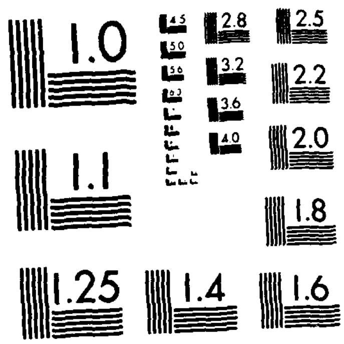
AD-A185 497 THE STOCHASTIC DYNAMIC TRAFFIC ASSIGNMENT PROBLEM(U) 1/1
MICHIGAN UNIV ANN ARBOR DEPT OF INDUSTRIAL AND
OPERATIONS ENGINEERING J R BIRGE ET AL. 1987 TR-87-16
UNCLASSIFIED N00014-86-K-0628 F/G 12/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

DTIC FILE COPY

AD-A185 497

The Stochastic Dynamic Traffic Assignment Probler

John R. Birge
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan, USA 48109

James K. Ho
Management Science Program
College of Business Administration
The University of Tennessee
Knoxville, Tennessee, USA 37916

Technical Report 87-16

Department of Industrial and Operations Engineering

The University of Michigan
College of Engineering
Ann Arbor, Michigan 48109-2117

12

The Stochastic Dynamic Traffic Assignment Problem

John R. Birge
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan, USA 48109

James K. Ho
Management Science Program
College of Business Administration
The University of Tennessee
Knoxville, Tennessee, USA 37916

Technical Report 87-16

The Stochastic Dynamic Traffic Assignment Problem

John R. Birge*

Department of Industrial and Operations Engineering

The University of Michigan

Ann Arbor, Michigan, USA 48109

James K. Ho**

Management Science Program

College of Business Administration

The University of Tennessee

Knoxville, TN, USA 37916

Abstract : This paper presents a method for solving a stochastic version of the dynamic traffic assignment problem. It shows that a globally optimal solution may be obtained by a sequence of linear optimizations. A decomposition algorithm for this procedure is presented that efficiently solves large-scale problems. Solution examples with up to sixty–six thousand variables are described.

Keywords: stochastic programming, traffic assignment, multi-stage problems.

* - This author's work was supported in part by National Science Foundation Grant No. ECS–8304065, Office of Naval Research Grant N00014-86-K-0628, and by the National Research Council under a Research Associateship at the Naval Postgraduate School, Monterey, California.

** - This author's work was supported in part by Office of Naval Research Grant N00014-87-K-0163.

1. Introduction

The problem of determining flows in a network to optimize an objective or satisfy equilibrium conditions has been studied in a variety of frameworks. The static problem has been extensively studied (see, for example, Dafermos and Sparrow [3] and Potts and Oliver [7]), but the dynamic problem has received less attention. Merchant and Nemhauser [5] gave a mathematical programming formulation of this dynamic traffic assignment problem. They showed that the resulting nonlinear, nonconvex program has a piecewise linear version such that a feasible, globally optimal solution exists among the set of linear program optima.

Ho [4] showed that a global optimum could be obtained by solving a sequence of linear programs. The number of such programs is at most $N + 1$ where N is the number of periods.

The previous studies have all assumed that the exogenous flows into nodes are known for all periods. This assumption is not generally true. Instead, the flows are unknown and represent random rates of arrivals into the system. The *stochastic* dynamic traffic assignment problem is then to determine flows along the arcs to minimize a convex function of traffic congestion. Flow decisions in one period depend on the realization of inputs from previous periods. The decisions seek to minimize expected costs over future periods.

In this paper, we formulate a piecewise linear version of the stochastic dynamic traffic assignment problem. This representation is analogous to those given by Ho and Merchant and Nemhauser. In this multistage model, uncertainties are resolved period by period as time progresses. We show that a successive linearization procedure, similar to Ho's, yields a globally optimal solution. We also present an algorithm based on the multistage decomposition method of Birge [1] to implement this procedure. Experimental results for this algorithm on problems with up to sixty-six thousand variables are given.

