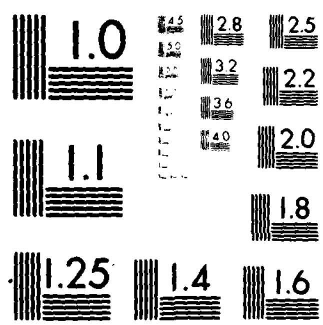
AD-A084 283 TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES F/G 12/2
SOLUTION APPROACHES FOR NETWORK FLOW PROBLEMS WITH MULTIPLE CRI--ETC(U)
DEC 79 D KLINGMAN, J MOTE N00014-78-C-0222
UNCLASSIFIED CCS-363 NL

1 OF 1
AD A

END
DATE
FILMED
6-80
DTIC



MICROCOPY RESOLUTION TEST CHART

(1)

LEVEL II

CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

14

CCS 363

6

SOLUTION APPROACHES FOR NETWORK FLOW PROBLEMS WITH MULTIPLE CRITERIA

by

10

Darwin Klingman*
John Mote**

12 41

11

December 1979

* Professor of Operations Research and Computer Sciences, The University of Texas at Austin, Department of General Business, BEB 608, Austin, TX 78712

** Systems Analyst, Analysis, Research, & Computation, Inc., P.O. Box 4067, Austin, TX 78765.

15

This research was supported in part by the Office of Naval Research contract no. N00014-78-C-0222. Reproduction in whole or in part is permitted for any purpose of the United States Government.

Accession For	
NTIS GRA&I	☑
DDC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
BEB 203E
The University of Texas at Austin
Austin, Texas 78712
(512) 471-1821

406 197

1. INTRODUCTION

The past thirty years have witnessed a marked increase in the acceptance of mathematical modeling and solution techniques by both the business and government sectors. This acceptance of the tools of operations research has been necessitated to some extent by the ever-increasing complexity in the decision-making environment. Typical problem areas faced by today's managers can involve hundreds of decision variables, often with many potential levels for each decision, hundreds of interacting constraints, and typically a number of conflicting policies or goals to be obtained. The decision maker must be able to simultaneously analyze the impact of each potential decision alternative. Then, in view of the set of goals for the problem, he must select the "best" decision to implement. To complicate matters even further, many of these problems must be addressed on a day-to-day basis as new, and often conflicting, information is provided to the decision maker.

Much of the success of operations research has been due to its ability to satisfactorily model (i.e., reduce to a "simple" mathematical abstraction) some of the important decision problems faced by management. Among the most successful fields within the broad category of operations research is the field of network flow analysis. Quite simply, this field deals with those problems that can be modeled to some degree as an interacting network of points or nodes and connecting links or arcs. Pictorially, a node may be used to represent such physical entities as a factory, a warehouse, or a customer, and an arc may be used to capture the production at a factory, the inventory level in a warehouse, or the shipment of a product to a customer.

Numerous algorithms have been developed for solving network flow problems that have a single objective function that is to be minimized or maximized. These include network specializations of more general purpose algorithms [1 , 4 , 6], as well as new algorithms solely designed for special classes of network problems [2 , 5 , 9]. Far less effort has been invested in the development of algorithms for solving network flow problems that have multiple objectives or criteria.

There have been a variety of approaches proposed for solving multiple criteria linear programming problems. The complexities of these techniques range from the very simple-minded to the highly involved.

Undoubtedly the simplest approach, and probably the one most practiced, is to solve the problem independently for each of the criteria functions. There is an obvious trade-off for this approach. On the one hand, since the individual problems are simply single criterion problems, they can be solved with any of the efficient algorithms designed for their class. But on the other hand, the decision maker is faced with (possibly) as many "solutions" as there are criteria.

Another approach for solving a linear programming problem with multiple criteria is to form a single surrogate criterion from the multiple criteria. Here the decision maker must specify exact weights, or trade-off values, for each of the criterion. Given these weights, the resulting single criterion problem can be solved with any of the existing algorithms for its class. Unlike the first approach, this surrogate criterion approach will produce a single "solution" to the decision maker. A drawback of this approach is that the exact weights for each criterion must be pre-specified. Of course, the decision maker is not restricted to a single choice of weights, but each such choice can possibly yield a new "solution" to the problem.

A third approach for solving multiple criteria problems is to incorporate some or all of the criteria into the problem in the form of constraints. For example, a manufacturing company might have the following three goals: maximize sales, maximize return on investment, and minimize employee overtime. One way to solve this problem is to add a constraint that total sales must exceed one million dollars, and a constraint that the average worker overtime cannot exceed five hours per week. The remaining criterion of maximizing return on investment can be used as the objective function. This problem can then be solved as a single criterion linear programming problem.

Another popular approach for solving multiple criteria problems is that of goal programming. With this approach, the decision maker typically specifies weight ranges (either priority or preemptive) for each of the criterion. Numerous goal programming specializations of the standard simplex linear programming algorithm have been developed for solving these problems.

A final approach, *multicriteria programming*, determines the set of all *nondominated* (efficient, admissable, or Pareto optimal) solutions to the problem. Loosely speaking, a solution is a nondominated solution if no other solution exists that is at least as good in terms of every criterion, and even better for at least one criterion.

In the single criterion case, enormous computational gains have been made by specializing the general purpose linear programming algorithms to take advantage of the special properties of network flow problems. Specifically, many arithmetic operations have been replaced by more efficient logical operations. In addition, the basis matrix and its inverse for many network flow problems can be represented as a simple graph structure (e.g., rooted tree). These graph structures can be stored and efficiently accessed by a computer as a set of linked lists.

Depending on the degree of complexity of the network problem being considered, numerous specialized network solution algorithms recently developed exhibit from one to three orders of magnitude improvements in solution times over the more general purpose linear programming algorithms. In view of this fact, this research effort sought to extend these advantages to network problems with multiple criteria. Two approaches were studied: a network variant of the multicriteria linear programming algorithm, and a network variant of the surrogate criterion approach.

A network variant of the multicriteria linear programming problem is presented ~~in Section~~ 2. The primal simplex multicriteria algorithm first developed by Yu and Zeleny [14, 15, 16] is specialized to handle the simple basis structure of the multicriteria uncapacitated transshipment problem. Specifically, the basis tree representation and updating techniques that have proven to be successful for single criterion network flow problems are used to substantially reduce the computational effort required for the multicriteria simplex algorithm.

In Section 2.1 the fundamental theoretical results for the general multicriteria linear programming problem are presented. A brief review of the relevant aspects of the network basis structure is given in Section 2.2. The specialized multicriteria primal simplex algorithm for the uncapacitated transshipment problem is presented ~~in~~ Section 2.3, and a small example problem is given in Section 2.4.

A network variant of the surrogate criterion linear programming approach is presented ~~in~~ Section 3. For sake of illustration, the shortest path problem is used as the class of networks to be solved. However, ~~the~~ approach outlined in this section can be easily extended to any of the other classes of network flow problems. In this section an interactive solution procedure is described that involves both the decision maker and the computer at each stage.

2.1 MULTICRITERIA LINEAR PROGRAMMING

The multicriteria solution procedure developed by Yu and Zeleny [14, 15, 16] is a generalization of the primal simplex algorithm for single criterion linear programming problems. In order to fully appreciate their generalization, some of the basic aspects of single criterion linear programming should be reviewed first.

The standard form of the single criterion linear programming problem is given by

$$\begin{aligned} &\text{Minimize} \quad cx \\ &\text{subject to:} \\ &\qquad Ax = b \\ &\qquad x \geqq 0 \end{aligned} \tag{1}$$

where A is the m x n matrix of activity coefficients, b is the m dimensional column vector of resource levels, c is the n dimensional row vector of objective function coefficients, x is the n dimensional column vector of decision variables, and 0 is an n dimensional column vector of zeros.

Without loss of generality it will be assumed that the matrix A has full row rank. A basis B is an m x m submatrix of A such that B also has full row rank. By partitioning problem (1) with respect to its basic and nonbasic components, the objective function may be rewritten as:

$$\text{Minimize} \quad \pi b + (c - \pi A)x \tag{2}$$

where $\pi = c_B B^{-1}$ is the m dimensional row vector of dual variables and c_B is the m dimensional subvector of c corresponding to the basic variables. From (2) it can be seen that a basic feasible solution to the problem is an optimal solution if

$$c - \pi A \geqq 0 \tag{3}$$

The quantity $c_k - \pi A_k$, where c_k is the k^{th} element of c and A_k is the k^{th} column of A, is called the *reduced cost* of variable x_k.

