



**OPTIMIZATION OF
EARTH DISTRIBUTION
IN LINEAR EXCAVATION
PROJECTS**

by

Juan Pablo Vásquez

December, 1991



**CONSTRUCTION
ENGINEERING &
MANAGEMENT**
PURDUE ■ UNIVERSITY

**Division of Construction
Engineering and Management
School of Civil Engineering
Purdue University
West Lafayette, Indiana 47907**

**OPTIMIZATION OF EARTH DISTRIBUTION IN
LINEAR EXCAVATION PROJECTS**

**AN INDEPENDENT RESEARCH STUDY
SUBMITTED TO THE FACULTY OF THE**

**SCHOOL OF CIVIL ENGINEERING
PURDUE UNIVERSITY**

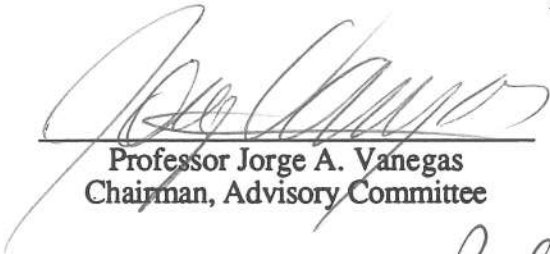
BY

JUAN PABLO VASQUEZ

**IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN CIVIL ENGINEERING**

DECEMBER 1991

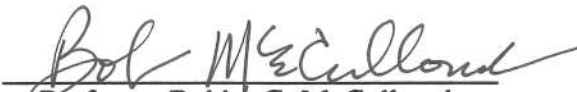
APPROVED:



Professor Jorge A. Vanegas
Chairman, Advisory Committee



Professor Daniel W. Halpin
Member, Advisory Committee



Professor Bobby G. McCullouch
Member, Advisory Committee

*This Research Project and
my Master of Science degree
are dedicated with love to
my future wife, Angie.*

ABSTRACT

This independent research study was developed in order to provide an alternative to the conventional methods currently used in planning and estimating earthmoving operations for linear projects. The original program was created by Mr. Robert S. Toomy at Georgia Institute of Technology in 1984 using BASIC. This program uses a section to section matching algorithm, and a variation of the classic transportation algorithm to balance in the most efficient way the distribution of earth among cuts, borrow pits, fills and landfills.

The new program developed in this research project enhances Mr. Toomy's original program by updating the code to QuickBASIC, and by making modifications in the program structure and in some of the calculation methods used. It also introduces new features. All these improvements were made focusing on increasing the speed, flexibility and reliability of the program, while providing the user with a friendly interface.

TABLE OF CONTENTS

ABSTRACT	i
TABLE OF CONTENTS.....	ii
LIST OF FIGURES.....	iii
1.0 INTRODUCTION.....	1
1.1 Background	1
1.2 Application	1
2.0 CURRENT APPROACH TO EARTH DISTRIBUTION.....	3
3.0 BASIC PROGRAM STRUCTURE	6
3.1 Data Input.....	6
3.2 Program Execution.....	9
<i>Section-to-Section Matching Process</i>	
<i>Transportation Algorithm</i>	
3.3 Output.....	14
4.0 PROGRAM ENHANCEMENTS	16
4.1 Structure Improvements	16
4.2 Calculations Improvements.....	18
<i>Different Soil Properties</i>	
<i>Volume Calculations</i>	
<i>Volume Results</i>	
<i>Unusable Volumes</i>	
<i>Incomplete Fills and Cuts</i>	
4.3 File Input and Output Improvements.....	25
4.4 Graphical Output.....	26
4.5 Screen Output.....	26
4.6 Printed Output.....	27
4.7 Compilation	27
5.0 CONCLUSIONS AND RECOMMENDATIONS.....	28
6.0 REFERENCES.....	31

APPENDIX A: SAMPLE OF THE ORIGINAL PRINTOUT

APPENDIX B: SAMPLE #1 OF THE NEW PRINTOUT

APPENDIX C: SAMPLE #2 OF THE NEW PRINTOUT

APPENDIX D: PROGRAM FLOW CHART

APPENDIX E: PROGRAM CODE

APPENDIX F: PROGRAM DISKETTE

LIST OF FIGURES

Figure 1.1 – Generic Layout for a Linear Excavation Project.....	2
Figure 2.1 – Volume vs. Distance Graph.....	3
Figure 2.2 – Ground Profile and Mass Diagram.....	4
Figure 3.1 – Section-to-Section Matching Process	10
Figure 3.2 – Source-Destination Distance Matrix	12
Figure 3.3 – Technology Table for a Transportation Problem.....	12
Figure 3.4 – Simulated Screen Display Example	15

1.0 INTRODUCTION

The purpose of this independent research project is to develop a new version of a program for optimization of earth distribution in linear excavation projects using the Microsoft Quick-Basic programming language, using the program previously developed by Mr. Robert S. Toomy at the Georgia Institute of Technology in 1984 as a starting point (Toomy 1984).

1.1 Background

The program made by Mr. Toomy presents an alternative to the traditional earth distribution method based on the mass diagram, which is very limited and sometimes inaccurate. Mr. Toomy's approach uses a microcomputer and the linear programming techniques, specifically the classical transportation algorithm. This program originally used the IBM Basic version that was available in 1984. Thus, it is a very slow program. For large projects, this problem could render the program inadequate.

The result of this study is a new and improved program that makes use of the more powerful capabilities of Quick-Basic, such as its ability to compile. Also, by introducing modifications in the program structure, the speed of the application will be considerably increased. Also its flexibility and capabilities were improved by introducing modifications in the input and output procedures, calculation processes and results.

1.2 Application

This program is designed to be used in any linear project that involves earthwork along its axis, such as a highway project. Typically, these linear projects involve successive cuts of ground and fills of terrain depressions. Therefore, there is a need to balance the volumes of cuts and fills in the most efficient way. In addition, to compensate any difference between the total cut and fill volumes, some projects have borrow pits from which to obtain earth for fills, and landfills to place excess earth from

cuts. In these cases, when the project is large or there are special conditions on the job site, it may be less expensive to place the earth from a cut in a landfill, rather than in a fill, or to furnish a fill with earth coming from a borrow pit, instead of a cut. Thus, it becomes necessary to optimize the complete project taking into account cuts, fills, borrow pits and landfills. A schematic example of the layout of a linear earth moving work is shown in figure 1.1.

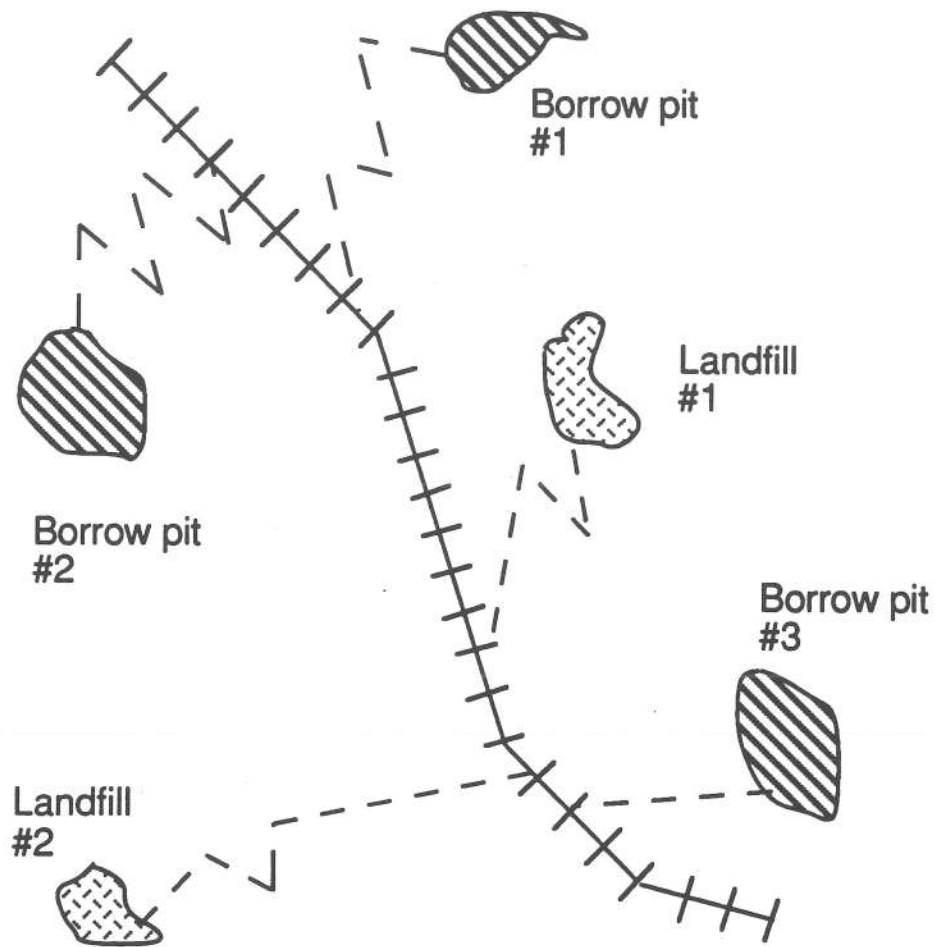


Figure 1.1.- Generic Layout for a Linear Excavation Project

2.0 CURRENT APPROACH TO EARTH DISTRIBUTION

Typically in a linear earth moving project, to be able to begin the optimization of earth distribution between cuts, fills, borrow pits and landfills, it is necessary to start by obtaining the following data:

- (1) *volumes of cut and fills.*— after the land survey is made, the global volumes of cut and fills are calculated for the project, based on the survey measurements;
- (2) *volumes of cut and fills for each station.*— the project is divided into equal length sections called stations (typically 100 ft.), calculating the volume of cut or fill for each;
- (3) *swell and shrink factors.*— the geotechnical studies determine the soil properties, specifically the swell and shrink factors;
- (4) *volume versus distance graph.*— with all these data, a graph of volume versus distance is made to give a general idea about the earth moving that needs to be made, as shown in Figure 2.1.

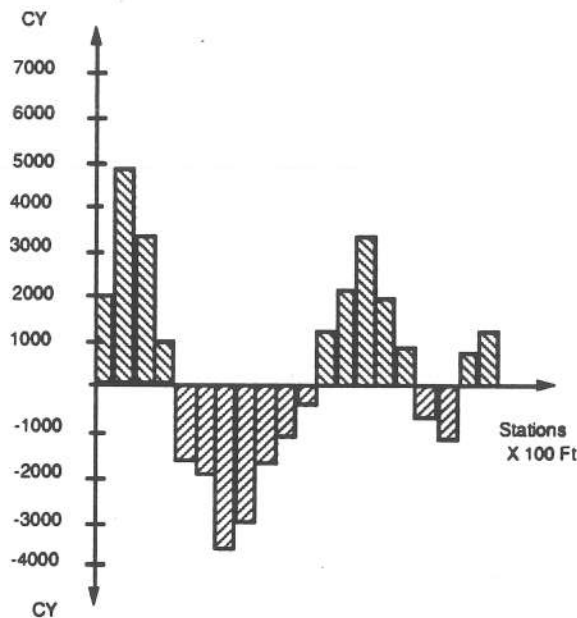


Figure 2.1.— Volume vs. Distance Graph

Traditionally, the method used for the earthmoving distribution is the **Mass Diagram**. From the volume versus distance graph, a cumulative volume graph is made starting from the first station through the end, taking into account the distances involved. This is shown in Figure 2.2.

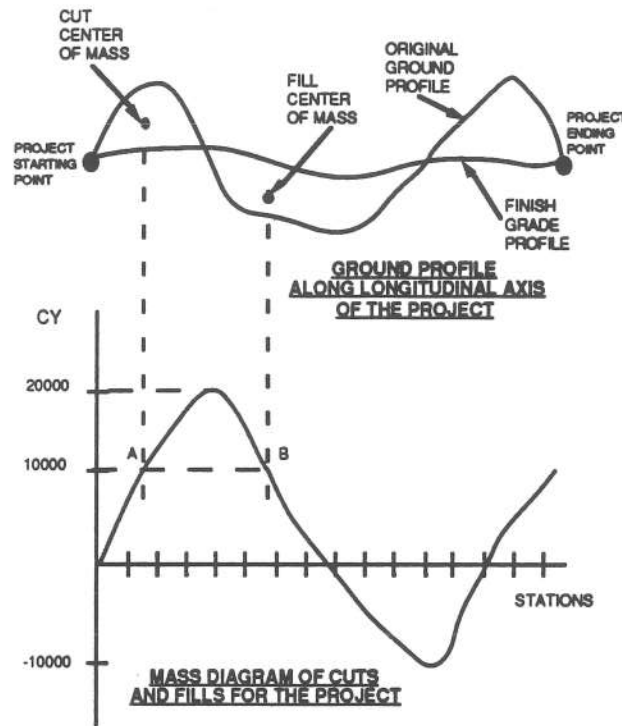


Figure 2.2.- Ground Profile and Mass Diagram
 [Developed from (Merritt 1983)]

The cumulative volume is plotted in the vertical axis against the distance measured along the center line of the project, typically in stations 100 ft. apart, along the horizontal axis. Swell factors are applied to the cuts to obtain the loose volume that needs to be transported, and shrink factors are applied to the fill volumes to determine the bank volume needed. Equations II.1 and II.2 show the calculation of these volumes.

$$V_1 = V_b * (1 + \text{Swell Factor}) \quad (\text{II.1})$$

$$V_f = V_b * (1 - \text{Shrink Factor}) \quad (\text{II-2})$$

V_b = Bank volume (cut volume in natural state)

V_1 = Loose volume (volume after swell)

V_f = Fill volume (volume after shrinkage)

The Mass Diagram helps to calculate the volume of earth and the distance to haul it, allowing to distinguish between the *free haul*, i.e., which does not have an extra cost, and the *overhaul*, which is billed as an extra cost for the project because the transportation distance exceeds the one agreed for the free haul.

The transportation distance is determined by the span between centers of mass of the group of cuts and fills used to balance each other, i.e., the volume of a cut is used as the volume for a fill. This span is measured in the Mass Diagram between the points A and B, each of which is located at the midpoint between a peak of the profile and the X axis, or a valley of the profile and the X axis, as shown in Figure 2.2.

The Mass Diagram method assumes that the balance between cuts and fills for each station will be made sequentially and cumulatively from the first station to the last station of the project, i.e., accumulated cuts go to fills as these are encountered. The problem is that sometimes, because of cost factors, it may be necessary for example, to import material from borrow pits, or to export excess material from cuts to landfills. This method does not allow to account easily for these deviations from the basic approach. Also, since this approach to earth distribution is primarily manual, it is time consuming, and furthermore, given the simplistic nature of the assumptions, the results may not be very accurate.

To overcome these limitations, the approach used by the program developed in this investigation uses an automatic section to section matching for cut and fill volumes that are close together. Also, it distributes the remaining volumes taking into account available borrow pits and landfills, by using a special variation of the classic transportation algorithm to obtain an efficient solution. The use of an automated method will help reduce earthwork costs by developing an efficient distribution, and since it allows for fast calculations, several scenarios with different assumptions may be tested to obtain an optimal solution.

3.0 BASIC PROGRAM STRUCTURE

This section summarizes the basic elements of the program structure.

3.1 Data Input

The program begins with the data input module. Data for a project can be introduced either directly using the keyboard, or imported from an existing file that was previously created with a text editor, or created with the program. This module of the program allows the review and modification of the data input as many times as it is necessary. The program will go beyond this module only when the user decides that all the data input is correct.

The data required for this module is:

- **PRONAMES** = Project Name; the name of the project.
- **NUMSEC** = Number of Sections, i.e., total number of sections in which the project is divided.
- **SECLEN** = Section Length, i.e., the distance in feet between the starting point of a section and its endpoint along the project axis. The project needs to be divided always into sections of equal length. Typically, the section length is 100 ft., but the program has the capability of allowing the user to determine different lengths.
- **VOL(I)** = Volume of each section, i.e., total volume in cubic yards of a given section. This volume has to be positive for the cuts and negative for the fills. These volumes are either provided by the survey team, or calculated beforehand from the data provided by the surveyors.
- **SWELL(I)** = Swell factor, i.e., factor to change from a bank volume to a loose volume. This factor is always is greater than 1. As an example, if the volume increases 15% going

from 1 CY when it is in bank state to 1.15 CY when it is loose, a swell factor of 1.15 must be entered.

