OPTIMISING

STORM-WATER DRAINAGE

NETWORKS

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor of Philosophy by

GODFREY ALAN WALTERS, M.A. (Cantab), C.Eng.

December 1981.

Ph.D. Thesis. G.A. WALTERS, 1981

OPTIMISING STORM WATER DRAINAGE NETWORKS

SUMMARY

This thesis examines ways in which the design of storm water drainage networks can be optimised and proposes, develops and tests some such methods.

The introduction is followed by a résumé of current design practice and an examination of previous work on the drainage optimisation problem. Methods of estimating the construction cost of a drainage network are detailed and functions proposed for modelling these costs.

The optimisation problem may logically be split into two areas, namely optimising fixed plan networks and optimising variable plan networks. The former involves the simultaneous selection of gradients and diameters for a network of pipes fixed in plan. A new Dynamic Programming model is proposed for this, having several advantages over previously published methods.

The main area of innovation is, however, in optimising variable plan networks. The general plan optimisation problem is seen to be far too complex for solution. However, taking the special case of road drainage networks, two possible modes of optimisation are defined. These are, firstly, the positioning of an unknown number of manholes along a drain running between two fixed manholes, and secondly, the positioning of an unknown number of cross-drains along a road carriageway. Both modes include the simultaneous choice of pipe gradients and diameters.

Models for these modes are proposed, with practical computer programs being developed and tested. Both models use a novel form of Dynamic Programming conceived and developed during this research.

The thesis ends with a brief outline of a Dynamic Programming solution to a rather different variable plan problem, followed by suggestions of areas for further study and conclusions of both a specific and a general nature.

Acknowledgement

I would like to thank all those who have helped and encouraged me during the course of this research. In particular I would like to express my gratitude to the following:

Dr. A.B.Templeman, Senior Lecturer in Civil Engineering at the University of Liverpool, for his thoughtful guidance and supervision.

The Highway Engineering Computer Branch of the Department of Transport for financing the project.

The Department of Engineering Science at the University of Exeter for assistance in the production of this thesis.

Kate, Andrew and Sally for being a loving family in trying circumstances.

CONTENTS

		page
Summary		
Acknowledgements		
Notation		
Chapter 1	The objectives	1
Chapter 2	Designing a drainage network	4
Chapter 3	Related research	15
Chapter 4	Costing a drainage network	23
Chapter 5	The fixed plan optimisation model - MANFIX	32
Chapter 6	Variable plan optimisation	92
Chapter 7	The variable manhole position model - MANVAR	102
Chapter 8	The variable cross-drain position model - CROSSVAR	148
Chapter 9	A further variable plan optimisation problem	187
Chapter 10	Conclusions and areas for further study	192
References		198
Appendix A	Cost calculations based on Spons Architects and Builders price book 1977	203
Appendix B	Cost calculations based on Farrar (ref. 1)	206
Appendix C	Fortran coding for program DPO	210
Appendix D	Fortran coding for program ASSEMB	221
Appendix E	Fortran coding for program MOD	225
Appendix F	Fortran coding for program SORT	239
Appendix G	Fortran coding for program MODEX	243

LIST OF FIGURES

Number	Title	Page
2.1	Variation of velocity with flow for partially full pipe	11
4.1	A typical drainage element	27
4.2	Cost functions	29
5.1	Tree of a typical drainage network	35
5.2a	N -stage serial system	43
5.2b	Section along non-branching drainage run	
5.2c	Single stage of a serial drainage system	
5.2d	N -stage serial drainage system	
5.3	Convergence of serial systems	46
5.4a	The basic serial system	48
5.4b	System with diameter constrains : non-serial	
5.4c	System with diameter constrains : serial	
5.4d	System with Rational method	
5.5a	Establishing upper bound on pipe level	54
5.5b	The design of a stage	
5.6	Design of a stage for basic N -stage serial system	57
5.7	Tracing the solution back through basic N-stage serial system	59
5.8	Drop manholes	60
5.9	Establishing quasi-input state costs, ... drop manholes	62
5.10	Establishing quasi-input state costs, ... branched system	65
5.11	Tracing the optimal solution back through a junction	66
5.12	Establishing quasi-input state costs, ... diameter constrained	68
5.13a	Minimum cover design	70
5.13b	Alternative designs for two pipe system	
5.14	Establishing bounds on DP procedure with diameter constraint	72
5.15	Design of a stage with diameter constraint	74
5.16	Tracing solution back : with diameter constraint	75
5.17	The Rational method of design	76
5.18	Subcatchment areas	78
5.19	Proposed MANFIX computer program	82
5.20	Optimal design by DDDP	84
5.21	Optimised design for housing estate at Peterborough (area A)	87
5.22	Optimised design for housing estate at Peterborough (area B)	88
5.23	Optimised design for housing estate at Peterborough : sections	89

Number	Title	Page
6.1	Variable plan modes	94
6.2	Typical highway storm drainage network	98
6.3	Layout of highway drainage networks	100
7.1a	Serial system for skeleton network	108
7.1b	Serial system for non-branching run	
7.1c	Ranges of manhole positions	
7.2a	Stages for modified serial system	112
7.2b	Stages for a non-branching drainage run	
7.2c	Modified serial system for drainage design	
7.3	ISDP applied to variable manhole position problem	115
7.4	Trace back for the variable manhole position problem	116
7.5	Establishing bounds on pipe level	119
7.6	Test network 2	122
7.7	Sensitivity of network cost to optimising parameters: DPO	124
7.8	Implementing the MANVAR model	126
7.9	Flow chart for program ASSEMB	128
7.10	Flow chart for program MOD	131
7.11	Skeleton plan for network 3	135
7.12a	A solution using MOD	136
7.12b	A minimum cover solution	
7.13	Construction cost vs execution time	137
7.14	Sensitivity of network cost to optimising parameters: MOD	139
7.15a	Network costs for combination of optimising parameters	140
7.15b	Sensitivity of network costs to time of entry	
7.16	Design example	142
7.17	Structure of DAPHOP	147
8.1a	Typical network	150
8.1b	Position of single cross-drain	
8.1c	Fibonacci search : network 3	
8.2	Fibonacci search over 88 internal points	153
8.3	Optimal position of a single cross-drain	155
8.4	Polytope search for two cross-drains	158
8.5a	Typical highway network	161
8.5b	Discrete cross-drain positions	
8.5c	Typical stage	
8.5d	Modified serial system	

Number	Title	Page
8.6	Design flows with variable network	165
8.7a	Cross-drain set with branches joining	166
8.7b	Cross-drain sets with a common base	
8.8	Implementing the CROSSVAR model	170
8.9	Flow chart for program SORT	171
8.10	Flow chart for program MODEX - main program	173
8.11	Flow chart for program MODEX - subroutine XDSET	174
8.12a	Network 4	178
8.12b	Network 5	
8.13	Cross-drain positions for network 4 using MODEX	179
8.14	Sensitivity of network cost to cross-drain resolution	184
8.15a	MODEX solution for network 5	185
8.15b	MODEX solution for network 3	
9.1	Connecting sources to single main drain	189
9.2	Flow chart for MULTICON	191

NOTATION

A	catchment area
b	trench width
C	construction cost for a drainage network
Cb	construction cost for a branch of the network
Ce	construction cost for an element of a network
Co	construction cost for the outfall manhole
C1	cost coefficient - pipe supply
C2	cost coefficient - wheeled excavator
C3	cost coefficient - labour
C4	cost coefficient - granular material
D	pipe internal diameter
DA	diameter of pipe entering manhole
Dmin	smallest permissible pipe diameter
Dus	diameter of largest pipe entering a manhole
d	decision in a serial system
$\underline{e}$	direction vector in optimal search procedure
F1	cost factor for excavation
G	correction matrix in optimal search procedure
$\underline{g}$	gradient vector in optimal search procedure
h	drop across drop-manhole
I	rainfall intensity
L	distance between manholes
Lb	length of a branch
Lmax	maximum manhole spacing
Lmin	minimum manhole spacing
N	return period of rainfall
Q	design discharge for pipe
Qf	full flow discharge for pipe
RZ	range of levels for pipe
r	return in a serial system
S	state in a serial system
SP	spacing of possible positions for intermediate manholes
s	pipe slope
smax	maximum pipe slope
smin	minimum pipe slope
t	duration of storm event
tc	time to concentration
te	time of entry
V	velocity of flow in a pipe

Vf	velocity at full flow
Vmax	maximum velocity
Vmin	minimum velocity
w	step length in optimal search procedure
Xd	distance from fixed manhole to downstream intermediate manhole
Xu	distance from fixed manhole to upstream intermediate manhole
x	distance from base cross-drain to cross-drain
$\underline{x}$	position vector in optimal search procedure
Y	depth of cover over pipe crown
Yav	average depth of cover along pipe
Ymax	maximum depth of cover
Ymin	minimum depth of cover
Yu	depth of cover at upper end of pipe
y	trench depth
Z	level of pipe
ZA	level of pipe entering manhole
ZB	level of pipe leaving manhole
Zd	level of pipe at lower end of an element
Zu	level of pipe at upper end of an element
Zus	level of lowest pipe entering a manhole
$\underline{\delta}$	correction vector in optimal search procedure

CHAPTER 1

THE OBJECTIVES

1.1 The Design Problem

1.2 The Research Objectives

CHAPTER 1 The Objectives

1.1 The Design Problem

Large sums of money, in excess of £100m (ref. 1) are spent annually on storm drainage networks in Britain alone. In broad terms storm drainage optimisation aims to ensure that the best value for money is obtained from this investment.

Ideally this requires that cost-benefit analyses be performed for all drainage schemes (see ref. 2) to ensure that the greatest benefit results, but in practice this is seldom done explicitly.

Instead drainage schemes are generally designed to a set of criteria chosen on the basis of experience. Such criteria give an informal balance between cost and public acceptability. It is of interest to note that this form of cost-benefit analysis is implicit in any engineering code of practice or set of design criteria.

The problem facing the designer is thus reduced to that of choosing a drainage scheme to meet all the design criteria whilst satisfying any constraints imposed by local conditions. The wider question of whether the design criteria are optimally suited to his particular problem is generally beyond his terms of reference, although he may occasionally use his "engineering judgment" to modify design criteria locally.

However, even with this reduced design problem, the designer is still left with, in general, an infinite number of possible solutions, all of which meet the design criteria whilst satisfying the constraints. Assuming that all these solutions have the same benefit, the scheme which involves least cost is the best, or optimal, solution.

1.2 The Research Objectives

It is the problem of finding the least cost solution for a drainage design problem, given a set of criteria and constraints, with which the research is concerned. The question of cost is discussed in Chapter 4, and is taken to be the cost of construction of the drainage network expressed in monetary terms. Given sufficient detailed information it could include future maintenance and running costs, but as these are often estimated to be a fixed proportion of the initial capital cost, their explicit inclusion does not seem justified.

Attention is limited to the system of underground pipes and manholes forming the storm-water drainage network. Excluded are all aspects of water quality and treatment and all effects on natural and artificial water courses downstream of the network outfall. Also excluded is any discussion of flow of storm-water overland before entering the pipe network or of detention storage within or outside the network.

Much of the research relates directly to road drainage as becomes apparent in the discussion of optimal plan layout (Chapter 6). However, an attempt has been made to retain generality wherever possible so that results and cònclusions are in many cases relevant to any storm-water or indeed foul sewerage network.

The possibility of optimising drainage design has only arisen since the advent of cheap and readily available electronic computers. Most medium and large design offices have computing facilities available and indeed much analysis of drainage designs is already performed by computer.

The objectives of this research therefore include an investigation of existing methods for storm drainage optimisation, the development of further practical methods for use on a computer, and the implementation and testing of such methods.

The bulk of the research in fact concentrates on optimising the plan layout of particular types of drainage network with practical computer programs being written and tested for these applications.

CHAPTER 2

DESIGNING A DRAINAGE NETWORK

2.1 Principles of Storm Drainage

2.2 Present Practice in Storm Water Drainage Design

2.3 System Constraints

- 2.3.1 Permissible Pipe Depth
- 2.3.2 Permissible Pipe Slope
- 2.3.3 Permissible Flow Velocity
- 2.3.4 Discharge
- 2.3.5 Pipe Level Continuity at Manholes
- 2.3.6 Pipe Diameter Continuity at Manholes
- 2.3.7 Pipe Diameter
- 2.3.8 Distance between Manholes

2.4 Glossary of Drainage Terms

Chapter 2 Designing a Drainage Network

2.1 Principles of Storm drainage

Storm drainage is provided to reduce nuisance and flood damage from incident rainfall. Flood damage may occur on natural catchments due to prolonged heavy rainfall, but man's influence greatly increases the problem. By changing moorland and forest into well drained agricultural land both the percentage of rain that flows off the land (the percentage runoff) and the speed at which this happens increases. Short, severe storms, which on natural catchments would perhaps be partially absorbed with the remaining runoff spread over a long period, may cause flooding of channels and fields on agricultural land.

The problem becomes far more severe in the urban catchment. High proportions of paved and otherwise largely impermeable areas, such as house-roofs, roads, carparks, footpaths, industrial yards, lead to large percentage runoffs occurring shortly after the rainfall. A very short storm, say a 10 minute cloudburst producing a total of 15 mm of rain, which would be insignificant in the countryside could cause severe local flooding in a town.

For this reason extensive storm drainage networks have been and continue to be built throughout urban areas. Traditionally the philosophy has been to remove the incident rainfall from surfaced areas as quickly as possible. Incidentally, however, thought is now being given to ways of temporarily detaining the runoff as near to the source as possible as a means of economising on the storm drainage network downstream. By slowing down the drainage flows the flood peak further down the system is considerably reduced. This allows the use of smaller pipes, or otherwise inadequate existing sewers, and can prevent damage to the natural watercourses into which storm sewers eventually flow.

Three types of urban sewer exist. There is the foul sewer taking sewage from domestic, industrial and commercial premises to a sewage treatment works or straight out to sea or even into an estuary or river.

There is the storm sewer taking only rainfall. This generally drains to the nearest convenient natural watercourse, but, if it originates from a road or industrial premises, it may lead to some form of treatment works or ponded storage before the water is released. The third type of sewer, rarely installed nowadays, is the combined storm and foul sewer, taking both sewage and rain-water. This is generally provided with storm overflows allowing water to flow out of the network into water-courses when excess flows develop due to storms.

This research concentrates largely on storm sewers, although many of the methods are also applicable to foul sewage networks. Both types may be regarded as "tree-like" networks with the base of the tree at the network outfall. Once the flow has entered the network at a branch it must follow one path and cannot diverge from that path. For this reason combined sewers with storm overflows operational cannot be classified in the same way.

Storm water drainage for new roads is an area of special interest in optimising drainage layout, (See Ch. 6). The design principles are however identical to normal urban storm water drainage.

2.2 Present Practice in Storm Water Drainage Design

It is the optimal design of storm-water drainage systems consisting of tree-like networks of pipes between manholes with which this research is concerned. This section examines how such systems are at present designed.

There are four logical stages:-

a) Identifying the correct design parameters.
b) Specifying the plan layout of the network.
c) Designing the gradients and diameters of the pipes.
d) Detailed specification of drain and manhole construction.

(a) and (d) are outside the scope of the present research, which thus concentrates on optimising the plan layout, pipe gradients and diameters.

The choice of correct design parameters is usually established by reference to the relevant national Codes of Practice (refs. 3 & 4) or locally based design guidelines. Such parameters would include a measure of the acceptable risk of flooding (generally given as the average period of occurence, or Return Period, of a storm giving flows equal to or greater than those designed for), the minimum acceptable velocity of flow in a pipe, the minimum acceptable cover over the crown of the pipe, and the maximum allowable distance between manholes.

Such Codes of Practice or guidelines would also cover such standard practices as

(a) keeping drains straight and at constant gradient and diameter between manholes,

(b) having a manhole at every pipe junction (except for gully connections),

(c) establishing flow capacity and flow velocity by assuming that pipes flow just full, (i.e. with no surcharge pressure), and by using an acceptable flow formula (e.g. Colebrook -White equations).

The second stage, that of specifying the plan layout of the network gives the designer considerable freedom of choice. He must use good judgement and experience to select from an infinite number of possible layouts one that is reasonably efficient and economical. If he wishes to do so he may select several networks and compare designs based on each. If he has sufficient information he may indeed estimate the likely construction cost of each and select the cheapest, thus performing a very basic optimisation, but this is rarely done.

With the layout specified and the position of all manholes fixed in plan, the designer must now specify the gradients and diameters of all pipes in the network. For this he needs to know the design flow for each pipe. Most drainage design in the U.K. is performed using either the Rational or the T.R.R.L. method for establishing design flows (see ref. 5). The Rational method is explained further in section 5.11.2. Very briefly, it enables the designer to calculate a flow for a pipe which is dependent on the total catchment area for the pipe, the average percentage runoff, and the time taken for flow to reach the downstream end of the pipe from the most remote part of the catchment.

The designer may now assign a gradient and diameter to each pipe such that its capacity is greater than or equal to its design flow. For an individual pipe there are likely to be several possible solutions. For example a large pipe at a shallow gradient will convey the same flow as a small pipe at a steep gradient. The number of different possible solutions for a network of pipes soon becomes very large indeed. For example with just 10 pipes in the network and with a choice of 3 different diameters for each pipe, 3^{10} or 59049 different possible solutions exist, assuming no design criterion or other constraint is violated.

Standard practice, however, is for the designer to place the pipe as close to the ground surface as is permissible. This could be governed by a minimum cover criterion or by a minimum velocity of flow constraint. The smallest pipe diameter is then chosen that will provide the required flow capacity. This procedure is based on the assumption that the shallowest solution is the cheapest, an erroneous supposition which can lead to designs considerably more expensive than necessary as will be shown in subsequent chapters.

The last part of the design process is the detailed design and specification for the drains and manholes. Although these may influence costs considerably, it is not the author's intention to investigate this part of the design process, except to say that in general the "detailed design" consists of the selection of appropriate standard designs from local or national guidelines.

2.3 System Constraints

The nature of the constraints on the design of a storm drainage network can fundamentally affect the optimisation procedure adopted. It is worth while here considering in some detail each of these possible constraints. They are as follows:

a) Permissible pipe depth

b) Permissible pipe slope

c) Permissible flow velocity

d) Discharge

e) Pipe level continuity at manhole

f) Pipe diameter continuity at manhole

g) Pipe diameter

h) Distance between manholes

2.3.1. Permissible Pipe Depth $Y_{min} \leq Y \leq Y_{max}$

In all designs a minimum depth of cover is required. This varies depending on the use of the land under which the pipes are to be laid. The current code of practice for sewerage (ref. 3) in the U.K. gives values of Ymin = 0.9 m under fields and gardens and 1.2 m under roads.

Sometimes a maximum depth of cover may be specified. Generally, however, the costs of deep excavation, and the extra requirements of stronger pipes, better bedding or concrete surrounds act to limit the depth. Cost functions can always be provided to reflect these practical costs. Hence, in theory, no strict upper limit need be placed on Y, and Ymax can often be omitted as a constraint.

2.3.2 Permissible pipe slope $s_{min} \leq s \leq s_{max}$

These constraints may sometimes be specified. Since flow is unsurcharged gravity flow, $s > 0$, but this is a necessary condition of constraint (d) and so need not be specified separately here.

The constraint $s_{min} \leq s$ is generally used where the designer considers it impracticable to lay pipes at slopes less than smin. For example, if the gradient is too small inaccuracies in laying could cause pipes to slope in the wrong direction with possible silting up at low flow conditions and trapping of air and partial surcharging at full-flow conditions. Similarly $s \leq s_{max}$ is a practical condition associated with pipe laying on steep slopes. Trouble can be caused with flexible jointed pipes on steep ground as these can slide down the slope if there is insufficient friction in the pipe bedding, particularly when the trench is being backfilled.

2.3.3. Permissible flow velocity $V_{min} \leq V \leq V_{max}$

There is generally some form of restriction on the velocity of flow in the pipe. This is usually in the form of a restriction on the velocity of flow (Vf) that would occur in the pipe flowing just full, but sometimes it is on the actual flow velocity (V) in the pipe at the design discharge (Q). Assuming that the pipe is being used reasonably efficiently with $Q/Qf > 0.25$, the full flow velocity will approximate to the design flow velocity as shown in Fig. 2.1.

Restricting the velocity to be above a minimum value is to prevent deposition of solids along the pipe invert, the minimum value generally being taken as 0.7 m/s (ref 3). A maximum flow velocity is to prevent excessive scour on the pipe walls. This can, of course, vary with differing pipe materials, but is often taken to be about 6.0 m/s. Recent experience suggests that this upper limit on velocity is not as important as was once thought (ref 3).

2.3.4. Discharge $Q \leq Qf$

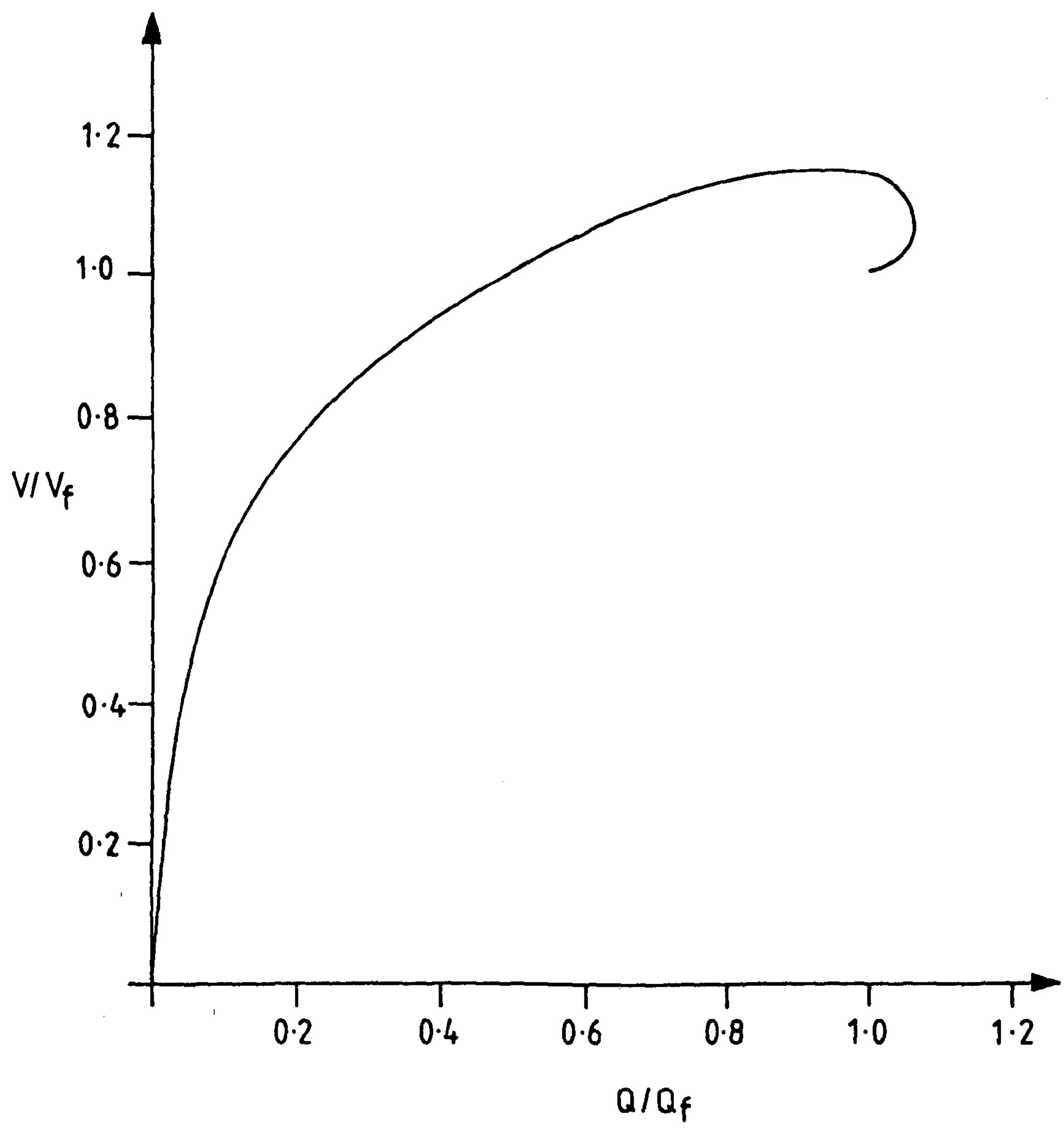
Each pipe in the system must be capable of discharging the design flow at that point without surcharging. The maximum unsurcharged flow down a pipe of given gradient and diameter D occurs when the pipe is flowing with a depth equal to about 0.94 D and is about 1.08 x Qf where Qf is the discharge in the pipe when it flows just full.

For practical purposes however, the maximum discharge is assumed to be Qf. A combination of pipe gradient and diameter must be chosen such that $Qf \geq$ design flow Q.

Q may be explicitly defined at the start of the design as in the case of conventional foul sewerage, or may depend on the pipe network upstream of the point being considered as in the case of storm sewers designed to the Rational (LLoyd- Davies) method (ref. 5), the Transport and Road Research Laboratory (TRRL) method (refs. 5, 6), and most other methods in common use.

2.3.5. Pipe level continuity at manholes $Zu \leq Zus$

The outgoing pipe from a manhole must be able to drain completely all the incoming pipes. Hence the outgoing pipe invert level (Zu) must be no higher than the lowest invert level of the incoming pipes (Zus).



VARIATION OF VELOCITY WITH FLOW
FOR PARTIALLY FULL PIPE

FIGURE 2·1

Moreover, if the outgoing pipe is flowing full and an incoming pipe is of smaller diameter, the incoming pipe will be submerged and hence surcharged unless the soffit of the incoming pipe is at or above the soffit of the outgoing pipe. This leads to the commonly adopted criterion that $Zu \leq Zus$, where Zu and Zus refer to soffit levels. Sometimes, however, designs are done to the alternative criterion $Zu \leq Zus$ where Zu and Zus are invert levels.

Strictly, both criteria are required if pipe diameters are allowed to reduce across a manhole in a downstream sense (see 2.3.6.). A full statement of the constraint then becomes: the downstream pipe soffit level must not exceed any upstream pipe soffit level, and the downstream pipe invert level must not exceed any upstream pipe invert level.

2.3.6. Pipe diameter continuity at manholes $D \geq Dus$

It is common practice to require that the diameter, D, of the outgoing pipe leaving a manhole is at least as big as the diameter, Dus, of any incoming pipe. There is no logical argument for this restriction on the grounds of pipe capacity, as a steep outgoing pipe could have a greater capacity than a larger incoming pipe at a flatter gradient.

It could however be argued that a reduction in pipe diameter at a manhole would increase the likelihood of blockages particularly in a foul or combined system.

2.3.7. Pipe Diameter D must be discrete, available diameter $\geq$ Dmin

Pipes are only available in discrete diameters. The sizes obtainable depend on the pipe material selected and on the pipe manufacturer. Some guidance can be obtained from the British Standard preferred diameters.

For clayware (ref. 7) these are as follows:

75 mm, 100 mm, 150 mm and then in 75 mm increments to 900 mm.

For asbestos-cement (ref. 8) they are in 25 mm increments from

100 to 250 mm, then 300 mm to 1050 mm in increments of 75 mm.

For unreinforced concrete pipes (ref. 9) they are 100 mm, then 150 mm to 600 mm in increments of 75 mm.

For prestressed concrete pipes (ref. 10) they are

450 mm to 1200 mm in 75 mm increments, then to 3000 mm in 150 mm increments.

For pitch-fibre (ref. 11) they are

100 mm to 225 mm in 25 mm increments.

Finally in uPVC (refs. 12 and 13) they are

110 mm, 160 mm, 200 mm, 250 mm, 315 mm 400 mm, 500 mm and 630 mm.

To prevent blockages, there is likely to be a limit to the smallest pipe size permitted for a drain. The current Building Drainage code in the U.K. (ref. 4) restricts drains to be 100 mm or over in diameter. Surface water drains for roads normally have D min = 150 mm.

2.3.8. Distance between Manholes

Where manholes are not required closer together for other reasons they should be spaced at distances not exceeding L max. This is to enable maintenance to be carried out, such as clearing blockages by rodding. L max is usually specified as a figure between 100 and 150 m.

2.4 Glossary of Drainage Terms

Pipe — Either: A pipeline of constant diameter and gradient joining two manholes. Pipes are normally straight, but for road drainage are sometimes curved in plan being at a constant offset from a curving road centreline.
Or: A component of a pipeline.

Manhole — An access chamber provided for maintenance, being for the purposes of this research a real manhole, a catchpit, an outfall or a rodding eye.

Diameter — The internal pipe diameter, (or pipe bore).

Invert — The lowest part of the internal pipe cross-section.

Soffit — The highest part of the internal pipe cross-section.

Crown — The highest part of the external pipe cross-section. In this research crown and soffit levels are considered identical.

French drain — A perforated pipe in a trench backfilled with sand or gravel, thus allowing water to enter the trench and flow into the pipe.

Carrier drain — A pipe that does not accept water anywhere along its length.

Gully	A grated inlet provided at a low point in a paved area, through which flow is led to a drain.
Carriageway drain	A road drain running parallel to the road centre-line and collecting water from the carriageway, either through gullies or as a French Drain.
Cross-drain	A road drain running across a road carriageway. A cross-drain is invariably a carrier drain.
Outfall	The point at which flow leaves the drainage network. For the purposes of this research it may be a manhole belonging to another drainage network or may be a true outfall into an open watercourse, the sea or a treatment works.
Cover	The vertical distance between the crown of a pipe and the ground level.

CHAPTER 3

RELATED RESEARCH

3.1 Introduction

3.2 Optimising a Fixed Plan Drainage Network

- 3.2.1. Linear Programming
- 3.2.2. Non-linear Programming
- 3.2.3. Geometric Programming
- 3.2.4. Dynamic Programming
- 3.2.5. Discrete Differential Dynamic Programming

3.3 Optimising the Plan Layout of a Drainage Network

- 3.3.1. Optimal Layout only
- 3.3.2. Combined Layout, Gradient and Diameter Optimisation.

3.4 Summary

Chapter 3 Related Research

3.1. Introduction

Previous research into optimal design of storm-water drainage networks can be divided conveniently into two categories:

a) optimal choice of diameter and vertical alignment of pipes for a network which is fixed in plan.

b) optimal plan layout of a network.

The former category has received steady attention over the last 15 years and this is summarised in section 3.2.

The latter category has been less well covered with only occasional publications. Work in this area is summarised in section 3.3.

3.2 Optimising a Fixed Plan Drainage Network

To date five techniques of optimisation have been used by various authors in an attempt to find a robust and rigorous method of minimising the cost of a fixed plan drainage network.

These five techniques will be dealt with in turn. They are

a) Linear Programming (LP)

b) Non-linear Programming (NLP)

c) Geometric Programming (GP)

d) Dynamic Programming (DP)

e) Discrete Differential Dynamic Programming (DDDP)

Historically, an interest in optimising drainage networks stems from the work of Haith (ref. 14) who used DP in 1966 to optimise sewer and drainage system vertical alignment. DP itself was originated by Bellman (ref 15) in 1957.

Attempts were made to use standard LP techniques and as sophisticated NLP algorithms became available these were also tried.

Recent work has reverted to better DP approaches, with DDDP being developed as an alternative to conventional DP.

3.2.1. Linear Programming

The theory and practical application of LP has been well developed over many years. Thus there are powerful algorithms available for the solution of any optimisation problem that can be linearised.

It is therefore tempting to linearise the drainage network problem, this being the approach adopted by several researchers.

Naturally problems occur with the non-linear nature of the function to be minimised (the objective function) and with the non-linear nature of some constraints on the function (see 2.3). Less obviously, the availability of pipes only in discrete sizes, too, causes trouble.

Dajani, Gemmell and Morlok (ref. 16) split the non-linear objective function into linear segments and developed sets of linear constraints to replace non-linear constraining functions. Later, Dajani and Hasit (ref. 17) adopted mixed integer equations as constraints to handle discrete pipe diameters.

General studies on optimisation of drainage networks by Yletyinen (ref. 18) and by Dobschutz (ref. 19) led them to adopt LP approaches. Again, more recent work by Iman, McCorquodale and Bewtra (Ref. 20), the principal aim of which was to incorporate flood damage costs into the cost functions, adopted LP as the means of optimisation.

3.2.2. Non-linear Programming

With the rapid advance of computers the feasibility of non-linear programming algorithms to deal with large numbers of variables has been widely investigated. Many large scale optimisation problems can now be tackled by NLP algorithms but there are still difficult areas. Principally these occur with discrete functions, discontinuities and non-linear constraints. As these are all features of drainage network optimisation (see section 5.3), it is clear that NLP cannot at present provide a complete answer.

Lemieux, Zech and Delarue (ref. 21) used Rosen's projected gradient method (ref. 22) to optimise a drainage network assuming that the objective function was convex and linearly constrained and that pipes were available in a continuous range of diameters. The solution was subsequently adapted to include commercial pipe sizes.

Price (Ref. 23) used a quasi-Newton algorithm and developed a method whereby the pipes in a network were adjusted to commerical sizes in a step-by-step approach. This also enabled network dependent design flows to be used. Essentially the method involved optimising the full network using approximate flows and continuous pipe diameters. The furthest upstream pipes were then altered to the nearest commercial diameters, pipe flows were simulated and the network downstream of these pipes optimised again. By repeating the process the optimum solution for the whole network was found. Price found the method insufficiently robust and generally inferior to a DDDP method that he also used. (see 3.2.5.).

3.2.3. Geometric Programming

A somewhat different approach is that of geometric programming (ref. 24) which is described in section 5.3.3.

Wilson (ref. 25) attempted to develop a general purpose tool for optimisation in the building industry using a GP computer model. He used sewer networks as an example to test his model with limited success, the discrete nature of available pipe diameters being a considerable problem. He concluded that the GP technique was "too powerful" for the drainage application and developed instead a tailor-made DP method.

3.2.4. Dynamic Programming

Dynamic programming using discrete values of pipe level is the basis of the present author's current research and is described in detail in Chapter 5.

DP has been applied to fixed plan drainage networks by Haith (ref. 14), Meredith (ref. 26), Merrit and Bogan (ref. 27), Wilson (ref. 25), Walsh and Brown (ref. 28) and recently by Froise and Burges (ref. 29) who incorporate storage elements into the network. Liang (ref. 30) applied DP to a general conduit network.

The above authors have produced models with varying degrees of success and validity. Most conclude that DP is a very effective approach to the drainage optimisation problem due to the serial nature of a drainage network and due to the ability of DP to handle discrete, non-linearly constrained discontinuous functions.

One of the problems with DP is the necessity to define the range of levels within which the pipe must lie at each manhole position. If this range is large and the spacing of discrete lèvels within it small, then a large number of discrete levels must be considered at each manhole. This leads to large computer storage and execution times.

The present author demonstrates that this can easily be avoided (see Chapter 5), but this apparent requirement for large computer resources led to DP being superseded by DDDP (see 3.2.5.).

Two other points were largely ignored by previous authors. Firstly, the fact that design flows are dependent on the network (see 5.5.5) and that if pipe diameters are constrained not to decrease in a downstream direction, this fundamentally affects the DP method (see 5.5.4 and ref. 31).

3.2.5. Discrete Differential Dynamic Programming

DDDP was developed from DP as a means of reducing computer storage and execution time requirements. Basically DDDP is an iterative DP approach and is described in detail in section 5.13.

It was first introduced into the field of water resources by Heidari, Chow, Kokotovic and Meredith of the University of Illinois (ref 32) and later developed at Illinois by Mays and Yen (ref. 33 and 34) for use with drainage design.

Yen, Tang and Mays produced a model incorporating Rational method design (ref. 35) and introduced the risk of flood damage into the cost function (ref. 36).

Mays and Wenzel (ref. 37) restructured the DDDP by redefining the basic stage in the serial system. The concept of isonodal lines (see 3.3 and ref. 38) was introduced, these being lines joining points an equal number of manholes upstream from the outfall. A stage then becomes the design of the network between isonodal lines. This was claimed to be more efficient than previous DDDP approaches.

Nopgomol and Askew (ref. 39) further developed DDDP or, as they called it, Incremental Dynamic Programming, within the general water resources context. They developed Multilevel Incremental Dynamic Programming to enable problems of higher dimensionality to be tackled by DDDP than were previously feasible.

Chow, Maidement and Tauxe (ref 40) compared the execution times for DP and DDDP programs used for drainage network design.

Price (ref. 23) adapted a DDDP method to allow for network dependent design flows.

3.3 Optimising the Plan Layout of a Drainage Network

Little research has been reported on optimising the plan layout of drainage networks. The problem is less well defined than the optimisation of fixed plan networks, there being many modes in which plan optimisation could occur (see section 6.1).

Published papers concentrate on particular aspects of layout optimisation, or on particulr types of network and not on a solution to the general problem.

Research can be split roughly into two categories:

a) finding the optimal layout with pipe diameters and gradients fixed (and therefore suboptimal).

b) optimising layout, pipe sizes and gradients for a special type of network.

3.3.1. Optimal layout only

Liebman (ref. 41) used a simple search procedure which attempted to improve an initially selected trial layout. All pipes had to be the same pre-determined size and were at predetermined slopes. The search consisted of changing one branch of the network at a time, the change being retained if the network cost was decreased. Flows in the system were fixed for each branch.

Barlow (ref. 42) proposed a heuristic method for establishing the route for major trunk sewers and then shortest-path-through-many-points and shortest-spanning-tree techniques to establish the complete layout.

Lowsley (ref. 43) proposed an implicit enumeration procedure based on defining a trunk sewer. Pipe sizes were fixed and the layout optimised for minimum excavation and pipe costs.

3.3.2. Combined layout, gradient and diameter optimisation

In his work on optimisation in the building industry, Wilson (ref. 25) attempted briefly to apply Geometric Programming to a particular drainage layout optimisation problem. He met with little success due to the large numbers of equality constraints, the problems of coincident manhole positions, and the generally large number of variables and constraints in all but the simplest of problems.

Argaman, Shamir and Spivak (ref. 38) proposed an interesting DP model for a particular type of network. The network consisted of a rectangular mesh of pipes which were defined as either local pipes or main pipes. Local pipes originated from a manhole, but had no connection from it. Hence they did not drain the manhole. Main pipes lead from a manhole, thus draining it. The network was a tree, hence only one main pipe could leave a manhole. Both main and local pipes collected water along their lengths.

Isonodal lines were defined as joining nodes an equal number of manholes upstream of the outfall. The layout optimisation consists of determining which pipes were main and which pipes were local and was performed using DP between isonodal lines.

