

# Oxford University Computing Laboratory

## Computer Manuals

4063

SCIENTIFIC & TECHNICAL SOFTWARE

Mathematical Programming. Part. **11**

Least Cost Mix

1900

4063(1)



SCIENTIFIC AND TECHNICAL SOFTWARE

# Mathematical Programming

## Part 11 Least Cost Mix

OXFORD UNIVERSITY COMPUTING LABORATORY

Copy 1

COMPUTING SERVICE

4063

Mathematical Programming 11





# MATHEMATICAL PROGRAMMING

## Part II Least Cost Mix

First Edition October 1967

### Amendment List I

Each amendment list contains one or more numbered amendments. These amendments are provided in the form of instructions to replace one or more existing pages, to add one or more new pages, or to alter an existing page.

When all additions and alterations have been made, the contents list and amendment record should be updated, and the amendment list should be filed at the back of the manual.

The following conventions are adopted in amendment lists:

- 1 Line number 1 is the first line following the section heading or, if no section reference is given, the first line of the page.
- 2 Line number -1 is the last line of the section or, if no section reference is given, the last line of the page.
- 3 *All* printed lines, including headings or mathematical expressions, are taken into account when a line number is determined.
- 4 The date of issue appears at the foot of all extra pages and all re-issued pages in the form (month, year).

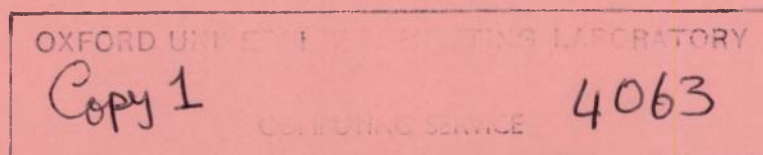
When a page is re-issued, the passages which have been materially altered contain a vertical mark in the outside margin.

- 5 When the recipient is asked to make a minor amendment to a page, he should note the amendment list number and the amendment number in the outside margin by writing, for example, AL3/2 to identify amendment 2 of amendment list 3.

Item	Part	Chapter	Page	Section	Line	Amendment
1		1	11			Remove and destroy. Replace with new page 11 attached.
2		3	19			Remove and destroy. Replace with new page 19 attached.

**International Computers and Tabulators Limited**

I.C.T. House, Putney, London, S.W.15







**1900**  
series

---

SCIENTIFIC AND TECHNICAL SOFTWARE

# **Mathematical Programming**

**Part 11 Least Cost Mix**

---

*Technical Publication 4063*

© *International Computers and Tabulators Limited 1967*

*First Edition October 1967*

*Issued by Technical Publications Department  
International Computers and Tabulators Limited  
Head Office: I.C.T. House, Putney, London S.W.15  
and printed in Great Britain by  
Printing and Stationery Department  
Letchworth, Hertfordshire*

Part II

LEAST COST MIX



CONTENTS

INTRODUCTION	1
Chapter 1 INPUT	5
Paper Tape	5
Punched Cards	5
Group Headings	5
Record Format	6
Groups	6
1 Title	6
2 Parameters	6
Min-max Constraints (Parameter 3)	7
Scaling for Ingredients (Parameter 8)	7
Scaling for Costs (Parameter 9)	7
Output Parameter	7
Name Groups	7
3 Product Names	8
4 Cost Row Names	8
5 Ingredient Names	8
6 Constituent Names	9
7 Analysis Matrix	9
Constituent Scaling	9
Punching	9
Organization of the Group	9
8 Product Specification	10
Constraint Bounds	11

Organization of the Group	11
Equalities	11
9 Exclusion Sets	12
10 Cost Rows	12
11 Triplets	13
End of Data	13
Chapter 2 OUTPUT	15
Output during Input	15
Summary List	15
Names	15
Check Sums	15
Analysis Matrix	15
Product Specifications	16
Exclusion Sets	16
Cost Rows	16
Triplets	16
Error Checking	16
Optional Input Data Printout	17
Output during Optimization	17
Iteration Print (Optional)	17
Output after Optimization	17
Results	17
Solution	17
Alternatives	18
Analysis	18
Active Constraints	18
Chapter 3 OPERATING INSTRUCTIONS	19
Chapter 4 AN EXAMPLE PROGRAM RUN	21
Appendix 1 PROBLEM SIZE LIMITATIONS	29
Appendix 2 DATA SHEETS	31

## INTRODUCTION

This manual describes the Least Cost Mix program which has been written for I.C.T. 1900 series computers. The program is designed to provide solutions for Least Cost Mix problems, particularly those that arise in the animal food-stuff industry.

### THE PROBLEM - GENERAL

The Least Cost Mix problem is that of producing a commodity from a mixture of ingredients, subject to certain constraints, at the least possible cost. When formulated in mathematical terms, the problem is a particular example of the general linear programming problem and the powerful techniques for solving this type of problem may be applied to it.

### THE ANIMAL FOOD - STUFF PROBLEM

Typical examples of the Least Cost Mix problem occur in the animal food-stuff industry, where the following situation is common.

A manufacturer makes several products that are different mixtures of a common set of basic ingredients, such as wheat, fishmeal, groundnut cake and dried milk. These ingredients are analyzed in terms of the amounts of essential constituents which they contain. The constituents are of two types: actual nutrient substances (such as oil, protein, riboflavin, etc.) and non-substantial measures (such as the number of calories contained in a unit of an ingredient). The amounts of these constituents contained in a product determine its value as a food-stuff and also distinguish it from other products. Each product is defined by constraints specifying upper or lower bounds (or both) for some of the constituents and for some of the ingredients contained in it.

There are a very large number of possible, or *feasible* mixtures of the basic ingredients that would satisfy the constraints for each product, and the factor which decides the best mixture from the point of view of the manufacturer is that of cost. Some ingredients are cheaper than others and the manufacturer's problem is to determine the cheapest (or *optimal*) mixture which satisfies the set of constraints for each product.

### SOLVING THE PROBLEM

To solve this problem, the following information must be known:

- 1 The amount of each constituent contained in a unit (e.g. 1 lb or 1 ton) of each ingredient.

This information is assembled in a table called the *Analysis Matrix*.