2. Multistage Stochastic Program Formulation

The multistage stochastic program assumes that decisions taken in period i depend on outcomes and decision made in periods $1, \ldots, i-1$. Period i decisions cannot, however, depend explicitly on future outcomes (i.e., they are ***nonanticipative***). In every period i, a finite number L_i of ***scenarios*** represent different sets of inputs. Each scenario l_i has an associated ***probability*** p_{l_i}. For every scenario l_i in period $i (i > 0)$, there corresponds a scenario $a(l_i)$ in period $i-1$ which is the ***immediate ancestor*** scenario of l_i. Each l_i also has a set $\mathcal{A}(l_i)$ of ***ancestor*** scenarios (one in each period $i' < i$) that is defined recursively as the set of currently defined ancestor scenarios and their immediate ancestors. There also exists a set of ***immediate descendant*** scenarios $\mathcal{D}_{i+1}(l_i) \subset \{1, \ldots, L_{i+1}\}$, in period $i+1$ for every i. The descendant scenarios partition $\{1, \ldots, L_{i+1}\}$ so that $\bigcup_{l_i \in \{1, \ldots, L_i\}} \mathcal{D}_{i+1}(l_i) = \{1, \ldots, L_{i+1}\}$ and $\mathcal{D}_{i+1}(i) \cap \mathcal{D}_{i+1}(j) = \emptyset$ if $i \neq j$. We let $\mathcal{D}(l_i)$ be the set of ***descendant*** scenarios consisting of immediate descendant scenarios of i and all future (recursively defined) immediate descendants up to the planning horizon.

The flow decisions are x_{ijl} for arc j in period i and scenario l. Other notation is consistent with Ho with decision variables λ_{ijl}^k representing weights on grid points c_j^k such that

$$x_{ijl} = \sum_{k=1}^{K(j)} c_j^k \lambda_{ijl}^k. \tag{1}$$

For period $i = 0$, we define $L_0 = 1$ and let x_{0j} and λ_{0j}^k be interchangeable with x_{0j1} and λ_{0j1}^k.

The notation includes:

$G = (\mathcal{N}, \mathcal{E})$ a directed graph;

$\mathcal{N}$ = set of nodes of G;

$\mathcal{E}$ = set of arcs (directed edges) of G;

N = planning horizon;

i = index of time period;

j = index of arc in $\mathcal{E}$; $j = 1, ..., a$;

q = index of node of $\mathcal{N}$; $q = 1, ..., n$;

n = index of destination node;

$A(q) = \{j \in \mathcal{E} \mid \text{arc } j \text{ leaves node } q\}$;

$B(q) = \{j \in \mathcal{E} \mid \text{arc } j \text{ enters node } q\}$;

$F_{il}(q) =$ external input at node q under scenario l;

$$h_{ij}(x_{ijl}) = \text{cost of } x_{ijl} = \sum_{k=1}^{K(j)} h_{ij}^k \lambda_{ijl}^k;$$

$d_{ijl} =$ amount of traffic admitted to arc j in period i under scenario l;

$$g_j(x_{ijl}) = \text{amount of traffic to exit from arc } j \text{ in period } i \text{ under scenario } l$$
$$= \sum_{k=1}^{K(j)} g_j^k \lambda_{ijl}^k.$$

We make the following assumptions as in Ho[4].

(A1) The arcs are not explicitly capacitated.

(A2) Saturation occurs for large enough x, i.e., $g_j^{K(j)} = 0$.

(A3) The nonnegative slope of the piecewise linear approximation to g_j is strictly decreasing.

(A4) The cost functions are structured so that

(a) $0 \leq h_{ij}^k \leq h_{ij}^{k+1}$ for all i, j and k;

(b) If j_2 is closer to the sink than j_1 along any simple path, then $h_{ij_2}^{K(j_2)} \leq h_{ij_1}^1$ for all i.

We add an assumption from Merchant and Nemhauser [5]:

(A5) $0 \leq g_j(x) \leq x$.

The stochastic dynamic traffic assignment problem (SDTAP) is then formulated as:

$$\min \sum_{j=1}^{a} \sum_{k=1}^{K(j)} h_{0j}^k \lambda_{0j}^k + \sum_{i=1}^{N} \sum_{l=1}^{L_i} p_l \{\sum_{j=1}^{K(j)} h_{ij}^k \lambda_{ijl}^k\} = G_{N+1}(\lambda) \qquad SDTAP.0$$

subject to

$$\sum_{k=1}^{K(j)} c_j^k \lambda_{1jl}^k = \sum_{k=1}^{K(j)} (c_j^k - g_j^k) \lambda_{0j}^k + d_{0j}, j = 1, ..., a; l = 1, ..., L_i; \qquad SDTAP.1$$

$$\sum_{k=1}^{K(j)} c_j^k \lambda_{i+1,jl'}^k = \sum_{k=1}^{K(j)} (c_j^k - g_j^k) \lambda_{ijl}^k + d_{ijl}, i = 1, \ldots, N-1;$$

$$j = 1, \ldots, a; l' \in D_{i+1}(l); l = 1, \ldots, L_i; \qquad SDTAP.2$$

$$\sum_{j \in A(q)} d_{0j} = F_0(q) + \sum_{j \in B(q)} \sum_{k=1}^{K(j)} g_j^k \lambda_{0j}^k, \forall q \neq n; \qquad SDTAP.3$$

$$\sum_{j \in A(q)} d_{ijl} = F_{il}(q) + \sum_{j \in B(q)} \sum_{k=1}^{K(j)} g_j^k \lambda_{ijl}^k, l = 1, \ldots, L_i;$$