The standard form of the R-criteria linear programming problem is given by

$$\begin{aligned} &\text{"Minimize"} \quad Cx \\ &\text{subject to:} \\ &\qquad Ax = b \\ &\qquad x \geqq 0 \end{aligned} \tag{4}$$

where C is an R x n matrix of criteria function coefficients such that the r^{th} criterion is specified by the r^{th} row of C. In general, no single solution can simultaneously minimize all R criteria, hence the quotes around *minimize*. Instead, the objective of a multicriteria linear programming problem is to determine the set of *nondominated* solutions to the problem.

A feasible solution x^o is a nondominated solution to problem (4) if no other solution x^1 exists such that $Cx^1 \leqq Cx^o$ and $Cx^1 \neq Cx^o$. An important observation is that a feasible solution x^o is a nondominated solution if and only if a vector $\lambda > 0$ exists such that x^o is an optimal solution to the single criterion linear programming problem:

$$\begin{aligned} &\text{Minimize} \quad \lambda Cx \\ &\text{subject to:} \\ &\qquad Ax = b \\ &\qquad x \geqq 0 \end{aligned} \tag{5}$$

The vector λ serves as a simple weighting function to reduce the R criteria to a single weighted (surrogate) criterion.

If problem (4) is partitioned with respect to its basic and nonbasic components, then an R x m matrix of "dual variables" is given by

$$\Pi = C_B B^{-1}$$

where C_B is the R x m submatrix of C corresponding to the basic variables.

Following along the lines of (3), a basis B yields a nondominated solution if a vector $\lambda > 0$ exists such that

$$\lambda(C - \Pi A) \geqq 0 \tag{6}$$

That is, B is a nondominated basis for problem (4) if it yields an optimal basis to problem (5) for some vector of positive weights.

In [13, 16] it was shown that the set of all nondominated bases for problem (4) is *connected*. This simply means that, given one nondominated basis, the complete set can be obtained by exhaustively examining all bases adjacent to the set of currently known nondominated bases. An initial nondominated basis can be found by solving the weighted linear programming problem (5) for any specified vector $\lambda > 0$.

It was also shown in [13, 16] that the nondominance of each adjacent basis can be determined by repeatedly solving the following *nondominance subproblem*

$$\begin{aligned} \text{Maximize} \quad & 1e \\ \text{subject to:} \quad & Ie + (C - \Pi^k A)y = 0 \qquad (7) \\ & e \geqq 0 \qquad y \geqq 0 \end{aligned}$$

where 1 is the R-dimensional row vector of ones and Π^k is the matrix of dual variables associated with the adjacent basis obtained by replacing the appropriate column of B with A_k. If problem (7) has an optimal objective function value of zero for a given adjacent basis, then that basis is nondominated. Otherwise, it is a dominated basis and does not need to be further considered.

The nondominance subproblem arises as the dual linear programming problem of

$$\begin{aligned} \text{Minimize} \quad & \lambda 0 \\ \text{subject to:} \quad & \lambda(C - \Pi^k A) \geqq 0 \\ & \lambda I \qquad\qquad \geqq 1 \end{aligned}$$

Clearly, this problem is equivalent to the problem of finding a vector $\lambda > 0$ such that (6) holds.

The nondominance subproblem is relatively easy to solve since an initial feasible basis, corresponding to the identity matrix, is readily available. However, this approach suffers from the fact that a nondominance subproblem must be solved for each of the n - m bases adjacent to each nondominated basis.

A seemingly more efficient procedure for determining the nondominance of the adjacent bases was considered in [11]. This procedure makes use of the concept of *effective constraints* to determine the nondominance of each adjacent basis without having to physically pivot each nonbasic column into the basis.

If B is a nondominated basis for problem (4), then

$$\Lambda = \{\lambda > 0 \mid \lambda(C - \Pi A) \geqq 0\}$$

is a nonempty polyhedral region over which B is an optimal basis for the corresponding weighted single criterion linear programming problem (5). Each nonbasic column of problem (4) generates a constraint for Λ of the form

$$\lambda(C_k - \Pi A_k) \geqq 0$$

where C_k (A_k) is the k^{th} column of C (A). Such a constraint is called an *effective constraint* if $C_k - \Pi A_k \neq 0$ and a vector $\lambda^o \in \Lambda$ exists such that $\lambda^o(C_k - \Pi A_k) = 0$. That is, a constraint is effective if it forms a boundary of Λ.

In [11] it was shown that an adjacent basis is a nondominated basis if and only if it corresponds to an effective constraint. This result eliminates the need to solve the nondominance subproblem for each adjacent basis. However, a procedure is needed for determining whether or not a constraint is effective.

In [11] it was shown that $\lambda(C_k - \Pi A_k) \geqq 0$ is an effective constraint if and only if there exists vectors $u \geqq 0$ and $v \geqq 0$ such that

$$u - v(C - \Pi A) = 1(C - \Pi A) \qquad (8)$$

with $u_k = 0$. That is, if nonnegative vectors exist such that (8) holds and the k^{th} component of u is zero, then the adjacent basis obtained by replacing the appropriate column of B with A_k is non-dominated.

2.2 BASIS STRUCTURE CHARACTERISTICS OF PURE NETWORK FLOW PROBLEMS

A brief review of the fundamental characteristics of pure network flow problems is presented in this section. In the following section these characteristics are combined with the results presented in Section 2.1 to yield a specialized primal simplex algorithm for the multicriteria uncapacitated transshipment problem.

Each of the n arcs of an uncapacitated transshipment problem corresponds to a column of the constraint matrix A of problem (1). Likewise, each of the m nodes corresponds to a row of A. If arc k is directed from node i_k to node j_k, then column k of A has a -1 coefficient in row i_k, a +1 coefficient in row j_k, and zeros in the other m - 2 rows.

When an uncapacitated transshipment problem is formulated as (1), then c is an n dimensional row vector of objective function coefficients or arc costs, b is an m dimensional column vector of net demands, and x is an n dimensional column vector of arc flow variables. Due to its special structure, the matrix A is referred to as a node-arc incidence matrix. Since A is a unimodular matrix [7], every basic feasible solution to the uncapacitated transshipment problem is integer if b is an integer vector.

Since the rank of A is m - 1 (assuming the network is connected), one row of A can be deleted. A basis B consists of m - 1 linearly independent columns (arcs) of A. It is well-known that every such basis can be represented as a rooted basis tree. All of the standard simplex operations that normally require the basis or basis inverse matrix cab be carried out as simple operations on the basis tree.

The determination of the dual variables $\pi = c_B B^{-1}$ is greatly simplified using the tree representation of the basis inverse. Specifically, the dual variables, or node potentials, can be defined by a simple one pass, top to bottom, left to right traversal of the rooted basis tree. Due to the special structure of the pure network flow problem, the optimality condition (3) takes the simple form:

$$c_k + \pi_{i_k} - \pi_{j_k} \geqq 0$$

for each arc k, where π_{i_k} (π_{j_k}) is the node potential associated with the origin (destination) node of arc k.

If $c_e + \pi_{i_e} - \pi_{j_e} < 0$ for some e, then the arc is said to be *pivot eligible*. Given such an entering arc, the standard primal simplex minimum ratio test for determining the new basis can be greatly streamlined. Specifically, the minimum ratio can be determined by simply examining the arcs on the unique basis equivalent path in the basis tree from node i_e to node j_e. The minimum ratio, and hence the leaving arc, is determined by identifying the smallest flow on the subset of the arcs whose orientation is in the direction from node j_e to node i_e in the basis equivalent path. Once the leaving arc r has been identified the change of basis (pivot) can be carried out by updating the flows on the arcs in the basis equivalent path and restructuring the basis tree to reflect the replacement of arc r by arc e.

2.3 MULTICRITERIA PRIMAL SIMPLEX TRANSSHIPMENT ALGORITHM

The multicriteria primal simplex algorithm presented in this section is a network specialization of the original algorithm of Yu and Zeleny. Specifically, the algorithm has been specialized to take advantage of the special basis structure of the uncapacitated transshipment problem.