- 2 The cost of each ingredient per unit weight (e.g. in £ per ton).

The list of the costs for all the ingredients in the analysis matrix is called the *Cost Row*.

- 3 The set of constraints that define each product.

A constraint specifying a lower bound is referred to as a *minimum constraint*, a constraint specifying an upper bound is referred to as a *maximum constraint*. The term *min-max constraint* is a common name for both types.

- 4 The set of *exclusions* to be implemented in the computation of each mixture.

An exclusion is a constraint that specifies the upper bound for an ingredient to be zero, thus completely excluding the ingredient from the mixture. In general, the analysis matrix will be constructed to include every ingredient that is ever available to the manufacturer. The function of an exclusion set is to limit the ingredients included in a computation to those that are currently available.

With this information it is possible to derive an optimal solution (assuming that a solution exists), treating the problem as a conventional linear programming problem.

There are various general purpose linear programming programs available that are suitable for this type of problem. However, a specialized program has been developed which requires no knowledge of linear programming on the part of the user.

In practice, it is usually found that optimal mixtures have to be computed for several sets of cost rows since ingredient prices vary for different suppliers and under changing market conditions. For the same reasons, and for reasons of varying availability, different sets of exclusions must be included in the computation. The specialized Least Cost Mix program enables the user to compute optimal mixtures for any combination from a number of products, cost rows and exclusion sets, while requiring a minimum of data preparation.

The program prints out the results of the computation in a form that is particularly suited to the requirements of the animal food-stuff manufacturer.

## THE LEAST COST MIX PROGRAM

### Configuration

The program may be run on a 1900 configuration having at least 16K core store and four magnetic tape decks, plus either a punched card reader or a paper tape reader and a line printer with at least 120 character positions. Core store requirements are detailed in Appendix 1.

### Data Requirements

The data input to the program is of two types:

- 1 Data defining the range of the problems to be computed.

This consists of the analysis matrix specifying the ingredients and constituents, the sets of constraints defining each product, the sets of exclusions and the various cost rows. The cost rows may be for actual or predicted ingredient costs.

- 2 Data defining the problems to be solved.

Each problem that is to be solved is defined by a *triplet* - a set of three numbers, being the index of the product for which the least cost mix is to be computed together with the indices of the exclusion set and cost row that are to be included in the computation. The program selects the groups of data specified by each triplet and combines them with the analysis matrix to form a linear programming problem. The need for the separate input of data for each problem is thereby avoided.

In addition, a set of parameter values must be input.

The program will accept data input on paper tape or punched cards. A complete description of the input requirements to the program is given in Chapter 1.

## **Results**

The program outputs several lists for each product/exclusion/cost row triplet; these are described fully in Chapter 2. Essentially, they are as follows:

- 1 a list of the amount of each ingredient per ton (or other unit) of the optimal mixture, together with its actual cost and the range of costs within which the solution would show the same ingredients to be in the optimal mixture;
- 2 a list of the ingredients which are not included in the optimal mixture, each with its actual cost and the (lower) cost below which it would be economical to include it;
- 3 a list of the amount of each constituent included in the optimal mixture;
- 4 a list of the constituents or ingredients which are at their upper or lower bounds as given by the min-max constraints.

For each of these, the program outputs the savings which would be obtained or the extra costs which would be incurred if the constraints were to be respectively relaxed or tightened and the limits within which these changes could be effected without altering the structure of the solution.

The cost of the optimal mixture is also printed.

The limitations on the size of problem that can be accepted and an indication of computing time are given in Appendix 1. Suggested designs for input data sheets are given in Appendix 2.



## Chapter I

# INPUT

Data may be input to the program using punched cards or eight-track paper tape.

The input data is punched as a set of records, as follows.

### PAPER TAPE

On paper tape, a record may contain up to 80 characters and is terminated by a newline character. Extra newline characters may be inserted between records to space out the lines of the printed version of the input data produced by a flexowriter. A facility of the program allows the input data to be optionally printed as part of the output and extra newline characters produce blank lines in such a printout.

Blank tape is ignored by the program and it is recommended that blocks of data be separated by lengths of blank tape to facilitate editing.

### PUNCHED CARDS

On punched cards, a record consists of the contents of one 80-column punched card. Blank cards may be inserted between records; these have the same effect on the optional data printout as extra newline characters on paper tape. There is no horizontal tabulation character (*tab*) on punched cards; in general, a tab can be represented by one or several spaces (blank columns) but in the special case mentioned below (see 'Record Format' page 6) only one space is permitted.

### GROUP HEADINGS

The input data is divided into eleven groups, each of which consists of several records. The first record in each group contains the heading of the group. Only the first four characters of a heading are recognized, the rest are ignored; thus comments may be included for printing in the heading line. A heading must consist of at least four characters.

The first character of a heading must be the first in the record, i.e. it must be the first non-blank character after the newline character(s) terminating the previous record on paper tape, or it must occupy column 1 on a punched card. A heading must not be preceded by a tab character or a space character.

Some groups contain sub-headings, which are subject to the same rules as headings.

## RECORD FORMAT

Records may be divided into two classes: those which contain headings, sub-headings or the title of the set of input data, and those which do not.

The first alphanumeric character in records of the latter class is always a number. This number may optionally be preceded by spaces or a tab and may be separated from any succeeding alphanumeric characters by spaces or tabs, with the following restriction:

Where the first number in such a record is followed by an alphabetic name, e.g. 27 WHEAT, the two items may not be separated by more than one space or one tab (one blank column on cards).

This restriction applies to all the 'Name' groups (see below).

Where a record contains only numeric characters, these may be separated by any number of spaces or tabs within the limit of 80 characters. All numbers are punched in free format.

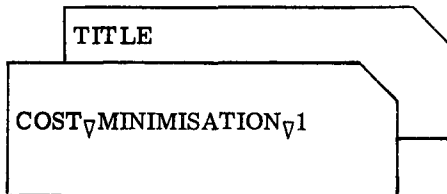
## GROUPS

The eleven data groups are described below. It is recommended that they be input in the order shown. In each case, an example of the group is given showing the format of the group as a printed item in which each line is equivalent to one record. The data must be punched so as to produce the layout indicated, e.g. the first group example would be punched as follows:

### On paper tape:

```
TITLE  
COST MINIMISATION 1
```

### On punched cards:



### 1 Title

Heading: TITLE

followed in the next record by the name for the set of input data.

