NBSIR 83-2745

Program Documentation for the Resource Recovery Planning Model

May 1983

Issued August 1983

Sponsored by
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Measurement Laboratory
Office of Recycled Materials
Washington, DC 20234

NBSIR 83-2745

PROGRAM DOCUMENTATION FOR THE RESOURCE RECOVERY PLANNING MODEL

Edward B. Berman

Edward B. Berman Associates, Inc.
Marblehead, MA 01945

and

Robert E. Chapman and Howard K. Hung

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Applied Mathematics
Washington, DC 20234

May 1983

Issued August 1983

Sponsored by
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Measurement Laboratory
Office of Recycled Materials
Washington, DC 20234

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

PREFACE

This research was conducted under the sponsorship of the Office of Recycled Materials, National Measurement Laboratory, National Bureau of Standards (NBS), by the Operations Research Division, Center for Applied Mathematics, National Engineering Laboratory, NBS. The Resource Recovery Planning (RRPLAN) model described in this report is an outgrowth of a cooperative project between the Office of Recycled Materials, the Operation Research Division, and Edward B. Berman Associates, Inc.

The RRPLAN model is a part of a larger program specified for the Office of Recycled Materials in the Resource Conservation and Recovery Act of 1976 (P.L. 94-580). This legislation directs the Secretary of Commerce, acting through NBS, to hold public hearings and publish guidelines for the development of specifications. These specifications are needed for the classification of materials which can be recovered from wastes now destined for disposal.

One of the goals of the Resource Conservation and Recovery Act, as it relates to this project, is to provide a framework for fostering the advances required to promote a more efficient mix of economic and environmental factors associated with the disposal and subsequent recovery and/or reuse of resources contained in municipal solid waste. The RRPLAN model addresses this goal by providing a methodology for the economic assessment of current and potential technologies, market potentials, and institutional barriers to resource conservation and recovery. The model is designed for use by both the public and private sectors in seeking to analyze the economic issues associated with alternative solid waste management programs. This report is designed as a technical reference document for setting up and maintaining the RRPLAN model.

The authors would like to express their gratitude to the many individuals whose cooperation helped them as model builders to better understand the needs of the intended users of the RRPLAN model. Appreciation is extended to Ms. Rosalie T. Matthews, Office of Recycled Materials, for her assistance, insight and many helpful comments throughout the course of this effort. Special appreciation is extended to Mssrs. Roy M. Allison and William G. Hall, Center for Applied Mathematics, for their extensive comments and editorial assistance which made the timely completion of this study possible. Special appreciation is also extended to Dr. Harvey Yakowitz, Office of Recycled Materials, whose stimulating discussions provided guidance and encouragement throughout this effort.

ABSTRACT

The Resource Recovery Planning (RRPLAN) model is designed with three purposes in mind. First and foremost, is the ability to generate a preferred (optimal) plan for resource recovery. Second, is the capability to evaluate a scenario specified by the decision maker for technical and economic feasibility. Third, is its use as a tool to facilitate the decision making process by providing answers to many what-if questions through an in-depth sensitivity analysis. In order to find the optimal solution, however, it is necessary to address three interdependent issues. The first two issues are concerned with the siting and sizing of solid waste management facilities, whereas the third concerns how to allocate commodities among the various facilities and potential markets. The existence of substantial economies of scale in the construction and operation of the types of facilities considered complicates the problem by introducing non-linearities into the objective function. RRPLAN uses a fixed-charge linear programming algorithm to deal with the two-parameter cost functions resulting from economies of scale. A heuristic post processor is also used to "force" the optimizer to systematically examine the solution domain in hopes of finding the true optimum. The model is written in FORTRAN and complies with the American National Standards Institute X3.9-1978 standard for that language.

Keywords: Economic analysis; facility location; fixed-charge problem; mathematical programming; optimization, resource recovery; solid waste management.

Table of Contents

List of Exhibits

List of Figures

List of Tables

Page

1. INTRODUCTION

Solid waste management is among the most complex municipal or regional governmental tasks facing policy makers today. The nonhomogeneous composition of the waste stream has resulted in a proliferation of approaches for handling the problem. Finding the best approach is complicated by noting that some communities have used mathematical models and met with little success whereas others have not and were able to achieve some reasonable solution. Over the last decade, many mathematical models have been developed and implemented with varying degrees of success. Although small communities would not need an elaborate model, techniques from engineering economics based on life cycle costing can still be quite useful in containing costs. Perhaps the greatest advantage offered by models however, is the way in which they help organize the thoughts of the decision maker so that important pieces of information are not overlooked. More recent modeling approaches have focused on this point; consequently, reliance on these models should produce useful results in dealing with the overall problem.

The economics of resource recovery is significantly more complicated than other alternatives however, requiring an in-depth analysis of facility design and cost as well as market size, structure and location. The Resource Recovery Planning (RRPLAN) model was especially designed to address the complicating factors mentioned above. On the one hand, RRPLAN explicitly incorporates potential economies of scale in the construction and operation of a solid waste processing facility. The model is thus able to support the basic tradeoff of savings from centralized processing versus the costs of additional haul required to bring it about. On the other hand, RRPLAN uses a detailed cost accounting system to attack the economic issues, carefully measuring the effects on overall program costs due to decisions affecting siting, routing, marketing and financing. By integrating the technical issues of processing with these four major deicision points, RRPLAN permits a wide variety of questions to be examined carefully.

1.1 PURPOSE OF MODEL

RRPLAN was developed in a cooperative program under the sponsorship of the National Bureau of Standards' Office of Recycled Materials by Edward B. Berman Associates, Inc., with the direct support of the NBS's Center for Applied Mathematics. RRPLAN has been tested extensively through pilot study applications to data from the State of Louisiana, central Mississippi, the California Solid Waste Management Board, and New York City.

The RRPLAN model described in this report may be thought of as a descendant of two earlier models. These models are known as WRAP (Waste Resource Allocation Program) and RAMP (Recovery And Market Planning). Both models were developed by the Mitre Corporation, the former through funding from the Environmental

Protection Agency.[1] There are substantial differences between the two models, especially regarding their software support systems and their treatment of market structure.

Both models use the same optimizer as RRPLAN. In WRAP, as in RRPLAN, a front end is available to build the equations for input into the optimizer, and a back end is available to interpret the solution. In RAMP, a more sophisticated equation structure is available, including the full market structure in RRPLAN, but the user must prepare equations for direct input into the optimizer, and must interpret its solution. The major focus of WRAP is on the identification of a preferred plan which includes the best candidate sites, the appropriate processing and disposal technology at each site, the sizing of each site and all transportation linkages among centers of waste generation, processing sites, and disposal sites. A major weakness of WRAP is its implicit assumption that any market for recoverables (e.g., ferrous products and newsprint) is unlimited. If market saturation is an important consideration, then WRAP's solution would represent an overly optimistic plan which could lead to serious cash flow problems if the plan were implemented. RAMP adds the saturation effect by incorporating both declining price and limited size markets.

As the description of the RRPLAN model unfolds, it will become evident that it is a synthesis of WRAP and RAMP. However, the model also incorporates all of the capabilities of WRAP and RAMP, and also incorporates numerous enhancements which render its cost accounting system far superior to those used in its predecessors. Furthermore, its more reasonable data requirements than WRAP coupled with the type and nature of its output should permit RRPLAN to greatly facilitate the regional planning and decision making processes. RRPLAN adds a sophisticated cost model, built-in source-separation options, an automatic dedicated transfer station function, user-defined cost, energy, and commodity categories, an extensive analysis of costs by source and site (including a projected full-cost tipping fee for each site), four new optimizing modes, and two new forcing modes. These will all be described in detail in this manual.

---

[1] Documentation for WRAP is included in the following reports: E.B. Berman, WRAP-A Model for Regional Solid Waste Planning: User's Guide, U.S. Environmental Protection Agency, SW-574, 1977; and V. Hensey, WRAP-A Model for Regional Solid Waste Planning: Programmer's Manual, U.S. Environmental Protection Agency, SW-573, 1977. All documentation for RAMP is included in the following report: E.B. Berman, Use of RAMP (Recovery And Market Planning) For The Evaluation of Policy Issues in Resource Recovery, mimeo, 1976.

1.2 RELATIONSHIP WITH THE FEDERAL INFORMATION PROCESSING STANDARDS

The NBS reports which describe the RRPLAN model are patterned after recommendations given in the Federal Information Processing Standards (FIPS) Publication 38[1] and NBS Special Publication 500-73[2]. The information contained in these documents permits one to define four general classes of publications. These four classes are associated with the programming and test stages of the software life cycle as defined in FIPS Publication 38. The four general classes of publications are:

(1) Management Summary Manual;
(2) User's Manual;
(3) Programmer's Manual; and
(4) Analyst's Manual.

The documentation for the RRPLAN model consists of two NBS reports. The first report, NBS Special Publication 657[3], includes the type of material described under the FIPS Management Summary and User's Manuals. The second report, NBSIR 83-2745, stresses the technical details described under the FIPS Programmer's and Analyst's Manuals.

The first manual is designed as a management tool. It provides the information necessary to assess the model's input requirements (including time, money, and other resources) and the usefulness of the model's results. The Management Summary Manual focuses on how the model can facilitate the decision making process rather than the specifics of how to set up and run the model. The RRPLAN equivalent of a Management Summary Manual is The Resource Recovery Planning Model: A New Tool for Solid Waste Management.

The second manual is designed as a reference document for a nonprogramming model user. Information contained in this manual is similar to the first but with increased emphasis on detail. In depth discussions of the following topics are included: the model's logical structure; the input data requirements; the results produced by the model; and the use of the model results. The RRPLAN equivalent of such a document is once again The Resource Recovery Planning Model: A New Tool for Solid Waste Management.

---

[1]National Bureau of Standards, Guidelines for Documentation of Computer Programs and Automated Data Systems, FIPS Publication 38, February 1976.

[2]National Bureau of Standards, Computer Model Documentation Guide, NBS Special Publication 500-73, January 1981.

[3]R. E. Chapman and E. B. Berman, The Resource Recovery Planning Model: A New Tool for Solid Waste Management, National Bureau of Standards, Special Publication 657, 1983.

The third and fourth documents are designed for use by programmers and analysts, respectively. The third manual provides guidelines for maintaining and modifying the model. These guidelines should be of sufficient detail to enable the programmer to understand the operation of the model and to trace through it for debugging, for making modifications, and for determining if and how the model can be converted to other computer systems. The fourth manual differs from the third in that its emphasis is on the model's functional structure, the algorithms used, and the techniques employed for model verification and validation. This report, NBSIR 83-2745, is the RRPLAN equivalent of the Programmer's and Analyst's Manuals described above.

The purpose of this report is to discuss the more technical aspects of the RRPLAN program documentation. A mathematical discussion of the computer program serves to introduce the basic philosophy behind the algorithm. Each routine is described, focusing on such topics as: (1) purpose; (2) calling sequence; (3) common blocks used; and (4) reports produced. A functional description of the model is used to facilitate the task of setting up RRPLAN on a user's computer system. A series of tests are outlined which should permit programmers to verify if the model produces correct solutions. The program is written in FORTRAN and complies with the guidelines set down in the ANSI X3.9-1978 software standard.[1]

---

[1]American National Standards Institute, American National Standard Programming Language FORTRAN, ANSI X3.9-1978, New York, 1978.

2. FUNCTIONAL DESCRIPTION OF RRPLAN

RRPLAN is a computer model designed with three purposes in mind. First and foremost, is the ability to generate a preferred plan for resource recovery. Second is the capability to evaluate a scenario specified by the decision maker for technical and economic feasibility. Third is its use as a tool to facilitate the decision making process by providing answers to many what-if questions through an in-depth sensitivity analysis.

2.1 PROBLEM STRUCTURE

The basic structure of RRPLAN consists of a set of equations and activities relating sources of solid wastes, sites where the wastes can be processed and markets for energy or materials recovery. The basic methodology employed by RRPLAN is illustrated in two diagrams. The first shows the major physical system details, whereas the second is a flowchart showing the model's logic and data flow. Turning to figure 2.1, it can be seen that the physical flows parallel the concepts just mentioned. Figure 2.1 is, however, merely a capsule summary of the overall problem because it focuses on the management of a single sources waste. In reality there are numerous sources, causing the array of linkages to become intertwined.

The first basic component of the problem structure focuses on the sources of solid waste (e.g., a community in a region or a district in a metropolitan area). Each source has associated with it a location, an estimated waste generation figure, a series of options for preseparating paper, glass and cans, and a set of transportation linkages to sites where the waste is processed.[1] Sites, as used in RRPLAN, may contain a landfill, a transfer station, an incinerator without heat recovery, or a resource recovery facility. Each site has associated with it a location, the type of processes it can accommodate (referred to as site-process combinations), the operating and capital costs requird to process the wastes, capacity considerations, outputs of wastes and marketable items, and transportation linkages. The model can handle markets with downward sloping demand curves, constraints on capacity, as well as ones which can receive unlimited quantities of recoverables. Each market has associated with it a location, a revenue schedule, and transportation linkages. The equations require all wastes to be transported to a site for processing, all wastes arriving at a site to be processed, all outputs of a site to be processed or sold, and that no capacities are exceeded. The activities in the model represent physical flows (e.g., transportation). processing (e.g., the amount of waste incinerated at a particular site), marketing (e.g., recoverables sold) as well as variables designed to preserve the structure of the problem.

[1]Each source has a variety of source separation options associated with it. They are: (1) no source separation (unconditional MSW); (2) always practice source separation (unconditional source separation); and (3) always practice source separation with a special proviso on the handling of paper (conditional source separation).

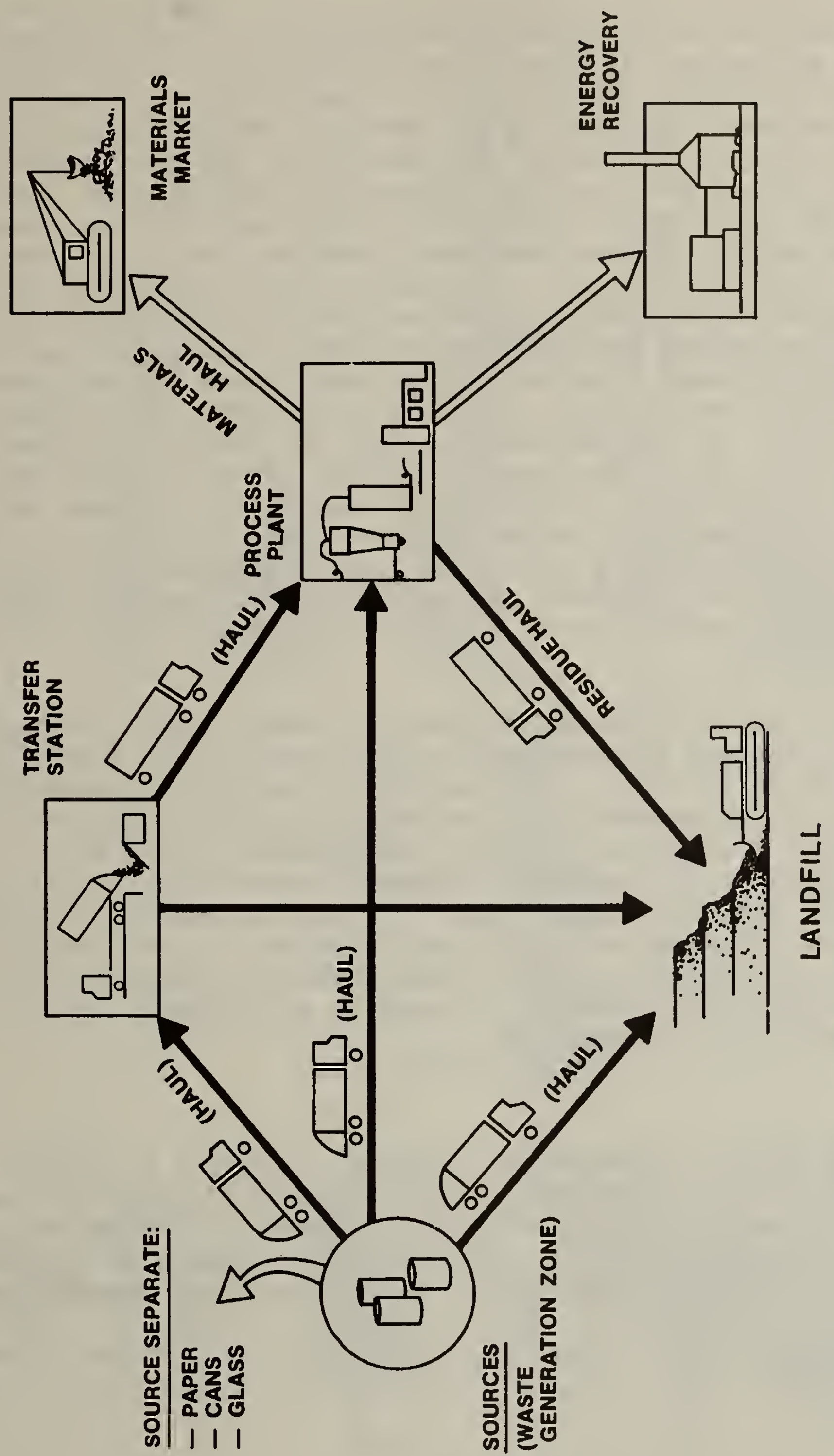
The model approaches the difficult problem of siting and sizing solid waste processing facilities by first approximating non linearities in the capital and operating cost functions with up to three linear segments. Each segment has an intercept (a fixed charge) and a slope (an incremental cost associated with increased processing activities). The introduction of fixed charges imposes certain complications causing the solution domain to be lumpy. This requires a specially designed optimization technique to generate meaningful solutions. The technique used in RRPLAN involves a fixed-charge linear programming algorithm with a forcing procedure to insure that the model can pass over an area of temporarily increasing cost in the solution domain (i.e., a lump) to find the true optimum.[1] RRPLAN adds new methods of forcing representing a significant improvement over methods used in other models in which each site (or site-process combination) which was in the solution is forced out of the solution and vice versa. This approach permits the solution domain to be searched in a more coordinated way by operating on all activities (e.g., transportation, processing, marketing) associated with a particular site (or site-process combination).

Figure 2.2 is a gross-level flowchart of the RRPLAN computer program in four pages of diagrams. The first and fourth pages deal with the MAIN program, whereas the second and third pages portray the logic and data flow within the optimizer. It is important to note that the optimizer is the same one used in WARP and RAMP. A number of enhancements have been incorporated, however, which lead to a more efficient use of the optimizer. These enhancements will be described as a part of the discussion of the optimizer. Some of the discussion which follows is of a more mathematical nature than in other parts of the report. Therefore, in order to promote a better understanding of the techniques discussed, terms which are of a mathematical nature will be defined.

MAIN PROGRAM (PART 1)

Referring now to the first page of the flowchart, we see a circular label with RRPLAN written inside. This point represents the beginning of the MAIN program which coordinates the flow of calculations among all subroutines. As a first step the values of all major variables are initialized prior to the reading of any user input data. The input is divided into three files. The first file is referred to as the case study file; it is represented as a deck of cards on the flowchart and is labled "Read User Input and Check for Validity." The second and third files are for packer and van tansportation distances. Both files are represented as a single deck of cards on the flowchart. Data from the case study file are read in a special sequence. These cards include information which sets control values, provides data on cost and energy categories, the types of waste and recoverable commodities considered, data on sources, sites, landfills, processing options and markets, as well as directions for extracting data from the packer and van transportation linkage files. After each card is read, the MAIN program checks for key values to determine if they are within tolerance and in the

[1] W.E. Walker, "A Heuristic Adjacent Extreme Point Algorithm for the Fixed Charge Problem," Management Science, Vol. 22 (1976), pp. 587-596.


MATERIALS MARKET
ENERGY RECOVERY
MATERIALS HAUL
PROCESS PLANT
TRANSFER STATION
(HAUL)
RESIDUE HAUL
(HAUL)
(HAUL)
(HAUL)
LANDFILL
SOURCE SEPARATE:
— PAPER
— CANS
— GLASS
SOURCES
(WASTE GENERATION ZONE)

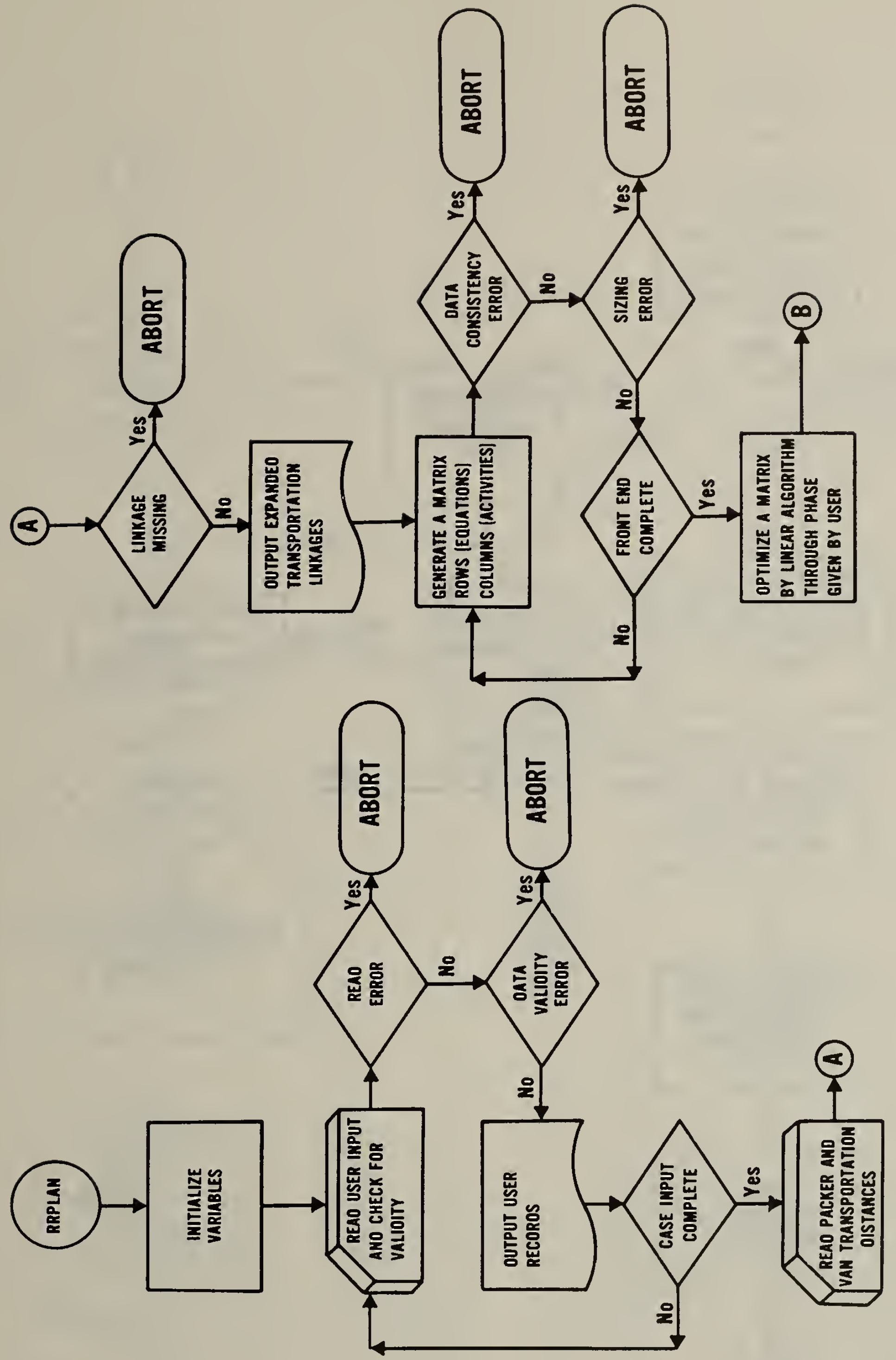
proper sequence. If one of these errors (i.e., tolerance values or proper sequence) is encountered, a message will be written out to the user which should facilitate the process of finding and correcting it. The program will then stop.[1] If no problem was encountered, the input data is formatted and printed back for the user. In this way the user can verify inputs as well as distinguish among a series of runs for the same case. The model reads each major type of data from a series of loops within the MAIN program. The ranges on the loops are specified by the information on the control card. Once all data from the case study file have been read, screened, and output, the packer and van distance files will be read. The logic in this portion of the MAIN program matches up the directions from the case study file with the distances[2] in the packer and van files. If a distance is missing (e.g., the case study file indicated that shipments from source I to site J were to take place but no distance from I to J was recorded) then the program writes a message to that effect and stops. The message should enable the user to find the location causing the problem. The specific problem can then be isolated by reviewing the data contained in the files. Transportation linkages are entered into the case study file and are specified as: (1) source to site; (2) site to site; and (3) site to market. After the files are read, the source to site category is expanded. The degree of its expansion depands on the source separation options offered and the type of paper market prevailing in the region.

All constraints and activities associated with the application problem are stored in an array referred to hereafter as the A matrix. The A matrix is constructed from the input data. Each row of the A matrix corresponds to an equation; each column corresponds to an activity. The rows and columns of the A matrix are defined in tables 2.1 and 2.2, respectively. If a data inconsistency or sizing error results, a message is written to the user to facilitate debugging and the program stops. If no error results, the MAIN program calls the optimizer to solve the problem stored in the A matrix. The optimizer (or linear programming module) which begins at statement label B on the second page of figure 2.2 consists of numerous subroutines. Each subroutine is summarized in appendix B.

---

[1]The edit-checking feature of RRPLAN was designed to prevent ill-formed problems from reaching the optimizer. Once the source of the error has been corrected, the user must resubmit the run for analysis. Although only one error per run is allowed by RRPLAN, the diagnostics provided by the model should permit the user to find all major input errors after several passes.

[2]The distances are measured in terms of miles, minutes or dollars per ton, as selected by the user.