Even with this special network layout, and with a procedure which was not entirely rigorous, the computational resources required for this method made it impractical.

Mays, Wenzel and Liebman (refs. 33, 44) used DP and DDDP with the concept of isonodal lines to develop a two phase screening model for practical optimal drainage layout design. The networks used are similar to the type studied by Argaman. Mays states that the method may not find the true global optimum due to the necessity of adopting a somewhat non-rigorous procedure.

3.4 Conclusion

Certain general conclusions can be drawn from the above summary of published research.

For the fixed plan drainage network problem, only those methods involving the use of DP or DDDP have met with any success, and none of these is entirely satisfactory (see Chapter 5). Methods involving LP, NLP or GP cannot deal with the discrete non-linear and discontinuous nature of the problem. Their use involves either over-simplification of the problem or adoption of a sub-optimal procedure.

For the variable plan problem, no rigorous procedure has been published for even the simplest of cases.

CHAPTER 4

COSTING A DRAINAGE NETWORK

4.1 Introduction

4.2 Cost of Measured Work

4.3 Cost Model

- 4.3.1. Cost of a Network
- 4.3.2. Cost of an Element
- 4.3.3. Cost Functions
- 4.3.4. Alternative Cost Functions

CHAPTER 4

Costing A Drainage Network

4.1 Introduction

A prerequisite of minimum cost design is the ability to cost a design, or at least to compare the relative costs of one design with another.

The cost of a drainage scheme from the point of view of the scheme's promoter would include such items as

(a) acquisition of land or easements,

(b) design and supervision costs,

(c) future likely maintenance and replacement costs

(d) the final contract costs.

The tendered contract price is the contractor's estimate of the cost of the job plus his profit and consists of

(a) cost of measured work

(b) lump sum items such as setting up temporary site buildings, insurance, temporary works and mobilising plant and labour

(c) profit and head office costs.

In optimising the design, it is the cost of the measured work that one attempts to minimise. On small schemes the measured work may well represent less than half the total cost. However, by minimising the cost of measured work, savings may also be made on some other items, such as maintenance and replacement costs and insurance, but these savings will not generally be directly proportional.

It would be unwise to compare two completely different schemes purely on the basis of the cost of measured work. However, the nature of drainage optimisation is that all schemes compared are generally similar with only slight differences in pipe slopes, diameters and manhole positions. Hence a comparison on the basis of the cost of measured work is usually valid.

4.2 Cost of measured work

Traditionally the prices of measured work, as presented in tender documents are given as rates per unit for the various items of construction multiplied by the estimated quantity for those items as given in the Bill of Quantities. The total price for measured work is then the sum of the prices calculated for all items. The actual cost of the measured work is found at the end of the contract by measuring all items as constructed and multiplying by the appropriate rates. This assumes that there is no great difference between the quantities as estimated in the Bill of Quantities and the final measurement.

If there is a significant difference, the contractor may have grounds for a claim for extra payment. For example, if the total length laid of a certain large diameter pipe has been reduced from say 200m to 20m, the contractor could argue that the cost of setting up the pipe-laying operation is not now being met by the rate quoted in his tender, and that had he known that a much smaller length was to be laid, he would have put in a much higher rate.

Of course, the reverse situation could arise, with the contractor making an unexpected windfall from an increase in quantity of a highly priced item, and on balance the two effects will tend to cancel out.

Returning to the design stage and the problem of comparing the costs of different schemes, one should ideally have a costing model that gives an increased rate per unit for low total quantities of that unit.

This however would be very difficult to achieve due to lack of sufficient data and the variations in individual contractors' working methods. Also the quantities involved in drainage tend to be of sufficient size for this effect to be generally negligible.

4.3 Cost Model

4.3.1 Cost of a Network

In building up a useable and realistic cost model for use in the optimising process two basic assumptions are made:

(1) The cost of a scheme = The sum of the independent costs of individual parts of the scheme.

(2) The rates used to calculate costs of individual parts of the scheme are independent of the quantities involved.

It is useful here to define a typical element in a drainage scheme. This can be taken as a length of pipeline between manholes, together with all the associated excavation and backfill, and the manhole immediately upstream of the pipeline. As can be seen from Fig. 5.1, in a network of n pipes where no two pipes have the same upstream manhole, there are n + 1 manholes (including an outfall manhole) and n elements.

Hence the cost of the total network (C) equals the sum of the element costs (Ce) plus the cost of the outfall (Co)

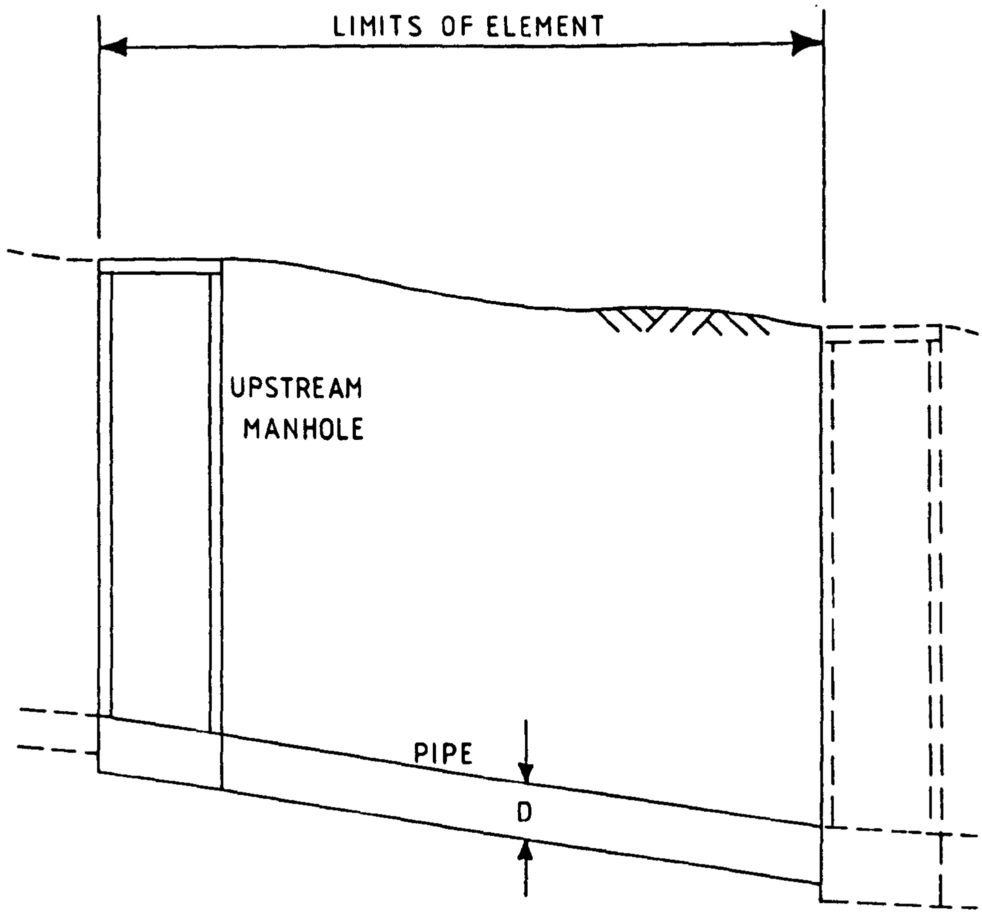
i.e. $$C = \sum_{1}^{n} Ce + Co \qquad \text{........................4.1}$$

4.3.2. Cost of an Element

A typical element is shown in Fig 4.1. Various parameters can be used to define the element for costing purposes. These must include the pipe diameter, pipe type and bedding type, and must also include some measure of the total volume of excavation and the depth of excavation. The upstream manhole diameter and depth are also required, as is some information as to soil type, dewatering requirements, whether a road surface has to be broken up and removed, the degree of reinstatement required, restriction of access to the work, and frequency of other services crossing the trenches.

Farrar (ref. 45) has collected data based on observations of site operations in the UK for laying sewer pipes of up to 600mm. From this he has derived a simple costing procedure, involving most of the above parameters, and hence generally applicable.

A rather less detailed approach can be used based on annually published cost data from the building industry (ref. 46) and a third approach would be to study the prices tendered by contractors for past drainage schemes.



A TYPICAL DRAINAGE ELEMENT

FIGURE 4·1

This last approach is, however, rather unsatisfactory for the following reasons:

(1) The breakdown of prices is not very detailed.

(2) The prices quoted do not necessarily reflect the actual costs to the contractor.

4.3.3. Cost Functions

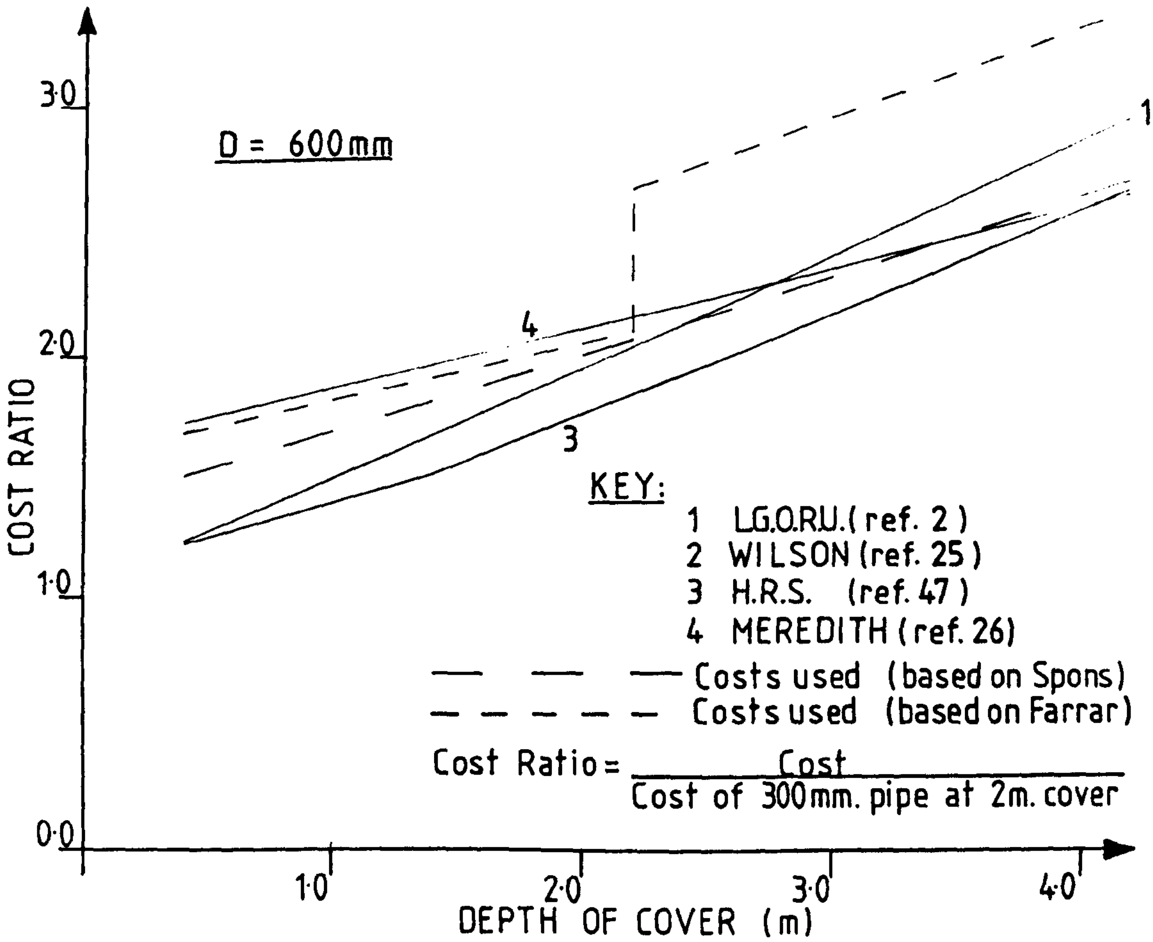
When previous authors on drainage optimisation have quoted the cost function they have used, it has ten been in a generalised form (see Table 4.1). Without knowing the values of the constants in these functions they are of little practical use.

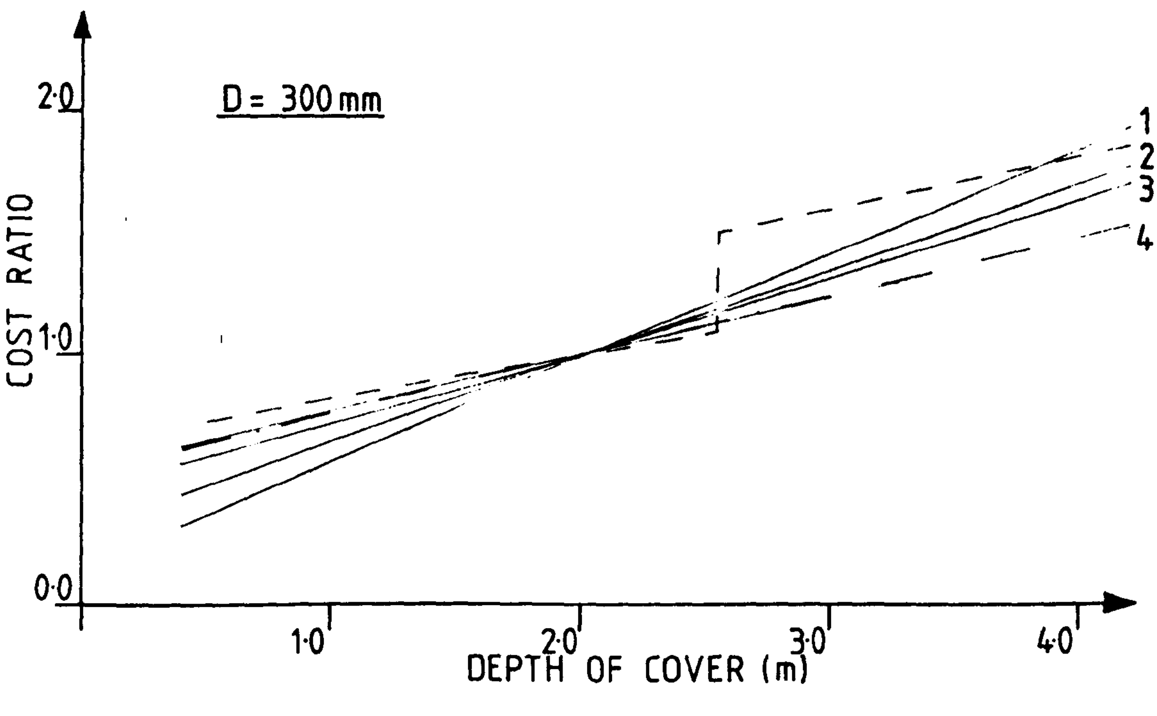
SOURCE	COST/UNIT LENGTH	NOTES
Lemieux, Zech and Delarue. (ref. 21)	$a + bD^n + eV$	e = unit cost of excavation
Meredith (ref. 26)	$10.98D + 0.8H - 5.98$	Cost in dollars D, H in feet H is depth to invert.
Dajani and Hasit (ref. 17)	$a + bD^2 + cH^2$	D, H in feet H is depth of excavation.
Barlow (ref. 42)	$aV + bD^n$	
Wilson (ref. 25)	$0.73D + 0.243H - 0.088$	H is depth to soffit. D, H in feet.

TABLE 4.1 - PUBLISHED COST FUNCTIONS FOR DRAINAGE

General Notes: a, b, c, n are unspecified constants.
D = pipe diameter
V = volume of excavation per unit length

Two authors however quote the specific form of their cost functions. These are included in Table 4.1 and are illustrated for two pipe sizes in a dimensionless form in Fig. 4.2. Also illustrated are costs taken from work done by the Hydraulics Research Station (ref. 47) and from a report by the Local Government Operational Research Unit (ref. 2).





COST FUNCTIONS

FIGURE 4.2

For the purposes of this research the author developed a set of cost functions for discrete pipe sizes, based on Spon's Architects and Builders Price Book (ref. 46) and prices quoted by pipe manufacturers. Details of the calculations are given in Appendix A.

In producing these cost functions, the following assumptions were made:

(1) Structural design of the pipe and bedding was to be in accordance with Department of the Environment recommendations (ref. 48) and was to be for pipes laid in a road carriageway.

(2) The cheapest satisfactory combination of pipe type and pipe bedding was to be used for a given pipe depth and diameter.

(3) Average soil conditions applied throughout, with no hard rock or exceptional dewatering requirements.

(4) There was no breaking up of road surface or reinstatement required.

(5) There was adequate working room for excavation.

(6) All excavation was by machine, there being no necessity for hand excavation.

(7) Surplus fill material could be disposed of on site.

These conditions are generally consistent with drainage schemes for new roads. The main exception would be requirement (3), as variable soil conditions and high water tables could be encountered in cuttings.

The cost functions developed give a rate per unit length for the finished pipeline, and for a given pipe diameter, depend only on the depth of cover (Y) over the pipe. These functions are based on prices in March 1977 and are as follows:

Pipe Diameter (mm)	Cost (£ per m)
150	2.8 + 4.1 Y
225	5.7 + 4.1 Y
300	8.9 + 4.1 Y
375	12.3 + 4.4 Y
450	15.9 + 4.7 Y
525	19.7 + 5.0 Y
600	23.7 + 5.3 Y

As these functions are linear with depth, it is reasonable to take the cost of a pipeline between manholes = L x f (Yav), where L is the distance between the centres of the manholes and Yav is the average cover along the length of the pipe. The cost of the upstream manhole depends on the depth of the manhole, measured from ground level to the lowest pipe invert, and on the manhole diameter. In turn the manhole diameter is determined by the biggest pipe entering or leaving the manhole. Assuming that pipe diameters cannot decrease down the pipe network (see 2.3.6), the largest pipe must be the pipe leaving the manhole. Now as the lowest invert is that of the outgoing pipe, the manhole cost is determined by the diameter and invert level of the outgoing pipe. But since depth to soffit = depth to invert - diameter of pipe, the cost of the upstream manhole of an element can be taken as f(D, Yu) where Yu = depth of cover at upstream end of the pipe. Hence the total cost of an element is a function of pipe length, pipe diameter, average depth of cover, and depth of cover at the upstream manhole i.e. Ce = f(L, D, Yav, Yu).

The cost of an element as used for this study is thus given below:

Pipe Diameter (mm)	Element Cost (£)
150	(2.8 + 4.1 Yav)L + 30 + 70 Yu
225	(5.7 + 4.1 Yav)L + 30 + 70 Yu
300	(8.9 + 4.1 Yav)L + 30 + 75 Yu
375	(12.3 + 4.4 Yav)L + 30 + 80 Yu
450	(15.9 + 4.7 Yav)L + 30 + 85 Yu
525	(19.7 + 5.0 Yav)L + 30 + 90 Yu
600	(23.7 + 5.3 Yav)L + 30 + 95 Yu

4.3.4. Alternative Cost Function

A more comprehensive set of cost equations was developed based on the work of Farrar (ref. 45) and is included in the final commercial drainage design computer program resulting from this research. Details of these functions are given in Appendix B.

The two sets of cost functions developed for use in this research are compared in Fig. 4.2 with previously published information, using two typical pipe diameters.

CHAPTER 5

THE FIXED PLAN OPTIMISATION MODEL - MANFIX

5.1 Introduction

5.2 Problem Definition

5.3 Optimising the Objective Function

- 5.3.1. The Objective Function
- 5.3.2. The Polytope or Simplex Method
- 5.3.3. Using A Smooth Continuous Objective Function

5.4 A Serial System

5.5 Drainage as a Serial System

- 5.5.1. Introduction
- 5.5.2. The Basic System
- 5.5.3. A Branching System
- 5.5.4. Non-decreasing Pipe Diameter
- 5.5.5. Design Flows that are not Pre-determined
- 5.5.6. Summary

5.6 Optimising a Serial System by Dynamic Programming

- 5.6.1. Introduction
- 5.6.2. Principle of Optimality
- 5.6.3. State Vector Space
- 5.6.4. Dynamic Programming Using Discrete Values

5.7 Optimising a Simple Fixed Plan Drainage Run by Dynamic Programming

- 5.7.1. Upper Bound on State Variable
- 5.7.2. Lower Bound on State Variable
- 5.7.3. Establishing Discrete Values of Level
- 5.7.4. Establishing a Feasible Design for an Element
- 5.7.5. Cost at Each Discrete Input State
- 5.7.6. Cost at Each Discrete Output State
- 5.7.7. Overall Minimum Cost
- 5.7.8. Optimal Solution

5.8 Inclusion of Drop-Manholes

- 5.8.1. Introduction
- 5.8.2. Defining the Quasi-Input State
- 5.8.3. Tracing Back

5.9 Optimising a Branched Drainage Network

5.9.1. Introduction

5.9.2. Procedure

5.9.3. Tracing Back Through a Branch Junction

5.10 Inclusion of the Constraint on Decreasing Pipe Diameter

5.10.1. Introduction

5.10.2. Procedure

5.10.3. Defining the Range of Diameters and their Discrete Values

5.10.4. Organising the Computation

5.11 Dependence of Flows on the Network Design

5.11.1. Introduction

5.11.2. The Rational or Lloyd-Davies Method

5.11.3. Other Methods of Calculating Storm Water Flows

5.11.4. The three Dimensional State Vector Approach

5.11.5. An Approximate Approach

5.12 The Final Fixed Plan Model - MANFIX

5.13 The Use of Discrete Differential Dynamic Programming

5.14 Experience and Results

5.14.1. Introduction

5.14.2. Experience of the Model

5.14.3. Results

5.15 Cost of Using MANFIX

5.16 Conclusion on the Fixed Layout Model - MANFIX

Chapter 5

The Fixed Plan Optimisation Model

5.1 Introduction

Whilst the main area of research for the present project was in the field of variable plan networks, it was an essential prerequisite to examine published work on fixed plan models. During this examination it was found that there were shortcomings (ref. 31) in all published methods, and that there was no one approach that seemed entirely satisfactory. Of the methods that were available, Discrete Differential Dynamic Programming (DDDP) seemed to have gained most acceptance and this is discussed in Section 5.13.

During the development of the variable plan models it became clear that a simple fixed plan Dynamic Programming (DP) model, essentially a subset of the variable plan model, could be of interest.

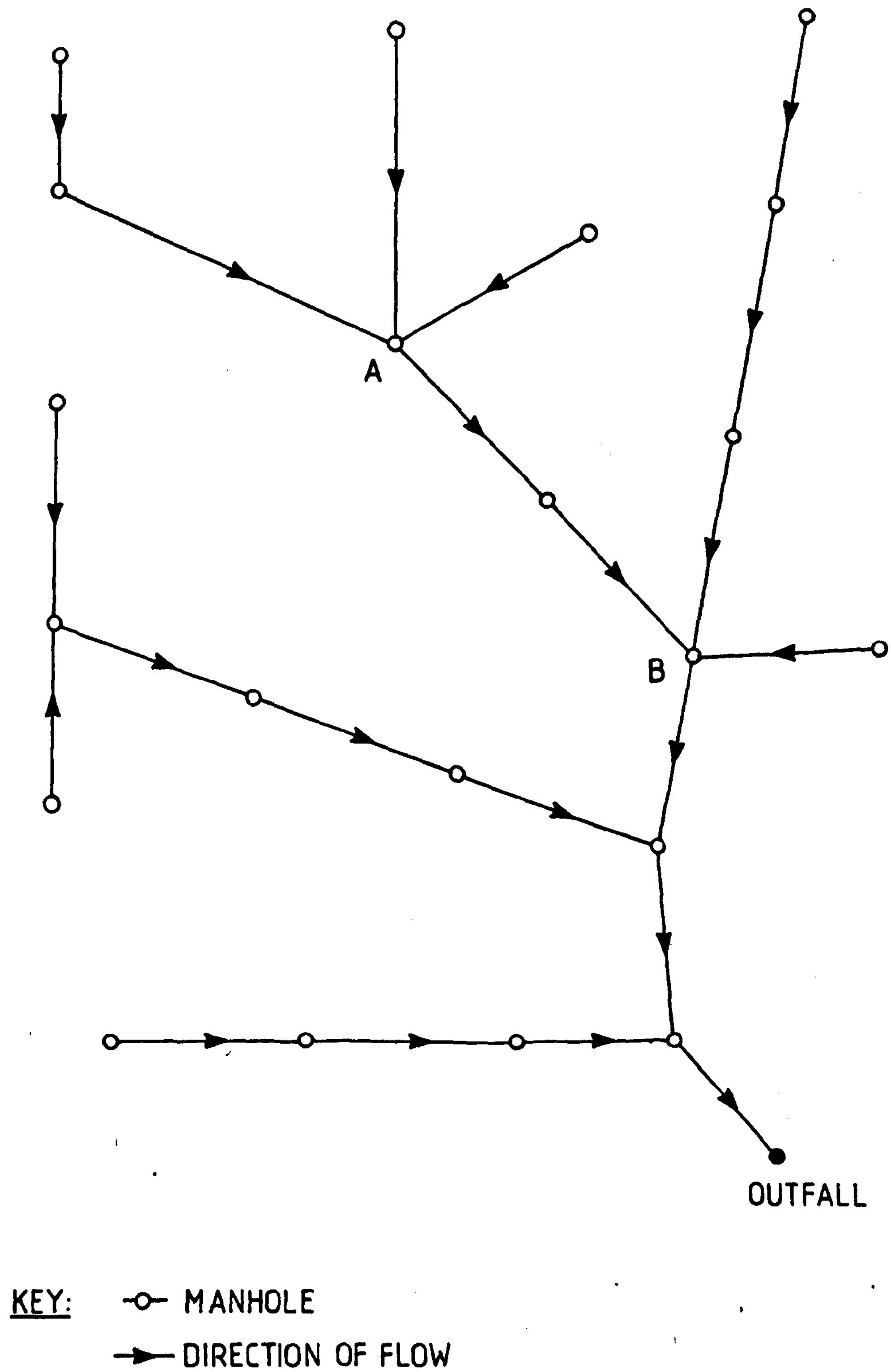
Although a separate computer program for the fixed plan model was not written, the model is presented in this chapter both for completeness and as an introduction to the more complicated variable plan problem. Results, conclusions and the choice of parameters are based on computer runs using the variable plan models (see Chapter 6) on fixed plan problems.

5.2 Problem Definition

Fixed Plan Optimisation represents the most basic level of improvement over current design methods, and is the simplest of the drainage network optimisation problems considered. It is also applicable to virtually all storm drainage networks, and with minor modifications, to foul sewer networks.

The designer specifies the plan layout of the pipes and the positions of all manholes. One tree of a typical network is shown in Figure 5.1. The problem is to find admissible pipe diameters and levels for every pipe in the system so that the total construction cost for the system is as small as possible, whilst all the technological and physical constraints imposed on the system are met.

As an example, consider a network of n pipes between (n + 1) manholes fixed in plan.



TREE OF A TYPICAL DRAINAGE NETWORK

FIGURE 5·1

The design of an element i (Figure 4.1) can be defined in terms of the pipe diameter D_i, pipe level at the upstream end Zu_i, and pipe level at the downstream end Zd_i. In general, given Zu_i and Zd_i, the smallest, and hence cheapest, pipe size that will carry the required flow and satisfy the design constraints, will be chosen. Hence the pipe diameter D_i is dependent on Zu_i and Zd_i and need not be considered as an independent variable. There are thus 2n variables in the problem.

The cost of constructing the pipe element $Ce_i = f(D, Yav, Yu)$ (see 4.3.3), where Yav and D are functions of Zu_i and Zd_i, and Yu is a function of Zu_i. Thus Ce_i is a function of Zu_i and Zd_i for a given design flow and set of ground levels. Hence the problem becomes one of minimising C where

$$C = Ce_1(Zu_1, Zd_1) + Ce_2(Zu_2, Zd_2) + \ldots\ldots \qquad \text{---- A}$$
$$+ Ce_i(Zu_i, Zd_i) + \ldots\ldots\ldots\ldots\ldots \quad , Ce_n(Zu_n, Zd_n)$$

subject to the following constraints(see 2.3)

$Ymin \leq Y \leq Ymax$
$smin \leq s \leq smax$
$Vmin \leq V \leq Vmax$
$Q \leq Qf$
$Zu \leq Zus$
$D \geq Dus$
D a discrete, available, diameter

The problem could indeed be further simplified by specifying that all pipes at a manhole must have the same level. The problem would then reduce to that of finding a pipe level at each of the (n + 1) manholes. There would thus be only (n + 1) variables.

The problem would then become:

Minimise C,

where $$C = Ce_1(Zu_1, Zd_1) + Ce_2(Zu_2, Zd_2) + \ldots \qquad \text{--------B}$$
$$+ Ce_i(Zu_i, Zd_i) + \ldots\ldots\ldots\ldots Ce_n(Zu_n, Zd_n)$$

with the (n - 1) equalities $Zd_j = Zu_k$ (j = 1, n - 1),
(where k depends on the connectivity of the network)
and the following constraints:

$Ymin \leq Y \leq Ymax$

$smin \leq s \leq smax$

$Vmin \leq V \leq Vmax$

$Q \leq Qf$

$D \geq Dus$

D a discrete, available, diameter

The disadvantage of this approach is that it is far too restrictive for practical drainage networks. Branches joining a main run will, for example, generally join at a much higher level. To restrict them to joining at the main pipe level could incur severe financial penalties.

Hence it is better to consider only the general form of the problem as in expression A.

5.3 Optimising the objective function

5.3.1 The objective function

The expression that is to be minimised, expression A, is known as the objective function and is here a non-linear function of 2n variables, where n, the number of pipes in the network, is unlikely to be less than 10 and could be as many as several hundred.

Consider the cost of a typical element $Ce_i(Zu_i, Zd_i)$. For Zu_i = constant, consider the range of values of Zd_i. Assuming available pipe diameters are in discrete sizes, there will be discrete values of diameter D_i for different parts of the range Zd_i. Hence there will be jumps in the cost of the element where the required diameter goes from one size to the next. The cost function for an element is thus discontinuous and therefore also non-differentiable.

Even if diameters were available in a continuous range of sizes the cost function for an element could still be discontinuous. This would occur if the cost function truly represented the cost of the different site practices involved in excavating pipe trenches to various depths.

For example, trench supports are not normally required for trench depths up to 1.5m, but trenches deeper than this must be supported for safety. Similar discontinuities with increasing depth could arise from the use of differing classes of pipe, types of bedding design or widths of trench.

The constraints on the optimisation are in the form of both linear and non-linear inequalities (see 2.3). For example, constraints on depth and pipe slope are linear inequalities, whereas those on flow velocities and discharge are non-linear inequalities.

Hence the objective function is a non-linearly constrained multi-variable non-linear discontinuous function. There are at present no suitable mathematical techniques available for the general solution of this type of optimisation problem.

5.3.2. The polytope or simplex method

For problems involving a very small number of variables, say less than about 10, a polytope algorithm could possibly be used.

Essentially the polytope technique applied to a problem with m variables involves the following procedure.

(a) Define the feasible zone of the m dimensional space within which the solution must lie.

(b) Define (m + 1) points within that space, preferably equally spaced, and evaluate the function at these points.

(c) Identify the worst (most expensive) points.

(d) Reflect the worst point through the centroid of the other points to obtain a new point.

(e) Evaluate the function at the new point, identify the new worst point and repeat from step (d).

The polygon may be expanded or contracted according to various rules. Other rules may also limit the deformity of the polygon and specify the procedure to adopt at constraint boundaries. The process continues until the polygon is reduced to a predetermined size and further iterations produce negligible improvement.

Unfortunately there are doubts as to its ability to find the optimal solution for even a moderate number of variables. In addition the process is relatively slow, requiring a large number of function evaluations.

Although the technique is known to be very robust, there could well be difficulties encountered in using it with an objective function such as expression A which has a number of large discontinuities corresponding to discrete values of pipe diameter.

5.3.3. Using a smooth continuous objective function

If one ignores the problem of discontinuities outlined in 5.3.1. and treats the function as smoothly continuous, a range of possible solution techniques emerge, depending on whether first and second derivatives of the function can be evaluated.

Consider first a problem in which there are no constraints. If first derivatives cannot be evaluated, even though they uniquely exist at all points, the minimum could be found by a linear search method using only function evaluations.

All such methods follow the general iteration $\underline{x}_{i+1} = \underline{x}_i + w_i.\underline{e}_i$ where $\underline{x}_{i+1}$ is the improved position and $\underline{x}_i$ is the old position of the vector x which defines the position of the search, w_i is a step length and $\underline{e}_i$ is the direction of the step.

The simplest of all such methods uses each axial direction in turn as the search direction $\underline{e}_i$, with the step w_i being determined by a linear search along that one direction. The current best point then moves parallel to each axis in turn.

Various algorithms have been developed by Hooke & Jeaves (ref. 49), Rosenbrock (ref. 50), Davies Swann & Campey (ref. 51) and others as improvements to the basic method.

As an alternative to linear search methods, a gradient method could be adopted by using information about the first and sometimes the second derivatives as well as the function values to help determine the direction of search $\underline{e}_i$.

The derivatives can be obtained either analytically if suitable formulae are available, or numerically from evaluations of the objective function, although this latter course has the disadvantage of extra function evaluations and possible problems with arithmetic calculation of very small quantities.

The most basic approach is to follow the line of steepest descent until the minimum value of the objective function along that line is reached, whereon a new direction is established and the process is repeated.

When second derivatives are available a far more powerful class of methods known as Newton methods may be used. Around its minimum value, the objective function can be assumed to be approximately quadratic, and for such a function it can be shown by Taylor expansion that the correction $\underline{\delta}$ for which x + $\underline{\delta}$ minimises the function can be written as $\underline{\delta} = -G^{-1}\underline{g}$ where $\underline{g}$ is the gradient vector and G is the matrix with elements $G_{jk} = \dfrac{\partial^2 f}{\partial x_j \partial x_k}$

Hence the iteration becomes $\underline{x}_{i+1} = \underline{x}_i - G^{-1}\underline{g}$

Modifications to the basic Newton method involving the use of only first derivatives and only function evaluations have been made, notably by Davidson (ref. 52) and Fletcher and Powell (ref. 53).

These Quasi Newton Algorithms tend to be the most efficient in terms of function evaluations, although if computer storage is critical a conjugate gradient method (ref. 54) may be more suitable.

All the algorithms so far mentioned are for the general unconstrained problem and in particular the storm drainage problem has both non-linear and linear inequality constraints.

One approach taken to such problems is to create penalty functions corresponding to the constraint boundaries so that the value of the function rises rapidly at the constraint thereby prohibiting the minimum value from being beyond the constraint boundary.

The function may then be minimised as an unconstrained problem. However severe problems can occur due to ill conditioning at the boundaries and generally several unconstrained problems have to be solved with varying values of penalty functions to obtain the true optimal solution.

Another approach is again to convert the problem to an unconstrained one, but this time by creating an augmented Lagrangian function (ref. 55). Alternatively the non-linearly constrained problem may be transformed into an equivalent linearly constrained exercise.

A rather different approach is to modify the search direction to avoid entering a non-feasible zone. Such techniques are known as projected gradient techniques. Essentially if the proposed search direction contravenes a constraint, a new direction is adopted, being the projection of the original onto the tangent plane of the constraint.

Yet another approach is that of Geometric Programming (ref. 24). The objective function must be expressed as a posynomial, (a function which is the sum of positive polynomial terms) and constraints should also be posynomial expressions. The method is based on the general geometric inequality theorem, which states that the arithmetic mean of a set of positive terms is always greater than their geometric mean, with equality when all the terms are equal.

Whilst the methods outlined above will, at least in theory, provide optimal solutions for a continuous smooth objective function there may still be severe problems due to lack of robustness particularly with complicated constraints.

The main problem, however, remains: The actual objective function is discontinuous. This could be avoided by allowing pipes to be of any size and ignoring practical discontinuities in the cost function corresponding to site practice or design. After obtaining the optimal solution, the pipe diameters must then be converted in some way to commercially available sizes. Attempts at doing this have met with only limited success (refs. 23, 25).

The conclusion therefore must be that there is no suitable technique for solving the general problem of which storm drainage optimisation is a particular example. It is therefore logical to examine whether storm drainage optimisation is in any useful way different from the general problem.

5.4 A Serial System

It is convenient here to introduce the concept of a serial system, for which a powerful alternative optimising approach is available.

The essence of such a system is that a quantity S, called the state, passes in one direction through a sequence of stages at each of which it is modified in value by decisions which produce returns. This is illustrated in fig. 5.2(a).

The quantity S has an initial value S_0 which is the input state to stage 1. In stage 1 decisions d_1 are made which change the value of S_0 to S_1 - the output state from stage 1 - and produce a stage return r_1. S_1 is then the input state for stage 2 at which decisions d_2 are made, producing stage returns r_2 and changing the value of S from S_1 to S_2. This process of making decisions at each stage which change the value of the state and produce stage returns continues until all N stages have been traversed and the state has a final value S_N.

The serial system must contain no loops. At any particular stage, say stage k, the only information known about the system is the input state S_{k-1} and the details within stage k. Hence the decision made, d_k, the return r_k and the output state S_k can only be influenced by the input state S_{k-1} and not by how that state was achieved (i.e. not by decisions d_1 to d_{k-1}).

5.5 Drainage as a Serial System

5.5.1. Introduction

The design of a drainage network may, with care, be treated as a serial system.

First consider the simplest case. This is a non-branching length of sewer consisting of N pipes between N+1 manholes as shown in Figure 5.2(b). The constraints listed in Chapter 2 section 3 will, in general, apply to the design. These are summarised below for convenience.

