15	x 0 r 55 c 10	15
q	13	13	7	7	1	
w	250	119	180	90	498.8	

Table 1

where x_{ij} is in units of aircraft; r_{ij} and w_j are in 100 passengers; c_{ij} is in \$1000 units, q_j^- is in \$10 units. This makes the unit of $c_{ij}\,x_{ij}$ and the unit of $q_j^-\,w_j$ the same so that we can sum up them in the objective function without consideration of units. Here we omit all the subscripts in our tables. Therefore, a, x, w, u, v, r, c, q and p represent a_i, x_{ij}, w_j, u_i, v_j, r_{ij}, c_{ij}, q_j^- and p_j correspondingly.

Let p_j be the probability density function of the demand for the jth product. Elmaghraby's data are as follows:

Route j	Interval(In hundreds of passengers)	density p
1	190 - 210	0.01
	210 - 240	0.005 / 3
	240 - 260	0.0175
	260 - 290	0.02 / 3
	290 - 310	0.01
2	0 - 100	0.003
	100 - 200	0.007
3	130 - 150	0.005
	150 - 170	0.010
	170 - 190	0.020
	190 - 210	0.010
	210 - 230	0.005
4	0 - 30	0.02 / 3
	30 - 70	0.005
	70 - 90	0.015
	90 - 110	0.010
	330 - 350	0.005
5	570 - 590	0.005
	590 - 610	0.040
	610 - 630	0.005

Table 2

In other places, the density functions are zeros.

Elmaghraby used the (x, w) in the first table as his starting point. As expected, $f = \text{Gr } x$ is already an optimal forest. In [6], Elmaghraby noticed that the cells containing a positive allocation are identical in his starting point and the optimal solution. He used the term configuration and defined it as any pattern of positive and zero cells in the tableau. Thus, he said in a matrix of N cells there were all possible configurations corresponding to the 2^N possible allocations of positive or zero entries in the cells.

He suggested assuming the configuration at the outset and solving the set of simultaneous equations. What he did not notice in [6] is that only some special configurations should be treated: that is, A-forests. We first see numerically how we can get the optimal solution by the method given in Section 4 if we know the optimal forest.

The A-forest f = Gr x is a one-tree forest. It does not contain slack variables. Suppose $v_1 = d$. By the third set of equations of (4.4), we get

$$u_1 = 18 - 16d, \qquad u_2 = 16 - 63d/11,$$
$$u_3 = 10.5 - 63d/22, \qquad u_4 = 17 - 9d,$$
$$v_2 = -0.1 + 63d/110, \qquad v_3 = 9d/22, \qquad (9.1)$$
$$v_4 = -2/15 + 21d/55, \qquad v_5 = -9/58 + 63d/638.$$

By the first and the second sets of equations of (4.4), we get

$$22w_1 + 12.6w_2 + 9w_3 + 8.4w_4 + 63w_5/29 = 10459. \qquad (9.2)$$

This is the numerical realization of (4.5). In this example, ϕ_j is continuously differentiable. According to the fourth set of equations of (4.4), we have

$$v_j = -\phi'_j(w_j), \qquad j = 1,2,3,4,5, \qquad (9.3)$$

where

$$\phi_j(w_j) = q_j^- \int_{w_j}^{\infty} (y - w_j) f_j(y) dy. \qquad (9.4)$$

Solving above equations, we get d = 9.8311. The values of u, v and $u_i + r_{ij} v_j - c_{ij}$ are as follows:

i	u + v - c					u
	j= 1	j= 2	j= 3	j= 4	j= 5	
1	0	-77.340	-44.687	-72.082	-82.232	-139.2976
2		0	0	0	-28.816	-40.3054
3		0		-1.310	0	-17.6527
4	0	-26.644	0	-24.933	-36.621	-71.4799
v	9.8311	5.5305	4.0218	3.6204	0.8156	

Table 3

From (9.3) and (9.4), we get w_j. Solving the first and the second sets of equations of (4.4), we get x_{ij}. The results are as follows:

i	x					a
	j= 1	j= 2	j= 3	j= 4	j= 5	
1	10	0	0	0	0	10
2		11.631	2.334	5.035	0	19
3		4.582		0	20.418	25
4	8.473	0	6.527	0	0	15
w	236.527	139.225	176.273	75.520	592.108	

Table 4

We see that all x_{ij}'s are nonnegative and that $u_i + r_{ij} v_j \leq c_{ij}$. for all i and j. Therefore, we have obtained an optimal solution. The optimal cost is \$ 1,699,456. The results are the same as Elmaghraby's (there was a mistake in w_1, therefore also in the optimal cost, in [6]).

10. Sensitivity Analyses of This Example

We can use above example as a numerical illustration of the discussions in Sections 5, 6 and 8.

(I). The cases when (x,w; f) remains an optimal forest triple.

There are three cases discussed in Section 4.8, when (x,w; f) remains an optimal forest triple, i.e., (A).(1), (A).(2), (B).

(1). Perturbation of a direct cost coefficient c_{hk}, where (h,k) is not in f.