This name may contain up to 80 characters and is actually printed at the head of the output.

#### Example

TITLE

COST MINIMISATION 1

### 2 Parameters

Heading: PARAMETERS

followed in the next record by nine numbers specifying the values of the following quantities:

- 1 number of constituents
- 2 number of ingredients
- 3 number of min-max constraints
- 4 output parameter

- 5 number of products
- 6 number of exclusion sets
- 7 number of cost rows
- 8 scale factor for ingredients (F)
- 9 scale factor for cost of mixture (G)

These nine numbers occupy one record.

**Min-max Constraints (Parameter 3)**

The sets of min-max constraints defining each product may contain different numbers of constraints. The value given to the appropriate parameter (3 above) is the maximum number of min-max constraints in any one product-defining set. An *equality* constraint (see page 11) is equivalent to two min-max constraints.

**Scaling for Ingredients (Parameter 8)**

The user specifies constituent amounts with respect to a basic unit of each ingredient, say 1 lb and will probably require the optimal mixture to be given in terms of a larger unit, say 1 ton. The ingredients scale factor (F) gives the ratio between these two units; thus, for a basic unit of 1 lb and an optimal mixture unit of 1 ton, F is set at 2240.

**Scaling for Costs (Parameter 9)**

The last of the nine parameters (G) is a scale factor for the cost of the mixture and is printed at the head of the results. If G is set to 1, then this cost will be printed in the same units as those used for the ingredient prices (e.g. sh/lb); if the cost of the mixture is to be printed in different units, then G must be set equal to the appropriate scale factor. Thus, if the ingredient costs are specified in shillings per lb but the cost of the mixture is required in £s per ton, then G should be set to  $2240/20 = 112$ .

**Output Parameter**

The value of the output parameter indicates to the program whether or not the user requires one or both of the optional printouts.

The optional printouts are:

- 1 a printout of the input data
- 2 an iteration print. This gives information about the program at various stages of the computation and it will normally be of limited interest to the user (see the section on 'Output During Optimization', Chapter 2, page 17).

The output parameter is set to one of the values: 0, 1, 2 or 3, the results being as follows:

<i>Output Parameter Value</i>	<i>Output</i>
0	both optional printouts are suppressed
1	the input data printout only
2	both optional printouts
3	the iteration print only

**Name Groups**

The next four groups specify the names of the products, cost rows, ingredients and constituents respectively. In all these groups the names must appear in the order of their indices which must be consecutive. Indices must not be omitted.

**3 Product Names**

Heading: PRODUCT NAMES

followed in successive records by the product names. Each product name is preceded by the index of the product. Product names may have up to eight characters and are listed in the order of their indices.

*Example*

PRODUCT NAMES

1 MIX A  
2 MIX B  
3 MIX C  
. .  
. .  
. .  
7 MIX G

**4 Cost Row Names**

Heading: COST ROW NAMES

followed in successive records by the name of each cost row. (The cost row name may be the name of the factory producing at the costs listed.) Each cost row name is preceded by the index of the cost row. Cost row names may have up to eight characters and are listed in the order of their indices.

*Example*

COST ROW NAMES

1 MILL X  
2 MILL Y  
3 MILL Z

**5 Ingredient Names**

Heading: INGREDIENT NAMES

followed in successive records by the names of the ingredients. Each ingredient name is preceded by its index which is the number of the analysis matrix column (see 'Analysis Matrix' page 9) allocated to that ingredient. Ingredient names may contain up to 16 characters and are listed in the order of their indices.

*Example*

INGREDIENT NAMES

1 LUCERNE MEAL  
2 BARLEY  
3 MAIZE  
. .  
. .  
. .  
16 DRIED MILK

**6 Constituent Names**

Heading: CONSTITUENT NAMES

followed on successive lines by the names of the constituents. Each constituent name is preceded by its index which is the number of the analysis matrix row (see 'Analysis Matrix' below) allocated to that constituent. Constituent names may contain up to 16 characters and are listed in the order of their indices.

*Example*

CONSTITUENT NAMES

- 1 DRY MATTER
- 2 PROTEIN
- 3 OIL
- . .
- . .
- . .
- 10 FISH MEAL

**7 Analysis Matrix**

Before punching, the analysis matrix is constructed as follows:

A column is allocated to each ingredient, the column number in each case being the index of the ingredient. Similarly, a row is allocated to each constituent. Each element of the matrix, located by the intersection of a row and column, consists of the amount of the corresponding constituent contained in one basic unit of the corresponding ingredient.

**Constituent Scaling**

The amounts of any one constituent which form a row of elements in the analysis matrix may be specified in any units, with the following condition: for reasons concerning the accuracy of the computation, *the actual numerical values forming the elements of the matrix should be neither very large nor very small compared with 1.*

For example, if the basic unit of ingredient is 1 lb and an ingredient contains 0.003 grammes per lb of vitamin C, then an appropriate unit for specifying this amount would be milligrammes per lb. The value of the corresponding analysis matrix element would then be 3.

It is important to remember that the units used for specifying constituent amounts in the analysis matrix are also used for specifying min-max constraints on the constituents.

The scaling of constituent amounts is unaffected by the scaling parameters F and G.

**Punching**

The analysis matrix is punched column by column, starting with column 1 and continuing in order. Each column is punched as a series of element values, each preceded by the appropriate constituent index. Elements having the value zero do not have to be punched. The order of the elements in a column is immaterial.

**Organization of the Group**

The group has two subheadings - COLUMN and ROWS - which precede the list of elements for each column. These subheadings are subject to the same rules as headings.

- Group heading : ANALYSIS MATRIX followed by
- subheading : COLUMN followed by
- 1 the number of the first column, followed by
- subheading : ROWS

followed, in successive records, by the row numbers. Each row number is followed, in the same record, by the numerical value of the element for the row.

