1989

The equivalence of transfer and generalized benders decomposition methods for traffic assignment

Barton, Russell R.; Hearn, Donald W.; Lawphongpanich, Siriphong

Transpt. Res.-B, Vol. 23B, No. 1, pp. 61-73, 1989
https://hdl.handle.net/10945/43132

Transpn. Res.-B. Vol. 23B. No. 1, pp. 61–73, 1989
Printed in Great Britain.

0191-2615/89 $3.00 + .00

THE EQUIVALENCE OF TRANSFER AND GENERALIZED BENDERS DECOMPOSITION METHODS FOR TRAFFIC ASSIGNMENT†

RUSSELL R. BARTON
School of Operations Research and Industrial Engineering, Upson Hall, Cornell University, Ithaca, NY 14853, U.S.A.

DONALD W. HEARN
Department of Industrial and Systems Engineering, 303 Weil Hall, University of Florida, Gainesville, FL 32611, U.S.A.

and

SIRIPHONG LAWPHONGPANICH
Department of Operations Research, U.S. Naval Postgraduate School, Monterey, CA 93943, U.S.A.

(*Received* 26 *June* 1987; *in revised form* 18 *March* 1988)

Abstract—In prior work we have given an intuitive development of Transfer Decomposition, a decomposition of the traffic assignment problem into two traffic assignment problems. The intent of this paper is to provide a rigorous basis for this technique by establishing that it is a generalized Benders decomposition. As an illustration of the result, we give a decomposition algorithm that is based on the familiar Frank–Wolfe method.

1. INTRODUCTION

In prior work, Barton and Hearn (1978) and Hearn (1984) have given intuitive developments of Transfer Decomposition, a decomposition for networks with application to aggregation of the standard traffic assignment problem. Essentially, this technique consists of partitioning a network in such a way that the original problem is transformed into two traffic assignment problems: a master problem and a subproblem. The intent of this paper is to provide a rigorous basis for this technique by showing it to be equivalent to a generalized Benders decomposition (Geoffrion, 1972) of the original traffic assignment problem.

At the outset, this equivalence may not be apparent because generalized Benders normally does not have a subproblem of the same form as the master. In the typical application of Benders Decomposition, the master problems are implicit linear programs solved by tangential approximation (Lasdon, 1970) and the subproblems are highly structured problems solved by efficient algorithms. However, the master and subproblems of Transfer Decomposition are both traffic assignment problems. This fact then leads to the development of a decomposition alogrithm that utilizes the familiar Frank–Wolfe method. A numerical example of the algorithm is given.

2. PROBLEM FORMULATION AND NOTATION

The standard traffic assignment problem may be written in node-arc formulation as:

(P1)

$$\text{minimize} \sum_{ij\in A_1} c_{ij}(x_{ij}) + \sum_{ij\in A_2} c_{ij}(y_{ij})$$

†This research was supported in part by the National Science Foundation under Grant CEE-8420830.

subject to

$$x_{ij} = \sum_k x_{ij}^k \quad ij \in A_1$$

$$y_{ij} = \sum_k y_{ij}^k \quad ij \in A_2$$

$$[B_1|B_2]\begin{bmatrix}\mathbf{x^k}\\ \mathbf{y^k}\end{bmatrix} = \mathbf{b^k} \quad k \in k$$

$$\mathbf{x^k}, \mathbf{y^k} \geq 0 \quad k \in k. \tag{1}$$

The first two sets of constraints are for notational covenience and will be implicit in the formulations that follow. Note that the double subscript ij is used to denote an arc (or link) from node i to j. The c_{ij} are assumed convex. The arcs have been arbitrarily split into two subsets, A_1 and A_2, with flows for commodity k (identified by destination) on arc ij denoted by x_{ij}^k for $ij \in A_1$ and y_{ij}^k for $ij \in A_2$. Arcs in A_1 and A_2, for example, might correspond to major arteries and local streets, respectively, in a traffic network. We write the vector $[\mathbf{x_{ij}}]^\mathbf{k}$ with components indexed by ij as $\mathbf{x^k}$. The vector $\mathbf{b^k}$ gives the net supply of "commodity k" at each node derived from a trip table T, where T_{nk} is the trip demand between origin n and destination k. Thus, commodities are indexed by destination; $b_n^k = T_{nk}$ and $b_k^k = -\sum_{n \neq k} b_n^k$.

The matrix $\mathbf{B} = [B_1|B_2]$ is the node-arc incidence matrix for the entire network, partitioned so that $B_1(B_2)$ contains columns corresponding to arcs in $A_1(A_2)$. In addition, we can also partition the rows so that $\mathbf{B}$ has the following appearance:

$$\begin{array}{cc} \text{Arcs} & \text{Arcs} \\ \text{in } A_1 & \text{in } A_2 \end{array}$$
$$\left[\begin{array}{c|c} B_{11} & 0 \\ \hline B_{12} & B_{22} \\ \hline 0 & B_{23} \end{array}\right]. \tag{2}$$

The rows for the submatrix $B_{11}(B_{23})$ correspond to nodes that are incident to arcs only from the set $A_1(A_2)$, and the rows for the submatrix $[B_{12}|B_{22}]$ correspond to nodes that are incident to arcs from both A_1 and A_2. We refer to this latter set nodes as the "interface" nodes since they represent junctions at which the flows transfer from A_1 (major arteries) to A_2 (local streets). For simplicity, it is assumed that all interface nodes and nodes corresponding to rows of B_{11} have neither supply nor demand, i.e. the corresponding components of the vector $\mathbf{b^k}$ are zero for all k. By this assumption, trips are permitted to originate from local streets (e.g. a residential area) and to terminate at other local streets (e.g. the downtown area). However, this assumption can be relaxed by adding additional dummy nodes and arcs with zero cost.