RRPLAN
INITIALIZE VARIABLES
READ USER INPUT AND CHECK FOR VALIDITY
READ ERROR
Yes
ABORT
No
DATA VALIDITY ERROR
Yes
ABORT
No
OUTPUT USER RECORDS
CASE INPUT COMPLETE
No
Yes
READ PACKER AND VAN TRANSPORTATION DISTANCES
A
A
LINKAGE MISSING
Yes
ABORT
No
OUTPUT EXPANDED TRANSPORTATION LINKAGES
GENERATE A MATRIX ROWS (EQUATIONS) COLUMNS (ACTIVITIES)
DATA CONSISTENCY ERROR
Yes
ABORT
No
SIZING ERROR
Yes
ABORT
No
FRONT END COMPLETE
No
Yes
OPTIMIZE A MATRIX BY LINEAR ALGORITHM THROUGH PHASE GIVEN BY USER
B

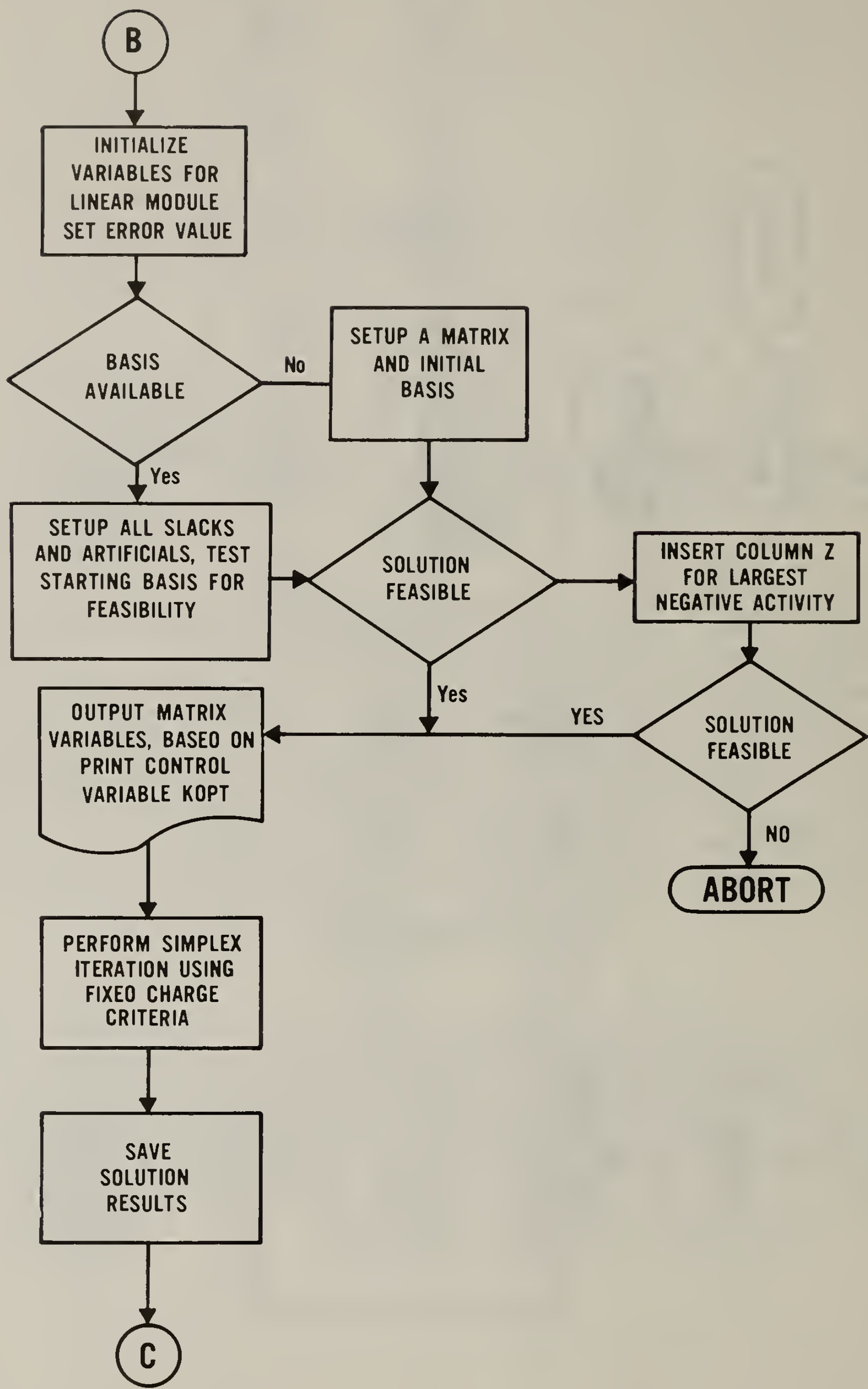

B
INITIALIZE VARIABLES FOR LINEAR MODULE SET ERROR VALUE
BASIS AVAILABLE
No
SETUP A MATRIX AND INITIAL BASIS
Yes
SETUP ALL SLACKS AND ARTIFICIALS, TEST STARTING BASIS FOR FEASIBILITY
SOLUTION FEASIBLE
INSERT COLUMN Z FOR LARGEST NEGATIVE ACTIVITY
Yes
YES
SOLUTION FEASIBLE
OUTPUT MATRIX VARIABLES, BASEO ON PRINT CONTROL VARIABLE KOPT
NO
ABORT
PERFORM SIMPLEX ITERATION USING FIXEO CHARGE CRITERIA
SAVE SOLUTION RESULTS
C

Figure 2.2 Flowchart of the RRPLAN Computer Program (continued)

OPTIMIZER (PART 2)

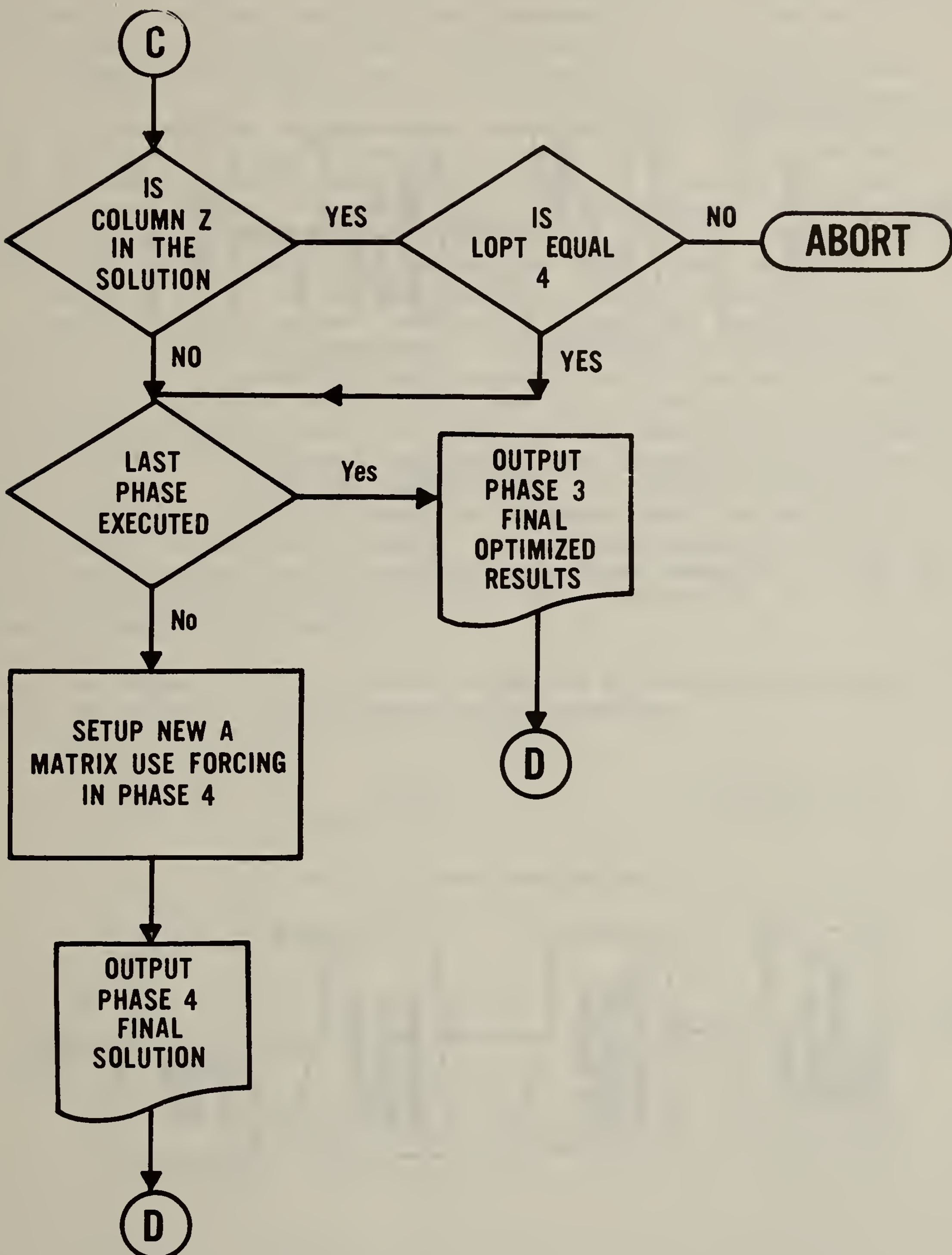
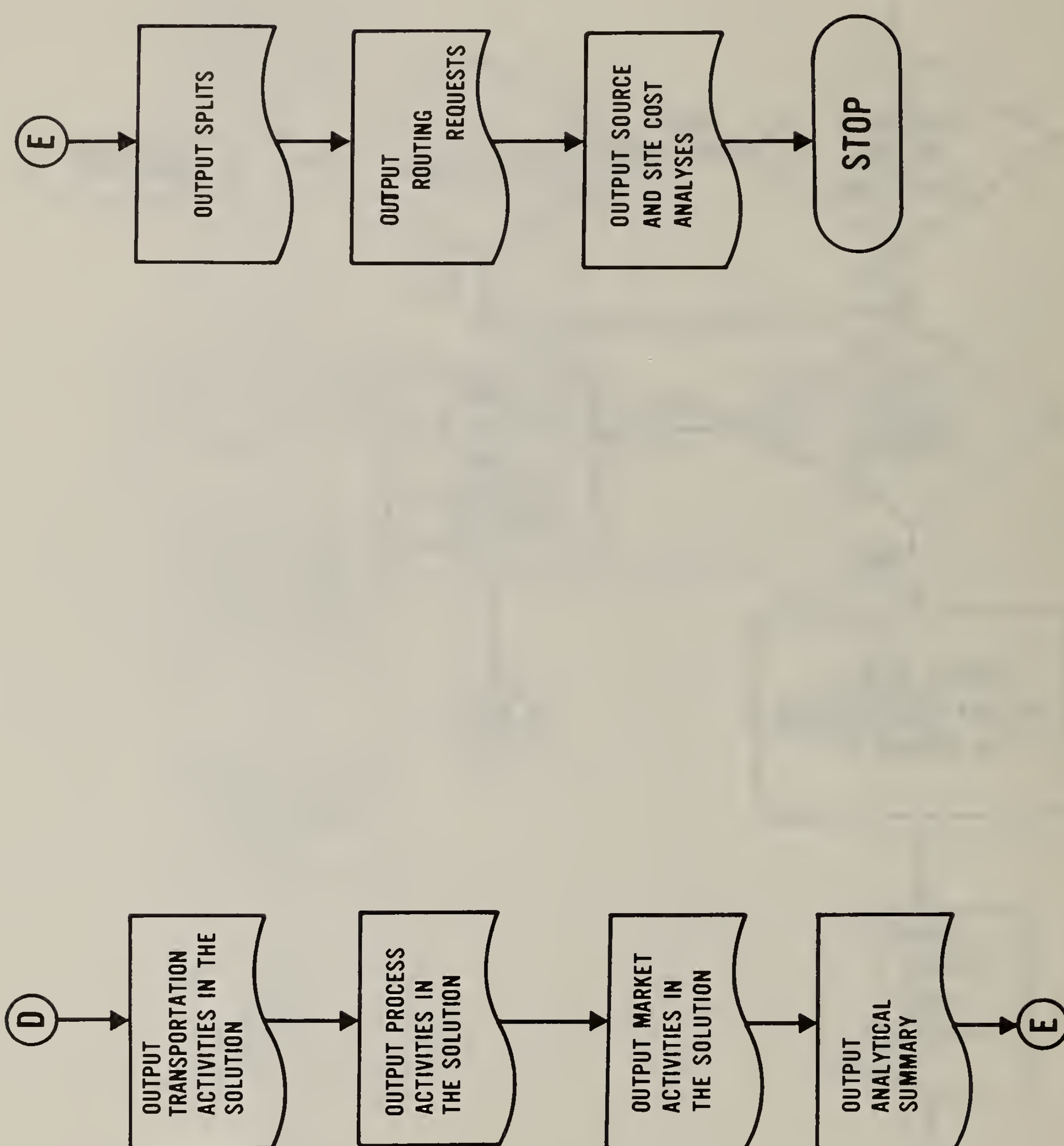


Figure 2.2 Flowchart of the RRPLAN Computer Program (continued)

MAIN PROGRAM (PART 2)



OPTIMIZER (PARTS 1 AND 2)

The model has five basic modes of optimization. Each mode serves to define the type of objective which is to be minimized or maximized. The first two modes of optimization minimize the total cost of the regional plan over a specified planning period. If both the timing and magnitude of cash flows are important, then the discounted costs of the regional plan should be minimized. If only the magnitude of cash flows is important then undiscounted costs should be minimized. It is important to note that the choice of discounted rather than undiscounted costs and vice versa may cause both the costs of the plan and the physical flows within the system to differ. The first mode, minimize lifetime discounted cost, is the preferred criterion for plan selection and evaluation. The third mode of operation seeks to maximize the net energy (energy produced (saved) from (due to) resource recovery activities minus all other energy inputs) of the regional plan. Such an objective function might be useful in comparing various waste-to-energy programs. The fourth mode of operation seeks to minimize a linear form (weighted sum) of program cost and net energy categories. This approach might prove useful in comparing mixtures of traditional and waste-to-energy programs if some form of matching formula for funds was in effect and will permit the user to weight cost or energy categories other than equally. The final mode of operation focuses on the topic of scenario evaluation. For example, a region may have a proposed plan which needs to be evaluated from the viewpoint of technical and economic feasibility. Typical questions addressed under this mode of operation would include the following. Are all facilities able to process the indicated waste stream without exceeding their rated capacity, or that of plants down the line which they feed into? Can resources be reallocated so that overall costs are reduced?

Assuming cost minimization is the sole objective of the decision maker, the problem to be solved by the optimizer may be stated as

$$\min \quad Z = \sum_{j=1}^{n} f_j(x_j) \qquad \text{equation 2.1}$$

subject to the linear constraints

$$\underset{\sim}{A}\underset{\sim}{X} = \underset{\sim}{b} \qquad \text{(i)}$$

$$\underset{\sim}{X} \geq \underset{\sim}{0} \qquad \text{(ii)}$$

where

$$f_j(x_j) = c_j x_j + k_j \delta_j \qquad \text{(iii)}$$

with

$$\delta_j = 0 \qquad \text{if } x_j = 0$$

$$= 1 \qquad \text{if } x_j > 0.$$

TABLE 2.1 Equations Used by RRPLAN

Equation Type	Purpose
Source balance equations	Requires all waste generated at each source to be entered into transportation.
Source paper balance equations	Requires all source-separated paper to be entered into transportation.
Site input residue balance equations	Generates a residue activity at a site equal to the residue arriving at the site.
Site input paper balance equations	Generates a paper activity at a site equal to the paper arriving at the site.
Site input balance equations for MSW and other commodities	Requires the total amount of MSW and other commodities arriving at a site to equal the amount of processing activity at the site.
Site output balance equations	Generates transportation activities of a commodity from a site equal to the process output of the commodity at that site.
Market input balance equations	Generates a total market activity equal to the amount of the commodity arriving at the market.
Site capacity constraints	Insures that processing at a site does not exceed capacity.
Land capacity constraints	Insures that land use does not exceed land available at any landfill.
Market bounds	Requires the sum of activity at a market in segments less than or equal to the jth $(1<j<5)$ segment to be less than or equal to the jth cumulative bound for the market.
Constraint on artificial processing activities	Provides an upper bound for artificial activities.

TABLE 2.2 Activities Considered by RRPLAN

Activity	Purpose
Transportation category 2	Site-to-site shipments.
Transportation category 3	Site-to-market shipments.
Transportation category 4	Source-to-site shipments of unseparated MSW.
Transportation category 5	Source-to-site shipments of unconditional source separated MSW (national paper market).
Transportation category 6	Source-to-site shipments of unconditional source separated MSW (local paper market).
Transportation category 8	Source-to-site shipments of pre-separated paper (local paper market).
Transportation category 9	Source-to-market shipments of pre-separated paper (local paper market).
Transportation categories 11-15	Source-to-site shipments of conditional source separated MSW (national paper market) with 1 to 5 trigger prices.
Residue activities	Measures thousands of tons per year of source-separated residue arriving at a site.
Paper activities	Measures thousands of tons per year of source-separated paper arriving at a site.
Process activities	Measures number of commodity units processed per year per linear segment at a site.
Market activities	Measures number of commodity units purchased per year per linear segment at a market.
Artificial processing activities	Take on epsilon values to preserve structure of the matrix.
NZ	Relieves infeasibilities associated with an advanced starting point.
NZX	Relieves infeasibilities when evaluating a prespecified plan.

The vector $\underset{\sim}{X}$ contains n elements known as activities (e.g., shipments of MSW from a source to a site). The tilde is used to distinguish the vector, $\underset{\sim}{X}$, from the elements x_j. The matrix A contains m rows and n columns. (It is assumed that n is greater than m.) The rows refer to the equations defined in table 2.1 and the columns to the activities defined in table 2.2. The c_j in condition (iii) are variable costs and the k_j are the fixed charges. (The k_j values are required to be greater than or equal to zero. If the activity has no fixed charges then k_j is equal to zero.)

At this point it would be appropriate to provide several definitions. An n-dimensional Euclidean space, E_n, is a set of objects called vectors which satisfy a series of requirements for multiplication by scalars, vector addition and an inner product measure with the added property that there exist n linearly independent[1] vectors and every set of n+1 vectors is linearly dependent. A basis is a set of n linearly independent vectors. A feasible solution to the problem is a vector $\underset{\sim}{X}^* = (x_1^*, \ldots, x_n^*)$ which satisfies conditions (i) and (ii). The feasible region is referred to as a polyhedron since its faces are polygons. A basic feasible solution to (i) is a solution obtained by setting n-m of the x_j to zero, solving for the remaining m variables and requiring that (ii) be satisfied. An extreme point is a feasible solution which can not be expressed as a convex combination[2] of any

---

[1]Independence implies that if $\underset{\sim}{P}_j \neq \underset{\sim}{0}$, j = 1, ..., n, then

$$\sum_{j=1}^{n} \delta_j \underset{\sim}{P}_j = \underset{\sim}{0}$$

obtains if and only if $\delta_j = 0$ for j = 1, ..., n. If at least one of the δ_j is non zero then the set of vectors is said to be linearly dependent.

[2]A convex combination of the points $\underset{\sim}{P}_1, \ldots, \underset{\sim}{P}_n$ is a point

$$\underset{\sim}{P} = \sum_{j=1}^{n} \delta_j \underset{\sim}{P}_j$$

where the δ_j are scalars, $\delta_j \geq 0$, and $\sum_{j=1}^{n} \delta_j = 1$.

two other distinct feasible solutions. From an important theorem in the theory of mathematical optimization, it can be asserted that basic solutions to the system AX = b completely identify all its extreme points.[1] Finally, if

all fixed charges, k_j, are non negative, the objective function is concave and it can be proven that a concave function defined over a convex polyhedron takes on its minimum at an extreme point.[2] The algorithm for solving the problem thus need only move from one vertex (extreme point) to another along the edges of the polyhedron, each time being sure to reduce the value of the objective function.

The logic which immediately follows label B (on the flowchart) attempts to alter the starting basis. An alteration is made if the starting basis is found to be infeasible due to a negative activity level. If this occurs, a special column (activity NZ in table 2.2) is substituted for the column (activity) with the <u>largest</u> negative activity level. One more attempt is made to have the basis accepted. If the second attempt fails, the program stops. This substitution approach is used only during the initialization phase.

---

[1]Since a basic solution is identified by a basis, m linearly independent vectors $\underset{\sim}{P}_j$ (out of a total of n), all the extreme points of

$$\underset{\sim}{AX} = \sum_{j=1}^{n} \underset{\sim}{P}_j x_j = \underset{\sim}{b}$$

are determined by computing n!/m!(n-m)! possible basic solutions of the system.

[2]A function f is concave if

$$\delta_1 f(x_1) + \delta_2 f(x_2) \leq f(\delta_1 x_1 + \delta_2 x_2)$$

for all

$$\delta_1 \geq 0, \delta_2 \geq 0, \quad \delta_1 + \delta_2 = 1 \text{ and all } x_1 \text{ and } x_2$$

in the convex domain of definition of f. A geometrical representation of a concave function, is one whose graph is never below the chords joining any two points on the graph. A proof of the footnoted statement may be found in W.M. Hirsch and G.B. Dantzig, "The Fixed Charge Problem," <u>Naval Research Logistics Quarterly</u>, Vol. 15 (1968), pp. 413-424.

The normal processing logic of the optimizer is determined by the user supplied input variable, LPHASE, which indicates the last phase of the execution desired. There are four possible phases of execution.

Phase 1: linear programming phase 1, generating a feasible solution (i.e., no negative activity levels).

Phase 2: linear programming phase 2, generating an optimal solution without consideration of fixed charges.

Phase 3: Walker algorithm[1] without forcing, generating a local optimum from adjacent extreme points which can improve the solution; fixed charges are considered.

Phase 4: Walker algorithm with forcing, seeking a general optimum solution by forcing selected sets of activities into the solution and rerunning phase 3; fixed charges are considered.

RRPLAN always enters the optimizer in phase 3. RRPLAN approaches the forcing issue in a more coherent manner than does WRAP or RAMP. Their method of single or double column forcing (i.e., phase 4) are of questionable efficiency, particularly where a site is linked to three or more sources or other sites. Column forcing operates from within the optimizer, and is therefore blind in the sense that there is no information within the optimizer on what the various columns represent. A great number of columns (for example, transportation activities) do not have intercepts, and thus do not need to be forced into the solution. Forcing such a column in becomes an indirect and inefficient way of forcing in the site which is at the receiving end of the transportation (if the site is not already in), or breaking one link to a site previously in the solution. If the column represents a site-process, it may facilitate the entry of a new site, but it will break no links to sites previously in the solution. What is more critical is that there is no way at all of forcing out a site that is well linked. Without that, the additional site, which might really be preferred, may be rejected because of its small size and because of the excess number of sites in the solution at the time of its consideration. Single column forcing can eventually force out a site which is linked to only one source or other site. Double column forcing can eventually force out a site which is linked to only two sources or other sites. But neither can assure that the optimizer will consider a solution without a site that is linked to three or more sources or other sites. In addition, double column forcing is expensive in computing time and cost.

---

[1]W.E. Walker, op.cit.

Because of the inefficiency of column forcing, two new forcing procedures called site forcing and site-process forcing were developed for RRPLAN. These new forcing methods operate from the MAIN program and are thus not blind. They operate in a coordinated way on several columns at once to force sites and site-processes in and out in an efficient manner. Site and site-process forcing procedures call the optimizer iteratively to perform a phase 3 for each separate force, and, as in column forcing, the better of the solution before and after the phase 3 is selected as the starting point for the next force. In site and site-process forcing, the list of sites (or site-processes) is passed through once, if a site (or site-process) is in the solution, it is forced out, and if it is not in the solution it is forced in.

In forcing out, very high costs (slope and intercept) are assigned to each of the columns representing the site (or site-process) and the optimizer is called for a phase 3. Then costs are reset, and the better of the solution before or after the force is selected as the basis for the next step. Forcing out is thus a one-step procedure.

In forcing in (really urging in), intercepts for all columns representing the site (or site-process) are set to zero, and the optimizer is called for a phase 3. If the site or site-process is still not in the solution, costs are reset, the better of the solutions before or after the force is selected as the basis, and the procedure passes on to the next site or site-process. If the site or site-process proves to have been "urged" into the solution, costs are reset, and the optimizer is recalled for an additional phase 3. The better of the solutions before or after the two-step force is selected as the basis for the next force, and the procedure passes on to the next site or site-process.

The algorithm as implemented in RRPLAN is designed to execute only phases 3 and 4. This is due to the fact that a starting basis is generated by RRPLAN and is always available to begin optimization in phase 3, thus bypassing the generation of a basis in phases 1 and 2. The logic flow would then move through label C to generate the current form of the A matrix in a sparse form (i.e., a form which recognizes that most of the entries are zero) to reduce the computations in phses 3 and 4. The fixed charge criterion is then used to find a solution better than the current one (or the initial basis as generated by the program). Iterations are performed until a locally optimal solution is found. The program then stores the current basis vector, the current objective value, the current level of the basic variables, and the last column to leave the basis. If the last phase was indicated as 3, then the program outputs the relevant input vectors and results based on the user-supplied print option variable, KOPT. Control is then returned to the MAIN program via label D. If the last phase was 4, (column forcing) then a sparse form of the current A matrix is generated, columns are forced into the current basis one or two at a time and the solution is tested for improvement. As before, the output is governed by the value of KOPT; control is then returned to the MAIN program via label D.

MAIN PROGRAM (PART 2)

The final page of the flowchart beginning with label D, represents the output of solutions found by the optimizer. All logic for this portion of the model is contained in the MAIN program. The first solution output consists of transportation activities in the solution. These activities are printed out according to their transportation category number (see table 2.2). The output includes the category number, the origin location, the destination location, the commodity shipped, and the number of commodity units shipped per year (e.g. Mlbs of steam).

The second output summarizes the process activity levels. Pass-through processing (the residue from source separation or source-separated paper) is listed first followed by a site-by-site description of processing. This information includes the site name and number, the process name and number, the location where processing is carried out, the linear segment on the cost function where processing takes place, the commodity and number of commodity units processed, and if appropriate the percent of capacity at which the plant operates. Information on dedicated transfer stations[1] and landfills are then presented.

The third type of solution output is concerned with the market activities appearing in the solution. Summary data for each market are output in sequence. A separate line of output is given for each segment of the market demand function (there may be up to five market segments). Cumulative sales and cumulative revenues for each market are also given. Information on sales and revenue of source-separated paper, cans and glass are then given. Implicit revenues (e.g., revenues from recoverables not explicitly traded and marketed by the model) are then presented by site. A revenue summary showing yearly averages for explicit commodities net of haul to the market (e.g. gross revenues minus haul costs), source-separted recoverables net of haul to the market, implicit (net) revenues and total net revenues is then printed out. Both discounted and undiscounted values are output for the revenue categories just mentioned. The fourth solution output is an analytical summary. It contains such items as the total tons of waste handled, the lifetime cost (both discounted and undiscounted), an accounting of costs by category and, if appropriate, information on energy.

[1]The model will consider siting a transfer station at each source in order to reduce the overall costs of the plan. For large communities this may be a wise approach because the savings in haul costs may greatly exceed the capital costs of the facility. The user has the option to suppress the dedicated transfer station logic if so desired. The choice of a dedicated transfer station at a source does not affect the size of the matrix.

RRPLAN also includes tracing. The logic flow picks up at label E. Tracing was incorporated into RRPLAN because three types of splits may occur in the solution due to site capacity constraints, landfill contraints and market limits. First, sources may be split among offload sites. Second, sites may be split among the processes offered. Finally, commodities arising from outputs of a process may be split among offload sites and markets. Optional routing requests are also included in tracing. For example, consider a site where wastes can be processed. In this case, the procedure will trace all site splits, through the output coefficients of each process, and commodity splits, to a series of sites and markets. The final tracing output summarizes source and site costs. These costs include linkage and offload costs for sources and on-site and net off-site costs. The site cost is a projected full-cost tipping fee for each site in the solution.

2.2 DATA STRUCTURES

The purpose of this section is to describe the three data files which define the application problem. In order to differentiate general formulations from specific applications, the following convention has been adopted: (1) general data are summarized in tables; (2) organizational characteristics are summarized in figures; and (3) the contents of a specific file are listed in exhibits. The tables focus on generic information only (e.g., variable name, format, purpose, range, etc.). Cross reference between exhibits 2.1, figure 2.3 and tables 2.7 through 2.30 should be sufficient to thoroughly understand how the model approaches an application problem.

The inputs are read in three files, which will be referred to as ACASE, the case study file, APKR, the packer distance file, and AVAN, the van distance file. ACASE is read on unit 5, APKR is read on unit 7, and AVAN is read on unit 8. Users must therefore provide as part of their input deck control cards which will enable the operating system to correctly read the contents of the three files.

The format specifications for the ACASE file may best be introduced through reference to the major input variables. These variables are defined in table 2.3; they may be classified into one or more of the three overlapping categories. These categories are those variables which: (1) set the range on do-loops; (2) determine the input status for conditional card groups; and (3) have an important influence on problem size.[1] The order in which all data from the ACASE file are read, is summarized in table 2.4. For all card groups except the paper market price distribution function data, the column labeled KEY corresponds to the entries in the first two columns of the card.

[1]Throughout the discussion which follows, the term card image can be substituted for card without loss of meaning.

Figure 2.3 illustrates how the ACASE file would appear in practice. All card groups are labeled. If the input status for a particular card group is conditional, the card is so labeled. The figure also illustrates which card groups are composed of a single card and which card groups require a block of cards.

All number assignment must begin with 01 and be consecutive, with no gaps. Thus the Jth card in a sequence will also have an identification number equal to J, and the last identification number will be equal to the total number of the entity entered on the control card (KEY 01). This requirement applies to the numbering of cost categories (KEY 08), energy categories (KEY 09), commodity categories (KEY 10), sources (KEY 11), sites (KEY 12), processes (KEY 14), markets (KEY 18), and cost summation categories (KEY 21). All cards within a numbered KEY type must be entered in sequence (e.g., cost category 1, cost category 2, etc.). Except for process cards (KEY 14, 15, and 16) and market cards (KEY 18, 19, and 20), all cards must be entered in order by KEY.

