TITLE:

An Algorithm for the Assignment Problem with Stochastic Side Constraints

AUTHOR(S):

MINE, Hisashi; FUKUSHIMA, Masao; ISHIKAWA, Kenji; SAWA, Isao

CITATION:

MINE, Hisashi ...[et al]. An Algorithm for the Assignment Problem with Stochastic Side Constraints. Memoirs of the Faculty of Engineering, Kyoto University 1984, 45(4): 26-35

ISSUE DATE:

1984-01-31

URL:

http://hdl.handle.net/2433/281255

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

An Algorithm for the Assignment Problem with Stochastic Side Constraints

By

Hisashi MINE*, Masao FUKUSHIMA*, Kenji ISHIKAWA**
and Isao SAWA***

(Received June 29, 1983)

Abstract

In this paper, we consider the assignment problem with stochastic side constraints, and propose a practical algorithm for solving it. Such a problem may arise, for example, when the assignment requires some scarce resources and the total amounts of those resources are subject to a random variation. Therefore, the problem seems quite general and significant in practice. This algorithm takes a special structure of the problem into account, and may be regarded as a heuristic modification of the method for two-stage linear programming under uncertainty. Although we cannot guarantee that the solution obtained by the proposed algorithm will coincide with the true optimal solution of the problem, our limited computational experience on small test problems indicates that good approximate solutions can be obtained in a fairly small computation time.

1. Introduction

The classical assignment problem is formulated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1.1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_{ij}=1 \quad \text{for } i=1, 2, \ldots, n, \tag{1.2}$$

$$\sum_{i=1}^{n} x_{ij}=1 \quad \text{for } j=1, 2, \ldots, n, \tag{1.3}$$

$$x_{ij}=0 \text{ or } 1 \quad \text{for all } i \text{ and } j, \tag{1.4}$$

where condition (1.4) implies

$$x_{ij}=\begin{cases} 1 & \text{if man } i \text{ is assigned to job } j \\ 0 & \text{otherwise.} \end{cases}$$

(1.2) and (1.3) require that there is a one-to-one assignment from the set of men

* Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University
** Sumitomo Metal Industries Ltd.
*** Information Processing Center, Kansai University

to that of jobs. The assignment problem is one of the most fundamental topics in network flow theory, and has been extensively studied in the context of linear programming. In particular, a number of effective solution techniques, e. g., [1] [2] [6] [8], have been developed and are widely used to solve various problems in practice.

In the classical assignment problem, the cost is minimized (or the profit is maximized) under the only condition that man i should be assigned to job j one to one. In practical applications, however, it may sometimes be necessary to incorporate additional constraints in the problem [4]. For example, suppose that we need some resources when we assign man i to job j. Then, we have to consider such constraints whereby the total amount of each resource to be used should meet the available amount of that resource. Furthermore, we suppose that the amount of each resource to be supplied is not known exactly when the decision is made, i. e., the right-hand-side of those constraints is a random vector. Then, the problem may be stated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{2. 1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_{ij}=1 \qquad \text{for } i=1,\ldots, n, \tag{2. 2}$$

$$\sum_{i=1}^{n} x_{ij}=1 \qquad \text{for } j=1,\ldots, n, \tag{2. 3}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} t_{ijk}x_{ij}=\xi_k \qquad \text{for } k=1,\ldots, m, \tag{2. 4}$$

$$x_{ij}=0 \text{ or } 1 \qquad \text{for all } i \text{ and } j, \tag{2. 5}$$

or, using the vector-matrix notation,

$$\text{Minimize} \quad cx \tag{3. 1}$$

$$\text{subject to} \quad Ax=b, \tag{3. 2}$$

$$Tx=\xi \tag{3. 3}$$

$$x\in X, \tag{3. 4}$$

where constraint (3. 2) corresponds to (2. 2)-(2. 3), constraints (3. 3) and (3. 4) correspond to (2. 4) and (2. 5), respectively, and $\xi=(\xi_1,\ldots, \xi_m)$ is a random vector.