3. TRANSFER DECOMPOSITION MOTIVATION AND THEORY

Transfer Decomposition was developed during a study of an ad hoc decomposition process used by transportation planners to solve large (10,000 links) traffic assignment problems (Wilson *et al.*, 1974; Hearn, 1978). The steps in this process were: (1) extract a relatively small subnetwork of primary interest, (2) transfer part or all of the complete trip demand matrix to a subnetwork demand matrix, and (3) flow the smaller tractable network. Although these steps may not yield an optimal solution, they resemble steps in Benders decomposition in that the extraction of a subnetwork of interest in Step 1 is simply the partitioning of columns of the node-arc incidence matrix as described above and the transferring of the trip demand matrix in Step 2 provides a *partial* "communication" link between the subnetwork and the rest of the network. In Transfer Decomposition, these steps are made rigorous and the missing communication link is provided.

In the remainder of this section, we develop Transfer Decomposition in an intuitive manner and show that the model structure is of the generalized Benders form. Finally, we also adapt the familiar Frank–Wolfe algorithm to the decomposition.

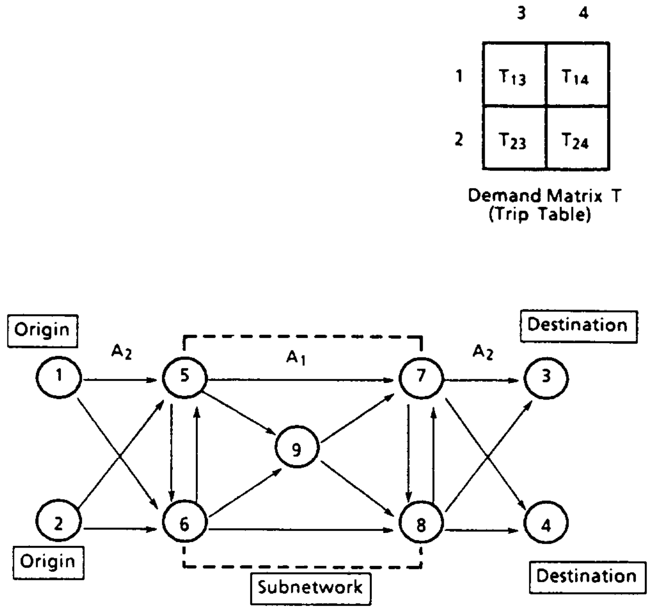


Fig. 1. Example problem.

Motivational example

Consider the example of Fig. 1. The arcs of the subnetwork of interest are indexed by A_1, and all others by A_2. Thus

$$A_1 = \{56, 65, 57, 59, 68, 69, 97, 98\}$$

$$A_2 = \{15, 16, 25, 26, 73, 74, 83, 84\}.$$

Note that the nodes have been numbered to yield a node-arc incidence matrix of form (2). Now assume that the flows y_{ij}, $ij \in A_2$ are *fixed* at values feasible to the flow conservation conditions, i.e. there exist $x_{ij}^k \geq 0$ such that (1) holds. Then, intuitively, and it will be proven below, there exists an *induced* demand matrix (trip table) for the subnetwork. By the additive nature of (P1), the optimal set of x_{ij} for the fixed y_{ij} may be obtained by solving

$$\min \sum_{ij \in A_1} c_{ij}(x_{ij})$$

subject to conservation of flow constraints on the subnetwork with respect to the demand matrix induced by the fixed y_{ij}. As we show later, the demand matrix of the subproblem need not retain the original commodity identification given in (P1). For example, flows of commodity "4" from node 1 to node 4 through nodes 5, 9, and 8 in (P1) are merely represented as flows from 5 to 8 in the subproblem, and are combined with all other flows leaving the subnetwork at 8 to form subnetwork commodity "8."

Based on these observations, the example problem is reformulated as two network flow problems: a master and a subproblem, as shown in Fig. 2. Note that the pseudo-links (broken lines) of the master network represent feasible paths on the subnetwork and the flow on these pseudo-links provide (induce) the trip demand matrix for the subproblem. This provides communication from the master problem to the subproblem, and *guarantess feasibility of the communicated demand matrix.* To complete the loop, the subproblem optimal solution supplies the master problem with pseudo-link "costs" obtained from the subproblem travel times.

From this example, one can easily see the connection with *aggregation* in the traffic assignment problem. The pseudo-links of the master problem represent a *path-flow* aggregation of the *link-flow* activity in the subnetwork. Evidently, if the travel times on the pseudo-links are somehow obtainable to sufficient accuracy, the master problem can