The two subheadings are repeated at the beginning of each set of column elements, separated by the appropriate column number.

*Example*

**ANALYSIS MATRIX**

**COLUMN**

1

**ROWS**

1 5.4

2 1.01

3 0.88

. .

. .

. .

**COLUMN**

2

**ROWS**

1 6.7

5 7.8

. .

. .

. .

**COLUMN**

13

**ROWS**

1 9.4

6 4

2 6.2

. .

. .

. .

**8 Product Specification**

Each product is specified by a series of min-max constraints on some of the constituents and ingredients. Each min-max constraint is input as a set of 4 numbers, as follows:

- 1 the number 1 or 2; 1 indicates that the constraint refers to an ingredient, 2 that the constraint refers to a constituent,
- 2 the index of the ingredient or constituent (i.e. the column number of an ingredient or the row number of a constituent in the analysis matrix),
- 3 the signed integer number -1 or +1; -1 indicates a minimum constraint, +1 a maximum constraint,
- 4 the actual value of the constraint bound.



**DETAILS**

2	7	+1	1.3
.	.	.	.
.	.	.	.
.	.	.	.

**MINMAX**

6  
.  
.  
.

**9 Exclusion Sets**

Each exclusion set is preceded by two subheadings - EXSET and EXCLUSIONS. There is no main heading for this group.

subheading : EXSET followed by

1 the index of the next exclusion set, followed by

subheading : EXCLUSIONS

followed, in successive records, by the indices of the excluded ingredients in any order.

*Example*

**EXSET**

1

**EXCLUSIONS**

3

9

5

**EXSET**

2

**EXCLUSIONS**

.  
. .  
.

**10 Cost Rows**

This group has two subheadings - SET INDEX and KOSTS - which head lists of ingredient costs. The group heading is PRICE TABLE; this is used to differentiate the group from the previous COST ROW NAMES group, as is the spelling of the subheading KOSTS. (The program recognizes only the first four characters of a heading or sub-heading.)

group heading : PRICE TABLE followed by

subheading : SET INDEX followed by

1 the index of the first cost row, followed by

subheading : KOSTS

followed, in successive records, by the ingredient index numbers. Each ingredient index is followed by the appropriate cost value in the same record.

Each cost row is preceded by the two subheadings separated by the appropriate cost row index.

*Example*

PRICE TABLE

SET INDEX

1

KOSTS

1 30

2 31

3 28

. .

. .

. .

SET INDEX

2

KOSTS

1 29

. .

. .

. .

SET INDEX

4

. .

. .

. .

**11. Triplets**

Each problem to be computed is defined by a set of three numbers called a triplet, as follows:

- 1 the index of the product
- 2 the index of the exclusion set
- 3 the index of the cost row.

Heading: TRIPLETS

followed by the triplets defining the problems. Each triplet occupies one record.

*Example*

TRIPLETS

1 1 1

2 2 1

2 2 2

**End of Data**

The input data must be terminated by the heading ENTER. This initiates the computation when recognized by the program, provided no errors have been found in the data.



## Chapter 2

# OUTPUT

The program produces output at three stages during a run:

during input (including an optional data print-out),  
during optimization (optional) and  
after optimization.

### OUTPUT DURING INPUT

As an aid in checking the input data, various items are printed while the data is being read in.

The title of the set of input data is printed at the head of the output; items are then output in the order in which the corresponding groups are input, as described below.

#### Summary List

The summary list gives the number of ingredients, constituents, products, cost rows, constraints, exclusion sets, relations and variables which are to be input. These numbers are copied or computed from the parameters.

Variables and relations are items which are generated and used by the program. The number of relations is equal to

the number of constituents + the number of constraints + 3

The number of variables is equal to

the number of ingredients + the number of constituents + 1

#### Names

After the product names have been read in, the program prints out:

P PRODUCT NAMES IN

where P is the number of products.

Similar printouts are made for cost-row, ingredient and constituent names.

#### Check Sums

##### *Analysis Matrix*

Under the heading

CHECKSUMS FOR ANALYSIS MATRIX

the program prints the following information for each ingredient column in the analysis matrix:

- index of the ingredient,
- number of constituents,
- the sum of all the elements in the column.

#### **Product Specifications**

Under the heading

##### **CONSTRAINT SUMS**

the program prints the following information for each set of constraints defining a product:

- index of the product,
- number of ingredients and number of constituents included in the specification of the product,
- sum of the indices of the ingredients and constituents included in the specification of the product,
- number of minimum constraints (preceded by a minus-sign),
- number of maximum constraints,
- sum of all constraint bounds.

#### **Exclusion Sets**

For each set of exclusions, the program prints the following information:

- number of exclusions,
- sum of the indices of the ingredients excluded.

#### **Cost Rows**

Under the heading

##### **COST ROW SUMS**

the program prints, for each cost row, the sum of the costs in the row preceded by the index of the row.

#### **Triplets**

The program prints the number of product/exclusion set/cost row triplets defining the problems to be solved.

When the last heading (ENTER) in the input data is read, the program prints **START COMPUTATION**.

#### **Error Checking**

The various check sums described above are output so that they may be compared with corresponding sums produced 'by hand'. Any failure to read an item will be indicated by a disparity between the two sets of sums.

In addition, the program performs a certain amount of error checking automatically. When the program detects an error, it halts and outputs a message on the console typewriter, of the form: **HALTED n**, and a corresponding message on the line printer, as in the following table:

<i>n</i>	<i>Message</i>
49	ERROR IN HEADING
50	HEADING MISSING
51	ERROR IN DATA
52	ERROR

The operator should then mark the input tape or card deck at the point where input has stopped and return the data to the originator for correction.

The most common errors are: the omission or mis-spelling of headings, the inclusion of tabs or spaces before headings and the omission of indices.

#### Optional Input Data Printout

If the output parameter (see Chapter 1, page 7) is set to 1 or 2, the program prints out the input data as it is read in. Thus, the user is able to carry out a complete check. The input data is printed interspersed with the summary list and check sums which are always output at this stage.