$$i = 1, \ldots, N; \forall q \neq n; \qquad SDTAP.4$$

$$\sum_{k=1}^{K(j)} c_j^k \lambda_{0j}^k = R_j, j = 1, ..., a; \qquad SDTAP.5$$

$$\left.\begin{cases} d_{0j} \geq 0, d_{ijl} \geq 0, & i = 1, \ldots, N, j = 1, \ldots, a, l = 1, \ldots, L_i; \\ \lambda_{0j}^k \geq 0, & j = 1, \ldots, a, k = 1, \ldots, K(j); \\ \lambda_{ijl}^k \geq 0, & i = 1, \ldots, N, j = 1, \ldots, a, \\ & l = 1, \ldots, L_i, k = 1, \ldots, K(j); \end{cases}\right\} \qquad SDTAP.6$$

$$\sum_{k=1}^{K(j)-1} \lambda_{0j}^k = 1, \sum_{k=1}^{K(j)-1} \lambda_{ijl} = 1; i = 1, \ldots, N, j = 1, \ldots, a; l = 1, \ldots, L_i; \qquad SDTAP.7$$

$$\left.\begin{cases} \lambda_{0j}^k > 0 & \text{for at most two consecutive } k, \\ \lambda_{ijl}^k > 0 & \text{for at most two consecutive } k. \end{cases}\right\} \qquad \textbf{OSP}$$

SDTAP is a multistage stochastic program with recourse (see Wets [8]). Decisions λ_{ijl}^k and d_{ijl} are made knowing $F_{i'l}$ for $i' \leq i$ with certainty but without knowing the realizations of future inflows $F_{i'l}$ for $i' > i$. The future inflows represent the discrete realizations of the random external input.

The *ordered solution property* constraint (OSP) makes SDTAP a nonconvex program. Additional optimization must, therefore, be performed to solve it. In Ho [4], it is shown that an optimal solution can be found for the deterministic problem by successively optimizing at most $N+1$ objective functions for the linear programming part of SDTAP. We show that a similar successive optimization approach yields a globally optimal soultion for the stochastic problem.

The difference between the procedure followed below and Ho's procedure is our order of optimization and choice of objectives. These objectives are used on individual scenarios to maintain the form of the stochastic program. We first state a result from Lemmae A,B,C of Ho and Lemma 1 of Merchant and Nemhauser.

Lemma 1. Let $y = \{\lambda_{ijl}^k, d_{ijl}\}$ be a feasible solution of SDTAP.1–7 that violates OSP for $i = r$, $j = s$, and $l = t$. There exists a feasible solution, $\overline{y} = \{\overline{\lambda}_{ijl}^k, \overline{d}_{ijl}\}$, to SDTAP.1–7 that differs from y only for $j = s$ when $i = r$ and $l = t$, and for arcs j on paths from s to n for $i > r$ and $l \in \mathcal{D}_i(t)$. The solution for λ in (1), $x_{rst} = \overline{x}_{rst}$, the solution given by $\overline{\lambda}$. The solution $\overline{y}$ satisfies OSP for $i = r$ and $j = s$ and:

a.) For all $q \in \mathcal{N}$, $i = 0, \ldots, N$, $l = 1, \ldots, L_i$, the total flow reaching q on or before i is at least as great for $\overline{y}$ as for y.

b.) For any scenario $l \in L_N$, the *scenario l objective value*,

$$G_{N+1}^l(\lambda) = \sum_{i=0}^{N} \sum_{l' \in \mathcal{A}(l)} \sum_{j=1}^{a} \sum_{k=1}^{K(j)} h_{ij}^k \lambda_{ijl'}^k \geq G_{N+1}^l(\overline{\lambda}).$$

c.) The solution $\{\overline{\lambda}_{ijl}^k, \overline{d}_{ijl}\}$ is the same for any value of $\{\lambda_{i'jl}^k, d_{i'jl}\}$, $i' > i$.

Proof: Applying Lemma 1 of Merchant and Nemhauser on each scenario $l = 1, \ldots, L_N$ and its ancestor scenarios, $\mathcal{A}(l)$, directly yields the results above except for the nonanticipativity requirement in SDTAP. We, therefore, need to obtain the same $\overline{\lambda}^k_{ijl'}$ and $\overline{d}_{ijl'}$ for all $l \in \mathcal{D}(l')$. By the construction in Merchant and Nemhauser, (c) above holds so that nonanticipativity is maintained.■

The following corollary is the basis for the successive optimization procedure.