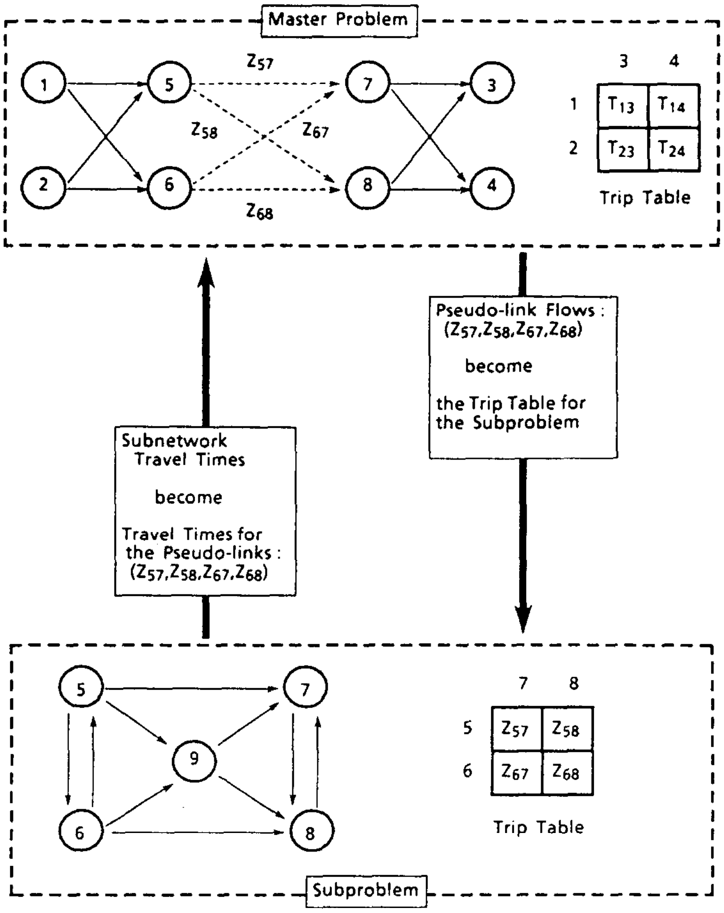


Fig. 2. Master and subproblem of Transfer Decomposition.

be taken as an aggregated version of the original. The connection with aggregation practices is discussed by Hearn (1984) and Friesz (1985). For related aggregation studies see Bovy and Jansen (1983) and Haghani and Daskin (1983).

From a computational viewpoint it appears that the original model (P1) can be solved by alternately solving the master and subproblems, possibly with different traffic assignment algorithms, if that should be advantageous.

In order to justify such uses, it is necessary that the decomposition be investigated theoretically.

Relation to generalized Benders decomposition

Geoffrion (1972) has provided a generalization of the original Benders decomposition that is applicable to nonlinear programs such as (P1) in which the variables are partitioned into two sets. To provide a rigorous theoretical framework for Transfer Decomposition, we will derive a generalized Benders problem (P3) that is equivalent (through (P2)) to (P1). We show that the original commodities can be dropped from the subproblem, producing (P4), the Transfer Decomposition formulation.

For the purpose of decomposing (P1), it is assumed that the partitioning of the node-arc incidence matrix into $[B_1|B_2]$ induces the existence of another node-arc incidence matrix $\mathbf{D}$ which facilitates the following reformulation of (P1):

(P2)

$$\min \sum_{ij\in A_1} c_{ij}(x_{ij}) + \sum_{ij\in A_2} c_{ij}(y_{ij})$$

subject to

$$\mathbf{D}\mathbf{z}^k + B_2\mathbf{y}^k = \mathbf{b}^k \quad k \in K$$

$$B_1\mathbf{x}^k = \mathbf{D}\mathbf{z}^k \quad k \in K$$

$$\mathbf{x}^k, \mathbf{y}^k, \mathbf{z}^k \geq 0 \quad k \in K.$$

In other words, it is assumed that (P1) is equivalent to (P2). Figure 3 illustrates the existence of the matrix **D** for the example in Figs. 1 and 2. Note that the pseudo-arcs correspond to columns of **D** and that $\overline{A}_1$ will denote the set of pseudo-arcs henceforth.

The matrix **D** represents possible paths between all pairs of interface nodes in the subnetwork. This allows the following result.

LEMMA 3.1.
Let $\Phi = \{\mathbf{Dz} : \mathbf{z} \geq 0\}$ and $\Omega = \{\mathrm{B}_1\mathbf{x} : \mathbf{x} \geq 0\}$. Then, $\Phi = \Omega$.

Proof. The space Φ corresponds to net node flows in the arc-chain representation for the subnetwork composing of arcs in A_1 and the space Ω corresponds to net node flows in the node-arc representation for the same subnetwork. By Theorem 2.2 of Ford and Fulkerson (1962) these are equivalent. ■

Let S denote the set of all interface nodes. Then, **D** can be constructed by finding for each node $s \in S$ a directed tree rooted at s that spans the maximum number of nodes in the subnetwork defined by A_1. For each $t \in S$ that is reachable from s, a column corresponding to a pseudo-arc (s, t) is added to **D**. However, the connectivity structure of the subnetwork corresponding to A_1 will be known in many instances, and the structure of **D** will be obvious. For example, if any interface node is reachable from any other

	56	57	59	65	68	69	78	87	97	98	:	15	16	25	26	73	74	83	84
1	0	0	0	0	0	0	0	0	0	0	:	1	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	:	0	0	1	1	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	:	0	0	0	0	-1	0	-1	0
4	0	0	0	0	0	0	0	0	0	0	:	0	0	0	0	0	-1	0	-1
5	1	1	1	-1	0	0	0	0	0	0	:	-1	0	-1	0	0	0	0	0
6	-1	0	0	1	1	1	0	0	0	0	:	0	-1	0	-1	0	0	0	0
7	0	-1	0	0	0	0	1	-1	-1	0	:	0	0	0	0	1	1	0	0
8	0	0	0	0	-1	0	-1	1	0	-1	:	0	0	0	0	0	0	1	1
9	0	0	-1	0	0	-1	0	0	1	1	:	0	0	0	0	0	0	0	0