### OUTPUT DURING OPTIMIZATION

#### Iteration Print (optional)

The Least Cost Mix program uses a method for deriving solutions to linear programming problems called the *Simplex* method. The Simplex method produces solutions (where feasible solutions exist) via a number of similar steps or iterations. If the output parameter (see Chapter 1, page 7) is set to 2 or 3, then the program outputs certain information at each iteration; this output is called the iteration print. A typical line of the iteration print might be:

```
8 COL 21 LAB 21 ZREAL 0.000 ZIMAG -0.200E 1 WMIN -0.104E3
ROW 8 LAB 1036 PIVOT 14.0000
```

The iteration print is mainly intended for the linear programming specialist and an explanation of all its details would require a greater knowledge of linear programming than is expected from the reader of this manual. However, the iteration print can provide the following useful information to the user.

- 1 Very large or very small values of the PIVOT indicate that better scaling may result in improved accuracy.
- 2 At a certain point in the program, the *imaginary objective function* ZIMAG (sometimes called the *infeasibility*) and WMIN both become zero (printed as 0.000E 0); this is the point at which a feasible solution is first obtained. From this point onwards, ZREAL, the negative value of the unscaled cost of the mixture, steadily becomes less negative while ZIMAG remains zero. Any deviation from this sequence may be an indication of contradictory constraints, and the data should be carefully checked in such cases.
- 3 The regular output of the iteration print indicates to the operator that the computation is proceeding normally.

### OUTPUT AFTER OPTIMIZATION

#### Results

When the solution of a problem has been obtained, the program prints out the cost of the mixture. The cost is given as the cost of a required quantity (e.g. 1 cwt) of the mixture as determined by the value of the costs scaling factor G (see Chapter 1, page 7) and the units used to express the ingredient costs.

The cost is printed both for the first feasible solution and for the final, optimal solution. In addition, the percentage improvement of the optimal cost over the feasible cost is printed. This percentage gives some indication of the tightness of the constraints.

The detailed results are then printed in the form of four tables with the headings: Solution, Alternatives, Analysis and Active Constraints.

#### Solution

This table lists the following information for each ingredient included in the optimal mixture:

- the index of the ingredient,
- the name of the ingredient,

the amount contained in the specified amount of the mixture,  
the cost of the ingredient as specified in the appropriate cost row,  
an upper and a lower cost limit indicating to what extent the cost of the ingredient could be altered without affecting the structure of the optimal mixture.  
(Where there is no such limit, the program prints a number of the order of plus or minus one million.)

As a check, the quantities of all the ingredients included in the mixture are added up; this sum should be equal to the quantity of mixture specified in terms of the basic units. Thus, if the required quantity of mixture is 1 ton and the basic unit is 1 lb., then the sum should equal 2240. In other words, the sum should be numerically equal to the scale factor F.

As a further check, the list of ingredients is preceded by the value of a check variable called the UNIT VARIABLE. This variable should always have the value 1.0000. Small deviations in the value of the unit variable from 1.0000 and of the sum from the value of F indicate inaccuracies in the computation; completely different values may be caused by contradictory constraints.

#### **Alternatives**

This table lists the following information for all the ingredients which are *not* included in the optimal mixture:

- the index of the ingredient,
- the name of the ingredient,
- the cost as specified in the appropriate cost row.

In addition, the program prints a lower limit for each ingredient which is the cost below which it would be economical to include the ingredient in the optimal mixture.

Ingredients which are not included in the optimal mixture because they are specified in the exclusion set for the problem are indicated by the letters EXC.

#### **Analysis**

This table lists the following information for each constituent contained in the optimal mixture:

- the index of the constituent,
- the name of the constituent,
- the amount of the constituent present in the optimal mixture.

This amount is given, under the heading VALUE, in the same units used in the analysis matrix and the constraints specification.

#### **Active Constraints**

This table lists those ingredients or constituents which are included in the optimal mixture in amounts which exactly satisfy the corresponding constraints. In other words, if an ingredient or constituent is listed in the table, then the amount of the material present in the optimal mixture is equal to an upper or lower bound of a constraint on the material.

In such cases it is possible to reduce the total cost by relaxing the constraints; the cost will be increased if the constraints are tightened. The table lists the following information for each material meeting the above condition:

- the index and name of the ingredient or constituent,
- the amount of the material contained in the optimal mixture (under the heading VALUE),
- an indication as to the type of bound which has been reached - MAX or MIN,
- limits between which the constraint could be tightened or relaxed without altering the structure of the optimal mixture,
- the amount saved or added to the cost as a result of relaxing or tightening the constraint to these limits.

## Chapter 3

# OPERATING INSTRUCTIONS

The information in this chapter is intended for the user's computer operators. The Least Cost Mix program is held in binary form on magnetic tape.

To run the Least Cost Mix program, the following operations must be carried out:

- 1 Load the program tape, and three scratch tapes with write-permit rings.
- 2 Load the input device (card reader or paper tape reader) with the input data card deck or paper tape.
- 3 Type the following messages on the console typewriter:

FOR PAPER TAPE:      FI #X3MM #TAPE

GO #X3MM

FOR PUNCHED CARDS:  FI #X3MM #TAPE

ON #X3MM 1

GO #X3MM

Handwritten marks at the top right corner.

Small handwritten mark on the right side.

Small handwritten mark on the right side.

Small handwritten mark at the bottom right corner.

## Chapter 4

### AN EXAMPLE PROGRAM RUN

The set of data input to the program for the run described in this chapter is called TEST DATA. The analysis matrix, cost rows, product specifications (min-max constraints) and exclusion sets are shown in Figures 1 to 4, in their pre-punching format. There are 16 ingredients, 10 constituents, 2 cost rows, 2 product specifications and 2 exclusion sets.

The format of the input data, as punched, is shown by sections of the printout of the input data, pages 25 and 26. The output parameter is given the value 1 so that the iteration print is suppressed.

The output during input is shown on page 24.

The solutions are shown for two problems:

Problem 1: MIX A, COST X, exclusion set 1

Problem 2: MIX B, COST X, exclusion set 2

on pages 27 and 28.