In entering the paper market price distribution function, all sixteen fields on one card should be used before starting the next card; there should be no gaps. A blank field will be interpreted as zero frequency. RRPLAN will read a number of fields equal to NOINT and they will be the first NOINT fields entered, at sixteen per card. It goes without saying that the cards should be entered in the same order as the frequency. Frequency intervals should be entered from lowest price to highest price.

The landfill cards (KEY 13) need not be sorted within KEY type. If the user wishes to sort them, a sort by location number is recommended. The SIPROC cards (KEY 17) need not be sequenced within KEY type. If the user wishes to sort them for purposes of a more orderly output, process number within site is recommended.

For processes, for each numbered process (in order of process number) enter all cards of the process (in card number order) before entering the first card of the next process. The number of third process cards for a process must be the same as the number of segments, NSEG(J), entered for that process on the first process card. For markets, for each numbered (in order of number), enter all cards of the market (in card number order) before entering the first card of the next market.

Transportation cards (KEY 22) must be sequenced by category type, ITTYP(J), entered on each transportation card in column 4, but they need not be sequenced within category type. In other words, all category 1 cards should be entered before the first category 2 card, and all category 2 cards should be entered before the first category 3 card. The transportation file must be followed by a single card with 23 entered as the read key.

Inputs are read, edit checked, and then listed by related card-group, before the next related card group is read. This procedure was designed to give the user rough guidance to the location of a problem in the data file if there is system failure to read an input (as for example a format mismatch). A sequence failure or a failure in edit check, if the card can still be read, will lead to an immediate report of the failure and an abort of the run. In a

Table 2.3 Major Input Variables

Variable Name	Purpose of Variable	Minimum Value	Maximum Value
INPAP[a]	Input mode for paper price distribution	0	1
KSEPO[b]	Run control for unconditional source separation	0	2
NCC	Number of cost categories	1	50
NCO	Number of commodity categories	5	20
NCOF(K)	Number of output coefficients for process K	0	7
NCS	Number of cost summation categories	0	10
NEC	Number of energy categories	0[c]	20
NLA	Number of landfills	0	30
NMK	Numbers of markets	0[d]	30
NOINT	Number of intervals in the paper price distribution	0[d]	200
NPR	Number of processes	2	30
NSEG(K)	Number of cost/energy linear segments for process K	1	3
NSGMK(K)	Number of segments for the Kth market	1	5
NSI	Number of sites	1	50
NSP	Number of site-process combinations	1	80
NSU	Number of sources		
NTRIG	Number of trigger prices used in the run	0	5

[a]If 0, input trigger prices and for each a percent of burn and an average price for marketed paper; if 1, input trigger prices and a frequency distribution for prices in the paper market. (Trigger prices are discussed in table 2.12)

[b]If 0, unconditional source separation is not available; if 1, unconditional source separation is available with a national paper market, if 2, unconditional source separation is available with a local paper market.

[c]If the objective criterion is maximize net energy, then the minimum value is 1.

[d]If a local paper market exits (KSEPO = 2), then the minimum value is 1.

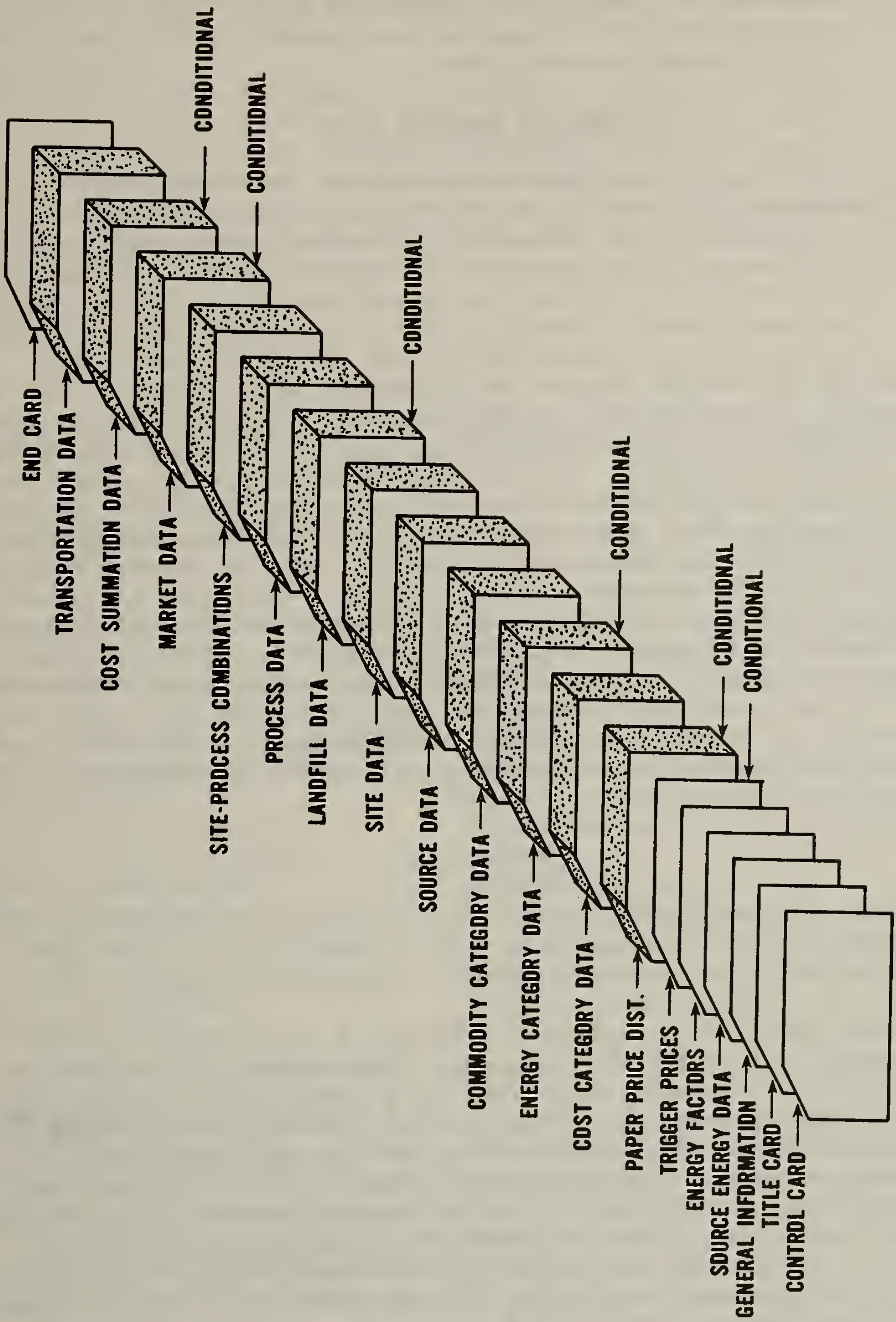
Table 2.4 User Input Groups

KEY	Card Group Type	Input Status
01	Control Card	Required always as first record.
02	Title Card	Required always as second record.
03	General Information	Required always as third record.
04	Source Energy Card	Required always as fourth record.
05	Energy Factors Card	Required always as fifth record.
06	Paper Market Data Card	Conditional: omit if KSEPO=2 or NTRIG=0.
07	Paper Market Price Trigger Prices	Conditional: omit if KSEPO=2 or NTRIG=0.
--	Paper Price Distribution	Conditional: omit if KSEPO=2 or NTRIG=0, or INPAP=0.
08	Cost Category Data	Required: NCC cards must be entered sequenced by number.
09	Energy Category Data	Conditional: omit if NEC=0, otherwise NEC cards must be entered sequenced by number.
10	Commodity Category Data	Required: NCO cards must be entered sequenced by number.
11	Source Data	Required: NSU cards must be entered sequenced by number.
12	Site Data	Required: NSI cards must be entered sequenced by number.
13	Landfill Data	Conditional: omit if NLA=0, otherwise NLA cards must be entered. No sequence required, but referred sequence is by location.
14	First Process Card	Required.[a]
15	Second Process Card	Required.[a]
16	Third Process Card	Required. [a,b]
17	Site-Process Combination Data	Required: NSP cards must be entered. No sequence required, but preferred sequence is by process number within site number.
18	First Market Card	Conditional: omit if NMK=0.[c]
19	Second Market Card	Conditional: omit if NMK=0.[c]
20	Third Market Card	Conditional: omit if NMK=0.[c]
21	Cost Summation Data	Conditional: omit if NCS=0, otherwise NCS cards must be entered sequenced by number.
22	Transportation Data	Required: sequenced by category type.
23	End Card for Transportation File	Required always as last record.

[a]Enter the first, second and third process cards sorted by process number (major) and card group (minor). Enter NPR sets.

[b]The number of third process cards in the set must be equal to NSEG(K), the number of linear segments for the Kth process, entered on the first process card.

[c]Enter the first, second and third market cards sorted by market number (major) and card group (minor). Enter NMK sets.


END CARD
TRANSPORTATION DATA
COST SUMMATION DATA
MARKET DATA
SITE-PROCESS COMBINATIONS
PROCESS DATA
LANDFILL DATA
SITE DATA
SOURCE DATA
COMMODITY CATEGORY DATA
ENERGY CATEGORY DATA
COST CATEGORY DATA
PAPER PRICE DIST.
TRIGGER PRICES
ENERGY FACTORS
SOURCE ENERGY DATA
GENERAL INFORMATION
TITLE CARD
CONTROL CARD
CONDITIONAL
CONDITIONAL
CONDITIONAL
CONDITIONAL
CONDITIONAL
CONDITIONAL

sequence failure, RRPLAN will identify the card that was read (group -- i.e., read key, and number within group) and the card that was expected. In an edit check failure, RRPLAN will identify the card group (i.e., the read key), the card number of sequence, and the integer or real number that failed the edit check (the latter will show a decimal point).

ESTIMATING PROBLEM SIZE

The A matrix is currently sized for a maximum of 90 rows (equations) and 360 columns (activities). The types of equations and activities which make up the A matrix were defined in tables 2.1 and 2.2; a formula for computing each dimension of the A matrix is presented in tables 2.5 and 2.6. The matrix is designed to be sparse (mostly zeroes) but it may have up to 8100 non-zero elements. If the user wishes to exercise the double column forcing option, then the maximum size of the A matrix is 50 rows by 100 columns. This limitation can be relaxed to 90 rows by 360 columns with the use of an override control. Since the site or site-process forcing options are almost certainly superior to double column forcing, the use of the override is not recommended for large problems. The key input variables which affect problem size, their purpose and allowable values were given in table 2.3. It is important to point out that if several of the key input variables are set at their maximum value during the course of a single run, the maximum size of the A matrix is likely to be exceeded. It is for this reason that a formula for computing the size of the A matrix prior to execution is included. With regard to transportation activities, linkages must be provided from each source and to each market. If unconditional source separation is offered in a region with a local paper market, then the linkages to the local paper market will be generated by the model. Process input linkages to each site must be provided as well as process output linkages designated for shipment to other sites for additional processing or disposal.

INPUT SUMMARY[1]

The input format for each group of cards are summarized in the tables which follow. Exhibit 2.1 is a listing of a typical ACASE file. RRPLAN uses integer, single precision real and character variables as inputs. All

[1]The card images described in tables 2.7 through 2.31 are tied to specific FORMAT statements within the MAIN program. Consequently, if one were to refer to the FORMAT statement, he would find a FORMAT specification identical to that which appears under the column labeled "Variable Specification". This was done to insure that anyone who may be called on to modify the source code will know not only the exact column alignment but also spaces available for the insertion of new variables. It is important to note that the tables shown in NBS Special Publication 657 do not include any reference to "fillers". It was thought that their inclusion might lead to confusion on the part of certain users. Similarly, if a variable has a maximum value of less than 100 and the variable is read according to an I3 specification, the variable is listed in NBS Special Publication 657 as I2 and the first column is listed as being one space to the right from that which appears in this report. Technical specialists should be aware of these differences as they appear in tables 2.7 through 2.31 and in the Dictionary of Input Variables. All variables for which such differences exist are marked clearly in appendix A.1.

variables which appear in the tables are defined as: (1) In, refers to an n-digit integer; (2) $F_j.k$ refers to a j-digit single precision real number with k digits to the right of the decimal point; and (3) mAp, refers to a character string with m blocks of p characters (i.e., mp characters).

In the discussion which follows, it is sometimes necessary to refer to a specific element within an array. So that this may be done efficiently, a labeling convention has been adopted and is used throughout the report. If the array is one-dimensional, the index is listed as K (i.e., ARRAY(K)). If the array is two-dimensional, the indices are J and K, respectively (i.e., ARRAY(J,K)). If the array is three-dimensional, the indices are I, J and K, respectively (i.e., ARRAY(I,J,K)).

Table 2.5 Guidelines for Calculating the Number of Rows in the A Matrix

Constraint Type	Allocate one row for each:
Source balance equations	Source
Source paper balance equations	Source coded for unconditional source separation with a local paper market
Site input residue balance equations	Site capable of receiving source separation residue
Site input paper balance equations	Site capable of receiving source separated paper
Site input balance equations for MSW	Site receiving MSW
Site input balance equations for other commodities	Site receiving any non-MSW commodity
Site output balance equations	Location-commodity combination
Market input balance equations	Market
Site capacity constraints	Capacitated site
Land capacity constraints	Land using location
Market bounds	Market segment for which there is an upper bound
Constraint on artificial processing activities	Problem formulation[a]

[a]The model builds in one column per site to take on a very small value (epsilon) without incurring a fixed cost. Such columns are needed to balance epsilon values on the right hand sides of site input balance equations, which inturn retain these equations in use even if the sites are actually inactive. Total inactivity of an equation could cause an inability to invert the basis during optimization. These artificial processing activities are upper bounded with a single constraint which is always the last row in the matrix.

Table 2.6 Guidelines for Calculating the Number of Columns in the A Matrix

Activity Type	Allocate one column for each:
Category 2 transportation	Site-to-site shipment declared on read KEY 22
Category 3 transportation	Site-to-market shipment declared on read KEY 22
Category 4 transportation	Source-to-site shipment of unseparated MSW[a]
Category 5 transportation	Source-to-site shipment of source separated MSW (National paper market)[b]
Category 6 transportation	Source-to-site shipment of source separated MSW (Local paper market)[c]
Category 8 transportation	Source-to-site shipment of preseparated paper (Local paper market)[c]
Category 9 transportation	Source-to-market shipment of preseparated paper (Local paper market)[c]
Category 11 transportation	Source-to-site shipment of source separated MSW (National paper market)[d]
Category 12 transportation	Source-to-site shipment of source separated MSW (National paper market)[d]
Category 13 transportation	Source-to-site shipment of source separated MSW (National paper market)[d]
Category 14 transportation	Source-to-site shipment of source separated MSW (National paper market)[d]
Category 15 transportation	Source-to-site shipment of source separated MSW (National paper market)[d]
Residue	Site at which source separation residue may arrive
Paper	Site at which source separated paper may arrive
Processing	Linear segment of a process at a given site
Artificial processing activities	Site[e]
NZ	Problem formulation[f]
NZX	Scenario evaluation[g]

[a]Assumes KMIX=1 on card 3 and IMIXSU(K)=1 for the source under consideration.

[b]Assumes KSEPO=1 on card 3 and ISEPSU=1 or 2 for the source under consideration.

[c]Assumes KSEPO=2 on card 3 and ISEPSU=1 or 2 for the source under consideration.

[d]Assumes KSEPO=1 on card 3 and ITRIGS (K)=1 for the source under consideration. The value of K is equal to the transportation category minus ten.

[e]See note a on table 2.5.

[f]Relieves infeasibilities associated with an advanced starting point.

[g]Relieves infeasibilities when performing a scenario evaluation. Is not to be counted unless LOPT=4.

Exhibit 2.1 Listing of a Typical Case Study File

```
01310030   20 1983   10.  0000  6.       8  4  2  4  4  5  5  2   9  3  1   20.
02 TEST CASE
03 23.211.8   8. 57.   5.0   6.0  25.49 15.  1 1 20.  1 2.   11131 1 .8  .60 .9
04  1 .8       1 .1        .25        .20        .15
0520.        12.0        12.0        15.2        2 1 10. 1-999.
06     15.  20.  25.
07  5  5.  12.
10.  20.  30.  20.  10.
08   1 MUNICIPAL COST          0  1.        2.
08   2 CONTRACTOR OP COST      0  1.        -3.
08   3 CONTRACTOR CAP COST     1  1.        1.     20   15 9.
08   4 SITE PREP COST          1  1.        0.     50   15 9.
09   1 FUEL INPUT, MBTU         -1.
09   2 STEAM OUTPUT, MBTU       -1.
10   1 MSW IN PKR, KTON        1 1.        1 0.        1 .20        1 0.
10   2 MSW IN VAN, KTON        1 1.        1 0.        1 .25        1 0.
10   3 S-S PAPER, KTON         1 .7        1 0.        1 .15        1 0.
10   4 S-S RES IN PKR, KTON 1 1.           1 0.        1 .20        1 0.
10   5 S-S RES IN VAN, KTON 1 1.           1 0.        1 .20        1 0.
10   6 STEAM, MLBS             2 1.        2 0.        2 1.         2 0.
10   7 PROC RESIDUE, KTON      2 .8        2 0.        1 .20        1 0.
10   8 INCIN FERROUS, KTON     2 .7        2 0.        1 .10        1 0.
119  1 LYNN                     1 340.       01111    1 1 2.0  1.5  1.2
119  2 MARBLEHEAD               2 7.         11           2.2  1.7  1.4
119  3 SALEM                    3 120.       01111        2.1  1.6  1.3
119  4 REVERE                   4 40.        01           1.9  1.4  1.1
129  1 SAUGUS PROC             05    2 10 1  41000.       1250.
129  2 WORCESTER PROC          06    2 00 2  41000.       1250.
129  3 AMESBURY LANDFILL       07    7 01 1  4200.        150.
129  4 SAUGUS SECONDARY PR     05    7 01 1  4400.        1100.
13   7 AMESBURY LANDFILL      99600.
13   5 SAUGUS LANDFILL        84000.
14   1 TRANS STA                20130           2 3 1 1
15   1  21.0
16   1120.       20.       .8       0.       1.8       1.8       .5       0.
16   1240.       35.       .7       0.       .50       .61       .4       0.
16   1360.       55.       .6       0.       .26       .24       .3       0.
14   2 WATER-WALL INCIN         21310           2 3 1 1
15   2  65.48     7.15       8.07
16   21300.      0.        70.      0.       7.3       0.        2.       0.
14   3 WATER-WALL INCIN         20320           2 3 1 1
15   3  65.48     7.15       8.07
16   31300.      224.16    80.      0.       7.3       7.0       2.       0.
16   321000.     1096.64   170.     0.       5.0       4.236     1.5      0.
14   4 LANDFILL                 700311.653      2 3 1 1
15   4
16   4140.       40.       1.6      0.       3.6       3.6       1.0      0.
16   4280.       70.       1.4      0.       1.0       1.22      .8       0.
16   43120.      110.      1.2      0.       .52       .48       .6       0.
14   5 SECONDARY PROC           70021.82        2 3 1 1
15   5
16   5140.       40.       1.6      0.       2.        2.        1.0      0.
16   5280.       70.       1.4      0.       1.        1.        0.8      0.
17   1    2 360.      1.0       1     2  1.2
17   2    1                     111   2  10.0
17   2    3           1.0       11    2  1.3
17   3    4                      11   2  10.0
```

Exhibit 2.1 Listing of a Typical Case Study File (Continued)

```
17   4   5             3.0         1    2  11.0
18   1 GE LYNN STEAM            5   6   2   1   2   1 225.4       -.2
19900.    1800.                                                                 1
203.0     2.0                           0.      0.                              1
18   2 NEW YORK FERROUS         8   8   1   1   2   1 10.2        -.2
19                                                                              2
2011.86                                 0.                                      2
18   3 WORCESTER STEAM          9   6   2   1   2   1 225.4       -.2
192000.  4000.                                                                  3
208.0     6.0                           0.      0.                              3
21   1 CONTRACTOR COST        2   2   3
22 1   1        1
22 1   1        2
22 1   2        1
22 1   2        2
22 1   3        1
22 1   3        2
22 1   4        1
22 1   4        2
22 2   2   6    1
22 2   7   5    3
22 2   7   5    4
22 2   7   6    3
22 2   7   6    4
22 3   6   5    1
22 3   8   5    2
22 3   6   6    3
22 3   8   6    2
23
```

Table 2.7 Control Card (KEY = 1)

The first card in the case study file sets the controls for the optimizer based on the values contained in the six entries LPHASE, NFORC, KOVER, KOPT, LOPT, KFOUT and KSUPP. The economic scope of the analysis is determined to a great degree by NYR, IFIRST, IDSCO and FLATO and to a lesser extent by ISQR, IXDED and PRPAP. All other entries serve to define end points of do loops through which much of the remaining data is read in.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	1	1
LPHASE	I1	3	3	4
NFORC	I1	4	1	2
KOVER	I2	5	0	99
KOPT	I1	7	1	3
LOPT	I1	8	0	4
FILLER	3X			
NYR	I2	12	1	99
FILLER	1X			
IFIRST	I4	15	-	--
DISCO	F8.4	19	0.0	50.0
ISQR	I1	27	0	1
IXDED	I1	28	0	1
KFOUT	I1	29	0	9
KSUPP	I1	30	0	1
FLATO	F8.4	31	0.0	50.0
FILLER	1X			
NCO	I2	40	5	20
FILLER	1X			
NCC	I2	43	1	50
FILLER	1X			
NEC	I2	46	0	20
FILLER	1X			
NSU	I2	49	1	60
FILLER	1X			
NSI	I2	52	1	50
FILLER	1X			
NPR	I2	55	2	30
FILLER	1X			
NSP	I2	58	1	80
FILLER	1X			
NLA	I2	61	0	30
FILLER	1X			
LSTLO	I3	64	1	999
FILLER	1X			
NMK	I2	68	0	30
FILLER	1X			
NCS	I2	71	0	10
FILLER	2X			
PRPAP	F5.2	75	-	--

Table 2.8 Title Card (KEY = 2)

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	2	2
KITLE(K)	19A4	3	---	---

Table 2.9 General Information Card (KEY = 3)

This card establishes values for the composition of the solid waste stream and the BTU content of its components. Information on collection costs, potential revenues and their associated cost allocations are given as well as controls which govern the use of the source separation options for the rest of the analysis. The last entries provide information on the relative costs of processing source separated paper and residue.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	3	3
PCPAP	F5.2	3	0.0	90.0
PCGLA	F5.2	8	0.0	90.0
PCCAN	F5.2	13	0.0	90.0
PCRES	F5.2	18	0.0	100.0
BTRES	F6.3	23	0.0	--
BTPAP	F6.3	29	0.0	--
PRGLA	F5.2	35	0.0	1000.0
PRCAN	F5.2	40	0.0	1000.0
ICCGLA	I2	45	1	NCC
ICCCAN	I2	47	1	NCC
CSPA	F5.3	49	0.0	1000.0
ISPA	I2	54	1	NCC
CSPB	F5.3	56	0.0	1000.0
ISPB	I2	61	1	NCC
KMIX	I1	63	0	1
KSEPO	I1	64	0	2
NTRIG	I1	65	0	5
INPAP	I1	66	0	1
ICCPAP	I2	67	1	NCC
RPRPAP	F4.3	69	0.05	20.0
ALLO	F4.3	73	0.0	1.0
RPRRES	F4.3	77	0.05	20.0

Table 2.10 Source Energy Card (KEY = 4)

KSPA and ESPA set the value for net energy in collection and its allocation; whereas KSPB and ESPB determine how much additional net energy is required for source separation. RPAH, RGLH and RCAH are factors to convert the dollar value of haul costs to an equivalent amount of net energy.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	4	4
FILLER	1X			
KSPA	I2	4	1	NEC
ESPA	F10.3	6	-	---
KSPB	I2	16	1	NEC
ESPB	F10.3	18	-	---
RPAH	F10.5	28	-	---
RGLH	F10.5	38	-	---
RCAH	F10.5	48	-	---

Table 2.11 Energy Factors Card (KEY = 5)

The energy values of source separated paper, glass and cans at the market and the appropriate energy categories are given on this card. The thermal content of MSW in millions of BTU are given as well as an energy category for BTU differentiation among MSW and source separated paper and residue. The last four entries are concerned with the forcing procedure. They permit the user to set controls for overriding the fixed and variable cost forcing values set internally.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	5	5
EPAPM	F10.5	3	-	---
IPAPM	I2	13	1	NEC
EGLAM	F10.5	15	-	---
IGLAM	I2	25	1	NEC
ECANM	F10.5	27	-	---
ICANM	I2	37	1	NEC
BTMSW	F10.5	39	0.05	---
IBTU	I2	49	1	NEC
FILLER	1X			
IFDRV	I1	52	0	1
FDRIV	F5.2	53	-	---
IVDRV	I1	58	0	1
VDRIV	F5.2	59	-	---

Table 2.12 Paper Market Data Card (KEY = 6)

The user has the option to specify a set of trigger prices below which paper will be burned for its BTU value and above which it will be sold for its fiber content. The PTRIG(1) through PTRIG(5) entries establish these trigger prices. The value of NTRIG given on card 3 determines how many trigger prices the user must input. Note that the trigger prices must be increasing. The model will select which trigger price and burn ratio are most appropriate for each source where they are offered. The user may also specify how much paper is burned and what the average sale price is through the values of BTRIG and ATRIG by setting INPAP equal to zero on card 3. Both sets of values must be increasing functions reflecting the fact that as the trigger price is increased, more paper will be burned and the average value at the market will be higher. If the user desires to input a frequency distribution for the price of paper, then these values are computed internally. In such a case, INPAP should be set to one on card 3 and no values should be entered for either ATRIG or BTRIG.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	6	6
FILLER	4X			
PTRIG(1)	F5.2	7		
PTRIG(2)	F5.2	12		
PTRIG(3)	F5.2	17	a	
PTRIG(4)	F5.2	22		
PTRIG(5)	F5.2	27		
BTRIG(1)	F4.1	32		
BTRIG(2)	F4.1	36		
BTRIG(3)	F4.1	40	b	
BTRIG(4)	F4.1	44		
BTRIG(5)	F4.1	48		
ATRIG(1)	F5.2	52		
ATRIG(2)	F5.2	57		
ATRIG(3)	F5.2	62	c	
ATRIG(4)	F5.2	67		
ATRIG(5)	F5.2	72		

a PTRIG(K) ⩾ PTRIG(K-1) + 0.001
b BTRIG(K) ⩾ BTRIG(K-1) + 0.0001
c ATRIG(K) ⩾ ATRIG(K-1) + 0.0001

Table 2.13 Paper Market Price Distribution Function Header (KEY = 7)

The data provided on this card define the size and shape of the paper price distribution. The minimum price is given by BOTTOM, whereas the maximum price is equal to the product of NOINT and SIZINT plus BOTTOM. The minimum width of the interval on the paper price distribution is defined so that all trigger prices are contained within the body of the paper price distribution.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	7	7
NOINT	I3	3	1	200
SIZINT	F5.3	6	a	1000
BOTTOM	F5.2	11	0.	>PTRIG(1)

a (PTRIG(NTRIG) - BOTTOM)/NOINT

Table 2.14 Paper Market Price Distribution Data

The paper price distribution is entered as frequencies over ranges of price in the market. The distribution is normalized before use so that the frequencies may be entered in any form that is convenient to the user.