B_1 B_2

	$\overline{57}$	$\overline{58}$	$\overline{67}$	$\overline{68}$	:	15	16	25	26	73	74	83	84
1	0	0	0	0	:	1	1	0	0	0	0	0	0
2	0	0	0	0	:	0	0	1	1	0	0	0	0
3	0	0	0	0	:	0	0	0	0	-1	0	-1	0
4	0	0	0	0	:	0	0	0	0	0	-1	0	-1
5	1	1	0	0	:	-1	0	-1	0	0	0	0	0
6	0	0	1	1	:	0	-1	0	-1	0	0	0	0
7	-1	0	-1	0	:	0	0	0	0	1	1	0	0
8	0	-1	0	-1	:	0	0	0	0	0	0	1	1
9	0	0	0	0	:	0	0	0	0	0	0	0	0

D B_2

	56	57	59	65	68	69	78	87	97	98
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	1	1	1	-1	0	0	0	0	0	0
6	-1	0	0	1	1	1	0	0	0	0
7	0	-1	0	0	0	0	1	-1	-1	0
8	0	0	0	0	-1	0	-1	1	0	-1
9	0	0	-1	0	0	-1	0	0	1	1

B_1

Fig. 3. The existence of the matrix **D** for the example in Figs. 1 and 2.

interface node using only arcs in A_1, then $\mathbf{D}$ is the node-arc incidence matrix of a complete directed graph on the node set S. This is often the case of the ad hoc decomposition process where the subnetwork consists of major highways.

By partitioning the variables into 2 sets: $(\mathbf{y}, \mathbf{z})$ and $\mathbf{x}$ and applying generalized Benders decomposition to (P2) , we obtain

(P3M)

$$\min v(\mathbf{z}) + \sum_{ij \in A_2} c_{ij}(y_{ij})$$

subject to

$$\mathbf{D}\mathbf{z}^k + B_2\mathbf{y}^k = \mathbf{b}^k \quad k \in K$$

$$\mathbf{y}^k, \mathbf{z}^k \geq 0 \quad k \in K.$$

(P3S)

$$v(\mathbf{z}) = \min \sum_{ij \in A_1} c_{ij}(x_{ij})$$

subject to

$$B_1\mathbf{x}^k = \mathbf{D}\mathbf{z}^k \quad k \in K$$

$$\mathbf{x}^k \geq 0 \quad k \in K.$$

By Theorem 2.1 of Geoffrion (1972) for the generalized Benders Decomposition we have

LEMMA 3.2

(P2) *and* (P3) *are equivalent.*

Note that Lemma 3.1 guarantees that if $(\mathbf{y}, \mathbf{z})$ feasible to (P3M), then there exists an $\mathbf{x}$ feasible to (P3S). However, the vector $\mathbf{D}\mathbf{z}^k$ does not generally have the same form as $\mathbf{b}^k$, i.e. the right hand side vector for (P3S) does not have an associated trip matrix. For the example in Figs. 1 and 2, let $\mathbf{z}^k = (10, 0, 20, 0)^t$ then

$$\mathbf{D}\mathbf{z}^k = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 20 \\ -10 \\ -20 \\ 0 \end{bmatrix}$$

and the original commodity index does not correspond to trips with the same destination in the subproblem, (P3S). In this case, $\mathbf{D}\mathbf{z}^k$ can be decomposed into two commodities as follows:

$$\mathbf{D}\mathbf{z}^k = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 0 \\ -10 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 20 \\ 0 \\ -20 \\ 0 \end{bmatrix}.$$

Below, it is shown that there exists a traffic assignment (TA) problem, (P4S), equivalent to (P3S).

Let $\mathbf{D}$ be a matrix of size $N \times L$, where N is number of nodes in the original network and L is number of pseudo-arcs, and d_{ij} denote the (i, j) element of $\mathbf{D}$. Since $\mathbf{D}$ is a node-arc incidence matrix, the rows of $\mathbf{D}$ are linearly dependent and sum to zero, i.e.

$$\sum_j \left(\sum_i d_{ij} \right) z_j^k = 0$$

or (3)

$$\sum_j [\mathbf{Dz^k}]_j = 0,$$

where $[w]_j$ denotes the jth component of $\mathbf{w}$. Let $J_k^+ = \{j : [\mathbf{Dz^k}]_j > 0\}$ and $J_k^- = \{j : [\mathbf{Dz^k}]_j < 0\}$. Equation (3) then implies that

$$\sum_{j \in J_k^+} [\mathbf{Dz^k}]_j = - \sum_{j \in J_k^-} [\mathbf{Dz^k}]_j = \alpha^k \geq 0.$$

For the remainder, it is assumed that $\alpha^k > 0$, for otherwise the kth commodity can be discarded from the subproblem. Let $T^k = \{(i, j) : i \in J_k^+ \text{ and } j \in J_k^-\}$; then, T^k indexes a set of "subtrips" for the original commodity k, and the lemma below shows how to construct a trip matrix from $\mathbf{Dz^k}$, $k \in K$.

LEMMA 3.3.
For each $k \in K$, there exists a set of constant β_{pq}^k for all $(p, q) \in T^k$ such that

(i) $\beta_{pq}^k \geq 0$

(ii) $\sum_{(p,q) \in T^k} \beta_{pq}^k = \alpha^k$

(iii) $\sum_{q \in J_k^-} \beta_{pq}^k = [\mathbf{Dz^k}]_p \quad p \in J_k^+$

(iv) $\sum_{p \in J_k^+} -\beta_{pq}^k = -[\mathbf{Dz^k}]_q \quad q \in J_k^-$.