- **SHRINK(I)** = **Shrink** factor, i.e., factor to change from a bank volume to a compacted fill volume. This factor is always less than 1. As an example, if the volume decreases 15% going from 1 CY when it is in bank state to 0.85 CY after it is compacted, a shrink factor of 0.85 must be entered.
- **UNITPRICE** = **Unit Price**, i.e., the unit price in dollars required to haul 1 CY of material a distance of 100 feet.
- **NBP** = **Number of Borrow Pits**, i.e., the number of borrow pits that are available for importing material when there is no more material available from the cuts to complete the fills. When the data was acquired from a file and the user wishes to increase the number of borrow pits, he/she can increase them up to 6 additional borrow pits. This slight limitation is due to the dimensioning the variables for the swell and shrink factors when the data is acquired from an existing file.
- **CBP(I)** = **Capacity of the Borrow Pits**, i.e., the volume in cubic yards of each borrow pit available for the project.
- **BPD(I)** = **Borrow Pit Distance**, i.e., the distance in feet between the borrow pit and the section where the trucks enter the project for dumping the borrowed material.
- **ES(I)** = **Entering Section**, i.e., the section number at which the truck can enter to the project coming from the borrow pit. The total distance between a borrow pit and the destination fill section where the import material will be dumped is equal to the distance between this section and the entering section plus the distance between the entering section and the borrow pit.
- **NLF** = **Number of Landfills**, i.e., the number of landfills that are available for dumping excess material when there are no more fills to place material coming from the cuts.

- **CLF(I)** = Capacity of the Landfills, i.e., the volume in CY of each landfill. When the data is acquired from an existing file, the number of landfills may be increased without any limitations.
- **LFD(I)** = Landfill Distance, i.e., the distance between the landfill and the departing section.
- **DS(I)** = Departing Section, i.e., the section number at which the truck can exit the project to go to the landfill. The total distance between any cut section and a landfill is equal to the distance between this section and the entering section plus the distance between the entering section and the landfill.

At the beginning of the program, the user will be asked if the data will be entered from a file or using the keyboard. If a file is the choice, the program will ask first the drive source, and then the file name. The drive source must be entered as " B: ". After the data is read from the file, it is possible to display it in the screen for review. All the data imported from an existing file may be modified. The main limitation is that the number of sections must remain the same. Also, the number of available borrow pits may be increased up to 6 additional ones. The number of landfills does not have any limitations.

The reason for these limitations is that with dimensioned variables all existing data will be lost if any attempt to redimension a previously dimensioned variable is made. For example, the number of sections cannot be increased because the variables that contain section data are dimensioned to a number equal to the number of existing sections in the data imported from an existing file. Also, while importing the data from an existing file, the swell and shrink variables are dimensioned to a quantity equal to the number of sections plus the number of borrow pits plus six, in order to allow for the possibility of increasing the number of borrow pits up to six. However, these limitations can be overcome by modifying the number of sections or borrow pits directly in the existing file using a text editor.

If the key board input is the choice, the user will be asked to enter the data as needed. The general data of the project is asked first: the project name, the length of the sections, the number of sections, the unit price and number of borrow pits. The number of borrow pits is requested at this time because it is needed to dimension the swell and shrink factor variables up to a quantity equal to the number of sections plus the number of borrow pits. After this is done, the volumes of each section must be entered one by one. Then, the data on swell and shrink factors is introduced. Here the program asks for the number of zones with different kind of soils. If there are two non-adjacent zones with the same type of soil, they count as different ones. For example, if a 10,000 ft. project there is type A soil in the first 3,000 feet, type B soil in the next 5,000 feet, and type A soil again in the last 2,000 feet, the user must enter a 3 as the number of different zones.

After the soil factors are entered, or after data is read from an existing file, the program will ask for any section that has a percentage of unusable soil. For this, the number of sections is inquired first, and then, one by one the percentage of unusable material must be entered. For example, if 20 % of the volume of a section cannot be used, 20 must be entered. After this is done, and if the data were not acquired from a file, the borrow pits and landfills are entered one by one.

At all times, after the input of any type of data is complete, the program will offer the possibility to display it for review, in order to check the data and make any necessary corrections.

After all data is input and all the corrections and modifications are made, these data can be saved in a file for further use, and also printed to keep a hard copy record.

3.2 Program Execution

Using the project shown in Figure 1.1 as an example, this section will provide a general description of the program execution.

The borrow pits and landfills are located on both sides of the project and each one has an access route to link it with the project at one of the project sections. The material available from cuts and

borrow pits will be used for fill sections (i.e., terrain depressions). The excess material or the cut volumes that are too expensive to be placed in the fill sections will be hauled to and dumped at the landfills. A schematic view of such a project might look as the example shown in Figure 2.1. The volumes above the grade level of the project represent the cuts, and the volumes below the grade level of the project represent the fill sections.

Section-to-Section Matching Process

After all the data is input, the first step in the program is to multiply each cut and borrow pit volume by its correspondent shrink factor. The total cut and fill volumes are then compared to know if extra borrow pit or landfill capacity is needed. Next, the section-to-section matching process takes place. Figure 3.1 shows the section-to-section matching process step-by-step.

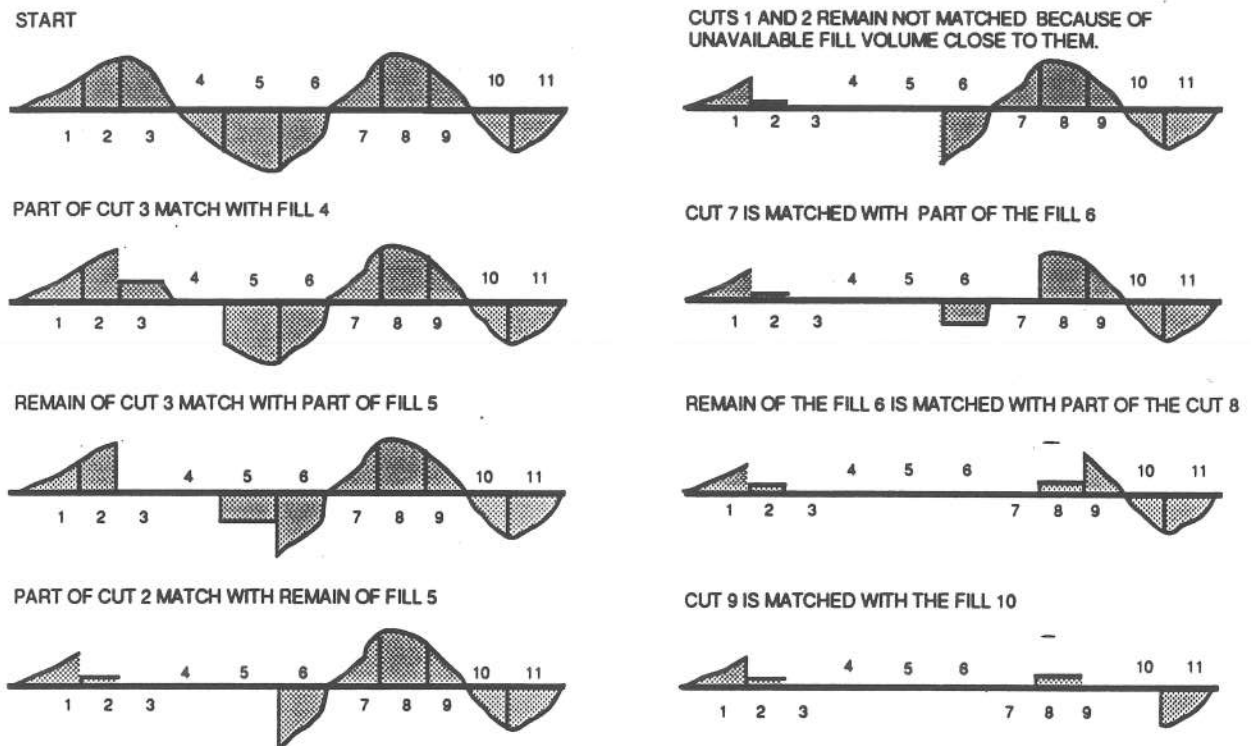


Figure 3.1.- Section-to-Section Matching Process

In this process, the program first identifies a grade point, i.e., a point where the section volumes change from a cut to a fill section. After a grade point is identified the program starts to compare the cut volumes from one side of the grade point, with the fill volumes in the other.

If a cut section volume is greater than the closest fill section, the volume matched is equal to the fill volume. Otherwise, the volume matched will be equal to the cut volume. Then, both volumes are reduced by the matched volume amount. This operation turns the smaller volume, between the cut and fill, to zero leaving a remaining volume in the larger one. If the remaining volume is a cut, the program will compare it with the next fill volume. Otherwise, it will search for the next cut volume.

This process will continue until all cuts or fills are finished. Then, the program will look for the next grade point and start again the process. A cut volume is left as excess material when there are no other fill volume available in the adjacent group of fill sections, and viceversa for any additional required materail for a fill section. The result for each volume matching step will be the volume (i,j), which means: shipped from a source (i) to a destination (j), and the distance (i,j) involved in the process.

At the end of the section-to-section matching process, some cut and fill volumes may remain without being used, and some will be completely matched and equal to zero. The remaining unused volumes will be taken into account for the transportation algorithm, while the others are discarded.

Transportation Algorithm

The remaining cut and fill sections, together with the borrow pits and landfills, are distributed using a variation of the classic transportation algorithm. In using this algorithm, the cuts and borrow pits are the sources, and the fills and landfills are the destinations. The project data will be arranged in a matrix containing the distances between the sources and the destinations, as shown in Figure 3.2. The distribution will be made in order to minimize the total transportation cost, which is proportional to the distance, between the source locations and the supply locations.

Transportation problems require a very large number of variables and constraints. Therefore the use of a standard simplex method will not be very efficient. Fortunately, the mathematical characteristics of the transportation model are so special, that its structural properties can be exploited in order to obtain a more efficient method of solution. One of its characteristics is that most of the coefficient in the constraints are zeros, and those which are not, appear in a recurrent pattern, as shown in Figure 3.3. This characteristic allows to develop a streamlined algorithm based in the simplex method. As a result, considerably savings in computational efforts are obtained.

		SOURCES								
		1	2	3	4	5	6	7	8	9
DESTINATIONS	1	c_{ij}	...							
	2							
	3									
	4									
	5									
	6									
	7									
	8									
	9									

Figure 3.2.- Source-Destination Distance Matrix

		Shipping Activities																
		x_{11}	x_{12}	x_{13}	...	x_{1n}	x_{21}	x_{22}	x_{23}	...	x_{2n}	...	x_{m1}	x_{m2}	x_{m3}	...	x_{mn}	
Supplies	1	1	1	1	...	1												$=S_1$
	2						1	1	1	...	1							$=S_2$
					\vdots
	m													1	1	1	...	1
Demands	1	-1					-1						-1					$=-D_1$
	2		-1					-1						-1				$=-D_2$
	...			-1					-1						-1			\vdots
	n					-1					-1						-1	$=-D_n$
		C_{11}	C_{12}	C_{13}	...	C_{1n}	C_{21}	C_{22}	C_{23}	...	C_{2n}	...	C_{m1}	C_{m2}	C_{m3}	...	C_{mn}	Min.

Figure 3.3.- Technology Table for a Transportation Problem
[Developed from (Wagner 1975)]

Hillier and Lieberman, and Wagner give a more detailed explanation of the transportation algorithm (Hillier & Lieberman 1986; Wagner 1975). The general description of the algorithm is as follows:

- (1) *Initialization step*: Construct an initial basic feasible solution. This means to find any solution that satisfies all the constraints, but not necessarily the most efficient. Go to the optimality test.
- (2) *Iterative step*: Here the program changes the initial solution redistributing the basic variable X_{ij} (amount shipped from source i to destination j) which has the highest negative rate (in absolute value) to modify the total cost Z . That means that the variable X_{ij} will be redistributed in order to obtain the higher reduction in the cost.
- (3) *Optimality test*: Derive the rate variables to obtain the new rates. If all the rates are bigger or equal to zero, the optimal solution has been found. Otherwise, the iterative step should be repeated.

The specific transportation algorithm used by Mr. Toomy in the original program, was directly taken from the book "Basic Programs for Production and Operations Management" of Pantumsinchai, Hassan and Gupta (Pantumsinchai et al. 1983). This algorithm is described next.

We define

- m = Number of supply points.
- n = Number of demand points.
- S_i = Supply amounts.
- D_j = Destination amounts.

having

- $i = 1, 2, \dots, m$
- $j = 1, 2, \dots, n$

Also, we define:

C_{ij} = The unit transportation cost (or distance) from the supply location i to the demand location j

X_{ij} = The amount to be shipped from the supply location i to the demand location j .

The problem is to minimize the total cost of the operation, that is:

$$\text{Minimize } Z = \sum_{i=1}^m \sum_{j=1}^n C_{ij} * X_{ij}$$

Subject to the following constraints:

- (1) Each source must be equal to the total amount shipped from it.

$$\sum_{j=1}^n X_{ij} = S_i$$

For $i = 1, \dots, m$

- (2) Each destination must be equal to the total amount shipped to it.

$$\sum_{i=1}^m X_{ij} = D_j$$

For $j = 1, \dots, n$

- (3) All the shipments must be equal or greater than 0.

$$X_{ij} \geq 0 \text{ for all } i \text{ and } j.$$

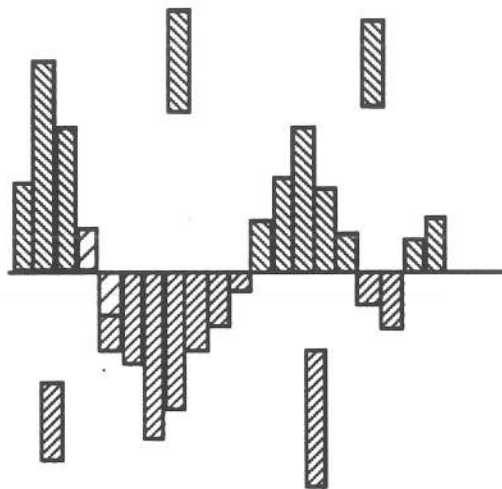
3.3 Output

The results of the program execution are shown by the program in two different ways. First, there is a graphical display on the screen made with scaled bars representing the volumes on the vertical axis, and the distances on the horizontal axis. The volumes are cuts when the bars are positive,

and fills when they are negative. The borrow pits are represented by scaled bars located in the upper part of the graph and the landfills by those located in the bottom. At the beginning of the graphical display, all the rectangles are colored green. The user must press ENTER at each step of the earth distribution process. The cuts and borrow pits will become black when emptied, and the fills and landfills will turn blue when filled. This graph will show step by step how the earth distribution is made from one source to a destination, providing the user at the end, with a good grasp of the overall earth distribution for the project. In each step, a printout in the screen will indicate the number of the source (i.e., cut section or borrow pit), and the number of the destination (i.e., fill section or landfill). Also the volume involved in each step will be also indicated, as shown in Figure 3.4.

EARTH DISTRIBUTION PLAN: LAWRENCE PROJ.

CUT 4 TO FILL 5 AMOUNT IS 500 CY



PRESS ENTER TO CONTINUE ?

Figure 3.4.- Simulated Screen Display Example

The second way to see the results of the program execution is via a printed output, which provides a listing of the results. These results are sorted in the three types of distribution that are possible:

- cut section to fill section,
- borrow pit to fill section, and
- cut section to landfill area.

These printouts have the number of the sections, borrow pits or landfills that are exchanging volumes, the distance involved in the process and the shrink and loose volumes hauled. Also, at the end of the printout, the total cost, the total loose volume and the total shrink volume are given. When there are remaining cut or fill sections, which were not matched in the process, a list of them and their remaining volumes is printed.

4.0 PROGRAM ENHANCEMENTS

This study enhanced the original program by Mr. Toomy in several ways. This section highlights each enhancement, and provides an explanation and description of why and how they were made.