Problem (3) can be regarded as a stochastic programming problem, for which various formulations have been proposed to obtain its determinist equivalent. One of such formulations is the so called two-stage program which can be stated in its simplest form as follows:

$$\text{Minimize} \quad cx+E_\xi\{\inf\ q^+y^+ + q^-y^- \mid x\} \tag{4. 1}$$

$$\text{subject to} \quad Ax=b, \tag{4. 2}$$

$$Tx+y^+-y^-=\xi, \tag{4. 3}$$

$$x\in X,\ y^+\geq 0,\ y^-\geq 0, \tag{4. 4}$$

where y^+ and y^- are *recourse* vector, q^+ and q^- are constant vectors such that $q^+ + q^- \geq 0$, and $E_\xi\{\cdot\}$ means expectation. The problem of this type is commonly called a complete problem [12], or a stochastic program with simple recourse [15]. It is interpreted as follows : The decision maker first selects an assignment, say $x=\hat{x}$, then he observes the random event $\xi=\hat{\xi}$, and finally he takes a corrective action (y^+, y^-) in such a way that $q^+y^+ + q^-y^-$ is minimized, subject to $y^+ - y^- = \hat{\xi} - T\hat{x}$, $y^+ \geq 0$, $y^- \geq 0$.

If the elements of x are not restricted to be 0-1 but are allowed to take continuous values, then problem (4) becomes a problem which is well known as the two-stage liner program under uncertainty. It may be solved by a method such as the one proposed in [12]. However, since the solutions of the assignment problem have to satisfy the 0-1 condition, problem (4) is considered a two-stage stochastic program in which the first stage variables, x_{ij}, are 0-1. For such problems, Wollmer [13] proposes an algorithm that combines the Van Slyke-Wets method [11] and an implicit enumeration method [5]. Yudin and Tsoy [14] also propose an algorithm for discrete stochastic programs under a slightly different setting.

As far as problem (4) is concerned, however, the deterministic constraint (4. 2) is merely the constraint of the ordinary assignment problem. Taking this special structure of the problem into account, we propose in this paper a practical algorthm for solving problem (4) approximately. The algorithm can be viewed as a heuristic modification of the algorithm for two-stage linear programs under uncertainty, developed by Van Slyke and Wets [11]. Although we cannot guarantee that the solution obtained by the proposed algorithm will coincide with the true optimal solution of problem (4), our limited computational experience on small test problems indicates that good approximate solutions can be obtained in a fairly small computation time.

2. Basic Properties of Two-Stage Stochastic Programs

We can rewrite problem (4) as the following equivalent problem:

$$\begin{aligned} &\text{Minimize} && cx + Q(x) \\ &\text{subject to} && Ax = b, \\ & && x \in X, \end{aligned} \tag{5}$$

where

$$Q(x) = E_\xi\{\inf\ q^+y^+ + q^-y^- \mid y^+ - y^- = \xi - Tx,\ y^+ \geq 0,\ y^- \geq 0\}.$$

In what follows, we assume that $E_{\xi_i}\{\xi_i\}$ exists for each i. Then, the function Q is everywhere finite and convex, and is expressed as

$$\begin{aligned} Q(x) &= \sum_{i=1}^{m} Q_i[(Tx)_i] \\ &= \sum_{i=1}^{m}\left\{-q_i^- \int_{\xi_i \leq (Tx)_i} \{\xi_i - (Tx)_i\}\, dF_i(\xi_i) + q_i^+ \int_{\xi_i > (Tx)_i} \{\xi_i - (Tx)_i\}\, dF_i(\xi_i)\right\} \end{aligned} \tag{6}$$

where F_i is the marginal distribution function for ξ_i. (See [12].)

Now let I be a set of indices and $\{x^l : l \in I\}$ be a set of feasible solutions to problem (5). For each x^l, define the vector $\pi^l=(\pi_1^l, \ldots, \pi_m^l)$ and the scalar ρ^l by

$$\left.\begin{aligned} \pi_i^l &= q_i^+ - (q_i^+ + q_i^-) \int_{t_i \le (Tx^l)_i} dF_i(\xi_i), \\ \rho_i^l &= q_i^+ E_{\xi_i}\{\xi_i\} - (q_i^+ + q_i^-) \int_{t_i \le (Tx^l)_i} \xi_i dF_i(\xi_i), \\ \rho^l &= \sum_{i=1}^{m} \rho_i^l. \end{aligned}\right\} \tag{7}$$

Then, π^l and ρ^l thus defined determine a supporting hyperplane of Q at x^l [11]. Namely, we have

$$Q(x) \ge \rho^l - \pi^l T x \qquad \text{for any } x$$

and

$$Q(x^l) = \rho^l - \pi^l T x^l.$$

Therefore, the problem:

$$\begin{aligned} \underset{x,\theta}{\text{Minimize}} \quad & cx+\theta \\ \text{subject to} \quad & \pi^l Tx + \theta \ge \rho^l \qquad \text{for } l \in I, \\ & Ax = b, \\ & x \in X, \end{aligned} \tag{8}$$

is an outer approximation of problem (5) in the sense that the minimum value of problem (8) is no greater than that of problem (5). Moreover, it is not difficult to see that, if the inequality

$$\hat{\theta} \ge Q(\hat{x}), \tag{9}$$

where $(\hat{\theta}, \hat{x})$ is an optimal solution to (8), is satisfied, then $\hat{x}$ is actually an optimal solution of problem (5). On the other hand, if the inequality (9) is violated, then $\hat{x}$ is added to the set $\{x^l\}$. As a result, we obtain a finer outer approximation problem of the form (8).