From Table 3, we see that: if (h,k) $\in$ {(1,2), (1,3), (1,4), (1,5), (4,2), (4,4), (4,5)}, c_{hk} can be any nonnegative number without changing the optimality of (x,w; f). For (h,k) = (2,5), c_{25} can be changed by $\Delta \in [-2.8156, +\infty)$; for (h,k) = (3,4), c_{34} can be changed by $\Delta \in [-1.3099, +\infty)$ without changing the optimality of (x,w; f). There is no change of the optimal cost in this case.

(2). Perturbation of a direct cost coefficient c_{hk}, where (h,k) is a column non-corner cell of f.

There is only one such a cell in our example: (h,k) = (1,1). Check Table 3. We know that (x,w; f) will remain an optimal forest triple if we decrease c_{11} to any extent or if we increase c_{11} by less than 44.6872. The optimal cost will be changed with a quantity of the change of $10c_{11}$.

(3). Perturbation of a transformation coefficient r_{hk}, where (h,k) is not in f.

As discussed in Section 8, any decrease of r_{hk} will not affect the optimality of (x,w; f). The maximal increase of c such that the optimality is not changed is $(c_{hk} - u_h - r_{hk} v_k) / v_k$, which is listed as follows:

i	j				
	1	2	3	4	5
1		13.9843	11.1112	19.9101	100.8284
2					3.4514
3				0.3618	
4		4.8177		6.8868	44.9012

Table 5

There is no change of the optimal cost in this case.

(II). f remains an optimal forest but (x,w) is changed.

This covers the cases of small perturbations of a direct cost coefficient c_{hk}, where (h,k) is in f and is not a column non-corner cell, or a transformation coefficient r_{hk}, where (h,k) is in f, or a source a_h, or a penalty coefficient q_k^+ or q_k^-. For example, suppose c_{22} is changed from 15 to 16. This does not change (9.2) but changes (9.1) into:

$u_1 = 18 - 16d,$ $u_2 = 16 - 63d/11,$

$u_3 = 10 - 63d/22,$ $u_4 = 17 - 9d,$

$v_2 = 63d/110,$ $v_3 = 9d/22.$ (10.1)

$v_4 = -2/15 + 21d/55,$ $v_5 = -4/29 + 63d/638.$

Combining (10.1) with (9.2), (9.3) and (9.4), we get

i	u + v - c					u
	j = 1	j = 2	j = 3	j = 4	j = 5	
1	0	-75.725	-44.617	-71.918	-81.713	-139.0511
2		0	0	0	-1.920	-46.2172
3		0		-1.351	0	-18.1086
4	0	-25.641	0	-25.034	-35.757	-71.3413
v	9.8157	5.6217	4.0155	3.6145	0.8313	

Table 6

i	x					a
	j = 1	j = 2	j = 3	j = 4	j = 5	
1	10	0	0	0	0	10
2		11.524	2.438	5.038	0	19
3		4.596		0	20.404	25
4	8.552	0	6.448	0	0	15
w	236.968	138.223	176.318	75.576	591.716	

Table 7

We see that all x_{ij}'s are nonnegative and that $u_i + r_{ij} v_j \leq c_{ij}$, for all i and j. Therefore, f is still an optimal forest with a slightly changed (x, w).

The optimal cost is $1,709,679. The original (x,w) is still feasible with an objective value $1,715,080. The change in x reflects that the optimal forest is not changed but the flows are dispersed from cell (2,2) to other cells of the forest f to balance the rise in the cost of (2,2).

(III). f is no longer an optimal forest when the perturbation is big enough.

For example, we decrease c_{34} from 9 to 7. This makes cell (3,4) cheap enough to enter the optimal forest. We first do a pivoting to let cell (3,4) enter and to let cell (3,2) exit. We get a new feasible solution:

i	x					a
	j = 1	j = 2	j = 3	j = 4	j = 5	
1	10	0	0	0	0	10
2		13.770	2.334	2.896	0	19
3		0		4.582	20.418	25
4	8.473	0	6.527	0	0	15
w	236.257	137.770	176.273	75.520	592.108	

Table 8

The objective value is now $1,696,383, which is less than the original objective value $1,699,456. Minimizing on the new forest, we get new x, w, u and v as follows:

i	u + v - c					u
	j = 1	j = 2	j = 3	j = 4	j = 5	
1	0	-77.367	-44.709	-72.057	-81.340	-139.3559
2		0	0	0	-1.463	-40.3263
3		-0.689		0	0	-17.5127
4	0	-26.654	0	-24.944	-35.329	-71.5127
v	9.8347	5.5326	4.0231	3.6217	0.8397	

Table 9

i	x					a
	j = 1	j = 2	j = 3	j = 4	j = 5	
1	10	0	0	0	0	10
2		13.810	2.304	2.886	0	19
3		0		4.603	20.397	25
4	8.454	0	6.546	0	0	15
w	236.089	138.103	176.262	75.507	591.507	

Table 10

Again, we know that we have got an optimal solution. since all x_{ij}'s are nonnegative and since all $u_i + r_{ij} v_j - c_{ij}$ are nonpositive. The optimal objective value is $1,696,353 now. It is a little less than the objective value we have got in the last tableau.