4.1 Structure Improvements

The use of Basic 7.0 and Quick Basic 4.5 makes new functions and features available. First, the line numbers were removed due to the use of structured loops and subroutines. Because some parts of the original program are very entangled, it was necessary to keep some line numbers as labels.

The IF THEN and GOTO loops were replaced by the IF THEN, ELSE, ELSEIF and END IF loops. The use of these loops results in a structured programming approach that runs smoothly and faster, permitting an easy understanding of the code while reading the program for further corrections.

As an example, the following loop

```
3040 IF SO=DXO THEN 3200
3050 IF SO>DXO THEN 3130
3060 M=M9+1
3070 N=N9
```

```

3080 S(M)=DXO-S0
3090 FOR J=1 TO N
3100 C(M,J)=0
3110 NEXT J
3120 GOTO 3220
3130 M=M9
3140 N=N9+1
3150 D(N)=SO-DOX
3160 FOR I=1 TO M
3170 C(I,N)=0
3180 NEXT I
3190 GOTO 3220
3200 M=M9
3210 N=N9
3220 M1=M+N-1

```

was replaced by

```

IF SO < DXO THEN
    M=M9+1
    N=N9
    S(M)=DXO-S0
    FOR J=1 TO N
    C(M,J)=0
    NEXT J
ELSEIF SO > DXO THEN
    M=M9
    N=N9+1
    D(N)=SO-DOX
    FOR I=1 TO M
    C(I,N)=0
    NEXT I
ELSE
    M=M9
    N=N9
END IF
M1=M+N-1

```

Furthermore, the availability of the functions ON ERROR GOTO and SELECT CASE, helped to create a subroutine for error trapping and handling. The ON ERROR GOTO function is placed next to a function that might potentially cause an error, e.g., a file input, file output or a printing output. When an error is detected by the ON ERROR GOTO function, the error handler subroutine is triggered. By using this subroutine, a wider variety of errors can be handled because the handler has a list of the possible errors. The number of the error is contained on the ERR variable. With this number, the handler subroutine uses the SELECT CASE function to print the corresponding error message and the solution for it. This allows a continuous operation of the program without interruptions and the potential loss of data. This error handling is especially critical when running the compiled program.

4.2 Calculations Improvements

Several improvements were made to the calculation procedures.

Different Soil Properties

Because it is possible to have different soil properties in a large project, the possibility to use different swell and shrink factors was added to this program. The program first asks for the number of zones with different soil properties in the project. Then, the soil properties and the boundary sections (beginning and end) for each zone are entered. The program automatically distributes the swell and shrink factors among the volumes of the sections within the boundaries, saving time in the data input.

To achieve this, the following algorithm was created:

```
CLS:LOCATE 12,12
INPUT " HOW MANY ZONES WITH DIFFERENT SOIL PROPERTIES DO THEPROJECT
HAVE ="; NPR
DIM GSHRINK(NPR),GWELL(NPR),START(NPR),XEND(NPR)
CLS
FOR I = 1 TO NPR
CLS
LOCATE 12,12
PRINT"WHICH SECTIONS COMPRISE SOIL TYPE # ("; I;")"
```



```

- LOCATE 14,12
- INPUT " NUMBER OF THE FIRST SECTION (START)=";START(I)
  LOCATE 16,12
  INPUT " NUMBER OF THE LAST SECTION (END)=";XEND(I)
  LOCATE 18,12
  INPUT " SHRINK FACTOR ( e.i 0.80 for 20% shrinkage from bankvol.) ="; GSHRINK(I)
  LOCATE 20,12
  INPUT " SWELL FACTOR ( e.i. 1.20 for 20% swell from bank vol.) ="; GSWELL(I)
  CLS
  NEXT I
  FOR I=1 TO NPR
  FOR K= START(I) TO XEND(I)
  SHRINK(K) = GSHRINK (I)
  SWELL(K) = GSWELL (I)
  NEXT K
  NEXT I

```

Volume Calculations

The original program made the earth distribution calculations using bank volumes. This implied that the fill volumes had to be divided by the shrink factor in order to match them with the cut volumes. Nevertheless, when having different shrink and swell factors, this approach is not valid anymore because the cuts will shrink in different proportions for different soils properties. Therefore, in the new version the earth distribution calculations are made in shrink volumes, which means that cut sections and the borrow pits are multiplied by their own shrink factors before any calculation is made, obtaining in this way the compacted volume that will complete the fill sections.

As a consequence of this change, the following routine

```

1290 FOR G=1 TO NUMSEC
1300 IF VOL(G) >= 0 THEN TCUT=TCUT+VOL(G) : GOTO 1320
1310 VOL(G)=INT(VOL(G)/SHRINK):SVOL(G)=VOL(G):TFILL=TFILL- VOL(G)
1320 SVOL(G)=VOL(G):IF SVOL(G) > BIGVOL THEN BIGVOL=SVOL(G)
1330 NEXT G

```

was changed for:

```
FOR G=1 TO NUMSEC
IF VOL(G) >= 0 THEN
    VOL(G)=INT(VOL(G))*SHRINK(G)
    TCUT = TCUT + VOL(G)
ELSE
    TFILL = TFILL - VOL(G)
END IF
SVOL(G) = VOL(G)
IF SVOL(G) > BIGVOL THEN BIGVOL = SVOL(G)
NEXT
```

Notice how the positive volumes (cuts) are now each multiplied by their corresponding shrink factor. Furthermore, the following routine was added in order to multiply the borrow pit volumes by their shrink factor.

```
FOR G=1 TO NBP
CBP(G) = CBP(G)*SHRINK(G+NUMSEC)
TBPIT=TBPIT+CBP(G)
NEXT
```

Because this last routine is placed in the program after the data is already saved, the capacity of the borrow pits CBP is saved in the same unit they were entered, i.e., in bank volume.

Also, the following counters were modified in order to use the swell factor of each section while calculating the total cost of the haulage:

```
2250 FOR C= 1 TO NUMSEC
2260 FOR F=1 TO NUMSEC
2270 IF XA(C,F)=0 THEN 2290
2280 TSCYS=TSCYS+XA(C,F)*SWELL*ABS(C-F)
2290 NEXT F
2300 NEXT C
```

They were changed to:

```
FOR C=1 TO NUMSEC
FOR F=1 TO NUMSEC
IF XA(C,F) <> 0 THEN
    TSCYS=TSCYS+XA(C,F)*SWELL(C)/SHRINK(C)*ABS(C-F)*UNITPRICE/100
    TOTVOL = TOTVOL+XA(C,F)*SWELL(C)/SHRINK(C)
    TOTSHRVOL=TOTSHRVOL+XA(C,F)
END IF
```

The following one:

```
5480 FOR L=1 TO M1
5490 IF X(L) = 0 THEN 5510
5495 I=A(K):J=B(L):IF C(I,J)=90000! THEN 5510
5500 TSCYS=TSCYS+X(L)*SWELL*C(I,J)
5510 NEXT L
```

was changed to

```
548 FOR L=1 TO M1
IF X(L)<>0 THEN
    I=A(L):J=B(L)
    IF C(I,J) <> 90000! THEN
        IF I <= M8 AND J<= N8 THE SV=CLOC(I)
        IF I > M8 I <= M9 AND J <= N8 THEN SV=I-M8+NUMSEC
        IF I <=M8 AND J > N8 AND J <= N9 THEN SV=CLOC(I)
        IF I <= M9 AND J <= N9 THEN
            TSCYS=TSCYS+X(L)*SWELL(SV)/SHRINK(SV)*C(I,J)*UNITPRICE/100
            TOTVOL = TOTVOL+X(L)*SWELL(SV)/SHRINK(SV)
            TOTSHRVOL=TOTSHRVOL+X(L)
        END IF
    END IF
END IF
NEXT L
```

Volume Results

The original program gave the total haulage volume results in bank volume. However, the loose volume is required to know how much volume must be transported and the shrunk one is needed to know the compacted volume that will be placed in the fills. This is not a big problem while having only one shrink and swell factor, because it is easy to calculate. However, if the project has two or more types of material with different soil properties, the transformation from bank to loose volume is very time consuming.

In order to overcome this difficulty, the results are given in both loose and compacted volumes, and to do so, two counters were added to calculate the totals, TOTVOL for the loose volume and TOTSHRVOL for the shrunk volume, as shown in the last routine.

Because the total cost of the operation is important, the variable TSCYS is multiplied now by the unit cost in order to have the volume hauled not only multiplied by the distance involved, but also multiplied by the hauling unit cost. In the original program the volumes used were bank volumes. This is not a correct approach because the volume hauled is loose volume instead of bank volume. Therefore, with the enhancement, the cumulative total price is obtained, multiplying each loose volume by the unit price and the haulage distance involved.

Unusable Volumes

Sometimes the sections may contain materials or types of soils that can not be used to fill. Thus, a routine was created to handle this eventuality. First, the program ask if there are sections with unusable volumes. Then it asks the number of the sections and the percentage of the volume that can not be used, reducing the volume of the section by the specified percentage. For example, if 20% of the volume of a section can not be used, 20 must be entered. The volume of the correspondent section will be reduced then by 20% by multiplying the volume by $(1-20/100)$, i.e., 0.8.

The following routine was added to deal with this problem:

```
INPUT " ARE THERE ANY SECTION WITH UNUSABLE VOLUME Y/N "; AS  
IF AS = "Y" THEN
```

```

LOCATE 14,12
INPUT " IN HOW MANY SECTIONS ";NUM
DIM SN(NUM)
FOR I = 1 TO NUM
CLS: LOCATE 4,8
INPUT "WHICH SECTION (ENTER THE SECTION NUMBER)"; SN(I)
PRINT
PRINT " HOW MUCH VOLUME IN % *CAN NOT BE USED* INSECTION("; SN(I);)"
PRINT
PRINT " e.i. enter 40 for 40% of wasted volume=";
INPUT WASTE
VOL(SN(I))=VOL(SN(I))*(1-WASTE/100)
NEXT

```

Incomplete Fills and Cuts

In the original program, sometimes some fills remain not completed at the end of the calculations, and the program does not notice it. The original program checks if the total cut plus borrow pits volume is larger than the total fill volume, and if the total fill plus landfill volume is larger than the total cut volume. If they are not, more landfill or borrow pit volume is requested to avoid that some cuts or fills would remain incomplete. However, even after this checking is satisfied, some fill sections may remain empty and the problem is still not noticed. This happens because for the transportation algorithm solution, sometimes it is cheaper to send the cut volumes to the landfills instead than to some fills. Also, because there is enough space available in the landfills, and no more volume available in the borrow pits.

To detect this problem, a programming routine was added to identify the remaining volumes and print them. In addition, some recommendations to overcome this situation are printed too. These recommendations are either to reduce the landfill capacity, or to increase the borrow pits volume close to the empty fills. In the case of remaining cuts, the recommendation is the opposite. Thus, when this problem occurs, the engineer in charge must try different possible solutions using his/her own criteria

depending on the resources available in the field. These different alternatives may be entered in the program in order to choose the best one depending on the outputs.

The routine used is as follows:

```
FOR I= 1 TO NUMSEC
  IF ABS(SVOL(I)) > 1 THEN
    REMAINFLAG=1
    PRINT "SECTION (;I;) HAS A REMAINING VOLUME OF "; SVOL(I)
  END IF
NEXT
IF REMAINFLAG = 1 THEN
  PRINT
  PRINT "RUN THE PROGRAM AGAIN DOING THE FOLLOWING CHANGES"
  PRINT ""
  PRINT "-FOR A REMAINING FILL, INCREASE THE CLOSER"
  PRINT " BORROW PIT BY A VOLUME EQUAL TO THE REMAINING"
  PRINT " VOLUME DIVIDED BY THE BORROW PIT SHRINK FACTOR OR"
  PRINT " REDUCE THE LANDFILLS CAPACITY"
  PRINT " "
  PRINT "-FOR A REMAINING CUT, INCREASE THE CLOSER"
  PRINT " LANDFILL BY A VOLUME EQUAL TO THE REMAINING"
  PRINT " VOLUME OR REDUCE THE BORROW PITS CAPACITY"
  PRINT ""
  PRINT " STRIKE ANY KEY TO CONTINUE"
  PAUSE$ = INPUT$(1)
END IF
```

When the graphic display is skipped because of a large number of sections, the volumes (SVOL) of the remaining sections matched in the transportation algorithm are not reduced by the corresponding amount of cut or fill volume. This implies that they do not become zero. Therefore, it is impossible to control if any section remains unbalanced.

For this reason, a routine was added which reduces the volumes of the sections, when the graphic display is skipped, by the amount balanced by the transportation algorithm. This routine is as follows:

```
653 IF GRAPHFLAG = 1 THEN
      TSCYS = TSCYS * 100
      FOR I=1 TO NUMSEC
        FOR J= 1 TO NUMSEC
          SVOL(I) = SVOL(I) - XA(I,J)
          SVOL(J) = SVOL(J) + XA(I,J)
        NEXT J
      NEXT I
      FOR L = 1 TO M1
        I = A(L)
        J = B(L)
        IF I <= M8 AND J <= N8 THEN
          SVOL(CLOC(I)) = SVOL(CLOC(I)) - X(L)
          SVOL(FLOC(J)) = SVOL(FLOC(J)) + X(L)
        END IF
        IF I > M8 AND I <= M9 AND J <= N8 THEN
          SVOL(FLOC(J)) = SVOL(FLOC(J)) + X(L)
        END IF
        IF J > N8 AND J <= N9 AND I <= M8 THEN
          SVOL(CLOC(I)) = SVOL(CLOC(I)) - X(L)
        END IF
      NEXT L
    END IF
```

4.3 File Input and Output Improvements

The file output routine was created in order to save all the input data of the project. In this way, it is very fast and easy to rerun the program. Also, it is possible to modify all the input data imported from an existing file. This allows to run the program for the same project many times, trying different assumptions and conditions in order to find the better solution, e.g., by reducing the capacity of the landfills, less cut volume will go to the landfills and will be going to the fills instead, reducing in this

way the volume used from the borrow pits. This sensitivity analysis allows the user to choose from many alternatives, the most economic and practical solution.

Because of the file output routine creation, the file input routine was modified to acquire also the borrow pits and landfills data.

4.4 Graphical Output Improvements

The original graphic output resolution was very poor because the capabilities of the old Basic were limited. Therefore, it was increased using SCREEN 8 instead of SCREEN 1. SCREEN 8 gives a 200 x 640 screen compared to the 200 x 320 of SCREEN 1. This modification allows not only to have a better resolution, but also to handle a project up to 192 sections, instead of the original 96, for the graphical display . If the project has more than 192 sections, the graphical display will be skipped, and the user will receive only the printed output data.

The display of the graphs, which is presented step-by-step in a sequential manner, is controlled by the new version manually instead of automatically. This change was made because the automatic control is made by a counter loop. Therefore, the difference in speed among computers will cause a slow pause when using a slow computer, and a fast pause in the opposite case. By having this part of the program done manually, the user has more control and will be able to appreciate better some critical part of the distribution.

4.5 Screen Output Improvements

The input data display was enhanced showing the shrink and swell factor of each section when the sections volumes are printed in the screen. Only the cuts (positive volumes) have their swell and shrink factor printed, because the fills are empty spaces that do not have any soil property.

Also a control of the quantity of data displayed in the screen was included to stop the displaying process allowing to review all the data carefully.

The use of commas for the display of numbers was introduced in order to permit an easy, fast and accurate review of the data displayed in the screen.

4.6 Printed Output Improvements

As it is shown in Appendix A, the original output data printed looks very confusing. To overcome this problem, the new output data is sorted using loops and printed in columns. This provides an elegant and easier to check printout, as shown in Appendix B.

In the original program, the volumes hauled between stations are given in bank volume. In order to know the exact amount of earth that must be transported, these volumes must be converted to loose volume. This calculation would consume a great amount of time in a large project. Therefore, the output was modified, in order to print the shrink and loose volumes for each hauled volume and for the total operation.

In the new version of the program, it is possible to obtain a printout of the data input. This allow to create a hard copy which is useful to keep together with all the project documentation. This output gives all the data of the cut and fill volumes, borrow pits and landfills.