	Dry Matter	Crude Protein	Oil	Sol. Carbohydrate	Fibre	Ash	Starch Equiv.	Protein Equiv.	Lime	Phosphoric Acid
	1	2	3	4	5	6	7	8	9	10
1 Lucerne Meal	91	22.3	2.9	36.4	18	11.4	50	13.5	2.73	.78
2 Barley	85.1	10	1.5	66.5	4.5	2.6	71	7.3	.07	.84
3 Maize	87	9.9	4.4	69.2	2.2	1.3	78	76	.02	.82
4 Oats	86.7	10.3	4.8	58.2	10.3	3.1	60	7.6	.14	.81
5 Wheat	86.6	12.1	1.9	69	1.9	1.7	72	9.6	.05	.86
6 Cotton Cake Und.	90.2	41.2	8	26.5	7.8	6.7	68	34.6		
7 Groundnut Cake D	89.7	46.8	7.5	23.2	6.4	5.8	73	41.3	.2	1.3
8 Linseed Cake	88.8	29.5	6	35.5	9.1	5.2	74	24.6	.51	1.7
9 Palm Kernel Cake	89	19.2	.8	46.5	13.4	3.9	73	17	.3	1.1
10 Blood Meal	86	81	.8	1.5		2.7	63	68.1	.05	.22
11 Fish Meal	87	61	3.5	1.5		21	59	53	10	09
12 Bone Meal	90.3	50.3	15	1		24	68	34.2	10.5	9.3
13 Sugar Beet Pulp	90	8.9	.6	59.1	18.3	3.1	61	5.1	1.2	.18
14 Dried Yeast	93.7	41.5	1	41.1	.2	9.6	68	32.5	.3	5.5
15 Soya Bean Meal	85.5	43.1	5.6	26.3	5.1	5.4	69	36.9	.3	2
16 Dried Milk	95.8	25.5	26.5	37.4		6.4				

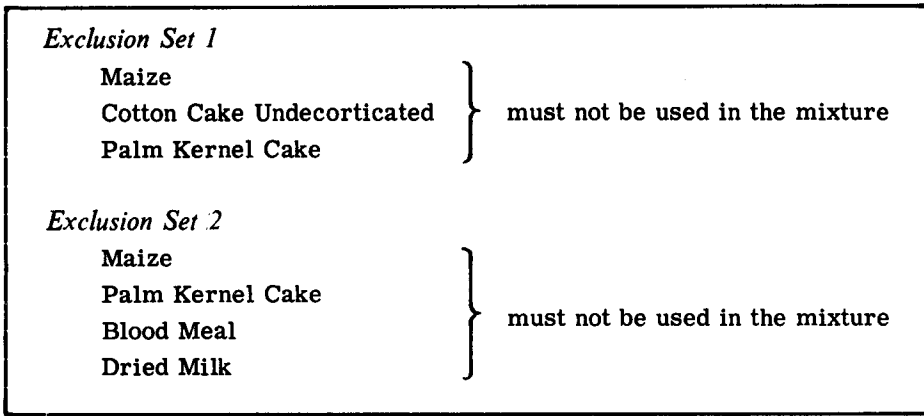
Figure 1 Analysis Matrix

	<i>Ingredients</i>	<i>1: Cost X</i>	<i>2: Cost Y</i>
1	Lucerne Meal	33.5	30
2	Barley	30	31
3	Maize	30.625	31
4	Oats	29.375	28
5	Wheat	29.5	29.5
6	Cotton Cake Und.	38.5	38
7	Groundnut Cake D	46.5	45
8	Linseed Cake	42.75	44
9	Palm Kernel Cake	30	30
10	Blood Meal	64	64
11	Fish Meal	68	69
12	Bone Meal	40	40
13	Sugar Beet Pulp	22.5	22
14	Dried Yeast	78	79
15	Soya Bean Meal	45	45.5
16	Dried Meal	135	137

Figure 2 Cost Rows

<i>Mix A</i>			
In one unit of MIX A, the amount of:			
phosphoric acid	must be between	0.75 and 2 units	
crude protein	must be between	17.5 and 22.5 units	
oil	must be between	2 and 6.0 units	
fibre	must be between	7.5 and 10 units	
lime	must not be less than	0.4 units	
wheat	must be between	0.175 and 0.225 units	
oats	must be equal to	0.3 units	
blood meal	must not be greater than	0.05 units	
<i>Mix B</i>			
In one unit of MIX B, the amount of:			
crude protein	must be between	22.5 and 25 units	
oil	must be between	3 and 4 units	
fibre	must be between	4 and 6 units	
lime	must be between	1 and 1.5 units	
fish meal	must be equal to	0.1 units	

Figure 3 Product Specifications



*Figure 4 Exclusion Sets*

LEAST COST MIX PROGRAM

TEST DATA

16 INGREDIENTS  
 10 CONSTITUENTS  
 2 PRODUCTS  
 2 COST ROWS  
 14 CONSTRAINTS  
 2 EXCLUSIONS  
 27 RELATIONS  
 27 VARIABLES

2 PRODUCT NAMES IN

2 COSTNAMES IN

16 INGREDIENT NAMES IN

10 CONSTITUENT NAMES IN

CHECKSUMS FOR ANALYSIS MATRIX

INGREDIENT 1	10	CONSTITUENTS	SUM	249,010000
INGREDIENT 2	10	CONSTITUENTS	SUM	249,410000
INGREDIENT 3	10	CONSTITUENTS	SUM	260,440000
INGREDIENT 4	10	CONSTITUENTS	SUM	241,930000
INGREDIENT 5	10	CONSTITUENTS	SUM	255,710000
INGREDIENT 6	8	CONSTITUENTS	SUM	283,000000
INGREDIENT 7	10	CONSTITUENTS	SUM	295,200000
INGREDIENT 8	10	CONSTITUENTS	SUM	274,910000
INGREDIENT 9	10	CONSTITUENTS	SUM	264,200000
INGREDIENT 10	9	CONSTITUENTS	SUM	303,370000
INGREDIENT 11	9	CONSTITUENTS	SUM	305,000000
INGREDIENT 12	9	CONSTITUENTS	SUM	302,600000
INGREDIENT 13	10	CONSTITUENTS	SUM	247,480000
INGREDIENT 14	10	CONSTITUENTS	SUM	293,700000
INGREDIENT 15	10	CONSTITUENTS	SUM	279,200000
INGREDIENT 16	5	CONSTITUENTS	SUM	191,600000