References

[1] V. Balachandran, "Generalized transportation networks with stochastic demands: an operator theoretic approach", Networks 9, pp.169-184 (1979).

[2] M. Bazaraa and C. M. Shetty, "Nonlinear Programming Theory and Algorithms", Wiley, New York, 1979.

[3] L. Cooper and L. J. LeBlanc, "Stochastic transportation problems and other network related convex problems", Naval Res. Logist. Quart., 24, pp.327-336 (1977).

[4] L. Cooper, "The stochastic transportation-location problem", Comput. Math. Appl. 4, pp.265-275 (1978).

[5] G. B. Dantzig, "Linear Programming and Extensions", Princeton University Press, Princeton, 1963.

[6] S. Elmaghraby, "Allocation under uncertainty when the demand has a continuous distribution function", Management Sci. 6, pp.270-294 (1960).

[7] A. R. Ferguson and G. B. Dantzig, "Notes on Linear Programming: Part XVI - The problem of routing aircraft", Aeronaut. Eng. Rev. 14, pp.51-55, 1955.

[8] A. R. Ferguson and G. B. Dantzig, "The allocation of aircraft to routes", Management Sci. 3, pp.45-73, 1956.

[9] L. R. Ford and D. R. Fulkerson, "Flows in Networks", Princeton University Press, Princeton, 1962.

[10] P. E. Gill and W. Murray, "Linearly constrained problems including linear and quadratic programming", In: "The State of the Art in Numerical Analysis" (D. Jacobs, ed.), pp.313-363, Academic Press, New York, 1977.

[11] P. E. Gill, W. Murray and M. H. Wright, "Practical optimization", Academic Press, New York, 1981.

[12] P. Kall, "Stochastic Linear Programming", Springer-Verlag, Berlin, 1976.

[13] J. L. Kennington and R. V. Helgason, "Algorithms for Network Programming", John Wiley & Sons, New York, 1980.

[14] L. Qi, "Base set strategy for solving linearly constrained convex programs", Comp. Sci. Tech. Rept. #505, University of Wisconsin-Madison, 1983.

[15] L. Qi, "An alternating method to solve stochastic programming with simple recourse", Comp. Sci. Tech. Rept. #515, University of Wisconsin-Madison, 1983. to appear in: Math. Programming Study.

[16] L. Qi, "Forest iteration method for stochastic transportation problem", Comp. Sci. Tech. Rept. #522, University of Wisconsin-Madison, 1983. To appear in: Math. Programming Study.

[17] L. Qi, "Finitely Convergent Methods for Solving Stochastic Linear Programming and Stochastic Network Flow Problems", Ph.D. Thesis, Computer Sciences, University of Wisconsin-Madison, 1984.

[18] S. M. Robinson, "Convex Programming", unpublished notes, 1981.

[19] R. T. Rockafellar, "Convex Analysis", Princeton University Press, Princeton, 1981.

[20] R. T. Rockafellar, "Monotropic programming: descent algorithms and duality", in "Nonlinear Programming 4", O. L. Mangasarian, R. R. Meyer and S. M. Robinson, eds., pp. 327-366, Academic Press, New York, 1981.

[21] I. M. Stancu-Minasian and M. J. Wets, "A research bibliography in stochastic programming 1955-1975", Oper. Res. 24, pp.1078-1179, 1976.

[22] B. Strazicky, "Some results concerning an algorithm for the discrete recourse problem", in "Stochastic Programming", (M. A. H. Dempster, ed.), pp. 263-274, 1980.

[23] W. Szwarc, "The transportation problem with stochastic demands", Management Sci. 11, pp.33-50 (1964).

[24] H. M. Wagner, "Principles of Operation Research with Applications to Managerial Decisions", Prentice-Hall, Englewood Cliffs, N. J., 1975.

[25] R. J-B. Wets, "Solving Stochastic Programs with Simple Recourse, I", Dept. of Math., University of Kentucky, 1974.

[26] R. J-B. Wets, "Stochastic programs with fixed recourse: the equivalent deterministic problem", SIAM Review 16, pp.309-339, 1974.

[27] R. J-B. Wets, "Solving Stochastic Programs with Simple Recourse", Stochastics 10, pp.219-242(1983).

[28] A. C. Williams, "A stochastic transportation problem", Oper. Res. 11, pp. 759-770 (1963).

[29] D. Wilson, "On a priori bounded model for transportation problem with stochastic demand and integer solutions", AIEE Transaction 4, pp. 186-193 (1972).

[30] D. Wilson, "Tight bounds for stochastic transportation problems", AIEE Transaction 5, pp. 180-185 (1973).

[31] D. Wilson, "A mean cost approximation for transportation problems with stochastic demand", Naval Res. Logist. Quart. 22, pp. 181-187 (1975).

[32] W. T. Ziemba, "Computational algorithms for convex stochastic programs with simple recourse", Oper. Res. 18, pp. 414-431 (1970).