3. Algorithm

Based on the observations in the previous section, we are now ready to state an algorithm for solving problem (5).

Step 0 : Set $I=\emptyset$ and $l=1$.

Step 1 : Solve the problem

$$\underset{x,\theta}{\text{Minimize}} \quad cx+\theta \tag{10}$$

$$\text{subject to} \quad \pi^r Tx+\theta \geq \rho^r, \quad r \in I,$$
$$Ax=b,$$
$$x \in X,$$

and let (x^l, θ^l) be its optimal solution. Here, if $I=\emptyset$, then θ is set equal to $-\infty$ and is ignored in the computation.

Step 2 : Calculate π^l and ρ^l by (7), and put $Q^l=\rho^l-\pi_l Tx^l$. If $Q^l \leq \theta^l$, then terminate as x^l is an optimal solution to problem (5). Otherwise, set $I=I \cup \{l\}$, increase l by one and return to Step 1.

This algorithm is essentially a simplified version of the one proposed by Van Slyke and Wets [11] for two-stage linear programs under uncertainty. Whenever we return to Step 1, a new constraint is generated and problem (10) is solved with the augmented constraints. Obviously, the algorithm terminates in a finite number of iterations, because if a certain feasible solution is generated twice during the iterations, i.e., $x^l=x^{l'}$ for some $l>l'$, then we must have $\theta^l=Q^l$ in Step 2. That is to say that the stopping criterion is satisfied. Therefore, the algorithm should terminate finitely, since the number of feasible solutions of problem (5) is finite.

Problem (10) is in general a 0-1 mixed integer program which can be solved, for example, by an implicit enumeration technique [5]. However, it seems quite expensive computationally to solve problem (10) by such a technique at each iteration. This is because a large number of iterations is usually required to obtain the termination of the algorithm. To avoid this computational burden, we employ here a simple method of obtaining an approximate solution to problem (10) by taking account of a special structure of the problem. This method is based on the subgradient algorithm [7] [10] for maximizing the dual of problem (10) and is implemented quite easily.

Let us define the Lagrangian for problem (10) by

$$\mathcal{L}(x, \theta, u)=cx+\theta+u(d-Bx-\theta e),$$

where u is the vector of Lagrange multipliers, d is the vector whose elements are ρ^r, $r \in I$, B is the matrix whose rows are $\pi^r T$, $r \in I$, and e is the vector of ones. Then, the Lagrangian dual problem for problem (10) is expressed as follows:

$$\begin{aligned} &\text{Maximize} \quad L(u) \\ &\text{subject to} \quad u \geq 0, \end{aligned} \tag{11}$$

where

$$L(u)=\inf_{x,\theta}\{\mathcal{L}(x, \theta, u) \mid Ax=b, x \in X\}.$$

Since

$$L(u)=ud+\inf_{x}\{(c-uB)x \mid Ax=b,\ x\in X\}+\inf_{\theta}\{(1-ue)\theta\},$$

it is easy to see that $L(u)>-\infty$ if and only if $ue=1$. Therefore, problem (11) is rewritten as

$$\begin{aligned} &\text{Maximize} && ud+\omega(u) \\ &\text{subject to} && ue=1, \\ & && u\geq 0, \end{aligned} \tag{12}$$

where

$$\omega(u)=\min_{x}\{(c-uB)x \mid Ax=b,\ x\in X\}. \tag{13}$$

It is noted that $\omega(u)$ is easily evaluated for each u by solving an ordinary assignment problem. Furthermore, it is not difficult to show that a subgradient of the function ω, which is a concave function, at the point u is given by $-B\bar{x}$, where $\bar{x}$ is an optimal solution of the assignment problem (13). (For the definition of subgradient, see [9].)

On the basis of the preceding arguments, Step 1 of the algorithm is modified as follows:

Step 1 *a*: Select u^0 and a small positive number ε, and set $k=0$.