Corollary 1. Consider a feasible solution $\hat{y} = \{\hat{\lambda}^k_{ijl}, \hat{d}_{ijl}\}$ of SDTAP.1–7. Suppose $y^* = \{\lambda^{k*}_{ijl}, d^*_{ijl}\}$ is an optimal solution of

$$\max \quad G_{i'l'}(\lambda) = \sum_{j=1}^{a} \sum_{k=1}^{K(j)} g^k_j \lambda^k_{i'jl'} \qquad (SUB - i'l')$$

subject to (SDTAP.1 - SDTAP.7)

$$(\lambda^k_{ijl}, d_{ijl}) = (\hat{\lambda}^k_{ijl}, \hat{d}_{ijl}); i < i', \forall j, l;$$

$$i \geq i', l \notin \mathcal{D}(l'), \forall j;$$

$$G^l_{N+1}(\lambda) \leq G^l_{N+1}(\hat{\lambda}), l = 1, \ldots, L_N.$$

Then, y^* has no OSP violations in period i' and scenario l'.

Proof: Suppose an OSP violation occurs in y^* in period i', scenario l', arc j'. By Lemma 1, there exists a solution $\overline{y} = \{\overline{\lambda}, \overline{d}\}$ that satisfies OSP on arc j' in scenario l' at period i' and that satisfies the constraints in (SUB–$i'l'$). By strict concavity in Assumption A3,

$$\sum_{k=1}^{K(j')} g^k_{j'} \overline{\lambda}^k_{i'j'l'} > \sum_{k=1}^{K(j')} g^k_{j'} \lambda^{k*}_{i'j'l'}. \qquad (2)$$

Hence, $G_{i'l'}(\overline{\lambda}) > G_{i'l'}(\lambda^*)$, a contradictions of λ^*'s optimality.■

This corollary can be applied repeatedly from period 1 to N for each scenario to obtain a full OSP solution. The repeated direct solution of the full multistage problem using standard linear programming procedures may, however, be too computationally burdensome to implement. The structure of SDTAP can, however, be exploited. In [1], a multistage nested Benders' decomposition method (NDSP) is described that significantly reduced solution times from simplex method solutions on a set of practical test problems. NDSP applies directly to SDTAP.1–7. Obtaining an OSP solution using objectives $G_{i'l'}$ requires, however, the addition of the intertemporal constraints, $G^l_{N+1}(\lambda) \leq G^l_{N+1}(\hat{\lambda})$. These constraints are not directly compatible in the decompostion procedure. The development below shows how to incorporate these constraints into a procedure to obtain OSP.

To simplify notation in the following development, let $y_{il} = \{\lambda^k_{ijl}, d_{ijl}\}$, let SDTAP.1 be

$$G_{N+1}(y) = \sum_{i=0}^{N}\sum_{l=1}^{L_i} p_{il} h_i y_{il}, \tag{3}$$

let the constraints, SDTAP.1,3, and 5, be

$$A_0 y_0 = b_0, \tag{4}$$

let the constraints, SDTAP.2 and 4, be

$$B_{i-1} y_{i-1,l'} + A_i y_{i,l} = \xi_{il}, \forall l \in \mathcal{D}(l'), i = 1,\ldots,N; l' = 1,\ldots,L_{i-1}; \tag{5}$$

and note that below we use j and k as general indices that are not restricted to denoting arcs and linear segments.

The basic (i,l)–subproblem solved in NDSP, given some solution, $\hat{y}_{i-1,a(l)}$, is then

$$
\begin{array}{rllr}
Q_{il}(\hat{y}_{i-1,a(l)}) = \min\ h_i y_{il} & +\theta_{il} & & (NDS-1) \\
\text{subject to } \ A_i y_{il} & & = \xi_{il} - B_{i-1}\hat{y}_{i-1,a(l)}; & (NDS-2) \\
D^r_{il} y_{il} & & \geq d^r_{il}, r = 1,\ldots,R_{il}; & (NDS-3) \\
E^s_{il} y_{il} & +\theta_{il} & \geq e^s_{il}, s = 1,\ldots,S_{il}; & (NDS-4) \\
y_{il} & & \geq 0. & (NDS-5)
\end{array}
$$

Constraints (NDS–3) are ***feasibility cuts*** to maintain a feasible solution in descendant scenarios l. The (NDS–4) constraints are ***optimality cuts*** on θ_{il}. They represent an outer linearization of G_{N+1} restricted to descendants of l. NDSP repeatedly solves subproblems NDS to obtain $(\hat{y}_{il}, \hat{\theta}_{il})$ until the optimality condition,

$$p_{il}\hat{\theta}_{il} \geq \sum_{l' \in \mathcal{D}_{i+1}(l)} p_{i+1,l'} Q_{i+1,l'}(\hat{y}_{il}), \tag{6}$$

is achieved for all i, l. (Note that $p_{il} = \sum_{l' \in \mathcal{D}_{i+1}(l)} p_{i+1,l'}$.)