Finally, in the previous program solutions of 0 volume transported were printed sometimes as a result. This problem was overcome by using an IF THEN filter before any printing subroutine printout. This filter does not allow any printout of a solution smaller than 1, which could be printed rounded to 0.

4.7 Compilation

The program was compiled in order to increase the speed and the flexibility to use it directly invoked from the operating system without the need to call Basic or Quick Basic in advance. This makes it easier to learn and use, because the user does not need to know Quick Basic or Basic to run it,

and he/she does not need to use or buy a Quick Basic package. All this reduces some of the barriers or resistance that the potential users may have to start using the program.

5.0 CONCLUSIONS AND RECOMMENDATIONS

This program will make available to people involved in earth moving operations, the use of a construction management tool. Although the program does not provide an exact solution because of the natural differences between what is planned in advance and the operations in the job site, it is a good and close approximation to a realistic solution, and it offers more advantages than the Mass Diagram method.

This program is not only fast, accurate, easy to operate and does not require any special skills, but it is also inexpensive because it does not need any special hardware or software. Additionally, its use is not time consuming. This characteristic gives it a powerful advantage, i.e., the capability to enable a user to try different alternatives for the solution in order to find the most economic and efficient one. This does not imply necessarily the least expensive solution (but which could be technically not feasible), but rather the most technically feasible at the lowest possible cost. The use of this program will allow users to make more accurate estimates of the earth moving costs, and to plan operations in a more efficient manner.

All these characteristics should make it easier to accept the program. With more people using this approach, new ideas and suggestions for improvement can be received as a feedback. This will lead to even more enhancements and additional features. The continuous improvement cycle will make it more powerful, flexible and suitable for many different project characteristic over time.

To conclude, the following recommendations are given in an attempt to assist researchers in the future to develop additional enhancements to this program.

A more structured programing.— With the use of more subroutines, the structure of the program can be improved. In doing so, most of the label numbers should be eliminated. However,

this kind of improvement is very difficult to make because some parts of the program are very entangled, and any changes should be made very carefully in order to maintain the logic of the program. As a result of this suggested enhancement, the program will not only be faster, but also easier to understand while reviewing its code.

Menu based format.— With the use of subroutines, it will be easier to introduce a menu based format in a future version. The menu based format will add flexibility to the program, will make it easier to use.

Matrix format output.— The printout of the results using a source and destination matrix will give a better idea of the earth distribution plan. The matrix could look as the Source-Destination Distance Matrix shown in figure 3.2 .

Add extra unit cost for borrow pits and landfills.— Sometimes the use of borrow pits and landfills can be charged with an extra cost. This extra cost can be added using an equivalent distance that reflects the extra cost. For example, the haulage unit cost for cut material within a project is \$1.00/cy/100ft.. Additionally, material from a borrow pit may have an extra cost of \$0.50 per cubic yard for any reason. This extra cost can be taken into account by increasing the borrow pit haulage distance by:

$$\frac{\$ 0.50 / \text{cy}}{\$1.00 / \text{cy} / 100\text{ft}} = 50 \text{ ft.}$$

This operation can easily be introduced in the program after the borrow pits and landfills data introduction section.

Dimensioned variables for distances.— Sometimes during construction, it is more difficult or costly to go through some sections than through others, e.g., using bypasses because of culverts installation or bridge construction, or having sections with steep slopes.

This problem can be overcome by increasing the length of the section taking into account

- (1) the distance involved in the bypass,

- (2) the higher slope, or
- (3) any extra cost (as with the borrow pits extra cost).

However, in the present program the lengths of the sections are assumed equal. Therefore, in order to modify the lengths, dimensioned variables for each section length should be used. To save time in the data introduction, the length can be assumed first as equal, automatically assigning to each section length variable a default value using a FOR NEXT loop. Then, in the next step, these distances can be modified manually for any section which needs to be modified.

A consequence of this change is that normally the distance between two stations is calculated by subtracting the stations' position numbers, and multiplying the result by the unique section length. With the dimensioned variables, a counter should be created in order to sum the lengths of all the sections in between the two sections. This can be made easily using an FOR NEXT loop.

Use of shadow prices.— The transportation algorithm provides shadow prices which can be used to make a sensitivity analysis. The shadow prices indicate the sensitivity of the total cost when changes are made in the different variables of the cost function Z . These shadow prices can be used by the user to know where to make changes in order to improve the solution.

6.0 REFERENCES

Hillier and Lieberman. (1986). Introduction to Operations Research. Holden-Day, Oakland, CA.

Merrit, Frederick. (1983). Standard Handbook for Civil Engineers. Frederick S. Merrit, West Palm Beach, FL.

Pantumsinchai and Pricha. (1983). Basic programs for production and operations management. Prentice Hall, Englewood Cliffs,NJ.

Toomy Robert. (1984). The Development of an Earth Distribution Analysis Program for Microcomputers, thesis presented to the Georgia Institute of Technology, at Atlanta, Georgia, USA, in partial fulfillment of the requirements for the degree of Master in Science.

Wagner, Harvey. (1975). Principles of Operations Research. Prentice Hall, Englewood Cliffs,NJ.

APPENDIX A:
SAMPLE OF THE ORIGINAL PRINTOUT

PROJECT TEST3B

PROJECT NUMBER

EARTH DISTRIBUTION PLAN

AMOUNT FROM CUT SECTION	5	TO FILL SECTION	4	IS	3	DISTANCE IS	1
AMOUNT FROM CUT SECTION	6	TO FILL SECTION	3	IS	4	DISTANCE IS	3
AMOUNT FROM CUT SECTION	6	TO FILL SECTION	4	IS	1	DISTANCE IS	2
AMOUNT FROM CUT SECTION	7	TO FILL SECTION	2	IS	1	DISTANCE IS	5
AMOUNT FROM CUT SECTION	7	TO FILL SECTION	3	IS	2	DISTANCE IS	4
AMOUNT FROM CUT SECTION	11	TO FILL SECTION	9	IS	3	DISTANCE IS	2
AMOUNT FROM CUT SECTION	11	TO FILL SECTION	10	IS	2	DISTANCE IS	1
AMOUNT FROM CUT SECTION	12	TO FILL SECTION	8	IS	4	DISTANCE IS	4
AMOUNT FROM CUT SECTION	12	TO FILL SECTION	9	IS	1	DISTANCE IS	3
AMOUNT FROM BORROW PIT	2	TO FILL SECTION	1	IS	2	DISTANCE IS	12
AMOUNT FROM BORROW PIT	1	TO FILL SECTION	2	IS	3	DISTANCE IS	3
AMOUNT FROM CUT SECTION	13	TO LANDFILL AREA	2	IS	3	DISTANCE IS	7
AMOUNT FROM CUT SECTION	14	TO LANDFILL AREA	2	IS	3	DISTANCE IS	8
AMOUNT FROM CUT SECTION	13	TO LANDFILL AREA	1	IS	2	DISTANCE IS	12
AMOUNT FROM CUT SECTION	12	TO LANDFILL AREA	1	IS	2	DISTANCE IS	11

TOTAL HAULAGE COST IN THIS PROJECT IS- \$ 181

TOTAL VOLUME IN CY IS- 36

APPENDIX B:

SAMPLE #1 OF THE NEW PRINTOUT

PROJECT TEST5 INPUT DATA

SECTION NUMBER	VOLUME (+) BANK (-) SHRINK	SHRINK FACTOR	SWELL FACTOR
1	-2		
2	-4		
3	-6		
4	-4		
5	2	0.85	1.25
6	4	0.80	1.29
7	2	0.90	1.10
8	-4		
9	-4		
10	-2		
11	4	0.95	1.05
12	7	1.00	1.00
13	3	0.75	1.30
14	2	0.97	1.08

BORROW PIT DATA FOR PROJECT: TEST5

BORROW PIT NUMBER	CAPACITY (CY)	ENTERING SECTION	DISTANCE X100 (FT)	SWELL FACTOR	SHRINK FACTOR
1	2	2	3	1.10	0.80
2	4	9	4	1.15	0.88

LANDFILL DATA FOR PROJECT:TEST5

LANDFILL NUMBER	CAPACITY (CY)	DEPARTING SECTION	DISTANCE X100 (FT)
1	5	6	5
2	6	12	6

PROJECT: TEST5

PROJECT NUMBER :

EARTH DISTRIBUTION PLAN

FROM CUT SECTION	TO FILL SECTION	DISTANCE x100 FT	SHRINK VOLUME	LOOSE VOLUME
5	4	1	2	3
6	3	3	2	3
6	4	2	2	3
7	3	4	2	2
11	9	2	2	2
11	10	1	2	2
12	8	4	4	4
12	9	3	2	2

FROM BORROW PIT	TO FILL SECTION	DISTANCE x100 FT	SHRINK VOLUME	LOOSE VOLUME
1	2	3	2	3
2	3	10	2	3
2	2	11	2	2

FROM CUT SECTION	TO LANDFILL AREA	DISTANCE x100 FT	SHRINK VOLUME	LOOSE VOLUME
12	2	6	1	1
13	2	7	3	5
14	2	8	2	2

TOTAL HAULAGE COST IN THIS PROJECT IS= \$ 176

TOTAL LOOSE VOLUME IN CY IS= 39

~~TOTAL SHRINK VOLUME IN CY IS= 30~~

SECTION (1) HAS A REMAINING VOLUME OF -2

RECOMENDATIONS:

RUN THE PROGRAM AGAIN DOING THE FOLLOWING CHANGES:

- FOR A REMAINING FILL INCREASE THE CLOSER BORROW PIT BY A VOLUME EQUAL TO THE REMAINING VOLUME DIVIDED BY THE BORROW PIT SHRINK FACTOR
 - FOR A REMAINING CUT INCREASE THE CLOSER LANDFILL BY A VOLUME EQUAL TO THE REMANING VOLUME.
-

APPENDIX C:

SAMPLE #2 OF THE NEW PRINTOUT

PROJECT LAWRENCEVILLE BYPASS INPUT DATA

SECTION NUMBER	VOLUME (+) BANK (-) SHRINK	SHRINK FACTOR	SWELL FACTOR
1	2,391	0.85	1.15
2	26,576	0.85	1.15
3	111,898	0.85	1.15
4	116,543	0.85	1.15
5	-97,189		
6	-64,951		
7	-32,856		
8	18,198	0.85	1.15
9	74,750	0.85	1.15
10	-9,936		
11	-114,890		
12	-34,000		
13	4,791	0.85	1.15
14	53,952	0.85	1.15
15	16,795	0.85	1.15
16	-5,811		
17	-15,417		
18	-60,327		
19	-27,494		
20	-22,810		
21	-15,589		
22	-56,092		
23	-7,400		
24	-23,555		
25	-66,397		
26	-5,025		
27	24,672	0.85	1.15
28	49,910	0.85	1.15
29	-73,668		
30	-3,741		
31	13,931	0.85	1.15
32	-10,727		
33	-13,878		
34	25,402	0.85	1.15
35	68,288	0.85	1.15
36	28,660	0.85	1.15
37	61,190	0.85	1.15
38	67,794	0.85	1.15
39	6,601	0.85	1.15
40	-101,693		
41	52,884	0.85	1.15
42	79,805	0.85	1.15
43	-69,369		
44	-765		
45	21,193	0.85	1.15
46	18,707	0.85	1.15
47	-321		
48	-17,232		
49	400	0.85	1.15
50	22,733	0.85	1.15

PROJECT LAWRENCEVILLE BYPASS INPUT DATA

SECTION NUMBER	VOLUME (+) BANK (-) SHRINK	SHRINK FACTOR	SWELL FACTOR
51	63,855	0.85	1.15
52	60,117	0.85	1.15
53	-67,302		
54	5,639	0.85	1.15
55	-13,259		
56	1,125	0.85	1.15
57	-70,116		
58	-69,383		
59	-69,383		
60	7,965	0.85	1.15

PROJECT: LAWRENCEVILLE BYPASS

PROJECT NUMBER :

EARTH DISTRIBUTION PLAN

FROM CUT SECTION	TO FILL SECTION	DISTANCE x100 FT	SHRINK VOLUME	LOOSE VOLUME
3	6	3	45,597	61,690
3	7	4	32,856	44,452
4	5	1	97,189	131,491
4	6	2	19,354	26,185
8	11	3	18,198	24,621
9	10	1	9,936	13,443
9	11	2	64,814	87,690
13	12	1	4,791	6,482
14	11	3	24,743	33,476
14	12	2	29,209	39,518
15	11	4	7,135	9,653
15	16	1	5,811	7,862
15	17	2	3,849	5,207
27	25	2	19,647	26,581
27	26	1	5,025	6,799
28	24	4	3,160	4,275
28	25	3	46,750	63,250
31	29	2	10,190	13,786
31	30	1	3,741	5,061
34	32	2	10,727	14,513
34	33	1	13,878	18,776
37	40	3	27,298	36,933
38	40	2	67,794	91,721
39	40	1	6,601	8,931
42	43	1	69,369	93,852
42	44	2	765	1,035
46	47	1	321	434
46	48	2	17,232	23,314
51	53	2	7,185	9,721
52	53	1	60,117	81,335
54	55	1	5,639	7,629
56	55	1	1,125	1,522
60	59	1	7,965	10,776
1	17	16	2,391	3,235
2	17	15	9,177	12,416
2	18	16	17,399	23,540
3	18	15	33,445	45,249
34	29	5	797	1,078
35	29	6	62,681	84,804
35	24	11	5,607	7,586
36	24	12	14,788	20,007
36	23	13	7,400	10,012
36	22	14	6,472	8,756
37	22	15	33,892	45,854
41	55	14	6,495	8,787
41	57	16	46,389	62,762

FROM CUT SECTION	TO FILL SECTION	DISTANCE x100 FT	SHRINK VOLUME	LOOSE VOLUME
42	57	15	9,671	13,084
45	57	12	14,056	19,017
45	58	13	7,137	9,656
46	58	12	1,154	1,561
49	58	9	400	541
50	58	8	22,733	30,756
51	58	7	37,959	51,356
51	59	8	18,711	25,315

FROM BORROW PIT	TO FILL SECTION	DISTANCE x100 FT	SHRINK VOLUME	LOOSE VOLUME
-----------------	-----------------	------------------	---------------	--------------

FROM CUT SECTION	TO LANDFILL AREA	DISTANCE x100 FT	SHRINK VOLUME	LOOSE VOLUME
------------------	------------------	------------------	---------------	--------------

TOTAL HAULAGE COST IN THIS PROJECT IS- \$ 7,547

TOTAL LOOSE VOLUME IN CY IS- 1,497,388

TOTAL SHRINK VOLUME IN CY IS- 1,106,765

DUE TO INSUFFICIENT CUT AND BORROW,
THE FOLLOWING FILL SECTIONS REMAIN:
(SHRINK VOLUME)