Step 1 *b*: Solve the assignment problem (13) with $u=u^k$, and let x^k be its optimal solution. Put $g^k=d-Bx^k$.

Step 1 *c*: Update u^k by the formula

$$u^{k+1}=P_U[u^k+\alpha^k(\hat{L}_k-L(u^k))\,g^k/\|g^k\|^2], \tag{14}$$

where α^k is a properly chosen step length [7] [10], $\hat{L}_k$ is an overestimate of the optimal value of problem (12), and P_U is the projection operator on the set $U=\{u \mid ue=1,\ u\geq 0\}$.

Step 1 *d*: If u^{k+1} satisfies $\|u^{k+1}-u^k\|/\|u^k\|<\varepsilon$, then put $x^t=x^k$ and $\theta^t=\min\{\theta \mid \theta e\geq d-Bx^k\}$. Go to Step 2. Otherwise, increase k by one and return to Step 1*d*.

It is noted that the value of $L(u^k)$ in (14) coincides with that of the objective function of problem (12), since u^k is contained in the set U. Thus, $L(u^k)$ is readily evaluated, using $\omega(u^k)$ obtained in Step 1*b*.

By employing this modification, the algorithm may be implemented quite easily, because Steps 1*a*-1*d* only require repeated solutions of ordinary assignment problems. It is noted, however, that the solution obtained by the algorithm, in which Step 1 is replaced by Steps 1*a*-1*d*, may not be an optimal solution to problem (5) in general, since the solution x^t generated by Steps 1*a*-1*d* does not necessarily solve problem (10).

Therefore, in order to obtain a better solution, the whole algorithm is eventually modified as follows:

Step 0: Set $I=\emptyset$, $l=1$ and $v^*=+\infty$.

Steps 1*a*-1*d*: As above.

Step 2: Calculate π^l, ρ^l and Q^l as before and put $v^l=cx^l+Q^l$. If $v^l<v^*$, then set $v^*=v^l$ and $x^*=x^l$. If $Q^l\leq\theta^l$, then terminate. x^* is an approximate soluton to problem (5). Otherwise, set $I=I\cup\{l\}$, increase l by one, and return to Step 1.

It is to be noted that this algorithm also terminates finitely due to the same reason as before.

4. Computational Results

We have implemented the proposed algorithm to solve problem (4). The algorithm was coded in FORTRAN IV, and the subroutine LSAPR [3] was used to solve ordinary assignment problems in Step 1b. The program was run on a FACOM M-200 computer at the Kyoto University Data Processing Center.

In our computational experiments, we solved three groups of small test problems with $n=7$ and $m=3$, each group consisting of ten problems. Each element of the random vector ξ was assumed to be distributed uniformly on the interval $[\alpha, \beta]$, where $[\alpha, \beta]=[30, 40]$ for Group 1, $[\alpha, \beta]=[10, 30]$ for Group 2, and $[\alpha, \beta]=[20, 50]$ for Group 3. Coefficients of the problems, c, q^+, q^-, T, were randomly generated within the ranges [50, 100], [0, 5], [0, 5] and [0, 10], respectively.

The computational results are summarized in Table 1. In order to evaluate the

Table 1.

(a) Group 1 : $[\alpha, \beta]=[30, 40]$

Problem	V^* ($\times 10^3$)	V ($\times 10^3$)	$\frac{V-V^*}{V^*}$ (%)	NCON.	NAS.	C. P. U. TIME (sec)
1	0.42880	0.42880	0.0	4	92	0.118
2	0.42429	0.42429	0.0	4	89	0.119
3	0.43779	0.43816	0.08457	2	37	0.057
4	0.41952	0.41952	0.0	6	129	0.184
5	0.47659	0.47802	0.29948	2	32	0.050
6	0.46421	0.46421	0.0	3	52	0.068
7	0.45512	0.45512	0.0	4	98	0.128
8	0.47342	0.47342	0.0	5	95	0.140
9	0.48807	0.51821	6.17378	2	32	0.043
10	0.44150	0.44999	1.92314	2	43	0.050
Average			0.84810	3.4	69.9	0.0957