For the algorithm below, we also define

$$p_{il} G^{il}_{N+1}(\hat{y}) = \sum_{i'=i}^{N} \sum_{l' \in \mathcal{D}(l)} p_{i'l'} h_{i'} \hat{y}_{i'l'}, \tag{7}$$

as the contribution to the objective of l and its descendant scenarios. The following algorithm obtains OSP in all i and l.

Nested Decomposition of Stochastic Traffic Assignment Problem (NDSTAP)

0. Intitialization. Use NDSP to obtain $\hat{y}$, a solution to SDTAP.1–7. Let $i = -1$, $l = 0$. Let $L_0 = 1$.

1. Check for OSP. Let $l = l + 1$. If $i > N$, stop; $\hat{y}$ is a full OSP solution.

If $l > L_i$, let $i = i + 1$, $l = 0$, return to 1.

Else, check OSP in $\hat{y}_{il}$. If OSP is satisfied, return to 1.

Else, go to 2.

2. Setup for OSP at (i, l). Let $\hat{y}_{i'l'}$ be as in the NDSP solution for all $i' = i + 1, \ldots, N$, and $l' \in \mathcal{D}(l)$. For each such subproblem (i', l'), let the NDS–3 and NDS–4 constraints be those defined at the end of Step 0. Let $K_{i'l'} = 0$ for all $i' = i + 1, \ldots, N; l' \in \mathcal{D}(l)$. Go to 3.

3. Maximize flow at i, l. Solve the subproblem:

$$\max \quad G_{il}(y_{il}) \tag{8.0}$$

$$\begin{aligned}
\text{subject to } A_i y_{il} & & &= \xi_{il} - B_{i-1}\hat{y}_{i-1,a(l)}; && (8.1)\\
h_i y_{il} & & -z_{il} &= -\textstyle\sum_{j=0}^{i-1}\sum_{l'\in A(l)} h_j \hat{y}_{jl'}; && (8.2)\\
h_i y_{il} & & +\theta_{il} &\le G_{N+1}^{il}(\hat{y}); && (8.3)\\
D_{il}^r y_{il} & & &\ge d_{il}^r, r = 1,\ldots,R_{il}; && (8.4)\\
E_{il}^s y_{il} & & +\theta_{il} &\ge e_{il}^s, s = 1,\ldots,S_{il}; && (8.5)\\
\bar{D}_{il}^k y_{il} & & +\bar{d}_{il}^{k1} z_{il} &\ge \bar{d}_{il}^{k2}, k = 1,\ldots,K_{il}; && (8.6)\\
y_{il} & & &\ge 0; && (8.7)
\end{aligned}$$

to obtain $(\hat{y}_{il}, \hat{z}_{il})$. Let $i' = i + 1$. Go to 4.

4. Check feasibility at i'. If $i' > N$, go to 1. Else, for all $l' \in \mathcal{D}(l)$ at i', solve

$$\min h_i y_{i'l'} \quad +\theta_{i'l'} \tag{9.0}$$

$$\begin{aligned}
\text{subject to } A_{i'} y_{i'l'} & & &= \xi_{i'l'} - B_{i'-1}\hat{y}_{i'-1,a(l')}; && (9.1)\\
h_{i'} y_{i'l'} & & -z_{i'l'} &= -\hat{z}_{i'-1,a(l')}; && (9.2)\\
D_{i'l'}^r y_{i'l'} & & &\ge d_{i'l'}^r, r = 1,\ldots,R_{i'l'}; && (9.3)\\
E_{i'l'}^s y_{i'l'} & & +\theta_{i'l'} &\ge e_{i'l'}^s, s = 1,\ldots,S_{i'l'}; && (9.4)\\
\bar{D}_{i'l'}^k y_{i'l'} & & +\bar{d}_{i'l'}^{k1} z_{i'l'} &\ge \bar{d}_{i'l'}^{k2}, k = 1,\ldots,K_{i'l'}, i' < N; && (9.5a)\\
& & -z_{i'l'} &\ge -G_{N+1}^{l'}(\hat{y}), i' = N; && (9.5b)\\
y_{i'l'} & & &\ge 0, && (9.6)
\end{aligned}$$

to obtain a new $\hat{y}_{i'l'}$.

If all subproblems (9) at i are feasible, let $i' = i' + 1$, return to 4.

Else, for (9) infeasible at (i', l'), add a feasibility cut, (9.5a), to $(i' - 1, a(l'))$, where

$$\bar{D}_{i'-1,a(l')}^k = \pi^1 B_{i'-1}; \tag{10.1}$$

$$\bar{d}_{i'-1,a(l')} = \pi^2; \tag{10.2}$$

$$\bar{d}^{k2}_{i'-1,a(l')} = \pi^1 \xi_{i'l'} + \pi^3 d_{i'l'} + \pi^4 e_{i'l'} + \pi^5 w; \tag{10.3}$$

$w = \bar{d}_{i'l'}$ if $i' < N$; $w = -G^{l'}_{N+1}(\hat{y})$ if $i' = N$, $k = K_{i'-1,a(l')} + 1$, and $\pi^j, j = 1, \ldots, 5$, are the multipliers corresponding to constraints (9.j), $j = 1, \ldots, 5$, respectively, that obtained the infeasibility condition. Update $K_{i'-1,a(l')}$ for each infeasible (i', l') subproblem (9). Let $i' = i' - 1$. If $i' = i$, go to 3. Else, return to 4.