Variable Name	Variable Specification	First Column	Minimum Value[a]	Maximum Value
FREQ(1)	F5.3	1	0.0	--
FREQ(2)	F5.3	6	0.0	--
FREQ(3)	F5.3	11	0.0	--
FREQ(4)	F5.3	16	0.0	--
FREQ(5)	F5.3	21	0.0	--
FREQ(6)	F5.3	26	0.0	--
FREQ(7)	F5.3	31	0.0	--
FREQ(8)	F5.3	36	0.0	--
FREQ(9)	F5.3	41	0.0	--
FREQ(10)	F5.3	46	0.0	--
FREQ(11)	F5.3	51	0.0	--
FREQ(12)	F5.3	56	0.0	--
FREQ(13)	F5.3	61	0.0	--
FREQ(14)	F5.3	66	0.0	--
FREQ(15)	F5.3	71	0.0	--
FREQ(16)	F5.3	76	0.0	--

a $\sum_{i=1}^{NOINT} FREQ(i) > 0$

Table 2.15 Cost Category Data (KEY = 8)

This is one of the NCC cost category cards; it deals with either annual recurring costs referred to within the model as operating costs or capital costs. The name of the cost category is stored in the alphanumeric variable NAMCC(J,K). IDCC(K) is set equal to zero or one indicating the type of cost. Costs are escalated according to the value of the the differential inflation rate DIFLAT(K). Values for WTCC(K) are only necessary if LOPT is equal to 3 or 4 in which case positive weights for WTCC(K) are required. At least one weight must be non zero for LOPT equal to 3. In the LOPT equal to 4 case, the weights are not checked, but the model will select among alternatives randomly if at least one weight is not entered. The LIFU(K), LIFC(K), RATE(K) and NREM(K) entries are not required for an operating cost, so they should be left blank.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	8	8
FILLER	1X			
ICC(K)	I3	4	1	NCC
FILLER	1X			
NAMCC(J,K)	5A4	8	-	---
FILLER	1X			
IDCC(K)	I1	29	0	1
FILLER	1X			
WTCC(K)	F8.3	31	-	---
DIFLAT(K)	F5.3	39	-	---
LIFU(K)	I3	45	1	200
LIFC(K)	I3	49	1	LIFU
RATE(K)	F4.1	52	0.5	80.0
NREM(K)	I2	56	0	99

Table 2.16 Energy Category Data (KEY = 9)

This is one of NEC energy cards. If NEC was listed as zero on card 1, then no cards are to be entered; the model will however, create an artificial energy category for accounting purposes. All energy categories represent <u>net</u> energy. Note that WTEC(K) should have weights entered which are negative or zero whether the category is expected to show a net energy surplus or a net energy requirement; for LOPT equal to 3 or 4 at least one weight should be non zero.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	9	9
FILLER	1X			
IEC(K)	I2	5	1	NEC
FILLER	1X			
NAMEC(J,K)	5A4	8	-	---
FILLER	1X			
WTEC(K)	F8.3	29	-	---

Table 2.17 Commodity Category Data (KEY = 10)

The commodity is numbered by the assignment of ICO(K) the commodity category cards; there are NCO such cards. The first five commodities must be:

1. Mixed MSW in a packer in KTPY (kilotons per year);
2. Mixed MSW in a van in KTPY;
3. Source-separated paper in a van in KTPY;
4. Source-separated residue in a packer in KTPY; and
5. Source-separated residue in a van in KTPY.

Once identified, commodities are referenced by number for site inputs, process inputs and outputs, market inputs, and transportation. Haul cost multipliers CCOCC(1,K) and CCOCC(2,K) and energy multipliers CCOEC(1,K) and CCOEC(2,K) are also entered on this card. These multipliers are applied against the packer and van distance file entries (van applies if ICO is other than 1 or 4).

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	10	10
FILLER	1X			
ICO(K)	I3	4	1	NCO
FILLER	1X			
NAMCO(J,K)	5A4	8	---	---
ICOCC(1,K)	I2	28	1	NCC
CCOCC(1,K)	F8.3	30	0.00001	1000.0
ICOCC(2,K)	I2	38	1	NCC
CCOCC(2,K)	F8.3	40	0.0	1000.0
ICOEC(1,K)	I2	48	0	NEC
CCOEC(1,K)	F8.3	50	---	---
ICOEC(2,K)	I2	58	0	NEC
CCOEC(2,K)	F8.3	60	---	---

Table 2.18 Source Data (KEY = 11)

This is one of NSU source cards. The second entry is designed for tracing activities to be carried out once the optimization has been completed. The next four entries show the source number, its name, location and rate of waste generation (in KTPY). All remaining entries are associated with the source separation options to be considered for this source. Any combination of the seven entries IMIXSU(K) through ITRIGS(5,K) may be offered. It is important to note that the source controls are dominated by the run controls (KMIX,KSEPO and NTRIG) listed on card 3. If the run control indicates that the alternative was not to be considered, then the corresponding source control will be read but will not be functional.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	11	11
ISUREQ(K)	I1	3	0	9
ISU(K)	I3	4	1	NSU
FILLER	1X			
NAMSU(J,K)	5A4	8	--	---
FILLER	1X			
LSU(K)	I3	29	1	LSTLO
TSU(K)	F10.3	32	0.001	---
FILLER	1X			
IMIXSU(K)	I1	43	0	1
ISEPSU(K)	I1	44	0	2
ITRIGS(1,K)	I1	45	0	1
ITRIGS(2,K)	I1	46	0	1
ITRIGS(3,K)	I1	47	0	1
ITRIGS(4,K)	I1	48	0	1
ITRIGS(5,K)	I1	49	0	1
IHCSEP(K)	I2	50	0	NCC
IHESEP(K)	I2	52	0	NEC
CHPA(K)	F5.2	54	0.0	1000.0
CHGL(K)	F5.2	59	0.0	1000.0
CHCA(K)	F5.2	64	0.0	1000.0

Table 2.19 Site Data (KEY = 12)

This is one of NSI site cards. As for sources, the second entry is designed for tracing activities to be carried out once the optimization has been completed. The next three entries show the site number, name and location. The incoming commodity is then given. For MSW-receiving sites, enter 2 for INCO(K). If the site is not MSW receiving, INCO(K) must be greater than 5, since the first five commodities are various forms of source generated waste. The next three entries determine if the site is capacitated, has land available for landfilling, and the number of processes to be considered. The last four entries deal with site preparation activities.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	12	12
ISIREQ(K)	I1	3	0	9
ISI(K)	I3	4	1	NSI
FILLER	1X			
NAMSI(J,K)	5A4	8	---	---
FILLER	1X			
LSI(K)	I3	29	1	LSTLO
FILLER	1X			
INCO(K)	I3	33	2	NCO
FILLER	1X			
ICAP(K)	I1	37	0	1
ILAND(K)	I1	38	0	1
NPROSI(K)	I2	39	1	NPR
FILLER	1X			
ICPREP(K)	I2	42	0	NCC
CCPREP(K)	F10.3	44	---	---
IEPREP(K)	I2	54	0	NEC
CEPREP(K)	F10.3	56	---	---

Table 2.20 Landfill Data (KEY = 13)

If the number of landfills, NLA, was listed as zero on card 1, then no cards of this type should be entered. There should be one card per location at which ILAND(K) on the site card was coded equal to one.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	13	13
FILLER	1X			
LLA(K)	I3	4	1	LSTLO
FILLER	1X			
NAMLA(J,K)	5A4	8	---	---
FILLER	1X			
AVLA(K)	F10.3	29	0.001	---

Table 2.21 First Process Card (KEY = 14)

Each process is identified by a process number, IPR(K), and by its position in the sequence 1 through NPR. With the exception of a process receiving MSW, each process must receive exactly one commodity. The process input commodity, INCOP(K), should be coded as 2 if it is MSW receiving; otherwise a value greater than 5 must be entered. If the process was in existence at the beginning of the study period IEXIS(K) should be coded as 1. The number of commodities output and the number of segments in the cost function are then given. The next two entries establish a means for recording landfill activites. The last four variables are designed for allocating costs (and energy) among two distinct categories (e.g., for costs, operation and maintenance versus capital).

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	14	14
FILLER	1X			
IPR(K)	I3	4	1	NPR
FILLER	1X			
NAMPR(J,K)	5A4	8	---	---
INCOP(K)	I3	28	1	NCO
IEXIS(K)	I1	31	0	1
NCOF(K)	I1	32	0	7
NSEG(K)	I1	33	1	3
ILACOF(K)	I1	34	0	1
USLA(K)	F8.5	35	a	---
FILLER	1X			
ICPR(J,1)	I2	44	1	NCC
ICPR(J,2)	I2	46	1	NCC
IEPR(J,1)	I2	48	0	NEC
IEPR(J,2)	I2	50	0	NEC

a If ILACOF(K)=0, then the value is irrelevant.
If ILACOF(K)=1, then the minimum is 0.0001.

Table 2.22 Second Process Card (KEY = 15)

For each process there is a card identifying which commodities are output and in what quantity as a function of the amount of waste processed. All outputs are measured in the units appropriate for that commodity (e.g., MBTU per unit input). This card must be entered even if the process does not produce an output (e.g., disposal in a landfill). In such a case the only entries are KEY and KEMP; they serve as placeholders. Enter a number of ICOF(K) and COF(K) equal to NCOF on KEY 14. Enter a number of ICOF(J,K) and COF(J,K) equal to NCOF(K) read on the previous card.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	15	15
FILLER	1X			
KEMP	I3	4	1	NPR
FILLER	1X			
ICOF(1,K)	I2	8	1	NCO
COF(1,K)	F8.5	10	0.00001	---
ICOF(2,K)	I2	18	1	NCO
COF(2,K)	F8.5	20	0.00001	---
ICOF(3,K)	I2	28	1	NCO
COF(3,K)	F8.5	30	0.00001	---
ICOF(4,K)	I2	38	1	NCO
COF(4,K)	F8.5	40	0.00001	---
ICOF(5,K)	I2	48	1	NCO
COF(5,K)	F8.5	50	0.00001	---
ICOF(6,K)	I2	58	1	NCO
COF(6,K)	F8.5	60	0.00001	---
ICOF(7,K)	I2	68	1	NCO
COF(7,K)	F8.5	70	0.00001	---

Table 2.23 Third Process Card (KEY = 16)

Basic processing costs are entered on the third process card as slopes and intercepts for each linear segment of the process. There are NSEG(K) such cards for each process. Two cost categories (and two energy categories) are provided, enabling the user to handle a wide variety of nonlinearities in the cost (and net energy) functions. The basic units of the model are thousand tons and thousand dollars. Most cost inputs may therefore be entered in dollars per ton because that is equivalent to thousand dollars per thousand tons. There are two exceptions. First, intercept costs must be entered in thousands of dollars. These include the site preparation cost, CCPREP and the two cost function intercepts, CINT(I,J,1) and CINT(I,J,2). Note that there will be NSEG(K) values for CINT(I,J,1) and CINT(I,J,2). Second, if the user has defined the unit of a commodity in other than thousand tons, the cost inputs into its process, including CSLO(I,J,1) and CSLO(I,J,2) and REVSP(K) of the commodity, must be in thousands of dollars per defined unit of the commodity.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	16	16
FILLER	1X			
KEMP	I3	4	1	NPR
LEMP	I1	7	1	3
CINT(I,J,1)	F9.2	8	--	---
CINT(I,J,2)	F9.2	17	--	---
EINT(I,J,1)	F9.2	26	--	---
EINT(I,J,2)	F9.2	35	--	---
CSLO(I,J,1)	F9.6	44	--	---
CSLO(I,J,2)	F9.6	53	--	---
ESLO(I,J,1)	F9.6	62	--	---
ESLO(I,J,2)	F9.6	71	--	---

Table 2.24 Site-Process Combination Data (KEY = 17)

The SIPROC or site-process cards offer processes at sites. There are NSP such cards. Each SIPROC offers one process at one site. An existing process (IEXIS(K) = 1) must be the only process at its site, but any number of new processes may be offered at the same site. Implicit revenues net of haul to the market for recoverables not specified through output coefficients and commodities are entered on this card. A similar arrangement is made for the energy values of recoverables not declared elsewhere. Due to the location of a site, its intended shipments, or for other reasons, it may be necessary to set an upper limit on capacity. The entry on CAPSP(K) and ISPSEG(J,1) through ISPSEG(J,3) affect the operations for the process activities at a candidate site.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	17	17
FILLER	1X			
ISPSI(K)	I3	4	1	NSI
FILLER	1X			
ISPPR(K)	I3	8	1	NPR
FILLER	1X			
CAPSP(K)	F9.3	12	a	---
FILLER	1X			
REVSP(K)	F9.3	22	--	---
FILLER	1X			
ISPSEG(J,1)	I1	32	b	1
ISPSEG(J,2)	I1	33	b	1
FILLER	1X			
ISPSEG(J,3)	I1	34	b	1
FILLER	1X			
IREVSP(K)	I2	36	0	NCC
KREVSP(K)	I2	39	0	NEC
EREVSP(K)	F10.5	41	--	---

a If ICAP(ISPSI(K))=1, then the minimum is 0.0001.
If ICAP(ISPSI(K))=0, then the value is irrelevant.

b ISPSEG(K) will be functional on if K NSEG on the first process card. The sum of functional ISPSEG(K) for a site-process combination must be greater than or equal to 1.

Table 2.25 First Market Card (KEY=18)

Markets are specified by commodity and location. Markets may be either limited or unlimited, and fixed price or declining price. A declining price market is represented by piecewise linear approximation with up to five linear segments per market. An unlimited fixed price market may be recorded here explicitly, or represented implicitly by a net revenue on the SIPROC cards. Energy values of the commodity in the market are also entered on this card. For the two energy categories, the value is assumed to be constant with respect to the marketed quantity, in contrast to the revenue, which may decline with respect to the marketed quantity in up to five segments. No cards of this type should be entered if NMK was coded as 0 on the control card. A limited market has NBDMK(K)=NSGMK(K). An unlimited market has NBDMK(K)=MSGMK(K)-1. A fixed price market has NSGMK(K)=1.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	18	18
FILLER	1X			
IMK(K)	I3	4	1	NMK
FILLER	1X			
NAMMK(J,K)	5A4	8	--	---
FILLER				
LMK(K)	I3	29	1	LSTLO
FILLER	IX			
ICOMK(K)	I3	33	1	NCO
FILLER	1X			
IECMK(1,K)	I3	37	0	NEC
FILLER	1X			
IECMK(2,K)	I3	41	0	NEC
FILLER	1X			
ICCMK(1,K)	I3	45	1	NCC
FILLER	1X			
ICCMK(2,K)	I3	49	1	NCC
FILLER	1X			
NSGMK(K)	I1	53	1	5
NBDMK(K)	I1	54	NSGMK(K)-1	NSGMK(K)
ECMK(1,K)	F10.5	55	--	---
ECMK(2,K)	F10.5	65	--	---

Table 2.26 Second Market Card (KEY=19)

This card allows the user to trace a declining price market structure setting up to five cumulative bounds above which the revenue per unit is reduced. This card must be entered even if the market is unlimited. In such a case, the only entries are KEY and KEMP; they serve as placeholders. No cards of this type should be entered if NMK was coded as 0 on the control card.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	19	19
BDMKA(1,K)	F7.2	3		
BDMKA(2,K)	F7.2	10		
BDMKA(3,K)	F7.2	17	a	
BDMKA(4,K)	FY.2	24		
BDMKA(5,K)	F7.2	31		
FILLER	36X			
KEMP	I3	74	1	NMK

a BDMKA(J,K) > BDMKA(J,K-1) + 0.0001

Table 2.27 Third Market Card (KEY=20)

The two sets of segments CFMKA(J,K) and CFMKB(J,K) allow for revenue sharing arrangements. The user should enter NSGMK market revenue coefficients for each set. No cards of this type should be entered if NMK was coded as 0 on the control card.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	20	20
CFMKA(1,K)	F7.3	3	0.0	--
CFMKA(2,K)	F7.3	10	a	--
CFMKA(3,K)	F7.3	17	a	--
CFMKA(4,K)	F7.3	24	a	--
CFMKA(5,K)	F7.3	31	a	--
CFMKB(1,K)	F7.3	38	0.0	--
CFMKB(2,K)	F7.3	45	b	--
CFMKB(3,K)	F7.3	52	b	--
CFMKB(4,K)	F7.3	59	b	--
CFMKB(5,K)	F7.3	66	b	--
FILLER	1X			
KEMP	I3	74	1	NMK

a CFMKA(J,K) < CFMKA(J,K-1)
b CFMKB(J,K) < CFMKB(J,K-1)

Table 2.28 Cost Summation Data (KEY=21)

This card permits the user to aggregate data from two or more cost categories for purposes of reporting costs. There are NCS such cards. Cost summation categories may be used to make explicit what portion of the total costs of the plan are being borne by the municipality, by a contractor providing services to the municipality, or represent cash flows into or out of the region. A cost category may be listed in more than one cost summation category without causing double-counting in the system costs. The variable NCCS(K) tells how many cost categories are aggregated into the K^{th} cost summation category.

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	21	21
FILLER	1X			
ICS(K)	I3	4	1	NCS
FILLER	1X			
NAMCS(J,K)	5A4	8	--	---
NCCS(K)	I2	28	1	10
FILLER	1X			
ICCS(1,K)	I3	31	1	NCC
FILLER	1X			
ICCS(2,K)	I3	35	1	NCC
FILLER	1X			
ICCS(3,K)	I3	39	1	NCC
FILLER	1X			
ICCS(4,K)	I3	43	1	NCC
FILLER	1X			
ICCS(5,K)	I3	47	1	NCC
FILLER	1X			
ICCS(6,K)	I3	51	1	NCC
FILLER	1X			
ICCS(7,K)	I3	55	1	NCC
FILLER	1X			
ICCS(8,K)	I3	59	1	NCC
FILLER	1X			
ICCS(9,K)	I3	63	1	NCC
FILLER	1X			
ICCS(10,K)	I3	67	1	NCC

Table 2.29 Transportation Data (KEY=22)

These cards specify transportation linkages. There are three types of linkages: (1) source-to-site; (2) site-to-site; and (3) site-to-market. The type of linkage is coded into ITTYP(K).

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	22	22
FILLER	1X			
ITTYP(K)	I1	6	1	3
FILLER	1X			
ITRCO(K)	I3	7	1	a
FILLER	1X			
ITROG(K)[b]	I3	10	1	LSTLO
FILLER	1X			
ITRSM(K)	I3	14	1	c

[a]If ITTYP(K)=1, the maximum value is NSU; otherwise, the maximum value is NCO.

[b]Not checked if ITTYP(K)=1.

[c]If ITTYP(K)=1 or 2, the maximum value is NSI; if ITTYP(K)=3, the maximum value is NMK.

Table 2.30 End Card for Transportation File (KEY=23)

Variable Name	Variable Specification	First Column	Minimum Value	Maximum Value
KEY	I2	1	23	23

Each source of waste generation, transfer or processing site, disposal site, and market is identified by a three-digit location code.[1] Two point-to-point distance files are input in either a triangular or square matrix format, one for packers and one for vans. The files establish a measure of distance between each pair of three-digit location codes in miles, minutes, or cost. These distance files are used by RRPLAN to calculate haul costs for various kinds of transportation activities. The packer distance file is read on unit 7 whereas the van distance file is read on unit 8. Each file must be fully sorted by major record and within a major record by minor record. In a triangular file, the major record is the higher numbered location and the minor record is the lower numbered location. In a square file, the major record is the destination location number and the minor record is the origin location number. Setting the control variable ISQR to zero indicates that triangular distance files are to be input whereas a value of one indicates that square files are to be input.

Three different kinds of expansion are performed on the transportation file by RRPLAN; they are: (1) location expansion, (2) source separation expansion; and (3) haul cost expansion. The file itself, as entered on card type 22, is skeletal. In the first expansion, RRPLAN extracts location numbers from the source, site, and market cards and adds them to the transportation file for sources and for destination sites and markets. In the second expansion, the category 1 transportation portion of the transportation file (source-to-site shipments) is expanded to provide categories 4, 5, 6, 8, 9, and 11 through 15 in accordance with the source separation options entered as run controls (KMIX, KSEPO, NTRIG) on card 3 and as source controls (IMIXSU(K), ISEPSU(K), ITRIGS(J,K)) on card groups 11.[2] As mentioned earlier, the run controls dominate the source controls. In the third expansion, haul costs are calculated for each entry in the expanded transportation file. This step uses the distance files for packers and for vans, and the haul cost multipliers, CCOCC(1,K), CCOCC(2,K), CCOEC(1,K), and CCOEC(2,K), entered on the commodity category cards.

If the distance in the files are recorded in miles, the haul cost multipliers should be dollars per ton-mile appropriate to the commodity. If the distances are in minutes, the haul cost multipliers should be dollars per ton-minute to the commodity. If the distances are in costs, said costs should be in the form of cost per ton over the distance between the two locations for the haul

[1]No location numbers are required for markets associated with implicit revenues (as declared on a SIPROC card).

[2]The activities corresponding to the expanded transportation file were presented in summary form in table 2.2.

of MSW, by packer for the packer file and by van for the van file. In that case, the haul cost (CCOCC(1,K) + CCOCC(2,K)) should be 1.0 for commodity number 1 (haul of MSW in a packer) and for commodity number 2 (haul of MSW in a van). Other haul costs should be the ratio of the haul cost for that commodity to the haul cost of MSW. The denominator is haul cost of MSW in a van for all other commodities except number 4 which is source separation residue hauled in a packer. Enregy requirements for haul are respectively per mile, per minute, and per dollar of MSW haul cost.

The triangular format distance files are arranged so that the entry for each pair of points appears only once. This is accomplished by arbitrarily defining the lower location number as the origin and the higher destination number as the destination. Exhibits 2.2 and 2.3 are examples of triangular distance files for packers and vans, respectively. Note that all entries are in dollars per ton. The distance tables shown in the exhibits are examples of complete linkage between each destination and all origins with lower location numbers. Although complete linkages will not cause any error if the said linkage was not declared on one of the type 22 transportation data cards, for regions with a large number of locations the files may become unwieldly. The general organization of the files is described in table 2.31.

2.3 COMMONS USED

The program data is made available to the various subroutines in several ways. The most frequently used method is COMMON storage. By using the COMMON storage accessing method, frequently used data may be referenced by the same mnemonic names throughout the program.

The variables in the COMMON storage areas may be user's input, or program defined data which may be character, integer, or floating point. No particular ordering is used in assigning variables to the areas. However, those areas used for user's input are allocated in the same order as the input records.

The COMMON storage areas are briefly identified here. These descriptions present a synopsis of the type of variables stored in each area. A detailed listing of variables in each area follows.

Unlabeled COMMON contains variables that are used by almost every routine in the program. The main logic variables and the matrix vectors generated from user inputs are allocated storage in this COMMON area. This COMMON area is also referred to as "blank" COMMON.

RRPLAN contains variables that are used in edit-checking and sizing the problem.

Table 2.31 Structure of the Packer and Van Distance Files

Card Type	Variable Specification	First Column	Purpose
First	20A4	1	Title[a]
Second[b]			
Third[b]			
Destination (1)[c]	I1	1	Flag[d]
FILLER	1X		
Destination (1)	I3	3	Destination Location No.
FILLER	I1		
Destination (2)	I3	2	First origin
FILLER	1X		
Destination (2)	F6.2	6	First distance
FILLER	2X		
Destination (2)	I3	14	Second origin
FILLER	1X		
Destination (2)	F6.2	18	Second distance
FILLER	2X		
Destination (2)	I3	26	Third origin
FILLER	1X		
Destination (2)	F6.2	30	Third distance
FILLER	2X		
Destination (2)	I3	38	Fourth origin
FILLER	1X		
Destination (2)	F6.2	42	Fourth distance
FILLER	2X		
Destination (2)	I3	50	Fifth origin
FILLER	1X		
Destination (2)	F6.2	54	Fifth distance
FILLER	2X		
Destination (3)	I1	1	Flag[e]

[a]Enter up to 80 alphanumeric characters to identify the distance file; identification should include whether packer or van, and the name of the region.

[b]No entries should be made on these cards.

[c]Enter a block of cards for each destination, with the destination, blocks ordered by destination location numer. Each destination block has one first card (Destination(1), one or more middle cards (Destination(2)), and one last card (Destination (3)). The Destination (2) cards should be used to enter origin/distance pairs, five pairs per card, in order of origin location number, leaving no gaps. The last pair in each destination block should be the diagonal element, origin equals destination, with a value of 0.

[d]The first entry must be coded as 1.

[e]The entry must be coded as 2.

Exhibit 2.2 Typical Distance File for Packers

```
DISTANCE FILE FOR PACKERS IN DOLLARS PER TON


1   1
  1 0.
2
1   2
  1 2.       2 0.
2
1   3
  1 1.5      2 1.4      3 0.
2
1   4
  1 1.5      2 3.       3 3.5      4 0.
2
1   5
  1 1.       2 2.5      3 2.0      4 1.       5 0.
2
1   6
  1 30.      2 32.      3 31.      4 29.      5 30.
  6 0.
2
1   7
  1 15.      2 16.      3 14.      4 18.      5 16.
  6 33.      7 0.
2
1   8
  1 50.      2 51.      3 49.      4 50.      5 50.
  6 38.      7 60.      8 0.
2
1   9
  1 30.      2 32.      3 31.      4 29.      5 30.
  6 0.       7 33.      8 38.      9 0.
2
```

Exhibit 2.3 Typical Distance File for Vans

```
      DISTANCE FILE FOR VANS IN DOLLARS PER TON
      1    1
         1 0.
      2
      1    2
         1 .6         2 0.
      2
      1    3
         1 .5         2 .45        3 0.
      2
      1    4
         1 .5         2 1.         3 1.1        4 0.
      2
      1    5
         1 .3         2 .8         3 .6         4 .3         5 0.
      2
      1    6
         1 10.        2 11.        3 10.5       4 9.5        5 10.
         6 0.
      2
      1    7
         1 5.         2 5.4        3 5.         4 6.         5 5.5
         6 11.        7 0.
      2
      1    8
         1 16.        2 16.2       3 15.8       4 16.        5 16.
         6 13.        7 20.        8 0.
      2
      1    9
         1 10.        2 11.        3 10.        4 9.8        5 10.
         6 0.         7 11.        8 13.        9 0.
      2
```

SBASI contains special variables used to store information from the basis column identification feature in the optimizer.

SWAPCO contains the variables which are used for the linear programming algorithm. The majority of the variables are vectors used for matrix manipulation and iterations. This COMMON area is the largest in the program.

The information on each COMMON area which follows includes for each variable: (1) variable name; (2) dimension (if any); (3) description of contents; (4) possible values; and (5) FORTRAN data type.

Table 2.32 "Unlabeled" COMMON STORAGE

Variable	Contents	Value	Data Type
KOPT	Control for output of optimizer	1,2, or 3	Integer
MUT	Read file number	5	Integer
JEF	Write file number	6	Integer
FILE1[a]	Not used	---	Integer
FILE2[a]	Not used	---	Integer
FILE3[a]	Not used	---	Integer
FILE4[a]	Not used	---	Integer
FILE5[a]	Not used	---	Integer
FILE6[a]	Not used	---	Integer
FILE7[a]	Not used	---	Integer
FILE8[a]	Not used	---	Integer
FILE9[a]	Not used	---	Integer
NFORC	Forcing control for phase 4	1,2	Integer
NPHAS	Last phase of model execution	3,4	Integer
INDSD	Control on steepest decent	2	Integer
FC(450)[b]	Column fixed cost array	Computed	Real
VC(360)[b]	Column variable cost array	Computed	Real
B(90)[b]	Righthand side vector	Computed	Real
BETAO(90)[b]	Old activity levels	Computed	Real
A(8100)[b]	Non-zero elements of "A" matrix	Computed	Real
RNA(90)[b]	Row ID for "A" matrix	1-90	Integer
CNA(450)[b]	Column ID for "A" matrix	1-450	Integer
INDIC(90)[b]	Type of equation	0,1	Integer
BASIS(90)[b]	Basis for optimization	Computed	Integer
ROW(8100)[b]	Matrix row values	1-90	Integer
COL(8100)[b]	Matrix column values	1-450	Integer
M[b]	Number of rows in "A" matrix	Computed 1-90	Integer
N[b]	Number of columns in "A" Matrix	Computed 1-450	Integer
NUMNZ[b]	Number of non-zero elements in the "A" matrix	Computed 1-8100	Integer
MAXM[b]	Maximum number of rows	90	Integer
MAXN[b]	Maximum number of columns	360	Integer
MAXA[b]	Maximum number of non-zero entries in the "A" matrix	8100	Integer

[a]These values were used in an earlier model (WRAP) and are referenced in the optimizer. RRPLAN uses the same optimizer as WRAP. The values are preset in the MAIN program to avoid unintentional errors.