(b) Group 2 : $[\alpha, \beta]=[10, 30]$

Problem	V^* ($\times 10^3$)	V ($\times 10^3$)	$\frac{V-V^*}{V^*}$ (%)	NCON.	NAS.	C. P. U. TIME (sec)
1	0.50611	0.50611	0.0	1	2	0.007
2	0.44123	0.44123	0.0	3	27	0.044
3	0.47133	0.47133	0.0	2	3	0.011
4	0.44939	0.44939	0.0	3	32	0.048
5	0.49069	0.49069	0.0	2	3	0.011
6	0.52995	0.52995	0.0	2	12	0.022
7	0.54471	0.54471	0.0	2	3	0.011
8	0.59567	0.60471	1.51756	2	7	0.015
9	0.49132	0.49132	0.0	1	2	0.007
10	0.53764	0.53764	0.0	2	3	0.011
Average			0.15176	2.0	9.4	0.0187

(c) Group 3 : $[\alpha, \beta]=[20, 50]$

Problem	V^* ($\times 10^3$)	V ($\times 10^3$)	$\frac{V-V^*}{V^*}$ (%)	NCON.	NAS.	C. P. U. TIME (sec)
1	0.45608	0.45608	0.0	1	2	0.007
2	0.44151	0.44151	0.0	5	63	0.098
3	0.44473	0.44473	0.0	1	2	0.007
4	0.43913	0.43913	0.0	5	79	0.111
5	0.49473	0.49496	0.04783	3	15	0.030
6	0.50183	0.50183	0.0	3	47	0.064
7	0.49676	0.49676	0.0	4	22	0.041
8	0.50936	0.50936	0.0	2	3	0.010
9	0.51741	0.53077	2.58275	3	48	0.069
10	0.47827	0.47827	0.0	4	33	0.047
Average			0.26306	3.1	31.4	0.0484

V^* : Exact optimal value.
V : Approximate optimal value found by the proposed algorithm.
$(V-V^*)/V^*$: Relative error.
NCON : Total number of constraints generated.
NAS : Total number of assignment problems solved.

proposed algorithm, true optimal solutions for the same test problems were also obtained by enumerating all feasible solutions. The proposed algorithm performed satisfactorily for those small test problems. For most of the problems, the objective value was attained within a few percent of the true optimal value. The numbers of problems for which the true optimal values were actually attained are six for Group 1, nine for Group 2 and eight for Group 3. In terms of the average CPU time and the average numbers of NCON and NAS, Group 1 was the most difficult to solve, while Group 3 was the second and Group 2 was the easiest. The reason for this may be

explained as follows: First, note that the function Q_i in (6) is quadratic on the interval $[\alpha, \beta]$, and is linear outside of this interval. Since the mean value of each element of T is 5.0 and only seven of the variables x_{ij} can take value one for a feasible solution, each component of the vector Tx is expected to take a value of 35.0 approximately. For Group 2, the fact of $35.0 \notin [\alpha, \beta]$ implies that a considerably large portion of the feasible solutions lies on one of the regions in which Q is linear. (See Fig. 1-*b*.) Thus, the function Q for Group 2 may be well approximated by a smaller number of supporting hyperplanes than for other groups. Moreover, comparing