The multicriteria uncapacitated transshipment problem is given by:

$$\begin{aligned} &\text{"Minimize"} \quad Cx \\ &\text{subject to:} \\ &\qquad Ax = b \\ &\qquad x \geqq 0 \end{aligned} \tag{9}$$

where C is an R x n matrix of criteria function coefficients, A is an m x n node-arc incidence matrix, b is an m dimensional column vector of net demands, and x is an n dimensional column vector of arc flow variables. Since the node-arc incidence matrix A is unimodular, every basic feasible solution to (9) is integer if b is an integer vector.

Due to the special basis structure of pure network flow problems, the matrix of "dual variables"

$$\Pi = C_B B^{-1}$$

can be treated as a simple generalization of the vector of node potentials presented in Section 2.2. Specifically, each node i of the R-criteria uncapacitated transshipment problem (9) has a R dimensional column vector of "node potentials" associated with it. These vectors of node potentials can be defined by a simple one pass, top to bottom, left to right traversal of the rooted basis tree. Due to the special structure of the multicriteria uncapacitated transshipment problem, a given basic feasible solution is a non-dominated solution if a vector $\lambda > 0$ exists such that

$$\lambda(C_k + \Pi_{i_k} - \Pi_{j_k}) \geqq 0$$

for every arc k, where C_k is the k^{th} column of C and Π_{i_k} (Π_{j_k}) is the vector of dual variables associated with the origin (destination) node of arc k.

Given an entering arc e and a leaving arc r, the change of basis can be carried out by augmenting the flow on arc e and the arcs on its basis equivalent path, and restructuring the basis tree to reflect the replacement of arc r by arc e. The process of augmenting flow on the basis equivalent path is identical to that used for the single criterion network flow problem. The restructuring of the basis tree, except for the updating of the vectors of node potentials, can also be done in the same efficient manner used for the single criterion problem. The updating of the node potentials can be carried out as a straightforward generalization of the single criterion approach.

Let e denote the entering arc and let $\Delta = C_e + \Pi_{i_e} - \Pi_{j_e}$ be the vector of *reduced costs* associated with it. If arc r is the leaving arc, then deleting it from the basis tree creates two subtrees. Let N_1 be the set of nodes in the subtree that contains node j_e. The updated vectors of node potentials $\hat{\Pi}_i$ are defined by

$$\hat{\Pi}_i = \begin{cases} \Pi_i & , \text{ if } i \notin N_1 \\ \Pi_i + \Delta, & \text{if } i \in N_1 \end{cases}$$

That is, it is only necessary to update the vectors of node potentials associated with one of the subtrees. In addition, the vector of node potentials for each node in the subtree is changed by a constant amount. This updating process can be efficiently carried out in one pass through the nodes N_1 of the subtree.

For single criterion network flow problems, the node potentials are used solely to determine the reduced costs of the arcs for pivoting purposes. For multicriteria network flow problems, the node potentials are also used for solving nondominance subproblems (7) or for determining effective constraints (8). Both of these approaches require knowledge of the vector of reduced costs

$$C_k - \Pi A_k = C_k + \Pi_{i_k} - \Pi_{j_k} \tag{10}$$

for each arc k. The actual solution of (7) or (8) can be carried out as before. However, since pivoting to an adjacent basis is so easy for network problems, the apparent advantage of the effective constraint approach over the nondominance subproblem approach may have disappeared. This is particularly true since the nondominance subproblem can be solved as a revised simplex problem using a R x R basis matrix, whereas the effective constraint problem requires a basis matrix that may be as large as (n - m) x (n - m). Clearly, for problems where the number of criterion is small, or the number of arcs is large, the difficulty of the effective constraint problem is much greater than that of the nondominance subproblem.

At this point a specialization of the nondominance subproblem will be presented. First it should be noted that the nondominance subproblem (7) can be restated as:

$$\begin{aligned} &\text{Minimize} \quad gy \\ &\text{subject to:} \quad Gy \leqq 0 \\ &\qquad\qquad\qquad\; y \geqq 0 \end{aligned} \tag{11}$$

where $G = C - \Pi^k A$, $g = 1G$, and Π^k is the R x m matrix of node potentials associated with the adjacent basis (tree) obtained by replacing the appropriate leaving arc with nonbasic arc k. Actually, in problem (11), it is only necessary to consider the columns of G that have at least one negative coefficient. At the bare minimum, this eliminates the m (zero) columns corresponding to the basis.

Since each constraint in problem (11) has a zero right hand side value, the problem either has an optimal objective function value of zero or is unbounded. If the optimal objective function value is zero, then the adjacent basis being considered is nondominated. Otherwise, the adjacent basis is dominated.

Problem (11) can be efficiently solved using the revised primal simplex algorithm with an explicit R x R basis inverse matrix. Let D denote the basis matrix for the problem. A readily available initial basis matrix is given by the identity matrix corresponding to the slack variables. Using this initial basis, the basis inverse is also the R x R identity matrix. The vector of dual variables ω associated with this initial basis is simply the zero vector.

Problem (11) has an optimal solution of zero if

$$g_k - \omega G_k \geqq 0$$

for each column K. If $g_c - \omega G_c < 0$ for some column c, then D is not an optimal basis and column c can be chosen as the pivot column for a standard primal simplex pivot.

Given a pivot column c, if the vector $D^{-1}G_c$ is nonpositive, then problem (11) has an unbounded solution and therefore the adjacent basis under consideration is dominated. Otherwise, any row r corresponding to a positive coefficient of the vector $D^{-1}G_c$ can be chosen as the pivot row and the standard simplex pivot can be carried out.

Since the number of criteria is not generally very large, the R x R basis inverse should probably be maintained in explicit form for fastest implementation.

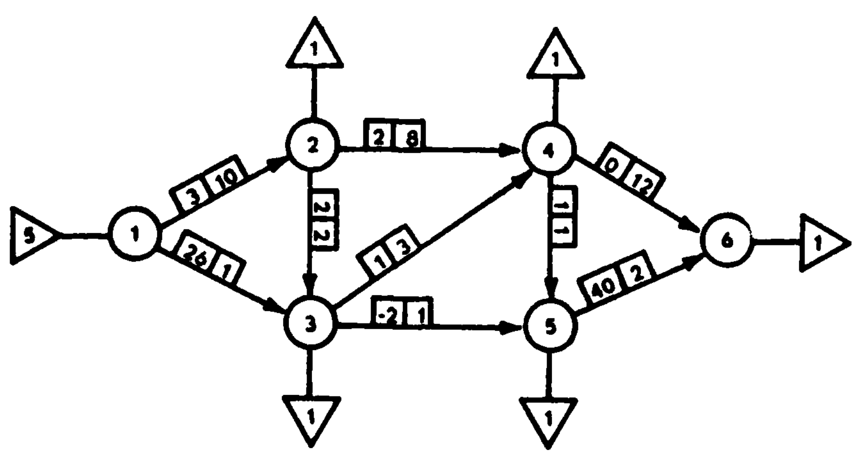
It should be noted that an easy check for nondominance can be made before actually resorting to solving problem (11) for the adjacent basis. The check is to examine the vector of reduced costs G_k for each nonbasic arc. Problem (11) has an unbounded solution if there exists a nonzero, nonpositive vector G_k.

2.4 EXAMPLE

A small six node and nine arc uncapacitated transshipment problem with two criteria functions is solved in this section. Figure 1 illustrates the network configuration of the problem.

FIGURE 1

EXAMPLE NETWORK



Node 1 has a supply of five units and each of the other nodes has a demand of one unit. These amounts are indicated in the triangles attached to each node. The two criteria coefficients are given in the boxes attached to each arc. The origin node (i_k), destination node (j_k), first criterion coefficient (C_k^1), and second criterion coefficient (C_k^2) for each of the nine arcs is given in Table 1. This problem is actually a bicriteria shortest path problem from node 1 to all other nodes.

The first step of the multicriteria simplex algorithm is to determine an initial nondominated basic feasible solution to the problem. This can be done by selecting any vector $\lambda > 0$ and solving the corresponding weighted single criterion uncapacitated transshipment problem (5). For sake of illustration, equal weights (i.e., $\lambda^1 = \lambda^2 = 1$) were chosen for this problem. The resulting initial nondominated basis tree is shown in Figure 2. The vectors of node potentials are indicated beside the corresponding nodes. The matrix of reduced costs $C - \Pi A$ is shown in Figure 3.