[b]Indicates that this value or dimension may be altered to handle different sized problems.

Table 2.33 "RRPLAN" COMMON Storage

Variable	Contents	Value	Data Type
KEY	KEY as read	—	Integer
KKEY	Proper read KEY	1-23	Integer
JAY	Sequence number read	—	Integer
JJAY	Proper sequence number	—	Integer
RHIGH	Maximum value	—	Real
RLOW	Minimum value	—	Real
RVAL	Value read	—	Real
IHIGH	Maximum value	—	Integer
ILOW	Minimum value	—	Integer
IVAL	Value read	—	Integer
L	Number of transportation activities	1-600	Integer
NTRA	Number of source-to-site links	1-300	Integer
IDCC(50)	Type of cost	0,1	Integer
MAXCOL[a]	Maximum number of columns	360	Integer
MAXROW[a]	Maximum number of rows	90	Integer

[a]Indicates that this value may be altered to handle different sized problems.

Table 2.34 "SBASI" COMMON Storage

Variable	Contents	Value	Data Type
NORIG	Last column before NZ activity	Computed	Integer
NROWOU	BASIS column entry replaced by NZ	Computed	Integer

Table 2.35 "SWAPCO" COMMON STORAGE

Variable	Contents	Value	Data Type
ALPHA(90)[a]	Matrix coefficients for a column	Computed	Real
BEP	Basic accuracy test variable	5.E-9	Real
BEP1	Basic accuracy test variable	5.E-4	Real
EP	Changing accuracy test variable	Computed	Real
EP1	Changing accuracy test variable	Computed	Real
HEP	Forcing test variable set to current EP	Computed	Real
HEP1	Forcing test variable set to current EP1	Computed	Real
TEP	Accuracy test variable	Computed	Real
TEP1	Accuracy test variable	Computed	Real
BETA(90)[a]	Righthand side	Computed	Real
BETAI(90)[a]	Initial righthand side	Computed	Real
ETA(90)[a]	Simplex transformation vector	Computed	Real
NU	Calculated PI product sum	Computed	Real
OLDOB	Objective value	Computed	Real
PI(90)[a]	Cost vector	Computed	Real
PSI	Calculated ALPHA(K) product sum	Computed	Real
PSIOF(360)[a]	Vector of various vector sums	Computed	Real
CST(360)[a]	Algorithm cost vector	Computed	Real
ETAFI(8100)[a]	Expansion matrix vector	Computed	Real
PHASE	Current phase or phase error	0,1,2,3,4	Integer
IBASE	Initial basis set	1	Integer
IETAP	Internal ETA vector pointer	Computed	Integer
IFORCE	Forcing method in use	Computed	Integer
LASTI	Initial last column to leave basis	Computed	Integer
INTR	Save initial number of transformations	Computed	Integer
INVT	Inversion control switch	0,1	Integer
LAST	Current last column to leave basis	Computed	Integer
LASTO	Previous last column	Computed	Integer
MAXTR	Not used	--	Integer
NIT	Counter for number of interactions	Computed	Integer
NTRAN	Counts transformations for a phase	Computed	Integer
OBASI(90)[a]	Solution basis as stored	Computed	Integer
PTR	Row pointer	Computed	Integer
ETAEN	Number of entries in expanded matrix	Computed	Integer

Table 2.35 "SWAPCO" COMMON STORAGE (continued)

Variable	Contents	Value	Data Type
ARTIF	Number of artificials in basis	Computed	Integer
IFASE	Phase to start processing	3	Integer
NPSUM	Expansion counter	Computed	Integer
LIMFI	Maximum expansion size	Computed	Integer
TOLCH	Switch to indicate test for tolerance	0,1	Integer
IBASI(90)[a]	Initial basis	Computed	Integer
COLPN(360)[a]	Column pointer values	1-450	Integer
COLCN(360)[a]	Count of non-zero's in a column	1-90	Integer
BASMR(450)[a]	Vector marking columns into basis	1-450	Integer
NRPVT(90)[a]	Pivot row vector	1-90	Integer
ETAND(8100)[a]	Row pointers for a given column in reverse order	1-90	Integer

[a]Indicates that this value or dimension may be altered to handle different sized problems.

3. OPERATIONAL CHARACTERISTICS

3.1 MODEL OUTPUTS

RRPLAN has two basic types of outputs, normal outputs and error messages. The normal outputs were discussed briefly in section 2.1. A line-by-line description of the output for the case example presented in exhibits 2.1-2.3 may be found in NBS Special Publication 657.

The remainder of this section will deal with error messages. In order to address the everyday problems of incorrect formatting, recording and sequencing, RRPLAN has an elaborate system for edit-checking the values of key input variables. If an error is encountered, a message is printed out to the user which should help to locate and correct the error. There are four basic types of error messages: (1) general errors; (2) linkage errors; (3) process-related errors; and (4) matrix manipulation errors.

General errors are the easiest to find and correct. They are associated with input values which are out of range, cards out of sequence, or sums of values that are out of tolerance. They are likely to be due to typographical errors (e.g., incorrect column alignment) or leaving out (or duplicating) a card. The diagnostics produced, the data output and remedies for such errors follow.

Diagnostic: VALUE OUT OF RANGE
Action: Program terminates
Data Output: KEY, card number of sequence, value on card
Remedy: Consult appendix A for proper range of values for the variable. The KEY and card number should be sufficient to identify on which card the error occurs. Check column alignment.

Diagnostic: SEQUENCE ERROR
Action: Program terminates
Data Output: KEY as read, KEY as should be, card number of sequence
Remedy: Check for missing cards in the previous block of cards. Check the settings on the loop controls given on the first card to see if any are either too low or too high. Check that cards within KEY are in sequence by identification number.

Diagnostic: ERROR SUM OF FREQUENCIES
Action: Program terminates
Data Output: Value of sum
Remedy: Locate negative frequencies as entered on the FREQ(K) cards. All frequencies must be non negative.

Diagnostic: TOTAL OF WEIGHTS
Action: Program terminates
Data Output: Value of sum (used only if LOPT=3: sum of cost weights minus sum of energy weights).
Remedy: Make sure all WTCC(K) weights are positive and all WTEC(K) weights are negative.

Linkage errors occur whenever the transportation network is incorrectly specified. These errors are more subtle than general errors. Hence these errors require greater care in locating and correcting. There are nine types of linkage errors, all but two of which produce output data as aids in troublesheeting. The two linkage errors which do not produce output data are: (1) more than 300 transportation activities; or (2) the lack of a market for source separated paper. The first error should not occur if attention is paid to the sizing issue outlined earlier. The second error can only occur when KSEPO is coded equal to 2 on the third card indicating a local paper market. In the event that the region is served by a local paper market, it will be necessary to define a market which receives source separated paper (commodity category number 3) as its input. Distances from all sources from which pre-separated paper could come should then be recorded in the two distance files. The seven other linkage errors all provide output which should enable the user to locate the file in which the problem is occurring (e.g., ACASE, APKR, or AVAN). The errors can then be located and corrected by following the prescriptions given below. Once the error is corrected, the user should perform one additional check. If the error involved one of the cards read on KEY 22, then each of the two distance files should be examined to determine if the linkage defined in the case study file has a distance recorded in each of the distance files.

Diagnostic: MORE THAN 300 TRANS ENTRIES
Action: Program terminates
Data Output: None
Remedy: Reduce problem size

Diagnostic: MAJOR RECORD FOR LOCATION CANNOT BE FOUND IN DISTANCE FILE
Action: Program terminates
Data Output: Location number, file number
Remedy: Determine if a linkage to or from that location is needed and if so construct a distance table. Check if the destination(s) card is missing.

Diagnostic: LOCATION CANNOT BE FOUND IN DISTANCE FILE
Action: Program terminates
Data Output: Destination location, Origin location, file number
Remedy: Determine if a linkage to or from that pair of locations is needed and if so provide an entry in the appropriate file(s).

Diagnostic: NO MARKET FOR SOURCE SEPARATED PAPER
Action: Program terminates
Data Output: None
Remedy: Only applies when KSEPO=2 indicating a local paper market. Define a location for the local paper market. (The input to a local paper market must be commodity number three.)

Diagnostic: TRANS CARD OUT OF SEQUENCE
Action: Program terminates
Data Output: Card number within those read on KEY 22
Remedy: Scan column 4 of the KEY 22 cards to see where the transportation category type ITTYP(K) reverses itself (e.g., goes from 2 to 1). All 1's should preceed all 2's which should preceed all 3's.

Diagnostic: SOURCE UNLINKED
Action: Program Terminates
Data Output: Source number
Remedy: Insert the source-to-site linkage card(s) within the ITTYP(K)=1 cards read on KEY 22.

Diagnostic: LOCATION COMMODITY IS NOT LINKED
Action: Program terminates
Data Output: Location number, commodity number
Remedy: Check the KEY 15 cards for a process with outputs that is matched to a site at the problem location. One of these output commodities is not being sent to another site or a market.

Diagnostic: INPUT LINKAGE MISSING FOR SITE
Action: Program terminates
Data Output: Site number
Remedy: Insert source-to-site or site-to-site linkage card(s) within either the ITTYP(K)=1 or 2 cards read on KEY 22.

Diagnostic: NO LINKAGE FOR MARKET
Action: Program terminates
Data Output: Market number
Remedy: Insert the site-to-market linkage card(s) within the ITTYP(K) = 3 cards on KEY 22.

Process-related errors are due to inconsistencies between two groups of input cards (e.g., sites and processes). For example, if a site listed its input commodity as MSW and a process which recived a non-MSW commodity was to be located at the site, an error would result. This is because each site is coded as receiving only one commodity. Methods for finding and correcting the errors are summarized below.

Diagnostic: DUPLICATED LANDFILL
Action: Program terminates
Data Output: Location number
Remedy: Define a new location number for one of the landfills or delete one landfill card at the location.

Diagnostic: NO SIPROCS FOR SITE
Action: Program terminates
Data Output: Site number
Remedy: Allocate one of the processes to this site.

Diagnostic: MORE THAN ONE SIPROC FOR SITE WITH EXISTING PROCESS
Action: Program terminates
Data Output: Site number
Remedy: Create a new site at the same location or delete extra process at the site.

Diagnostic: DUPLICATED SIPROC
Action: Program terminates
Data Output: Site number, process number
Remedy: Check site number and process number for an error; one or the other must be changed.

Diagnostic: NO VALID SEGMENTS ON SIPROC
Action: Program terminates
Data Output: SIPROC number
Remedy: Check the values coded into the ISPSEG(J,K), K=1,2,3, on the SIPROC card against the card 14 value of NSEG(J) for the process considered.

Diagnostic: LANDFILL REQUIRED AT LOCATION BY SITE
Action: program terminates
Data Output: Location number, site number
Remedy: Check the values coded into LIACOF(K) on the first process card and ILAND(K) on the site card. If a landfill is necessary, one should be created at the location.

Diagnostic: INPUT COMMODITY MISMATCH, SIPROC
Action: Program terminates
Data Output: Input commodity for the site,input commodity for the process.
Remedy: Either the input commodity for the site or for the process matched up to it on the SIPROC card must be changed. Both commodities must be the same in order for the match to be valid.

Matrix manipulation errors identify a problem encountered within the optimizer. These errors are likely caused by an incorrectly formulated problem. For example, the problem as formulated by the user may have a duplicate row or column in the A matrix which would make inversion impossible. Print switches in the optimizer have been set so that the error messages which follow will not occur normally. This was done to minimize extraneous outputs during site or site-process forcing. If the program terminates prior to the output of the preferred plan, it is an indication of a matrix manipulation error. The print switches can then be turned on to help isolate the problem by setting KSUPP=1.

Diagnostic: ****THIS PROGRAM HALTS BASIS NOT GOOD****
Action: Program terminates
Data Output: None
Remedy: Check application problem and starting basis values. Starting basis is causing infeasible solution.

Diagnostic: THE CURRENT BASIS IS NOT INVERTABLE. REINVERSION WAS PROCEEDING FROM THIS INITIAL BASIS (LIST) TO THIS CURRENT BASIS (LIST) WITH THE ACTIVITIES INDICATED BELOW ALREADY TRANSFORMED (LIST). THE ACTIVITY_____ WITH THE CURRENT FORM (LIST) HAS NO COMPUTATIONALLY FEASIBLE PIVOT ELEMENT.
Action: Program terminates
Data Output: Basis lists
Remedy: Reexamine application problem definition.

Diagnostic: THE CURRENT BASIS IS NOT INVERTABLE. IT HAS A NEGATIVE ACTIVITY LEVEL, BASIS (LIST), LEVEL (LIST).
Action: Program terminates
Data Output: Basis lists
Remedy: Reexamine application problem definition.

Diagnostic: NUMBER OF ELEMENTS IN INITIAL-A-MATRIX, INCLUDING REQUIRED SLACK AND SURPLUS VARIABLES HAS EXCEEDED THE LIMIT ALLOWED FOR STORAGE. PROBLEM TERMINATED DUE TO INITIALIZATION PROCEDURES OVERFLOWING AVAILABLE STORAGE.
Action: Program terminates
Data Output: None
Remedy: Refer to matrix sizing instructions for the method of predetermining number of elements that will be in the A matrix.

Diagnostic: NUMBER OF COLUMNS, INCLUDING REQUIRED SLACK AND SURPLUS VARIABLES, HAS EXCEEDED THE LIMIT ALLOWED FOR STORAGE. PROBLEM TERMINATED DUE TO INITIALIZATION PROCEDURES OVERFLOWING AVAILABLE STORAGE.
Action: Program terminates
Data Output: None
Remedy: Refer to sizing instructions for determining maximum number of columns that will be established for the matrix.

Diagnostic: DUPLICATE ROW AND COLUMN LOCATIONS ENCOUNTERED DURING SORT ROUTINE
Action: Program terminates
Data Output: None
Remedy: Reexamine problem and matrix structuring specifications.

Diagnostic: UNBOUNDED SOLUTION WITH COLUMN_____IN THE BASIS.
Action: Program terminates
Data Output: Column number
Remedy: Reexamine application and starting basis entries.

Diagnostic: MATRIX OVERFLOW
Action: Program terminates
Data Output: None
Remedy: Reexamine application problem. Check sizing procedure then use larger version if necessary.

Diagnostic: NO FEASIBLE SOLUTION
Action: Program terminates
Data Output: None
Remedy: Reexamine application and input variables. Recheck NBS Special Publication 657 for help on preparing inputs.

Diagnostic: ERROR Z COLUMN IN BASIS. PROGRAM HALTS
Action: Program terminates
Data Output: None
Remedy: Recheck application problem and input specifications. Final solution is not feasible.

Diagnostic: ERROR NEW COLUMN TO BE INSERTED. OLD COLUMN =__, NEW COLUMN =__
Action: Program continues. Changes columns in the basis solution to remove negative activity levels.
Data Output: Column numbers
Remedy: Recheck application and starting basis input values.

3.2 MODEL TEST RESULTS

The purpose of this section is twofold. First, it provides a description of activities carried out to insure model portability and summarizes run times experienced. Second, it includes a description of the validation activities used to insure that the model does in fact what it should do (as stated in the documentation). The second issue is included because model builders are becoming more aware of the need for a coherent means of validation. Such activities, especially with regard to third party assessments, are summarized in a series of papers dealing with energy models.[1] The discussion of model validation in the latter part of this section will draw on the information contained in the above-mentioned report.

The issue of model portability was addressed in two ways. First, the model was written in FORTRAN for which a nationally accepted standard exists. RRPLAN was written in FORTRAN 77 and tested extensively against the ANSI standard.[2] The model is designed to be run on any system which: (1) complies with the X3.9 ANSI standard; and (2) can accomodate an intermediate-sized programs. Second, the model was tested independently on two types of hardware with different word lengths and memory capabilities. In all cases the solutions produced by the model were identical. In order to increase the likelihood that the model is portable, a test case with solutions for comparison is included with the source code whenever a request for the model is made.

All preliminary developmental work on the RRPLAN model was done on a Sperry Univac 1100/82. This machine is the NBS mainframe computer; it has a 36-bit word. When the model was run on the Univac; it had a memory requirement of 110,000 words. A wide variety of test problems were run on the Sperry Univac 1100/82. All run times presented in this section are based to these test problems.

---

[1]P.B. Saunders, editor, Selected Assessment Strategies Applied to Short-Term Energy Models, National Bureau of Standards, NBSIR 83-2672, March 1983.

[2]American National Standards Institute, American National Standard Programming Language FORTRAN, ANSI X3.9-1978, New York, 1978.

Three sets of test problems were analyzed to see what effect problem size and the type of forcing selected would have on run time. The results of these tests are summarized in tables 3.1 and 3.2. Table 3.1 provides background information on each test problem. Five basic attributes which are closely related to the nature of the input data are presented first. These attributes are: (1) the number of sources; (2) the number of sites; (3) the number of landfills; (4) the number of site-processes; and (5) the number of markets. These attributes were chosen for their impact on the row and column dimensions of the A matrix.

Two other attributes, a sparseness factor and a complexity factor, are also presented. These attributes were chosen due to their effect on computational difficulty.[1] The sparseness factor is equal to the number of non-zero elements in the A matrix divided by the total number of elements in the A matrix (i.e., the product of the number of rows and columns). Since the algorithm assumes that the A matrix is sparse, lower values of the sparseness factor should improve the efficiency of the algorithm (e.g., due to less sorting and manipulation of row and column pointers). The complexity factor is equal to the ratio of the number of columns containing a non-zero intercept (i.e., a fixed charge) to the total number of columns. The term complexity factor is used because the presence of fixed charges may cause the program to converge to a local optimum. Other things being equal, the greater the number of fixed charges, the greater the likelihood that the program will converge to a local optimum.

The three problem sizes may be thought of as small-intermediate, intermediate-large, and large. The first problem is the test case application discussed in this report and in NBS Special Publication 657. It has 30 rows and 54 columns. The second problem represents a study of an SMSA. It has 62 rows and 188 columns. This study consisted of 17 sources, 12 sites, 4 landfills, and 2 markets. The third problem represents a study of an entire state. This skelatal analysis consisted of 17 sources, 21 sites, 2 landfills, and 5 markets. All three test problems exercised the source separation options.

The run times experienced for each of the three test problems are reported in table 3.2. Results for seven types of forcing are reported. The first case shows the run times experienced when no forcing is used. This approach incorporates the Walker algorithm[2] which uses a modified selection rule governing the selection of the variables to enter and leave the basis. Fixed

---

[1]The values of these factors are indicative of those experienced in carrying out a variety of analytical studies. It is expected that the ranges shown in table 3.1 will bracket most user's application problems.

[2]W.E. Walker, op. cit.

Table 3.1 Background Information on Test Problems

Critical Attribute	Problem Size 30 x 54	62 x 188	86 x 282
Number of sources	4	17	17
Number of sites	4	12	21
Number of landfills	2	4	2
Number of site processes	5	12	30
Number of markets	3	2	5
Sparseness factor	0.094	0.046	0.024
Complexity factor	0.167	0.085	0.206

Table 3.2 Run Times Experienced for Selected Test Problems

Type of forcing used	Problem Size 30 x 54	62 x 188	86 x 282
None	24 sec.	1 min. 4 sec.	4 min. 6 sec.
Site (IN/OUT)	37 sec.	5 min. 10 sec.	--
Site (OUT)	--	4 min 39 sec.	31 min. 37 sec.
Site-Process (IN/OUT)	40 sec.	5 min. 18 sec.	--
Site-Process (OUT)	--	4 min. 41 sec.	45 min. 21 sec.
Single column	31 sec.	3 min. 20 sec.	--
Double column	2 min. 11 sec.	>20 min.	--

charges are considered in this approach. Data on both types of site forcing and site-processing forcing are then reported. If KFOUT was other than 9 (see Appendix A), then each site (site-process) will be forced both in and out of the basis. If KFOUT was coded as 9 on the control card, then each site (site-process) will be forced out of the basis; no attempt to force in will be made. The last two rows of table 3.2 show run times for single and double column forcing. These were the only types of forcing available in WRAP and RAMP.

A review of table 3.2 reveals that double column forcing is much more demanding in CPU time than any of the other methods. Since it is unlikely that double column forcing will find a solution which is demonstrably better than site or site-process forcing, its use should be discouraged. Furthermore, although single column forcing has a slight advantage in measured CPU time over both types of site and site-process forcing, the possibility that the program can not move away from a local optimum which is well linked can not be ignored. The slight increase in computational expense associated with site and site-process forcing therefore seems quite reasonable. Finally, the model was able to successfully generate solutions for very large problems in times well within the range of CPU times not uncommonly experienced at computer facilities such as NBS.

Validation Activities

The validation of a complex model aims at demonstrating that the model bears a close resemblance to the physical system. The validation process is, in reality, three separate tasks: (1) technical validity; (2) operational validity; and (3) dynamic validity. Since program documentation provides a basis from which the validity of the model can be assessed, it will be discussed prior to the three major tasks of the validation process.

Documentation

From a model user's point of view, documentation (the written description of the model) is essential if the model is to be useable, useful, and used. Since the abstract model is a mathematical representation whereas the operational RRPLAN model appears as a computer code, it is necessary to verify that there exists a unique relationship between the abstract (mathematical) model and the operational model. The relationship between the abstract and operational model can best be understood through reference to section 2.1. Documentation also requires that portability be established (not just that the program can be run on a variety of machines, but that it produces the desired result). Portability in that sense was the subject of the earlier discussion. Documentation also serves to explain all relevant relationships between inputs, outputs and analysis. The model's documentation also should provide some measure of user friendliness. Complex models often involve subtle techniques which, in the absence of a buffer between the user and the model, could cause frustration and lead to a highly inefficient use of the model. Documentation and an executive code (e.g., extensive edit-checking and message generating capabilities) should serve to shield the user from unnecessary detail without withholding any information which is essential to confidently use the model. The RRPLAN model addresses this issue through reference to two NBS reports: (1) Special Publication 657; and (2) NBSIR 83-2745. Each report discusses a particular aspect of the model. These aspects are (1) management and application; and (2) operation and maintenance. Each report is designed to be self contained. Where necessary, connections between the reports are given. The source code provides ample information to users so they can find and correct errors in their field. The source code also contains comments within each routine, should it become necessary to make changes to the source code.

Technical Validity

Technical validity requires the identification of all model assumptions, including those dealing with data requirements and sources. As a first step, one should identify all stated and implied assumptions, all decision variables, and any hypothesized relationship between variables. This step sheds light on the correspondence between the model and the real world phenomena it attempts to explain (see figure 2.1). Three types of assumptions may be readily defined. First, the mathematical assumptions include its functional form and the continuity of its relationships (e.g., economies of scale in processing). A second type, content assumptions, define all model terms and variables. They should also define the scope and limitations of the model. (This topic is discussed in NBS Special Publication 657.) The final type, causal assumptions, are concerned with the assumed or hypothesized relationships between terms and variables (e.g., the way the A matrix is constructed).

Ideally, one would like to build a model which would produce true conclusions whenever all of the assumptions are true. To translate such abstract concepts into a form which is concrete and testable, it is necessary to:

(i) determine if the model's calculations are correct and accurate;

(ii) analyze if the logical flow of data and intermediate results are correct and consistent; and

(iii) ensure that variables and relationships have not been omitted.

Substantial portions of both NBS reports address these issues.

Operational Validity

Operational validity is concerned with whether or not the model can produce bad answers for proper ranges of parameter values (i.e., the model should be robust in that a user would find it difficult to make the model yield (in terms of the decision maker) an ostensibly wrong anwer). Sensitivity analysis is related to, but distinct from, robustness. This technique seeks to systematically vary the values of the model parameters to determine how much (i.e., how sensitive) the solution changes. This issue is discussed in detail in Chaper 5 of NBS Special Publication 657.

The last aspect of operational validity and the most difficult is implementation validity. Implementation validity is concerned with the extent to which the real world system being modeled will respond in a manner indicated by the recommended solution. This task is difficult because if a decision maker knew how the system would respond to a given change in a parameter or decision variable, there would be considerably less need for a model. Implementation validity has been addressed in an informal matter. Since the model has not been released to the general public (at the time of this writing), it is impossible to state definitively whether its results will merely vindicate the decision maker's judgment or provide genuine insight.

Dynamic Validity

Dynamic validity is concerned with determining how the model will be maintained during its life cycle so it will continue to be an acceptable representation of the real system. The two aspects associated with dynamic validity are updating and review. In updating, the person incorporating the changes needs to be satisfied that the model developers have established a procedure by which information is collected and analyzed to determine if and when model parameters or model structure need to be changed.[1] It is also important that a process exists by which such changes can be incorporated into the model and disseminated to users (e.g., through tape transfers). A regular schedule for reviewing the success or failure of the model during its life cycle is also necessary. These reviews should be carried out regularly and should focus on documenting any systematic divergences between the solution predicted and the actual outcomes. The implications and means of accomplishing any proposed model changes should also be commented on.

3.3 SOFTWARE EXCHANGE

The preferred transfer medium for the model is 9-track magnetic tape. All tapes provided by NBS will be written in either ASCII or EBCDIC. The request for the source code should specify which bit configuration is required for the user's operating system. All tapes provided by NBS will be unlabeled. The 2400 foot real will be recorded with a density of 1600 frames per inch (FPI). Since one of the files recorded on the tape is the test case output, a logical record length of 132 characters is used throughout. A fixed block length of 1320 characters is also used. Table 3.3 summarizes the information on magnetic tapes.

---

[1]The decision to develop RRPLAN was in part a recognition that neither WRAP nor RAMP were adequate to efficiently address the issues at hand.

Table 3.3 Specifications for Transfer of 9-Track Unlabeled Magnetic Tapes

Characteristic	Specification
Bit configuration	ASCII <u>or</u> EDCDIC
Density	1600 FPI
Logical record length	132 characters
Block length	1320 characters

Software exists at NBS for producing either ASCII or EBCDIC punched card code. NBS will therefore provide a deck of punched cards if such a request is made. Since the deck will consist of only the source code and data files discussed below, punched cards should only be used when no compatible magnetic tape equipment is available on the user's operating system.

Whenever a request for the RRPLAN source code is made, the tape sent to the requestor will contain six files. These files contain: (1) background information; (2) the RRPLAN source code; (3) the ACASE file; (4) the APKR file; (5) the AVAN file; and (6) the test case output file.

The first file contains a table of contents for the tape. It describes what is in each file and how the five remaining files relate to each other. It also contains information on what portions of the model have been updated and the reason for the change since the publication of NBS Special Publication 657 and NBSIR 83-2745.

The second file contains the RRPLAN source code. The programmer is free to choose which compiler options are to be used. It may be advisable to exercise the code optimization option if such an option is available on the host system. Since RRPLAN has almost 90,000 data bank words, it may be necessary to exercise an option which generates code which allows data bank addresses to exceed a prespecified amount.

The third, fourth and fifth files are the test case user data files. They contain a known correct set of data for use in initial testing on the host system. Once these tests have been performed, it is recommended that these files be provided to users so that they can check out their knowledge of the model with a known set of input data. Similarly, users can duplicate the data files and through an editor introduce errors to test their troubleshooting skills. These files are thoroughly described in NBS Special Publication 657 where they are used to illustrate how the model operates. Guidelines for constructing the files based on the RRPLAN worksheets are also given in NBS Special Publication 657.

The last file contains a complete set of solutions to the problem defined by the test case data. The primary purpose of this file is to insure model validity across operating systems. That is, it can be used as a reference point to determine what effects, if any, a different operating system has on the model's results.