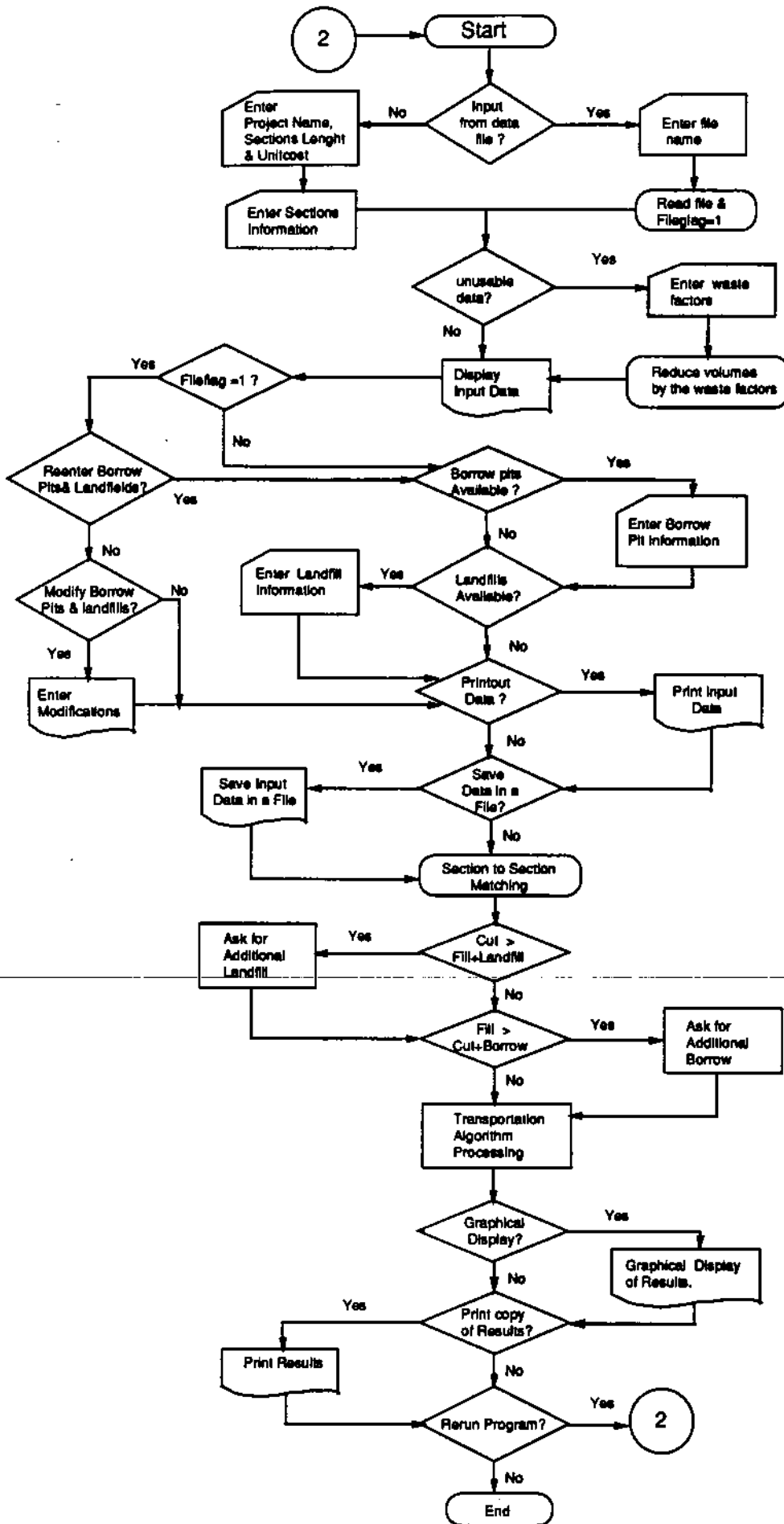
FILL SECTION	18	- VOLUME -	9,483
FILL SECTION	19	- VOLUME -	27,494
FILL SECTION	20	- VOLUME -	22,810
FILL SECTION	21	- VOLUME -	15,589
FILL SECTION	22	- VOLUME -	15,728
FILL SECTION	59	- VOLUME -	42,707

RECOMENDATIONS:

RUN THE PROGRAM AGAIN DOING THE FOLLOWING CHANGES:

- FOR A REMAINING FILL INCREASE THE CLOSER BORROW PIT BY A VOLUME EQUAL TO THE REMAINING VOLUME DIVIDED BY THE BORROW PIT SHRINK FACTOR
 - FOR A REMAINING CUT INCREASE THE CLOSER LANDFILL BY A VOLUME EQUAL TO THE REMANING VOLUME.
-

APPENDIX D:
PROGRAM FLOW CHART



APPENDIX E:
PROGRAM CODE

PROGRAM CODE

```
SCREEN 0: CLEAR : CLS : KEY OFF
PRINT : PRINT
PRINT " *****"
PRINT " * Quick BASIC *"
PRINT " * EARTH DISTRIBUTION PROGRAM *"
PRINT " * *"
PRINT " *****"
PRINT : PRINT
PRINT " QUICK BASIC PROGRAM FOR AUTOMATIC EARTH DISTRIBUTION ANALYSIS "
PRINT " WRITTEN BY JUAN PABLO VASQUEZ AT PURDUE UNIVERSITY IN SEPTEMBER"
PRINT " 1991, BASED IN THE PROGRAM MADE BY MR. ROBERT S. TOOMY FROM"
PRINT " GEORGIA INSTITUTE OF TECHNOLOGY IN MAY 1984. "
PRINT : PRINT : PRINT
INPUT "**** PRESS RETURN TO CONTINUE ****", AS
CLS : PRINT : PRINT
PRINT " OPTIMIZATION OF THE DISTRIBUTION OF CUT AND FILL"
PRINT " FOR LINEAR PROYECTS"
PRINT
PRINT " DATA REQUIRED:"
PRINT " - NUMBER OF SECTIONS"
PRINT " - CUT IN CY (-)"
PRINT " - FILL IN CY (+)"
PRINT " - SWELL AND SHRINKAGE FACTORS "
PRINT " - BORROW PIT (IF ANY) NUMBER, LOCATION AND CAPACITY"
PRINT " - LANDFILL (IF ANY) NUMBER, LOCATION AND CAPACITY"
PRINT
PRINT " RESULTS:"
PRINT " - VOLUME OF EACH CUT SECTION GOING TO A FILL SECTION"
PRINT " - DISTANCE INVOLVED"
PRINT " - TOTAL HAULAGE IN STATION-YARDS FOR THE PROJECT"
PRINT : PRINT : PRINT
INPUT "**** PRESS RETURN TO CONTINUE ****", AS
-----
SCREEN 0: CLEAR : CLS : LOCATE 12, 12
10 INPUT "WILL INPUT BE READ FROM A DATA FILE (Y/N)"; ASS
IF ASS = "Y" THEN
5 CLS : LOCATE 12, 12
  INPUT "FROM WHICH DRIVE ex. B: "; DRIVES
  LOCATE 14, 12
  INPUT "NAME OF THE INPUT FILE"; NIFS
  CLOSE
  ON ERROR GOTO HANDLER
  OPEN DRIVES + NIFS FOR INPUT AS #1
  INPUT #1, PRONAMES, PNUMS, NUMSEC, SECLN, UNITPRICE, NBP
  DIM VOL(NUMSEC), XA(NUMSEC, NUMSEC), SVOL(NUMSEC), SWELL(NUMSEC + NBP + 6),
    SHRINK(NUMSEC + NBP + 6)
  FOR I = 1 TO NUMSEC
  INPUT #1, I, VOL(I), SWELL(I), SHRINK(I)
  NEXT
  INPUT #1, NBP
  DIM CBP(NBP), ES(NBP), BPD(NBP)
```



```

FOR I = 1 TO NBP
INPUT #1, I, CBP(I), ES(I), BPD(I), SWELL(NUMSEC + I), SHRINK(NUMSEC + I)
NEXT
INPUT #1, NLF
DIM CLF(NLF), DS(NLF), LFD(NLF)
FOR I = 1 TO NLF
INPUT #1, I, CLF(I), DS(I), LFD(I)
NEXT
FILEFLAG = 1
CLOSE
ELSE
IF ASS <> "N" THEN
PRINT " PLEASE ANSWER WITH Y OR N ": GOTO 10
ELSE
CLS : LOCATE 11, 12
INPUT "PROJECT NAME "; PRONAMES
INPUT "TOTAL NUMBER OF SECTIONS"; NUMSEC
INPUT "SECTION LENGHT IN FEET"; SECLEN
INPUT " COST $/CY/100 FT "; UNITPRICE
INPUT "NUMBER OF BORROW PITS"; NBP
DIM VOL(NUMSEC), XA(NUMSEC, NUMSEC), SVOL(NUMSEC), SWELL(NUMSEC + NBP),
SHRINK(NUMSEC + NBP)
FOR G = 1 TO NUMSEC
CLS : LOCATE 13, 8
PRINT "VOLUME IN CY OF CUT (-) OR (+) FILL FOR SECTION ("; G; ") = ";
INPUT ; VOL(G)
NEXT
CLS : LOCATE 12, 12
INPUT " HOW MANY ZONES WITH A DIFFERENT SOIL PROPERTY DO THE PROJECT
HAVE ="; NPR
DIM GSHRINK(NPR), GSWELL(NPR), START(NPR), XEND(NPR)
CLS
FOR I = 1 TO NPR
CLS
LOCATE 12, 12
PRINT " WHICH SECTIONS COMPRISE SOIL TYPE # ("; I; ")"
LOCATE 14, 12
INPUT " NUMBER OF THE FIRST SECTION (START)="; START(I)
LOCATE 16, 12
INPUT "NUMBER OF THE LAST SECTION (END)="; XEND(I)
LOCATE 18, 12
INPUT " SHRINK FACTOR (e.i 0.80 for 20% shrink from bank vol.) = "; GSHRINK(I)
LOCATE 20, 12
INPUT " SWELL FACTOR (e.i. 1.20 for 20% swell from bank vol. ) = "; GSWELL(I)
CLS
NEXT I
FOR I = 1 TO NPR
FOR K = START(I) TO XEND(I)
SHRINK(K) = GSHRINK(I)
SWELL(K) = GSWELL(I)
NEXT K
NEXT I
END IF
END IF

```

```

14 CLS
LOCATE 12, 12
INPUT "IS THERE ANY SECTION WITH UNUSABLE VOLUME Y/N "; AS
IF AS = "Y" THEN
    LOCATE 14, 12
    INPUT " IN HOW MANY SECTIONS"; NUM
    DIM SN(NUM)
    FOR I = 1 TO NUM
    CLS : LOCATE 4, 8
    INPUT " WHICH SECTION (ENTER THE SECTION NUMBER)"; SN(I)
    PRINT
    PRINT " HOW MUCH VOLUME IN % *CAN NOT BE USED* IN SECTION("; SN(I); ")"
    PRINT
    PRINT "e.i Enter 40 for 40% of wasted volume="; : INPUT WASTE
    VOL(SN(I)) = VOL(SN(I)) * (1 - WASTE / 100)
    NEXT
    CLS
    PRINT : PRINT
    PRINT "NEW VOLUMES ARE : "
    PRINT "-----"
    PRINT
    FOR I = 1 TO NUM
    PRINT "VOL("; SN(I); ") = "; VOL(SN(I))
    NEXT
    PRINT
    PRINT "PRESS ANY KEY TO CONTINUE": AAS = INPUT$(1)
    ELSEIF AS <> "N" THEN
    LOCATE 14, 12
    PRINT "PLEASE ANSWER Y OR N"
    GOTO 14
END IF
IF FILEFLAG = 1 THEN
15 CLS
    LOCATE 12, 12
    INPUT "DO YOU WANT TO RENAME THE PROJECT Y/N "; AS
    IF AS = "Y" THEN
        LOCATE 14, 12
        INPUT " ENTER NEW NAME "; PRONAMES$
    ELSEIF AS <> "N" THEN
        LOCATE 14, 12
        PRINT "PLEASE ANSWER Y OR N"
        GOTO 15
    END IF
16 CLS
    LOCATE 12, 12
    INPUT "DO YOU WANT TO CHANGE THE UNIT COST Y/N "; AS
    IF AS = "Y" THEN
        LOCATE 14, 12
        INPUT " COST $/CY/100 FT "; UNITPRICE
    ELSEIF AS <> "N" THEN
        LOCATE 14, 12
        PRINT "PLEASE ANSWER Y OR N"
        GOTO 15
    END IF
END IF

```

```

20 CLS
25 LOCATE 12, 12
INPUT "DO YOU WANT TO REVIEW YOUR INPUT DATA (Y/N)"; A$
IF A$ = "Y" THEN
  CLS
  PRINT "      PROJECT "; PRONAMES; " INPUT DATA"
  PRINT "-----"
  PRINT " SECTION      VOLUME  SHRINK  SWELL"
  PRINT "      (+) BANK  FACTOR  FACTOR"
  PRINT "      (-) SHRINK          "
  PRINT "-----"
  COUNTER2 = 0
  FOR G = 1 TO NUMSEC
  IF VOL(G) > 0 THEN
    PRINT USING " ##      #,###,###      #.##      #.## "; G; VOL(G); SHRINK(G); SWELL(G)
  ELSE
    PRINT USING " ##      #,###,###      "; G; VOL(G)
  END IF
  COUNTER2 = COUNTER2 + 1
  IF COUNTER2 = 16 THEN
    PRINT " PRESS ANY KEY TO CONTINUE"
    PAUSE$ = INPUT$(1)
    COUNTER2 = 0
    CLS
    PRINT "      PROJECT "; PRONAMES; " INPUT DATA"
    PRINT "-----"
    PRINT " SECTION      VOLUME  SHRINK  SWELL"
    PRINT "      (+) BANK  FACTOR  FACTOR"
    PRINT "      (-) SHRINK          "
    PRINT "-----"
  END IF
  NEXT G
  PRINT : PRINT
ELSEIF A$ <> "N" THEN
  PRINT " PLEASE ANSWER WITH Y OR N "; GOTO 25
END IF
IF A$ = "N" THEN LOCATE 14, 12
30 INPUT "DO YOU WISH ANY CHANGE (Y/N)"; B$
IF B$ = "Y" THEN
  CLS : LOCATE 12, 12
  INPUT "ENTER THE SECTION NUMBER TO BE CHANGED"; G: PRINT
  LOCATE 14, 12
  INPUT "ENTER THE VOLUME OF THE SECTION IN CUT(-) OR FILL(+)"; VOL(G)
  GOTO 20
ELSEIF B$ <> "N" THEN
  PRINT " PLEASE ANSWER WITH Y OR N "; GOTO 30
END IF
TFILL = 0: TCUT = 0: TBPIT = 0: TLF = 0
IF FILEFLAG = 1 THEN
  CLS
35 LOCATE 12, 12
  INPUT "DO YOU WANT TO REENTER YOUR BORROW PIT AND LANDFILL DATA Y/N"; A$
  IF A$ = "N" THEN
    GOTO 50
  ELSEIF A$ <> "Y" THEN
    LOCATE 14, 12
    PRINT "PLEASE ANSWER Y OR N"
    GOTO 35

```

```

END IF
LOCATE 6, 12: PRINT " YOU CAN MODIFIED ALL THE DATA BUT YOU CAN ONLY ADD"
LOCATE 8, 12: PRINT " SIX (6) NEW BORROW PITS. ( THE NEW LANDFILL HAVE"
LOCATE 10, 12: PRINT " NOT ANY LIMITATION"
END IF
CLS
40 LOCATE 12, 12
INPUT "ARE BORROW PITS AVAILABLE (Y/N)"; AS
IF AS = "Y" THEN
  CLS : LOCATE 12, 12
  ERASE CBP, ES, BPD
  INPUT "NUMBER OF BORROW PITS"; NBP
  DIM CBP(NBP), ES(NBP), BPD(NBP)
  FOR K = 1 TO NBP
    CLS
    LOCATE 10, 12
    PRINT "CAPACITY IN CY FOR BORROW PIT #("; K; ")="; : INPUT CBP(K)
    LOCATE 12, 12
    PRINT "WHAT IS THE NEAREST SECTION THAT TRUCKS FROM THE"
    LOCATE 13, 17
    INPUT "BORROW PIT CAN ENTER THE PROJECT AREA"; ES(K)
    LOCATE 15, 12
    PRINT "WHAT IS THE DISTANCE FROM THE BORROW PIT TO THIS"
    LOCATE 16, 17
    INPUT "SECTION IN FEET"; BPD(K)
    BPD(K) = BPD(K) / SECLEN
    LOCATE 18, 12
    INPUT "SWELL FACTOR AS A FRACTION OF BANK VOLUME (I.E 1.25)"; SWELL(NUMSEC + K)
    LOCATE 20, 12
    INPUT "SHRINK FACTOR AS A FRACTION OF BANK VOLUME (I.E 0.95)"; SHRINK(NUMSEC + K)
    NEXT
    CLS
  ELSEIF AS <> "N" THEN
    PRINT " PLEASE ANSWER WITH Y OR N ": GOTO 40
  END IF
50 CLS
55 LOCATE 12, 12
INPUT "DO YOU WANT TO SEE THE BORROW PIT DATA (Y/N)"; AS
IF AS = "Y" THEN
  CLS
  PRINT : PRINT "----- BORROW PIT DATA FOR PROJECT: "; PRONAMES
  PRINT "-----"
  PRINT : PRINT
  PRINT " BORROW PIT CAPACITY ENTERING DISTANCE SWELL SHRINK "
  PRINT " NUMBER (CY) SECTION x100 (FT) FACTOR FACTOR "
  PRINT "-----"
  FOR K = 1 TO NBP
    PRINT USING " ## ###,### ### #,### ### #.## "; K; CBP(K); ES(K);
    BPD(K); SWELL(NUMSEC + K); SHRINK(NUMSEC + K)
  NEXT
  PRINT : PRINT : PRINT : PRINT
ELSEIF AS <> "N" THEN
  PRINT " PLEASE ANSWER WITH Y OR N ": GOTO 55
END IF