The NDSTAP steps 1 to 4 differ from NDSP iterations in the use of the linking constraints (8.2) and (9.2). NDSTAP also does not add any optimality cuts (8.5) and (9.4) because (9.5b) guarantees optimality.

Other strategies can also be used in proceeding between periods. The above strategy solves all problems at one period proceeding to the next period. An alternative is to solve all descendant subproblems for the current scenario at i' before proceeding to the next i' scenario. The above implementation required few iterations, however, and was used for the computational tests. Convergence of the NDSTAP iterations to an OSP solution is given in the following theorem.

Theorem 1. NDSTAP terminates in a finite number of iterations of Steps 1 to 4 with a solution $\{\hat{y}_{il}\}$ that satisfies OSP for all i and l.

Proof: OSP is obtained for $i = 0$ if Steps 2 to 4 terminate for $i = 0$, according to Corollary 1. Given OSP at $i = 0$, Corollary 1 and induction establish OSP for all i and l if Steps 2 to 4 terminate finitely for each non–OSP solution found.

To show that Steps 2 to 4 terminate finitely, note that, because the constraints (8.3–5) and (9.3–4) are outer linearizations, any feasible solution $\{y_{il}\}$ such that $G_{N+1}(y) \leq$

$G_{N+1}(\hat{y})$ must satisfy (8.3–5) and (9.3–4) for all i and l. These constraints, therefore, do not exclude any solutions y that satisfy

$$G^l_{N+1}(y) \leq G^l_{N+1}(\hat{y}) \text{ for all } l = 1, \ldots, L_N. \tag{11}$$

Since (8.2) and (9.2) are always satisfiable, we must establish that

1. there exists y_{il} and $\{y_{i'l'} \mid i' > i, l' \in \mathcal{D}(l)\}$ that are feasible for (8.1–5) and (9.1–4,5b) for all i', l';
2. Constraints (8.6) and (9.5a) do not exclude any feasible y that satisfies (11);
3. Only a finite number of constraints (8.6) and (9.7) are generated.

The first condition is guaranteed by Corollary 1 as stated above. Condition 2 is true by induction. Assume that $v = \hat{y}_{N-1,a(l')}$, $u = \hat{z}_{N-1,a(l')}$ is input to subproblem (9) for (N, l'). Then, there exist multipliers $\pi^1, \ldots, \pi^5$ ($\pi^3 = \pi^4 = 0$ since (9.3) and (9.4) are vacuous for $i = N$), such that

$$\begin{aligned} \pi^1 A_N + \pi^2 h_N \quad &\leq 0, \\ -\pi^2 \quad -\pi^5 &= 0, \\ \pi^5 \quad &\geq 0, \end{aligned} \tag{12}$$

and

$$\pi^1 \xi_{Nl'} - \pi^1 B_{N-1} v - \pi^2 u - \pi^5 G^{l'}_{N+1}(\hat{y}) > 0. \tag{13}$$

In order for v and u to be feasible, therefore, we require

$$\pi^1 B_{N-1} v + \pi^2 u \geq \pi^1 \xi_{Nl'} - \pi^5 G^{l'}_{N+1}(\hat{y}). \tag{14}$$

Constraint 14 is the feasibility cut (9.5a) as defined in (10). Hence, no constraints (9.5a) for $i' = N - 1$ exclude any feasible y. By induction, the same argument applies for all constraints (8.6) and (9.5a), establishing Condition 2.

Condition 3 is true by induction also. For $i = N$, only extreme solutions of (12) need be considered. Therefore, only a finite number of cuts (14) are generated. Hence, only a finite number of constraints (9.5) are added to subproblems for $i' = N - 1$. By induction, this is true for all i'.■

3. Computational Results

The algorithm NDSTAP was coded in FORTRAN as an extension of the NDSP code ([1]). The code applies to problems with up to three periods ($N = 2$) with multiple descendants in each period. Scenarios with single immediate descendants can be grouped together, however, to solve problems with $N > 2$ and $L_2 = \cdots = L_N$. This situation corresponds, in practice, to the ability to predict future inflows with certainty given observation of the first three periods of inflows.