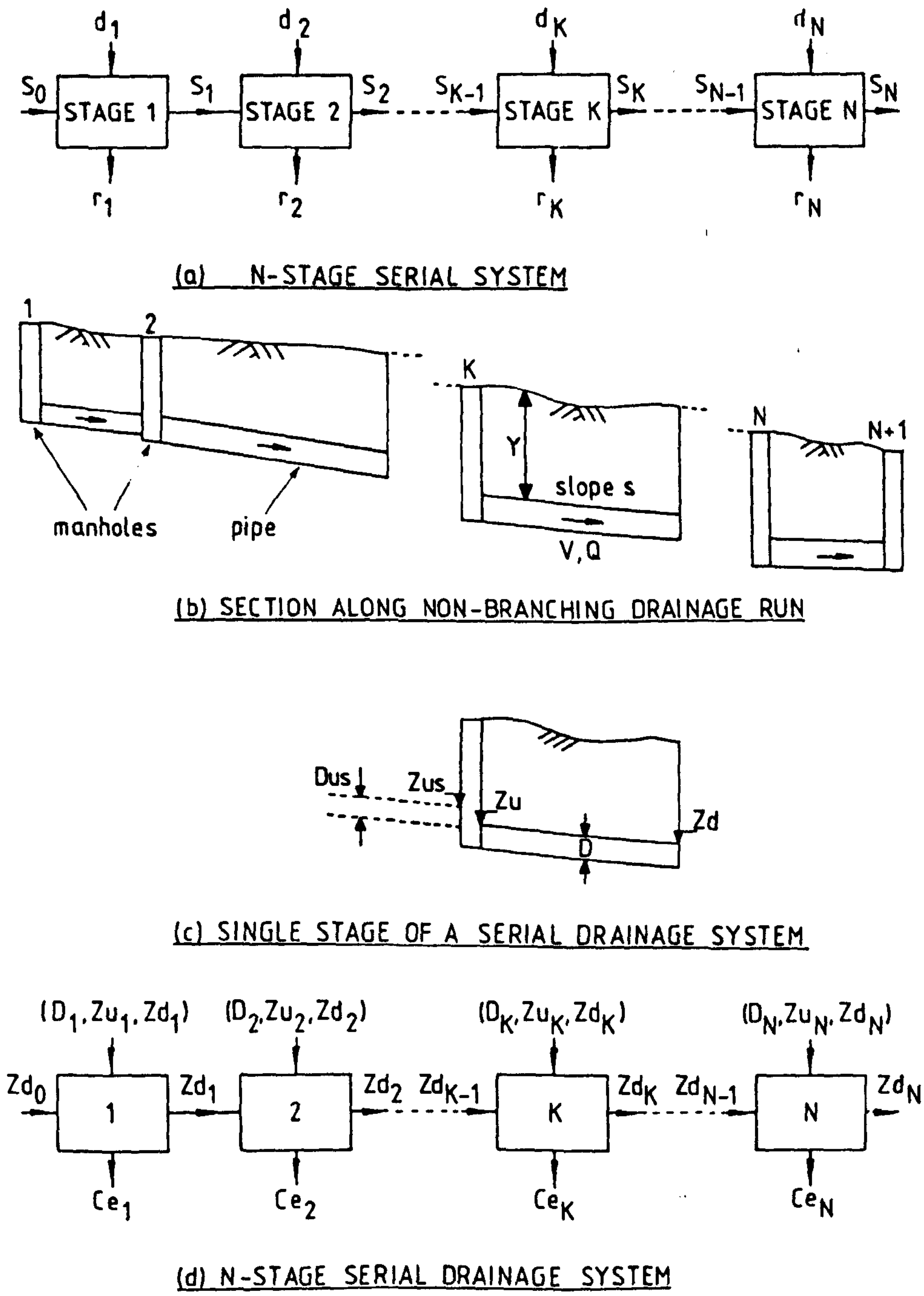

d1
d2
dK
dN
S0
STAGE 1
S1
STAGE 2
S2
SK-1
STAGE K
SK
SN-1
STAGE N
SN
r1
r2
rK
rN
(a) N-STAGE SERIAL SYSTEM
1
2
K
N
N+1
Y
slope s
manholes
pipe
V, Q
(b) SECTION ALONG NON-BRANCHING DRAINAGE RUN
Dus
Zus
Zu
Zd
D
(c) SINGLE STAGE OF A SERIAL DRAINAGE SYSTEM
(D1, Zu1, Zd1)
(D2, Zu2, Zd2)
(DK, ZuK, ZdK)
(DN, ZuN, ZdN)
Zd0
1
Zd1
2
Zd2
ZdK-1
K
ZdK
ZdN-1
N
ZdN
Ce1
Ce2
CeK
CeN
(d) N-STAGE SERIAL DRAINAGE SYSTEM

FIGURE 5.2

a) $Ymin \leq Y \leq Ymax$

b) $smin \leq s \leq smax$

c) $Vmin \leq V \leq Vmax$

d) $Q \leq Q\ full$

e) $Zu \leq Zus$

f) $D \geq Dus$

g) D is a discrete, available diameter

Constraints a, b, c, e and g present no problems to the concept of drainage as a serial system. Zu and Zus refer to the pipe soffit levels although there is no theoretical argument against using the invert or pipe centre line as the reference for the pipe levels.

Constraint (d) raises the question of the design flow Q. For simplicity first assume that all design flows are known before the design starts. This is generally not the case for storm-water drainage and is a question which will be considered later (see section 5.5.5).

Constraint (f) fundamentally changes the nature of the system and so the system will be considered with or without this constraint. Initially, consider the simpler case of the constraint not applying.

5.5.2. The basic system

Consider a single stage of the system as shown in Figure 5.2 (c). This consists of a pipe together with its upstream manhole. The complete system consists of N such stages starting from stage 1 at the upstream end of the sewer and ending in stage N which has a downstream manhole as well as the usual upstream one. Let the input state to stage K be the soffit level of the pipe entering the upstream manhole Zus_K. One can now make a decision on pipe levels and diameter for this stage based on the input state and design flow such that all constraints are satisfied. There will in general be many possible decisions. The choice of pipe levels and diameter will incur a return for the stage which is here considered to be the construction cost of the element, and will produce an output state, the level at the downstream end of the pipe. This output state forms the input state to the next, (K + 1)th, stage.

The serial nature of the system is shown diagrammatically in Figure 5.2(d). Note that $Zus_K \equiv Zd_{K-1}$ and that the input state is not the level of the pipe leaving the upstream manhole but merely the highest level at which that pipe could be set. Hence the decision on pipe levels can involve a change in level or 'drop' across a manhole. For an isolated drainage run, the input state to stage 1, Zd_0 and the output state from stage N, Zd_N are not required, but where the run forms part of a larger network they will be used, as outlined in the following section.

5.5.3. A branching system

Having shown that a simple non-branching sewer can be treated as a serial system, it is now necessary to consider a branching system.

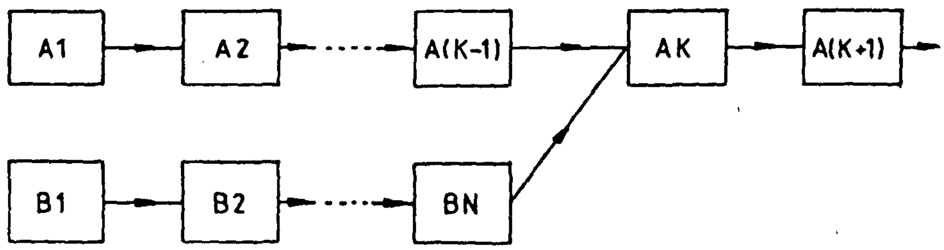
Drainage systems are typically arranged as tree-like networks as shown in Figure 5.1. There are no loops, at least not in newly designed networks, although old existing systems often have cross connections and diverging flows.

As it is the design of new networks under consideration it is reasonable to assume that there are no loops and that sewers never diverge, but always converge. The convergence of two serial systems is illustrated in Figure 5.3(a). The only complication is that the input to stage K, has to be determined from both the ouput state from stage A_{K-1} and from the output state from stage B_N. This is done by redefining the input state as the soffit level of the lowest pipe entering the upstream manhole. It is in fact rather more convenient to rearrange the serial system slightly as shown in Figure 5.3(b). Instead of one sewer joining a main sewer, we now have two sewers leading into a third sewer. These are exactly equivalent but the latter arrangement is easier to handle computationally.

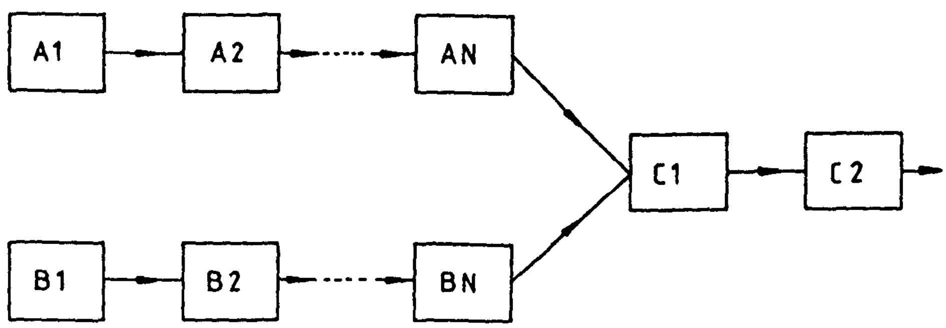
5.5.4 Non-decreasing pipe diameter

As mentioned in section 5.5.1., constraint (f) fundamentally changes the nature of the serial system. This fact has not generally been recognised by previous authors (e.g. refs. 28, 34, 44) and has led to the use of incorrect algorithms (ref. 31).

Consider first the basic system as described in 5.5.2. and illustrated in Figure 5.4(a). At stage K, information about the upstream system (stages 1 to K-1) is conveyed purely by the state variable Z. A decision



(a)



(b)

CONVERGENCE OF SERIAL SYSTEMS

FIGURE 5.3

as to the design of stage K is made on the basis of the information available at that stage, i.e. the input state Z, the design flow Q and the constraints. None of these constraints are affected in any way by the design or conditions outside stage K.

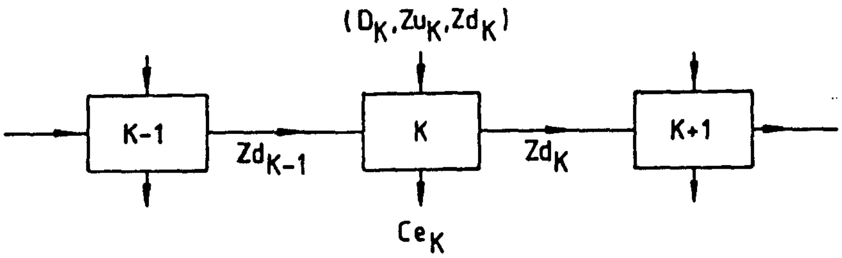
Next consider the introduction of constraint (f), i.e. pipe diameter must not be less than the upstream pipe diameter. Where previous authors have used this constraint, they have treated the system serially as described above and have used the constraint as just another condition on the selection of suitable levels and diameter for each stage as it is designed. This, however, destroys one of the essential features of a serial system, that there should be no loops. The decisions at a stage are no longer made purely on information available at that stage. They now use a constraint which is affected by the design of the previous stage. This is illustrated in Figure 5.4(b).

This problem can however be handled correctly with a certain amount of rearrangement. Instead of a single state variable Z, an additional variable, D, the upstream pipe diameter can be introduced. The state is now defined by the values of Z and D. Where several pipes enter a manhole, D is defined as the diameter of the largest of these pipes. A decision at stage K can now once again be made using only the information available at that stage, i.e. the input state (Z, D) the design flow Q and the full set of constraints. None of the constraints now refer to information not available either as input to that stage or as information within the stage.

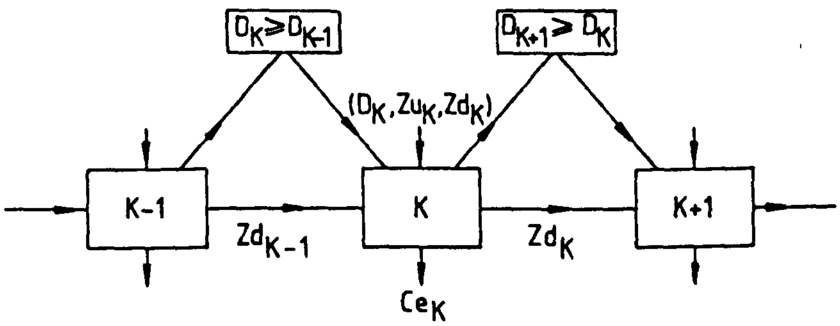
The new serial system is illustrated in Figure 5.4(c).

5.5.5. Design flows that are not pre-determined

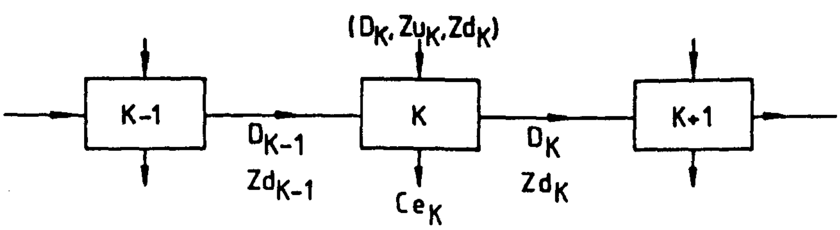
It is normal practice in the design of stormwater drainage networks for the flow at points in the network to be dependent on the design of the network upstream. This is true both for the simple Rational (Lloyd-Davies) method of design and for more sophisticated procedures using routing techniques, e.g. The Transport and Road Research Laboratory Hydrograph method, (refs. 5, 6).



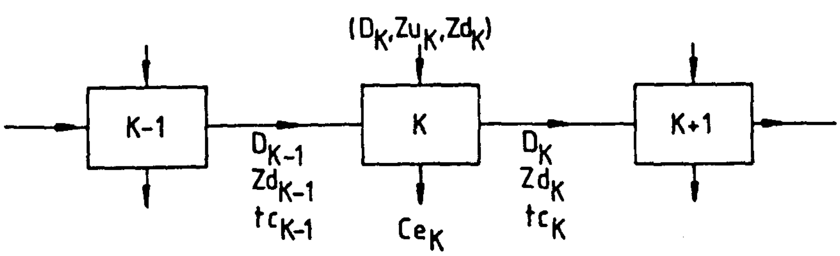
(a) THE BASIC SERIAL SYSTEM



(b) SYSTEM WITH DIAMETER CONSTRAINT : NON-SERIAL



(c) SYSTEM WITH DIAMETER CONSTRAINT; SERIAL



(d) SYSTEM WITH RATIONAL METHOD

FIGURE 5.4

As an illustration consider the Rational method. Essentially the flow at a point in a stormwater network is dependent on the total equivalent impermeable catchment area upstream of that point and on the maximum time taken by stormwater to reach that point from any point upstream. This time is known as the time to concentration. The greater the time to concentration, the less the design rainfall rate and hence the less the design flow. Whilst the impermeable area is fixed and may be determined before the start of any design, the time to concentration depends on the diameter, slope and roughness of all pipes upstream of the point considered. Hence the design flows cannot be predetermined.

To treat this situation as a true serial system, one has to use a third state variable, the time to concentration. The flow at stage K may then be determined from the stage input and decisions made strictly internally for that stage.

Such a serial system is shown in Figure 5.4(d).

As will be shown later, this concept is of limited usefulness (see section 5.11).

5.5.6. Summary

Design of drainage may be treated as a serial system with one, two, or three state variables depending on the nature of the constraints and the design flows.

Converging, tree-like networks present no problems to the concept of serial systems.

5.6 Optimising a Serial System by Dynamic Programming

5.6.1. Introduction

In 1957 Richard Bellman wrote a book entitled Dynamic Programming (ref. 15). This text introduced a novel mathematical approach to the problem of optimising multi-stage decision processes, and the name Dynamic Programming has been retained for the general approach he devised.

5.6.2. Principle of Optimality

Bellman stated in his principle of optimality that "An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision". This is intuitively obvious and as Bellman argues, a proof by contradiction is immediate. Applying this principle recursively, an alternative and equivalent statement can be made, namely "any subarc of an optimal path is itself optimal".

It is this principle that allows the decomposition of a multi-variable serial optimisation problem into the successive optimisation of problems with small numbers of variables.

5.6.3. State vector space

The input to a stage can be considered as a vector, with one dimension for each state variable. A state variable may be a continuous function, or may only exist at discrete values. Hence the state vector may be continuous, discontinuous or discrete in nature.

An optimisation problem concerns finding the set of values for the state vector at each stage such that the total return from the system is maximised or minimised. As minimisation is merely maximising a negative return, the argument may be restricted to maximisation.

It is in the nature of Dynamic Programming that the optimal values of the state vector are not known until all the stages have been considered. At each stage, however, a range of values of the state vector is considered. This range must be predefined, and must include the final optimal value of the vector. Thus an allowable state vector space is defined at each stage prior to the optimisation process.

5.6.4. Dynamic programming using discrete values

There are many different ways of optimising serial systems using Dynamic Programming. All use a similar approach, and the way described below is the most useful for the drainage network problem.

The state vector is considered to be discrete valued, whether or not this corresponds to a physical reality.

The vector space is predetermined at each stage, and so are all the individual discrete values of the vector within that space. Hence for a particular stage K, there is a set of possible values for the input state vector, and a set of possible values for the output state vector.

Assume that for each discrete value of the input state vector at stage K, the total optimal return for stages 1 to (K-1) is known. Consider a particular discrete value of the output state vector. The optimal way of arriving at that output state must now be obtained. This is done as follows:-

(1) consider a discrete input state.

(2) optimise the design from the discrete input state to the discrete output state by making the decisions which maximise the individual stage return whilst conforming to any design constraints. This may in itself be a complicated optimisation problem or may be trivial as in the case of drainage networks. There may indeed be no feasible solution, in which case a very large negative return can be assigned to this combination of input and output states.

(3) Add the stage return to the total optimal return for stages 1 to (K-1) for the discrete input state considered.

(4) If this total return is greater than for any previous way of arriving at the same output state, the value of the return is retained, as are references to the decisions that led to it, and any previously stored values for this output state are discarded. If the return is less, then the value of the return and the details of the design are ignored.

(5) If there are any discrete input states that have not yet been considered, return to (1).

Hence the optimal return has been obtained for a discrete value of the output state vector, and the stage decisions which led to this optimal return are known.

This process must now be repeated for each discrete value of the output state vector. Hence we end up with a set of values for the optimal return for each discrete value of the output state vector at stage K. These can be used to form a set of optimal returns for each discrete value of the input state vector at stage K+1.

The process may now be repeated for stages (K+1) to N of the N stage system.

On completion of the Nth stage, the returns for each discrete output state can be examined and the maximum return selected. This is then the value of the optimum return from the system.

In itself this is of little use. What is required is the set of decisions at each stage that led to this optimum return. This can be established by tracing the optimal solution back through the system as follows:

(1) Identify the output state corresponding to the optimal return at output from stage N.

(2) Identify the decision for stage N that led to this output state, together with the corresponding input state, (a set of such data having previously been stored for each output state).

(3) For the particular input state identified, identify the corresponding output state for the previous stage.

(4) Identify the decisions for this stage that led to this output state, together with the corresponding input state.

(5) Repeat (3) and (4) until the first stage is reached, whereupon a complete set of optimal decisions for the system will have been identified.

5.7 Optimising a simple fixed plan drainage run by Dynamic Programming

The simplest drainage network is that of a single pipe run with manholes at fixed positions along it. The sizes and slopes of all the pipes for the optimum design may be obtained in the following manner.

Consider first the case of a single state variable, the pipe soffit level at a manhole. Flows are assumed to be independent of the pipe network design, and pipe diameters are not constrained, thus being free to decrease in diameter downstream. Assume also for simplicity that drops in level across manholes are not permitted (see 5.8). A stage is as defined in 5.5.2. The input state is the pipe soffit level at the upstream end of the pipe. The output state is the pipe soffit level at the downstream end of the pipe. The output from state K equals the input for stage (K+1).

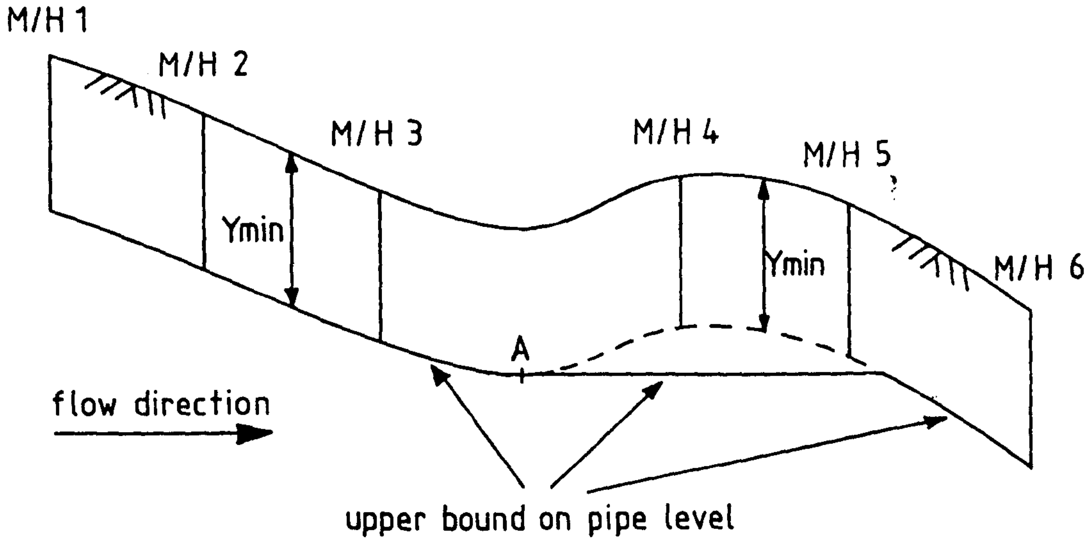
It is now necessary to consider the range of permissible states at each stage. For a typical drainage problem the range to consider is not at all obvious. On the one hand the range chosen must be sufficient to guarantee the inclusion of the global optimum, and yet not so large as to incur severe computational penalties. The problem may be circumvented by adopting a rather different D.P. approach called Discrete Differential Dynamic Programming or D.D.D.P. (ref. 32) as described in section 5.13. As D.D.D.P. is unsuitable for variable plan problems which lead on from the problem at present under consideration, it is necessary to find a rational basis for defining the range of states using the more conventional D.P. approach.

5.7.1. Upper bound on state variable

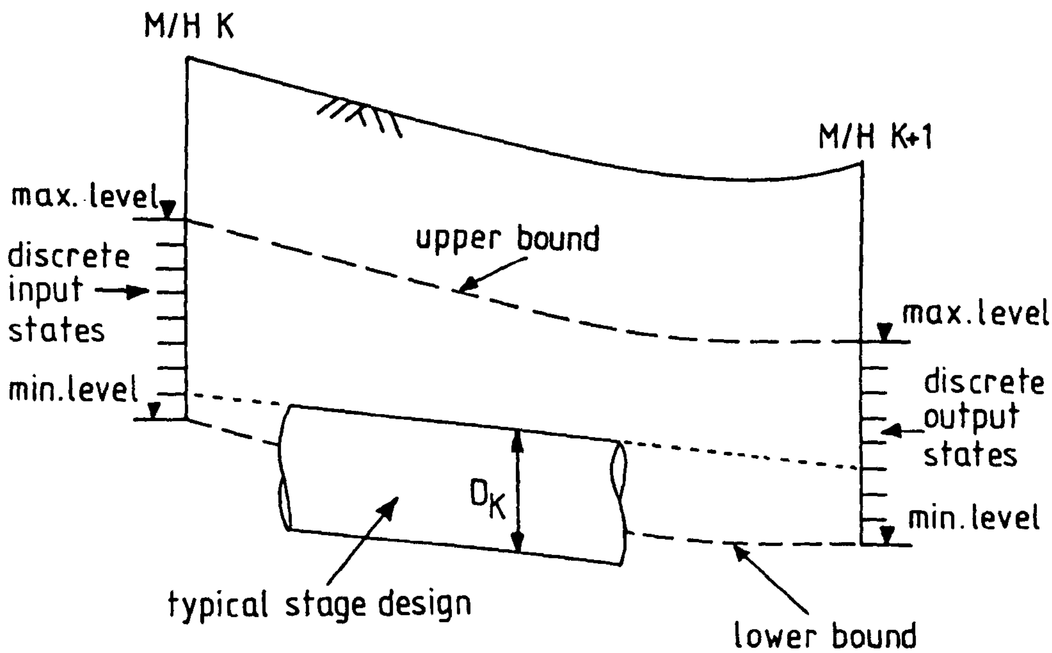
There will generally be a restriction on depth of cover, (constraint (a) of section 2.3), such that the pipe soffit level must be less than the ground level minus the depth of cover.

Hence there is some sort of upper limit on the range of levels that should be considered at each stage. Considering the pipe run shown in Figure 5.5 (a), one can see that although this would form a reasonable bound for pipe lengths (1)→(2) and (2)→(3), it would not be realistic for the rest of the run. Obviously the level downstream of A cannot exceed the level at A. If there is a minimum gradient specified (constraint (b) of section 2.3) this may be applied downstream of A to give a modified upper limit.

If there is no minimum gradient specified, the upper limit may still be restricted by a constraint on minimum velocity of flow in the pipe. However, by choosing a very large diameter pipe, the pipe slope to



(a) ESTABLISHING UPPER BOUND ON PIPE LEVEL



(b) THE DESIGN OF A STAGE

FIGURE 5.5

achieve a minimum velocity restriction will approach zero. So unless there is a restriction of maximum pipe size, the minimum velocity constraint does not in itself form a restriction on maximum pipe levels.

To summarise, the upper bound pipe level at a general point, P, is the lesser of:

(1) (ground level at that point) - (specified minimum depth of cover)

(2) (Upper bound pipe level at any point upstream) - m x (distance from that point), where m is maximum of (zero, specified minimum gradient, minimum gradient to provide specified minimum velocity with largest available pipe).

5.7.2. Lower bound on state variable

There may be a constraint on the maximum depth of cover (constraint (a) of section 2.3) which will give a lower bound on pipe level.

Whether or not this limit exists, experience of practical designs shows that it is reasonable to consider a lower bound at a fixed depth below the upper bound as determined in the previous section, giving a zone of fixed depth within which the optimal solution should lie. The selection of the correct depth to ensure optimality is a matter of judgment and experience and will be considered later (See 5.14.2).

The only other way in which the lower level could be limited is if a minimum outfall level is specified, but this would rarely form a practical limit for most of the network.

5.7.3. Establishing discrete values of level

Having specified the upper and lower limits on the state variable, pipe soffit level, at every manhole in the system, it is now necessary to define the discrete values of level that the variable may take.

To guarantee a true optimal solution, it is necessary to specify an infinite number of discrete values. However, in most cases a close approximation to the optimal solution may be obtained by adopting only a few discrete values. The choice of the number adopted is again one of judgment and experience (see 5.14.2), and is a balance between the marginal cost savings on the network designed and the extra running costs of the computer program.

The arrangement of a typical stage in the network is shown in Figure 5.5(b).

5.7.4. Establishing a feasible design for an element

Given a discrete downstream level and a discrete upstream level, the design for an element then consists of selecting the pipe diameter that will give the least construction cost for the element whilst satisfying all the constraints listed in section 2.3. In practice it is assumed that element costs increase with increased pipe diameter hence the smallest feasible diameter is chosen.

For many combinations of upstream and downstream level, there may be no feasible pipe diameter.

5.7.5. Cost at each discrete input state

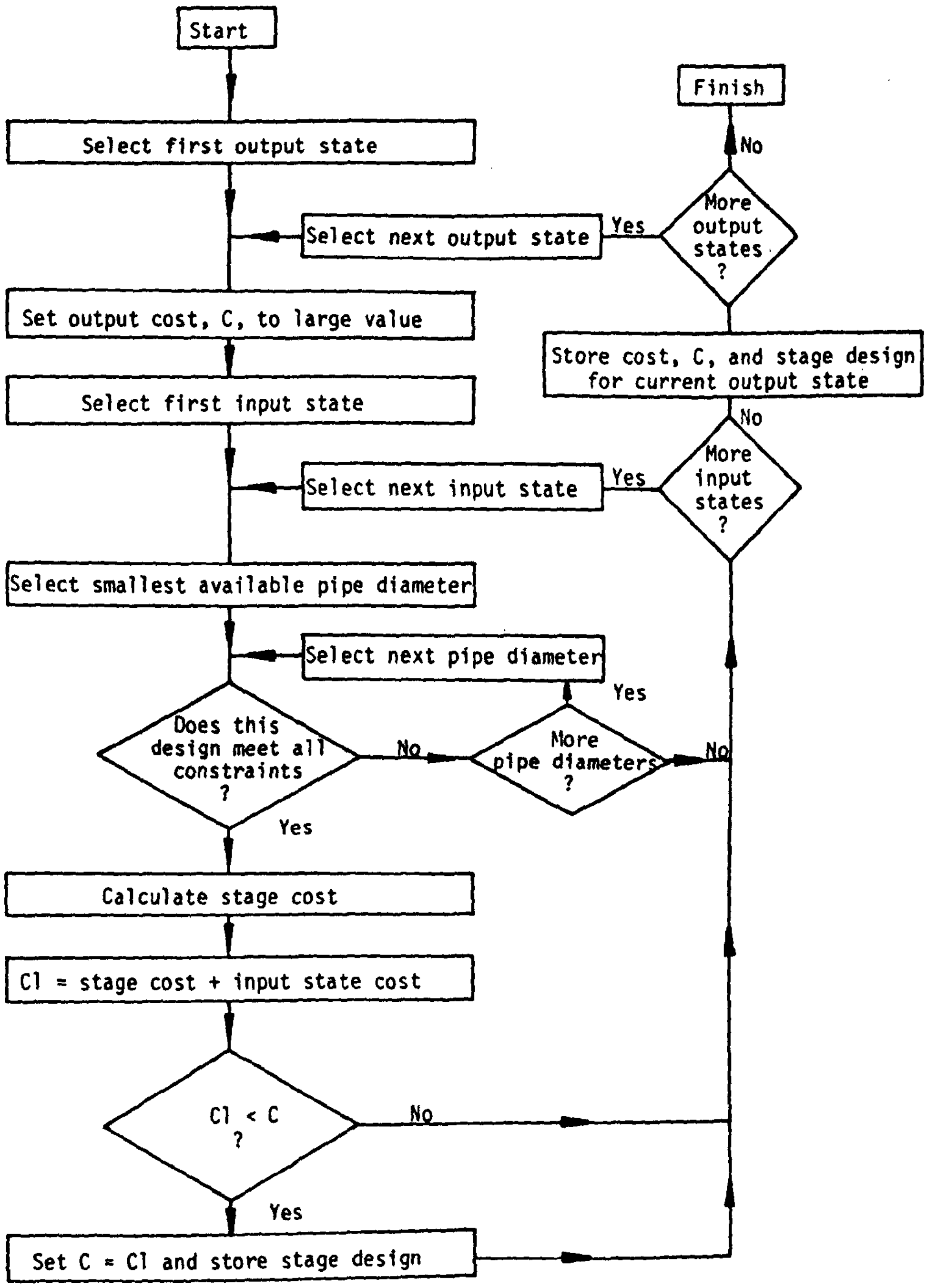
In dealing with a typical stage K, it is assumed that the minimum cost of arriving at each discrete input state is known, i.e. for each input state the optimal set of decisions for stages 1 to (K-1) and the returns (costs) resulting from them have already been determined.

5.7.6. Cost at each discrete output state

The problem now becomes that of determining the decisions in stage K that produce the minimum cost of arrival at each output state from stage K, where the minimum cost of arrival is the sum of the cost of arrival at the input to stage K plus the cost of the decisions taken in stage K to get from the input state to the particular output state.

For a particular output state, each input state is taken in turn. For that particular input state the smallest pipe is selected that will meet all the constraints listed in 2.3. The stage cost for this solution is added to the cost at the input state to give a cost at the output state.

When all input states have been examined, the overall cheapest cost of arrival at the output state is identified. This cost and the stage decisions that led to it are retained, all other costs and decisions relating to that output state being abandoned.



DESIGN OF A STAGE, FOR BASIC N-STAGE SERIAL SYSTEM

FIGURE 5.6

Hence the minimum cost of arrival at each output state is established. This is illustrated in Figure 5.6.

5.7.7. Overall Minimum Cost

It has now been shown that, given a set of minimum total costs for the input states to stage K, it is possible to obtain a set of minimum costs for the input states to stage (K+1).

As the costs for the input states to stage 1 are known, being generally zero, the process can be applied recursively along the serial system to obtain the set of minimum costs for the last (Nth) stage. This set of costs can then be examined and the cheapest will be the overall cheapest solution for the serial system.

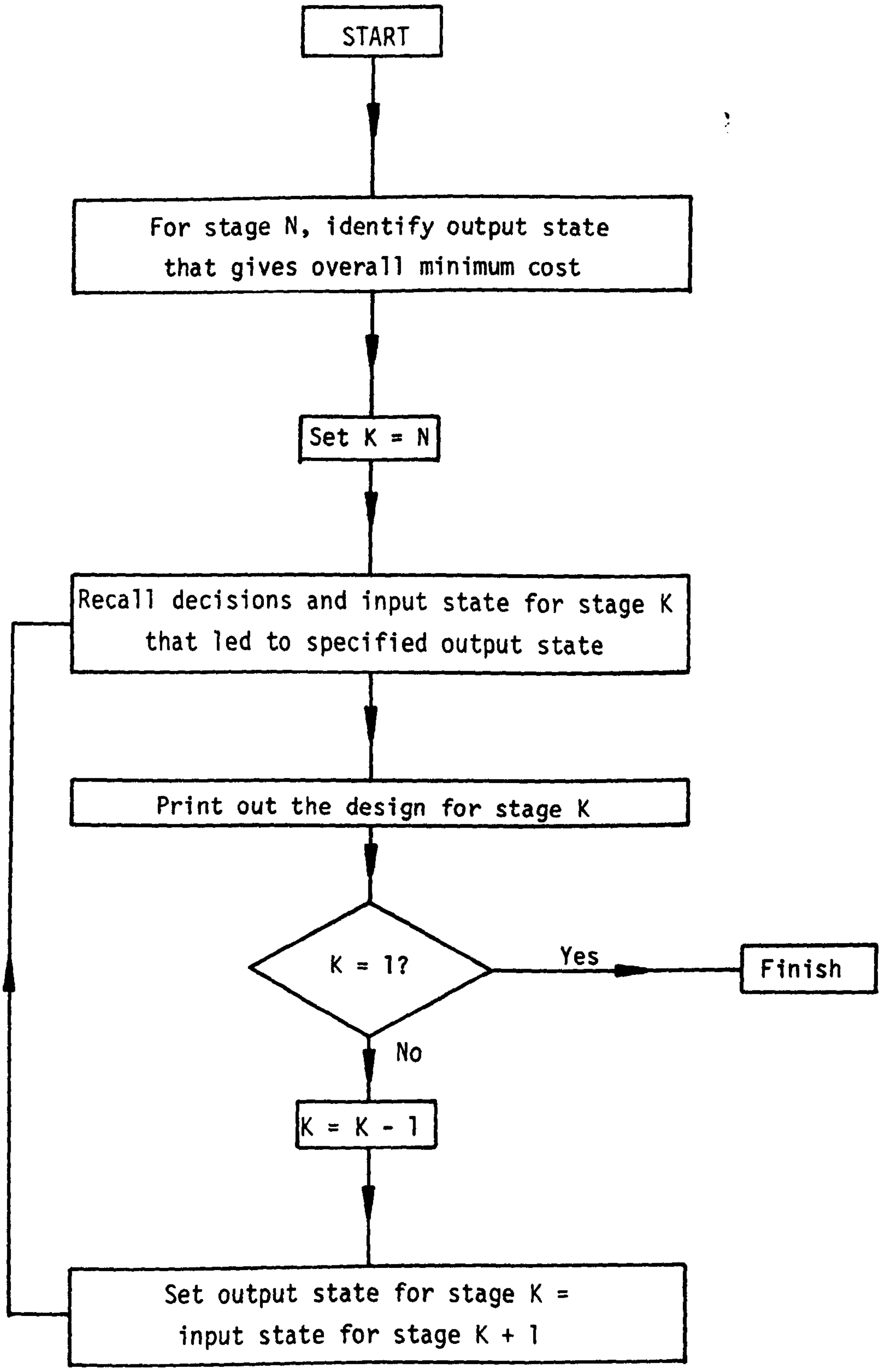
5.7.8. Optimal Solution

In itself this is of little value. It is the decisions that led to the minimum cost solution that are important, hence a trace-back as described in section 5.6.4 is performed to establish the pipe levels and diameters used. This is illustrated in Figure 5.7.

5.8. Inclusion of Drop-Manholes

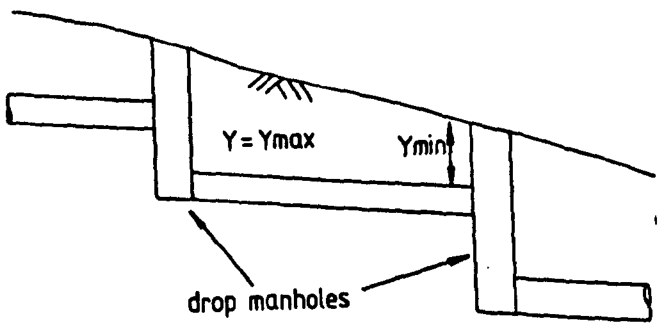
5.8.1. Introduction

A drop-manhole is one in which there is a change in level between the incoming and outgoing pipes. Such structures are required where ground levels change rapidly. Maximum slope or maximum velocity restrictions (see 2.3) may cause the outgoing pipe to be lower than the incoming for there to be any feasible solution (see Figure 5.8(a).) Alternatively if there is an obstruction it may be more economical to drop levels across a manhole (see Figure 5.8(b)). Small changes in level can normally be accommodated without incurring extra costs but large changes may well necessitate different and more expensive forms of manhole construction. Typical of these are the Back-Drop manholes described in the British Standard code of practice for Building Drainage (ref. 4).

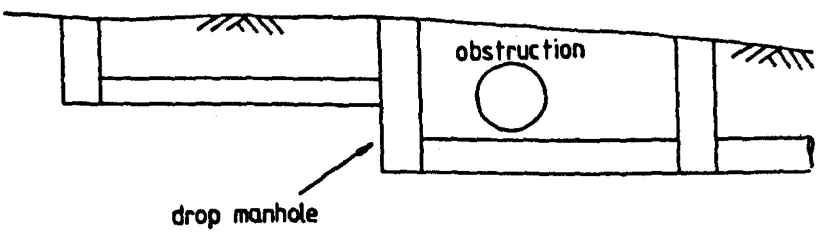


TRACING THE SOLUTION BACK THROUGH BASIC N-STAGE SERIAL SYSTEM

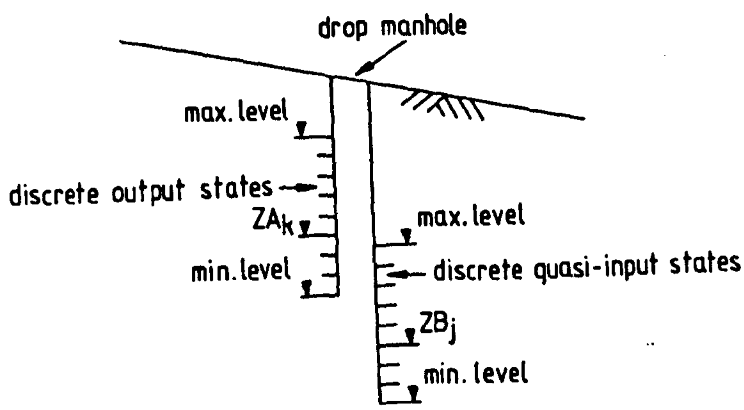
FIGURE 5.7



(a) DUE TO STEEPLY SLOPING GROUND



(b) DUE TO OBSTRUCTION



(c) QUASI-INPUT STATES

DROP MANHOLES

FIGURE 5.8

5.8.2 Defining the Quasi-Input State

The main theoretical difficulty in dealing with drops across manholes is that the input state, defined as the lowest pipe level entering the upstream manhole, is no longer the level of the outgoing pipe.