TABLE 1

ARC DATA

k	1	2	3	4	5	6	7	8	9
i_k	1	1	2	2	3	3	4	4	5
j_k	2	3	4	3	4	5	6	5	6
C_k^1	3	26	2	2	1	-2	0	1	40
C_k^2	10	1	8	2	3	1	12	1	2

FIGURE 2

FIRST NONDOMINATED BASIS TREE

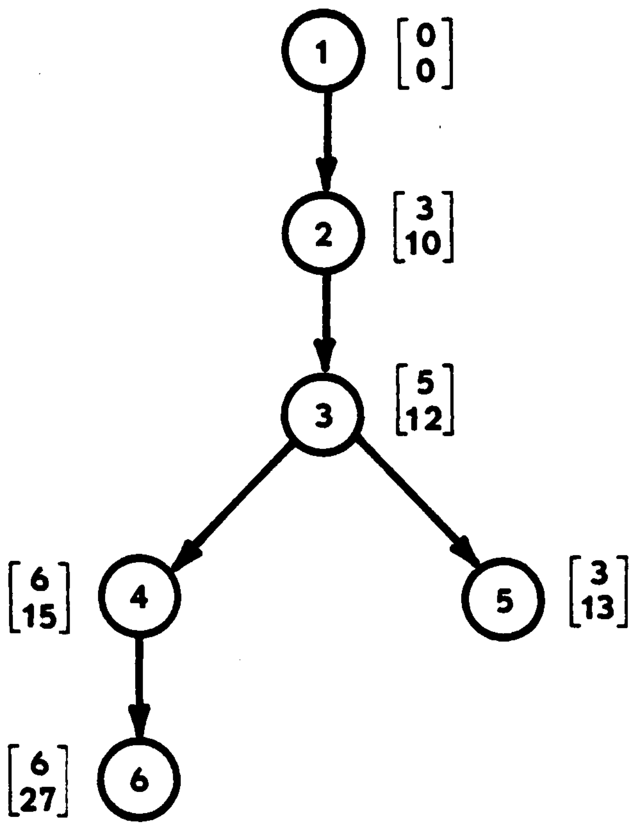


FIGURE 3

FIRST REDUCED COST MATRIX

$$\begin{bmatrix} 0 & 21 & -1 & 0 & 0 & 0 & 0 & 4 & 37 \\ 0 & -11 & 3 & 0 & 0 & 0 & 0 & 3 & -12 \end{bmatrix}$$

Three bases adjacent to the nondominated one shown in Figure 2 must be checked for nondominance. The adjacent basis associated with arc 8 does not have to be considered since its vector of reduced costs is nonnegative. The nondominance of the adjacent bases corresponding to arcs 2, 3, and 9 can either be checked by the nondominance subproblem or the effective constraint approach. However, since the problem only has two criteria, the nondominated region Λ can be plotted and the nondominance of the adjacent bases determined by inspection. Figure 4 shows the nondominated region associated with the basis tree given in Figure 2. The adjacent bases associated with arcs 2 and 3 are nondominated since their reduced costs form effective constraints of Λ.

Figure 5 illustrates the nondominated basis tree that is obtained if arc 5 is replaced by arc 3 in the initial nondominated basis tree. Again, the vectors of node potentials are indicated beside the corresponding nodes. Note that the node potential vectors for nodes 4 and 6 changed by a constant amount $\left(\Delta = \begin{bmatrix} -1 \\ 3 \end{bmatrix}\right)$ while the other node potential vectors remained the same. The reduced cost matrix is shown in Figure 6, and Figure 7 shows the nondominated region associated with this basis. There is only one effective constraint and it corresponds to the first nondominated basis that was found, so no further examination needs to be made in this direction.

The next step is to back up to the initial nondominated solution (Figure 2) and introduce arc 2 into the basis. This yields