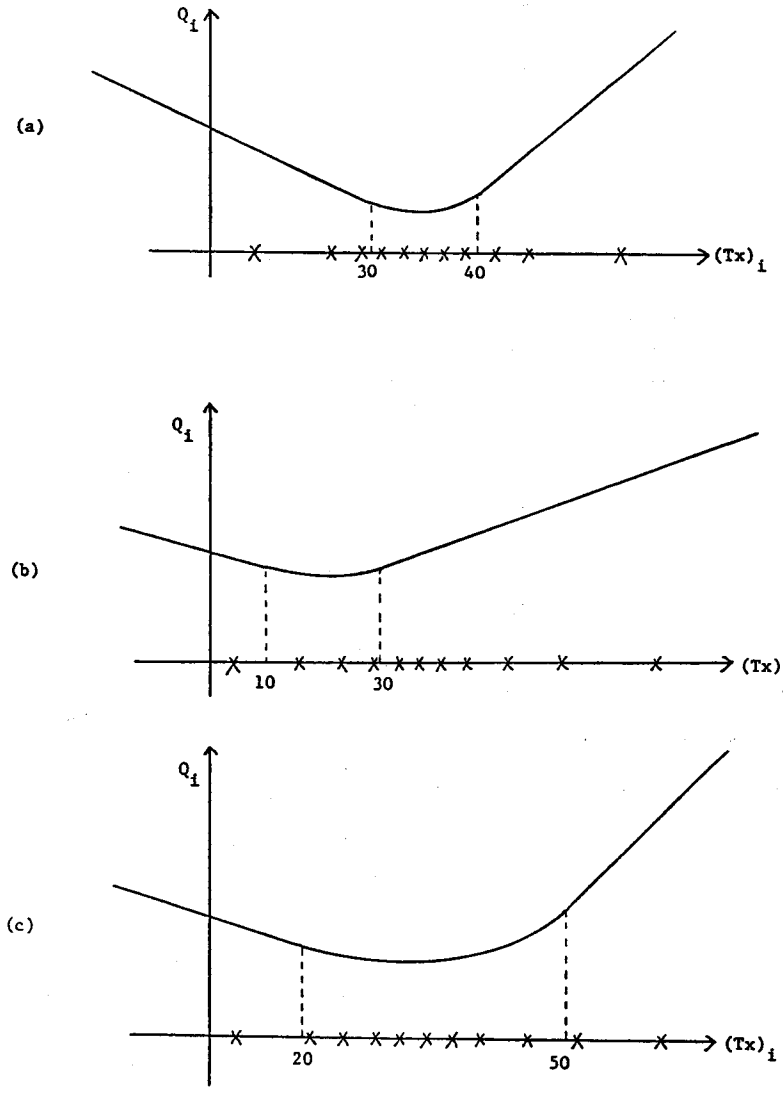

Fig. 1 × : feasible points

Group 1 with Group 3, it may be observed that a larger number of the feasible solutions for Group 1 are scattered in both parts outside the region in which Q is nonlinear (See Fig. 1-*a*, *c*). For that reason, it would be more difficult to get relatively good approximations for Q in Group 1 than in Group 3.

5. Conclusion

We have formulated the assignment problem with stochastic side constraints as a two-stage stochastic program, and proposed an algorithm for obtaining an approximate solution of the problem. We have also made an experiment on small test problems so as to compare the solutions obtained by the algorithm with the true optimal solutions. The computational results were encouraging. Although we have not solved large problems, it might be expected that the proposed algorithm would produce fairly good approximate solutions even for large problems in a relatively small computation time.

Acknowledgment

The authors would like to thank Professor T. Ibaraki for his helpful discussions.

References

1) Barr, R. S., F. Glover and D. Klingman, "The Alternating Basis Algorithm for Assignment Problems", Mathematical Programming, **13**, 1-13 (1977).
2) Bertsekas, D. P., "A New Algorithm for the Assignment Problem", Mathematical Programming, **21**, 152-171 (1981).
3) Burkard, R. E. and U. Derigs, *Assignment and Matching Problems: Solution Methods with FORTRAN-Programs*, (Springer-Verlag, Berlin Heidelberg, 1980).
4) Chen, S. and R. Saigal, "A Primal Algorithm for Solving a Capacitated Network Flow Problem with Additional Linear Constraints", Networks, **7**, 59-79 (1977).
5) Garfinkel, R. S. and G. L. Nemhauser, *Integer Programming*, (John Wiley, New York, 1972).
6) Hatch, R. S., "Bench Marks Comparing Transportation Codes Based on Primal Simplex and Primal-Dual Algorithms" Operatons Research, **23**, 1167-1172 (1975).
7) Held, M., P. Wolfe and H. Crowder, "Validation of Subgradient Optmization", Mathematical Programming, **6**, 62-88 (1974).
8) Kuhn, H. W., "The Hungarian Method for the Assignment Problem", Navel Research Logistics Quarterly, **2**, 83-97 (1955).
9) Rockafelar, R. T., *Convex Analysis*, (Princeton University Press, Princeton, N. J., 1970).
10) Shapiro, J. F., "A Survey of Lagrangean Techniques for Discrete Optimization", *Annals of Discrete Mathematics 5*, 113-138 (1979).
11) Van Slyke, R. M. and R. Wets, "L-Shaped Linear Programs with Applcations to Optimal Control and Stochastic Programming", SIAM Journal on Applied Mathematics, **17**, 638-663 (1969).
12) Wets, R. "Programming under Uncertainty : The Complete Problem", Z. Wahrscheinlichkeitstheorie verw. Geb. **4**, 316-339 (1969).
13) Wollmer, R. D., "Two Stage Linear Programming under Uncertainty with 0-1 Integer First Stage Variables", Mathematical Programming, **19**, 279-288 (1980).
14) Yudin, D, B. and E. V. Tsoy "Integer-Valued Stochastic Programming", Engineering Cybernetics, **12**, 1-8 (1973).
15) Ziemba, W. T., "Stochastic Programs with Simple Recourse", *Mathematical Programming in Theory and Practice*, 213-273, (North-Holland, 1974).