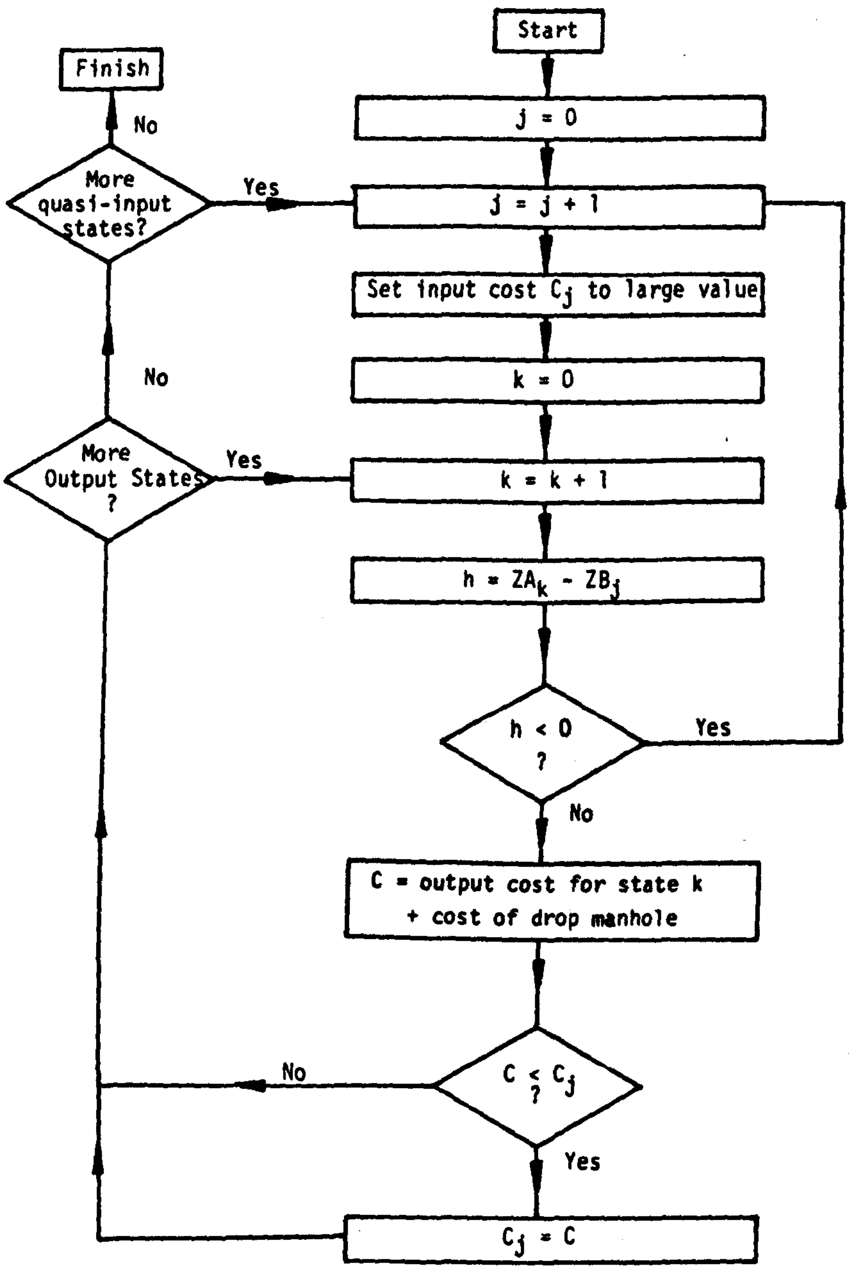
This difficulty can be overcome by one of two methods

(a) consider the manhole itself as a stage with the input state corresponding to the incoming pipe level and the output state corresponding to the outgoing pipe level, the decisions being the drop across the stage and the return being the cost of the manhole. This approach has been adopted by some previous authors (ref.23, 34) but further modifications are required when dealing with converging networks and on balance this approach was considered unnecessary.

(b) Set up a "quasi input state", corresponding to the level of the outgoing pipe, this being the pipe level at the upstream end of the stage under consideration. This approach was developed for and used in the current research. Referring to Figure 5.8(a), the maximum level of the pipe leaving a manhole is determined by the level of the ground at the downstream end of the pipe and by the maximum permissible pipe slope. It is thus sensible to use this as an upper bound limit on the quasi input state. From this upper bound, a lower bound limit can be deduced with experience. Hence m discrete values of the quasi input-state may be determined. For each of these it is necessary to know the total optimal upstream cost.

Referring to Figure 5.8(c), consider a typical quasi-input state j, level ZB_j. Any output state, k, level ZA_k from the previous stage combined with a suitable value of drop, h, could give rise to this state, provided $ZA_k \geq ZB_j$. The optimal upstream cost associated with state j is then the least of (cost to output state k + cost of drop from ZA_k to ZB_j) for all k such that $ZA_k \geq ZB_j$. This procedure is shown in the flow chart of Figure 5.9.

This procedure can be incorporated into the D.P. method already detailed to achieve the overall optimum cost for the network.



Note: j = quasi-input state number
k = output state number

ESTABLISHING QUASI-INPUT STATE COSTS, ALLOWING FOR DROP MANHOLES

FIGURE 5.9

5.8.3. Tracing Back

It now remains to ensure that one can perform a trace back up the system to obtain the set of optimal decisions that led to the overall optimum cost. This would conventionally require there to be m references stored one for each discrete input state, labelling the output state corresponding to the optimal upstream cost.

However, as a computationally easier alternative, one can omit the m references and re-establish the optimal upstream state for the single value of quasi-input state specified by the trace-back. This is a similar procedure to that described in the last part of section 5.8.2, and illustrated in Figure 5.9, except that it is only one quasi-input state that is considered.

As the trace-back is performed only once, the extra computation involved is negligible, and the savings made in data handling and storage can be significant.

5.9 Optimising a branched drainage network

5.9.1. Introduction

In 5.5.3. it was shown that a converging system such as the typical tree-like drainage network could be treated as several serial systems linked together. Hence there should be no difficulty implementing a D.P. approach to optimise such a system and this is indeed the case.

5.9.2 Procedure

For a tree-like network the order of design should be such that when a particular branch is being designed, all the branches upstream of it should already have been designed.

Take a typical branch of the network, consisting of several lengths of pipe between manholes, with several branches joining the most upstream manhole (e.g. branch AB of Figure 5.1).

Assume that for each upstream branch a set of optimal costs has been established for each discrete output state on the most downstream stage. In general the ranges of discrete output states and the range of permissible pipe levels at the upstream end of the typical branch under consideration will all be different.

Consider a set of quasi-input states for the most upstream stage of the branch. The optimal cost for each quasi-input state may now be obtained by combining the costs of the output states of all upstream branches in a suitable way, adding in the cost of a drop manhole if this is required.

A flow chart for the above process is shown in Figure 5.10.

Having established a set of costs corresponding to the quasi-input states, the design process may then proceed in the normal way, resulting in a set of costs for each output state of the most downstream stage in the branch.

5.9.3 Tracing back through a branch junction

There remains the problem of tracing back the overall optimum solution through the junction. The optimum quasi-input state will have been identified. Adopting a procedure similar to that described in 5.8.3., the optimum output state may be obtained for each branch in turn. This is illustrated in the flow chart of Figure 5.11.

5.10. Inclusion of the constraint on decreasing pipe diameters

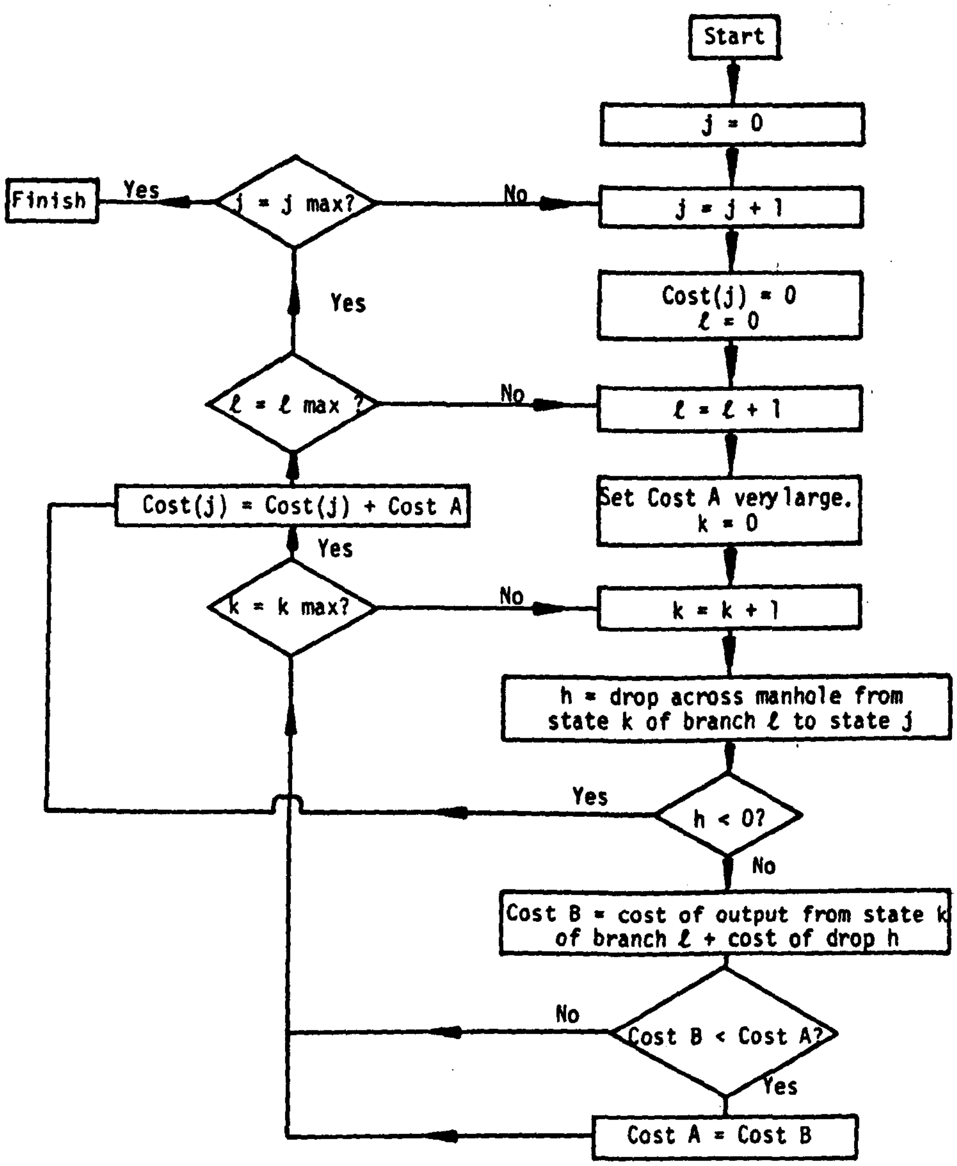
5.10.1. Introduction

A common requirement in drainage network design is that pipe diameters should never decrease in a downstream direction, i.e. the pipe leaving a manhole must be at least as big as the largest pipe entering (constraint f, section 2.3).

It was shown in section 5.5.4. that if the network is to be represented as a serial system, it becomes necessary to introduce a second state variable D, the pipe diameter.

5.10.2. Procedure

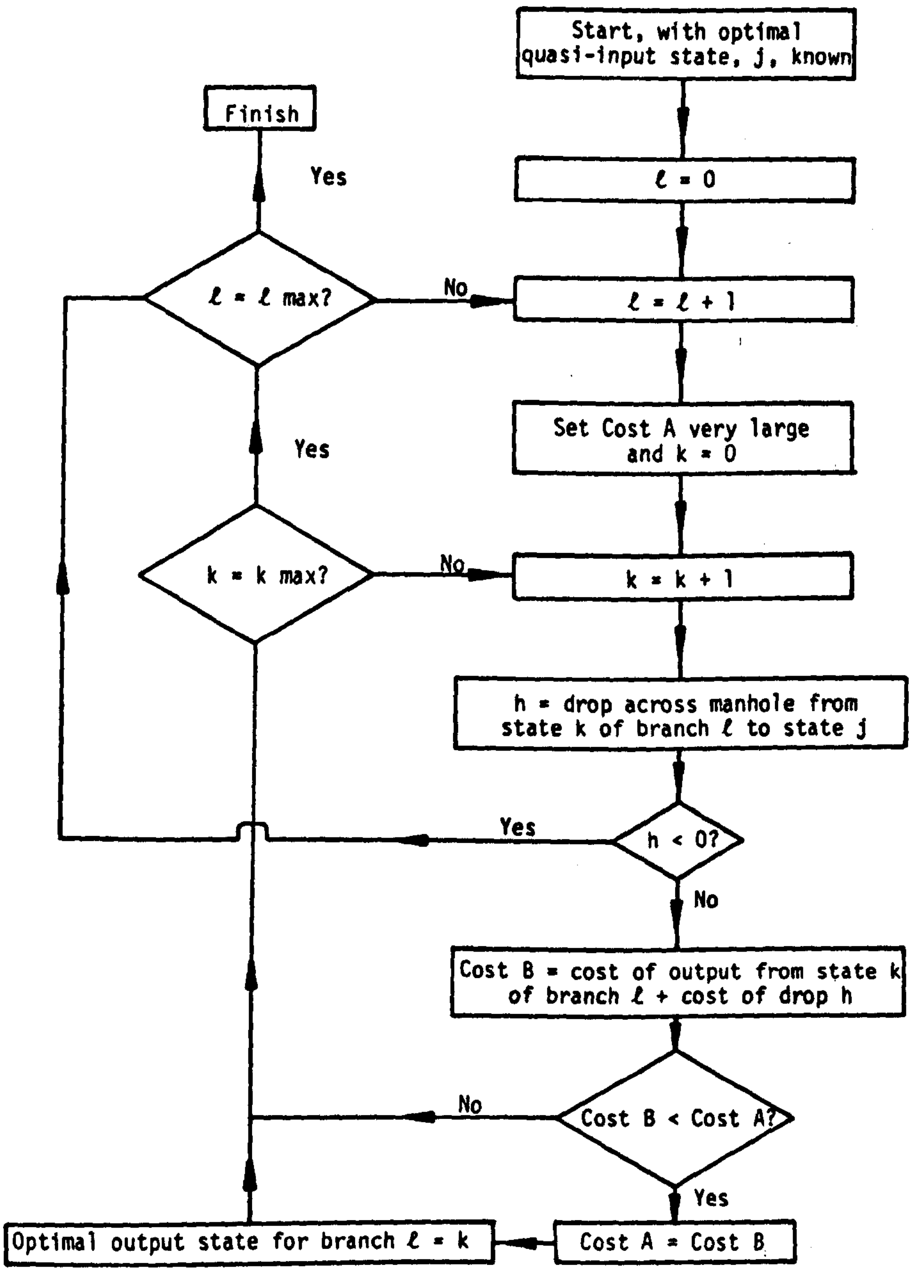
Thus the output state from a stage is a two dimensional vector (Z, D), where Z is the pipe soffit level and D is the pipe diameter.



Note j = quasi-input state number: 1,2,3,...., j max
k = output state number : 1,2,3,...., k max
ℓ = upstream branch reference:1,2,3,...., ℓ max

ESTABLISHING QUASI-INPUT STATE COSTS FOR A BRANCHED SYSTEM

FIGURE 5.10



Note k = output state number: 1,2,3,..., k max
ℓ = upstream branch reference: 1,2,3,...,ℓ max

TRACING THE OPTIMAL SOLUTION BACK THROUGH A JUNCTION

FIGURE 5.11

The input state for the next stage is then, strictly, the level and diameter (ZA, DA) of the pipe entering the upstream manhole. Where branches converge this becomes the minimum pipe level and maximum pipe diameter of all pipes entering the upstream manhole.

As in sections 5.8 and 5.9, it is convenient to define a quasi-input state, here consisting of pipe level and diameter (Z, D) of the pipe at the upstream end of the current stage, i.e. the level and diameter of the pipe on exit from the upstream manhole. It is then necessary to find for each state (Z, D) the least cost of arriving at (Z, D) from any input state (ZA, DA) such that ZA $\geq$ Z and DA $\leq$ D allowing for the cost of any drop manhole feature associated with the value of (ZA - Z).

This procedure is similar to that described in 5.8.2. and is detailed in the flow chart of Figure 5.12.

5.10.3. Defining the range of diameters and their discrete values

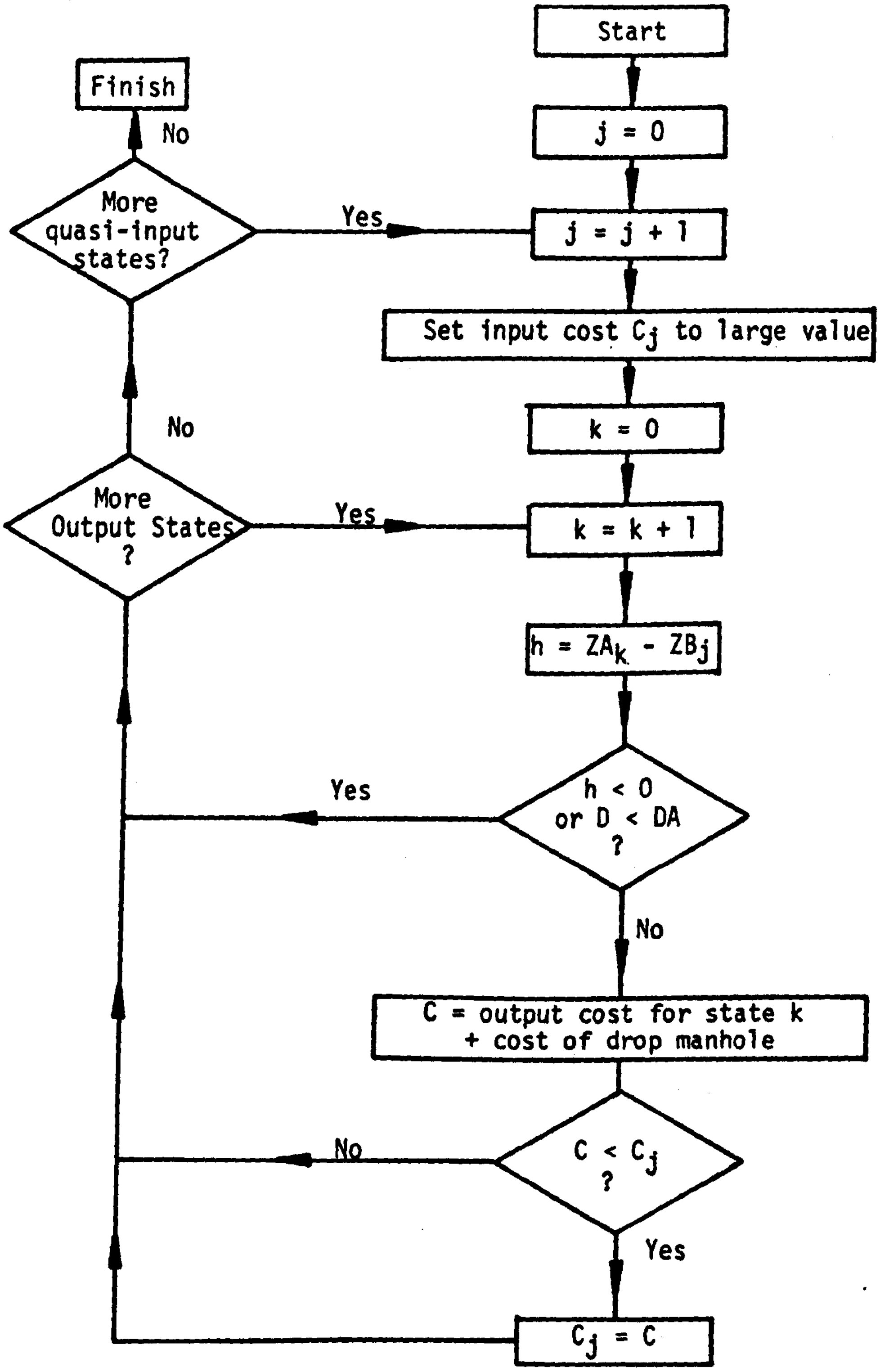
The D.P. method adopted requires that the range of values of diameter D should be defined at every stage in the system, and D should adopt discrete values at these stages.

The latter condition is automatically met by the fact that pipes are only available in discrete increments of size.(see 2.3, constraint (g)).

The actual diameters available may depend on the pipe material and manufacturer. Hence it is necessary for the designer to identify the range of pipes that are available to him, and the pipes he wishes to use on each particular length of drain.

For example, assume a particular network consists of lengths of French drain and Carrier Drains (as defined in Section 1.3). The designer may choose to use perforated clay pipes of diameters 100 and 150 mm and porous concrete pipes of diameters 228, 309 and 380 mm for the French drains, with Carrier drains of Asbestos Cement selected from the range 300mm to 600mm in increments of 75mm.

One convenient way of dealing with these pipe variations is to specify a pipe class for every length of drain in the system and separately specify the pipes that are available in each class.



Note: j = quasi-input state number

k = output state number

ESTABLISHING QUASI-INPUT STATE COSTS FOR A DIAMETER CONSTRAINED SYSTEM

FIGURE 5.12

An example of a typical pipe class is shown in Table 5.1. Note that other pipe properties can conveniently be attributed to each pipe in this way.

A TYPICAL PIPE CLASS

TABLE 5.1

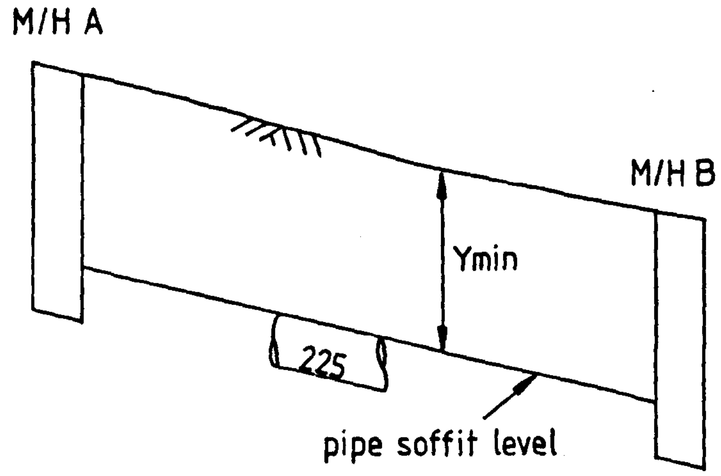
PIPE CLASS A (for French Drains)

No	Diameter (mm)	Material	Roughness (mm)	Min. Velocity (m/s)	Max. Velocity (m/s)
1	100	Perf. Clay	0.5	0.7	10.0
2	150	"	"	"	"
3	228	Porous Conc.	1.0	0.7	6.0
4	309	"	"	"	"
5	380	"	"	"	"

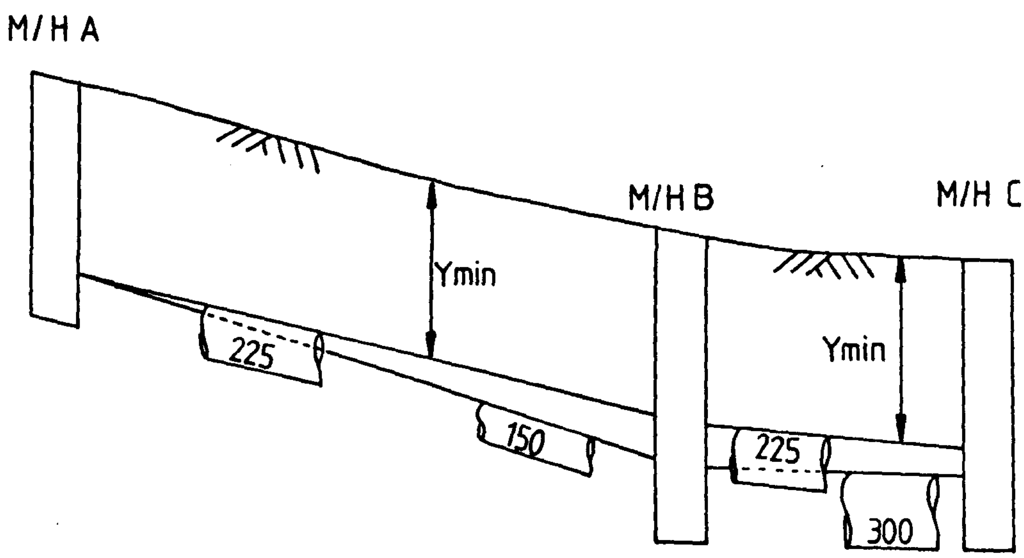
The overall available range of discrete diameters has now been established for a particular pipe length. This could therefore be used as the range of the state variable D. Such a procedure is, however, likely to be very inefficient where the pipe class contains more than a few diameters.

In some way it is necessary to establish upper and lower bounds on the size of pipe to enable realistic ranges of diameter to be taken.

Consider a single length of pipe between manholes A and B (Figure 5.13 (a)). Design the pipe first to the minimum possible depth of cover. This will involve either minimum cover or minimum gradient or minimum velocity constraints (see 2.3). Then consider any other design. This will necessarily be at or below the level of the first solution. Use of a smaller pipe diameter at a steeper slope may give a cheaper solution. However, using a larger pipe must give a more expensive solution as the extra pipe cost cannot be compensated by reducing the trench excavation. Hence the optimal solution cannot involve a pipe diameter greater than that for a minimum cover solution.



(a) MINIMUM COVER DESIGN



(b) ALTERNATIVE DESIGNS FOR TWO PIPE SYSTEM

FIGURE 5.13

It would be very convenient if such an argument could be extended to cover a network of pipes rather than just one single pipe. Unfortunately this is not theoretically justified as can be seen from Figure 5.13(b). In this case the combined costs of the two pipes AB, BC, of diameters 150mm and 300mm could be less than for the two 225mm pipes at minimum cover. However results show (see 7.10.3) that in practice the optimal solution almost always has a diameter less than or equal to the 'minimum cover' solution. This condition is likely to apply to any network for designs involving sensible methods of costing the pipe elements and for reasonable ranges of pipe diameters and hence forms a realistic method of obtaining an upper bound on the diameter.

These same results show a second important feature. This is that the optimal solution tends to be confined within one or two increments of diameter of the minimum cover solution. This gives a method of establishing a lower bound on the pipe diameter.

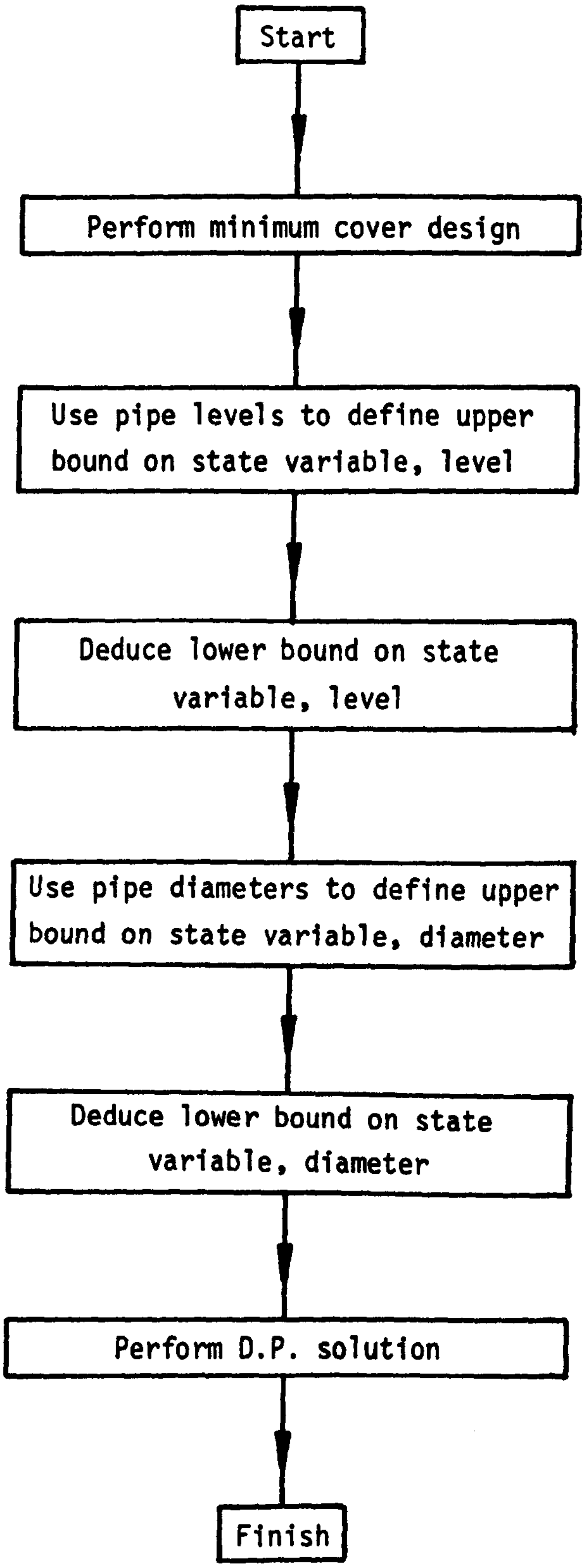
It can now be seen that bounds on both level and diameter may be achieved by performing a minimum cover design and using the levels and diameters so produced to define the upper limits on the state variables for a Dynamic Programming process. The lower limits may be taken for level as a fixed distance below the upper limit and for diameter as a fixed number of increments below the upper limit.

A flow chart to illustrate this process is given in Figure 5.14.

5.10.4. Organising the computation

The method of computation for an individual stage is similar to that for just one state variable. Essentially every output state (defined by the vector (Z, D)) is considered in turn, with each quasi-input state taken as a possible source of the optimal solution. A series of checks are made on the feasibility of this design.

If there are m discrete levels and n discrete pipe diameters, there are $m \times n$ states and hence $m^2 \times n^2$ 'designs' to consider. This is a large increase over the single state variable case. There is, however, one great computational simplification. The pipe design



ESTABLISHING BOUNDS ON D.P. PROCEDURE WITH DIAMETER CONSTRAINT

FIGURE 5.14

is now completely defined as both pipe diameter and levels are specified. Previously (see 5.7.4.) for the single state variable case it was necessary to design the element by finding the smallest suitable diameter. Using the Colebrook-White equation the diameter cannot be obtained explicitly for a given slope and discharge, hence some procedure using enumeration or iteration was necessary. That included in the flow chart of Fig. 5.6 is one simple possibility. Hence, although two state variables are used, computational effort is not greatly increased.

For each output state feasible solutions are compared in cost with the cheapest being retained. The design of a stage is summarised in the flow chart of Fig 5.15.

Tracing back the final optimal solution presents no new difficulties and is organised similarly to those processes described in 5.6.4 and 5.8.3 . A flow chart of the trace back procedure is shown in Fig. 5.16.

5.11 Dependence of flows on the network design

5.11.1 Introduction

In most methods of design for stormwater networks the design flow at a point in the system is dependent on the size, slope and roughness of some or all of the pipes upstream of that point.

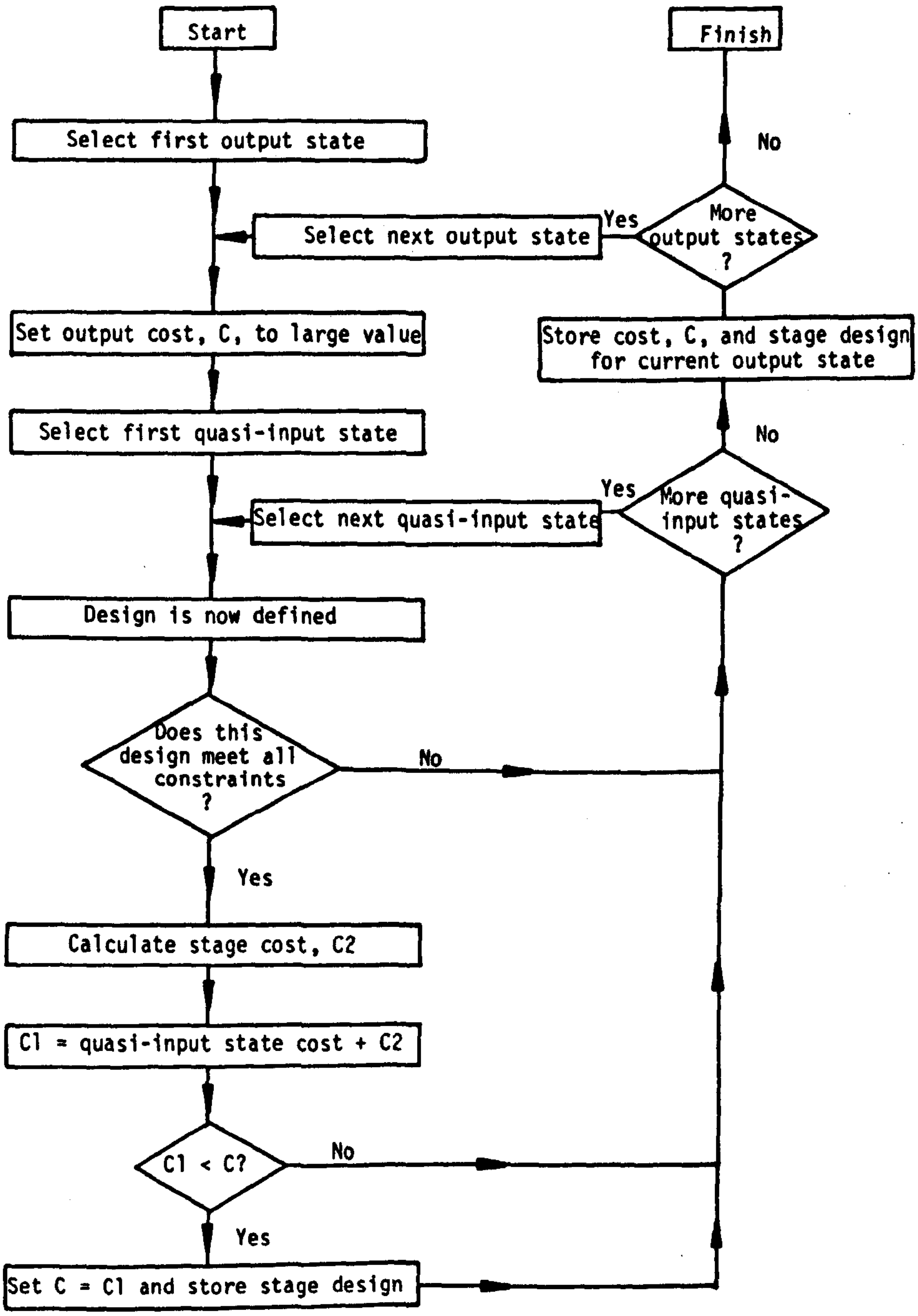
As described in section 5.5.5 this leads to a three dimensional state vector for a true serial representation and hence a rigorous D.P. approach.

5.11.2 The Rational or Lloyd-Davies method

The most common method of calculating flows for small stormwater drainage networks is the Rational or Lloyd-Davies method (ref. 5). This will be used to demonstrate the application of both the rigorous and an approximate approach to the problem of network dependent design flows.

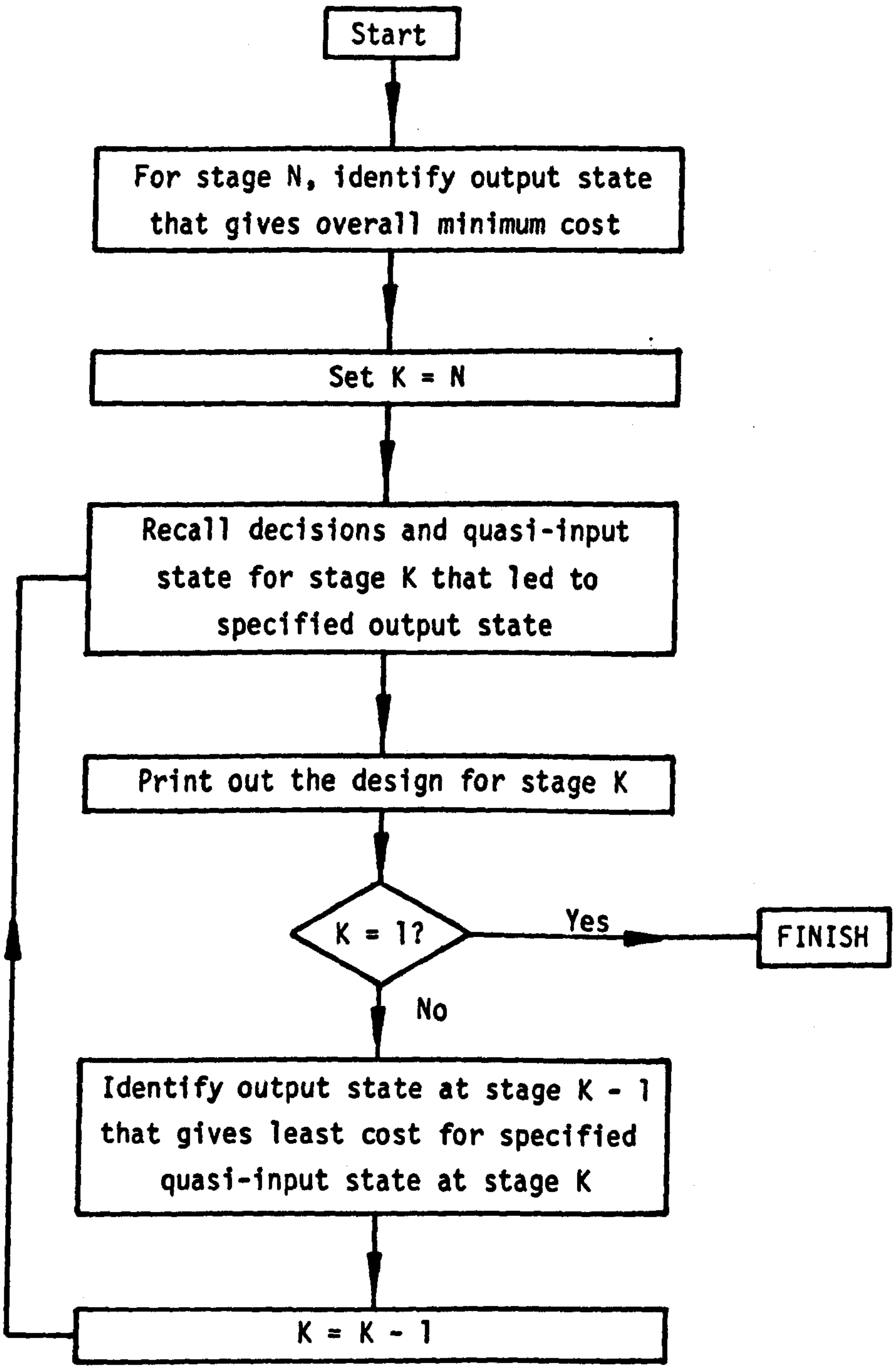
A flow chart showing the Rational method is shown in Fig. 5.17. The essential feature is that the design flow in a pipe in the network depends on the time it takes for water to flow from the most remote point upstream of that pipe to the downstream end of the pipe. This time, (the time to concentration), consists of the time it takes the water to enter the pipe network (the time of entry) plus the time taken to flow down the pipes to the downstream end of the pipe under consideration, assuming that the pipes are flowing full (the time of flow).