FIGURE 4

FIRST NONDOMINATED REGION

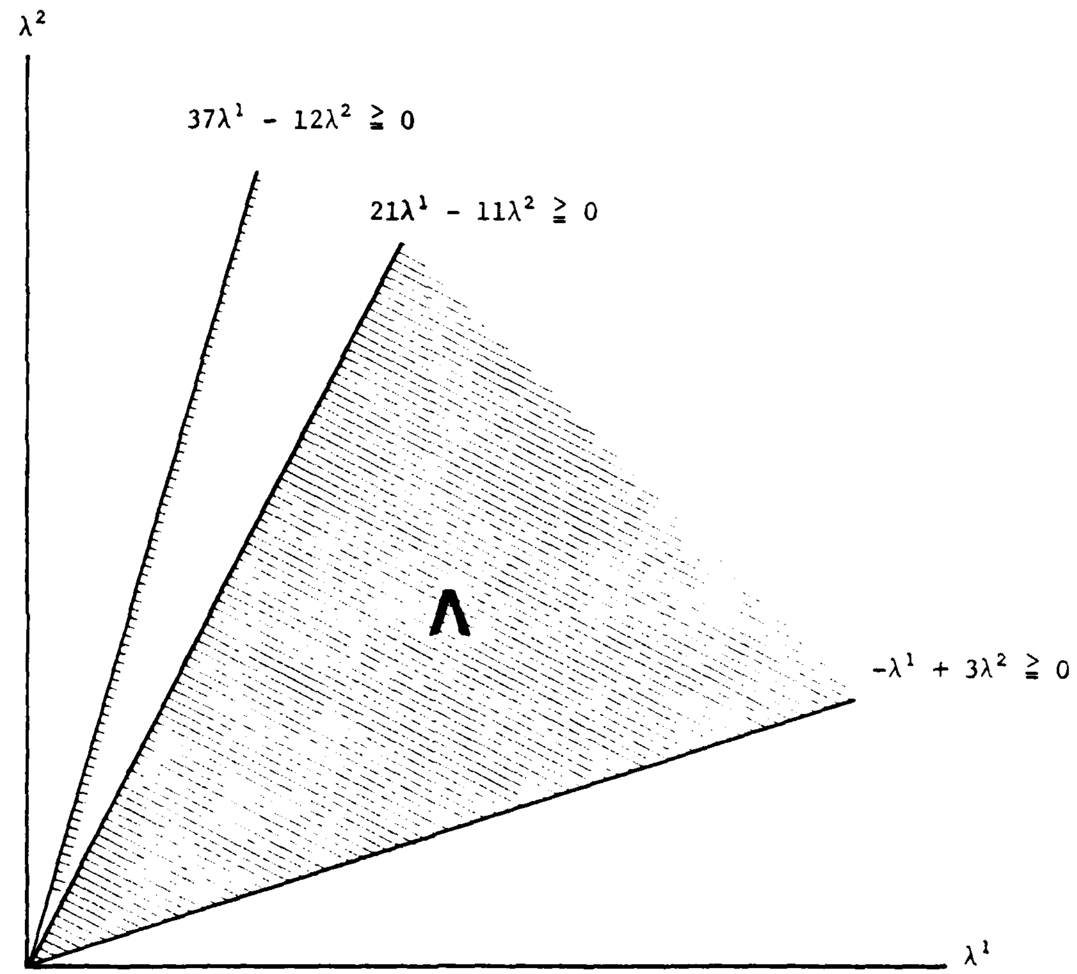


FIGURE 5

SECOND NONDOMINATED BASIS TREE

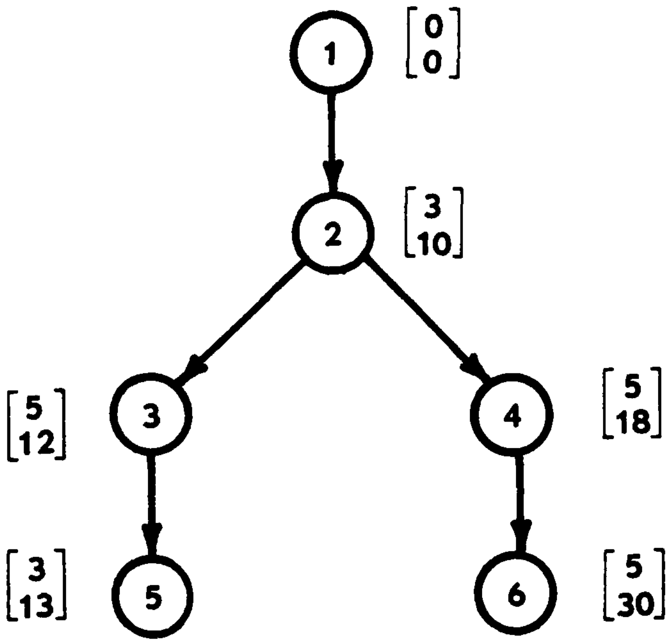


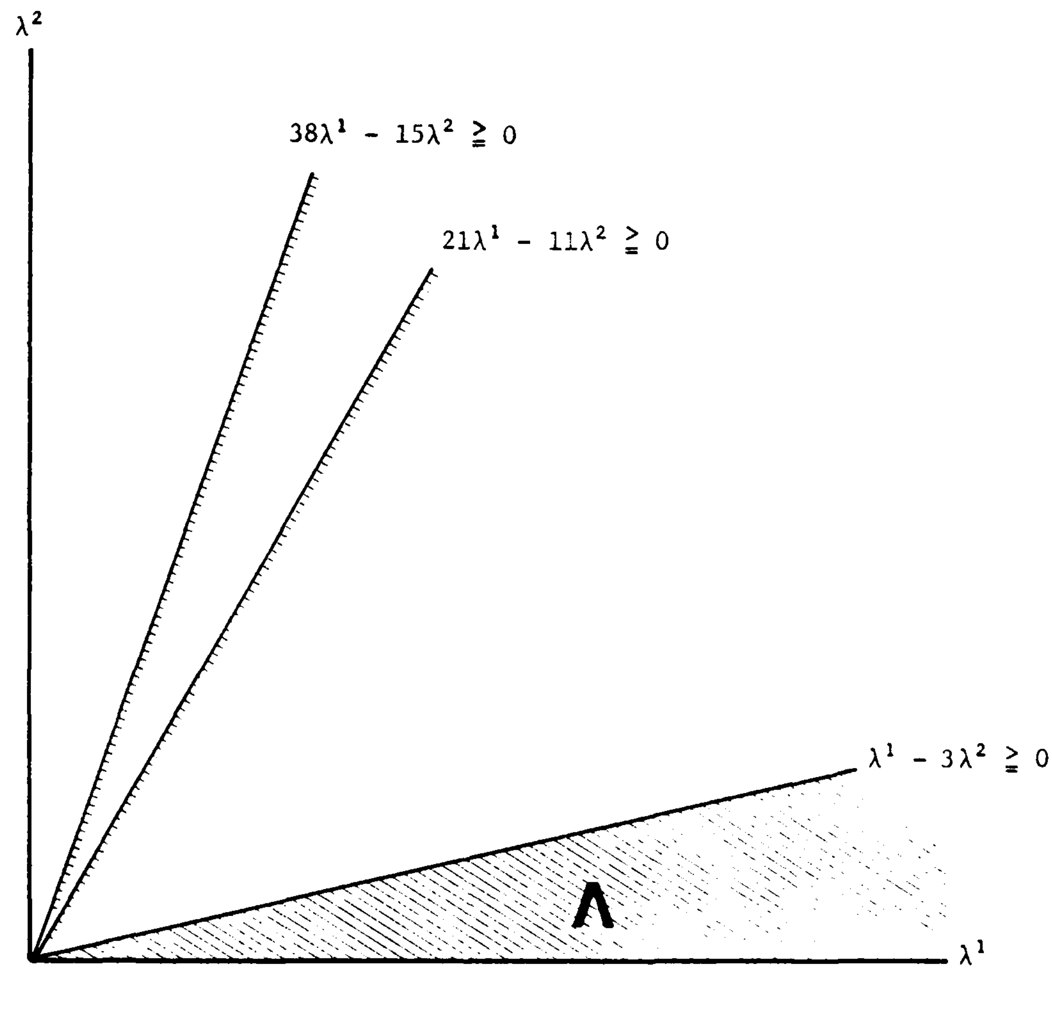
FIGURE 6

SECOND REDUCED COST MATRIX

$$\begin{bmatrix} 0 & 21 & 0 & 0 & 1 & 0 & 0 & 3 & 38 \\ 0 & -11 & 0 & 0 & -3 & 0 & 0 & 6 & -15 \end{bmatrix}$$

FIGURE 7

SECOND NONDOMINATED REGION



the third nondominated basis shown in Figure 8. The associated matrix of reduced costs is given in Figure 9 and its nondominated region is shown in Figure 10. There are two effective constraints for this basis but only the one corresponding to arc 9 yields a nondominated basis that has not already been encountered.

FIGURE 8

THIRD NONDOMINATED BASIS TREE

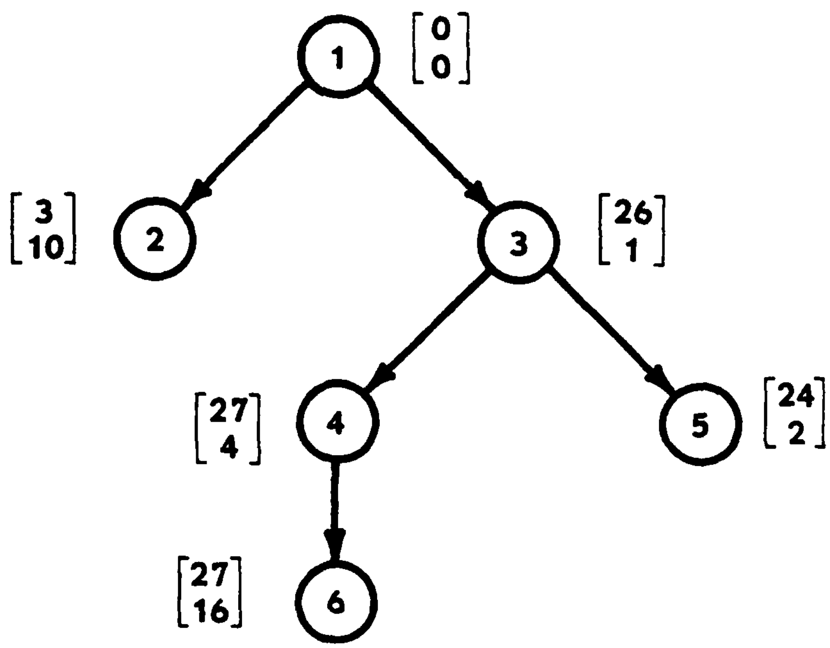


FIGURE 9

THIRD REDUCED COST MATRIX

$$\begin{bmatrix} 0 & 0 & -22 & -21 & 0 & 0 & 0 & 4 & 37 \\ 0 & 0 & 14 & 11 & 0 & 0 & 0 & 3 & -12 \end{bmatrix}$$