APPENDIX A DICTIONARY OF TERMS

A.1 DICTIONARY OF INPUT VARIABLES

This section provides an in-depth summary of all input terms. The dictionary is arranged in alphabetical order by variable name. Each array dimension is defined and, where applicable, references are made to the effects caused by variable settings in other parts of the input deck. In all but a few cases, a mnemonic, which should help associate the variable with its purpose, is given. The key letters in the related name are capitalized and underlined. The specification declared in the FORMAT statement through which the variable is read in is then given. The read KEY, card type, and first column on the card available for this variable follow immediately. If the variable has a range, the range is given. Each variable is then cross referenced to a worksheet and a specific question. A complete description of the worksheets is given in NBS Special Publication 657. The variable is then defined; any additional information is also noted at that time.

With regard to the worksheet-question responses in the glossary, there are three cases which serve as exceptions. The first relates to the location number. The question on the worksheet refers to the location as a street address or intersection. Once all worksheets have been filled out, each address or intersection is assigned a number. It is this number which is used for all distance calculations. If the term "Note A" is used, it is an indication that the user must assign the location number and construct the distance table from the data on the worksheets. The second exception relates to the end points of do loops. This case is designated by "Note B". Variables falling into this category correspond to counts on the number of times a particular worksheet was filled out. Each of the variables in this category also serve to define an end point for a sequence of sources, sites, markets, etc. Therefore, when the worksheets are counted, they should also be assigned a sequence number. The third exception relates to run controls. This case is designated as "Note C". Six other variables are also grouped into this category. They are: the distance file format control, ISQR; the dedicted transfer station control, IXDED; and the four fixed and variable cost drives, IFDRV, IVDRV, FDRIV, VDRIV.

It was mentioned in chapter 2 that several variables were read according to a FORMAT specification which differed from the "Variable Specification" given in NBS Special Publication 657. So that technical specialists who may have to modify the source code and/or work with users will be aware of these differences, the specification and column qualifiers of all such variables are marked with an asterisk (*). Any differences between this report and NBS Special Publication 657 can therefore be reconciled easily through reference to this appendix.

In the discussion which follows, it is sometimes necessary to refer to a specific element within an array. So that this may be done efficiently, a labeling convention has been adopted and is used throughout the appendix. If the array is one-dimensional, the index is listed as K (i.e., ARRAY(K)). If the array is two-dimensional, the indices are J and K, respectively (i.e., ARRAY(J,K)). If the array is three-dimensional, the indices are I, J and K, respectively (i.e., ARRAY(I,J,K)).

ALLO

Mnemonic: ALLOcation
Specification: F4.3 KEY: 3 First Column: 73
Minimum: 0.0 Maximum: 1.0
Worksheet: I Question: 9.C
Definition: Allocation of paper processing cost to actual paper processed as waste (ALLO) versus potential paper processed as waste (1-ALLO).

AREAD(5)

Specification: F6.2
Worksheet: Note A
Definition: The Kth distance on the card read in either the packer or van distance file.

ATRIG(5)

Mnemonic: Average TRIGger price
Specification: F5.2 KEY: 6 First Column: 52
Minimum: ATRIG(K-1) + 0.0001(ATRIG(K)) K>1
Worksheet: II Question: 4.a
Definition: Average price of paper sold in the market for the Kth trigger price.

AVLA(30)

Mnemonic: AVailable LAnd
Specification: F10.3 KEY: 13 First Column: 29
Minimum: 0.001
Worksheet: VIII Question: 3
Definition: Land available at the Kth landfill.

BDMKA(5,30)

Mnemonic: BounD MArKet
Specification: F7.2 KEY: 19 First Column: 3
Minimum: BDMKA(J-1,K) + 0.0001(BDMKA(J,K))
Worksheet: XI Question: 4.a
Definition: Bound for the Jth segment of the Kth market.

BOTTOM

Mnemonic: BOTTOM price
Specification: F5.2 KEY: 7 First Column: 11
Worksheet: II Question: 2.a
Definition: Lower limit of first (lowest price) interval in dollars per ton in the paper market price distribution.

BTMSW

Mnemonic: BTu content of MSW
Specification: F10.5 KEY: 5 First Column: 39
Minimum: 0.05
Worksheet: I Question: 5.a
Definition: Millions of BTUs per ton of MSW.

BTPAP

Mnemonic: BTu content of PAPer
Specification: F6.3 KEY: 3 First Column: 29
Minimum: 0.0
Worksheet: I Question: 5.b
Definition: Millions of BTUs per ton of source separated paper when processed as solid waste.

BTRES — Mnemonic: BTu content of source separation RESidue
Specification: F6.3 KEY: 3 First Column: 23
Minimum: 0.0
Worksheet: I Question: 5.c
Definition: Millions of BTUs per ton of source separation residue.

BTRIG(5) — Mnemonic: percent Burn by TRIGger price
Specification: F4.1 KEY: 6 First Column: 32
Minimum: BTRIG(K-1) + 0.0001(BTRIG(K)) K>1
Worksheet: II Question: 4.b
Definition: Percent of source separated paper burned for the Kth trigger price.

CAPSP(80) — Mnemonic: CAPacity for a Site-Process
Specification: F9.3 KEY: 17 First Column: 30
Minimum: 0.0001 if the site is coded as capacitated
Worksheet: X Question: 3
Definition: Capacity the process at the site under consideration assuming the process is the only one at that site.

CCOCC(2,20) — Mnemonic: haul Cost-COmmodity-Cost Category
Specification: F8.3 KEY: 10 First Column: 30
Minimum: 0.00001 if J=1 Maximum: 1000.0
Worksheet: V Question: 2
Definition: The Jth haul cost factor for the Kth commodity category (CCOCC(J,K)).

CCOEC(2,20) — Mnemonic: haul Cost-COmmodity-Energy Category
Specification: F8.3 KEY: 10 First Column: 50
Worksheet: V Question: 3
Definition: The Jth haul cost factor for the Kth energy commodity (CCOEC(J,K)).

CCPREP(50) — Mnemonic: Constant Cost for site PREParation
Specification: F10.3 KEY: 12 First Column: 44
Worksheet: VII Question: 5.b
Definition: Site preparation cost for the Kth site in thousands of dollars.

CEPREP(50) — Mnemonic: Constant Energy for site PREParation
Specification: F10.3 KEY: 12 First Column: 56
Worksheet: VII Question: 5.d
Definition: Site preparation energy requirement for the Kth site in full energy units.

CFMKA(5,30) — Mnemonic: CoeFficient MarKet A
Specification: F7.3 KEY: 20 First Column: 3
Minimum: 0.0 Maximum: CFMKA(J-1,K)
Worksheet: XI Question: 4.d
Definition: Jth revenue coefficient for the Kth market (slope of the total revenue function for the Jth segment) for the first cost category.

CFMKB(5,30)

Mnemonic: CoeFficient MarKet B
Specification: F7.3 KEY: 20 First Column: 38
Minimum: 0.0 Maximum: CFMLB(J-1,K)
Worksheet: XI Question: 4.d
Definition: Jth revenue coefficient for the Kth market (slope of the total revenue function for the Jth segment) for the second cost category.

CHCA(60)

Mnemonic: Cost for Hauling CAns
Specification: F5.2 KEY: 11 First Column: 64
Worksheet: VI Question: 6.e
Definition: Cost per ton to haul cans from source K to the market.

CHGL(60)

Mnemonic: Cost for Hauling GLass
Specification: F5.2 KEY: 11 First Column: 59
Worksheet: VI Question: 6.d
Definition: Cost per ton to haul glass from source K to the market.

CHPA(60)

Mnemonic: Cost for Hauling PAper
Specification: F5.6 KEY: 11 First Column: 54
Worksheet: VI Question: 6.c
Definition: Cost per ton to haul paper from source K to the market.

CINT(30,3,2)

Mnemonic: Cost INTercept
Specification: F9.2 KEY: 16 First Column: 8
Worksheet: IX Question: 6.c
Definition: Intercept (fixed charge) cost in thousands of dollars for the Ith process, Jth segment, Kth cost type.

COF(7,30)

Mnemonic: output COeFficient
Specification: F8.5 KEY: 15 First Column: 10
Worksheet: IX Question: 5
Definition: The Jth output coefficient (output per unit of incoming commodity) for the Kth process.

CSLO(30,3,2)

Mnemonic: Cost SLOpe
Specification: F9.6 KEY: 16 First Column: 44
Worksheet: IX Question: 6.c
Definition: Slope (variable) cost in dollars per ton for the Ith process, Jth segment, Kth cost type.

CSPA

Mnemonic: Cost SeParation A
Specification: F5.3 KEY: 3 First Column: 49
Minimum: 0.0 Maximum: 1000.0
Worksheet: I Question: 8.a
Definition: Total cost of collection in dollars per ton of unseparated MSW.

CSPB

Mnemonic: Cost SeParation B
Specification: F5.3 KEY: 3 First Column: 56
Minimum: 0.0 Maximum: 1000.0
Worksheet: I Question: 8.b
Definition: Incremental cost of source separation in dollars per preseparated ton.

DIFLAT(50)

Mnemonic: DIfferential inFLATion rate
Specification: F5.3 KEY: 8 First Column: 39
Worksheet: III Question: 4
Definition: Differential annual inflation rate in percent for the Kth cost category.

DISCO

Mnemonic: DISCOunt rate
Specification: F8.4 KEY: 1 First Column: 19
Minimum: 0.0 Maximum: 50.0
Worksheet: I Question: 2
Definition: Annual discount rate in percent.

ECANM

Mnemonic: Energy value of CANs at the Market
Specification: F10.5 KEY: 5 First Column: 27
Worksheet: I Question: 10.c
Definition: Energy value of cans per ton in the market when delivered there.

ECMK(2,30)

Mnemonic: Energy Coefficient MarKet
Specification: F10.5 KEY: 18 First Column: 55
Worksheet: XI Question: 5
Definition: Jth energy market value per ton when marketed in the Kth market.

EGLAM

Mnemonic: Energy value of GLAss at the Market
Specification: F10.5 KEY: 5 First Column: 15
Worksheet: I Question: 10.b
Definition: Energy value of glass per ton in the market when delivered there.

EINT(30,3,2)

Mnemonic: Energy INTercept
Specification: F9.2 KEY: 16 First Column: 26
Worksheet: IX Question: 7.b
Definition: Intercept energy in full energy units for the Ith process, Jth segment, Kth energy type.

EPAPM

Mnemonic: Energy value of PAPer at the Market
Specification: F10.5 KEY: 5 First Column: 3
Worksheet: I Question: 10.a
Definition: Energy value of paper per ton in the market when delivered there.

EREVSP(80) Mnemonic: Energy REVenue SiProc
Specification: F10.5 KEY: 17 First Column: 41
Worksheet: X Question: 5.d
Definition: Energy of recoverables not entered into the commodity file (implicit revenues) net of energy requirement of haul to the market. In the back end, EREVSP(K) is redefined as the total implicit revenue per average year for the Kth SIPROC in undiscounted cost.

ESLO(30,3,2) Mnemonic: Energy SLOpe
Specification: F9.6 KEY: 16 First Column: 62
Worksheet: IX Question: 7.b
Definition: Energy slope for the Ith process, Jth segment, Kth energy type.

ESPA Mnemonic: Energy SeParation A
Specification: F10.3 KEY: 4 First Column: 6
Worksheet: I Question: 11.a
Definition: Net energy requirement for collection of unseparated waste per ton.

ESPB Mnemonic: Energy SeParation B
Specification: F10.3 KEY: 4 First Column: 18
Worksheet: I Question: 11.b
Definition: Net incremental energy requirement for source separation per ton of preseparated waste.

FDRIV Mnemonic: Fixed cost DRIVer
Specification: F5.2 KEY: 5 First Column: 53
Worksheet: Note C
Definition: Fixed cost value used for driving into the solution an activity not in the solution when either site or site-process forcing is used.

FLATO Mnemonic: inFLATion rate
Specification: F8.4 KEY: 1 First Column: 31
Minimum: 0.0 Maximum: 50.0
Worksheet: I Question: 3
Definition: General annual inflation rate in percent.

FREQ(200) Mnemonic: FREQuency
Specification: F5.3 KEY: First Column: 1
Worksheet: II Question: 3
Definition: The Kth frequency; frequency for the Kth interval in the paper market price distribution function.

IBTU Mnemonic: Identity BTU
Specification: I2 KEY: 5 First Column: 49
Worksheet: I Question: 13.d
Definition: Energy category for BTU differentiation among MSW, paper and source separation residue.

ICANM

Mnemonic: Identity CANs Market
Specification: I2 KEY: 5 First Column: 37
Worksheet: I Question: 13.c
Definition: Energy category for cans in the market.

ICAP(50)

Mnemonic: Identity CAPacity
Specification: I1 KEY: 12 First Column: 37
Worksheet: VII Question: 6.a
Definition: Capacity code for the Kth site: 0 if the site is uncapacitated; 1 if the site is capacitated.

ICC(50)

Mnemonic: Identify Cost Category
Specification: I3* KEY: 8 First Column: 4*
Worksheet: III Question: 1.b
Definition: Identification number of the Kth cost category.

ICCCAN

Mnemonic: Identity Cost Category CANs
Specification: I2 Key: 3 First Column: 47
Worksheet: I Question: 14.c
Definition: Cost category number for cans in the market.

ICCGLA

Mnemonic: Identity Cost Category GLAss
Specification: I2 Key: 3 First Column: 45
Worksheet: I Question: 14.b
Definition: Cost category number for glass in the market.

ICCMK(2,30)

Mnemonic: Identity Cost Category MarKet
Specification: I2 Key: 18 First Column: 46
Worksheet: XI Question: 4
Definition: Jth cost (i.e., revenue) for the Kth market.

ICCPAP

Mnemonic: Identity Cost Category PAPer
Specification: I2 Key: 3 First Column: 67
Worksheet: I Question: 14.a
Definition: Cost category number for national paper market. Not used for explicit paper markets under KSEPO = 2.

ICCS(10,10)

Mnemonic: Identity Cost Category Summation
Specification: I3* Key: 21 First Column: 31*
Worksheet: XII Question: 2
Definition: The identity of the Jth cost category in the Kth cost summation.

ICO(20)

Mnemonic: Identity COmmodity
Specification: I3* Key: 10 First Column: 4*
Worksheet: V Question: 1.b
Definition: Identification number for the Kth commodity.

ICOCC(2,20) Mnemonic: Identity COmmodity Cost Category
Specification: I2 Key: 10 First Column: 28
Worksheet: V Question: 2
Definition: Cost category for the Jth haul cost factor for the Kth commodity.

ICOEC(2,20) Mnemonic: Identity COmmodity Energy Category
Specification: I2 Key: 10 First Column: 48
Worksheet: V Question: 3
Definition: Energy category for the Jth energy level factor for the Kth commodity.

ICOF(7,30) Mnemonic: Identity COeFficient
Specification: I2 Key: 15 First Column: 8
Worksheet: IX Question: 5
Definition: The commodity category number for the Jth coefficient of the Kth process.

ICOMK(30) Mnemonic: Identity COmmodity MarKet
Specification: I3* Key: 18 First Column: 33*
Worksheet: XI Question: 3
Definition: The commodity category number for the Kth market.

ICPR(30,2) Mnemonic: Identity Cost PRocess
Specification: I2 Key: 14 First Column: 44
Worksheet: IX Question: 6
Definition: The Kth cost category for the Jth process.

ICPREP(50) Mnemonic: Identity Cost site PREParation
Specification: I2 Key: 12 First Column: 42
Worksheet: VII Question: 5.a
Definition: The cost category number for the Kth site's preparation costs.

ICS(10) Mnemonic: Identity Cost Summation
Specification: I3* Key: 21 First Column: 4*
Worksheet: XII Question: 1.b
Definition: Identification number of the Kth cost summation category.

IDCC(50) Mnemonic: IDentity code Capital Cost
Specification: I2 Key: 21 First Column: 5
Worksheet: III Question: 2
Definition: Cost type code: 0 is operating cost; 1 is capital cost.

IEC(20) Mnemonic: Identity Energy Category
Specification: I3* Key: 9 First Column: 4*
Worksheet: IV Question: 1.b
Definition: The identification number of the Kth energy category.

IECMK(2,30)

Mnemonic: Identity Energy Category MarKet
Specification: I3* Key: 18 First Column: 37*
Worksheet: XI Question: 5
Definition: The identification number of the Jth energy category for the Kth market.

IEPR(30,2)

Mnemonic: Identity Energy PRocess
Specification: I2 Key: 14 First Column: 48
Worksheet: IX Question: 7
Definition: The identification number of the Kth energy category for the Jth process.

IEPREP(50)

Mnemonic: Identity Energy PREParation
Specification: I2 Key: 12 First Column: 54
Worksheet: VII Question: 5.c
Definition: The energy category number for the Kth site's preparation energy.

IEXIS(30)

Mnemonic: Identity EXIStence
Specification: I1 Key: 14 First Column: 31
Worksheet: IX Question: 3
Definition: Existing process code: 0 is new; 1 is existing.

IFDRV

Mnemonic: Identity Fixed cost DRiVer
Specification: I1 Key: 5 First Column: 52
Worksheet: Note C
Definition: Driver control for fixed costs. If 0, fixed costs will be set to zero to "urge in" an activity in site and site-process forcing; if 1, FDRIV will be used as the fixed cost to "urge in" during site and site-process forcing.

IFIRST

Mnemonic: Identity FIRST year
Specification: I4 Key: 1 First Column: 15
Worksheet: I Question: 1.a
Definition: The identity of the first year in the run.

IGLAM

Mnemonic: Identity GLAss Market
Specification: I2 Key: 5 First Column: 25
Worksheet: I Question: 13.b
Definition: The energy category number for glass when delivered to the market.

IHCSEP(60)

Mnemonic: Identity Haul Cost SEParation
Specification: I2 Key: 11 First Column: 50
Worksheet: VI Question: 6.a
Definition: The cost category number for haul of paper, glass and cans to the market for the Kth source.

IHESEP(60) Mnemonic: Identity Haul Energy SEParation
Specification: I2 Key: 11 First Column: 52
Worksheet: VI Question: 6.b
Definition: The energy category number for haul of paper, glass and cans to the market for the Kth source.

ILACOF(30) Mnemonic: Identity code LAnd COeFficient
Specification: I1 Key: 14 First Column: 34
Worksheet: IX Question: 9
Definition: Landfill coefficient code: 0 if process does not use on-site landfill; 1 if it does.

ILAND(50) Mnemonic: Identity code LAND
Specification: I1 Key: 12 First Column: 38
Worksheet: VII Question: 6.b
Definition: Landfill code: 0 if no landfill is available at the Kth site; 1 if a landfill is available.

IMIXSU(60) Mnemonic: Identity MIXed now SoUrce
Specification: I1 Key: 11 First Column: 43
Worksheet: VI Question: 5.a
Definition: Code for MSW option (no source separation) at source K: if 0, option is not offered; if 1 option is.

IMK(30) Mnemonic: Identity MarKet
Specification: I3* Key: 18 First Column: 4*
Worksheet: XI Question: 1.b
Definition: Identification number of Kth market.

INCO(50) Mnemonic: INput COmmodity
Specification: I3* Key: 12 First Column: 33*
Worksheet: VII Question: 4
Definition: The commodity number of the input to the Kth site.

INCOP(30) Mnemonic: INput COmmodity Process
Specification: I3* Key: 14 First Column: 28*
Worksheet: IX Question: 2
Definition: The commodity number of the input to the Kth process.

INPAP Mnemonic: INput mode for PAPer
Specification: I1 Key: 3 First Column: 66
Worksheet: II Question: 4
Definition: Input mode for paper price distribution: if 0, input trigger prices, percents of burn, and average prices; if 1, input trigger prices and a paper market price distribution function.

IPAPM Mnemonic: Identity PAPer Market
Specification: I2 Key: 5 First Column: 13
Worksheet: I Question: 13.a
Definition: The energy category for paper in the market.

IPR(30)

Mnemonic: Identity PRocess
Specification: I3* Key: 14 First Column: 4*
Worksheet: IX Question: 1.b
Definition: The identification number of the Kth process.

IREAD(5)

Mnemonic: Identity READ
Specification: I3
Worksheet: Note A
Definition: The Kth minor location number on a card.

IREVSP(80)

Mnemonic: Identity REVenue Site-Process
Specification: I2 Key: 17 First Column: 36
Worksheet: X Question: 5.a
Definition: The cost category number for accumulating implicit revenues associated with the Kth site-process combination.

ISEPSU(60)

Mnemonic: Identity SEParation SoUrce
Specification: I1 Key: 11 First Column: 44
Worksheet: VI Question: 5.b
Definition: Code for unconditional source separation at source K: if 0, option is not offered; if 1 or 2, national or local paper option as defined by KSEPO is offered.

ISI(50)

Mnemonic: Identity SIte
Specification: I3* Key: 12 First Column: 4*
Worksheet: VII Question: 1.b
Definition: Identification number of the Kth site.

ISIREQ(50)

Mnemonic: Identity SIte REQuest
Specification: I1 Key: 12 First Column: 3
Worksheet: VII Question: 2
Definition: Site routing request code: if 0, routing not requested; if greater than zero, ISIREQ(K) levels of routing are requested for site K, if site K is in the solution.

ISPA

Mnemonic: Identity SeParation A
Specification: I2 Key: 3 First Column: 54
Worksheet: I Question: 14.d
Definition: Cost category number for total cost of collection, CSPA.

ISPB

Mnemonic: Identity SeParation B
Specification: I1 Key: 3 First Column: 61
Worksheet: I Question: 14.e
Definition: Cost category number for incremental cost of source separation, CSPB.

ISPPR(80)

Mnemonic: Identity Site-Process, PRocess
Specification: I3* Key: 17 First Column: 8*
Worksheet: X Question: 2.b
Definition: The process number of the Kth site-process combination.

ISPSEG(80,3)

Mnemonic: Identity Site-Process SEGment
Specification: I1 Key: 17 First Column: 32
Worksheet: X Question: 4
Definition: Kth linear segment code for the Jth site-process combination: if 0, the segment is not offered; if 1, the segment is offered.

ISPSI(80)

Mnemonic: Identity Site-Process, SIte
Specification: I3* Key: 17 First Column: 4*
Worksheet: X Question: 1.b
Definition: The site number of the Kth site-process combination.

ISQR

Mnemonic: Identity SQuaRe
Specification: I1 Key: 1 First Column: 27
Worksheet: Note C
Definition: Distance file format control: if 0, model is prepared for a triangular distance file; if 1, model is prepared for a square distance file. The file is read major record, minor record, distance. In a triangular file, the major record is the higher numbered location and the minor record is the lower numbered location. In a square file, the major record is the destination location number and the minor record is the origin location number.

ISU(60)

Mnemonic: Identity SoUrce
Specification: I3* Key: 11 First Column: 4*
Worksheet: VI Question: 1.b
Definition: The identification number of the Kth source.

ISUREQ(60)

Mnemonic: Identity SoUrce REQuest
Specification: I1 Key: 11 First Column: 3
Worksheet: VI Question: 2
Definition: Source routing request code: if 0, routing is not requested; if greater than zero, ISUREQ(K) levels of routing are requested for source K, if source K is in the solution.

ITRCO(600)

Mnemonic: Identity TRansportation COmmodity
Specification: I3* Key: 22 First Column: 6*
Worksheet: XIII
Definition: Transportation source/commodity code; Kth transportation activity: if source-origin, categories 1,4,5,6,8,9,11-15, source number; if site-origin, categories 2 and 3, commodity number.

ITRIGS(5,60)

Mnemonic: Identity TRIGger Source
Specification: I1 Key: 11 First Column: 45
Worksheet: VI Question: 5.c
Definition: Source separation option code for the Jth trigger price at the Kth source: if 0, option is not offered; if 1 option is offered.

ITROG(600) — Mnemonic: Identity TRansportation ORigin
Specification: I3   Key: 22   First Column: 10
Worksheet: XIII
Definition: The origin location number of the Kth transportation activity.

ITRSM(600) — Mnemonic: Identity TRansportation Site/Market
Specification: I3*   Key: 22   First Column: 14*
Worksheet: XIII
Definition: Offload identification of the Kth transportation activity: if category 3 or 9, ITRSM(K) is market number; otherwise ITRSM(K) is site number.

ITTYP(600) — Mnemonic: Identity Transportation TYPe
Specification: I1   Key: 22   First Column: 4
Worksheet: XII
Definition: Category type code for the Kth transportation activity: if 1, ITTYP(K) is source-to-site; if 2, ITTYP(K) is site-to-site; if 3, ITTYP(K) is site-to-market.

IVDRV — Mnemonic: Identity Variable cost DRiVer
Specification: I1   Key: 5   First Column: 58
Worksheet: Note C
Definition: Driver control for variable cost: if 0, VC(K) is used in "urging in"; if 1, VDRIV is used.

IXDED — Mnemonic: Identity eXclusion for DEDicated transfer
Specification: I1   Key: 1   First Column: 28
Worksheet: Note C
Definition: Dedicated transfer station control; if 0, normal dedicated transfer station procedure operates; if 1, dedicated transfer station procedure is suppressed.

KEMP — Specification: I3*
KEY(1): 15   First Column: 4*
KEY(2): 16   First Column: 4*
Definition(1): Process number.
KEY(3): 19   First Column: 74*
KEY(4): 20   First Column: 74*
Definition(2): Market number.

KEY — Specification: I2
Definition: Card group control: appears in columns 1 and 2 of all numbered cards.

KFOUT — Mnemonic: Kontrol Forcing OUT
Specification: I1   Key: 1   First Column: 29
Worksheet: Note C
Definition: Control for forcing out only in site and site-process forcing: if 0 through 8, procedure includes forcing in and forcing out; if 9, procedure uses forcing out only.

KITLE

Mnemonic: Kontrol tITLE
Specification: 19A4 Key: 2 First Column: 3
Definition: 76 alphanumeric digits for name of run.

KMIX

Mnemonic: Kontrol MIX
Specification: I1 Key: 3 First Column: 63
Worksheet: I Question: 6.a
Definition: Run control for availability of unconditional MSW as an option: if 0, option is not available; if 1 it is.

KOPT

Mnemonic: Kontrol OPTion
Specification: I1 Key: 1 First Column: 7
Worksheet: Note C
Definition: Output control for the optimizer: 1 is full output; 2 is matrix summary and final solution; 3 is matrix summary and solution of phase 3 and 4, if phase 4 is requested.

KOVER

Mnemonic: Kontrol OVERride
Specification: I2 Key: 1 First Column: 5
Worksheet: Note C
Definition: Control for site and site-process forcing and orverride of row/column limit control for double-forcing, as in NFORC = 2. If 0, a phase 3 or phase 4 solution will be generated, according to the setting of LPHASE. If 1, site forcing will be undertaken, LPHASE will be reset to 3. If 2, site-process forcing will be undertaken, LPHASE will be reset to 3. If 99, the NFORC = 2 matrix size limit of 50 rows and 100 columns is overridden, to become 90 rows and 360 columns. Any other value of KOVER is reset to 0.

KREVSP(80)

Mnemonic: Kategory REVenue Site-Process
Specification: I2 Key: 17 First Column: 39
Worksheet: X Question: 5.c
Definition: Energy category for EREVSP(K), the net energy value of any implicit revenues associated with the Kth site-process combination.

KSEPO

Mnemonic: Kontrol SEParatiOn
Specification: I1 Key: 3 First Column: 64
Worksheet: I Question: 6.b
Definition: Run control for availability of unconditional source separation: if 0, option is not available, if 1, available with national paper market; if 2, available with local paper market.

KSPA

Mnemonic: Kode SEParation A
Specification: I2 Key: 4 First Column: 4
Worksheet: I Question: 13.e
Definition: Energy category for collection of unseparated waste.

KSPB

Mnemonic: Kode SeParation B
Specification: I2 Key: 4 First Column: 16
Worksheet: I Question: 13.f
Definition: Energy category for incremental energy requirement due to source separation.