The NDSTAP code takes advantage of single–descendant scenarios in the last period by combining periods 2 to N into a single subproblem:

$$\begin{aligned}
&\min && G_{N-k}(y_{2l}, \ldots, y_{Nl}) && && (15.0)\\
&\text{subject to} && A_2 y_{2l} && = \xi_{2l} - B_1 \hat{y}_{1,a(l)}; && (15.1)\\
& && B_{jl} y_{jl} \quad + A_{j+1,l} y_{j+1,l} && = \xi_{jl}, j = 2, \ldots, N-1; && (15.2)\\
& && \textstyle\sum_{j=2}^{N} h_{jl} y_{jl} && \leq G^l_{N+1}(\hat{y}) - h_1 \hat{y}_{1,a(l)} - h_0 \hat{y}_{0,a(a(l))}; && (15.3)\\
& && G_{N-s}(y_{2l}, \ldots, y_{Nl}) && \geq \bar{G}_{N-s}, s = 0, \ldots, k-1 (\text{ if } k > 0); && (15.4)\\
& && y_{jl} && \geq 0, j = 2, \ldots, N; && (15.5)
\end{aligned}$$

where $G_{N-s}(y_{2l}, \ldots, y_{Nl}) = \sum_{j=2}^{N-s} g_j y_{jl}$ is the total flow from each arc in periods 2 to $N-s$, and $\bar{G}_{N-s}$ is the value of G_{N-s} obtained for subproblem (15) when solved with objective G_{N-s}. Subproblem (15) is solved instead of (8) for increasing values of k (starting at $k = 0$) until an OSP solution is obtained for $(y_{2l}, \ldots, y_{Nl})$. This is the successive optimization procedure in Ho, which again achieves OSP finitely. Its advantage over solving (8) and (9) repeatedly is that a single solution of (15) may obtain OSP for all $y_{il}, i = 2, \ldots, N$, whereas several subproblems (8) may be required. The use of a similar objective with several descendant scenarios (i.e., when $L_i < L_{i+1}$), however, still requires several optimizations and is not advantageous over (8) and (9). The strategy in NDSTAP requires only one additional optimization to obtain OSP in period 0, which may be the only decision actually implemented before another problem is solved.

NDSTAP takes further advantage of the structure of (15) by checking for an *OSP basis*, i.e., a basis of (15) such that λ_{ijl}^{k} is *basic* for at most two consecutive k. When an OSP basis is found, we need only check its feasibility in other subproblems (15) for different l to find an OSP solution for these scenarios. In practice, this quickly produces OSP solutions for all scenarios.

The test problems were generated with the same structure as Ho's test problems in [4], using Ho's TAPGEN generator to create a deterministic problem and the STGEN generator used in [1] to create the stochastic problem. All examples use the network in [4] with seven nodes and twelve arcs. They also have five periods ($N = 4$).

The test problems include varying numbers of scenarios and different period 0 input flows. The scenario numbers ($L_2 = L_3 = L_4$) range from six to eighty–one. The period 0 inputs are characterized as "high," "medium" or "low." Each test problem has three period–one scenarios ($L_1 = 3$). The input flows are summarized in Table 1. The period-two scenarios are constructed from combinations of the period–two values in Table 1 for nodes four, five and six.

Table 1. Input Flows

NODE	*PERIOD*					
	0			1		
	Input Level			Scenario		
	High	Medium	Low	1	2	3
1	10	30	50	10	15	30
2	10	30	50	20	30	30
3	10	30	50	5	10	30
4	10	30	50	0	30	30
5	10	30	50	0	30	30
6	10	30	50	0	30	30
NODE	*PERIOD*					
	2		3		4	
1	30		20		5	
2	70		40		30	
3	20		10		5	
4	(10,20,30)		30		40	
5	(15,25,35)		60		50	
6	(0,15,30)		10		20	

The size characteristics of the linear programs (SDTAP.1–7) for each test problem are given in Table 2. That table gives the number of period–two scenarios, the number of constraints and the number of variables for each SDTAP solved.

Table 2. Test Problem Characteristics

Problem Numbers	Scenarios	Constraints	Variables
1–3	6	720	5,952
4–6	12	1,296	10,800
7–9	24	2,448	20,496
10–12	81	7,920	66,552

The test problems were solved on the IBM 3033/4381 Computer Network at the Naval

Postgraduate School, Monterey, California, using the VS FORTRAN compiler under the VM/CMS operating system. The results from solving each problem are given in Table 3. The number of NDSP "major iterations" refers to the number of subproblems, (NDS), solved (excluding multiple solutions of last period subproblems with the same ancestor input). NDSTAP "major iterations" refer to the number of subproblems (8), (9) and (15) solved. "Simplex iterations" refer to the number of simplex iterations performed on subproblems NDS, (8), (9) and (15). The CPU second times (CPUs) do not include initial input and final output times but do include time for iteration logging.