The Rational method design philosophy assumes that the rainfall can be treated as having a constant intensity during a storm event. Generally the shorter the length of storm the higher is the



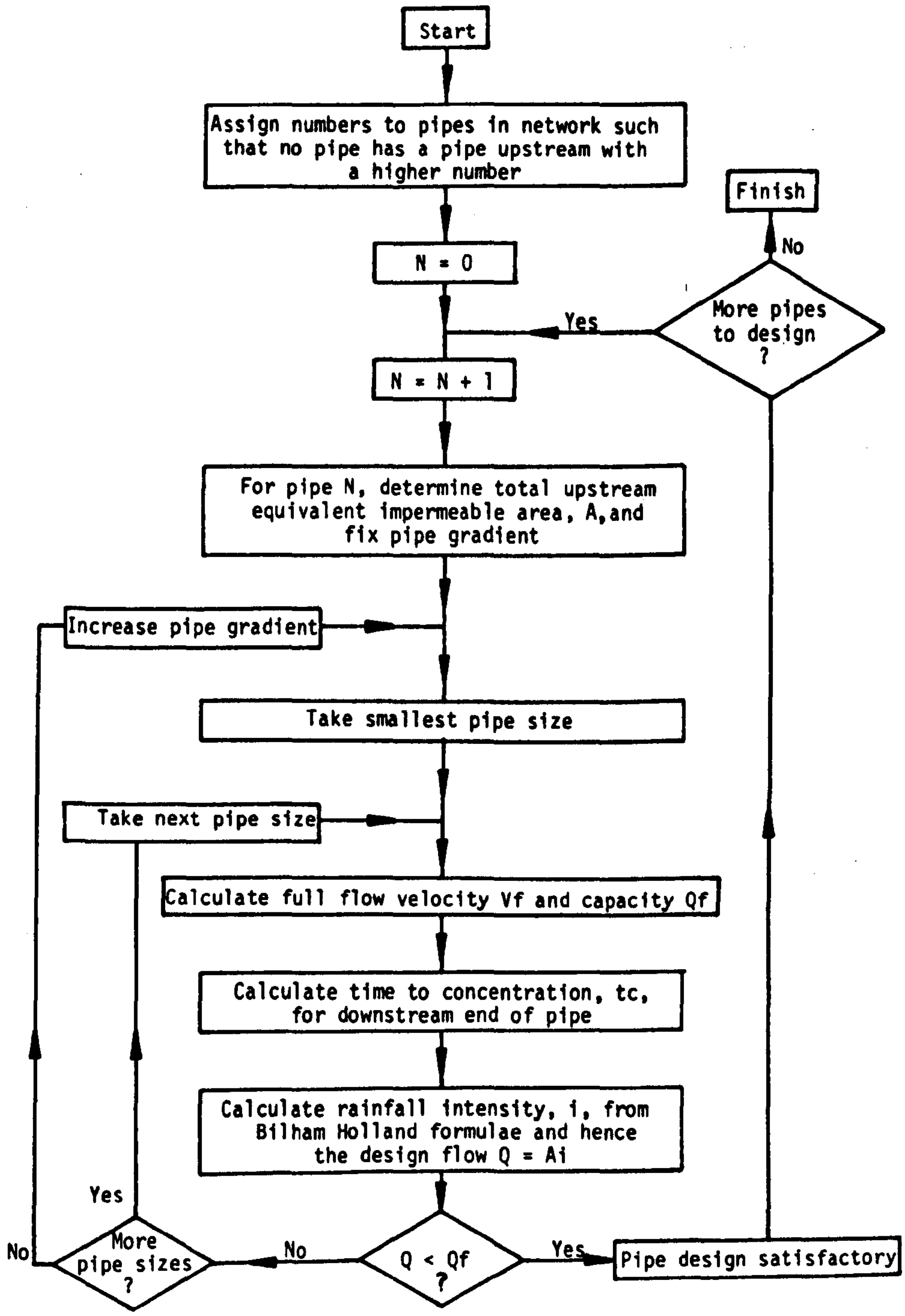
DESIGN OF A STAGE, WITH DIAMETER CONSTRAINT

FIGURE 5.15



TRACING THE SOLUTION BACK THROUGH N-STAGE SERIAL SYSTEM WITH DIAMETER CONSTRAINT

FIGURE 5.16



THE RATIONAL METHOD OF DESIGN

FIGURE 5.17

rainfall intensity. It is assumed that the rainfall is evenly distributed over the catchment area.

Consider two subcatchments 1 and 2 draining into a common pipe AB (Fig. 5.18). Consider a storm of length t such that $tc(2) < t < tc(1)$, where tc(1), tc(2) are the times to concentration at point B for flow from subcatchments 1 and 2.

As $t > tc(2)$ all of subcatchment 2 contributes to the flow at B, but as $t < tc(1)$ only part of subcatchment 1 contributes to the flow at B. It is assumed that the design flow at B increases with increasing values of t until a critical situation is met when $t = tc(1)$. At this stage both of the subcatchment areas contribute in full to the flow at B. However if t is increased beyond this, the rainfall intensity is reduced, and hence the design flow decreases.

So, in general, for a particular pipe in the network a length of storm is selected equal to the time to concentration to the downstream end of the pipe.

Statistical rainfall data has been compiled which can either be used directly for a given location in Great Britain (Ref. 5) or formulae based on this data can be used to obtain an average rainfall intensity for a given return period and length of storm, the return period being the average period of time between events that exceed the chosen event.

One common formula used in Britain is the Bilham formula with the Holland modification (Ref.5). This is given below:

<u>Bilham:</u>

$$I = \frac{60}{t}(Nt \times 202.26)^{1/3.55} - 2.54 \qquad \text{------(C)}$$

where I is rainfall intensity in mm/hr
N is return period in years
t is length of storm in minutes

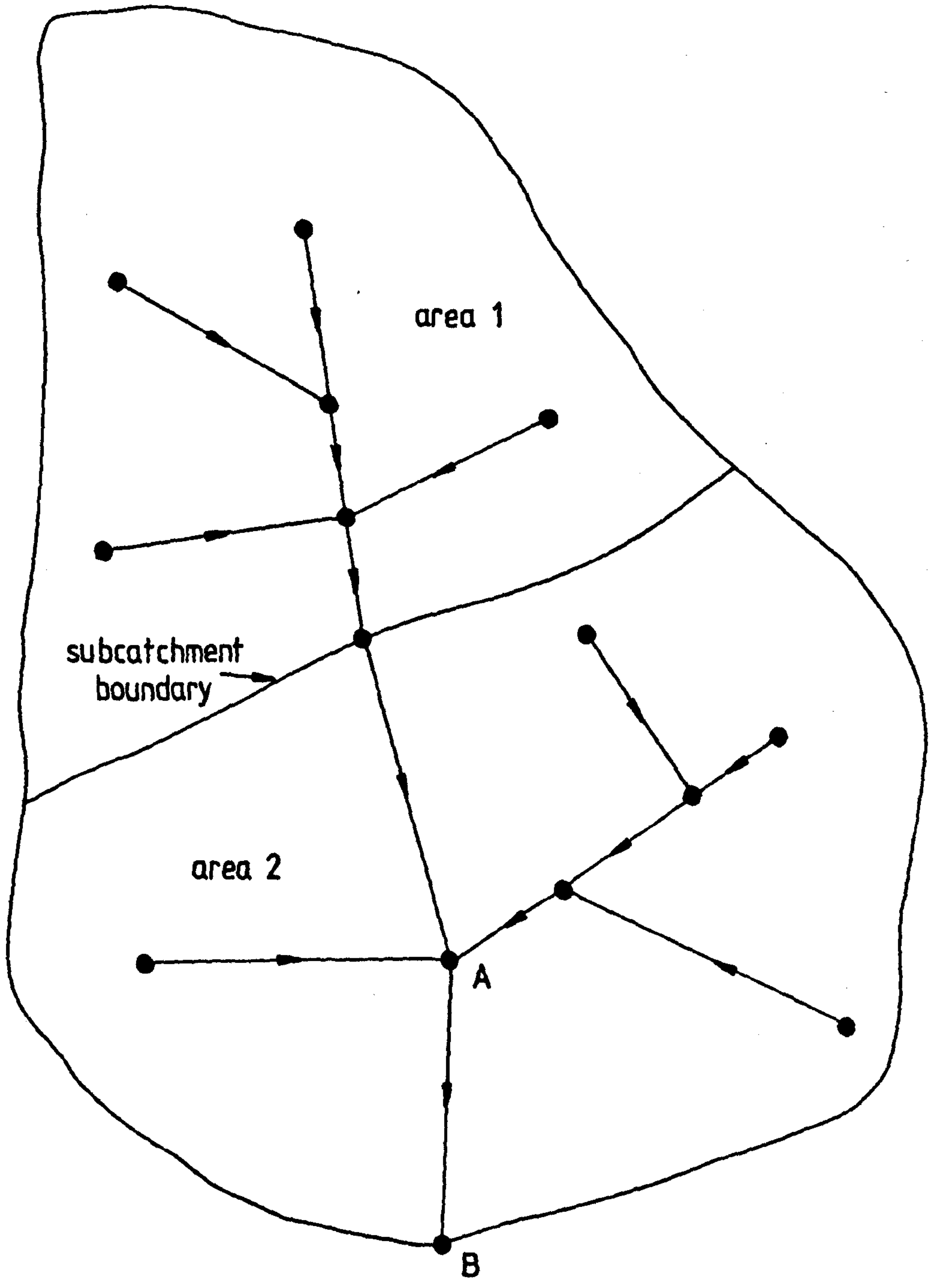
<u>Holland modification</u> : for $I > 33.0$ mm/hr

$$\mathrm{Ln}\left[\frac{15240}{NIt}\left(\frac{It}{1524}+0.1\right)^{3.55}\right] = 1 - 0.0314\,I \qquad \text{---- (D)}$$

These formulae have been used throughout this research although there is no theoretical reason why tabular data should not be used.

Examination of equations C and D show that the rainfall intensity I is only given explicitly for values of $I \leqslant 33.0$ mm/hr. The Holland modification has to be solved iteratively for values of $I > 33.0$ mm/hr with the Bilham formula giving a reasonably close initial value for the iteration.

Having obtained the rainfall intensity the design flow at a point in the network is then (the rainfall intensity) x (the catchment area upstream of that point).



SUBCATCHMENT AREAS

FIGURE 5.18

5.11.3 Other methods of calculating stormwater flows

Various other methods of calculating stormwater design flows have been proposed and used. These are conveniently summarised in Ref. 1. In all these methods the design flow at a point in the network is in some way dependent on the design of that part of the network upstream of the point.

The most prevelant of these alternative methods is the Transport and Road Research Laboratory Hydrograph method, this being widely used in the U.K. for the design of large drainage networks. A full description of the method is given elsewhere (Refs. 5, 6) but briefly it involves the use of a time varying rainfall intensity and takes into account the storage or routing effects of the pipes through which the water flows. The main effect is that an increase in pipe diameter creates greater storage within the pipe, which in turn diminishes the peak of the time varying flow out of the pipe. Hence the design flow at a point in the system is dependent on the upstream pipes, although the precise nature of the dependence is much more difficult to establish than with the Rational method. To devise a rigorous Dynamic Programming approach for such a design system would be very difficult and totally impracticable.

5.11.4 The three-dimensional state vector approach

As described in section 5.5.5, using the Rational method of design, drainage may be considered as a true serial system by using three state variables. So, in theory, a rigorous DP approach could be devised. It is , however, of interest to consider the computational effort involved in such a strategy, bearing in mind also that such a method would only be relevant to the Rational design philosophy which is likely to be superseded.

DP is generally considered to be efficient when there are one or two state variables. More state variables incur severe computational penalties as can be seen from the general approach used for all detailed computations. For a stage this consists of selecting each discrete output state and considering all possible ways of arriving at that state from each input state.

If there are ,say, i(n) discrete values of the n dimensions defining a state then there are $(i(1) \times i(2) \times i(3) \times \ldots\ldots i(n))^2$ designs to consider for each stage with $(i(1) \times i(2) \times \ldots\ldots i(n))$ values of cost and an equal number of trace-back references to store.

For example, if $i(1) = i(2) = \ldots\ldots i(n) = 10$

for $n = 1$ number of designs = 100

for $n = 2$ number of designs = 10 000

for $n = 3$ number of designs = 1 000 000

for $n = 4$ number of designs = 100 000 000

Experience has shown that values of $i(1)$ and $i(2)$ could be reduced to 7 and 3 respectively for level and diameter state variables. Using these values the number of designs for incorporation of a third state variable becomes $7^2 \times 3^2 \times (i(3))^2 = 441(i(3))^2$. A suitable value of $i(3)$ is unlikely to be less than 7. This would give at least $441 \times 7^2 = 21\ 609$ designs per stage.

Although a method using 20 000 designs per stage may be possible for the fixed plan problem, it is certainly not desirable, and as a basis for a variable plan model it can quickly be set aside as impracticable.

5.11.5 An approximate approach

Having shown that a completely rigorous approach is impracticable, it is now necessary to examine the practicability of a reasonable but non-rigorous method.

The development of such a method was actually performed for a variable plan model of which the fixed plan model under discussion is a special case. The details of the development will thus be presented in Chapter 7, "The variable manhole position model". For completeness, however, a summary of the development is given here, as far as it is applicable to the simpler fixed plan network.

The first step taken was to assume that all flows were fixed, ie. did not depend on the pipe network upstream. The method of fixing the flows was less obvious.

The initial approach was to calculate a time of flow for all pipes using a uniform flow velocity (eg. 1.5 m/s). Hence a rainfall and design flow could be calculated for each pipe. When the DP design was complete a comparison of the actual flow velocities and hence actual flows could be made with the assumed values.

It was soon seen that this led to unacceptably large discrepancies. However an iterative approach based on this was a logical development. The flow velocities resulting from the new design were thus used to calculate new times to concentration and design flows for the next DP design. The iterations continued until the variation in flow from one iteration to the next was within acceptable limits. This usually occured within four or five iterations.

The method was seen to be rather clumsy and somewhat prone to problems of convergence (see 7.9.2).

A more satisfactory approach was to define the initial design flows as being equal to the design flows for a minimum cover design. Very rapid convergence then ensued. As a minimum cover design was already used to establish the limits on pipe level and diameter (see 5.10.3), the minimum cover design flows were readily available.

It was additionally found that in general the diameters of pipes designed by the first DP design did not subsequently change in further iterations. The pipe slopes merely altered to accomodate changes in design flows.

5.12 The final fixed plan model - MANFIX

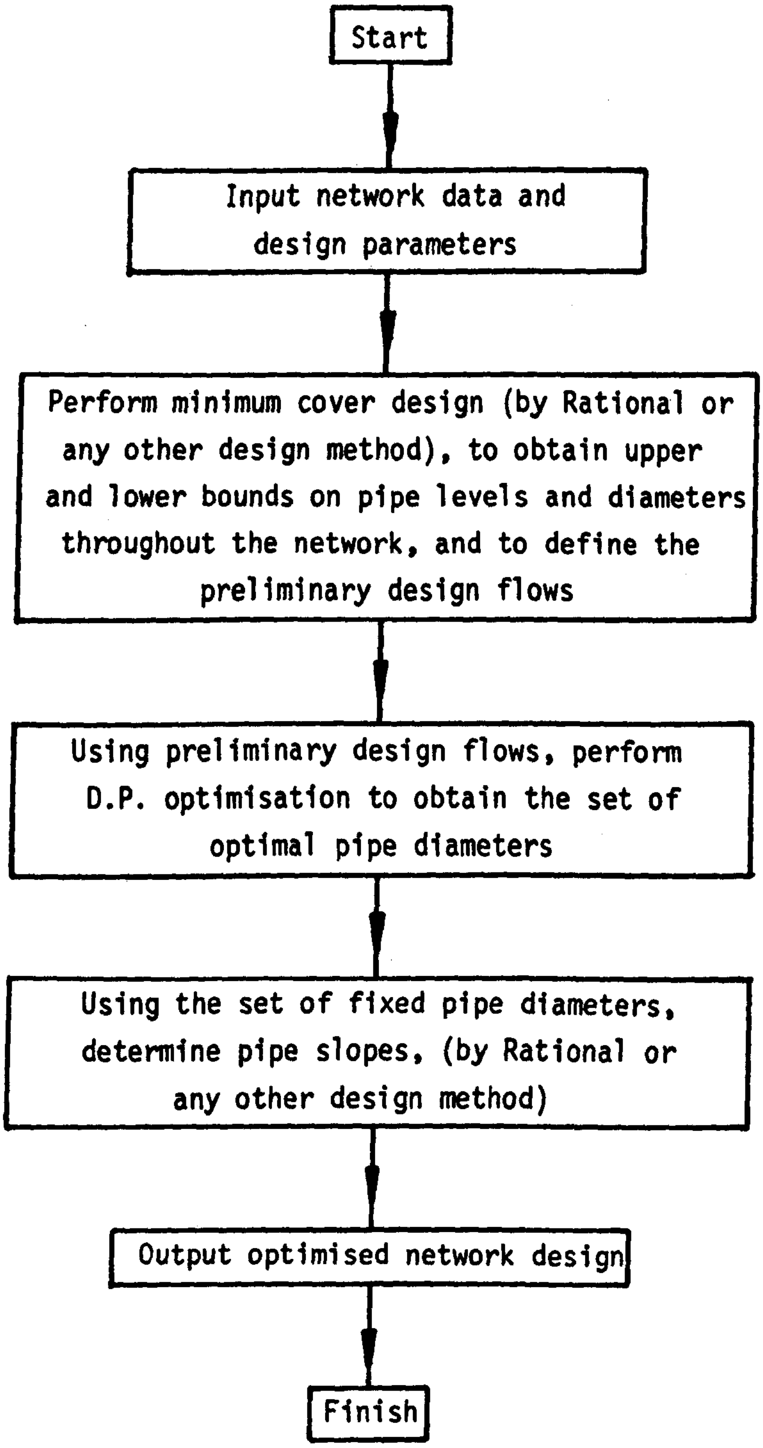
The observation that the designed pipe diameters did not change after the first iteration provided a very useful method of truncating the iterative procedure.

Instead of allowing the iteration to proceed until flows were acceptably stable, an exact explicit solution could be obtained by taking the diameters produced by the first iteration as fixed and then performing a normal Rational method design to obtain pipe slopes, and incidentally design flows.

Although a separate computer program was never written for the fixed plan model, results using the variable plan program on fixed plan examples showed the method to be sound.

For convenience the fixed plan model will be refered to as the MANFIX (manholes fixed) model. For completeness a flow chart for a proposed MANFIX computer program is shown in Fig. 5.19.

One very important feature of MANFIX is that it is not necessarily restricted to the use of the Rational method of design. In principle any design method could be used to establish design flows and bounds on the state variables based on a minimum cover condition. These flows and bounds can then be used in the core of the program - the DP process - to determine pipe diameters only. These diameters canthen be used in the selected design method to determine pipe slopes.



PROPOSED MANFIX COMPUTER PROGRAM

FIGURE 5.19

5.13 The use of Discrete Differential Dynamic Programming

DDDP (ref. 32) was refered to in Section 5.1 as being the most economic existing approach for the fixed plan optimisation problem. As such it is worthwhile discussing the method and comparing it with the MANFIX model.

Stormwater drainage using DDDP has been described in some detail elsewhere (refs. 23,34). Hence only the principles will be presented here.

A simple drainage run between three manholes is illustrated in Fig. 5.20. The DDDP approach to optimising the design of such a run is as follows :

(a) Specify a "trial trajectory", this being an initial guess at the longitudinal profile of the pipeline between the manholes.

(b) Specify an initial "band width", this being the width of a "corridor" centred on the trial trajectory, giving the limits within which the pipe profile may lie.

(c) Select a small number (3 or 5) of discrete depths at each manhole equally spaced across the corridor.

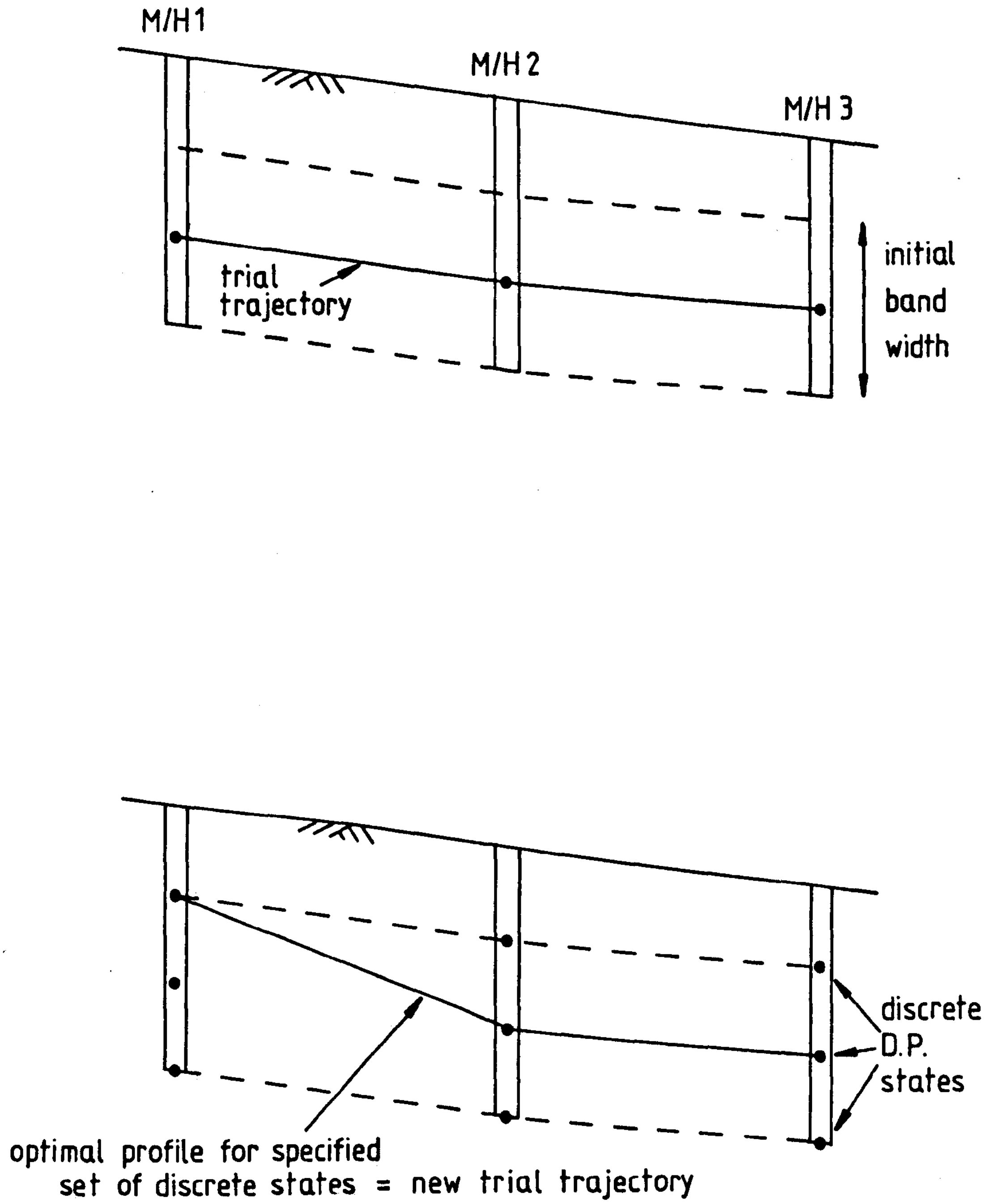
(d) Use conventional DP to select the optimum profile using the discrete depths.

(e) Use the optimum profile as the trial trajectory for another iteration using the same band width. This forms the primary iteration.

(f) When the optimum profile coincides with the trial trajectory, decrease the band width and repeat the process. This is the secondary iteration.

(g) Continue decreasing the band width until the required accuracy is obtained.

DDDP is claimed to be much more efficient computationally than DP (ref 40), due to the small number of discrete levels considered in the corridor for any one iteration. Hence if the possible range of levels at the manholes is large the potential saving over DP could be remarkable. For example, if the pipe levels were required to an accuracy of 0.01m and there was a possible range of levels of , say, 3m at each manhole, a conventional DP approach would require 300 discrete levels at a manhole, with $300^2 = 90\ 000$ possible designs to consider at each stage. Using 3 discrete levels in DDDP there are $3^2 = 9$ designs per stage per iteration. If it takes 3 primary iterations to achieve a stable trajectory, there are then $3 \times 9 = 27$ designs per secondary iteration. To reduce a 3m band width to 0.01m with a reduction by, say, a factor of 0.7 at each secondary iteration,



OPTIMAL DESIGN BY D.D.D.P.

FIGURE 5.20

requires n secondary iterations , where $3 \times 0.7^n = 0.01$. Hence n = 16, and the total number of designs per stage = 16 x 27 = 432, compared to 90 000 for a conventional DP design.

However, it has been shown that by using MANFIX the possible range of levels can be reduced substantially by first performing a minimum cover design. Further, the required accuracy for pipe levels can be very coarse, with only the optimal pipe diameters being required, the pipe gradients being obtained from a conventional design procedure using the optimal diameters.

Hence MANFIX will be more comparable to the DDDP approach than will conventional DP methods.

For example with MANFIX, if the range of levels is restricted to 0.9m and the levels are required to 0.1m, there are 10 discrete levels and $10^2 = 100$ designs per stage.

Similar values for a DDDP approach give $3^2 \times 3 \times n$ designs per stage where $1 \times 0.7^n = 0.1$, giving n = 6 and number of designs = 162.

It would appear that a DDDP approach is possibly less efficient than a carefully prepared DP approach such as MANFIX, though it may be more likely to find a true optimal solution in unusual circumstances.

One disadvantage with DDDP is that a computer program is necessarily more complex than for DP. For the purpose of the present research the main disadvantage is that the concepts of a trial trajectory and a decreasing band width are incompatible with the variable plan problem.

An additional consideration is that storm drainage design by DDDP has only been presented using a one dimensional state vector. Hence the constraint on non-decreasing pipe diameters (see 2.3), if required, is handled incorrectly in published material (ref. 31). It would be possible to have a DDDP approach to storm drainage using a two dimensional state vector, but this has not yet been tried. Also, for a complete approach, some method of approximating stormwater design flows is needed, perhaps similar to that in MANFIX.

5.14 Experience and results

5.14.1 Introduction

As the main research was concerned with variable plan networks a computer program for MANFIX was not written. Hence experience relates to the development and use of the variable plan programs and will be presented fully in Chapter 7. Results are those using variable plan programs for fixed plan examples.

5.14.2 Experience of the model

The following general comments can be made :

(a) DP optimised solutions generally followed closely to minimum cover solutions and were often identical for many of the upper branches in a network. Where they differed it was rarely by more than 0.5m in depth or by more than two increments in pipe diameter.

(b) a solution with a cost close to the true minimum could be obtained by using a coarse DP grid with discrete levels at spacings of between 0.1 and 0.15m.

(c) for standard pipe diameter increments (75mm) and a sensible minimum gradient (1 in 250), near optimal solutions could be expected with confidence using six discrete levels over a range of 0.7m and considering just three diameters for each pipe.

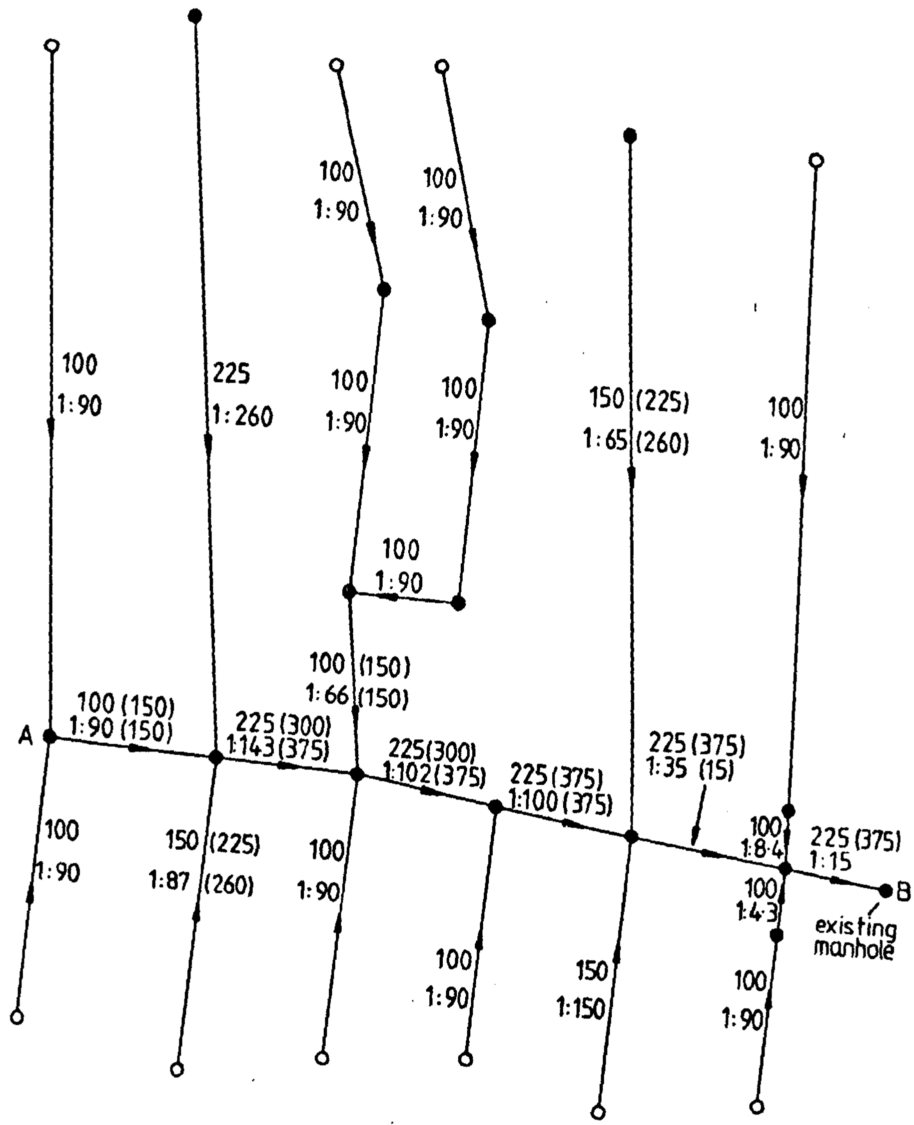
5.14.3 Results

All the optimally designed networks showed cost savings of between 5% and 15% over networks designed to a minimum cover solution. This is consistent with the findings of other authors. (refs. 27, 28, 29)

A set of results for the stormwater drainage of a small housing estate is given in Figs. 5.21, 5.22 and 5.23. The original design against which the optimal design is compared does contain some inconsistencies and cannot be regarded as a perfect minimum cover design. It is however considered typical of present practice. Qualitatively the results show a feature typical of optimised designs compared to traditional designs. The pipe diameters are unaltered at the upstream ends, the main savings being on reduced diameters towards the outfalls. Where depths of cover are increased it is only necessary to do so very slightly to accommodate the increase in gradients.

In fact at both outfalls into existing sewers the optimised networks give invert levels slightly higher than the original scheme, due to smaller pipes being required at the same or slightly increased depths of cover.

From a practical viewpoint the optimal scheme is preferable in that less of the pipe network is at minimum gradient. Hence there would be less trouble from siltation and blockage. Increases in depth of cover tend to be minimal and hence would not add significantly to access problems in cases of failure or new connections.



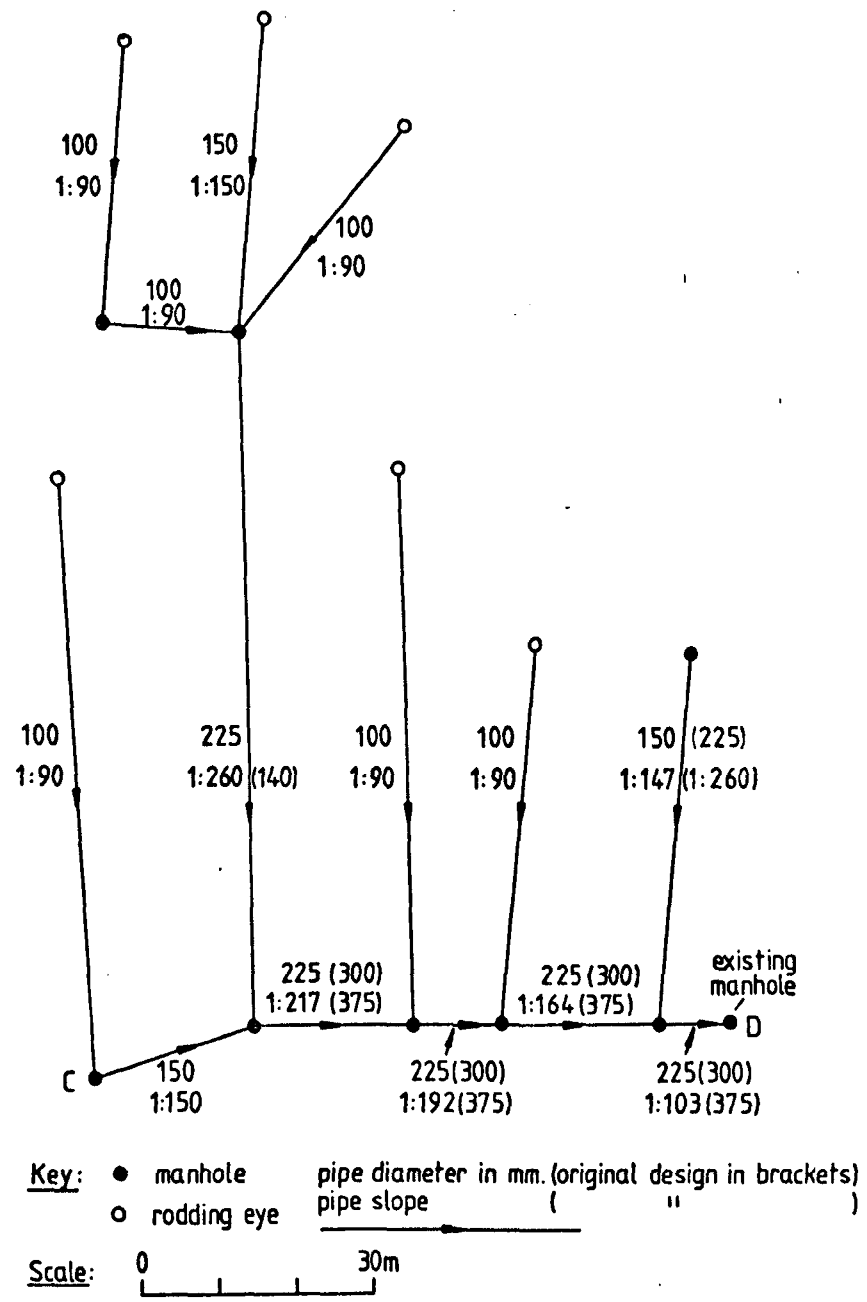
Key: ● manhole
○ rodding eye

pipe diameter in mm. (original design in brackets)
pipe slope (")

Scale: 0 30m.