FIGURE 10

THIRD NONDOMINATED REGION

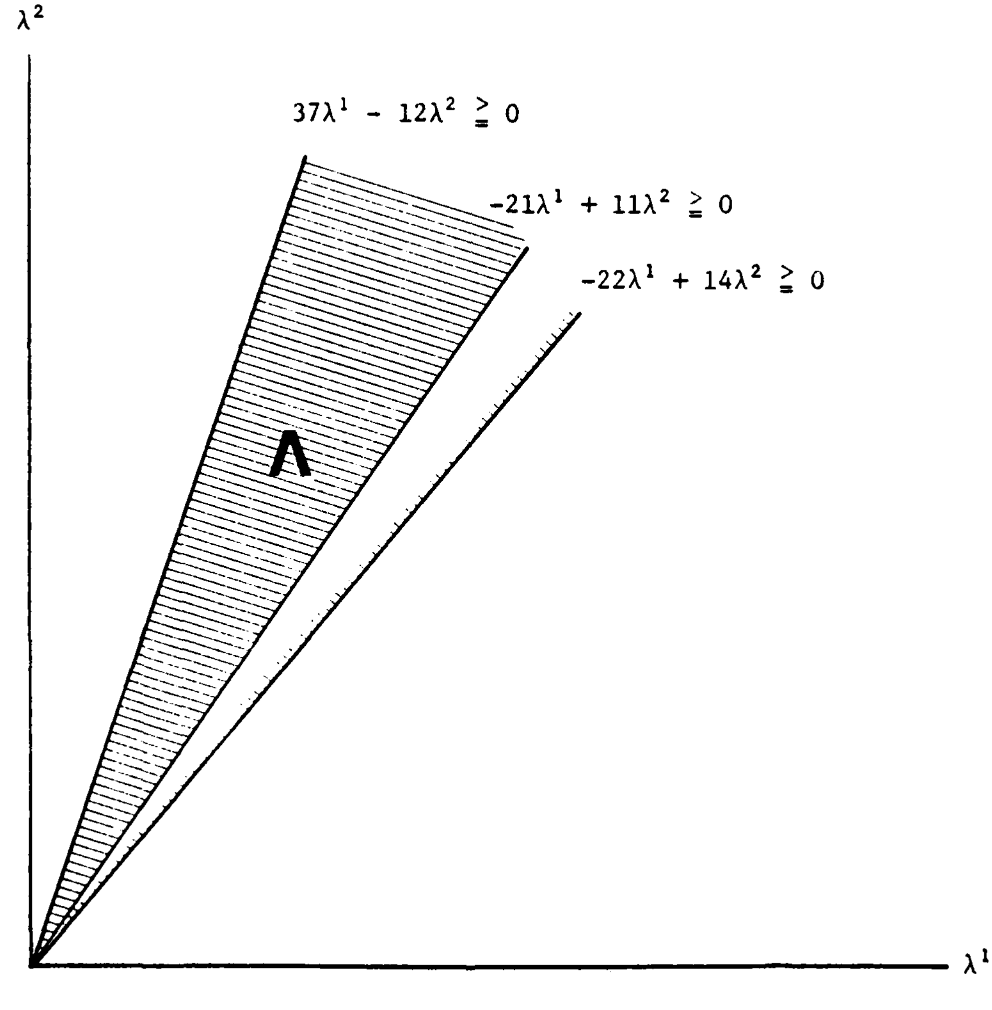


Figure 11 shows the nondominated basis tree obtained by replacing arc 7 by arc 9. The matrix of reduced costs is given in Figure 12 and the nondominated region is illustrated in Figure 13. The only effective constraint corresponds to an already known nondominated basis.

FIGURE 11

FOURTH NONDOMINATED BASIS TREE

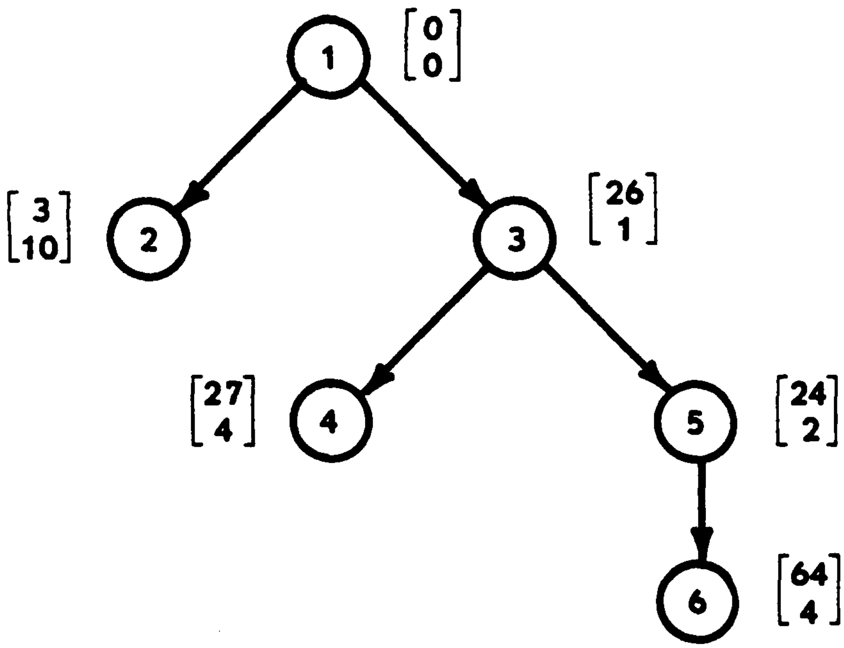


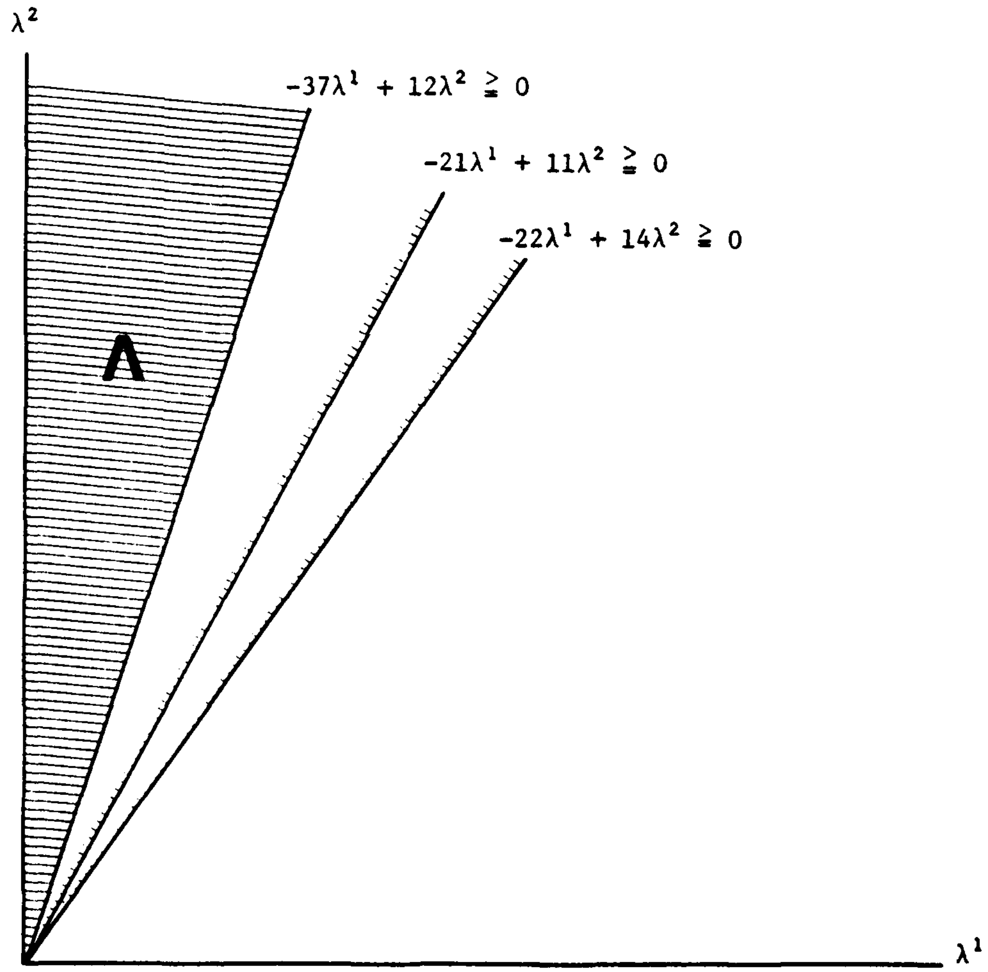
FIGURE 12

FOURTH REDUCED COST MATRIX

$$\begin{bmatrix} 0 & 0 & -22 & -21 & 0 & 0 & -37 & 4 & 0 \\ 0 & 0 & 14 & 11 & 0 & 0 & 12 & 3 & 0 \end{bmatrix}$$

FIGURE 13

FOURTH NONDOMINATED REGION



All four nondominated basic solutions to this small example problem have been determined. Any linear combination of adjacent nondominated basic solutions is also a nondominated solution. However, for this small example problem, the only integer nondominated solutions are the four basic solutions. This is true for any multiobjective shortest path or assignment problem, but it is not necessarily true for a more general transshipment problem.

3.1 SURROGATE CRITERION NETWORK METHOD

In this section, an interactive solution procedure is presented for solving shortest path problems with multiple criteria. The procedure is based on the surrogate criterion approach. Each iteration of the solution procedure involves both the decision maker and the computer. Based on a set of weights assigned by the decision maker for each criterion, the computer is used to solve a weighted objective (surrogate criterion) shortest path problem. Any efficient algorithm can be used for the optimization process. However, since label-correcting shortest path algorithms can be re-started from an earlier solution, it is believed that a simplex-based code such as C5 [2] is the best choice for the optimizer.