CONSTRAINT SUMS

PRODUCT	ING	CON	INDEX	MIN	MAX	BOUND
1	5	9	77	-7	7	69,700100
2	2	8	60	-5	5	67,200100

COST ROW SUMS

1 763,2500  
 2 1202,5000

1 3 EXCLUSIONS, TOTAL 18

2 4 EXCLUSIONS, TOTAL 38

2 TRIPLETS IN

TITLE  
TEST DATA

PARAMETERS  
10 16 14 3 2 2 2 2240 1

PRODUCT NAMES  
1 MIX A  
2 MIX B

COST ROW NAMES  
1 COST X  
2 COST Y

INGREDIENT NAMES  
1 LUCERNE MEAL....  
2 BARLEY.....  
3 MAIZE.....  
4 OATS.....  
5 WHEAT.....  
6 COTTON CAKE UND.  
7 GROUNDNUT CAKE D  
8 LINSEED CAKE....  
9 PALM KERNEL CAKE  
10 BLOOD MEAL.....  
11 FISH MEAL.....  
12 BONE MEAL.....  
13 SUGAR BEET PULP.  
14 DRIED YEAST.....  
15 SOYA BEAN MEAL..  
16 DRIED MILK.....

CONSTITUENT NAMES  
1 DRY MATTER.....  
2 CRUDE PROTEIN...  
3 OIL.....  
4 SOL CARBOHYDRATE  
5 FIBRE.....  
6 ASH.....  
7 STARCH EQUIV....  
8 PROTEIN EQUIV...  
9 LIME.....  
10 PHOSPHORIC ACID.

ANALYSIS MATRIX

COLUMN 1  
ROWS  
1 91  
2 22.3  
3 2.9  
4 36.4  
5 18  
6 11.4  
7 50  
8 13.5  
9 2.73  
10 .78

COLUMN 2  
ROWS  
1 85.1  
2 10  
3 1.5  
4 66.5  
5 4.5  
6 2.6  
7 71  
8 7.3  
9 .07  
10 .84

COLUMN 3  
ROWS  
1 87  
2 9.9  
3 4.4  
4 69.2  
5 2.2  
6 1.3  
7 78  
8 7.6  
9 .02  
10 .82

COLUMN 4  
ROWS  
1 86.7  
2 10.3  
3 4.8  
4 58.2  
5 10.3  
6 3.1  
7 60  
8 7.6  
9 .14  
10 .81

SPECIFICATION

MINMAX

DETAILS

2	10	-1	0.75
2	10	+1	2
2	2	-1	17.5
2	2	+1	22.5
2	3	-1	2
2	3	+1	6.0
2	5	-1	7.5
2	5	+1	10
2	9	-1	0.4
1	5	-1	0.175
1	5	+1	0.225
1	4	-1	0.3
1	4	+1	0.3001
1	10	+1	0.05
1	11	+1	0.1001

PRICE TABLE

SET INDEX

KOSTS

1	33.5
2	30
3	30.625
4	29.375
5	29.5
6	38.5
7	46.5
8	42.75
9	30
10	64
11	68
12	40
13	22.5
14	78
15	45
16	135

EXSET

2

EXCLUSIONS

3  
9  
10  
16

TRIPLETS

1	1	1
2	2	1

ENTER

PROBLEM 1  
 PRODUCT 1 MIX A  
 EXCLUSIONS 1  
 COST 1 COST X

COST OF MIXTURE  
 FEASIBLE 46,077 OPTIMAL 29,124 IMPROVEMENT 36.79 PERCENT

SOLUTION

	INGREDIENT	QUANTITY	ACTUAL COST	COST LIMITS	
				UPPER	LOWER
	UNIT VARIABLE...	1,0000			
4	OATS.....	671.96	29,375	1000029,375	22,868
5	WHEAT.....	391.98	29,500	1000029,500	23,870
10	BLOOD MEAL.....	23.55	64,000	70,464	32,976
12	BONE MEAL.....	371.28	40,000	46,330	-4901,470
13	SUGAR BEET PULP.	781.21	22,500	27,204	7,632
	TOTAL	2239,99			

ALTERNATIVES

	INGREDIENT	ACTUAL COST	LOWER LIMIT	
1	LUCERNE MEAL....	33,500	29,801	
2	BARLEY.....	30,000	22,674	
3	MAIZE.....	30,625	22,631	EXC
6	COTTON CAKE UND.	38,500	41,229	EXC
7	GROUNDNUT CAKE D	46,500	43,550	
8	LINSEED CAKE....	42,750	33,308	
9	PALM KERNEL CAKE	30,000	27,792	EXC
11	FISH MEAL.....	68,000	46,371	
14	DRIED YEAST.....	78,000	37,575	
15	SOYA BEAN MEAL..	45,000	40,932	
16	DRIED MILK.....	135,000	32,187	

PROBLEM 1  
 PRODUCT 1 MIX A  
 EXCLUSIONS 1  
 COST 1 COST X

ANALYSIS

	CONSTITUENT	VALUE
1	DRY MATTER.....	88,423
2	CRUDE PROTEIN...	17,500
3	OIL.....	4,476
4	SOL CARBOHYDRATE	50,327
5	FIBRE.....	9,805
6	ASH.....	6,315
7	STARCH EQUIV....	63,806
8	PROTEIN EQUIV...	12,123
9	LIME.....	2,210
10	PHOSPHORIC ACID.	2,000

	ACTIVE CONSTRAINTS	VALUE	CHANGE TO	SAVE	CHANGE TO	EXTRA COST
10	PHOSPHORIC ACID. MAX	2.00000	2.16657	0.11590	1.77189	0.15872
2	CRUDE PROTEIN... MIN	17.50000	16.74388	0.43554	20.33978	1.63576
5	WHEAT..... MIN	0.17500	0.16198	0.07330	0.22501	0.28155
4	OATS..... MIN	0.30000	0.27250	0.17892	0.30012	0.00075