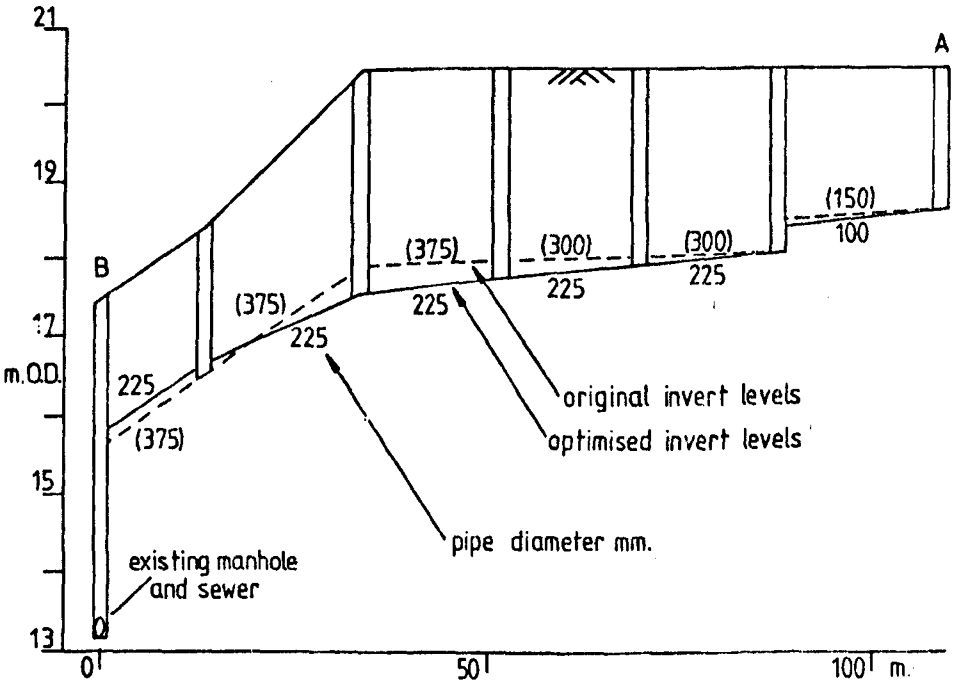
OPTIMISED DESIGN FOR HOUSING ESTATE AT PETERBOROUGH (AREA A)

FIGURE 5.21

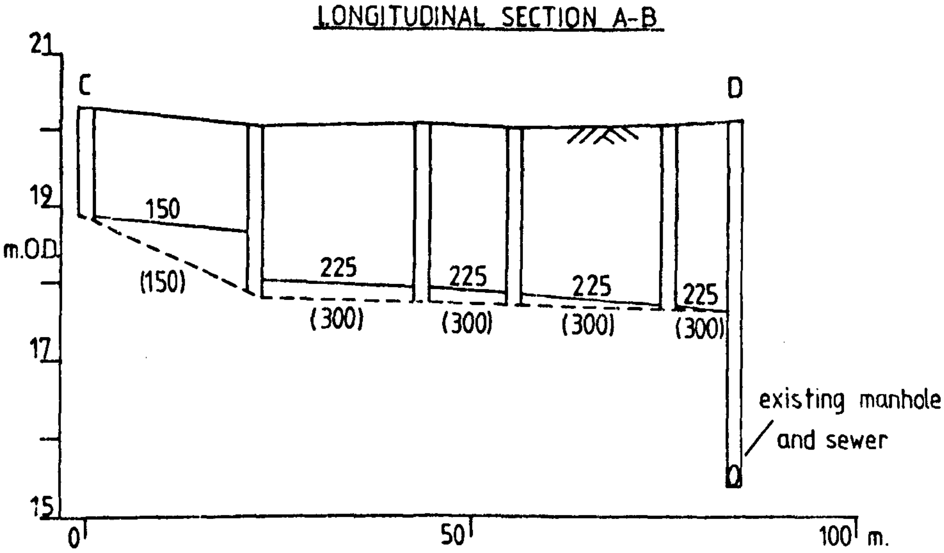


OPTIMISED DESIGN FOR HOUSING ESTATE AT PETERBOROUGH (AREA B)

FIGURE 5.22



LONGITUDINAL SECTION A-B



LONGITUDINAL SECTION C-D

OPTIMISED DESIGN FOR HOUSING ESTATE AT PETERBOROUGH
SECTIONS

FIGURE 5.23

Quantitatively the DP optimised scheme represents a saving in construction cost of about 12% over the original scheme.

5.15 Cost of using MANFIX

Clearly the extra computing costs involved in using an optimisation program should not exceed the likely savings on construction costs. If all designs led to actual constructed schemes it would be reasonable to allow the cost of optimising to approach the likely savings over a non-optimised scheme. However, for a number of reasons this may not be acceptable. These include :

(a) Allowance for non-productive and superseded design runs: the cost of these runs must be paid for from the savings on productive runs.

(b) High computer costs imply a requirement for large computers or long run times. If these facilities are not immediately available designs may be delayed for up to several days by awaiting turnround on multi-user machines.

It is difficult to estimate the actual cost of using MANFIX as present results were obtained from a variable plan program. However, as an example, the resources used in the design of the housing estate networks were 22 secs. execution time on an ICL 1906S computer with a core store requirement of 54K. In 1979 this cost about £3. This is insignificant compared to the saving on construction cost of the scheme, which is calculated as £2300 at March 1977 prices.

It is clear from this that the computer costs can be a small fraction of the likely savings. Indeed performing a design using an optimising computer program may well be cheaper than manually designing the scheme, which is the usual design office practice.

5.16 Conclusions on the fixed layout model - MANFIX

It has been shown that an effective and simple Dynamic Programming model can be used for the optimised design of stormwater drainage networks. The efficiency of such a model is likely to be greater than that of a DDDP model or any other existing fixed plan optimising model.

The novel features of the model include the following:

(a) Establishing upper bounds on levels and diameters, and an estimate of design flows, by performing a conventional minimum cover design before the optimising process.

(b) Limiting the DP to a coarse grid over a narrow band, and using only the pipe diameters thus obtained.

(c) Using a second state variable, pipe diameter, to handle the constraint on non-decreasing pipe diameters rigorously.

(d) Performing a final conventional design using the pipe diameters obtained from the DP optimisation, thus producing the correct design flows and pipe gradients.

(e) The ability to use, in theory, any design method (Rational, TRRL, etc), since the core of the model, the DP program, is unaffected by the design method.

The cost of design using such a model need be little more and may in fact be less than current design costs.

MANFIX requires only limited computer resources and could be tailored for use on a mini-computer within a design office as well as being a fully supported design program on a main-frame computer.

The type of design produced by MANFIX is in general sensible and preferable in at least one respect other than cost to the minimum cover design often produced manually. Minimum cover designs very often have most of the network at minimum gradients or flow velocities. MANFIX produces designs with more of the network at steeper slopes, thus reducing possible trouble with siltation and blockages. Some trench depths are increased but the increases are often minimal and very rarely exceed 0.5m. Indeed due to reduced pipe diameters at the downstream end of the network, trench depths can often be decreased.

CHAPTER 6

VARIABLE PLAN OPTIMISATION

6.1 Introduction

6.2 Highway Storm Drainage Networks

6.3 Potential for Optimisation

CHAPTER 6 Variable Plan Optimisation

6.1 Introduction

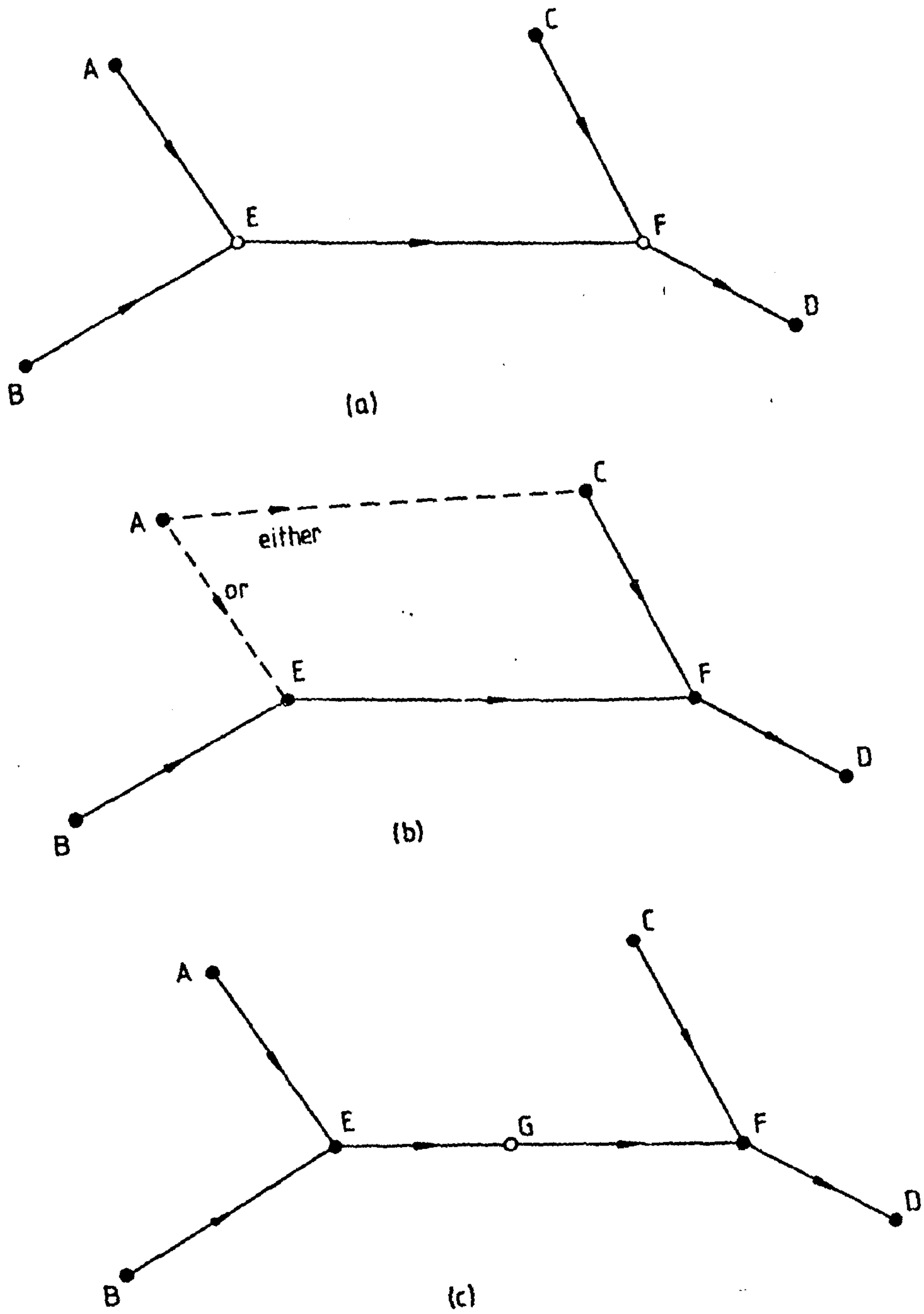
The main object of this research was to investigate the possibility of optimising the plan layout of storm water drainage networks for roads and to produce a practical working computer program for optimal drainage design if this was indeed feasible.

In a general storm drainage network, plan optimisation could be performed in many different modes. The simplest mode can be defined as follows: given a network of pipes and manholes some of which are fixed in position, find the optimum position of all other manholes together with the optimum gradients and diameters for all pipes. Such a network is shown in figure 6.1a. Some of the manholes are fixed in position (i.e. A, B, C, D), others are variable (i.e. E, F). For this example the problem is then to find the plan coordinates of manholes E and F together with the slopes and diameters of all pipes.

A second mode of optimisation is where the connectivity, or in other words the basic network layout, is unspecified. In figure 6.1b, for example, the flow from A may go either to E or to C, the choice being part of the optimisation.

A third mode of optimisation is where the number of manholes is unspecified. For example in figure 6.1c variable position manhole G may or may not exist in the optimal solution.

As a general case could combine all three such modes of optimisation it is clear that for anything but very small networks the complexities of general variable plan optimisation become formidable. Any attempt to create a general variable plan optimisation model would at present be completely futile.



VARIABLE PLAN MODES

FIGURE 6.1

The procedure adopted was therefore to examine the characteristics of several particular types of drainage network and to examine ways in which variable plan optimisation might be achieved.

As the research was primarily concerned with road drainage, the type of network normally designed for new major roads was initially considered, and is dealt with in the following two chapters. In addition one further type of network is specified and examined in chapter 9. This is the case of joining several sources of flow to a single main drainage run.

For the road drainage type of network two variable plan models were developed, MANVAR (variable manholes) and CROSSVAR (variable cross-drains), computer programs for these being written and tested. A model for the final type of network considered is proposed but a program has not been written or tested.

A fully documented and tested commercial version of MANVAR has been written for the Highway Engineering Computer Branch of the Department of the Environment, and will be released soon as an optional mode of operation for their current Drainage Design Program DAPHNE (ref 56).

6.2 Highway Storm Drainage Networks

An essential element in most modern highway design is the provision of a drainage system to remove incident rainfall. The road profile is used to direct run-off to the road edges or possibly to the central reservation in the case of a dual carriageway. Occasionally, especially on minor roads in rural areas, the designer will allow run-off to pass over a grass verge and into an open ditch. In general, however, piped drains are provided running roughly parallel to the road edges. The run-off may flow straight into these as in the case of a "French drain", this being a gravel filled trench with a perforated or porous pipe to collect the water at the bottom of the drain. Alternatively water may first be collected by open channels formed by a kerb and the crossfall on the road, thence passing through gullies sited in the channel into conventional closed pipes.

Generally kerbs and gullies are used throughout urban areas, and in rural areas where the road is on an embankment.

French drains are used in cuttings in rural areas and also along the central reservation of motorways and dual carriageway roads. They often have the additional duty of keeping the road foundation drained. This purpose is however ignored in this research as the flows involved are minimal compared with stormwater flows.

For convenience French drains and "gully -fed" drains will be referred to as "carriageway drains", as their primary duty is to collect the run-off from carriageways. Carriageway drains are generally either laid at a constant offset from the road centreline, thus being curved in plan where the road is curved, or laid in straight lines between manholes which are at a uniform offset from the road centreline.

At intervals it is generally necessary to convey the water from carriageway drains across the road. This is done by the provision of drains consisting of conventional closed pipes in trenches that are very carefully backfilled and compacted and almost always run directly across the carriageways. These will be referred to as "cross-drains".

Cross-drains, as well as being constructed to a higher specification, are often designed for more severe storm events than the rest of the drainage network. This is a sensible precaution as access in the event of failure is very expensive and overloading in a severe storm could lead to dangerous flooding of the road carriageways.

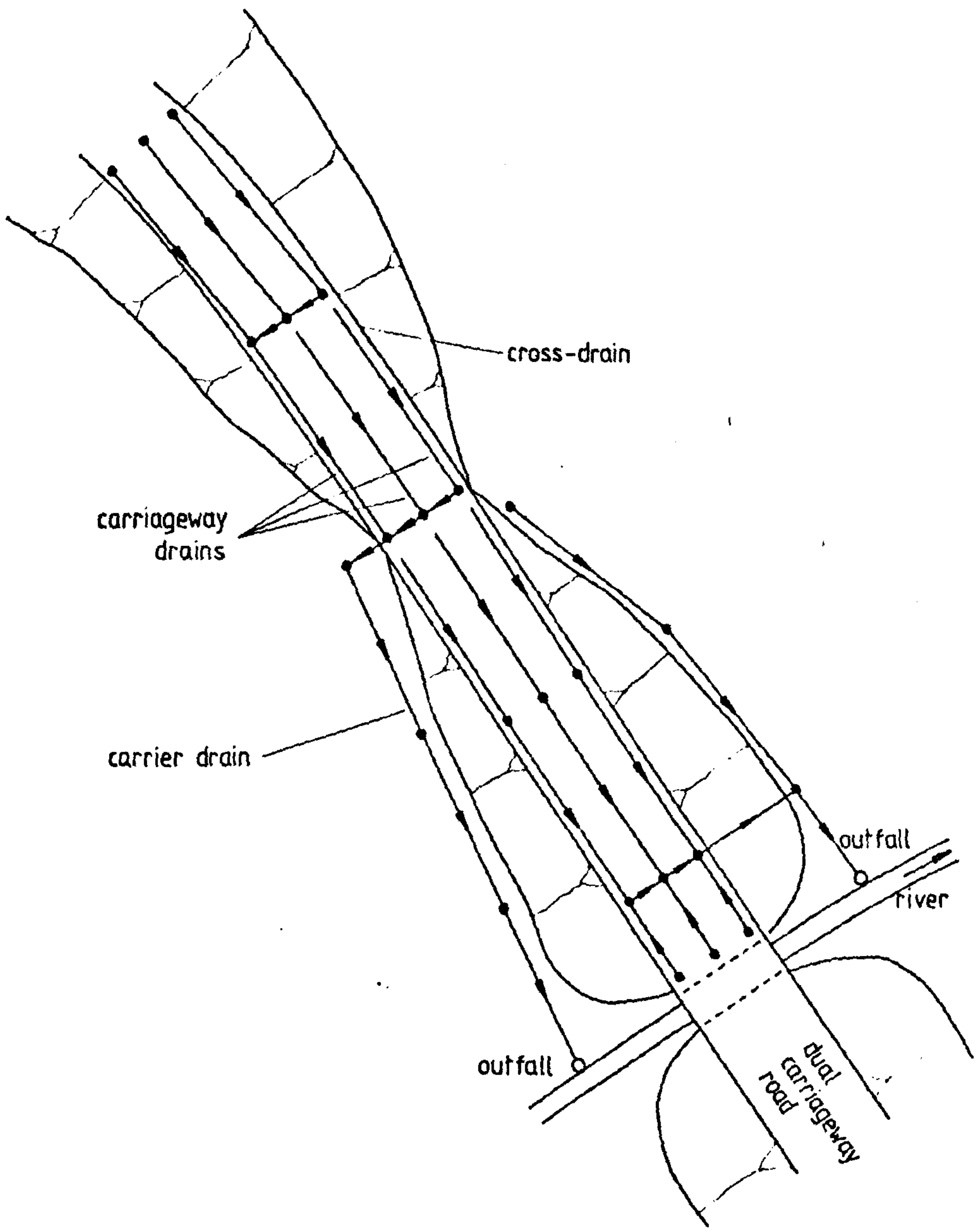
A third type of drain exists which will be referred to as a "carrier drain". Carrier drains convey water from carriageway or cross-drains to the outfalls of the network. Water can only enter carrier drains at manholes. Carrier drains and carriageway drains may sometimes share the same trench.

It is general practice to provide manholes for maintenance purposes at all drain junctions, changes in pipe size or changes in pipe gradient and at intervals along all drains subject to maximum spacing restrictions. Cross-drains will not have any intermediate manholes except one in the central reservation where such a reservation exists.

A manhole will usually be placed at the head of a drainage run, but sometimes a "rodding eye" will be provided instead, thus giving a cheaper form of access. "Rodding eyes" can be considered as cheap manholes for the purpose of the present research.

Figure 6.2 shows a typical dual-carriageway storm-water drainage system, consisting of carriageway drains, cross-drains, carrier drains and outfalls. The drains form two tree-like networks.

Design of the networks conventionally start with the designer drawing a plan of the pipework layout, specifying the position of all manholes and calculating the catchment areas for all pipes. Values of runoff coefficient (runoff/rainfall) are specified for the various parts of the catchment (e.g. carriageway, verge, hard shoulder). The storm severity is selected by the choice of a return period. As highway pipes usually have diameters of less than 600 mm, Road Note 35 (ref. 5) allows design flows to be calculated by the Rational (Lloyd-Davies) method. The designer proceeds with the sequential design of all pipes in the network, starting with those at the upstream ends and then working downstream. In general the designer will place all pipes at minimum possible cover and select the pipe diameters that will convey the required flow at the resultant gradients.



TYPICAL HIGHWAY STORM DRAINAGE NETWORK

FIGURE 6.2

6.3 Potential for Optimisation

It has been shown in chapter 5 that, given the position of all manholes, the diameters and slopes of all pipes may be optimally designed using MANFIX.

What scope is there, however, for altering the position of manholes to improve the design still further? To answer this question, it is desirable to examine the procedure adopted by the designer in positioning the manholes. He must first decide what type of drain is necessary and over what length it is required. In addition he must choose the offset of carriageway drains from the road centreline. These decisions can be regarded as fixed and invariable in any optimisation. For example in Fig.6.3(a), there must be drains between A and D, B and E, C and F, and also, therefore, manholes at A, B, C, D, E and F.

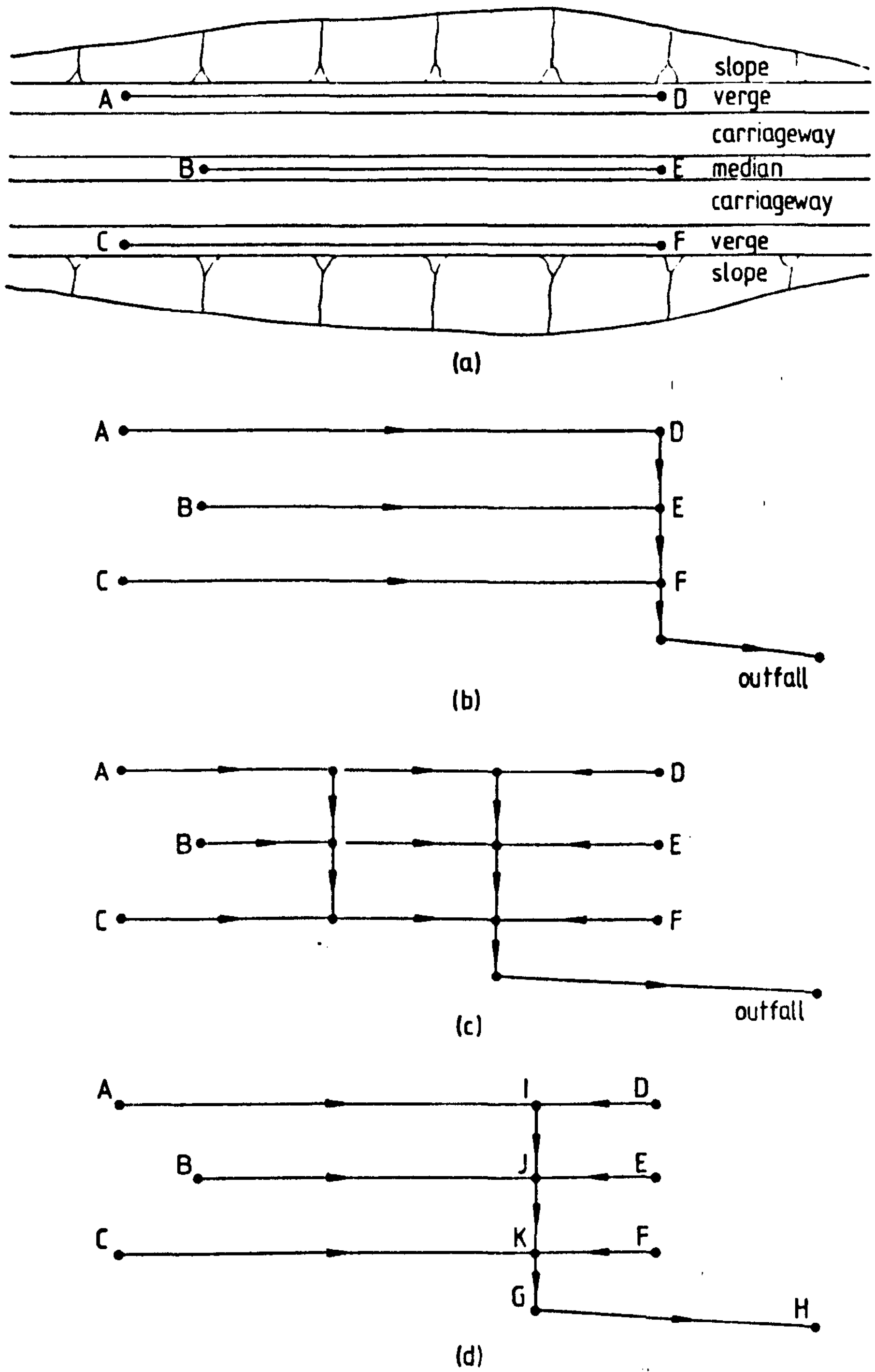
There remains a certain flexibility about how these drains are connected to an outfall.

There can for example, be one or more crossdrains with one outfall as in Fig. 6 .3 (b) and (c).

Other schemes could involve several outfalls.

In practice the number and position of the outfalls will be largely governed by factors other than minimum cost design. Water authority requirements and availability of land being two important constraints in the U.K.

Again there will almost always be a cross-drain at the lowest point along the length of carriageway under consideration. Hence it is reasonable to assume that in figure 6.3 (d) manholes A, B, C, D, E, F, G, H, I, J, K are all effectively fixed in plan.



LAYOUT OF HIGHWAY DRAINAGE NETWORKS

FIGURE 6.3

The designer then has to decide first of all if any additional cross-drains are required and if so, where. Secondly he must decide on the number and position of all intermediate manholes, remembering that manholes must be provided at changes in pipe diameter and slope and at intervals not greater than a certain maximum spacing for maintenance requirements.

These decisions are essentially of an economic nature as the functional efficiency of the network is not likely to be altered by such decisions and yet the network cost may well be. Yet these decisions are usually made on engineering judgement and experience. Unfortunately they may well be taken by engineers with limited experience and consequently little foundation for engineering judgement. Even experienced engineers would have little, if any, quantitative evidence of costs on which to base their judgement. Cost comparisons of alternative layouts are rarely, if ever, performed.

These decisions, on the number and position of cross-drains and intermediate manholes, therefore appeared to be prime candidates for computer based optimisation methods.

A model for optimising the number and position of intermediate manholes is presented in chapter 7, whilst chapter 8 presents a model for additionally optimising the number and position of cross-drains.

CHAPTER 7

THE VARIABLE MANHOLE POSITION MODEL - MANVAR

7.1. Introduction.

7.2. Defining the Problem.

7.3. The Design Flow.

7.4. Method of Approach to the Optimisation Problem.

7.5. A Dynamic Programming Approach.

- 7.5.1. Introduction.
- 7.5.2. The Basic Skeleton Serial System.
- 7.5.3. The Design of a Run by Normal D.P.

7.6. Indeterminate Stage Dynamic Programming.

- 7.6.1. Introduction.
- 7.6.2. A Modified Serial System.
- 7.6.3. Intermediate Manholes and the Modified Serial System.
- 7.6.4. Applying I.S.D.P. to the Variable Manhole Problem.
- 7.6.5. Efficiency of I.S.D.P. for the Variable Manhole Problem.

7.7. The Set of Discrete Possible Manhole Positions.

7.8. Establishing the Ranges of Value for the State Variables.

7.9. Dependence of Flows on Network Design.

- 7.9.1. Introduction.
- 7.9.2. An Approximate Approach.

7.10. Experience and Results of Using Preliminary Program DPO.

- 7.10.1. Introduction.
- 7.10.2. The Test Networks.
- 7.10.3. General Results.
- 7.10.4. Selection of Optimising Parameters.

7.11. The Variable Manhole Position Model - MANVAR.

- 7.11.1. Introduction.
- 7.11.2. Structure of MANVAR.
- 7.11.3. Implementation of MANVAR.

7.12. Program ASSEMB.

7.12.1. Introduction.

7.12.2. The Program.

7.12.3. Input.

7.12.4. Output.

7.12.5. Use of ASSEMB.

7.13. Program MOD.

7.13.1. Introduction.

7.13.2. The Program.

7.13.3. Input.

7.13.4. Output.

7.14. Results from Using ASSEMB and MOD.

7.14.1. Introduction.

7.14.2. Checks on the consistency of MOD and DPO.

7.14.3. The Effect of Varying the Optimising Parameters.

7.14.4. Varying the Design Parameters.

7.14.5. Tests on Other Networks.

7.15. Conclusions from Using MOD.

7.16. A Commercial Program.

7.16.1. Introduction.

7.16.2. The Existing Program DAPHNE.

7.16.3. The Optimising Version DAPHOP.

7.17. Conclusions on the MANVAR Model.

Chapter 7. The variable manhole position model - MANVAR

7.1 Introduction

An extensive literature search in the fields of drainage and optimisation failed to unearth any published material of relevance to the problem of positioning an unknown number of manholes along a drainage run.

Previous research in the optimisation of variable plan drainage networks has generally concentrated on problems of connectivity (refs. 33,38,44) rather than the problem of a variable plan position for a manhole. A review of multivariable optimisation techniques showed that they generally deal with problems in which the number of variables is known. Here the number of manholes, and hence the number of variables, is initially unknown.

7.2 Defining the Problem

For a typical tree-like network the problem is to find the number of intermediate manholes along each non-branching run, together with their positions, together with the diameters and levels of all pipes, such that the total construction cost of the network is as small as possible whilst all the technological and physical constraints imposed on the system are met.

One of the constraints given in section 2.3 is a condition that manholes should not be spaced at more than a given distance apart, Lmax, along each run.

As an example consider a network of m pipe runs between (m + 1) fixed manholes.

Consider a typical pipe run I, (I = 1 to m).
Pipe run I consists of an unknown number of manholes, N(I).
Define an element as a pipe with its upstream manhole (see Fig. 4.1).

The design of an element (I,J) can then be defined in terms of the pipe diameter D(I,J), upstream and downstream pipe levels Zu(I,J), Zd(I,J), and upstream and downstream distances Xu(I,J), Xd(I,J) from a fixed manhole.

In general given Zu and Zd, Xu and Xd, the smallest and hence cheapest pipe size that will carry the required flow and satisfy the flow constraints will be chosen. Hence the pipe diameter D(I,J) is dependent on Zu(I,J), Zd(I,J), Xu, Xd and need not be considered as an independent variable.

There are then 4 × N(I) variables for a typical branch where N(I) is itself an additional variable, and with the constraints that Xd(I,J) = Xu(I,J+1) and Xu(I,1) = 0, Xd(I,N(I)) = length of branch, Lb(I).

The cost of constructing a pipe element is a function of element length, pipe diameter, average depth and upstream depth. In terms of the independent variables the cost of an element Ce(I,J) = f{Zu(I,J), Zd(I,J), Xu(I,J), Xd(I,J)} Let the cost of constructing a pipe run be Cb(I). Hence the problem becomes one of minimising C where

$$C = \sum_{I=1}^{m} Cb(I)$$

$$= \sum_{I=1}^{m} \sum_{J=1}^{N(I)} Ce(I,J)$$

where

$$Ce(I,J) = f\big(Zu(I,J),\ Zd(I,J),\ Xu(I,J),\ Xd(I,J)\big)$$

and N(I) are variable parameters in the minimisation, subject to a set of constraints of the form:

$$Xd(I,J) = Xu(I,J+1) \qquad \begin{cases} I = 1,m \\ J = 1,N(I)-1 \end{cases}$$

$$Xu(I,1) = 0$$

$$Xd\big(I,N(I)\big) = L(I)$$

and constraints $0 \leq Xd(I,J) - Xu(I,J) \leq Lmax$

and constraints $Ymin \leq Y \leq Ymax$

$smin \leq s \leq smax$

$Vmin \leq V \leq Vmax$

$Q \leq Qf$

$Zu \leq Zus$

$D \geq Dus$

D a discrete, available, diameter.

7.3 The Design Flow

The design flow Q for an element will in general depend on the total catchment area, A, for the element and, if the Rational method of design is used, on the time to concentration, tc, (i.e. time taken for runoff to reach the downstream end of the element from the furthest upstream point).

Hence $Q = f(A, tc)$

The catchment area in general increases with distance along a run.

Hence $A = f(Xd)$

The time to concentration, tc, depends on the diameters, slopes and lengths of all pipes upstream of the element and on the diameter, slope and length of the pipe in the element.

If an approximation can be made for the time to concentration such that it is independent of the pipe diameters and gradients upstream and merely dependent on position, then $tc = f(Xd)$.

Hence as $Q = f(A, tc)$

$Q = (f(Xd)$

7.4 Method of Approach to the optimisation problem

The main difficulties involved in forming an optimisation model for this application are listed below.

a) Unknown number of variables.

b) Non-linear, non-differentiable objective function.

c) Discrete values of pipe diameter.

Difficulty (a) could be partially overcome by assuming that a sufficiently large number of intermediate manholes exist thus giving a definite number of variables and allowing solutions in which many of these manholes are coincident. This is very inefficient and the presence of singularities in the solution would present great problems to any known multi-variable optimisation algorithm.

Even if this first difficulty could be overcome, for the reasons discussed in section 5.3 there would still be formidable problems in producing a robust, economic and effective model based on conventional multi-variable optimisation algorithms.

As Dynamic Programming had been shown to be effective in fixed-plan drainage optimisation it was decided to investigate whether it was also suitable for the problem of variable manhole positions.

7.5 A Dynamic Programming approach.

7.5.1 Introduction

As has been shown in Chapter 5 Dynamic Programming is very efficient when dealing with serial systems, and can cope well with discrete valued variables and discontinuous objective functions. Fixed plan drainage networks were shown in section 5.5. to be serial systems suitable for D.P. The questions in dealing with the variable manhole position problem are whether some or all of the network can be considered a serial system, and whether such a serial system is amenable to D.P.

Throughout this chapter it is assumed that diameters may not decrease in a downstream direction (See 2.3.(g)). Hence pipe diameter is a necessary state variable (See 5.5.4).

7.5.2 The basic skelton serial system

If the fixed manhole positions are assumed to define a basic skeleton of drainage runs, (e.g. Fig. 6.3(d)), the design of each run could then be considered as a stage in a serial system. This is evident by direct comparison with the fixed plan serial system described in section 5.5.

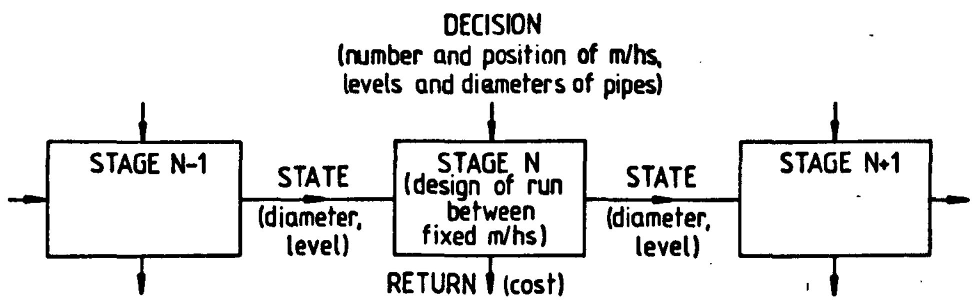
Such a serial system is illustrated in Figure 7.1(a).

An individual stage involves decisions on the number and position of intermediate manholes along a run together with decisions on the diameter and slopes of all pipes.

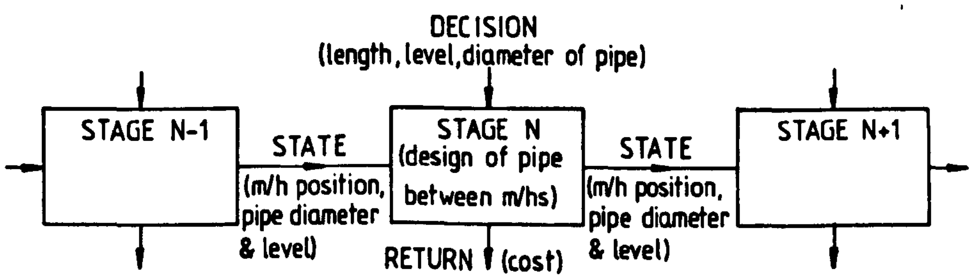
If a method of producing an optimal design for an individual run can be found, it follows from the nature of serial systems that the optimal solution for the whole network can be established by D.P.

7.5.3 The design of a run by normal D.P.

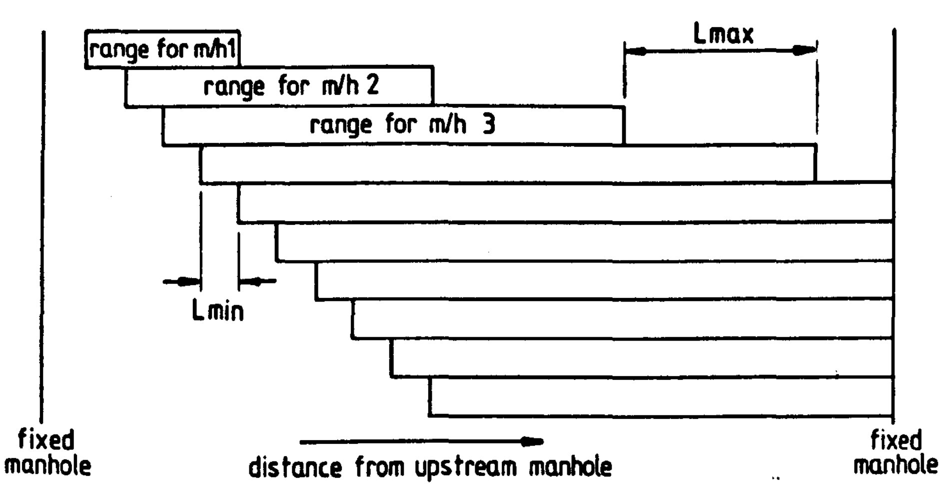
The same principal difficulties exist in trying to optimise the design of an individual run as do for the problem of optimising the complete network (see 7.4). Admittedly the scale of the problem is



(a) SERIAL SYSTEM FOR SKELETON NETWORK



(b) SERIAL SYSTEM FOR NON-BRANCHING RUN



(c) RANGES OF MANHOLE POSITIONS

FIGURE 7.1

much reduced. Even so conventional multi-variable optimisation is unlikely to be a fruitful line of approach.

Hence D.P. was again investigated carefully as the most likely way of achieving satisfactory results.

The first approach was to assume that there were a fixed number of stages in the design of a run. Each stage consisted of the design of a pipe with its upstream manhole, the design consisting of the length, diameter and levels of the pipe. To allow the number of manholes to be variable, a stage could consist of a pipe of zero length, in which case the stage return (cost) would be zero. Such a system would need to have pipe level, pipe diameter and position of manhole as state variables, and is shown diagrammatically in Fig. 7.1(b).

Assume that manholes have a maximum spacing, and also a minimum spacing. The possible positions of successive intermediate manholes is then as shown in Fig. 7.1(c).

If the minimum spacing = Lmin , the maximum spacing = Lmax and the length of run = Lb , then the possible range of position for the Nth intermediate manhole is the lesser of

$$N \times (Lmax - Lmin) \quad \text{and} \quad (Lb - N \times Lmin)$$

and the total number of possible stages is the integer value of (Lb/Lmin).

It can be seen that, if the total length of run is large compared to the minimum spacing, the possible range of position for a manhole could itself be large.

For highway drainage a typical run length could exceed 1 km, with a minimum spacing of, say, 30 m and a maximum of, say, 150 m. This would give 33 stages and a range of 790 m for the position of the seventh manhole.

The number of stages could probably be substantially reduced without affecting the solution in almost all cases. Indeed, with experience, the range of position for the manholes could probably be reduced somewhat.

It seems therefore that a practicable solution may be possible using the above approach. The main disadvantages are

(a) three state variables

(b) large range of values for the manhole position state variable

(c) number of stages has to be pre-determined. Hence some redundant stages are inevitable.

To illustrate these problems consider briefly the typical highway case outlined above.

Assuming that discrete values are adopted for the state variables in the D.P. process (see Section 5.6.4), the state variables level, diameter and position may have ℓ, m and n discrete values respectively.

There are thus $(\ell \times m \times n)^2$ designs to consider at each stage. If there are N stages and ℓ, m and n are constant for each stage, there are $N \times (\ell \times m \times n)^2$ elemental designs for a complete run.

Assume values of ℓ and m are the same as the typical values adopted in the MANFIX model, i.e. $\ell = 7$, $m = 3$. For a run length of 1 km assume that the number of stages can in practice be reduced to 10 and the maximum range of positions to, say, 450 m. Then if the discrete values of the position state vector are taken at 30 m intervals, n = 16 and the total number of designs = $10 \times (7 \times 3 \times 16)^2$ = 1,128,960.