```

```

IF AS = "N" THEN LOCATE 14, 12
60 INPUT "DO YOU WISH TO MAKE ANY CHANGES (Y/N)"; ASS
IF ASS = "Y" THEN
    120 CLS
    LOCATE 8, 12
    INPUT "WHAT IS THE NUMBER OF THE BORROW PIT"; K
    LOCATE 10, 12
    PRINT "CAPACITY IN CY FOR BORROW PIT #("; K; ")="; : INPUT CBP(K)
    LOCATE 12, 12
    PRINT "WHAT IS THE NEAREST SECTION THAT TRUCKS FROM THE"
    LOCATE 13, 17
    INPUT "BORROW PIT CAN ENTER THE PROJECT AREA"; ES(K)
    LOCATE 15, 12
    PRINT "WHAT IS THE DISTANCE FROM THE BORROW PIT TO THIS"
    LOCATE 16, 17
    INPUT "SECTION IN FEET"; BPD(K)
    BPD(K) = BPD(K) / SECLN
    LOCATE 18, 12
    INPUT "SWELL FACTOR AS A FRACTION OF BANK VOLUME (I.E 1.25)"; SWELL(NUMSEC + K)
    LOCATE 20, 12
    INPUT "SHRINK FACTOR AS A FRACTION OF BANK VOLUME (I.E 0.95)"; SHRINK(NUMSEC + K)
    GOTO 50
ELSEIF ASS <> "N" THEN
    PRINT " PLEASE ANSWER WITH Y OR N ": GOTO 60
END IF
IF BPFLAG = 1 THEN 70
IF FILEFLAG = 1 THEN
    CLS :
86    LOCATE 12, 12
    INPUT "DO YOU WANT TO REENTER YOUR LANDFILL DATA Y/N"; AS
    IF AS = "N" THEN
        GOTO 80
    ELSEIF AS <> "Y" THEN
        LOCATE 14, 12
        PRINT "PLEASE ANSWER Y OR N"
        GOTO 86
    END IF
END IF
END IF
90 CLS
95 LOCATE 12, 12
INPUT "ARE LANDFILLS AVAILABLE (Y/N)"; ASS
IF ASS = "Y" THEN
    CLS : LOCATE 12, 12
    INPUT "NUMBER OF LANDFILLS="; NLF
    ERASE CLF, DS, LFD
    DIM CLF(NLF), DS(NLF), LFD(NLF)
    FOR H = 1 TO NLF
100    CLS : LOCATE 10, 12
        PRINT "WHAT IS THE CAPACITY OF THE LANDFILL ("; H; ")=";
        INPUT CLF(H)
        LOCATE 12, 12
        PRINT "FROM WHAT SECTION CAN TRUCKS LEAVE THE PROJECT"
        LOCATE 13, 12
        INPUT "AREA FOR THE LANDFILL"; DS(H)
        LOCATE 15, 12
        PRINT "WHAT IS THE DISTANCE FROM THIS SECTION TO THE"
        LOCATE 16, 12
        INPUT " LANDFILL IN FEET"; LFD(H): LFD(H) = LFD(H) / SECLN

```

```

        IF HFLAG = 1 THEN 80
        NEXT H
ELSEIF ASS <> "N" THEN
    PRINT " PLEASE ANSWER WITH Y OR N ": GOTO 95
END IF
80 CLS
85 LOCATE 12, 12
INPUT "DO YOU WANT TO SEE THE LANDFILL DATA (Y/N)"; ASS
IF ASS = "Y" THEN
    CLS : PRINT : PRINT
    PRINT " LANDFILL DATA FOR PROJECT: "; PRONAMES
    PRINT " _____"
    PRINT : PRINT
    PRINT " LANDFILL      CAPACITY      DEPARTING      DISTANCE"
    PRINT " NUMBER        (CY)        SECTION        x100 (FT)"
    PRINT " _____"
    FOR H = 1 TO NLF
    PRINT USING "  ##      #,###,###      ###      #,###"; H; CLF(H); DS(H); LFD(H)
    NEXT
    PRINT : PRINT
ELSEIF ASS <> "N" THEN
    PRINT " PLEASE ANSWER WITH Y OR N ": GOTO 85
END IF
IF ASS = "N" THEN LOCATE 14, 12
110 INPUT " DO YOU WISH TO MAKE ANY CHANGE (Y/N)"; ASS
IF ASS = "Y" THEN
130 CLS
    LOCATE 8, 12: INPUT "WHAT IS THE NUMBER OF THE LANDFILL"; H
    HFLAG = 1
    GOTO 100
ELSEIF ASS <> "N" THEN
    PRINT " PLEASE ANSWER WITH Y OR N "
    GOTO 110
END IF
TLF = 0
FOR H = 1 TO NLF
    TLF = TLF + CLF(H)
NEXT
IF LFFLAG = 1 THEN 70
131 CLS : LOCATE 12, 6
INPUT " DO YOU WANT TO SAVE THE DATA IN A FILE Y/N "; AS
IF AS = "Y" THEN
132 INPUT " FROM WHICH DRIVE ex. B: "; DRIVES
    INPUT "WHAT IS THE NAME OF THE FILE THAT WILL STORE THIS DATA "; NIFS
    CLOSE
    ON ERROR GOTO HANDLER
    OPEN DRIVES + NIFS FOR OUTPUT AS #1
    WRITE #1, PRONAMES, PNUMS, NUMSEC, SECLN, UNITPRICE, NBP
    FOR I = 1 TO NUMSEC
    WRITE #1, I, VOL(I), SWELL(I), SHRINK(I)
    NEXT
    WRITE #1, NBP
    FOR I = 1 TO NBP
    WRITE #1, I, CBP(I), ES(I), BPD(I), SWELL(I + NUMSEC), SHRINK(I + NUMSEC)
    NEXT
    WRITE #1, NLF
    FOR I = 1 TO NLF
    WRITE #1, I, CLF(I), DS(I), LFD(I)

```

```

NEXT
CLOSE
IF ERR = 53 THEN
    CLS : LOCATE 12, 12
    INPUT "DRIVE NOT READY , CHECK IT AND PRESS ENTER"; P$
    GOTO 132
    RESUME
END IF
ELSEIF A$ <> "N" THEN
    PRINT " PLEASE ANSWER WITH Y OR N "
    INPUT " **** PRESS ENTER TO CONTINUE****"; A$
    GOTO 131
END IF
IF LFFLAG = 1 THEN 70
BIGVOL = 0
FOR G = 1 TO NBP
    CBP(G) = CBP(G) * SHRINK(G + NUMSEC)
    TBPIT = TBPIT + CBP(G)
NEXT
FOR G = 1 TO NUMSEC
    IF VOL(G) >= 0 THEN
        VOL(G) = INT(VOL(G) * SHRINK(G))
        TCUT = TCUT + VOL(G)
    ELSE
        TFILL = TFILL - VOL(G)
    END IF
    SVOL(G) = VOL(G)
    IF SVOL(G) > BIGVOL THEN BIGVOL = SVOL(G)
NEXT
70 CLS
PRINT : PRINT
PRINT "      INPUT DATA FOR PROJECT "; PRONAMES
PRINT "      *****"
PRINT : PRINT
PRINT " TOTAL NUMBER OF SECTIONS ="; NUMSEC
PRINT USING " TOTAL LOOSE VOLUME OF CUT = ###,###,### CY "; TCUT
PRINT USING " TOTAL AMOUNT OF FILL      = ###,###,### CY"; TFILL
PRINT USING " PROJECT LENGHT          = ###,###,### YD"; NUMSEC * SECLN
PRINT " NUMBER OF BORROW PITS      ="; NBP
PRINT " CAPACITY OF BORROW PITS    ="; TBPIT
PRINT " NUMBER OF LANDFILLS        ="; NLF
PRINT " CAPACITY OF LANDFILLS      ="; TLF
PRINT : PRINT
INPUT "*****PRESS RETURN TO CONTINUE*****", ASS
IF BPFLAG = 1 THEN
    IF LFFLAG = 1 THEN
75        CLS
            PRINT " DO YOU WANT TO SAVE THE MODIFIED DATA IN A FILE Y/N ": A$ = INKEY$
            IF A$ = "Y" THEN
                GOTO 132
            ELSEIF A$ <> "N" THEN
                PRINT " PLEASE ANSWER WITH Y OR N "
                PRINT " **** PRESS ANY KEY TO CONTINUE****": B$ = INKEY$
                GOTO 75
            END IF
        END IF
    END IF
END IF

```

```

IF TFILL > TCUT + TBPIT THEN
  CLS : LOCATE 10, 2
  PRINT "REQUIRED FILL EXCEEDS AVAILABLE CUT"
  PRINT USING "AND BORROW PIT CAPACITY BY ##,###,###,### "; TFILL - (TCUT + TBPIT); :
  PRINT "CY"
  PRINT
140 INPUT "DO YOU WANT TO ADD MORE CAPACITY (Y/N)"; ASS
  IF ASS = "Y" THEN
    BPFLAG = 1: GOTO 120
  ELSEIF ASS <> "N" THEN
    PRINT " PLEASE ANSWER Y OR N ": GOTO 140
  END IF
  NOXBP = 1
  END IF
IF TCUT > TFILL + TLF THEN
  CLS : LOCATE 10, 2
  PRINT "REQUIRED CUT EXCEEDS REQUIRED FILL "
  PRINT USING "AND LANDFILL CAPACITY BY ##,###,###,###"; TCUT - (TFILL - TLF); : PRINT "CY"
  PRINT
150 INPUT "DO YOU WANT TO ADD MORE LANDFILL CAPACITY (Y/N)"; ASS
  IF ASS = "Y" THEN
    LFFLAG = 1: GOTO 130
  ELSEIF ASS <> "N" THEN
    PRINT " PLEASE ANSWER Y OR N "
    GOTO 150
  END IF
  NOXLF = 1
END IF
155 CLS
LOCATE 12, 12
INPUT "DO YOU WANT TO PRINT YOUR INPUT DATA (Y/N)"; AS
IF AS = "Y" THEN
  ON ERROR GOTO HANDLER
  LPRINT "      PROJECT "; PRONAMES; " INPUT DATA"
  LPRINT "-----"
  LPRINT " SECTION      VOLUME   SHRINK   SWELL"
  LPRINT " NUMBER      (+) BANK  FACTOR   FACTOR"
  LPRINT "              (-) SHRINK      "
  LPRINT "-----"
  COUNTER2 = 0
  FOR G = 1 TO NUMSEC
  IF VOL(G) > 0 THEN
    LPRINT USING " ##      #,###,####      #.##      #.## "; G; VOL(G); SHRINK(G);
    SWELL(G)
  ELSE
    LPRINT USING " ##      #,###,#### "; G; VOL(G)
  END IF
  COUNTER2 = COUNTER2 + 1
  IF COUNTER2 = 50 THEN
    LPRINT CHR$(12)
    COUNTER2 = 0
    LPRINT : LPRINT
    LPRINT "      PROJECT "; PRONAMES; " INPUT DATA"
    LPRINT "-----"
    LPRINT " SECTION      VOLUME   SHRINK   SWELL"
    LPRINT " NUMBER      (+) BANK  FACTOR   FACTOR"
    LPRINT "              (-) SHRINK      "
    LPRINT "-----"

```



```

END IF
NEXT G
LPRINT CHR$(12)
IF NBP <> 0 THEN
  COUNTER2 = 0
  LPRINT : LPRINT
  LPRINT " BORROW PIT DATA FOR PROJECT: "; PRONAMES
  LPRINT " _____"
  LPRINT : LPRINT
  LPRINT " BORROW PIT CAPACITY ENTERING DISTANCE SWELL SHRINK "
  LPRINT " NUMBER (CY) SECTION X100 (FT) FACTOR FACTOR "
  LPRINT " _____"
  FOR K = 1 TO NBP
    LPRINT USING " ### #,###,### ### #,### #.## #.## "; K; CBP(K);
    ES(K); BPD(K); SWELL(NUMSEC + K); SHRINK(NUMSEC + K)
    COUNTER2 = COUNTER2 + 1
  IF COUNTER2 = 50 THEN
    LPRINT CHR$(12)
    COUNTER2 = 0
    LPRINT : LPRINT
    LPRINT " BORROW PIT DATA FOR PROJECT: "; PRONAMES
    LPRINT " _____"
    LPRINT : LPRINT
    LPRINT " BORROW PIT CAPACITY ENTERING DISTANCE SWELL
    SHRINK "
    LPRINT " NUMBER (CY) SECTION x100 (FT) FACTOR FACTOR "
    LPRINT " _____"
  END IF
  NEXT
  LPRINT CHR$(12)
END IF
IF NLF <> 0 THEN
  COUNTER2 = 0
  LPRINT : LPRINT
  LPRINT " LANDFILL DATA FOR PROJECT: "; PRONAMES
  LPRINT " _____"
  LPRINT : LPRINT
  LPRINT " LANDFILL CAPACITY DEPARTING DISTANCE"
  LPRINT " NUMBER (CY) SECTION x100 (FT)"
  LPRINT " _____"
  FOR H = 1 TO NLF
    LPRINT USING " ## #,###,### ### #,###"; H; CLF(H); DS(H); LFD(H)
    COUNTER2 = COUNTER2 + 1
  IF COUNTER2 = 50 THEN
    LPRINT CHR$(12)
    COUNTER2 = 0
    LPRINT : LPRINT
    LPRINT " LANDFILL DATA FOR PROJECT: "; PRONAMES
    LPRINT " _____"
    LPRINT : PRINT
    LPRINT " LANDFILL CAPACITY DEPARTING DISTANCE"
    LPRINT " NUMBER (CY) SECTION x100 (FT)"
    LPRINT " _____"
  END IF
  NEXT
  LPRINT CHR$(12)
END IF
ELSEIF AS <> "N" THEN

```

```

LOCATE 14, 12
PRINT " PLEASE ANSWER WITH Y OR N ": GOTO 155
END IF
CLS : LOCATE 10
PRINT " *****"
PRINT " * * * * * "
PRINT " * SECTION TO SECTION MATCHING * "
PRINT " * * * * * "
PRINT " *****"
G = 1
IF VOL(G) <= 0 THEN C = 1: GOTO 190
165 G = G + 1
IF G = NUMSEC + 1 THEN 220
IF VOL(G) >= 0 THEN 165
F = G
C = G - 1
170 IF VOL(C) + VOL(F) < 0 THEN 180
XA(C, F) = ABS(VOL(F))
VOL(C) = VOL(C) + VOL(F)
VOL(F) = 0
F = F + 1
IF F = NUMSEC + 1 THEN 220
IF VOL(F) < 0 THEN 170
G = F
GOTO 165
180 XA(C, F) = VOL(C)
VOL(F) = VOL(F) + VOL(C)
VOL(C) = 0
C = C - 1
IF C = 0 THEN 190
IF VOL(C) > 0 THEN 170
190 C = C + 1
IF C = NUMSEC + 1 GOTO 220
IF VOL(C) <= 0 GOTO 190
F = C - 1
200 IF VOL(C) + VOL(F) < 0 THEN 210
XA(C, F) = ABS(VOL(F))
VOL(C) = VOL(C) + VOL(F)
VOL(F) = 0
F = F - 1
IF F <> 0 THEN
    IF VOL(F) < 0 THEN 200
END IF
G = C
GOTO 165
210 XA(C, F) = VOL(C)
VOL(F) = VOL(C) + VOL(F)
VOL(C) = 0
C = C + 1
IF C = NUMSEC + 1 THEN 220
IF VOL(C) >= 0 THEN 200
GOTO 190
220 FOR C = 1 TO NUMSEC
FOR F = 1 TO NUMSEC
IF XA(C, F) <> 0 THEN
    TSCYS = TSCYS + XA(C, F) * SWELL(C) / SHRINK(C) * ABS(C - F) * UNITPRICE / 100
    TOTVOL = TOTVOL + XA(C, F) * SWELL(C) / SHRINK(C)
    TOTSHRVOL = TOTSHRVOL + XA(C, F)