The number of OSP violations found for each test problem in Step 1 of NDSTAP are given in Table 4 by period (with periods 2 to 4 grouped together). Whenever an OSP feasible basis in (15) was found for some scenario l, the corresponding OSP solution was used for that scenario so that no OSP violations were recorded for l.

Table 3. Test Problem Iterations and Times

Problem		*NDSP*		
	Period 0 Input	Major Iterations	Simplex Iterations	CPUs
1	High	46	326	0.104
2	Medium	62	346	0.124
3	Low	42	353	0.172
4	High	46	402	0.192
5	Medium	62	449	0.316
6	Low	40	364	0.253
7	High	46	468	0.443
8	Medium	62	535	0.597
9	Low	40	480	0.587
10	High	52	656	1.640
11	Medium	68	1,024	3.636
12	Low	50	729	2.160
Problem		*NDSTAP*		
	Period 0 Input	Major Iterations	Simplex Iterations	CPUs
1	High	7	177	0.066
2	Medium	7	151	0.059
3	Low	7	171	0.075
4	High	7	228	0.103
5	Medium	8	190	0.128
6	Low	6	231	0.162
7	High	7	229	0.140
8	Medium	7	247	0.188
9	Low	7	297	0.290
10	High	7	245	0.204
11	Medium	7	339	0.944
12	Low	6	557	1.234

Table 4. OSP Violations

Problem	*Period*			*Total*
	0	1	2,3,4	
1	2	15	19	36
2	1	12	16	29
3	0	6	20	26
4	1	13	22	36
5	1	15	18	34
6	0	7	18	25
7	1	14	20	35
8	1	15	23	39
9	0	7	31	38
10	1	12	19	32
11	1	16	25	42
12	0	4	34	38

Note that higher input flows generally produce more OSP violations (Table 4). Ho also observed this in deterministic problems. Observe in Table 3, however, that higher input flow problems are generally easier to solve. This may be due to a reduced number of feasible bases for higher input flows leading to fewer iterations.

The results in Tables 3 and 4 indicate that NDSTAP can efficiently produce OSP solutions for large–scale linear programs SDTAP. The test problems were not solved by a direct simplex method implementation because time and memory requirements exceeded the VM/CMS limitations.* The true advantage of NDSTAP is, indeed, for problems that

* To give some comparison, however, a smaller SDTAP problem (without OSP) with the same network, 511 constraints, and 1318 variables was solved in [1]. NDSP solved this problem in 0.76 CPU seconds on an Amdahl V8 with 35 major iterations and 92 simplex iterations. MINOS (Murtagh and Saunders [6]) solved the same problem on the same machine in 10.37 CPU seconds with 337 simplex iterations.

cannot be solved by a direct simplex method implementation. We demonstrate here that OSP solutions can efficiently be obtained for examples of these difficult problems.

4. Conclusion

This paper presented a formulation of a stochastic dynamic traffic assignment problem with uncertain input flows and dynamic decisions given knowledge of previous inputs. A successive optimization procedure to obtain a globally optimal solution was then developed and shown to converge. The efficiency of this nested decomposition procedure, NDSTAP, was demonstrated on a set of test problems with up to almost eight thousand linear programming constraints and over sixty–six thousand variables. An NDSTAP implementation solved all problems in under five CPU seconds on an IBM 3033/4381. These results demonstrate the practicality of decompostion approaches for stochastic traffic assignment problems and other general multistage stochastic linear programs.

REFERENCES

[1] J.R. Birge, "Decomposition and partitioning methods for multi–stage stochastic linear programs", **Operations Research 33** (1985), 989–1007.

[2] J.R. Birge, "An L-shaped method computer code for multi-stage stochastic linear programs", in *Numerical Methods in Stochastic Programming*, Y. Ermoliev and R. Wets, eds., (Springer–Verlag, Berlin, 1987).

[3] S.C. Dafermos and F.T. Sparrow, "The traffic assignment problem for a general network", **Journal of Research of the National Bureau of Standards 73B** (1969) 91–118.

[4] J.K. Ho, "A successive linear optimization approach to the dynamic traffic assignment

problem", **Transportation Science 14**(1980) 295–305.

[5] D.K. Merchant and G.L. Nemhauser, "A model and an algorithm for the dynamic traffic assignment problem", **Transportation Science 12**(1978) 183–199.

[6] B.A. Murtagh and M.A. Saunders, "Large-scale linearly constrained optimization", **Mathematical Programming 14** (1978) 41–72.

[7] R.B. Potts and R.M. Oliver, *Flows in transportation networks* (Academic Press, New York, 1972).

[8] R. J-B. Wets, "Stochastic programming: solution techniques and approximation schemes", in: A. Bachem, M. Groetschel, and B. Korte, eds., *Mathematical programming: state-of-the-art 1982* (Springer-Verlag, Berlin, 1983) pp. 506-603.

END
DATE
FILMED
DEC.
1987