KSUPP

Mnemonic: Kontrol SUPPression
Specification: I1 Key: 1 First Column: 30
Worksheet: Note C
Definition: Control for suppression of intermediate information on search for an initial feasible solution: if 0, information is suppressed; if 1, information is provided.

LEMP

Specification: I1 Key: 16 First Column: 7
Worksheet: IX
Definition: Segment number for the subject process.

LIFC(50)

Mnemonic: LIFe Capitalization
Specification: I3 Key: 8 First Column: 49
Worksheet: III Question: 5.b
Definition: Years for capitalization of the loan for the Kth cost category.

LIFU(50)

Mnemonic: LIFe Useful
Specification: I3 Key: 8 First Column: 45
Worksheet: III Question: 5.a
Definition: Years of useful life for the Kth cost category.

LITLE(20)

Mnemonic: List tITLE for distances
Specification: 20A4
Worksheet: Note A
Definition: 80 alphanumeric digits available for naming the distance files.

LLA(30)

Mnemonic: Location LAndfill
Specification: I3 Key: 13 First Column: 4
Worksheet: Note A
Definition: Location number of the Kth landfill.

LMK(30)

Mnemonic: Location MarKet
Specification: I3 Key: 18 First Column: 29
Worksheet: Note A
Definition: Location number of the Kth market.

LOPT

Mnemonic: Logic OPTimization
Specification: I1 Key: 1 First Column: 8
Worksheet: Note C
Definition: Optimization mode control: if 0, minimum discounted cost; if 1, minimum undiscounted cost, if 2, maximum net energy; if 3, minimum linear form; if 4, feasibility test of a specified plan.

LPHASE

Mnemonic: Last PHASE
Specification: I1 Key: 1 First Column: 3
Worksheet: Note C
Definition: Last phase control: if not 4, last phase is 3 (optimization with fixed costs, but without forcing); if 4, last phase is 4 (column forcing, in accordance with setting of NFORC).

LSI(50)

Mnemonic: Location SIte
Specification: I3 Key: 12 First Column: 29
Worksheet: Note A
Definition: Location number of the Kth site.

LSTLO

Mnemonic: LaST LOcation
Specification: I3 Key: 1 First Column: 64
Worksheet: Note A
Definition: Last location in distance file. Edit checked against location numbers in input deck, but not against distance file.

LSU(60)

Mnemonic: Location SoUrce
Specification: I3 Key: 11 First Column: 29
Worksheet: Note A
Definition: Location number of Kth source.

NAMCC(50,5)

Mnemonic: NAMe of Cost Category
Specification: 5A4 Key: 8 First Column: 8
Worksheet: III Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth cost category.

NAMCO(20,5)

Mnemonic: NAMe of COmmodity
Specification: 5A4 Key: 10 First Column: 8
Worksheet: V Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth commodity.

NAMCS(10,5)

Mnemonic: NAMe of Cost Summation
Specification: 5A4 Key: 21 First Column: 8
Worksheet: XII Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth cost summation category.

NAMEC(20,5)

Mnemonic: NAMe of Energy Category
Specification: 5A4 Key: 9 First Column: 8
Worksheet: IV Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth energy category.

NAMLA(30,5)

Mnemonic: NAMe of LAndfill
Specification: 5A4 Key: 13 First Column: 8
Worksheet: VII Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth landfill.

NAMMK(30,5)

Mnemonic: NAMe of MarKet
Specification: 5A4 Key: 18 First Column: 8
Worksheet: XI Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth market.

NAMPR(30,5)

Mnemonic: NAMe of PRocess
Specification: 50A4 Key: 14 First Column: 8
Worksheet: IX Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth process.

NAMSI(50,5)

Mnemonic: NAMe of SIte
Specification: 5A4 Key: 12 First Column: 8
Worksheet: VII Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth site.

NAMSU(60,5)

Mnemonic: NAMe of SoUrce
Specification: 5A4 Key: 11 First Column: 8
Worksheet: VI Question: 1.a
Definition: 20 alphanumeric digits for the name of the Jth source.

NBDMK(30)

Mnemonic: Number of BounDs MarKet
Specification: I1 Key: 18 First Column: 54
Worksheet: XI Question: 4.a
Definition: Number of bounds for the Kth market.

NCC

Mnemonic: Number of Cost Categories
Specification: I2 Key: 1 First Column: 43
Minimum: 1 Maximum: 50
Worksheet: Note B
Definition: Number of cost categories.

NCOF(30)

Mnemonic: Number of COeFficients
Specification: I1 Key: 14 First Column: 32
Minimum: 0 Maximum: 7
Worksheet: IX Question: 5
Definition: Number of output ceofficients for the Kth process.

NCS

Mnemonic: Number of Cost Summation categories
Specification: I2 Key: 1 First Column: 71
Minimum: 0 Maximum: 10
Worksheet: Note B
Definition: Number of cost summation categories.

NEC

Mnemonic: Number of Energy Categories
Specification: I2 Key: 1 First Column: 46
Minimum: 0 Maximum: 20
Worksheet: Note B
Definition: Number of energy categories.

NFORC

Mnemonic: Number of FORCes
Specification: I1   Key: 1   First Column: 4
Worksheet: Note C
Definition: Column force control. Operates only when KOVER = 0 and LPHASE = 4. If 1, single column force; if 2, double column force; if 0 or greater than 2, NFORC is reset to 1.

NLA

Mnemonic: Number of LAndfills
Specification: I2   Key: 1   First Column: 61
Minimum: 0   Maximum: 30
Worksheet: Note B
Definition: Number of landfills.

NMK

Mnemonic: Number of MarKets
Specification: I2   Key: 1   First Column: 68
Minimum: 0   Maximum: 30
Worksheet: Note b
Definition: Definition of markets.

NOINT

Mnemonic: Number Of INTervals
Specification: I3   Key: 7   First Column: 3
Minimum: 1   Maximum: 200
Worksheet: II   Question: 2.b
Definition: Number of intervals in the paper market price distribution function.

NPR

Mnemonic: Number of PRocesses
Specification: I2   Key: 1   First Column: 55
Minimum: 2   Maximum: 30
Worksheet: Note B
Definition: Number of processes.

NPROSI(50)

Mnemonic: Number of PROcesses at SIte
Specification: I2   Key: 12   First Column: 39
Worksheet: X
Definition: The number of processes offered at the Kth site.

NREM(50)

Mnemonic: Number years REMaining
Specification: I2   Key: 8   First Column: 56
Worksheet: III   Question: 5.d
Definition: Years of useful life remaining for existing capital for the Kth cost category.

NSEG(30)

Mnemonic: Number of SEGments
Specification: I1   Key: 14   First Column: 33
Minimum: 1   Maximum: 3
Worksheet: IX
Definition: Number of linear segments in the Kth processes cost/energy function.

NSGMK(30) Mnemonic: Number SeGments MarKet
Specification: I1 Key: 18 First Column: 53
Minimum: 1 Maximum: 5
Worksheet: XI Question: 4.d
Definition: Number of linear segments in the Kth market.

NSI Mnemonic: Number of SItes
Specification: I2 Key: 1 First Column: 52
Minimum: 1 Maximum: 50
Worksheet: Note B
Definition: Number of sites.

NSP Mnemonic: Number of Site Processes
Specification: I2 Key: 1 First Column: 58
Minimum: 1 Maximum: 80
Worksheet: Note B
Definition: Number of site processes.

NSU Mnemonic: Number of SoUrces
Specification: I2 Key: 1 First Column: 49
Minimum: 1 Maximum: 60
Worksheet: Note B
Definition: Number of sources.

NTRIG Mnemonic: Number of TRIGgers
Specification: I1 Key: 3 First Column: 65
Minimum: 0 Maximum: 5
Worksheet: II Question: 4.a
Definition: Number of trigger prices used in the run.

NYR Mnemonic: Number of YeaRs
Specification: I2 Key: 1 First Column: 12
Minimum: 1 Maximum: 99
Worksheet: I Question: 1.b
Definition: Number of years in the study period.

PCCAN Mnemonic: Per Cent CANs
Specification: F5.2 Key: 3 First Column: 13
Minimum: 0.0 Maximum: 90.0
Worksheet: I Question: 4.c
Definition: The percentage of cans in MSW by weight.

PCGLA Mnemonic: Per Cent GLAss
Specification: F5.2 Key: 3 First Column: 8
Minimum: 0.0 Maximum: 90.0
Worksheet: I Question: 4.b
Definition: The percentage of glass in MSW by weight.

PCPAP Mnemonic: Per Cent PAPer
Specification: F5.2 Key: 3 First Column: 3
Minimum: 0.0 Maximum: 90.0
Worksheet: I Question: 4.a
Definition: The percentage of paper in MSW by weight.

PCRES

Mnemonic: Per Cent RESidue
Specification: F5.2 Key: 3 First Column: 18
Minimum: 0.0 Maximum: 100.0
Worksheet: I Question: 4.d
Definition: The percentage of residue in MSW by weight.

PRCAN

Mnemonic: PRice CANs
Specification: F5.2 Key: 3 First Column: 40
Minimum: 0.0 Maximum: 1000.0
Worksheet: I Question: 7.c
Definition: The price of cans in the market in dollars per ton.

PRGLA

Mnemonic: PRice of GLAss
Specification: F5.2 Key: 3 First Column: 35
Minimum: 0.0 Maximum: 1000.0
Worksheet: I Question: 7.b
Definition: The price of glass in the market in dollars per ton.

PRPAP

Mnemonic: PRice PAPer
Specification: F5.2 Key: 1 First Column: 75
Worksheet: I Question: 7.a
Definition: Average price of paper as realized in the market in dollars per ton. Used only if KSEPO = 1; reset by internally generated PRPAP if NTRIG greater than 0 and INPAP = 1.

PTRIG(5)

Mnemonic: Price TRIGger
Specification: F5.2 Key: 6 First Column: 7
Worksheet: II Question: 4.a
Definition: Kth trigger price in dollars per ton.

RATE(50)

Mnemonic: interest RATE
Specification: F4.1 Key: 8 First Column: 52
Minimum: 0.05 Maximum: 80.0
Worksheet: I Question: 2
Definition: If IDCC(K) = 1, RATE(K) is the interest rate in percent for capitalization of the Kth cost category. Not checked for IDCC(K) = 0.

RCAH

Mnemonic: Ratio CAn Haul
Specification: F10.5 Key: 4 First Column: 48
Worksheet: I Question: 12.c
Definition: Conversion factor for haul of cans to the market: ratio of energy haul requirement per dollar cost of haul.

REVSP(80)

Mnemonic: REVenue Site-Process
Specification: F9.3 Key: 17 First Column: 22
Worksheet: X Question: 5.b
Definition: Implicit revenue in dollars per input ton into the Kth site-process, net of cost of haul to the market. Represents the sum of net revenues arising from all recoverables not entered into the model (i.e., the commodity list) as explicit commodities.

RGLH

Mnemonic: Ratio GLass Haul
Specification: F10.5 Key: 4 First Column: 38
Worksheet: I Question: 12.b
Definition: Conversion factor for haul of glass to the market: ratio of energy haul requirement per dollar cost of haul.

RPAH

Mnemonic: Ratio PAper Haul
Specification: F10.5 Key: 4 First Column: 28
Worksheet: I Question: 12.a
Definition: Conversion factor for haul of paper to the market: ratio of energy haul requirement per dollar cost of haul.

RPRPAP

Mnemonic: Ratio PRocessing PAPer
Specification: F4.3 Key: 3 First Column: 69
Minimum: 0.05 Maximum: 20.0
Worksheet: I Question: 9.a
Definition: Ratio of paper to MSW as a cost driver in processing paper as waste.

RPRRES

Mnemonic: Ratio PRocessing RESidue
Specification: F4.3 Key: 3 First Column: 77
Minimum: 0.05 Maximum: 20.0
Worksheet: I Question: 9.b
Definition: Ratio of source separation residue to MSW as a cost driver in processing residue.

SIZINT

Mnemonic: SIZe of INTerval
Specification: F5.3 Key: 7 First Column: 6
Worksheet: II Question: 2.c
Definition: Size of interval in the paper market price distribution function in dollars of price per ton.

TSU(60)

Mnemonic: Tons SoUrce
Specification: F10.3 Key: 11 First Column: 32
Worksheet: VI Question: 4
Definition: Tonnage at the Kth source in thousands of tons per year.

USLA(30) — Mnemonic: USe of LAnd
Specification: F8.5 Key: 14 First Column: 35
Minimum: 0.0001 if ILACOF(K) = 1
Worksheet: IX Question: 4
Definition: Use of land on site by process K in acre feet per thousand tons input.

VDRIV — Mnemonic: Variable cost DRIVer
Specification: F5.2 Key: 5 First Column: 59
Worksheet: Note C
Definition: Variable cost used in place of VC(K) if IVDRV=1 for urging into the solution an acticity outside of the solution in site and site-process forcing.

WTCC(50) — Mnemonic: Weight Cost Category
Specification: F8.3 Key: 8 First Column: 31
Worksheet: III Question: 3
Definition: Weight of the Kth cost category for use in LOPT=3 or 4. Should be positive or zero.

WTEC(20) — Mnemonic: Weight Energy Category
Specification: F8.3 Key: 9 First Column: 29
Worksheet: IV Question: 2
Definition: Weight for the Kth energy category for use in LOPT = 3 or 4. Should be negative or zero.

A.2 DICTIONARY OF INTERNAL VARIABLES

The variables which are defined in this section are used for intermediate calculations and hence are not directly under the control of the user, as the model's inputs. Three types of variables are covered: (1) any variable which appears in a DIMENSION statement; (2) any variable which appears in a COMMON statement; and (3) any variable which appears in an output report but is not an input. This discussion should provide enough information to enable programmers to make changes to the source code dictated by either user needs or operating system characteristics.

A(8100) — Definition: The Kth non-zero coefficient as entered into the optimizer. See COL(K) and ROW(K).

B(90) — Definition: The right-hand side of the Kth row.

BASIS(90) — Definition: The column index of the Kth activity in the basis. As generated by the front end, BASIS(K) is an element in the advanced starting point (initial feasible solution) for input into the optimizer. In the back end, it is an element of the generated optimal solution.

BETAO(90) — Definition: The activity level (solution value) of the Kth activity in the solution.

BHO(90) — Mnemonic: B HOld
Definition: Vector used to hold B(K) (right-hand side) during optimization, in order to reset B(K) for the next call to the optimizer.

CC(50) — Definition: Cumulative lifetime undiscounted inflated cost per dollar of first-year cost (for operating cost) or per dollar of capital cost (for capital cost) for the Kth cost category.

CCW(50) — Definition: CC(K)*WTCC(K) for LOPT = 1, otherwise, DC(K)*WTCC(K). Weighted lifetime cost per dollar of first-year cost (for operating cost) or per dollar of capital cost (for capital cost) for the Jth cost category.

CLEVN(100) — Mnemonic: Cost LEVel New
Definition: The Kth continuation factor at the current level, new array (new array is in process of accumulation).

CLEVO(100) — Mnemonic: Cost LEVel Old
Definition: The Kth continuation factor at the current level, old array (old array is in process of further reticulation).

CMCAN — Mnemonic: CuMulator CANs
Definition: Cumulator for cans in thousand tons per year (from source separation).

CMGLA — MNemonic: CuMulator GLAss
Definition: Cumulator for glass in thousand tons per year (from source separation).

CMPAP — Mnemonic: CuMulator PAPer
Definition: Cumulator for paper in thousand tons per year (from source separation). For national paper market only.

CNA(450) — Mnemonic: Column NAme
Definition: Name of Jth column. In the matrix before expansion, CNA(K) is the column index. Artificials are 2000 plus the row number of the equation. Slacks are 3000 plus the row number of the constraint.

COL(8100) — Definition: The column index of the Kth coefficient (non-zero) as prepared for entry into the optimizer. See A(K) and ROW(K).

CSPLLC(40,7) — Mnemonic: Cost SPLit Location-Commodity
Definition: Cost per ton for transportation, Kth split in Jth location-commodity of the solution.

CSPLSU(60,9) — Mnemonic: Cost SPLit SoUrce
Definition: Cost per ton for transportation, Kth split in Jth source of the solution

CUMC
Mnemonic: CUMulator undiscounted Cost
Definition: Cumulator for lifetime undiscounted cost.

CUMD
Mnemonic: CUMulator Discounted cost
Definition: Cumulator for lifetime discounted cost.

CUME
Mnemonic: CUMulator Energy
Definition: Cumulator for lifetime energy.

CUMMC
Mnemonic: CUMulator Market undiscounted Cost
Definition: Cumulator for marketable commodities in undiscounted cost; cumulates gross revenues at the market.

CUMMD
Mnemonic: CUMulator Market Discounted cost
Definition: Cumulator for marketable commodities in discounted cost; cumulates gross revenues at the market.

CUMPC
Mnemonic: CUMulator Paper Cost (undiscounted)
Definition: Cumulator for gross revenue of marketed paper (national paper market only) in undiscounted dollars.

CUMPD
Mnemonic: CUMulator Paper cost (Discounted)
Definition: Cumulator for gross revenue of marketed paper (national paper market only) in discounted dollars.

CUMRC
Mnemonic: CUMulator Revenues Cost (undiscounted)
Definition: Cumulator for net implicit revenues in undiscounted dollars.

CUMRD
Mnemonic: CUMulator Revenues cost (Discounted)
Definition: Cumulator for net implicit revenues in discounted dollars.

CUMTCC
Mnemonic: CUMulator Transportation-Commodity Cost (undiscounted)
Definition: Cumulator for costs of haul to market of explicit commodities in undiscounted dollars.

CUMTCD
Mnemomic: CUMulator Transportation-Commodity cost (Discounted)
Definition: Cumulator for costs of haul to market of explicit commodities in discounted dollars.

CUMTSC
Mnemonic: CUMulator Transportation Source separation Cost (undiscounted)
Definition: Cumulator for cost of haul to market of source-separation recoverables in undiscounted dollars.

CUMTSD
Mnemonic: CUMulator Transportation Source separation cost (Discounted)
Definition: Cumulator for costs of haul to market of source-separation recoverables in discounted dollars.

DC(50)	Definition: Cumulative lifetime discounted inflated cost per dollar of first-year cost (for operating cost) or per dollar of capital cost (for capital cost) for the Kth cost category.
DIST(600)	Definition: Distance (in miles, minutes or cost) by packer for the Kth transportation activity.
DISTV(600)	Definition: Distance (in miles, minutes or cost) by van for the Kth transportation activity.
FC(450)	Mnemomic: Fixed Cost Definition: Fixed cost (i.e., intercept) for Kth column; objective function as entered into the optimizer.
FFFF(60)	Definition: A temporary holding array for holding fixed costs for later resetting during site and site-process forcing. The Kth fixed cost in the array.
FSPLSU(60,9)	Mnemonic: Factor SPLit SoUrce Definition: Real tons (tons as actually delivered) for the Kth split (Kth offload in solution) of the Jth source.
ICOLOC(50)	Mnemonic: Identity COmmodity-LOcation (Commodity) Definition: The commodity number of the Kth commodity-location in the matrix, as prepared for entry into the optimizer.
ICSP(40,5)	Mnemonic: Identity Compact Site-Process Definition: Index number in the full siproc file (i.e., as input) corresponding to the Kth process in the solution of the Jth site in the solution.
IEXG(240)	Mnemonic: Identity EXpansion seGment Definition: The segment number of the Kth entry in the expanded site-process file (site-processes expanded by number of segments, as prepared for entry into the optimizer).
IEXP(240)	Mnemonic: Identity EXpansion Process Definition: The process number of the Kth entry in the expanded site-process file. See IEXG(K).
IEXS(240)	Mnemonic: Identity EXpansion Site Definition: The site number of the Kth entry in the expanded site-process file. See IEXG(K).
IGX(150)	Mnemonic: Identity seGment eXpanded Definition: The segment number of the Kth entry in the expanded market file (markets expanded by number of segments, as prepared for entry into the optimizer).

ILCOC(40) — Mnemonic: Identity Location-COmmodity (Commodity)
Definition: The commodity number of the Kth location-commodity in the solution.

ILCOL(40) — Mnemonic: Identity Loction-COmmodity (Location)
Definition: The location number of the Kth location-commodity in the solution.

ILEVN(100) — Mnemonic: Identity LEVel New
Definition: The site number of the Kth continuation at the current level; new array is in process of accumulation.

ILEVO(100) — Mnemonic: Identity LEVel Old
Definition: The site number of the Kth continuation at the current level; old array is in process of further reticulation.

IMX(150) — Mnemonic: Identity Market eXpanded
Definition: The market number of the Kth entry in the expanded market file. See IGX(K).

INDIC(90) — Mnemonic: INDICator
Definition: (1) In front end and optimizer, row K type code: 0 is equation; 1 is upper-bound constraint. (2) in back end, for Kth activity in solution:
if transportation activity, INDIC(K) equals category (ITTYP(K))
if residue activity, INDIC(K) equals 21
if paper activity, INDIC(K) equals 22
if processing activity, INDIC(K) equals 23
if marketing activity, INDIC(K) equals 31.

IPAP(50) — Mnemonic: Identity PAPer
Definition: The site number of the Kth paper activity, as prepared for input into the optimizer.

IREMCF(40,5,7) — Mnemonic: Identity REMaining CoeFficient
Definition: Identity code, coefficient valuation: if 0, the Kth coefficient of the Jth process in the solution of the Ith site in the solution has been fully valued; if 1, that coefficient has not been fully valued.

IREMLC(40) — Mnemonic: Identity REMaining Location-Commodity
Definition: Identification code, location-commodity valuation: if 0, the Kth location-commodity has been fully valued; if 1, it has not been fully valued.

IREMSI(40) — Mnemonic: Identity REMaining SIte
Definition: Identity code, site valuation: if 0, site has been fully costed; if 1, it has not.

IRES(50) — Mnemonic: Identity RESidue
Definition: The site number of the Kth residue activity.

IRLCSP(40,7)	Mnemonic: Identity Remaining Location-Commodity SPlit Definition: Identification code, location-commodity split valuation: if 0, the Kth split of the Jth location-commodity in the solution has been fully costed; if 1, it has not been fully costed.
IRSP(40,5)	Mnemonic: Identity Remaining Site-Process Definition: Identification code, site-process valuation: if 0, Kth process in solution of Jth site in solution has been fully costed; if 1, it has not.
ISPLSU(60,9)	Mnemonic: Identity SPLit SoUrce Definition: Offload identification number for Kth split of Jth source in solution; if K = 1, 2, or 3, ISPLSU(J,K) is undefined; if K is greater than 3, ISPLSU(J,K) is site number.
ISPLLC	Mnemonic: Identity SPlit Location-Commodity Definition: Identification number of Kth split of Jth location-commodity in the solution. If JSPLLC(J,K) = 2, ISPLLC(J,K) is site number of offload; if JSPLLC(J,K) = 3, ISPLLC(J,K) is market number of offload.
ITRDS(600)	Mnemonic: Identity TRansportation DeStination Definition: Destination location number of the Kth transportation activity.
ITSTA(600)	Mnemonic: Identity Transfer STAtion Definition: Transfer station identity code for the Kth transportation activity: if), there is no dedicated transfer station for the haul; if 1, 2, or 3, ITSTA(K) is the segment number of the transfer station process (process 01).
ITY	Definition: Category number, ITTYP(K) of the current transportation activity (read).
JEF	Definition: Write output device number for the optimizer. Used by RRPLAN to communicate output suppression information to the optimizer.
JREF(300)	Definition: The transportation file index, retained during sorting of locations for running against distance files.
JSPLLC(40,7)	Definition: Type code for ISPLLC(J,K). If 2, ISPLLC(J,K) is a site number; if 3, ISPLLC(J,K) is a market number.
KFLAG(22)	Definition: A series of flags to identify the first solution activity of a type for purpose of header control.

LBAS(90) Mnemonic: Last BASis
Definition: (1) During optimization, array is a temporary hold for the previous basis. (2) In the back end, for the Kth activity in the solution, if transportation activity, LBAS(K) is the index number in the transportation file; if a processing activity, it is the index number in the (input) site-process file; if a marketing activity, it is the index number in the expanded market file; if paper or residue activity, it is the site number.

LFLAG Definition: LFLAG = KSUPP.

LFLAGX Definition: LFLAGX = KFOUT.

LISTO(60) Definition: Array used to hold site numbers for site forcing and site-process numbers for site-process forcing for the current site or site-process being forced.

LOCD(300) Mnemonic: LOCation Destination
Definition: Array for temporary holding of destination locations for sorting and match against the distance files.

LOCO(300) Mnemonic: LOCation Origin
Definition: Array for temporary holding of origin locations for sorting and match against the distance files.

M Definition: Number of rows.

MUT Definition: Read input device number for the optimizer. Used by RRPLAN to communicate output suppression information to the optimizer.

N Definition: Number of columns.

NCOLO Mnemonic: Number of COlumn LOcation
Definition: The number of commodity locations in the matrix, as prepared for input into the optimizer.

NCSP(40) Mnemonic: Number of Compact Site-Processes
Definition: The number of processes in the solution at the Kth site in the solution.

NEX Mnemonic: Number EXpanded
Definition: The number of activities in the expanded site-process file in the matrix, as prepared for input into the optimizer.

NFSP Mnemonic: Number of First Site-Process
Definition: The column number of the first activity in the matrix that is a processing activity.

NFTR(15) Mnemonic: Number of First TRansportation
Definition: The index number in the transportation file corresponding to the first entry that is category K.

NMX — Mnemonic: Number of Markets eXpanded
Definition: The number of activities in the expanded market file.

NORIG — Mnemonic: N ORIGinal
Definition: N (i.e., the number of columns) before the addition of special column Z(column number NZ), which is always the last column before slacks, and which is called by the optimizer to relieve constraints in finding an initial feasible solution.

NPAP — Mnemonic: Number of PAPer
Definition: The number of paper activities in the matrix.

NRES — Mnemonic: Number of RESidue
Definition: The number of residue activities in the matrix.

NSLACK — Mnemonic: Numer of SLACKs
Definition: The number of slacks in the matrix.

NTRA — Mnemonic: Number of TRansportation A
Definition: The number of category type 1 transportation entries (source origin) in the input file. Since these entries are not in the matrix, NTRA becomes the difference between the index in the transportation file and the column index in the matrix. Category types greater than 3 are all obtained by expansion from the type 1 entries in the input file, and are all in the matrix.

NTRAN — Mnemonic: Number of TRANsportation (MAIN)
Number of TRANsformation (SWAP)
Definition: In MAIN, the number of transportation activities in the transportation file. NTRNAS = NTRAN. NTRAN is reset to 0 before calling SWAP.

NTRANS — Mnemonic: Number of TRANSportation
Definition: The number of transportation activities in the transportation file. NTRANS = NTRAN.

NTRB — Mnemonic: Number of TRansportation B
Definition: The number of category type 2 transportation activities in the input file and in the matrix.

NTRC — Mnemonic: Number of TRansportation C
Definition: The number of category type 3 transportation activities in the input file and in the matrix.

NUMNZ — Mnemonic: NUMber Non Zero
Definition: The number of non-zero coefficients in the matrix.

NXTR(15)	Mnemonic: Number eXpansion TRansportation Definition: The number of transportation activities in category type K (K>3).
NZ	Mnemonic: Number Z Definition: The number of column Z. (See NORIG)
NZX	Mnemonic: Number Z eXtra Definition: Column Z extra is identical to column Z (i.e., it relieves all constraints) except that it has a lower cost in the objective function. It is offered only under LOPT = 4. It is called in by SWAP to replace column Z and thus bypasses the stop at the end of phase 3 if column Z is found in the solution.
OBASI(90)	Mnemonic: Old BASIs Definition: The Kth activity in the basis as output by the optimizer (i.e. the solution).
OLDOB	Mnemonic: OLD OBjective Definition: The objective function, or solution value, as output by the optimizer.
PCSP(40,5)	Mnemonic: Per Cent Site Process Definition: The output of the Kth process in the solution at the Jth site in the solution, as a percent of the output at the Jth site in the solution.
REQLA(30)	Mnemonic: REQuirement for LAnd Definition: The total requirement for land use in acre-feet at the Kth landfill.
REXC(240)	Mnemonic: Relist EXpansion Capacity Definition: The capacity in thousand tons per year of the Kth activity in the expanded site-process file, as prepared for input to the optimizer.
RK	Definition: Level of current activity in the back end.
RNA(90)	Mnemonic: Row NAme Definition: The name of the Kth row; it is equal to the row index.
ROW(8100)	Mnemonic: ROW Definition: The row index of the Kth non-zero coefficient in the matrix.
RSTACT(90)	Mnemonic: ReSTore ACTivity Definition: An array for holding, for possible later restoration, the activity levels of the previous solution in site and site-process forcing.