SOLUTION COMPLETED

PROBLEM 2  
 PRODUCT 2 MIX B  
 EXCLUSIONS 2  
 COST 1 COST X

COST OF MIXTURE  
 FEASIBLE 34,641 OPTIMAL 33,829 IMPROVEMENT 2,34 PERCENT

SOLUTION

INGREDIENT	QUANTITY	ACTUAL COST	COST LIMITS	
			UPPER	LOWER
UNIT VARIABLE,..	1,0000			
5 WHEAT.....	1113,24	29,500	31,928	24,371
6 COTTON CAKE UND.	406,88	38,500	43,383	34,730
11 FISH MEAL.....	224,00	68,000	1000068,000	44,216
12 BONE MEAL.....	50,46	40,000	44,715	-13,908
13 SUGAR BEET PULP.	445,42	22,500	28,367	-9,997
TOTAL	2240,00			

ALTERNATIVES

INGREDIENT	ACTUAL COST	LOWER LIMIT	
1 LUCERNE MEAL....	33,500	26,966	
2 BARLEY.....	30,000	27,863	
3 MAIZE.....	30,625	28,392	EXC
4 OATS.....	29,375	26,065	
7 GROUNDNUT CAKE D	46,500	40,965	
8 LINSEED CAKE....	42,750	33,484	
9 PALM KERNEL CAKE	30,000	28,324	EXC
10 BLOOD MEAL.....	64,000	55,904	EXC
14 DRIED YEAST.....	78,000	40,949	
15 SOYA BEAN MEAL..	45,000	39,960	
16 DRIED MILK.....	135,000	35,152	EXC

PROBLEM 2  
 PRODUCT 2 MIX B  
 EXCLUSIONS 2  
 COST 1 COST X

ANALYSIS

CONSTITUENT	VALUE
1 DRY MATTER.....	88,053
2 CRUDE PROTEIN...	22,500
3 OIL.....	3,205
4 SOL CARBOHYDRATE	51,030
5 FIBRE.....	6,000
6 ASH.....	5,319
7 STARCH EQUIV....	67,696
8 PROTEIN EQUIV...	18,140
9 LIME.....	1,500
10 PHOSPHORIC ACID.	1,573

ACTIVE CONSTRAINTS		VALUE	CHANGE TO	SAVE	CHANGE TO	EXTRA COST
2 CRUDE PROTEIN...	MIN	22,50000	21,58493	0,94249	25,00000	0,93570
5 FIBRE.....	MAX	6,00000	9,12919	1,01443	4,00000	0,64837
9 LIME.....	MAX	1,50000	3,12150	0,68483	1,24853	0,10621
11 FISH MEAL.....	MIN	0,10000	0,04186	1,38280	0,10010	0,00238

SOLUTION COMPLETED

OUTPUT COMPLETED

LEAST COST MIX PROGRAM

## Appendix I

### PROBLEM SIZE LIMITATIONS

The maximum numbers of the various data items which the Least Cost Mix program is capable of accepting are as follows:

40 ingredients	20 constituents	10 products
25 constraints	8 exclusion sets	4 cost rows
40 problems (triplets)		

(25 constraints means that any product can be defined by up to 25 constraints; equality constraints are counted as two.)

#### **Core Store Requirements**

The current version of the program requires 11,913 words of core storage.

#### **Computing Time**

The computing time on a 1903 is approximately 7 minutes per problem of the size of the example given in Chapter 4.



## Appendix 2

# DATA SHEETS

Suggested designs for input data sheets are given on the following pages.

### GENERAL PUNCHING INSTRUCTIONS

Punch only data enclosed in heavy lines.

Each line in the sheets represents a record. A record is either a card, or a series of paper tape characters ending in a newline character (producing one line when printed out).

Extra newlines or blank cards can be inserted between lines if required.

On tape, lengths of blank tape should be inserted between groups of data to facilitate corrections and modifications.

Alphabetical headings must start on column 1 when using cards, and they must not be preceded by spaces or tabs when using tape.

Records starting with a number can be preceded by spaces or tabs on tape, or blank columns on cards.

In general, items in a record can be separated by any number of spaces or tabs (or blank columns in cards), but there must be one and only one space or tab or blank column between a number (index) followed by an alphabetical name.

Note that main group headings (such as ANALYSIS MATRIX) are punched only at the head of the first sub-group, except in the case of EXSET.



**1900 Series Least Cost Mix  
program data**

Reference: \_\_\_\_\_

Originator: \_\_\_\_\_

Date: \_\_\_\_\_

<b>TITLE</b>	
--------------	--

Up to 80 Characters

**PARAMETERS**

Number of	Number of			Scale Factors					
	Constituents	Ingredients	Constraints	Output Parameters	Products	Excl. Sets	Cost Rows	F	G
≤ 20		≤ 40	≤ 25	≤ 3	≤ 10	≤ 8	≤ 4		



# 1900 Series Least Cost Mix program data

Reference: \_\_\_\_\_

Originator: \_\_\_\_\_

Date: \_\_\_\_\_

## PRODUCT NAMES

Index	Name
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Indices 1, 2 .....  
up to number  
of product

## COST ROW NAMES

Index	Name
1	
2	
3	
4	

Indices 1, 2 .....  
up to number  
of cost rows

Up to 8 Characters  
per name





**1900 Series Least Cost Mix  
program data**

Reference: \_\_\_\_\_

Originator: \_\_\_\_\_

Date: \_\_\_\_\_

**CONSTITUENT NAMES**

Index	Name
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

Indices 1, 2 ..... up to number of constituents  
Up to 16 characters per name









**1900 Series Least Cost Mix  
program data**

Reference: \_\_\_\_\_

Originator: \_\_\_\_\_

Date: \_\_\_\_\_

**PRICE TABLE**

**SET INDEX**

Cost Row  
Index

**KOSTS**

Index	Cost
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

Indices 1, 2 ..... up to  
number of ingredients.



**1900 Series Least Cost Mix  
program data**

Reference: \_\_\_\_\_

Originator: \_\_\_\_\_

Date: \_\_\_\_\_

**KOSTS** continuation sheet

Index	Cost
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	