Proof. Conditions (i), (iii), and (iv) constitute the constraint set of the Hitchcock–Koopman transportation problem. Equation (3) above guarantees that total demand equals total supply, which implies that the feasible region is nonempty. Thus, there exist a set of β_{pq}^k satisfying (i), (iii), and (iv). Then, condition (ii) is obtained from summing condition (iii) over p or condition (iv) over q. ■

Since columns of the transportation problem have the form $(\mathbf{e_p} - \mathbf{e_q})$ where $\mathbf{e_p}$ is the pth unit vector in R^N,

$$\mathbf{Dz^k} = \sum_{(p,q) \in T^k} \beta_{pq}^k (\mathbf{e_p} - \mathbf{e_q}) = \sum_p \sum_q \beta_{pq}^k (\mathbf{e_p} - \mathbf{e_q}),$$

where $\beta_{pq}^k = 0$ if $(p, q) \notin T^k$ and the summation over p and q both range from 1 to N. That is, Lemma 3.3 permits us to decompose the vector $\mathbf{Dz^k}$ into vectors that are indexed by node numbers, i.e. ps and qs.

Given that β_{pq}^k is chosen for all p, q, and k, then we can define for any given $\mathbf{z^k}$

$$\mathbf{w^q} = \sum_k \sum_p \beta_{pq}^k (\mathbf{e_p} - \mathbf{e_q}) \quad \text{for all } q$$

and $S(\mathbf{w^q}) = \{\mathbf{f^q} : B_1\mathbf{f^q} = \mathbf{w^q} : \mathbf{f^q} \geq 0, q = 1, \ldots, N\}$. Note that w_q depends on βs which, in turn, depends on $\mathbf{z^k}$. Moreover,

$$\mathbf{w^q} = \sum_k\sum_p \beta^k_{pq}\mathbf{e_p} - \left(\sum_k\sum_p \beta^k_{pq}\right)\mathbf{e_q}$$

implies that $w^q_j \geq 0, j = 1, \ldots, (q-1), (q+1), \ldots, N$, and $w^q_q = -\sum_{j\neq q} w^q_j$. Hence, $\mathbf{w^q}$ has the same form as $\mathbf{b^k}$ in problem (P1) and one can easily derive a subnetwork trip matrix from $\mathbf{w^q}$. Thus, $S(\mathbf{w^q})$ is a feasible region for a TA problem. Define the feasible region for (P3S) as follows:

$$S(\mathbf{z^k}) = \{\mathbf{x^k} : B_1\mathbf{x^k} = \mathbf{Dz^k}; \mathbf{x^k} \geq 0, k \in K\}.$$

Then, the lemma below establishes the equivalence of the two feasible regions.

LEMMA 3.4.
$S(\mathbf{z^k})$ is equivalent to $S(\mathbf{w^q})$ in that $\mathbf{x^k} \in S(\mathbf{z^k})$ induces a $\mathbf{f^q} \in S(\mathbf{w^q})$ and vice versa.

Proof. For any $\mathbf{x^k} \in S(\mathbf{z^k})$, $\mathbf{x^k} \geq 0$ represents a flow vector satisfying $B_1\mathbf{x^k} = \mathbf{Dz^k}$. Consider the set of arc (i, j) with positive flow, i.e. the ijth component of the vecotr $\mathbf{x^k}$ is positive. Because $\mathbf{x^k}$ is feasible, these arcs with positive flow must form paths which connect pairs (p, q) in T^k. Then, $\mathbf{x^k}$ must be composed of flow vectors, $\mathbf{x^{k(p,q)}}$, for $(p, q) \in T^k$, which has the same dimension as $\mathbf{x^k}$ and represents sending some β^k_{pq} units of flow for commodity k along a path consisting of arcs with positive flow and connecting p to q. That is,

$$\mathbf{x^k} = \sum_p\sum_q \mathbf{x^{k(p,q)}}, \text{ and}$$
$$B_1\mathbf{x^{k(p,q)}} = \beta^k_{pq}(\mathbf{e_p} - \mathbf{e_q}), \tag{4}$$

where $\mathbf{x^{k(p,q)}} = 0$ and $\beta^k_{pq} = 0$ for (p, q) not in T^k. Since $\mathbf{x^k}$ is feasible, β^k_{pq} must also satisfy the conditions in Lemma 3.3. From (4), we have that

$$B_1\sum_k\sum_p \mathbf{x^{k(p,q)}} = \sum_k\sum_p \beta^k_{pq}(\mathbf{e_p} - \mathbf{e_q}) = \mathbf{w^q}.$$

Let $\mathbf{f^q} = \Sigma_k\Sigma_p \mathbf{x^{k(p,q)}}$; then $\mathbf{f^q} \in S(\mathbf{w^q})$.

For the converse, if $\mathbf{f^q} \in S(\mathbf{w^q})$, then $B_1\mathbf{f^q} = \mathbf{w^q}$. By a similar argument, we have that

$$\mathbf{f^q} = \sum_p\sum_q \mathbf{f^{k(p,q)}}, \text{ and}$$
$$B_1\mathbf{f^{k(p,q)}} = \beta^k_{pq}(\mathbf{e_p} - \mathbf{e_q}),$$

where $\mathbf{f^{k(p,q)}} = 0$ when $\beta^k_{pq} = 0$, and β^k_{pq}, for all $(p,q) \in T^k$, satisfy conditions in Lemma 3.3. By letting $\mathbf{x^k} = \Sigma_p\Sigma_q \mathbf{f^{k(p,q)}}$, $\mathbf{x^k}$ is a member of $S(z^k)$. ■

Define (P4) as follows:

(P4M)
$$\min v(\mathbf{z}) + \sum_{ij\in A_2} c_{ij}(y_{ij})$$

subject to

$$\mathbf{Dz^k} + B_2\mathbf{y^k} = \mathbf{b^k} \quad k \in K$$
$$\mathbf{y^k} : \mathbf{z^k} \geq 0 \quad k \in K.$$

(P4S)
$$v(\mathbf{z}) \equiv \min \sum_{ij\in A_1} c_{ij}(f_{ij})$$

subject to

$$B_1\mathbf{f}^q = \mathbf{w}^q \quad q = 1, \ldots, N$$
$$\mathbf{f}^q \geq 0 \quad q = 1, \ldots, N.$$

THEOREM 3.1. (P1) *and* (P4) *are equivalent.*

Proof. (P1) and (P2) are equivalent by Lemma 3.1, (P2) and (P3) are equivalent by Lemma 3.2, and (P3) and (P4) are equivalent by Lemma 3.4.■

This theorem, which shows Transfer Decomposition to be a generalized Benders decomposition, has as corollaries the many results proven by Geoffrion (1972) for generalized Benders problems. In particular, we cite

COROLLARY 3.1a. *$v(\mathbf{z})$ is convex for any $\mathbf{z}$ feasible in* (P4M).

COROLLARY 3.1b. *For a fixed* $\mathbf{y}$, *$\mathbf{z}$ feasible in the master problem of* (P4M), *any set of optimal multipliers of* (P4S) *is a set of subderivatives of $v(\mathbf{z})$. The function v is differentiable if and only if the set of multipliers is unique.*

COROLLARY 3.1c. *If a solution of the master problem of* (P4M) *is within ϵ_1 of optimality and the corresponding solution of* (P4S) *is within ϵ_2 of optimality, then the combined solution is within $\epsilon_1 + \epsilon_2$ of optimality for* (P1).

For the example in Figs. 1 and 2, the constraints of (P4S) can be written as:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 1 \end{bmatrix} \mathbf{f}^q = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ 0 \end{bmatrix}$$

and the superfluous rows of zeros can be deleted to obtain a reduced set of constraints that is truly a set of flow conservation equations. Moreover, $v(\mathbf{z}) = v(z_{57}, z_{58}, z_{67}, z_{68})$ in the example. It can then be easily shown that the subderivatives of $v(\mathbf{z})$ are the path times from node 5 to 7, 5 to 8, 6 to 7, and 6 to 8 at the optimal subnetwork flow. In general, the subderivatives of $v(\mathbf{z})$ are the path times for pairs (p, q) in $\overline{A}_1$.

In summary, the development given here justifies the intuitive development given earlier. The master problem (P4M) is a traffic assignment problem as is the subproblem (P4S). To demonstrate the usefulness of this result, we next utilize a popular traffic assignment algorithm within the Transfer Decomposition framework.

4. A TRANSFER DECOMPOSITION ALGORITHM

Adapting Frank–Wolfe to Transfer Decomposition

Previous analysis verifies that both (P4M) and (P4S) have the same form as the original problem (P1). Thus, any algorithm for (P1) can be employed for both the master and subproblems of Transfer Decomposition. One popular method for large-scale problems is the algorithm of Frank and Wolfe (1956) as adapted by LeBlanc *et al.* (1975). In the following we adapt the Frank–Wolfe algorithm to (P4) by using it for (P4M) and allowing any traffic assignment algorithm for (P4S). In this description, $\overline{A}_1$ denotes the set of pseudo-links.

Algorithm

Step 0 (*Initialization*). Select tolerances ϵ_1 and ϵ_2 for problems (P4M) and (P4S). Choose an arbitrary set of flows $(\mathbf{y}, \mathbf{z})$ feasible for problem (P4M). Set $LB = -\infty$.

Step 1 (*Solve* P4S). Construct the subnetwork demand matrix from $\mathbf{z}$. Solve (P4S) for $v(\mathbf{z})$ within ϵ_2 by *any* user-equilibrium traffic assignment algorithm. Let τ_{ij} = path times (dual variables) for $(i, j) \in \bar{A}_1$.

Step 2 (*Linearize* P4M). Linearize the arc costs of (P4M) for all $ij \in A_2 \cup \bar{A}_1$ by the formulas

$$d_{ij} = \begin{cases} \dfrac{dc_{ij}(y_{ij})}{dy_{ij}} & ij \in A_2 \\ \tau_{ij} & ij \in \bar{A}_1. \end{cases}$$

Assign all demands (in T) to minimum paths with respect to d_{ij}. Denote this set of flows by $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$.

Step 3 (*Update lower bound and test for termination*). Let

$$\overline{LB} = \sum_{ij} c_{ij}(y_{ij}) + v(\mathbf{z}) + \sum_{ij \in A_2} d_{ij}(\bar{\mathbf{y}} - \mathbf{y}) + \sum_{ij \in \bar{A}_1} d_{ij}(\bar{\mathbf{z}} - \mathbf{z}).$$

$$\text{Set } LB = \max[LB, \overline{LB}]. \text{ If } \sum_{ij \in A_2} c_{ij}(y_{ij}) + v(\mathbf{z}) - LB \le \epsilon_1 + \epsilon_2, \text{ stop.}$$

Otherwise, go to Step 4.

Step 4 (P4M *Line Search*). Search the line segment $[(\mathbf{y}, \mathbf{z}), (\bar{\mathbf{y}}, \bar{\mathbf{z}})]$ for an improved solution, $(\mathbf{y}'', \mathbf{z}'')$, of (P4M). Replace $(\mathbf{y}, \mathbf{z})$ by $(\mathbf{y}'', \mathbf{z}'')$ and go to Step 1.