Hence, although the method could well be successful in achieving near optimal results in most cases, it seems likely that the computer time required may be unrealistically large.

7.6 Indeterminate Stage Dynamic Programming

7.6.1 Introduction

In an attempt to improve on the DP approach of section 7.5.3, a new concept in DP was developed. This will be called, for convenience, Indeterminate Stage Dynamic Programming (ISDP).

As the name suggests the stages are not predetermined either in number or position but result from the DP optimisation.

When applied to the intermediate manhole problem an elegent and effective method results.

7.6.2 A modified serial system

A modified serial system was adopted in which the input state to a stage results from the output from one of a range of possible previous stages. This is best explained by a simple example.

Consider a set of stages a, b, c, d etc. as shown in Figure 7.2(a). Consider stage d. Allow the input to stage d to be the output from any of stages a, b, or c with the actual choice being one of the decisions D. Depending on the decision D, stage c may or may not be redundant, or both b and c may be redundant.

Likewise stage c could have either stage a or stage b as input.

Hence the following are all possible serial systems; a b c d, a b d, a c d, a d, with the actual serial system adopted being dependent on the decisions made at c and d. Note that there could be 2, 3 or 4 stages in the final serial system.

7.6.3 Intermediate manholes and the modified serial system

Consider a run along which an unknown number of intermediate manholes are to be placed, with fixed manholes Y and Z at the upstream and downstream ends.

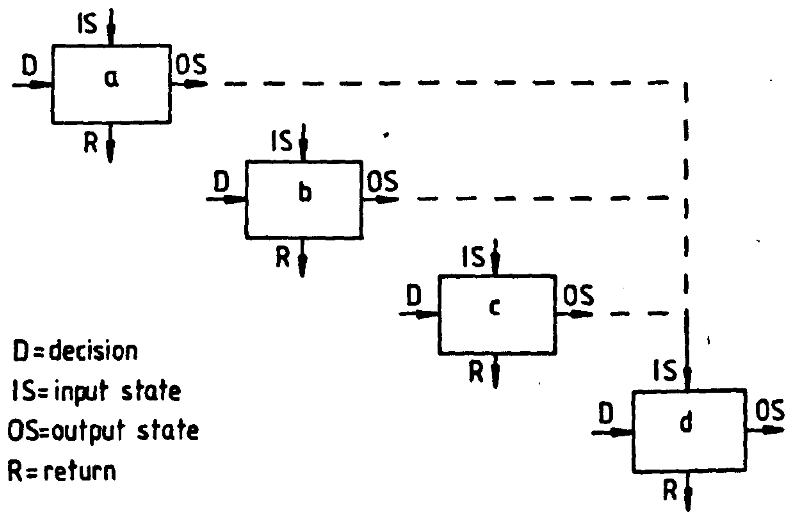
It is possible to define a modified serial system as described in 7.6.2 in the following way.

Define a set of possible discrete intermediate manhole positions a, b, c etc. along the run (Fig. 7.2(b)). Let each of these correspond to the downstream end of a stage in the modified serial system. The input to one of these stages is then the output from one of the upstream stages. Figure 7.2(c) shows the modified serial system.

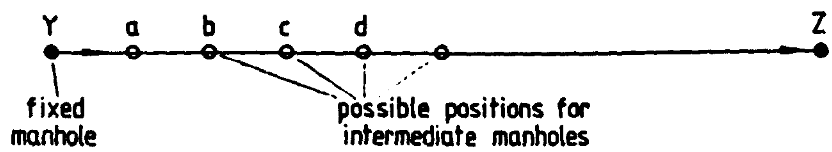
7.6.4 Applying ISDP to the variable manhole problem

The dynamic programming is now performed in a standard way except that instead of considering the input state for a stage as being any one of the output states from the one stage immediately upstream, the input state must now be considered as any one of the output states from any feasible previous stage. A previous stage can be infeasible if the distance between the stages is greater than the maximum manhole spacing or smaller than the minimum spacing if this is specified.

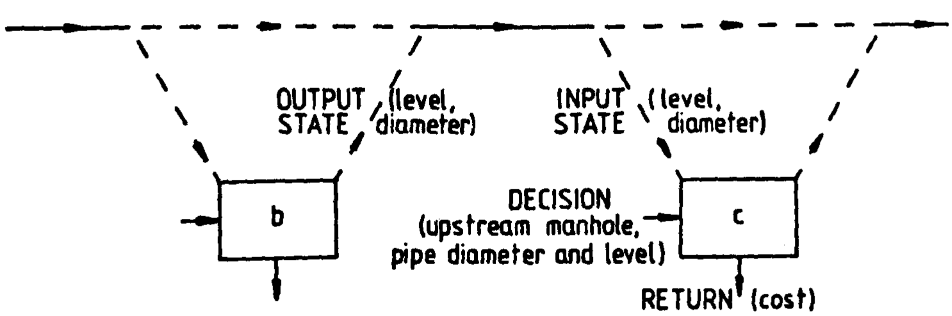
As an example consider the case of possible intermediate manholes a, b, c etc. along fixed run YZ (see Fig. 7.2(b)).



(a) STAGES FOR MODIFIED SERIAL SYSTEM



(b) STAGES FOR A NON-BRANCHING DRAINAGE RUN



(c) MODIFIED SERIAL SYSTEM FOR DRAINAGE DESIGN

FIGURE 7.2

Assume that ranges of the state variables, level and diameter, have been established at each possible intermediate manhole position along the run, and that discrete values of these variables have been specified.

Stage a has the fixed manhole Y as its upstream manhole and ends at the possible intermediate manhole position a. In general there will be a set of input states for stage a corresponding to discrete values of the input state variables, together with a set of costs corresponding to these input states. If Y is an upstream end of the network, the input states will still exist but the associated costs will be zero.

Then, for a particular output state from stage a, select the way of arriving at that state from any input state such that the total upstream cost is least whilst satisfying all the constraints.

Repeat this for every output state, thus obtaining a set of minimum total upstream costs at the output from stage a, and a set of references to identify the input state corresponding to that minimum cost.

Now consider stage b. This stage may either have the fixed manhole Y, or the possible intermediate manhole a as its upstream manhole provided that the distance from Y to b is less than the maximum spacing and that the distance from a to b is greater than the minimum spacing. Hence for a particular output state from stage b, select the way of arriving at that state from any input state either at manhole Y or at manhole a such that the total upstream cost is least, whilst satisfying all the constraints.

Repeat this for every output state, thus obtaining a set of minimum total upstream costs at the output from stage b, and a set of references to identify the upstream manhole and input state corresponding to that minimum cost.

Similarly Y, a and b can be considered as feasible upstream manholes for stage c provided the maximum or minimum spacing constraints are not violated. If, say, the distance from Y to c is greater than the maximum spacing then Y is not considered as a feasible upstream manhole for this (or subsequent) stages, and stage c is optimised using only manholes a and b.

This process is continued until the final fixed manhole Z is reached, this last stage being treated in an identical way to give a set of minimum costs and a set of upstream references. The process is illustrated in Figure 7.3.

If Z is the outfall to the network the costs at Z may now be examined and the output state giving the least cost selected. This gives the origin for the trace back up the run YZ. If, however, the network continues downstream of the fixed manhole Z, the trace-back origin for YZ will be obtained as part of the trace back over the whole network.

The trace back up run YZ then proceeds as follows. The upstream reference for the origin will give the upstream manhole and output state for the optimal solution. This manhole position and output state will in turn have an upstream reference to another manhole position and output state. Hence the trace back up the branch will eventually lead to the fixed manhole Y. This is illustrated in Figure 7.4.

In this way the positions of the manholes, pipe diameters and pipe levels will have been simultaneously chosen to give the least cost solution.

7.6.5 Efficiency of ISDP for the variable manhole problem

As a comparison with the DP approach proposed in 7.5.3 consider the number of designs required for the same 1 km run using consistent parameters.

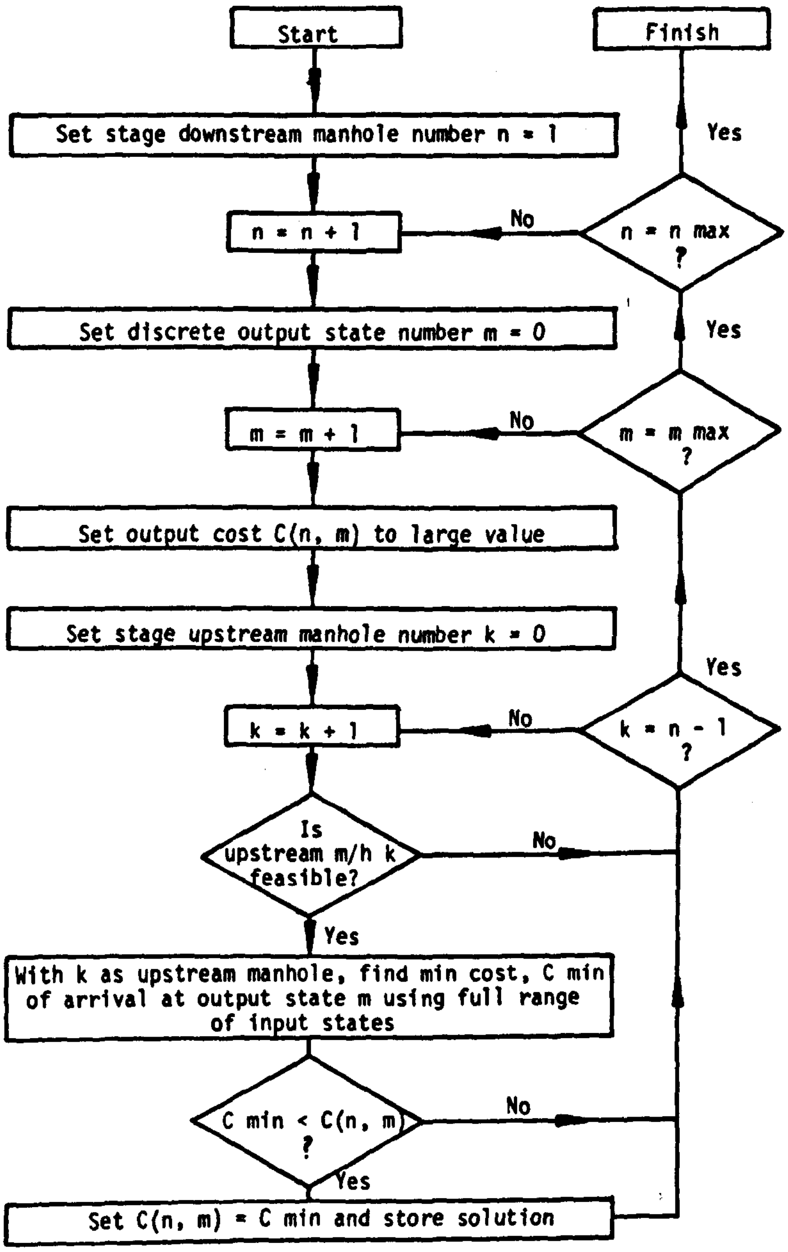
Hence take intermediate manholes at 30 m spacing, with, say, a maximum manhole spacing of 150 m and a minimum of 30 m. Take 7 discrete levels and 3 discrete diameters.

There are a total of 34 manholes, giving 33 stages. Each stage has a maximum of 5 possible upstream manholes, with 7 × 3 = 21 input states per manhole.

Hence the maximum number of designs per stage = 21 × 21 × 5 and the maximum number of elemental designs for the run

$$= 21 \times 21 \times 5 \times 33$$
$$= 72,765$$

This compares with 1,128,960 for the DP approach.

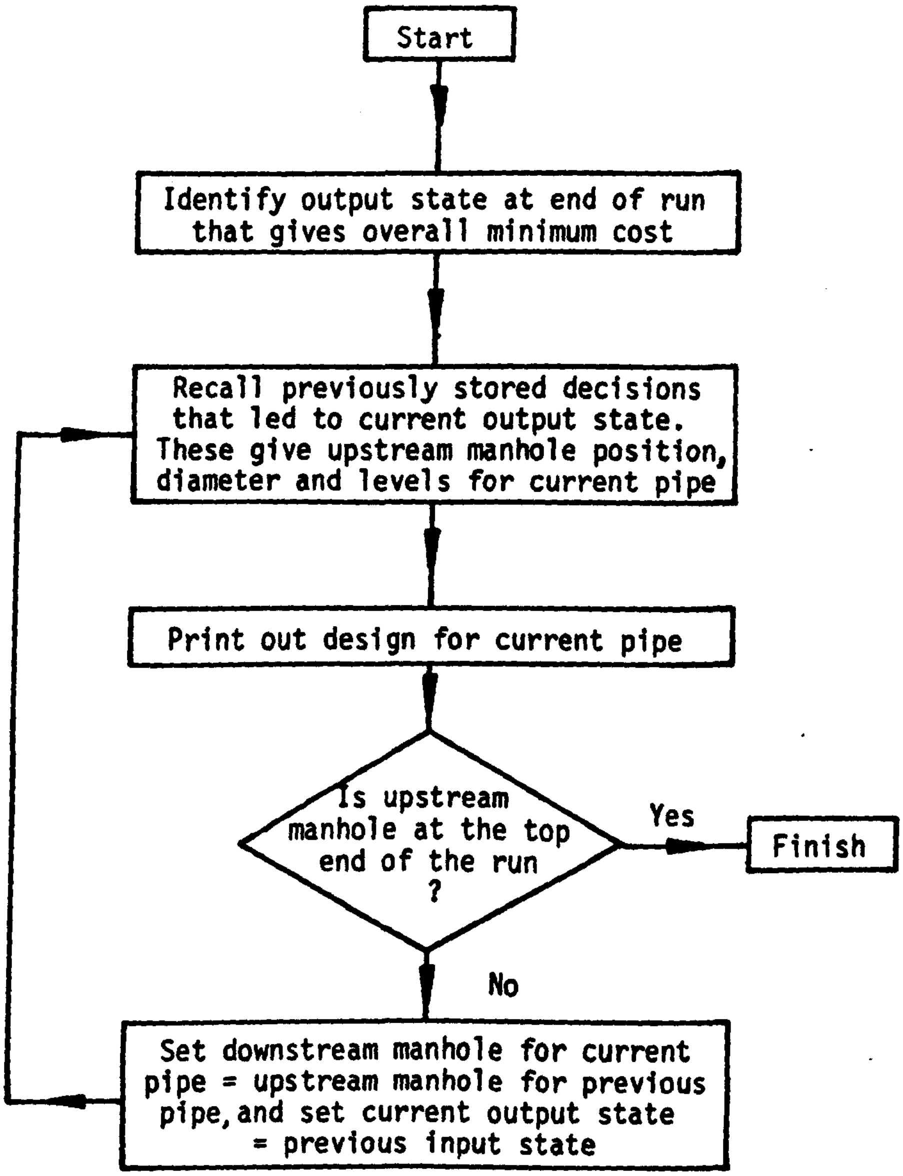


Notes 1) possible manhole positions are numbered in downstream direction from fixed manhole 1 to fixed manhole n max

2) m max = number of discrete states at each manhole

ISDP APPLIED TO VARIABLE MANHOLE POSITION PROBLEM

FIGURE 7.3



TRACE-BACK FOR THE VARIABLE MANHOLE POSITION PROBLEM

FIGURE 7.4

7.7 The set of discrete possible manhole positions

One of the key concepts involved in the ISDP approach to the intermediate manhole problem is the establishment of a set of discrete possible positions for the intermediate manholes.

To obtain a true optimal solution the intermediate manholes should not be constrained to a set of discrete positions. Hence in theory an infinite number of discrete manhole positions is required to achieve an optimal solution. Obviously the number has to be limited in practice, and this is in keeping with the discrete values adopted for the continuous state variable, pipe level. For practical highway drainage there is another justification for using a set of discrete possible positions for the manholes, this being the preference of highway engineers to the placing of manholes at convenient chainages along a road. For example a designer may well wish to have all manholes at chainages which are multiples of 10 m. Establishing a set of possible manhole positions at all such chainages along a length of carriageway drain would then give a practicable and elegant solution to the problem.

One further advantage of this approach is that manholes may be excluded from certain parts of a run by simply not specifying any manhole positions along that part. This may be necessary at, for example, bridges, culverts and road junctions.

7.8 Establishing the ranges of value for the state variables

It was assumed in section 7.6.4 that a set of discrete levels and diameters had been established at each possible intermediate manhole position as an essential prerequisite of the dynamic programming method.

In Chapter 5, the fixed plan model, this was achieved by producing a minimum cover design, (see section 5.10.3) and using this as the upper limit of both pipe level and pipe diameter. The lower limit was then a fixed distance below the upper limit for level and a fixed number of pipe diameters below the upper limit for diameters.

It would be very useful if the same approach could be used for the variable manhole problem. However it is not immediately obvious how to perform a minimum cover design when the manhole positions are undetermined.

If an arbitrary set of manhole positions is assumed and a minimum cover design is performed, the resulting information on maximum level and diameter will only be relevant to the manhole positions used and not to all the other positions which were possible but not selected. For example in Figure 7.5(a) if A.B.C...G are possible manhole positions, and A and G are used to establish a minimum cover design, upper bound pipe levels at B,C,D,E,F are all lower than they could be in the true optimal solution.

It is, therefore, necessary to consider a minimum cover design based on the inclusion of manholes at all possible manhole positions. Such a design is of course unrealistic but does nonetheless provide a useful upper limit on level at each possible manhole position. Experience shows that it also provides a satisfactory upper limit on diameter.

From these upper limits, lower limits of level and diameter at manholes can be established from experience (see section 7.10).

It should be noted that the range of levels at a possible manhole position as defined by the upper and lower limits applies only to a solution with a manhole at that position. Hence the final optimal solution is not constrained within a range of levels along each pipe length, only within ranges of levels at each final manhole position (see Figure 7.5(b)).

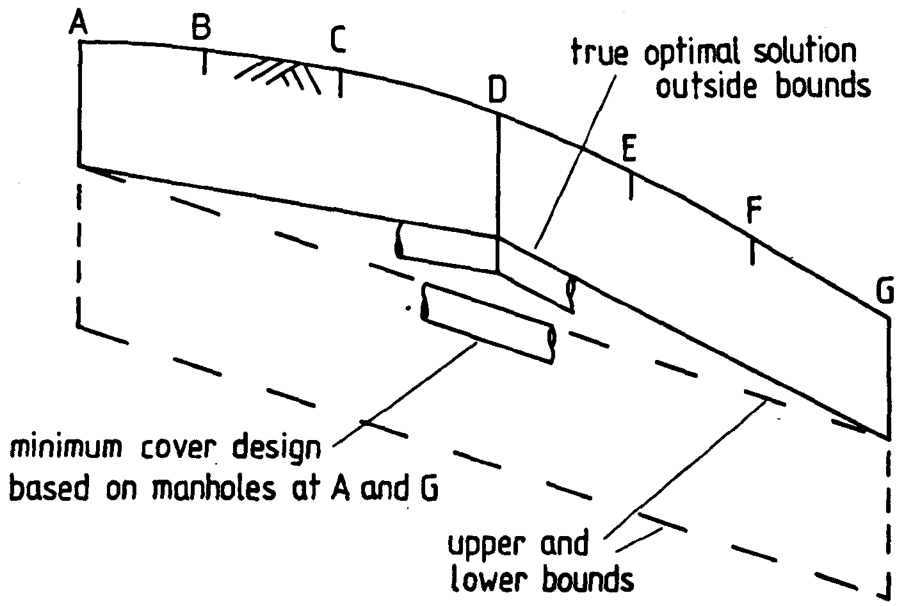
The number of discrete values and diameters used in the DP are chosen by experience (see section 7.10).

7.9 Dependence of flows on network design

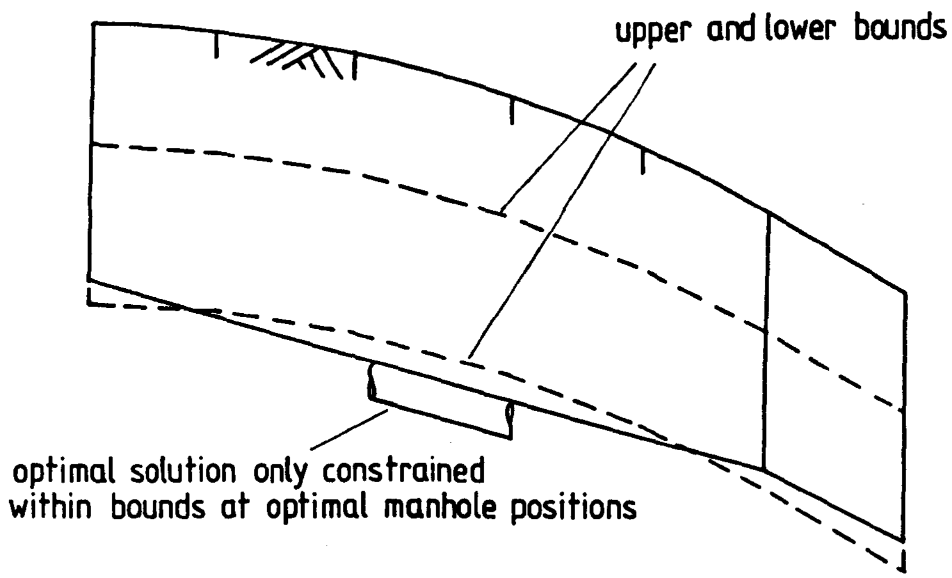
7.9.1 Introduction

It was noted in section 5.11 that design flows are usually dependent on the pipe network upstream of the point under consideration, and for a rigorous DP approach using the most common, the Rational, design method three state variables are required, the third being the time to concentration. So far in this chapter it has been assumed that design flows are fixed.

It was shown in section 5.11.5 that an approximate method could be used with success for the fixed manhole case. Such a method is now required for the intermediate manhole problem before a full optimisation model can be formulated and the development of such a method is given in the following sections.



(a) USING SELECTED MANHOLE POSITIONS



(b) USING ALL MANHOLE POSITIONS

ESTABLISHING BOUNDS ON PIPE LEVELS

FIGURE 7.5

7.9.2 An Approximate Approach

The simplest approximate approach is to assume that all design flows are fixed at some initial value and do not thereafter vary as part of the design process.

The method of fixing the initial values is rather less obvious and may be dependent on the design method. Here the Rational method is used in the development. The first approach adopted was to assume that the velocities of flow in the final solution would, for the purposes of calculating the design flows, be equal to a single uniform value of, say, 1.0 or 1.5 m/s. The actual design flows can then be calculated as a function of the total upstream equivalent impermeable catchment area and the time to concentration using the Rational method (see section 5.10.2).

With this simplification the design flow does not depend on the actual optimal set of manholes chosen or on the design of the pipe diameters or gradients. Hence a unique design flow can be specified for each possible manhole position in every run of the network. A computer program DPO was written to implement an ISDP model using this procedure.

Unfortunately comparison of flow velocities and computed flows for the resultant "optimal" design showed that large discrepancies resulted from such an approximate method. Velocities ranged from 0.5 to 3 times the original assumed value with resultant design flows out by up to 30%.

It was decided to use an iterative approach, with the new set of velocities from the "optimal" design being used to recalculate the design flows and the computer program DPO was altered to implement this. The velocities change abruptly at the manhole positions selected by the "optimal" design, but not at the other possible vacant manhole positions. This was recognised as a drawback which could lead to problems of convergence, but the method was nonetheless pursued to gain experience.

The iterations were continued until flow velocities converged to within a given tolerance. The first one or two iterations generally gave small changes in manhole positions and pipe diameters. Thereafter changes were generally limited to the pipe gradients. On one example the process failed to converge, with the solution hunting between two different manhole layouts. Generally, however, a fully consistent and near optimal solution was obtained within about five iterations.

The method proved rather expensive in computer resources and could not be relied upon to converge properly. However, it was found that very rapid convergence could be achieved by first performing a minimum cover design by a conventional design procedure using all possible manhole positions, and using the resultant flows from this as the starting point of the iterations. The manhole positions and diameters resulting from the first iteration usually remained fixed during subsequent iterations with only the pipe gradient changing to accommodate changes in design flows. Moreover such a minimum cover design is also necessary to establish economical ranges for the pipe levels and diameters, hence this was the method adopted.

7.10 Experience and results of using preliminary program DPO

7.10.1 Introduction

DPO was written to develop and test ISDP and the approximate procedures for dealing with network dependent design flows.

Consequently there were frequent alterations and improvements to DPO during its working life with the program being finally superseded by MOD (see section 7.13) and DAPHOP (see section 7.16).

Hence only the general results, major limitations and conclusions will be presented here, as detailed results are somewhat meaningless in the light of subsequent improvements and alterations.

The Fortran coding for the final version of DPO is given in Appendix C.

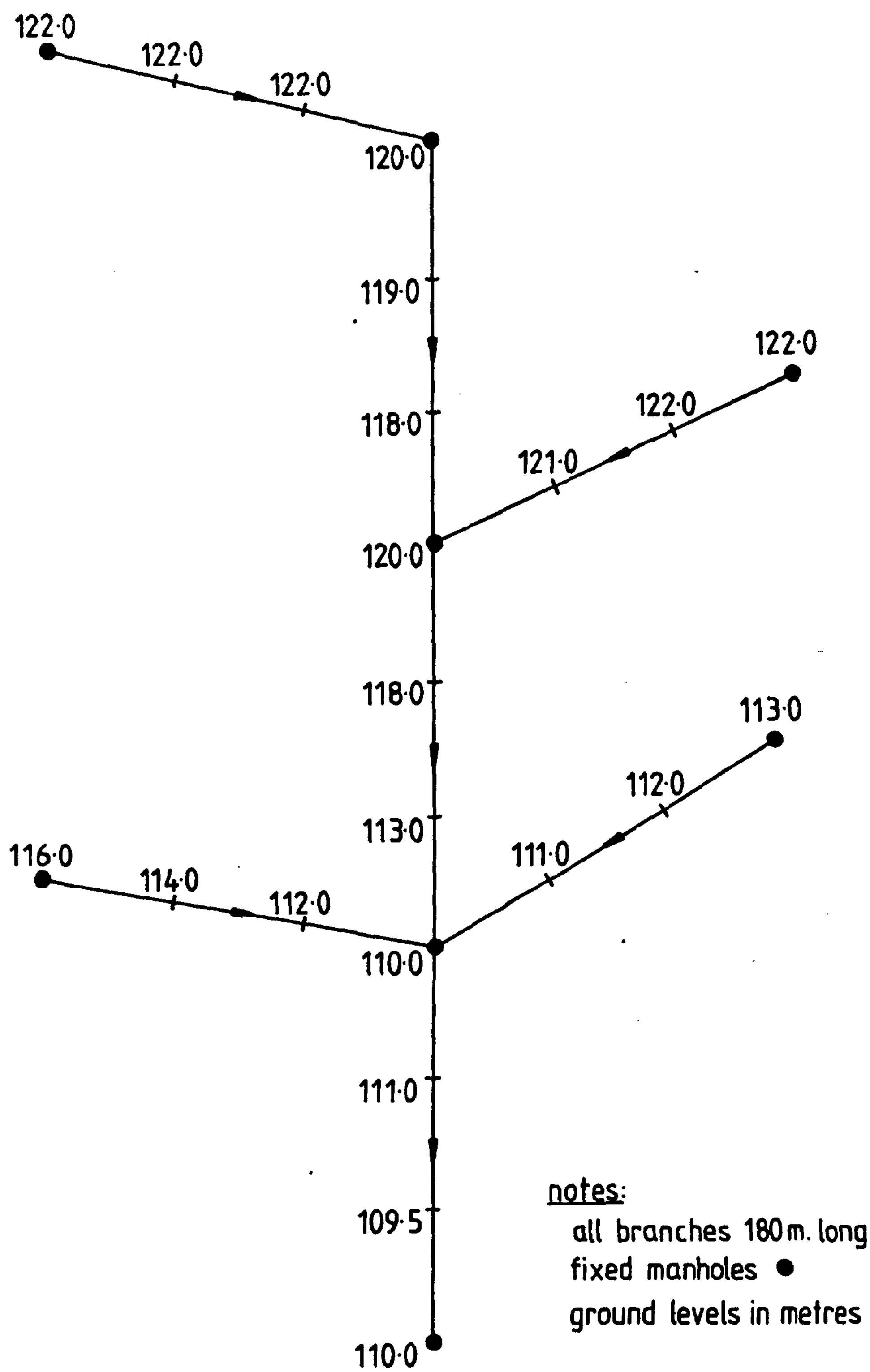
7.10.2 The test networks

Several test networks were used in the development of DPO, the one shown in figure 7.6 (Network 2) being used extensively in the investigation of the sensitivity of the solution to the choice of parameters.

7.10.3 General results

The following general observations could be made about the preliminary runs of DPO.

a) Manhole positions and pipe diameters were relatively insensitive to the design flows used at each iteration, whereas pipe slopes varied considerably between iterations.



TEST NETWORK 2

FIGURE 7.6

b) Manhole positions were quite sensitive to the choice of spacing of the discrete pipe depth state variable.

c) Optimal designs tended to lie within a 0.5 m zone below a level defined by a pipe at minimum possible cover.

d) Pipe diameters were always less than, and within two increments of, the diameter obtained from a minimum cover design.

e) Optimal costs were 5-15% lower than those of equivalent designs using the conventional minimum cover criterion.

f) Solutions obtained with discrete pipe depth increments less than about 0.15 m and adequate range of depths generally gave optimal manhole positions and pipe diameters on the first iteration.

7.10.4 Selection of Optimising Parameters

Thirteen runs were executed using DPO on network 2 (figure 7.6) for the purpose of establishing the sensitivity of the "optimal" solution to the choice of the two main optimising parameters. These parameters are

a) Spacing of discrete levels for pipe level state variable;

b) Spacing of possible intermediate manholes.

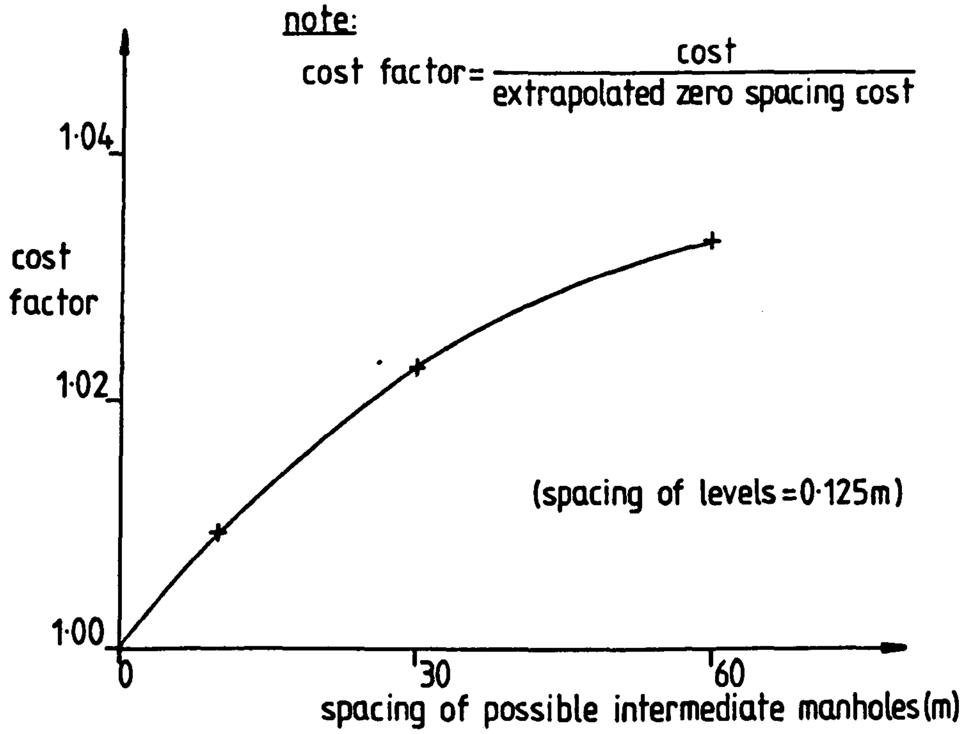
Checks were made on the results to ensure that there was a sufficient range of pipe levels and a sufficient number of discrete pipe diameters considered for the optimal solution to be within the bounds of the available values. For example if the results showed that the solution lay at or close to the lower bound of the pipe levels, the range of levels was increased without altering the spacing of levels and the optimal solution re-computed. If the solutions were found to be identical, it was assumed that the solution was then indeed optimal.

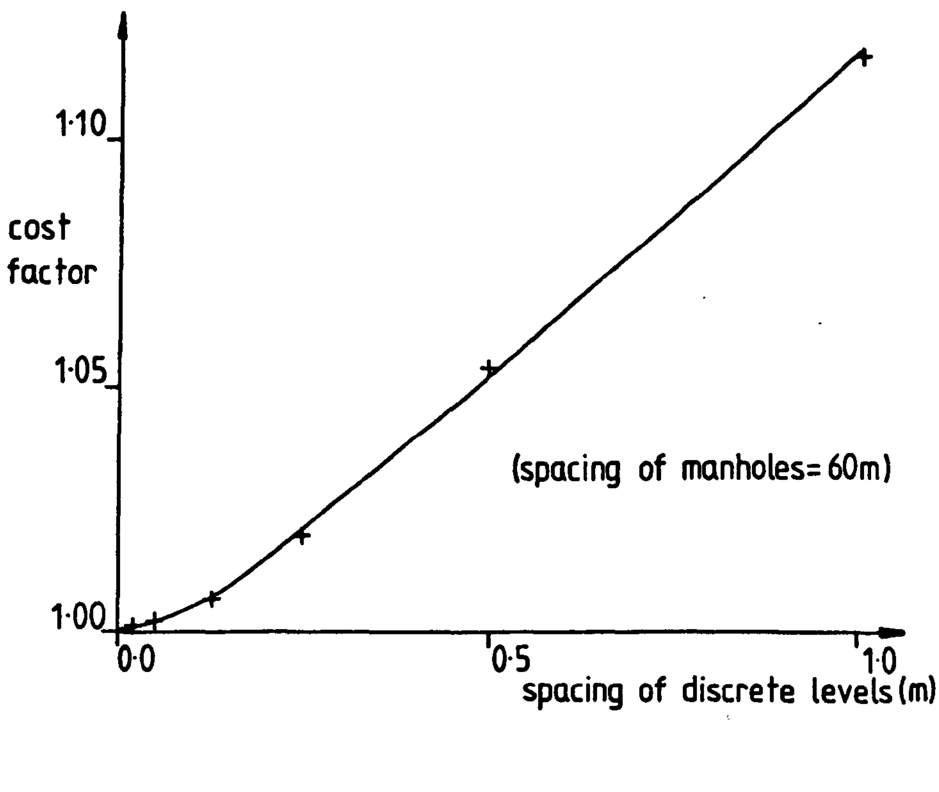
The results of these runs are shown in the graphs of figure 7.7, but should be treated with caution as they relate to a single network and are insufficient in number to establish any specific conclusions.

7.11 The variable manhole position model - MANVAR

7.11.1 Introduction

The results of the preceding section led to the formation of a new model which is both economic and robust. For convenience it is referred to as the MANVAR model, and is the basis of a fully commercial program, DAPHOP, which is described in section 7.16.





SENSITIVITY OF NETWORK COST TO OPTIMISING PARAMETERS: PROGRAM DPO

FIGURE 7.7

7.11.2 Structure of MANVAR

MANVAR essentially consists of four distinct stages. These are:

a) producing a set of possible intermediate manhole positions;

b) producing a minimum cover design and limits on the ISDP design;

c) performing a coarse ISDP design;

d) performing a final exact solution.

Stage (a) consists of taking the skeleton layout of the network and producing sets of possible intermediate manhole positions along all relevant runs. Ground levels and catchment areas are also produced for all the generated manhole positions.

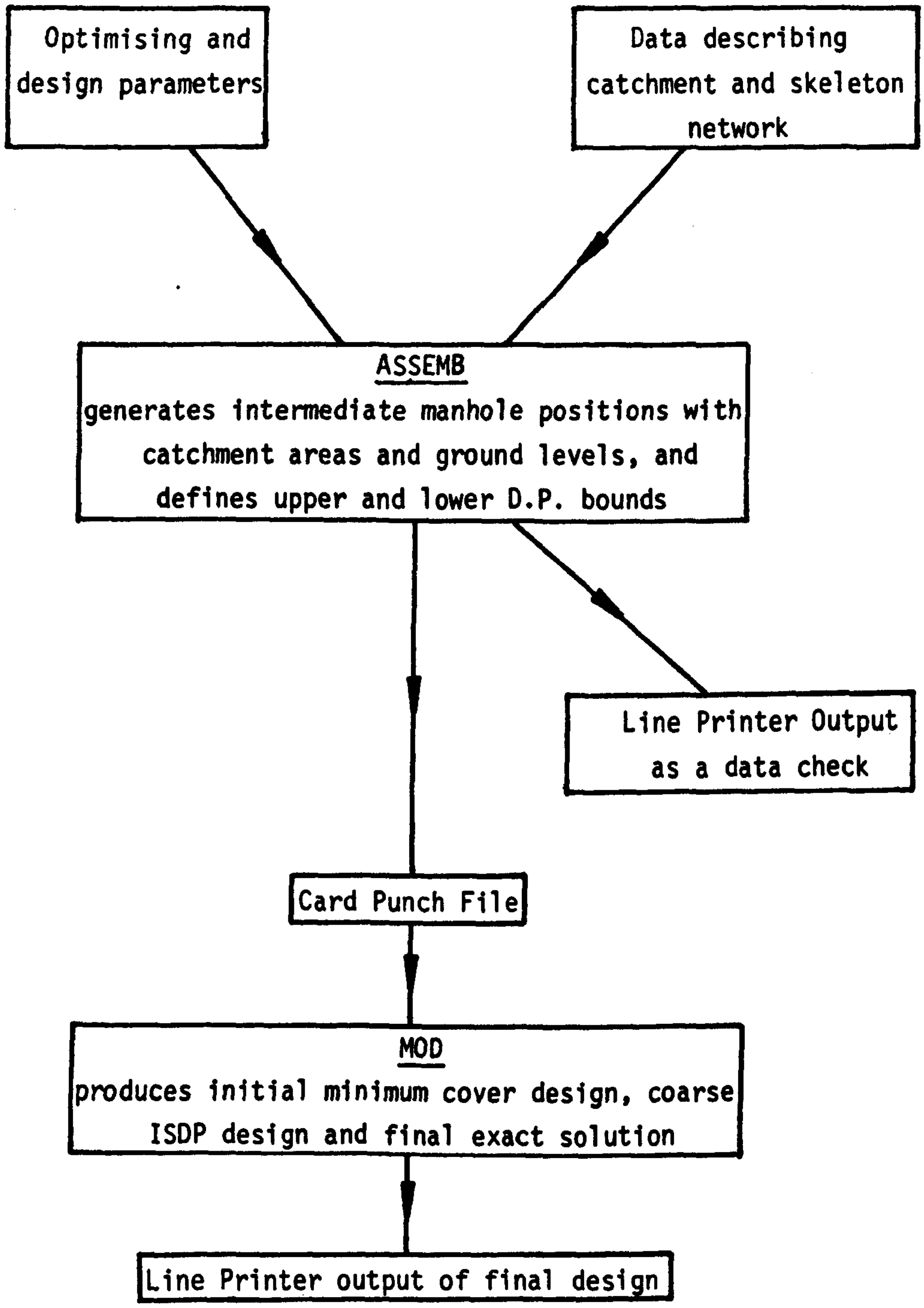
Stage (b) consists of performing a minimum cover design on the network assuming a manhole at every possible location using the design method of one's choice (Rational, TRRL etc.) to obtain upper limits on the pipe levels and diameters and the design flow at every manhole location. Lower limits on level and diameter are also set at this stage.