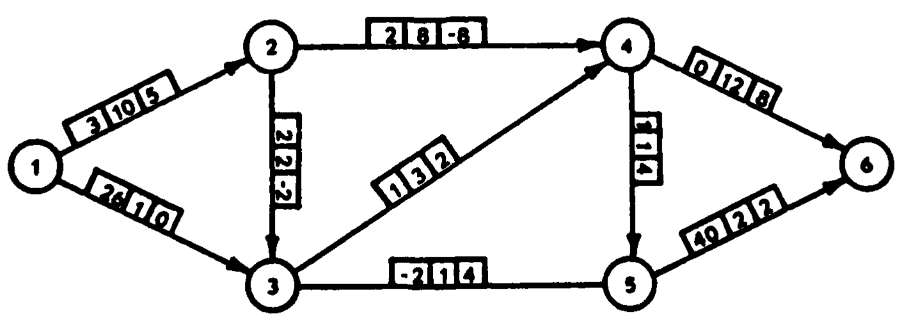
After solving the surrogate criterion shortest path problem, the decision maker is presented with the values of each criteria function for the current solution. Then, depending upon his satisfaction with these values, he either stops with a satisfactory solution, or modifies his criteria weights and returns control to the optimizer.

A number of interactive systems such as this have been developed for solving large-scale multicriteria problems [3, 8, 10, 12, 17]. The general belief is that the decision maker who actively participates in the solution process gains valuable insight into the actual problem being solved. Not only does this insight help him direct the search by the computer for the best compromise solution, but it also provides him with a better grasp of the interacting characteristics of the problem.

A small six node and nine arc network with three criteria functions is given in Figure 14. This example will be used to demonstrate the analysis of a multicriteria shortest path problem. Node 1 is the source node for this example. The length of each arc, in terms of the three different criteria functions, is indicated in the boxes attached to the arc. For instance, the arc from node 1 to node 2 has a length of 3 according to the first criterion, a length of 10 for the second, and a length of 5 for the third. These three criteria may be thought of as reflecting the cost, speed, and comfort level associated with using the arc.

FIGURE 14

EXAMPLE NETWORK



Figures 15, 16, and 17 illustrate the shortest path trees obtained by independently considering criterion one, two, and three, respectively. That is, if criterion one is assigned a weight of one and criteria two and three are assigned weights of zero, then the shortest path tree shown in Figure 15 is obtained. The vectors shown beside each node in these figures is the length of the path from the source to the node in terms of the three criteria. For instance, in Figure 15 the length of the path from the source to node 5 is 3 in terms of the first criterion, 13 for the second, and 7 for the third.

FIGURE 15

CRITERION 1 - SHORTEST PATH TREE

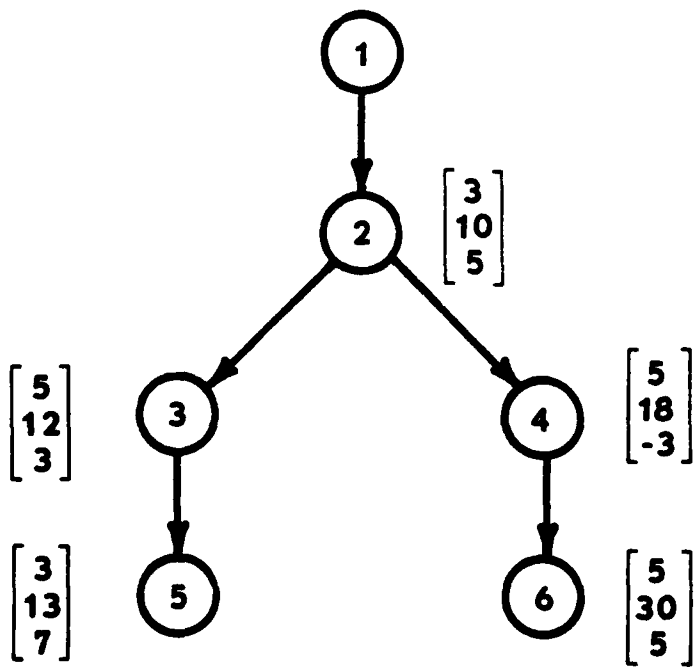


FIGURE 16

CRITERION 2 - SHORTEST PATH TREE

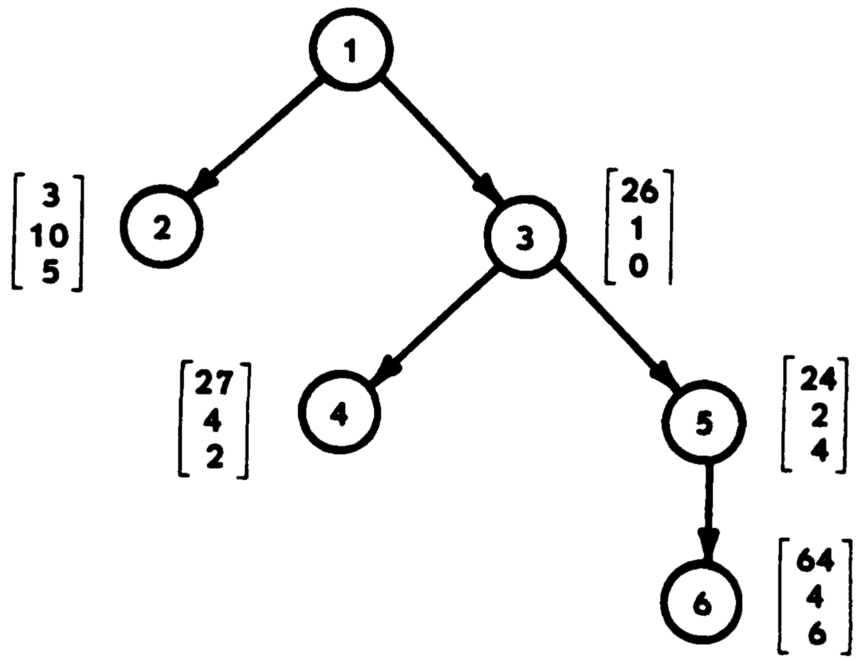
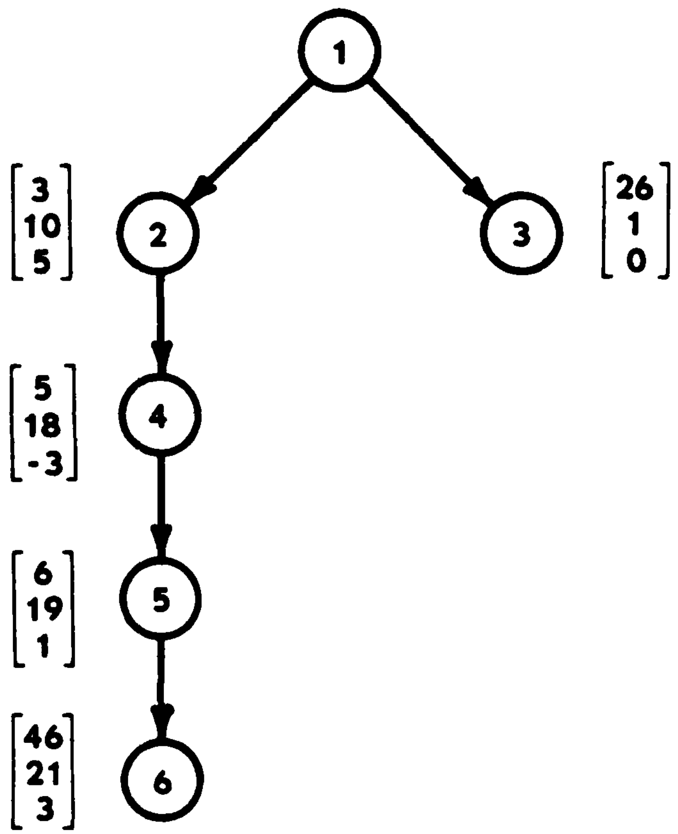


FIGURE 17

CRITERION 3 - SHORTEST PATH TREE



Figures 15 and 16 illustrate the underlying conflict between the first two criteria. That is, the best solution for criterion one (Figure 15) is very poor for criterion two, and similarly, the best solution for criterion two (Figure 16) is very poor for criterion one.