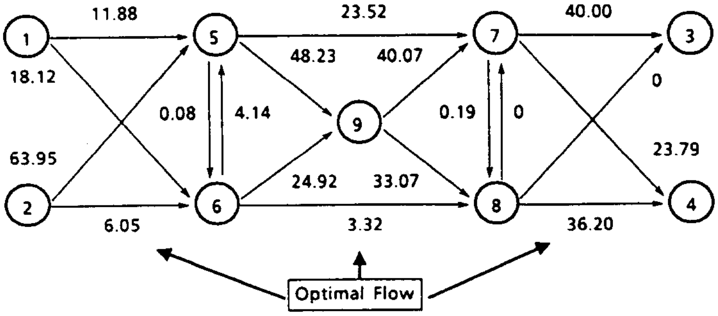
Of course, particular care must be exercised in Step 4, the line search. Since each iteration requires a resolving of problem (P4S), the method is only practical if just a few points on the line need be evaluated. A heuristic variation performs the line search, not on the objective of (P4M), but on a function $g(\mathbf{y}, \mathbf{z})$ that is defined to be the maximum of all tangent planes generated in prior iterations. The information required to evaluate $g(\mathbf{y}, \mathbf{z})$ is easily obtained by employing Corollary 3.1b. It must be stored, so the heuristic would require more computer memory, but the trade-off with time required to solve the subproblems could be substantial. This heuristic does not produce monotonically decreasing objective values for (P4M).

Convergence properties of this algorithm can be established, but the details are beyond the scope of this paper, except to note that the known convergence results of the Frank–Wolfe method insure global convergence whenever $v(\mathbf{z})$ is continuously differentiable (see Corollary 3.1a).

A numerical example

It is instructive to consider a numerical example of the algorithm. Figure 4 gives traffic assignment data for the example network. The "volume delay" formula, typical of transportation planning models, represents the time to traverse each link as a function of the link volume, or flow. For this problem, the c_{ij} are integrals of the volume delay formula. The constant T_0 is the uncongested travel time for the link, and the "Capacity" term is capacity only in a penalty sense. The optimal objective value of (P1) is for the *user equilibrium* model of traffic flow. The fixed trip demand matrix requires that a total of 100 trips traverse the network from origins 1 and 2 to destinations 3 and 4. The number beside each link represents a user-optimal solution in terms of total link flow.

The initial iterations on this problem are illustrated in Fig. 5. The subnetwork trip tables passed from (P4M) to (P4S) are in the first column. In effect, these are extreme points of the domain of (P4M) with respect to the pseudo-link flow variables (z_{57}, z_{58}, z_{67}, z_{68}). Step 4 of the algorithm, the line search, transfers a convex combination of all past trip tables to the subproblem. These are shown in the second column. Step 2 of the algorithm solves the subnetwork traffic assignment problems with the trip table of



Optimal Objective Value = 1453

Volume Delay =

$T_0(1.0 + 0.15(\text{Flow/Capacity})^4)$

	3	4
1	10	20
2	30	40

Trip Table

Link	T_0	Capacity
1-5	5	10
1-6	6	16
2-5	3	35
2-6	9	18
5-6	1	50
5-7	5	25
5-9	2	35
6-5	1	50
6-8	5	25

Link	T_0	Capacity
6-9	2	35
7-3	3	25
7-4	6	24
7-8	1	50
8-7	8	39
8-3	6	43
8-4	1	50
9-7	2	35
9-8	2	25

Fig. 4. Example problem data.

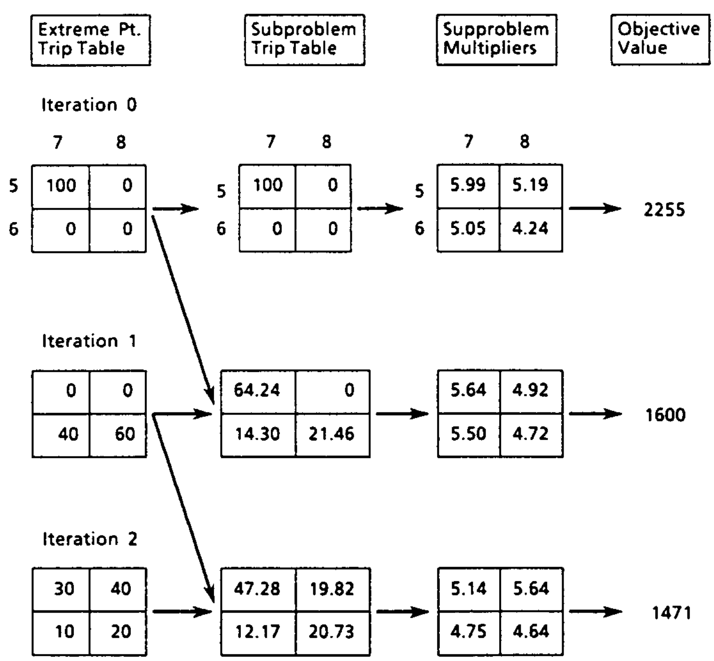


Fig. 5. Iterations of the algorithm on the example problem.