```

```

END IF
NEXT F
NEXT C
M8 = 0: N8 = 0
FOR G = 1 TO NUMSEC
IF VOL(G) > 0 THEN
    M8 = M8 + 1
ELSEIF VOL(G) < 0 THEN
    N8 = N8 + 1
END IF
NEXT G
CLS : LOCATE 10
PRINT " *****"
PRINT " * * * * * "
PRINT " * TRANSPORTATION ALGORITHM WORKING * "
PRINT " * * * * * "
PRINT " *****"
M9 = M8 + NBP
N9 = N8 + NLF
S = M9 + 1: D = N9 + 1: R = D + S
DIM A(R), B(R), C(S, D), D(D), S(S), K(D), R(S)
DIM U(S), V(D), E(R), F(R), H(R), T(R), X(R)
DIM CLOC(S), FLOC(D)
I = 0: J = 0
FOR G = 1 TO NUMSEC
IF VOL(G) > 0 THEN
    I = I + 1
    CLOC(I) = G
    S(I) = VOL(G)
ELSEIF VOL(G) < 0 THEN
    J = J + 1
    FLOC(J) = G
    D(J) = ABS(VOL(G))
END IF
NEXT G
FOR I = 1 TO M8
FOR J = 1 TO N8
C(I, J) = ABS(CLOC(I) - FLOC(J))
NEXT J
NEXT I
IF NBP <> 0 THEN
FOR K = 1 TO NBP
I = M8 + K
S(I) = CBP(K)
FOR J = 1 TO N8
C(I, J) = ABS(FLOC(J) - ES(K)) + BPD(K)
NEXT J
NEXT K
END IF
IF NLF <> 0 THEN
FOR H = 1 TO NLF
J = N8 + H
D(J) = CLF(H)
FOR I = 1 TO M8
C(I, J) = ABS(CLOC(I) - DS(H)) + LFD(H)
NEXT I
NEXT H
IF NBP <> 0 THEN

```

```

        FOR I = (M8 + 1) TO M9
        FOR J = (N8 + 1) TO N9
        C(I, J) = 90000!
        NEXT J
        NEXT I
    END IF
END IF
DXO = 0
FOR J = 1 TO N9
DXO = DXO + D(J)
NEXT J
SO = 0
FOR I = 1 TO M9
SO = SO + S(I)
NEXT I
IF SO < DXO THEN
    M = M9 + 1
    N = N9
    S(M) = DXO - SO
    FOR J = 1 TO N
    C(M, J) = 0
    NEXT J
ELSEIF SO > DXO THEN
    M = M9
    N = N9 + 1
    D(N) = SO - DXO
    FOR I = 1 TO M
    C(I, N) = 0
    NEXT I
ELSE
    M = M9
    N = N9
END IF
M1 = M + N - 1
323 GOSUB DETERMINE
325 GOSUB INITIALIZE
IF I9 = 0 THEN 548
GOSUB FIND
GOSUB IMPROVE
GOTO 325
DETERMINE:
E7 = .001
FOR I = 1 TO M
R(I) = E7
U(I) = S(I) + E7
NEXT I
FOR J = 1 TO N - 1
K(J) = 0
V(J) = D(J)
NEXT J
K(N) = M * E7
V(N) = D(N) + K(N)
L1 = 0
344 FOR L = 1 TO M
IF U(L) <> 0 THEN 348
NEXT L
GOTO 373
348 I = L

```

```

349 CO = 1E+20
FOR J = 1 TO N
IF V(J) < 0 THEN
    IF CO > C(I, J) THEN
        CO = C(I, J)
        L = J
    END IF
END IF
NEXT J
J = L
L1 = L1 + 1
A(L1) = I
B(L1) = J
IF U(I) <= V(J) THEN
    X(L1) = U(I)
    V(J) = V(J) - U(I)
    U(I) = 0
    K(J) = K(J) - R(I)
    T(L1) = R(I)
    GOTO 344
ELSE
    X(L1) = V(J)
    U(I) = U(I) - V(J)
    V(J) = 0
    R(I) = R(I) - K(J)
    T(L1) = K(J)
    GOTO 349
END IF
373 IF L1 < M1 THEN
    PRINT "ERROR, No. OF BASIC CELLS ARE NOT EQUAL TO M+N+1"
    STOP
ELSE
    FOR L = 1 TO M1
        X(L) = X(L) - T(L)
        IF X(L) < E7 THEN X(L) = 0
    NEXT L
END IF
RETURN
INITIALIZE:
FOR I = 1 TO M
    R(I) = -1
NEXT I
FOR J = 1 TO N
    K(J) = -1
NEXT J
M3 = 1
N3 = 0
I = A(1)
U(I) = 0
R(I) = 0
FOR L = 1 TO M1
    IF A(L) = I THEN
        J = B(L)
        K(J) = I
        V(J) = C(I, J)
        N3 = N3 + 1
    END IF
NEXT L

```

```

404 FOR J = 1 TO N
IF K(J) >= 0 THEN
    FOR L = 1 TO M1
    IF B(L) = J THEN
        IF R(A(L)) < 0 THEN
            I = A(L)
            R(I) = J
            U(I) = C(I, J) - V(J)
            M3 = M3 + 1
        END IF
    END IF
    END IF
NEXT L
END IF
NEXT J
LO = 1
GOTO 432
418 FOR I = 1 TO M
IF R(I) > 0 THEN
    FOR L = 1 TO M1
    IF A(L) = I THEN
        IF K(B(L)) <= 0 THEN
            J = B(L)
            K(J) = I
            V(J) = C(I, J) - U(I)
            N3 = N3 + 1
        END IF
    END IF
    END IF
NEXT L
END IF
NEXT I
LO = 2
432 IF N3 = N THEN
    IF M3 < M THEN
        IF LO = 1 THEN 418
        GOTO 404
    END IF
ELSE
    IF LO = 1 THEN 418
    GOTO 404
END IF
GXO = 0!
I9 = 0
FOR I = 1 TO M
FOR J = 1 TO N
G1 = U(I) + V(J) - C(I, J)
IF G1 > 1E-08 THEN
    IF G1 > GXO THEN
        GXO = G1
        I9 = I
        J9 = J
    END IF
END IF
NEXT J
NEXT I
RETURN
FIND:
H(1) = I9
LO = I9

```

```

L = 0
455 L = L + 1
IF H(L) >= 0 THEN
    IF R(LO) = 0 THEN 466
    H(L + 1) = -R(LO)
    R(LO) = -R(LO)
    LO = -R(LO)
    GOTO 455
END IF
H(L + 1) = K(LO)
K(LO) = -K(LO)
LO = -K(LO)
GOTO 455
466 HO = L
T(1) = -J9
LO = J9
L = 0
471 L = L + 1
IF T(L) >= 0 THEN
    IF R(LO) > 0 THEN
        T(L + 1) = -R(LO)
        R(LO) = -R(LO)
        LO = -R(LO)
        GOTO 471
    END IF
ELSE
    IF K(LO) > 0 THEN
        T(L + 1) = K(LO)
        K(LO) = -K(LO)
        LO = -K(LO)
        GOTO 471
    END IF
END IF
TT = L
LO = T(TT)
FOR L = 1 TO HO
    IF H(L) = LO THEN 489
NEXT L
489 HO = L
E(1) = I9
F(1) = J9
L1 = 1
HO = HO - 1
IF HO > 0 THEN
    FOR L = 1 TO HO
        L1 = L1 + 1
        IF H(L) <= 0 THEN
            E(L1) = H(L + 1)
            F(L1) = -H(L)
        ELSE
            E(L1) = H(L)
            F(L1) = -H(L + 1)
        END IF
    NEXT L
END IF
TT = TT - 1
IF TT > 0 THEN
    FOR L = 1 TO TT

```

```

L1 = L1 + 1
J = TT - L + 2
IF T(J) >= 0 THEN
    E(L1) = T(J)
    F(L1) = -T(J - 1)
ELSE
    E(L1) = T(J - 1)
    F(L1) = -T(J)
END IF
NEXT L
END IF
RETURN
IMPROVE:
FOR L = 2 TO L1
FOR J = 1 TO M1
IF E(L) = A(J) THEN
    IF F(L) = B(J) THEN
        H(L) = J
        GOTO 528
    END IF
END IF
NEXT J
528 NEXT L
XO = 1E+20
L2 = L1 / 2
FOR L = 1 TO L2
L3 = L * 2
IF X(H(L3)) < XO THEN
    XO = X(H(L3))
    L4 = H(L3)
END IF
NEXT L
X(L4) = X(L4) + XO
A(L4) = I9
B(L4) = J9
FOR L = 1 TO L2
I = L * 2
J = I - 1
IF J < 1 THEN
    X(H(J)) = X(H(J)) + XO
END IF
X(H(I)) = X(H(I)) - XO
NEXT L
RETURN
548 FOR L = 1 TO M1
IF X(L) < 0 THEN
    I = A(L): J = B(L)
    IF C(I, J) < 90000! THEN
        IF I <= M8 AND J <= N8 THEN SV = CLOC(I)
        IF I > M8 AND I <= M9 AND J <= N8 THEN SV = I - M8 + NUMSEC
        IF I <= M8 AND J > N8 AND J <= N9 THEN SV = CLOC(I)
        IF I <= M9 AND J <= N9 THEN
            TSCYS = TSCYS + X(L) * SWELL(SV) / SHRINK(SV) * C(I, J) * UNITPRICE / 100
            TOTVOL = TOTVOL + X(L) * SWELL(SV) / SHRINK(SV)
            TOTSHRVOL = TOTSHRVOL + X(L)
        END IF
    END IF
END IF
END IF

```



```

NEXT L
CLS
SCREEN 8
COLOR 2, 0
LOCATE 1, 2: PRINT "EARTH DISTRIBUTION PLAN: "; LEFT$(PRONAMES, 12)
LOCATE 2, 2: PRINT "-----"
R = 50 / BIGVOL
S = 580 / NUMSEC
IF S < 3 THEN
  CLS
  LOCATE 8, 12
  PRINT "NUMBER OF SECTION TOO LARGE FOR GRAPHICAL DISPLAY"
  GRAPHFLAG = 1
  GOTO 653
END IF
FOR I = 1 TO NUMSEC
  IF SVOL(I) > 0 THEN
    LINE (S * I, 100) - ((I + 1) * S, 100 - SVOL(I) * R - 1), 2, BF
  ELSEIF SVOL(I) < 0 THEN
    LINE (S * I, 100) - ((I + 1) * S, 100 - SVOL(I) * R + 1), 2, BF
  END IF
NEXT I
FOR K = 1 TO NBP
  LINE (ES(K) * S, 75) - ((ES(K) + 1) * S, 75 - CBP(K) * R), 2, BF
NEXT K
FOR H = 1 TO NLF
  LINE (DS(H) * S, 175) - ((DS(H) + 1) * S, 175 - CLF(H) * R), 2, BF
NEXT H
LOCATE 23, 1: INPUT "PRESS ENTER TO CONTINUE"; ASS
FOR I = 1 TO NUMSEC
  FOR J = 1 TO NUMSEC
    IF XA(I, J) <> 0 THEN
      LOCATE 5, 1
      PRINT USING "CUT ## TO FILL ## AMOUNT IS #####.# C.Y. "; I; J; XA(I, J)
      FOR N = 1 TO XA(I, J) * R
        LINE (I * S + 1, (99 - (SVOL(I) * R)) + N) - ((I + 1) * S - 1, (99 - (SVOL(I) * R)) + N), 0
        LINE (J * S + 1, (101 - SVOL(J) * R) - N) - ((J + 1) * S - 1, (101 - (SVOL(J) * R)) - N), 1
      NEXT N
      SVOL(I) = SVOL(I) - XA(I, J)
      SVOL(J) = SVOL(J) + XA(I, J)
      LINE (I * S, 100) - ((I + 1) * S, 100), 2
      FOR T = 1 TO 150
        NEXT T
      LOCATE 23, 1
      INPUT "PRESS ENTER TO CONTINUE"; ASS
    END IF
  NEXT J
NEXT I
FOR L = 1 TO M1
  IF X(L) < .1 THEN 628
  LOCATE 5, 1: PRINT " "
  I = A(L)
  J = B(L)
  IF I <= M8 AND J <= N8 THEN
    LOCATE 5, 1
    PRINT USING "CUT ## TO FILL ## AMOUNT IS #####.# C.Y."; CLOC(I); FLOC(J); X(L)
    FOR N = 1 TO X(L) * R

```

```

LINE (CLOC(I) * S + 1, (99 - (SVOL(CLOC(I)) * R)) + N) - ((CLOC(I) + 1) * S - 1, (99 - (SVOL(CLOC(I)) *
R)) + N), 0
LINE (FLOC(J) * S + 1, 101 - SVOL(FLOC(J)) * R - N) - ((FLOC(J) + 1) * S - 1, 101 - SVOL(FLOC(J)) * R
- N), 1
NEXT N
SVOL(CLOC(I)) = SVOL(CLOC(I)) - X(L)
SVOL(FLOC(J)) = SVOL(FLOC(J)) + X(L)
LINE (CLOC(I) * S, 100) - ((CLOC(I) + 1) * S, 100), 2
LOCATE 23, 1
INPUT "PRESS ENTER TO CONTINUE"; ASS$
ELSEIF I > M8 AND I <= M9 AND J <= N8 THEN
K = I - M8
LOCATE 5, 1
PRINT USING "B. PIT ## TO FILL ## AMOUNT IS #####.# C.Y."; K; FLOC(J); X(L)
FOR N = 1 TO X(L) * R
LINE (ES(K) * S + 1, 75 - (CBP(K) * R) + N) - ((ES(K) + 1) * S - 1, 75 - (CBP(K) * R) + N), 0
LINE (FLOC(J) * S + 1, 101 - SVOL(FLOC(J)) * R - N) - ((FLOC(J) + 1) * S - 1, 101 - SVOL(FLOC(J)) * R
- N), 1
NEXT N
SVOL(FLOC(J)) = SVOL(FLOC(J)) + X(L)
CBP(K) = CBP(K) - X(L)
LINE (ES(K) * S, 75) - ((ES(K) + 1) * S, 75), 2
LOCATE 23, 1
INPUT "PRESS ENTER TO CONTINUE"; ASS$
ELSEIF J > N8 AND J <= N9 AND I <= M8 THEN
H = J - N8
LOCATE 5, 1
PRINT USING "CUT ## TO L. FILL ## AMOUNT IS #####.# C.Y."; CLOC(I); H; X(L)
FOR N = 1 TO X(L) * R
LINE (CLOC(I) * S + 1, (99 - (SVOL(CLOC(I)) * R)) + N) - ((CLOC(I) + 1) * S - 1, (99 - (SVOL(CLOC(I)) *
R)) + N), 0
LINE (DS(H) * S + 1, 175 - (CLF(H) * R) + N) - ((DS(H) + 1) * S - 1, 175 - (CLF(H) * R) + N), 1
NEXT N
CLF(H) = CLF(H) - X(L)
LINE (CLOC(I) * S, 100) - ((CLOC(I) + 1) * S, 100), 2
SVOL(CLOC(I)) = SVOL(CLOC(I)) - X(L)
LINE (DS(H) * S, 175) - ((DS(H) + 1) * S, 175), 2
LOCATE 23, 1
INPUT "PRESS ENTER TO CONTINUE"; ASS$
END IF
628 NEXT L
LOCATE 4, 1: PRINT ""
TSCYS = TSCYS * 100
LOCATE 4, 1
PRINT USING "      TOTAL COST IN $ IS = ###,###,###,###"; TSCYS: LOCATE 5, 1
PRINT USING "      TOTAL VOLUME IN C.Y. IS = ###,###,###"; TOTVOL
LOCATE 23, 1: INPUT "PRESS ENTER TO CONTINUE"; ASS$
LOCATE 23, 1: PRINT ""
SCREEN 0: WIDTH 80
IF NOXBP = 1 THEN
REMAINFLAG = 1
PRINT : PRINT : PRINT
LOCATE 4, 6: PRINT "DUE TO INSUFFICIENT CUT AND BORROW,"
LOCATE 6, 6: PRINT "THE FOLLOWING FILL SECTIONS REMAIN:": PRINT : PRINT
FOR L = 1 TO M1
IF X(L) <> 0 THEN
I = A(L)
J = B(L)