Multiple conflicting goals, as illustrated by this example, are quite common in practice. Their occurrance calls for a *compromise* solution. That is, a solution to the problem may not be best in terms of any one criterion, but should be acceptable in terms of all criteria. Such a compromise solution, for this small example, is illustrated in Figure 18. This solution was obtained by simultaneously considering all three criterion. Specifically, the first criterion was assigned a weight of 10, the second a weight of 10, and the third a weight of 1. Although the solution in Figure 18 is not the best in terms of any single criterion, it is a valid compromise solution to the problem.

3.2 MULTSP: AN INTERACTIVE SOLUTION PROGRAM

A simple in-core, interactive multicriteria shortest path code, MULTSP, was developed in order to investigate the solution procedure suggested in Section 3.1. This code uses a specialized version of the C5 label-correcting code [2] to solve the individual surrogate criterion shortest path problems. A re-start capability was added to the basic C5 code in order to capitalize on the fact that the p^{th} and $(p + 1)^{st}$ surrogate criterion shortest path problems are generally very similar.

FIGURE 18

COMPROMISE SHORTEST PATH TREE

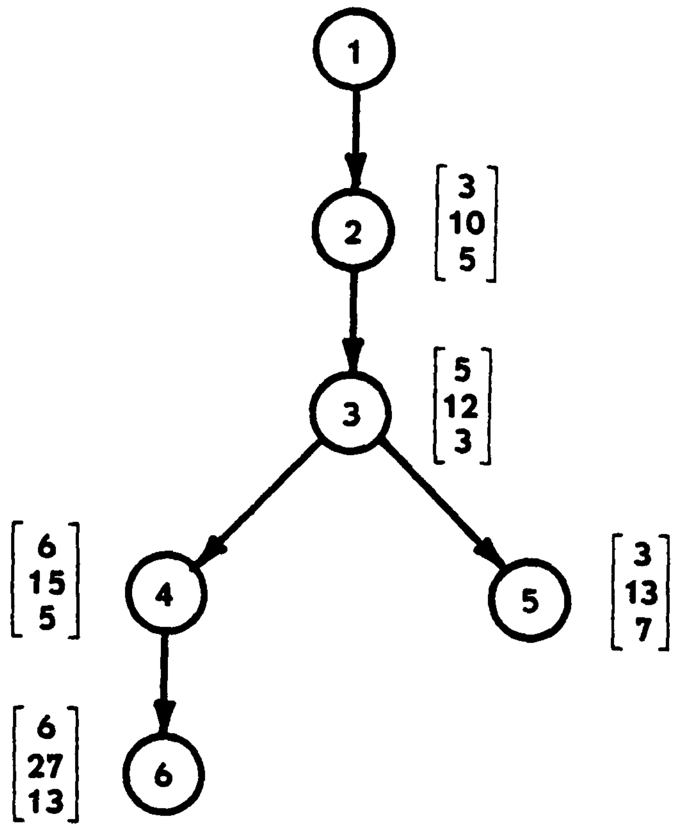


TABLE 2

RE-START IMPACT

PROBLEM NUMBER	CRITERIA WEIGHTS			ITERATIONS REQUIRED USING RE-START	ITERATIONS REQUIRED NOT USING RE-START
	1	2	3		
1	1	0	0	210	210
2	0	1	0	83	211
3	0	0	1	100	214
4	1	1	1	23	208
5	1	1	5	16	206
6	1	5	5	16	208
TOTAL NUMBER OF ITERATIONS				448	1257

Table 2 presents some test results using MULTSP on a 100 node, 1000 arc, 3 criteria shortest path network problem. This table indicates the computational advantage of using a label-correcting code with the re-start capability. Specifically, for this example, the re-start capability reduced the total number of iterations required to determine the three single criterion optimal solutions and three compromise solutions from 1257 to 448. As indicated in the table, the reduction in the number of iterations is most pronounced as "fine-tuning" of the compromise solution is carried out.

Table 3 provides the test results for a 500 node, 2500 arc, 3 criteria shortest path problem. Determining the first solution required 846 iterations. By using the re-start capability of MULTSP, each of the next thirteen solutions only required an average of 44 additional iterations. It is interesting to note that determining

TABLE 3

REQUIRED ITERATIONS

PROBLEM NUMBER	CRITERIA WEIGHTS			ITERATIONS REQUIRED
	1	2	3	
1	1	0	0	846
2	10	1	0	14
3	5	1	0	14
4	1	1	0	90
5	1	5	0	95
6	1	10	0	24
7	0	1	0	45
8	0	10	1	58
9	0	5	1	29
10	0	1	1	98
11	0	1	5	73
12	0	1	10	10
13	0	0	1	9
14	1	1	1	74
TOTAL NUMBER OF ITERATIONS				1479

all fourteen solutions required less than .18 c.p.u. seconds on The University of Texas' CDC 6600 (MNF compiler). Clearly, when a code can solve problems as large as this one so rapidly, the decision maker faced with a multicriteria planning problem has a valuable tool in his arsenal.

REFERENCES

1. R. Barr, F. Glover, and D. Klingman, "An Improved Version of the Out-of-Kilter Method and a Comparative Study of Computer Codes," *Mathematical Programming*, 7, 1 (1974) 60-87.

2. R. Dial, F. Glover, D. Karney, and D. Klingman, "A Computational Analysis of Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees," *Networks*, 9 (1979) 215-248.

3. A. Geoffrion, J. Dyer, and A. Feinberg, "An Interactive Approach for Multicriteria Optimization with an Application to the Operation of an Academic Department," *Management Science*, 19 (1972) 357-368.

4. F. Glover, D. Karney, and D. Klingman, "Implementation and Computational Comparisons of Primal, Dual, and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems," *Networks*, 4, 3 (1974) 191-212.

5. F. Glover, D. Klingman, J. Mote, and D. Whitman, "Comprehensive Computer Evaluation and Enhancement of Maximum Flow Algorithms," Research Report CCS 360, Center for Cybernetic Studies, The University of Texas at Austin, 1979.

6. F. Glover, D. Klingman, and A. Napier, "An Efficient Dual Approach to Network Problems," *OPSEARCH*, 9, 1 (1972) 1-19.

7. I. Heller and C. Tompkins, "An Extension of a Theorem of Dantzig's," in *Linear Inequalities and Related Systems*, H. W. Kuhn and A. W. Tucker (eds.), Princeton University Press, Princeton, New Jersey, 1956.

8. J. Hultz, D. Klingman, G. T. Ross, and R. Soland, "An Interactive Computer System for Multicriteria Facility Location," Research Report, Analysis, Research, and Computation, P.O. Box 4067, Austin, Texas 78765, 1979.

9. M. Hung and W. Rom, "Solving the Assignment Problem by Relaxation," Working Paper, College of Business Administration, Cleveland State University, 1978.

10. G. T. Ross and R. Soland, "A Multicriteria Approach to the Location of Public Facilities," Rapport Technique No. EP78-R-16, Ecole Polytechnique de Montreal, 1978.

11. L. Seiford and P. L. Yu, "Potential Solutions of Linear Systems: The Multi-Criteria Multiple Constraint Levels Program," *Journal of Mathematical Analysis and Applications*, 69, 2 (1979) 283-303.

12. R. Steuer and A. Schuler, "An Interactive Multiple-Objective Linear Programming Approach to a Problem in Forest Management," *Operations Research*, 26, 2 (1978) 254-269.

13. P. L. Yu, "Cone Convexity, Cone Extreme Points, and Nondominated Solutions in Decision Problems with Multiobjectives," *JOTA*, 14, 3 (1974) 319-377.

14. P. L. Yu and M. Zeleny, "Linear Multiparametric Programming by Multicriteria Simplex Methods," *Management Science*, 23, 2 (1976) 159-170.

15. P. L. Yu and M. Zeleny, "The Set of All Nondominated Solutions in Linear Cases and a Multicriteria Simplex Method," *Journal of Mathematical Analysis and Applications*, 49, 2 (1975) 430-468.

16. M. Zeleny, *Linear Multiobjective Programming*, Springer-Verlag, New York, 1974.

17. S. Zionts and J. Wallenius, "An Interactive Programming Method for Solving the Multiple Criteria Problem," *Management Science*, 22, 6 (1976) 652-663.