RSTOBJ	Mnemonic: ReStore OBJective Definition: The value of the objective function, solution value, of the previous solution held for possible restoration. Used in site and site-process forcing.
TCSI(40)	Mnemonic: Tonnage Compact SIte Definition: Tonnage in thousands of tons per year at the Kth site in the solution.
TEMPT	Definition: An arbitrarily small number.
TOLCO(40,7)	Mnemonic: TOnnage Location COmmodity Definition: Tonnage of the Kth split of the Jth location-commodity in the solution in percentage of the total tonnage at the location-commodity.
TOTC	Mnemonic: TOTal Cost (undiscounted) Definition: Total undiscounted life-cycle cost for the solution.
TOTD	Mnemonic: TOTal cost (Discounted) Definition: Total discounted life-cycle cost for the solution.
TOTE	Mnemonic: TOTal Energy Definition: Total lifetime net energy for the solution.
TOTT	Mnemonic: TOTal Tons Definition: Total lifetime tons for the solution.
TSPLSU(60,9)	Mnemonic: Tons SPLit SoUrce Definition: Tonnage at the Kth split in the solution for the Jth source, in thousands of tons per year.
VC(450)	Mnemonic: Variable Cost Definition: Variable cost (i.e. slope) of activity K in the matrix objective function.
VVVV(60)	Definition: A temporary holding array for variable costs for later resetting during site and site-process forcing. The Kth variable cost in the array.

APPENDIX B DESCRIPTION OF SUBROUTINES

This appendix provides a brief description of all subroutines in the model. For the MAIN program, a more detailed description by functional block is given in chapter 2. The appendix is arranged in alphabetical order by subroutine name. Each subroutine is described on a single program summary sheet. This summary sheet includes: (a) the name of the subroutine; (b) the call statement; (c) a narrative description; (d) the calling routines; (e) the called routines; (f) the commons referenced; and (g) any messages generated. The information provided on the summary sheets in conjunction with the model flowchart shown in chapter 2 should facilitate the programmer's task of effectively maintaining the model. The interactions among subroutines which are explicitly stated on the summary sheets should also assist the programmer in making any modifications to the source code dictated by user needs or peculiarities of the operating system.

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: BTRAN

CALL STATEMENT: BTRAN

DESCRIPTION:

Calculates the current pricing vector, PI, from basis vector costs and updates it by all previous transformations. The transformations are obtained by reading the ETAFI vector in reverse sequence from the last to first entry. The current price is calculated as the value of the matrix product of the basic variable costs and the inverse activity, $C_B \cdot B^{-1}$.

CALLED BY: LP,NEWA,SETUB,SIMPL,SWITC,TINV

CALLS: DOTPR,ETARB

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: CHECK

CALL STATEMENT: CHECK

DESCRIPTION:

This subroutine edit-checks the values of key inputs. If an edit-checked input is found to be out of range, a message is generated and the run is stopped. The message provides the read key and the card number in the sequence upon which the out-of-range input was discovered. The out-of-range value is then presented as either a real number or an integer, as appropriate to the out-of-range input.

CALLED BY: MAIN

CALLS: None

COMMONS REFERENCED: RRPLAN

MESSAGES GENERATED:

ERROR: VALUE OUT OF RANGE. PROGRAM ABORTS.
READ KEY: -- CARD NUMBER OF SEQUENCE: --

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: COMEI

CALL STATEMENT: COMEI(COLMI)

DESCRIPTION:

This subroutine prices out the A matrix with current prices of basis vectors and determines which column is to be brought into the basis. The column is used for iterations in phases 1 and 2. The selected column number is stored in COLMI. This vector is then brought into the basis.

CALLED BY: LP

CALLS: DOTPR,SWITC,PASSA

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: DOTPR

CALL STATEMENT: DOTPR(X,Y,DOT,T)

DESCRIPTION:

Calculates the dot product of two vectors. It uses Winograd's algorithm for the computation. The calculated value is the objective function for the computation. The calculated value is the objective function for the current basis. When execution is in phases 3 and 4, the calculated value is the fixed charge objective function for the current basis.

CALLED BY: BTRAN,COMEI,FORCE,LP,SWAP,SWITC,TINV

CALLS: None

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: ETAGE

CALL STATEMENT: ETAGE (COLMI,R)

DESCRIPTION:

This subroutine performs a simplex transformation which creates a new vector, ETA. The pivotal element is column COLMI and row R. The ETA vector is one factor in the product form of the basis inverse. Any change in the basis is noted and the basis vector levels and pointers are updated.

CALLED BY: LP,REINV,SIMPL

CALLS: ETARI,FTRAN

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: ETARB

CALL STATEMENT: ETARB(R)

DESCRIPTION:

Reads the next sparse image ETA vector, in reverse sequence, into an expanded vector for use by BTRAN. It sets the argument R equal to the pivot row number for the transformation.

CALLED BY: BTRAN

CALLS: None

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: ETARF

CALL STATEMENT: ETARF(R)

DESCRIPTION:

ETARF reads the next sequential ETA vector in sparse form into an expanded ETA vector for use by subroutine FTRAN. This vector is read in a forward direction.

CALLED BY: FTRAN

CALLS: None

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: ETARI

CALL STATEMENT: ETARI(R)

DESCRIPTION:

This subroutine maintains a sequential file of ETA vectors produced by the preceeding simplex transformations. The vector is used for product calculations or inverse computations. A new vector is written onto the end of the array in sparse form if the storage limit has not been exceeded. The argument "R" is used to determine the pointers necessary.

CALLED BY: ETAGE

CALLS: None

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: FORCE

CALL STATEMENT: FORCE(ICOL,IFLAG)

DESCRIPTION:

FORCE is a column forcing subroutine for use in phase 4 execution. It saves the values necessary for a quick BASIS restoration and then forces a column into the current basis. If a better solution is not found, then the entering BASIS is restored and the forcing flag, IFLAG, is set for return to the calling procedure.

CALLED BY: SWAP

CALLS: DOTPR,SAVE,SETUB,SIMPL

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: FSORT

CALL STATEMENT: FSORT

DESCRIPTION:

FSORT is a utility subroutine which sorts the 'A' matrix by row and column. It is used during the initialization. Only the non-zero matrix elements are retained. The sparse matrix and the row and column pointer tables are rearranged into column order.

CALLED BY: SETUA

CALLS: None

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED:

DUPLICATE ROW AND COLUMN LOCATIONS ENCOUNTERED DURING SORT ROUTINE

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: FTRAN

CALL STATEMENT: FTRAN(TVEC)

DESCRIPTION:

FTRAN is called to update the matrix column that is being brought into the BASIS. The column is modified by all previous transformations which are kept in the product form of the inverse. The transformation vector, TVEC, is assigned the matrix product of the current basis inverse times the transformation vector. The principal objective is to determine which vectors are to be removed from the current BASIS.

CALLED BY: ETAGE,LP,NEWA,REINV,SIMPL

CALLS: ETARF

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: INIT

CALL STATEMENT: INIT

DESCRIPTION:

This subroutine creates and checks the initial basis and adds artificials and slacks to get a full identity matrix. A test of the optional user supplied solution is initiated. Execution is terminated if the initial basis is not feasible even after the 'Z' column substitution. When the basis is feasible, the logic sets up variables for phase 3 execution.

CALLED BY: SWAP

CALLS: PASSA,SETUB

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED:

****THIS PROGRAM HALTS BASIS NOT GOOD****

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: KERR

CALL STATEMENT: KERR

DESCRIPTION:

This subroutine checks that no capital costs are used for those cost categories which are annual recurring costs (i.e., operating costs). If a capital cost is used improperly, a message is written and the run is stopped.

CALLED BY: MAIN

CALLS: None

COMMONS REFERENCED: RRPLAN

MESSAGES GENERATED:

CAPITAL COST CATEGORY USED IMPROPERLY. PROGRAM ABORTS.

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: LP

CALL STATEMENT: LP

DESCRIPTION:

This subroutine is an implementation of the revised simplex method for solving linear programming problems. It performs the phase I and phase 2 iterations. It uses the product form of the inverse and Winograd's algorithm for computing dot products. An arbitrary initial basis will be accepted if feasible. The coded logic is primarily calls to other subroutines. The subroutine will also solve the application problem for a strictly linear programming solution exclusive of fixed charges.

CALLED BY: SWAP

CALLS: BTRAN,COMEI,ETAGE,FTRAN,PASSA,REMOV,TINV

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: NEWA

CALL STATEMENT: NEWA

DESCRIPTION:

This subroutine generates the sparse form of the current "A" matrix. It creates the sparse form to help eliminate redundant computations during phases 3 and 4. If the matrix is too dense the process is not recoverable and flags are set to terminate the entire execution.

CALLED BY: SWAP

CALLS: BTRAN,FTRAN,PASSA,REINV

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: OUT

CALL STATEMENT: OUT(LINK,ISET)

DESCRIPTION:

This subroutine outputs the input vectors and results based on the user-supplied print option variable, KOPT, which is internally equated to two variables, LINK and ISET. If LINK = 0, the OUT routine is not called. If LINK is greater or less than zero, then outputs are determined by the value is ISET.

CALLED BY: SWAP

CALLS: None

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED:

Three types of outputs can be requested by the user:

1. Matrix arrays and counters;
2. Results for intermediate phases (vector format); and
3. Results for the final phase (vector format).

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: PASSA

CALL STATEMENT: PASSA(GETCO)

DESCRIPTION:

This subroutine reads one specific column, GETCO, of the 'A' matrix in full form and stores it in array ALPHA. (For every row there is an entry.)

CALLED BY: COMEI,INIT,LP,NEWA,REINV,SIMPL,TINV

CALLS: None

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: PROC

CALL STATEMENT: PROC

DESCRIPTION:

This subroutine prints headings for the process activity summary report.

CALLED BY: MAIN

CALLS: None

COMMONS REFERENCED: None

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: REINV

CALL STATEMENT: REINV

DESCRIPTION:

This subroutine reinverts the current basis after isolating basis vector entries already in original positions. Levels of activity are recomputed after basis reinversion.

CALLED BY: NEWA,SETUB,TINV

CALLS: ETAGE,FTRAN,PASSA

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED:

THE CURRENT BASIS IS NOT INVERTABLE. REINVERSION WAS PROCEEDING.
THE CURRENT BASIS IS NOT INVERTABLE. IT HAS A NEGATIVE ACTIVITY LEVEL.

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: REMOV

CALL STATEMENT: REMOV(COLMI,ROWOU)

DESCRIPTION:

This subroutine checks for negative entries in the basis columns in the matrix rows corresponding to the artificials. If a negative entry is found, the artificial for that row is removed from the basis. If an artificial is not removed, then the removed row, ROWOU, has the minimum activity level for column, J, i.e., BETAO(I) over matrix coefficients A(I,J).

CALLED BY: LP

CALLS: None

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED:

UNBOUNDED SOLUTION WITH COLUMN –– IN THE BASIS

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: SAVE

CALL STATEMENT: SAVE(JBASE,TOBJ,JLAST,TBETA)

DESCRIPTION:

This subroutine stores the current basis vector, the current objective value, the current level of the basic variables, and the last column to leave the basis.

CALLED BY: FORCE,SETUB,SWAP

CALLS: DOTPR

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: SEQUO

CALL STATEMENT: SEQUO

DESCRIPTION:

This subroutine checks that all cards are entered in the proper sequence. If a card is out of sequence, a message is generated and the run is stopped. The message provides: the read key (i.e., the card type indicated); the proper read key (i.e., the card type expected); the card number in the sequence; and identification number (e.g., source 3). The run will stop if either the read key and the proper read key do not match, or if the card number in the sequence and the identification number do not match. All cards are checked for sequence.

CALLED BY: MAIN

CALLS: None

COMMONS REFERENCED: RRPLAN

MESSAGES GENERATED:

SEQUENCE ERROR. PROGRAM ABORTS. READ KEY: -- PROPER READ KEY: --

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: SETUA

CALL STATEMENT: SETUA

DESCRIPTION:

This subroutine augments the 'A' matrix to get the canonical form with all equality constraints. It arranges a sparse matrix and counts the number of zero entries in a column.

CALLED BY: SWAP

CALLS: FSORT

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED:

PROBLEM TERMINATED DUE TO INITIALIZATION PROCEDURES OVERFLOWING AVAILABLE STORAGE

NUMBER OF ELEMENTS IN INITIAL 'A' MATRIX INCLUDING REQUIRED AND SURPLUS VARIABLES HAS EXCEEDED THE LIMIT ALLOWED FOR STORAGE WHICH IS CURRENTLY SET AT 8100

NUMBER OF COLUMNS, INCLUDING REQUIRED SLACK AND SURPLUS VARIABLES, HAS EXCEEDED THE LIMIT ALLOWED FOR STORAGE WHICH IS CURRENTLY SET AT 360

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: SETUB

CALL STATEMENT: SETUB(ARRAY,LST,BTA)

DESCRIPTION:

SETUB restores the basic vector to computation status. If a reinversion occurred naturally during iterations, then subroutine REINV is called to do a reinversion. If the starting basis was not acceptable, then subroutine ZCOLIN is called to substitute the 'Z' column for the basis column with the largest negative activity level and reinversion is tried again.

CALLED BY: FORCE,INIT,SWAP

CALLS: BTRAN,REINV,SAVE,ZCOLIN

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED:

ERROR NEW COLUMN TO BE INSERTED. OLD COLUMN -- NEW COLUMN --

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: SIMPL

CALL STATEMENT: SIMPL(COLMI)

DESCRIPTION:

This subroutine uses the fixed charge criterion to find a solution better than that found by the LP subroutine, or the initial basis as provided by the user or generated by the program. If simple forcing is used then iterations are performed until a local optimum solution is found.

CALLED BY: FORCE,SWAP

CALLS: BTRAN,DOTPR,ETAGE,FTRAN,PASSA,TINV

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: SIZE

CALL STATEMENT: SIZE(M,N,NUMNZ)

DESCRIPTION:

This subroutine checks that the matrix size or number of non-zero elements within the matrix does not exceed the limits specified in the MAIN program. The limits are currently: 90 rows; 360 columns (before slacks); and 8100 non-zero elements. If one of these limits is exceeded, a message is written and the run is stopped.

CALLED BY: MAIN

CALLS: None

COMMONS REFERENCED: RRPLAN

MESSAGES GENERATED:

NUMBER OF NON-ZERO COEFFICIENTS EXCEEDS 8100. PROGRAM ABORTS.
NUMBER OF ROWS EXCEEDS 90. PROGRAM ABORTS.
NUMBER OF COLUMNS EXCEEDS 360. PROGRAM ABORTS.

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: SWAP

CALL STATEMENT: SWAP

DESCRIPTION:

This subroutine is the main control routine for the Fixed Charge Solution System which uses the SWIFT algorithm developed by Walker. The principal logic flow is controlled and executed from this procedure. Certain error check variables are initialized by the routine. When execution is completed or abnormally terminated, control returns to the MAIN routine.

CALLED BY: MAIN

CALLS: DOTPR,FORCE,INIT,LP,NEWA,OUT,SAVE,SETUA,SETUB,SIMPL

COMMONS REFERENCED: Unlabeled,SBASI,SWAPCO

MESSAGES GENERATED:

MATRIX OVERFLOW
NO FEASIBLE SOLUTION
Z COL IN BASIS. PROGRAM HALTS

PROGRAM SUMMARY SHEET

SUBROUTINE NAME:	SWITC
CALL STATEMENT:	SWITC
DESCRIPTION:	This subroutine calculates the objective value at the end of phase 1. If artificials remain at a positive level there is no feasible solution. If the solution is feasible then the variable costs are stored in the cost vector, the phase is set to 2 and the basis is transformed for phase 2 execution.
CALLED BY:	COMEI
CALLS:	BTRAN,DOTPR
COMMONS REFERENCED:	Unlabeled and SWAPCO
MESSAGES GENERATED:	None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: TINV

CALL STATEMENT: TINV

DESCRIPTION:

This subroutine performs a matrix reinversion under two conditions: (1) the current vector is full; (2) the last tolerance check has more than a negligible error. Tolerance checks are made every fifth iteration unless the last check shows appreciable error. If the last tolerance check showed appreciable error, tolerance is checked on every iteration and reinversion occurs when an intolerable error appears, i.e., when the absolute value of the differences between the previous cost and the current cost divided by the current cost exceeds .5.

$$(\text{ERROR} = \left| (\text{PRCOST} - \text{COST})/\text{COST} \right| > .5)$$

CALLED BY: LP,SIMPL

CALLS: BTRAN,DOTPR,PASSA,REINV

COMMONS REFERENCED: Unlabeled and SWAPCO

MESSAGES GENERATED: None

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: TORT

CALL STATEMENT: TORT

DESCRIPTION:

This subroutine checks that the number of transportation activities created during the expansion of the file associated with the source separation options and the paper market classification does not exceed 300. If more than 300 activities are created, a message is written and the run is stopped.

CALLED BY: MAIN

CALLS: None

COMMONS REFERENCED: RRPLAN

MESSAGES GENERATED:

MORE THAN 300 TRANSPORTATION ENTRIES DURING EXPANSION. PROGRAM ABORTS.

PROGRAM SUMMARY SHEET

SUBROUTINE NAME: ZCOLIN

CALL STATEMENT: ZCOLIN (BASIS,BETA,IOLD,INEW,M)

DESCRIPTION:

ZCOLIN is a special linear program routine which changes a starting BASIS. If a BASIS has a negative activity level in the initialization phase of the linear program algorithm then this routine is called. The activity levels are checked and the one with the largest negative value is replaced by a special column pointer "Z". The phase switch is reinitialized and control returns for a recheck of the new BASIS.

CALLED BY: SETUB

CALLS: None

COMMONS REFERENCED: SBASI

MESSAGES GENERATED: None

References

American National Standards Institute, American National Standard Programming Language FORTRAN, ANSI X3.9 1978, New York, 1978.

Berman, E.B., Use of RAMP (Recovery And Market Planning) For The Evaluation of Policy Issues in Resource Recovery, mimeo, 1976.

Berman, E.B., WRAP - A Model for Regional Solid Waste Planning: User's Guide, U.S. Environmental Protection Agency, SW-574, 1977.

Chapman, R.E., and E.B. Berman, The Resource Recovery Planning Model: A New Tool for Solid Waste Management, National Bureau of Standards, Special Publication 657, 1983.

Hensey, V., WRAP - A Model for Regional Solid Waste Planning: Programmer's Manual, U.S. Environmental Protection Agency, SW-573, 1977.

Hirsch, W.M., and G.B. Dantzig, "The Fixed Charge Problem," Naval Research Logistics Quarterly, Vol. 15 (1968), pp. 413-424.

Saunders, P.B., editor, Selected Assessment Strategies Applied to Short-Term Energy Models, National Bureau of Standards, NBSIR 83-2672, March 1983.

Walker, W.E., "A Heuristic Adjacent Extreme Point Algorithm for the Fixed Charge Problem," Management Science, Vol. 22 (1976), pp. 587-596.

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date (Yr. Mo. Day)	02. Summary prepared by (Name and Phone)	03. Summary action
83 07 19	Robert E. Chapman (301) 921-3855	New [x] Replacement [ ] Deletion [ ]
04. Software date (Yr. Mo. Day)	**05. Software title**	Previous Internal Software ID
83 03 24	Resource Recovery Planning (RRPLAN) Model	
06. Short title		**07. Internal Software ID**

08. Software type
- [ ] Automated Data System
- [x] Computer Program
- [ ] Subroutine/Module

09. Processing mode
- [ ] Interactive
- [x] Batch
- [ ] Combination

10. Application area

General:
- [ ] Computer Systems Support/Utility
- [x] Scientific/Engineering
- [ ] Bibliographic/Textual
- [x] Management/Business
- [ ] Process Control
- [ ] Other

Specific: Solid Waste Management

11. Submitting organization and address

Center for Applied Mathematics
National Bureau of Standards
Washington, D.C. 20234

12. Technical contact(s) and phone

Robert E. Chapman (301) 921-3855
Edward B. Berman (617) 631-8099

13. Narrative

RRPLAN identifies the most efficient means of processing a region's solid waste stream. The existence of substantial economies of scale in the construction and operation of the types of facilities considered, however, complicates the problem by introducing non-linearities into the objective function. RRPLAN uses a fixed-charge linear programming algorithm to deal with the two-parameter cost functions resulting from economies of scale. Due to these fixed charges, it is possible that the optimizer will converge on a local rather than global optimum. A heuristic post processor is therefore used to "force" the optimizer to examine other portions of the solution domain in hopes of finding the true optimum.

14. Keywords

Economic analysis; facility location; fixed-charge problem; mathematical programming; optimization; resource recovery; solid waste management

15. Computer manuf'r and model	16. Computer operating system	17. Programing language(s)	18. Number of source program statements
Sperry Univac	1100/82	FORTRAN 77	APPX 6500
19. Computer memory requirements	**20. Tape drives**	**21. Disk/Drum units**	**22. Terminals**
512 K Bytes	U9V/U9S (9 Track 1600 or 6250 FPI)		

23. Other operational requirements

24. Software availability
- Available [x]
- Limited [ ]
- In-house only [ ]

25. Documentation availability
- Available [x]
- Inadequate [ ]
- In-house only [ ]

26. FOR SUBMITTING ORGANIZATION USE

185-101

Standard Form 185
1974 July
U.S. Dept. of Commerce—NBS
(FIPS. Pub. 30)

INSTRUCTIONS

01. Summary Date. Enter date summary prepared. Use Year, Month, Day format: YYMMDD.

02. Summary Prepared By. Enter name and phone number (including area code) of individual who prepared this summary.

03. Summary Action. Mark the appropriate box for new summary, replacement summary or deletion of summary. If this software summary is a replacement, enter under "Previous Internal Software ID" the internal software identification as reported in item 07 of the original summary, and enter the new internal software identification in item 07 of this form; complete all other items as for a new summary. If a software summary is to be deleted, enter under "Previous Internal Software ID" the internal software identification as reported in item 07 of the original summary; complete only items 01, 02, 03 and 11 on this form.

04. Software Date. Enter date software was completed or last updated. Use Year, Month, Day format: YYMMDD.

05. Software Title. Make title as descriptive as possible.

06. Short Title. (Optional) Enter commonly used abbreviation or acronym which identifies the software.

07. Internal Software ID. Enter a unique identification number or code.

08. Software Type. Mark the appropriate box for an Automated Data System (set of computer programs), Computer Program, or Subroutine/Module, whichever best describes the software.

09. Processing Mode. Mark the appropriate box for an Interactive, Batch, or Combination mode, whichever best describes the software.

10. Application Area.
General: Mark the appropriate box which best describes the general area of application from among:

Computer Systems Support/Utility	Process Control
Management/Business	Bibliographic/Textual
Scientific/Engineering	Other

Specific: Specify the sub-area of application; e.g.: "COBOL optimizer" if the general area is "Computer Systems Support/Utility"; "Payroll" if the general area is "Management/Business"; etc. Elaborate here if the general area is "Other."

11. Submitting Organization and Address. Identify the organization responsible for the software as completely as possible, to the Branch or Division level, but including Agency, Department (Bureau/Administration), Service, Corporation, Commission, or Council. Fill in complete mailing address, including mail code, street address, city, state, and ZIP code.

12. Technical Contact(s) and Phone: Enter person(s) or office(s) to be contacted for technical information on subject matter and/or operational aspects of software. Include telephone area code. Provide organization name and mailing address, if different from that in item 11.

13. Narrative. Describe concisely the problem addressed and methods of solution. Include significant factors such as special operating system modifications, security concerns, relationships to other software, input and output media, virtual memory requirements, and unique hardware features. Cite references, if appropriate.

14. Keywords. List significant words or phrases which reflect the functions, applications and features of the software. Separate entries with semicolons.

15. Computer Manufacturer and Model. Identify mainframe computer(s) on which software is operational.

16. Computer Operating System. Enter name, number, and release under which software is operating. Identify enhancements in the Narrative (item 13).

17. Programing Language(s). Identify the language(s) in which the software is written, including version; e.g., ANSI COBOL, FORTRAN V, SIMSCRIPT II.5, SLEUTH II.

18. Number of Source Program Statements. Include statements in this software, separate macros, called subroutines, etc.

19. Computer Memory Requirements. Enter minimum internal memory necessary to execute software, exclusive of memory required for the operating system. Specify words, bytes, characters, etc., and number of bits per unit. Identify virtual memory requirements in the Narrative (item 13).

20. Tape Drives. Identify number needed to operate software. Specify, if critical, manufacturer, model, tracks, recording density, etc.

21. Disk/Drum Units. Identify number and size (in same units as "Memory"—item 19) needed to operate software. Specify, if critical, manufacturer, model, etc.

22. Terminals. Identify number of terminals required. Specify, if critical, type, speed, character set, screen/line size, etc.

23 Other Operational Requirements. Identify peripheral devices, support software, or related equipment not indicated above, e.g., optical character devices, facsimile, computer-output microfilm, graphic plotters.

24. Software Availability. Mark the appropriate box which best describes the software availability from among: Available to the Public, Limited Availability (e.g.: for government use only), and For-In-house Use Only. If the software is "Available", include a mail or phone contact point, as well as the price and form in which the software is available, if possible.

25. Documentation Availability. Mark the appropriate box which best describes the documentation availability from among: Available to the Public, Inadequate for Distribution, and For In-house Use Only. If documentation is "Available", include a mail or phone contact point, as well as the price and form in which the documentation is available, if possible. If documentation is presently "Inadequate", show the expected availability date.

26. For Submitting Organization Use. This area is provided for the use of the organization submitting this summary. It may contain any information deemed useful for internal operation.

GPO : 1974 O - 554-655

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET *(See instructions)*	1. PUBLICATION OR REPORT NO.	2. Performing Organ. Report No.	3. Publication Date

4. TITLE AND SUBTITLE

Program Documentation for the Resource Recovery Planning Model

5. AUTHOR(S)

Edward B. Berman, Robert E. Chapman and Howard K. Hung

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Office of Recycled Materials
National Measurement Laboratory
National Bureau of Standards
Washington, D. C. 20234

10. SUPPLEMENTARY NOTES

[X] Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

The Resource Recovery Planning (RRPLAN) model is designed with three purposes in mind. First and foremost, is the ability to generate a preferred (optimal) plan for resource recovery. Second, is the capability to evaluate a scenario specified by the decision maker for technical and economic feasibility. Third, is its use as a tool to facilitate the decision making process by providing answers to many what-if questions through an in-depth sensitivity analysis. In order to find the optimal solution, however, it is necessary to address three interdependent issues. The first two issues are concerned with the siting and sizing of solid waste management facilities, whereas the third concerns how to allocate commodities among the various facilities and potential markets. The existence of substantial economies of scale in the construction and operation of the types of facilities considered complicates the problem by introducing non-linearities into the objective function. RRPLAN uses a fixed-charge linear programming algorithm to deal with the two-parameter cost functions resulting from economies of scale. A heuristic post processor is also used to "force" the optimizer to systematically examine the solution domain in hopes of finding the true optimum. The model is written in FORTRAN and complies with the American National Standards Institute X3.9-1978 standard for that language.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

Economic analysis; facility location; fixed-charge problem; mathematical programming; optimization; resource recovery; solid waste management.

13. AVAILABILITY

[X] Unlimited
[ ] For Official Distribution. Do Not Release to NTIS
[ ] Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
[ ] Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

15. Price

USCOMM-DC 6043-P80