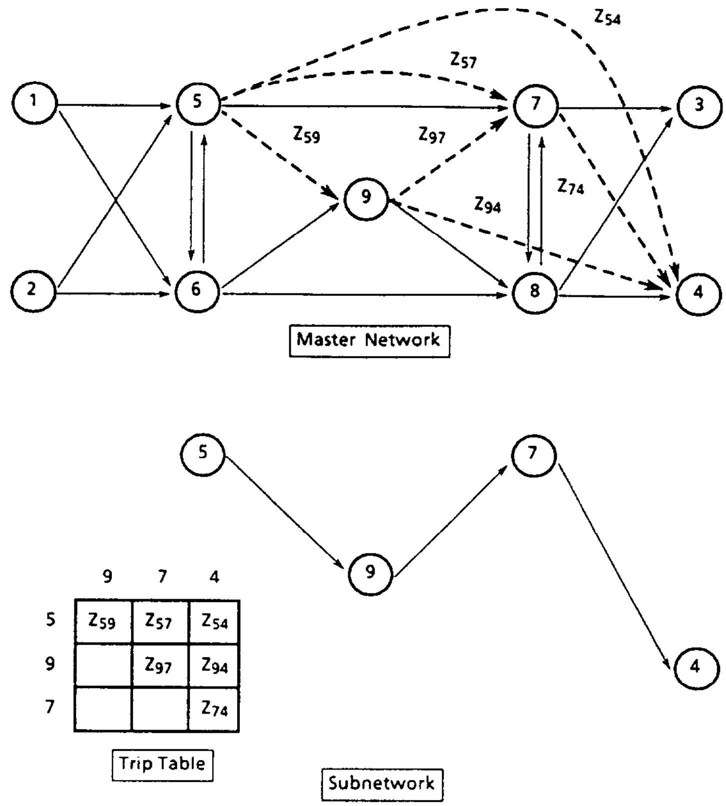


Fig. 6. Alternative Transfer Decomposition of the example problem.

column 3 and produces the multipliers (travel times) shown in the third column. Successive values of the objective for (P4M) are shown in the final column. Note that in only three iterations the subnetwork trip table, a convex combination of the three trip tables of column two produces a solution within 2% of the optimal value. The travel times (multipliers) for the traversal of the subnetwork are approaching their optimal values of (5.60, 6.00, 4.60, 5.00).

5. ALTERNATIVE TRANSFER DECOMPOSITIONS

The subnetwork of the decomposition in Fig. 1 is a "natural" one in that the network divides into disjoint segments and the pseudo-links have an easily visualized interpretation. Other choices of a subnetwork are possible, however. Consider Fig. 6, where the same network as before has been decomposed so that the links (59, 97, 74) comprise the subnetwork. One advantage of this decomposition is that the subnetwork is a tree and therefore the execution of Step 1, solution of subproblem (P4S), is trival. The disadvantage is that the outer network is three links larger than the original network. Note, however, that the subnetwork demand matrix is not unique for this choice of decomposition (i.e. the β_{pq}s in Lemma 3.3 are not necessarily unique). If links associated with z_{54} and z_{94} were removed from the outer network, the subnetwork could be as shown and the demand matrix would only contain z_{59}, z_{97}, and z_{74}. The important difference is that the pseudo-links of the outer network always have "costs" that sum to $v(\mathbf{z})$, even if the pseudo-link replaces an original link.

6. SUMMARY

Transfer Decomposition provides a mathematical representation for the ad hoc decomposition process. This makes it possible to characterize the strengths and weaknesses of the transportation planning process. The decomposition method may be efficient for models where A_1 is large but with a small number interface nodes or where the network induced by A_1 is of a special structure. The final example shows that Transfer Decomposition is quite general and the decomposition can be varied to suit different needs.

Acknowledgement—Several comments by referees have helped to clarify the exposition of this paper, particularly on the existence of **D** and the proof of Lemma 3.3.

REFERENCES

Barton R. R. and Hearn D. W. (1978) Study of network aggregation in transportation planning models. Mathtech Final Report, Contract DOT-TSC-1443. Princeton, New Jersey.

Bovy P. H. L. and Jansen G. R. M. (1983) Network aggregation effects upon equilibrium assignment outcomes: An empirical investigation. *Transpn. Sci.* **17**(3), 240–262.

Ford L. and Fulkerson D. R. (1962) *Flows in Networks*. Princeton University Press, Princeton, New Jersey.

Frank M. and Wolfe P. (1956) An algorithm for quadratic programming. *Naval Res. Logist. Q.* **3**, 95–110.

Friesz T. L. (1985) Transportation network equilibrium, design and aggregation: Key developments and research opportunities. *Transp. Res.* **19A**(5/6), 413–427.

Geoffrion A. M. (1972) Generalized Benders decomposition. *J. Optim. Theory Appl.* **10**(4), 237–260.

Haghani A. E. and Daskin M. S. (1983) A network design application of an extraction algorithm for network aggregation. *Transp. Res. Rec.* **944**, 37–46.

Hearn D. W. (1978) Network aggregation in transportation planning. Mathtech Final Report, Contract DOT-TSC-RSPD-78-8. Princeton, New Jersey.

Hearn D. W. (1984) Practical and theoretical aspects of aggregation problems in transportation planning models. In *Transportation Planning Models*, 257–287(Edited by M. Florian). Elsevier Science Publishers, Amsterdam.

Lasdon L. S. (1970) *Optimization Theory for Large Systems*. MacMillan Publishing Company, New York.

LeBlanc L. J., Morlok E. K. and Pierskalla W. P. (1975) An accurate and efficient approach to equilibrium traffic assignment on congested networks. *Transp. Res.* **9**(5), 309–318.

Wilson E. M., Matthias J. S. and Bets M. J. (1974) A traffic assignment planning model: The load-node concept. *Transp. Res.* **8**, 75–84.