```

```

        IF I = M9 + 1 AND J <= N8 THEN PRINT " FILL SECTION"; FLOC(J); : PRINT USING " VOLUME =
        #,###,###"; X(L)
    END IF
NEXT L
LOCATE 23, 1: INPUT "PRESS RETURN TO CONTINUE", ASS
ELSEIF NOXLF = 1 THEN
    REMAINFLAG = 1
    LOCATE 4, 6: PRINT "DUE TO INSUFFICIENT LANDFILL CAPACITY,"
    LOCATE 6, 6: PRINT " THE FOLLOWING CUT SECTIONS REMAIN:": PRINT : PRINT
    FOR L = 1 TO M1
        IF X(L) <> 0 THEN
            I = A(L)
            J = B(L)
            IF I <= M8 AND J <= N9 + 1 THEN PRINT " CUT SECTION"; CLOC(I); : PRINT USING " VOLUME =
            #,###,###"; X(L)
        END IF
    NEXT L
    LOCATE 23, 1: INPUT "PRESS RETURN TO CONTINUE", ASS
    END IF
653 IF GRAPHFLAG = 1 THEN
    TSCYS = TSCYS * 100
    FOR I = 1 TO NUMSEC
        FOR J = 1 TO NUMSEC
            SVOL(I) = SVOL(I) - XA(I, J)
            SVOL(J) = SVOL(J) + XA(I, J)
        NEXT J
    NEXT I
    FOR L = 1 TO M1
        I = A(L)
        J = B(L)
        IF I <= M8 AND J <= N8 THEN
            SVOL(CLOC(I)) = SVOL(CLOC(I)) - X(L)
            SVOL(FLOC(J)) = SVOL(FLOC(J)) + X(L)
        END IF
        IF I > M8 AND I <= M9 AND J <= N8 THEN
            SVOL(FLOC(J)) = SVOL(FLOC(J)) + X(L)
        END IF
        IF J > N8 AND J <= N9 AND I <= M8 THEN
            SVOL(CLOC(I)) = SVOL(CLOC(I)) - X(L)
        END IF
    NEXT L
    END IF
END IF
IF NOXBP <> 1 THEN
    IF NOXLF <> 1 THEN
        CLS
        FOR I = 1 TO NUMSEC
            IF ABS(SVOL(I)) > .1 THEN
                REMAINFLAG = 1
                PRINT "SECTION ("; I; : PRINT USING ") HAS A REMAINING VOLUME OF
                #,###,### CY"; SVOL(I)
                COUNTER3 = COUNTER3 + 1
                IF COUNTER3 = 16 THEN
                    COUNTER3 = 0
                    PRINT "PRESS ANY KEY TO CONTINUE"
                    A$ = INPUT$(1)
                END IF
            END IF
        NEXT
    END IF
NEXT

```

```

        END IF
    END IF
    IF REMAINFLAG = 1 THEN
        CLS
        PRINT
        PRINT " RECOMENDATIONS:"
        PRINT
        PRINT " RUN THE PROGRAM AGAIN DOING THE FOLLOWING CHANGES:"
        PRINT ""
        PRINT " - FOR A REMAINING FILL INCREASE THE CLOSER BORROW PIT "
        PRINT " BY A VOLUME EQUAL TO THE REMAINING VOLUME DIVIDED "
        PRINT " BY THE BORROW PIT SHRINK FACTOR"
        PRINT ""
        PRINT " - FOR A REMAINING CUT INCREASE THE CLOSER LANDFILL "
        PRINT " BY A VOLUME EQUAL TO THE REMANING VOLUME."
        PRINT
        PRINT " STRIKE ANY KEY TO CONTINUE"
        PAUSES = INPUT$(1)
    END IF
    654 CLS
    LOCATE 12, 12
    INPUT "DO YOU WANT A PRINTED COPY OF YOUR RESULTS (Y/N)"; ASS
    IF ASS = "N" THEN
        GOTO 699
    ELSEIF ASS <> "Y" THEN
        PRINT " PLEASE ANSWER WITH (Y) OR (N)"
        PRINT "":
        INPUT " *****PRESS ENTER TO CONTINUE*****"; QQQQQ
        GOTO 654
    END IF
    CLS : LOCATE 12, 12: PRINT "PRINTER WORKING"
    ON ERROR GOTO HANDLER
    LPRINT : LPRINT " PROJECT: "; PRONAMES
    LPRINT : LPRINT " PROJECT NUMBER :"; PNUMS
    LPRINT : LPRINT : LPRINT " EARTH DISTRIBUTION PLAN"
    LPRINT " -----": LPRINT
    COUNTER = 0
    LPRINT " FROM CUT TO FILL DISTANCE SHRINK LOOSE "
    LPRINT " SECTION SECTION x100 FT VOLUME VOLUME"
    FOR C = 1 TO NUMSEC
    FOR F = 1 TO NUMSEC
    IF XA(C, F) >= 1 THEN
        LPRINT USING " ### ## ### ###,### ###,### "; C; F; ABS(C - F); XA(C,
        F); XA(C, F) * SWELL(C) / SHRINK(C)
        COUNTER = COUNTER + 1
    IF COUNTER = 46 THEN
        LPRINT CHR$(12)
        COUNTER = 0
        LPRINT " FROM CUT TO FILL DISTANCE SHRINK LOOSE "
        LPRINT " SECTION SECTION x100 FT VOLUME VOLUME"
    END IF
    END IF
    NEXT F
    NEXT C
    FOR L = 1 TO M1
    IF X(L) >= 1 THEN
        I = A(L)
        J = B(L)

```

```

IF I <= M8 AND J <= N8 THEN
  LPRINT USING "    ###    ##    ####    ##,###    ##,### "; CLOC(I); FLOC(J);
  C(I, J); X(L); X(L) * SWELL(CLOC(I)) / SHRINK(CLOC(I))
  COUNTER = COUNTER + 1
  IF COUNTER = 46 THEN
    LPRINT CHR$(12)
    COUNTER = 0
    LPRINT "    FROM CUT    TO FILL    DISTANCE    SHRINK    LOOSE "
    LPRINT "    SECTION    SECTION    x100 FT    VOLUME    VOLUME"
  END IF
END IF
END IF
NEXT L
LPRINT ""
LPRINT "    FROM BORROW TO FILL    DISTANCE    SHRINK    LOOSE "; ""
LPRINT "    PIT    SECTION    x100 FT    VOLUME    VOLUME"
FOR L = 1 TO M1
  IF X(L) >= 1 THEN
    I = A(L)
    J = B(L)
    IF I > M8 AND I <= M9 AND J <= N8 THEN
      LPRINT USING "    ###    ##    ####    ##,###    ##,### "; I - M8; FLOC(J);
      C(I, J); X(L); X(L) * SWELL(I - M8 + NUMSEC) / SHRINK(I - M8 + NUMSEC)
      COUNTER = COUNTER + 1
      IF COUNTER = 45 THEN
        LPRINT CHR$(12)
        COUNTER = 0
        LPRINT "    FROM BORROW TO FILL    DISTANCE    SHRINK    LOOSE "; ""
        LPRINT "    PIT    SECTION    x100 FT    VOLUME    VOLUME"
      END IF
    END IF
  END IF
END IF
NEXT L
LPRINT ""
LPRINT "    FROM CUT    TO LANDFILL DISTANCE    SHRINK    LOOSE "
LPRINT "    SECTION    AREA    x100 FT    VOLUME    VOLUME"
FOR L = 1 TO M1
  IF X(L) >= 1 THEN
    I = A(L)
    J = B(L)
    IF I <= M8 AND J > N8 AND J <= N9 THEN
      LPRINT USING "    ###    ##    ####    ##,###    ##,### "; CLOC(I); J - N8;
      C(I, J); X(L); X(L) * SWELL(CLOC(I)) / SHRINK(CLOC(I))
      COUNTER = COUNTER + 1
      IF COUNTER = 50 THEN
        LPRINT CHR$(12)
        COUNTER = 0
        LPRINT "    FROM CUT    TO LANDFILL DISTANCE    SHRINK    LOOSE "
        LPRINT "    SECTION    AREA    x100 FT    VOLUME    VOLUME"
      END IF
    END IF
  END IF
END IF
NEXT L
LPRINT : LPRINT : LPRINT : LPRINT
LPRINT USING "    TOTAL HAULAGE COST IN THIS PROJECT IS= $ ##,###,###,###"; TSCYS
LPRINT : LPRINT USING "    TOTAL LOOSE VOLUME IN CY IS= ##,###,###"; TOTVOL
LPRINT : LPRINT USING "    TOTAL SHRINK VOLUME IN CY IS= ##,###,###"; TOTSHRVOL
LPRINT CHR$(12)

```

```

IF NOXBP = 1 THEN
  COUNTER = 0
  LPRINT : LPRINT : LPRINT : LPRINT : LPRINT
  LPRINT " DUE TO INSUFFICIENT CUT AND BORROW,"
  LPRINT " THE FOLLOWING FILL SECTIONS REMAIN:"
  LPRINT " (SHRINK VOLUME)": LPRINT
  FOR L = 1 TO M1
    IF X(L) < 0 THEN
      I = A(L)
      J = B(L)
      IF I = M9 + 1 AND J <= N8 THEN LPRINT USING " FILL SECTION ###"; FLOC(J); :
      LPRINT USING " - VOLUME = #,###,###"; X(L)
      COUNTER = COUNTER + 1
    END IF
    IF COUNTER = 45 THEN
      LPRINT CHR$(12)
      COUNTER = 0
    END IF
  NEXT L
END IF
IF NOXLF = 1 THEN
  COUNTER = 0
  LPRINT : LPRINT : LPRINT : LPRINT : LPRINT
  LPRINT " DUE TO INSUFFICIENT LANDFILL CAPACITY,"
  LPRINT " THE FOLLOWING SECTIONS REMAIN:"
  LPRINT " (SHRINK VOLUME)": LPRINT
  FOR L = 1 TO M1
    IF X(L) < 0 THEN
      I = A(L)
      J = B(L)
      IF I <= M8 AND J = N9 + 1 THEN LPRINT USING " CUT SECTION ###"; CLOC(I); :
      LPRINT USING " - VOLUME = #,###,### "; X(L)
      COUNTER = COUNTER + 1
    END IF
    IF COUNTER = 45 THEN
      LPRINT CHR$(12)
      COUNTER = 0
    END IF
  NEXT L
END IF
IF NOXBP < 1 THEN
  IF NOXLF < 1 THEN
    COUNTER = 0
    FOR I = 1 TO NUMSEC
      IF ABS(SVOL(I)) > 1 THEN
        LPRINT : LPRINT : LPRINT
        LPRINT "SECTION ("; I; ") HAS A REMAINING VOLUME OF "; SVOL(I)
        COUNTER = COUNTER + 1
        IF COUNTER = 45 THEN
          LPRINT CHR$(12)
          COUNTER = 0
        END IF
      END IF
    NEXT I
  END IF
END IF
END IF

```

```

IF REMAINFLAG = 1 THEN
  IF COUNTER > 45 THEN LPRINT CHR$(12)
  LPRINT : LPRINT : LPRINT
  LPRINT " RECOMENDATIONS:"
  LPRINT
  LPRINT " RUN THE PROGRAM AGAIN DOING THE FOLLOWING CHANGES:"
  LPRINT ""
  LPRINT " - FOR A REMAINING FILL INCREASE THE CLOSER BORROW PIT "
  LPRINT " BY A VOLUME EQUAL TO THE REMAINING VOLUME DIVIDED "
  LPRINT " BY THE BORROW PIT SHRINK FACTOR"
  LPRINT ""
  LPRINT " - FOR A REMAINING CUT INCREASE THE CLOSER LANDFILL "
  LPRINT " BY A VOLUME EQUAL TO THE REMANING VOLUME."
  LPRINT
  LPRINT CHR$(12)
END IF
699 CLS
LOCATE 12, 12
INPUT "DO YOU WANT TO RUN THE PROGRAM AGAIN WITH NEW DATA(Y/N)"; CS
IF CS <> "N" THEN
  IF CS = "Y" THEN
    CLEAR
    GOTO 5
  ELSE
    PRINT "PLEASE ANSWER WITH (Y) OR (N)"
    PRINT "": PRINT ""
    INPUT "***** PRESS ENTER TO CONTINUE*****"; CS
    GOTO 699
  END IF
END IF
END IF
CLS : LOCATE 12, 20: PRINT " THE END"
END
703 CLS : LOCATE 10, 6
PRINT "INSUFFICIENT LANDFILL FOR OPTIMAL SOLUTION"
PRINT "-----"
PRINT : PRINT "DO YOU WANT TO INCREASE LANDFILL CAPACITY (TYPE Y),"
PRINT : INPUT " OR CONTINUE WITH LESS THAN OPTIMAL SOLUTION (TYPE N)", AS$
IF AS <> "N" THEN
  IF AS = "Y" THEN
    CLEAR
    GOTO 130
  ELSE
    PRINT "PLEASE ANSWER WITH (Y) OR (N)"
    PRINT "": PRINT ""
    INPUT "***** PRESS ENTER TO CONTINUE*****"; CS
    GOTO 703
  END IF
END IF
END IF
C(I, J) = 90000!: INSUF = 1: GOTO 323
HANDLER:
SELECT CASE ERR
CASE 25
  CLS : LOCATE 12, 12
  PRINT "DEVICE FAULT, PLEASE CONNECT THE PRINTER"
  LOCATE 14, 12
  PRINT "PRESS ANY KEY TO CONTINUE "
  PAUSE$ = INPUT$(1)
  RESUME

```

CASE 27

```
CLS : LOCATE 12, 12
PRINT "PRINTER OUT OF PAPER, PLEASE FEED THE PRINTER"
LOCATE 14, 12
PRINT "PRESS ANY KEY TO CONTINUE "
PAUSES = INPUT$(1)
RESUME
```

CASE 53

```
CLS : LOCATE 12, 12
PRINT "FILE NOT FOUND, PLEASE ENTER DRIVE AND FILE NAME AGAIN"
LOCATE 14, 12
INPUT "FROM WICH DRIVE ex. B: "; DRIVES
LOCATE 16, 12
INPUT "NAME OF THE INPUT FILE"; NIF$
RESUME
```

CASE 54

```
CLS : LOCATE 12, 12
PRINT "BAD FILE MODE, PLEASE ENTER DRIVE AND FILE NAME AGAIN"
LOCATE 14, 12
INPUT "FROM WICH DRIVE ex. B: "; DRIVES
LOCATE 16, 12
INPUT "NAME OF THE INPUT FILE"; NIF$
RESUME
```

CASE 57

```
CLS : LOCATE 12, 12
PRINT "DEVICE I/O ERROR"
LOCATE 14, 12
PRINT "PRESS ANY KEY TO CONTINUE "
PAUSES = INPUT$(1)
RESUME
```

CASE 71

```
CLS : LOCATE 12, 12
PRINT "DISK NOT READY, PLEASE CHECK DISK DRIVE"
LOCATE 14, 12
PRINT "PRESS ANY KEY TO CONTINUE "
PAUSES = INPUT$(1)
RESUME
```

CASE 64

```
CLS : LOCATE 12, 12
PRINT "BAD FILE NAME, PLEASE ENTER DRIVE AND FILE NAME AGAIN"
LOCATE 14, 12
INPUT "FROM WICH DRIVE ex. B: "; DRIVES
LOCATE 16, 12
INPUT "NAME OF THE INPUT FILE"; NIF$
RESUME
```

END SELECT

APPENDIX F:
PROGRAM DISKETTE

Verbatim

DataLife