Stage (c) consists of a single ISDP design using a coarse grid of discrete levels, the grid of discrete diameters and the sets of possible intermediate manholes. This gives a "coarse optimal" solution, for which the actual flows will differ somewhat from the true design flows, but which, from result (f) of section 7.10.3, will generally give the optimal values of manhole positions and pipe diameters for all pipes in the network.

Stage (d) consists of taking the new network of pipes of known diameter as defined by the optimal set of manhole positions and pipe diameters, and designing the pipe gradients using the chosen design method, thereby ensuring a fully consistent final design. This effectively truncates the iterative procedure of section 7.9.2 thereby making a far more efficient model with little or no penalty incurred.

7.11.3 Implementation of MANVAR

MANVAR was implemented as two computer programs, ASSEMB and MOD, linked by a data file (see figure 7.8). Input to MOD may be either from ASSEMB or direct from the user.



IMPLEMENTING THE MANVAR MODEL

FIGURE 7.8

7.12 Program ASSEMB

7.12.1 Introduction

As explained in section 7.7 an essential prerequisite of the ISDP process is the establishment of sets of possible intermediate manhole positions along all relevant runs.

This can be done by the user first deciding on the required spacing of the intermediate manholes and then calculating and inputting all such manhole positions. This was done manually for the earliest variable manhole test runs but is tedious and is a task best performed by the computer. Hence at an early stage in the development of MANVAR a subsidiary computer program called ASSEMB was written, part of the function of which was to take the skeleton layout input by a user and produce a full set of possible intermediate manhole positions together with their associated ground levels and catchment areas.

The second function was to define the upper limit on the pipe levels for the whole network, working from information on ground levels, obstructions, minimum pipe gradient and connectivity only. It is assumed that pipes at manholes are positioned such that the downstream soffit level is at or below the soffit level of the lowest upstream pipe. Hence this upper limit can be defined without reference to pipe diameters or flows, and hence in fact without the design method being defined.

The final function was to arrange data to a form convenient for the main program.

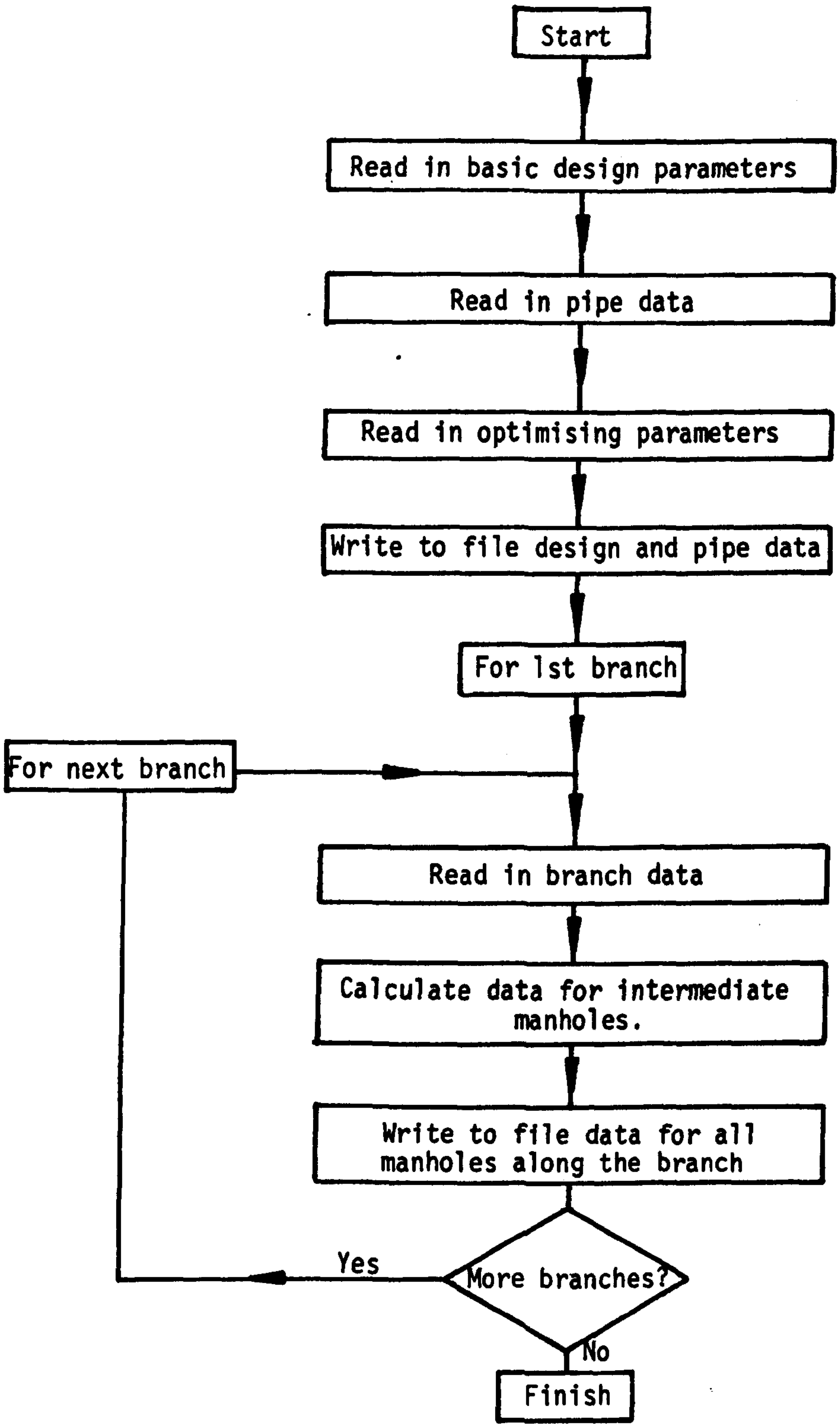
7.12.2 The program

A flow chart showing the essential features of ASSEMB is given in figure 7.9 and a full listing of the Fortran Program is given in Appendix D.

7.12.3 Input

The input to ASSEMB consists of

a) the basic design parameters (e.g. minimum pipe gradient, minimum cover, minimum flow velocity, minimum and maximum manhole spacing).
b) available pipe sizes.
c) optimising parameters for main program.
d) type, length and catchment width for each drainage run in the



FLOW CHART FOR PROGRAM ASSEMB

FIGURE 7.9

network. (The "type" determines whether or not there is any catchment to be assigned along the length of the run, and whether intermediate manholes are to be placed along the run. For the purpose of assigning catchments to the intermediate manholes it is assumed, where relevant, that the catchment area is of a uniform width parallel to the run.)

e) ground levels along each run.

f) details of obstructions along each run.

g) the connectivity of the network.

7.12.4 Output

The output from ASSEMB forms the complete input to the next part of MANVAR and consists of the following:

a) optimising parameters

b) pipe sizes

c) basic design parameters

d) number of runs in the network

e) for each run: number of possible manholes
: positions along branch of each m/h
: cumulative catchment area for each m/h
: pipe soffit level for a min. cover design (= upper limit for DP design) for each m/h
: lower limit for DP design for each m/h
: range of feasible upstream connections for each m/h (governed by max. and min. m/h spacing)
: ground level data along run
: identification of any runs upstream

f) problem size (total number of possible manholes, total number of ground levels, probable maximum number of manholes in final design).

7.12.5 Use of ASSEMB

For ease of data preparation ASSEMB may be used either interactively or remotely. Data generated by ASSEMB is written onto a card-punch file for compatibility with manually created data, and to allow small modifications, (e.g. to the optimising parameters) by changing individual lines of the output without re-running ASSEMB. The card-punch file could be listed onto actual punch cards, but on the computer system used for this research it was more convenient to use card image files within the computer memory.

7.13 Program MOD

7.13.1 Introduction

The remainder of the MANVAR model is implemented by the computer program MOD.

Production of a minimum cover design could be performed manually or by an existing computer program {e.g. DAPHNE (ref. 56) or TRRL (ref. 6)}. However MOD incorporates a minimum cover design procedure producing a minimum cover design for a network consisting of all possible intermediate manholes. The design method used is the Rational, with rainfall calculated by the modified Bilham formulae (ref. 5).

The heart of the program is the ISDP design of the network. This can only be performed using a computer due to the very large number of calculations involved (see 7.6.5). Computer storage and execution time become critical factors influencing the structure of the program and the choice of parameters for the optimisation.

The final part of MOD produces a fully consistent design for the network, based on the manhole positions and diameters chosen by the ISDP optimisation. This design essentially consists of finding the correct pipe gradients, the Rational method again being used.

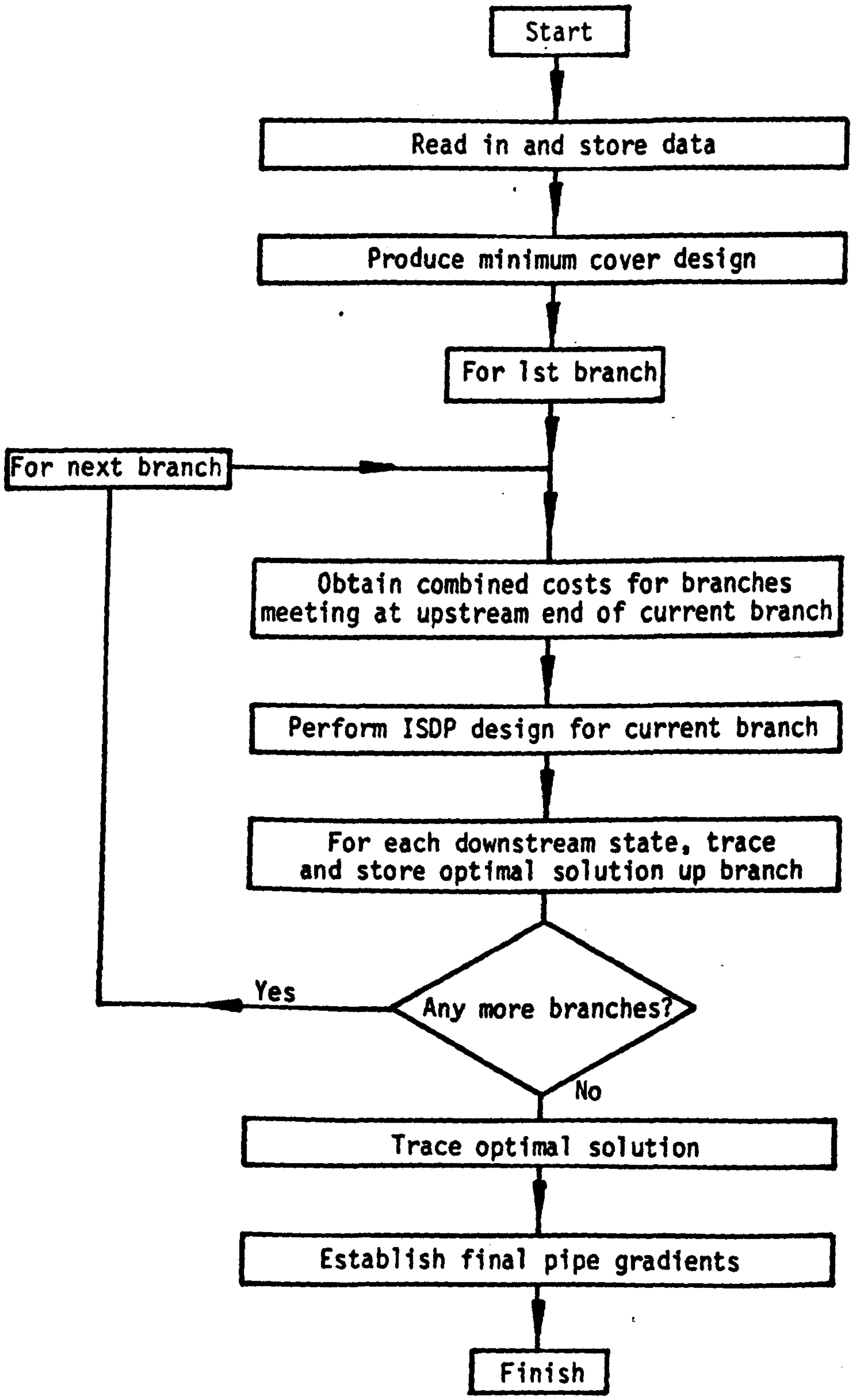
7.13.2 The program

MOD consists of approximately 830 lines of standard Fortran, there being a main program and thirteen subroutines. A listing of the full program is given in Appendix E.

A flow chart showing the principal features of the program is given in figure 7.10.

To minimise storage requirements most data is stored in four large arrays which are dynamically addressed, thus preventing large redundant areas of storage, and reducing memory requirements to a modest size for a main-frame computer.

The program was written in National Computer Centre (NCC) Standard Fortran (ref. 57) with further limitations imposed by HECB standards (ref. 58). This was to enable the final commercial version to be machine independent and fully transferable to any large computer. Many of the subroutines used in MOD were used in the commercial version and elsewhere. This does however incur some penalties in



FLOW CHART FOR PROGRAM MOD

FIGURE 7.10

terms of the number of lines of coding and the execution time of the program. These are, however, felt to be minor compared to the benefits of interchangeability.

7.13.3 Input

Input data is either generated by ASSEMB or can be created manually. In either case it is on cards (either real cards or a card image file) and has the same format as the output from ASSEMB (see 7.12.4).

7.13.4 Output

The output essentially consists of the final optimum design giving all final manhole places, pipe diameters, levels and gradients, together with flows, pipe capacities and details of cost.

Additional information is optionally available giving details of the initial minimum gradient design and details of the optimisation process but these were primarily for diagnosis in the event of failure during program development.

7.14 Results from using ASSEMB and MOD

7.14.1 Introduction

The computer runs using MOD may be divided into four groups:

a) checking that results are consistent with DPO
b) finding the effects of varying the optimising and design parameters
c) checking performance on various networks
d) preliminary investigations for the variable cross-drain problem

The results of a, b and c are presented and discussed below, and are tabulated in Table 7.1. Results from d will be presented and discussed in Chapter 8.

7.14.2 Checks on the consistency of MOD and DPO

Due to minor changes in the costing routines and in the method used to calculate rainfall, MOD could not be expected to be in full agreement with DPO.

Three examples using network 2 (fig. 7.6) were tested on MOD (see Table 7.1) and the results compared to those using DPO. A large measure of agreement was noted, with pipe diameters and manhole positions identical. There were small differences in pipe gradients

Network	Manhole Spacing SP (m)	Zone Depth RZ (m)	Number of Levels	Number of Diameters	Minimum Spacing (m)	Time of Entry te (mins)	Cost of Construction (£)	Execution Time (secs)
2	10	0.5	5	3	30	2	14064	19
2	30	0.5	5	3	30	2	14204	4
2	60	0.5	5	3	30	2	14329	1
3	120	0.5	5	3	30	2	108084	3
3	60	0.5	5	3	30	2	107450	6
3	30	0.5	5	3	30	2	107323	19
3	10	0.5	5	3	30	2	106572	140
3	30	1.0	11	3	30	2	106984	75
3	30	1.0	11	3	30	2	106693	72
3	30	0.5/1.5	6	3	30	2	107114	26
3	30	0.5/1.5	8	3	30	2	106864	42
3	30	0.5/1.5	11	3	30	2	106643	75
3	30	0.6/1.2	7	3	30	2	106693	33
3	30	2.0	21	3	30	2	106220	243
3	30	3.0	31	3	30	2	106220	518
3	10	2.0	21	3	30	2	105860	1884
3	120	0.6	7	2	30	2	108084	4
3	60	0.6	7	2	30	2	107121	9
3	30	0.6	7	2	30	2	107021	30
3	10	0.6	7	2	30	2	106725	223
3	30	1.5	16	3	60	2	106220	108
3	20	1.0	11	3	60	2	106902	107
3	20	1.0	11	3	60	4	104067	107
3	20	1.0	11	3	60	6	102028	107
3	20	1.0	11	3	60	8	100103	106
3	20	1.0	11	3	60	10	98769	109

TESTS USING MOD

TABLE 7.1

and costs, however, but these were thought acceptable. It was concluded that MOD was largely consistent with DPO.

7.14.3 The effect of varying the optimising parameters

The results of 23 runs using MOD on network 3 (figure 7.11) are tabulated in Table 7.1 Seventeen of these runs had identical design parameters, but varying optimising parameters. Execution times varied from 3 seconds to 1884 seconds with network construction costs ranging from £108100 to £105900 respectively, representing savings of from 2% to 4% over a likely minimum gradient solution (see fig. 7.12(b)) costing £110,100. A typical optimal solution is shown in fig. 7.12(a). Fig. 7.13 shows program execution time plotted against network costs on a logarithmic scale. The general trend shows that the optimal solution is approached as the time spent on computing increases. There must be an absolute value for the true optimal solution but this is unknown. If the intermediate manhole spacing was decreased to a very small distance and the spacing of levels greatly decreased whilst retaining a wide level zone this optimal solution would be approached. However, the execution time would be enormous and hence computing costs would be greatly in excess of any possible saving on construction cost.

On the other hand, very small execution times still show sizeable savings on construction costs, for negligible computing costs.

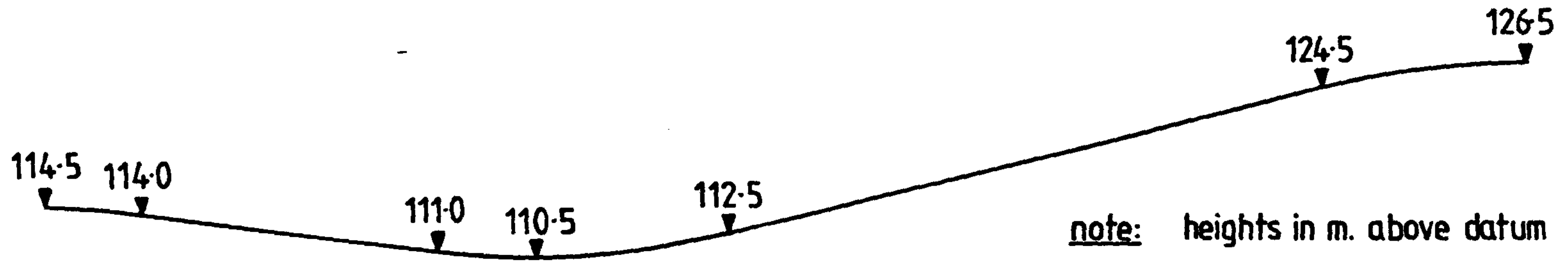
Obviously a balance must be achieved between computer costs and likely savings as explained in section 5.15.

Costs of running MOD on Liverpool University's I.C.L. 1906S computer work out at approximately £0·16 per second. Hence for a 100 second job the execution cost is £16.

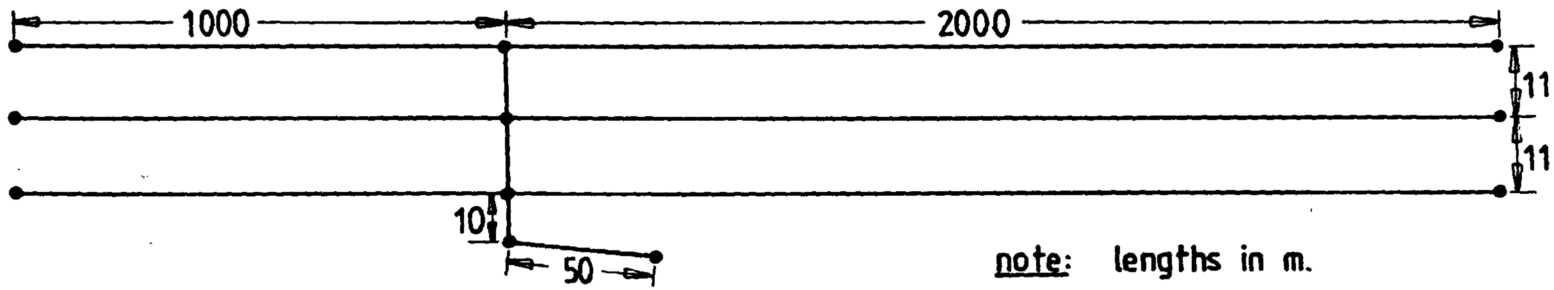
For practical considerations (see section 5.15) it was felt that the ratio of (extra saving on construction costs)/(extra computing costs) should not fall below about 20 in a commercial program. This, applied on a diminishing returns basis, leads to a maximum execution time of about 150 secs for this network on the 1906S.

This, then, forms a restraint on the selection of the optimising parameters. The parameters that effect the performance of the optimisation and the program execution time are:
a) spacing of possible intermediate manholes (SP)
b) number of discrete pipe levels considered (M)

FIGURE 7.11

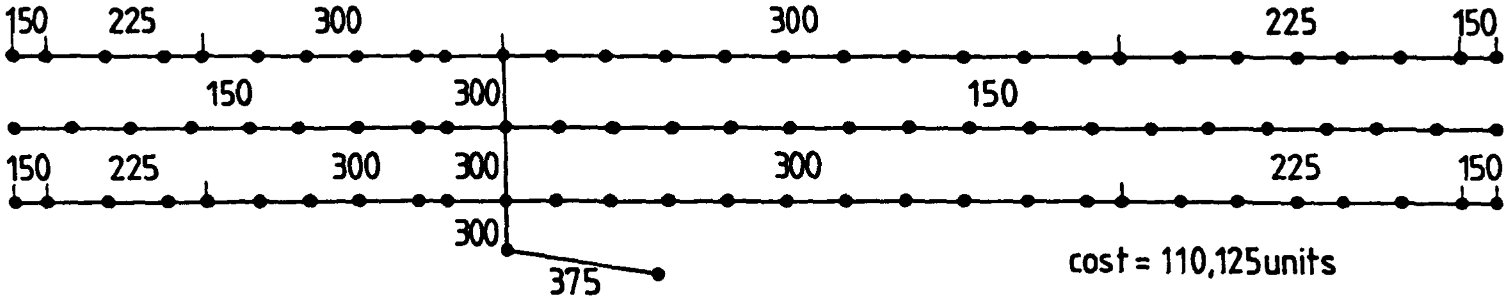


SECTION ALONG ROAD CENTRE-LINE

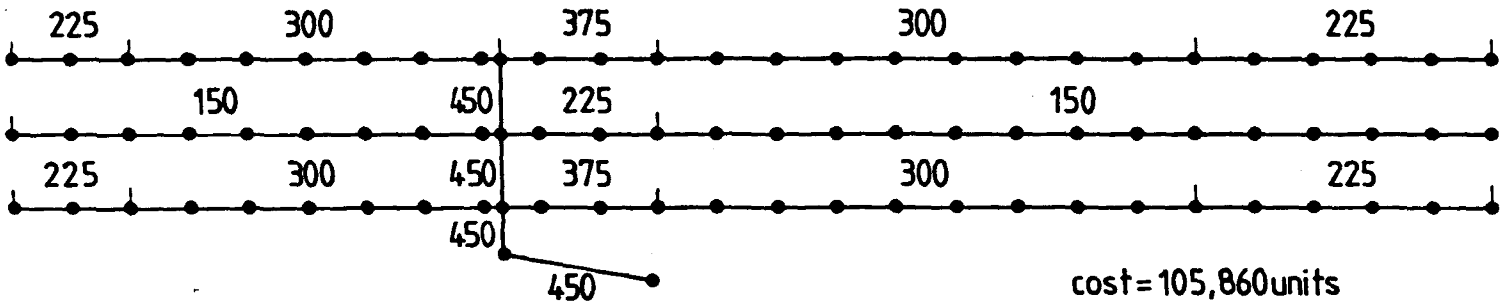


SKELETON PLAN FOR NETWORK 3

FIGURE 7.12

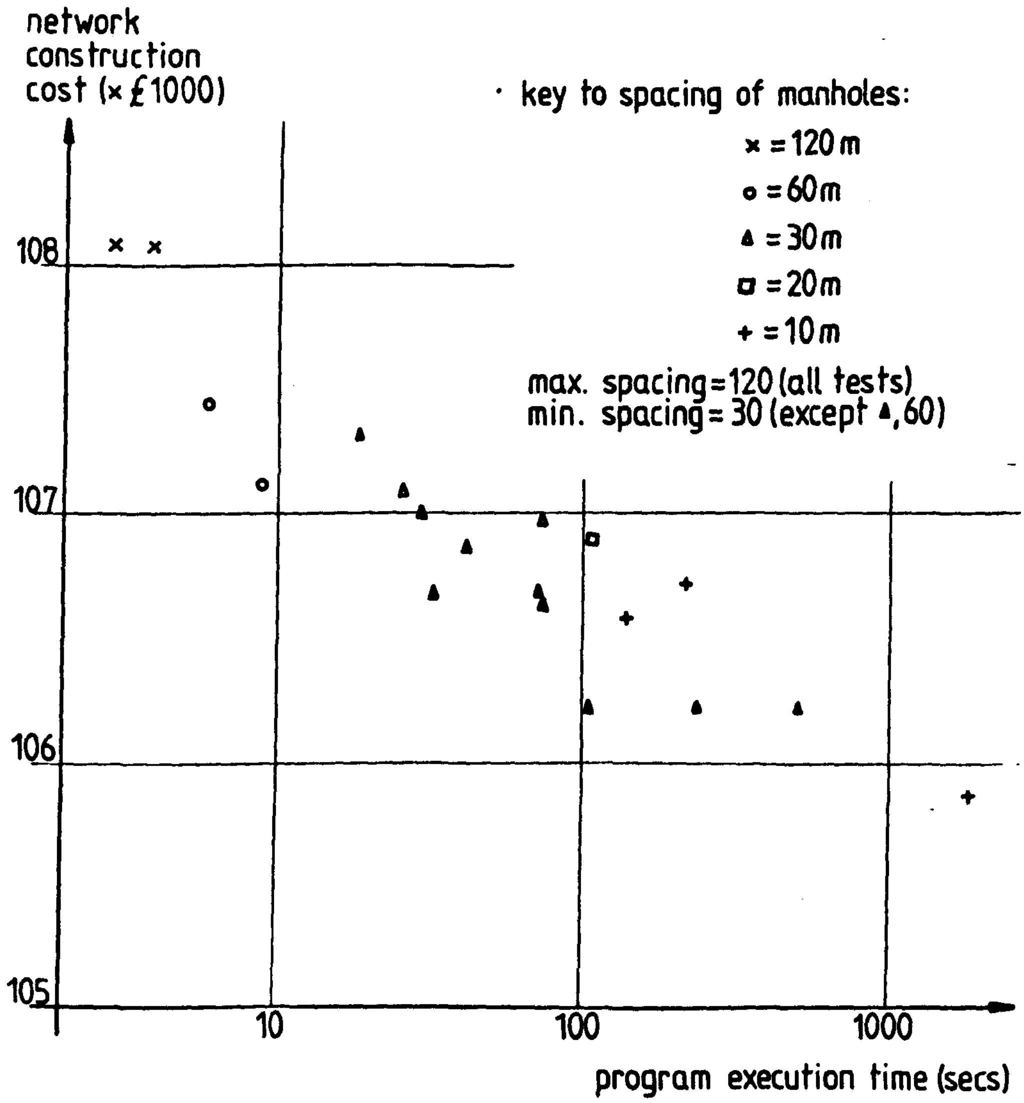


(a) A SOLUTION USING MOD



(b) A MINIMUM COVER SOLUTION

note: diameters in mm.



CONSTRUCTION COST vs EXECUTION TIME

FIGURE 7.13

c) range of pipe levels considered (RZ)
d) range of pipe diameters considered (J)

For the reasons described in section 7.10.3 (f) the spacing of discrete pipe levels was kept within the range 0.1 to 0.15, there being no apparent advantage in decreasing the spacing below this, and with increases of spacing likely to cause sub-optimal manhole positions and diameters. Hence M and RZ are linked by RZ $\doteqdot$ (M-1) × 0.1. Generally the range of diameters considered was kept to 3, but reduced to 2 for four of the runs.

Hence in practice SP and RZ become the only two important parameters.

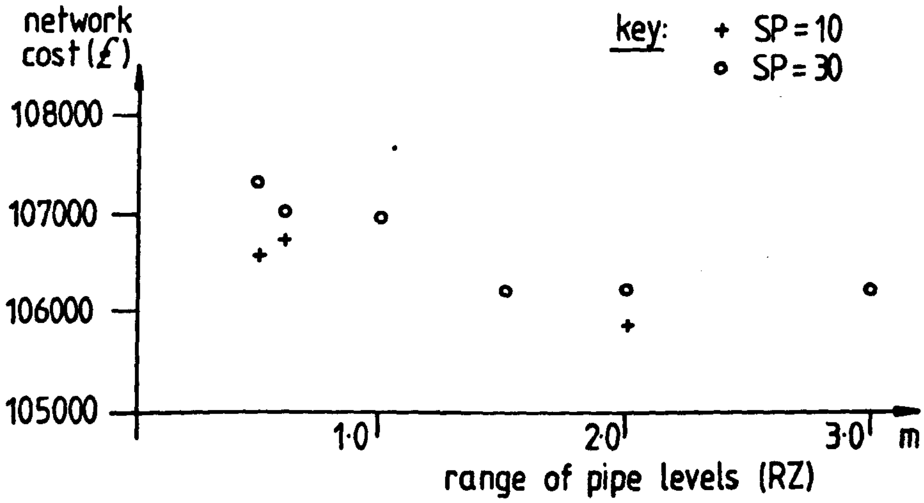
The effects of varying these are shown in figure 7.14. It can be seen that there appears to be a zone depth above which there is no further reduction in cost, this occurring at about 1.5 m. However, decreasing manhole spacing leads to increased savings with no such obvious limit.

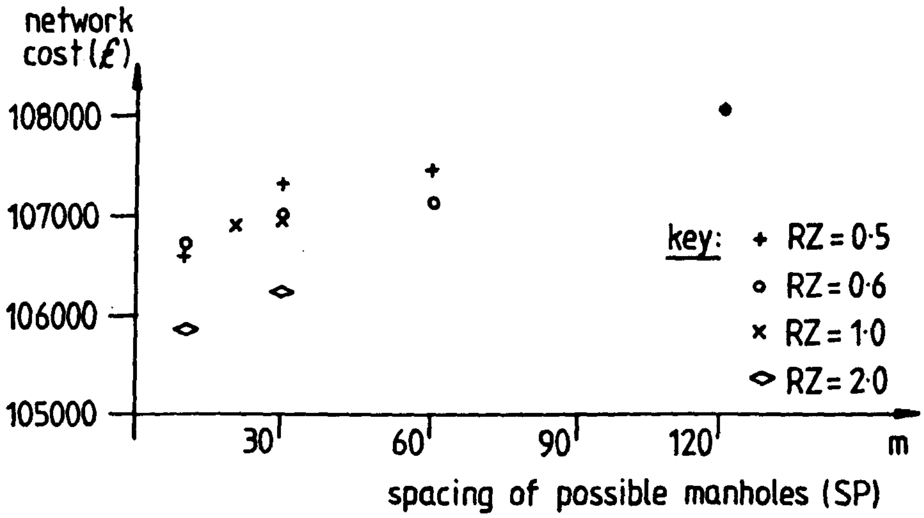
The results are combined in fig. 7.15(a), the costs being given as approximate contours. The line joining points with execution times of 150 secs is also sketched in. If the execution time is restricted to this value, the correct choice of parameters for the most effective use of computer time is a zone depth of about 1.5 m with a manhole spacing of about 30 m.

Several runs were performed (Table 7.1) with RZ locally widening towards the end of long pipe branches especially where these branches ended well above the level of the main pipe in the minimum cover design. Such runs were found to be more efficient than those using constant values of RZ, using generally about half the computer time to achieve the same cost savings.

Four runs were performed using only 2 possible pipe diameters. These showed only small savings in computer time. The penalties involved in using only 2 diameters were not large in this example, but neither were the advantages. Hence it was felt unnecessary and rather unwise to adopt less than 3 diameters in general.

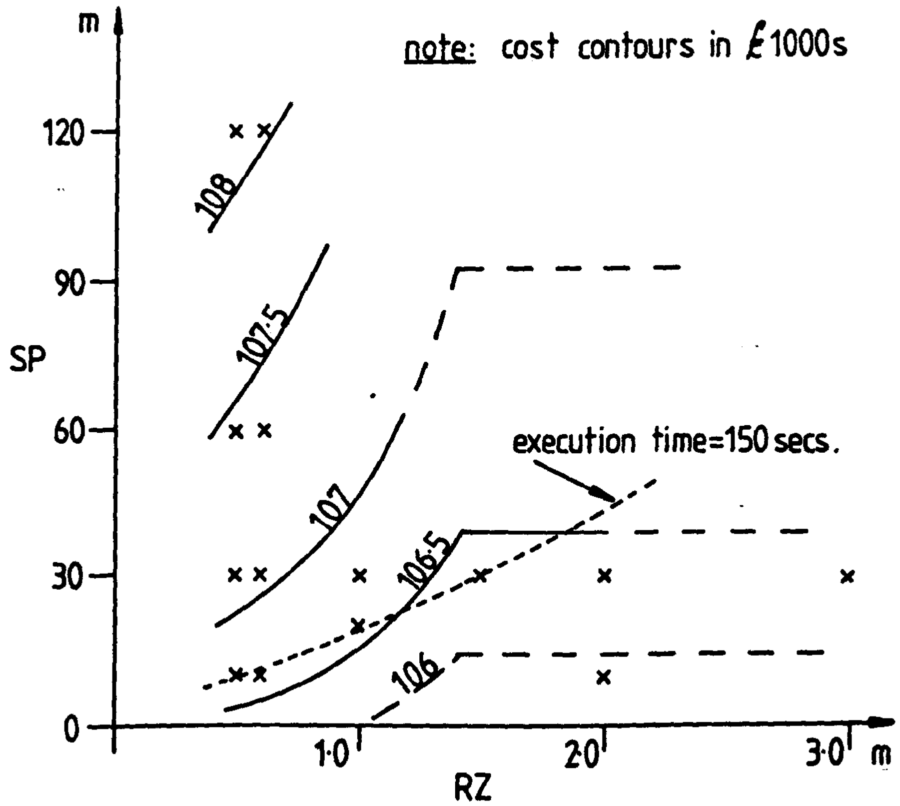
Most runs used a minimum manhole spacing of 30 m. The exceptions used a minimum spacing of 60 m and in the one case where comparison was possible this was rather more efficient, using less computer time than for an equivalent run with a 30 m minimum spacing and no cost penalty.



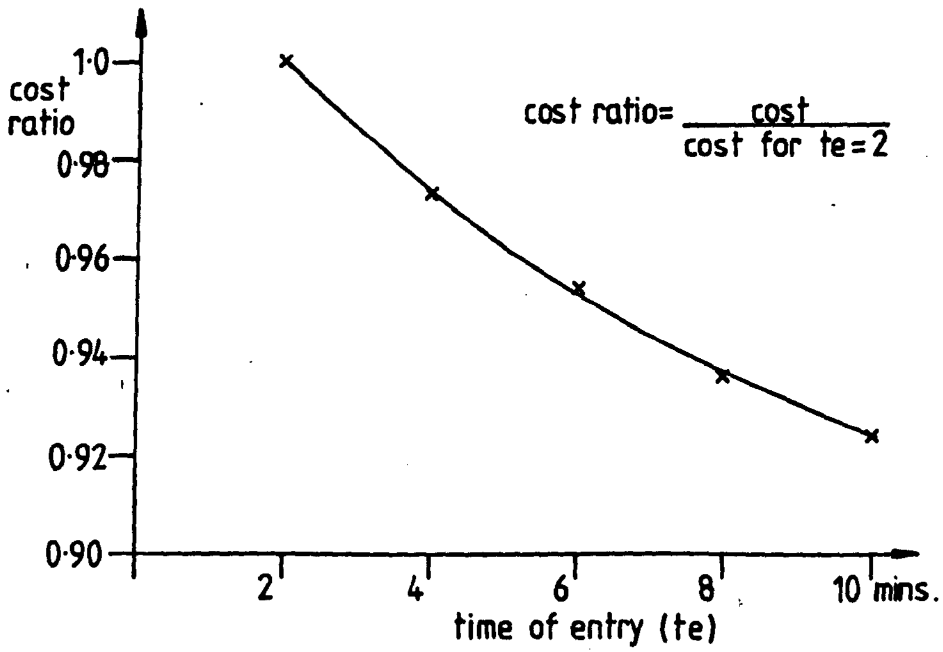


SENSITIVITY OF NETWORK COST TO OPTIMISING PARAMETERS: PROGRAM MOD

FIGURE 7.14



(a) NETWORK COSTS FOR COMBINATION OF OPTIMISING PARAMETERS



(b) SENSITIVITY OF NETWORK COST TO TIME OF ENTRY

FIGURE 7.15

7.14.4 Varying the design parameters

Having established an optimal design program for drainage design it is relatively easy and informative to find the effects on the design and cost of construction of varying certain of the design parameters.

Such parameters, the values of which are at present selected on a rather arbitrary basis, could include minimum cover over the pipe, minimum slope, pipe roughness, maximum manhole spacing, storm return period and time of entry of runoff into the pipe system. Such a study was not the objective of this research, but five runs were performed on Network 3 (fig. 7.11) using varying times of entry (te) to the pipe system (see Rational Method, section 5.11.2) as it was felt that this may be greatly underestimated in current design practice.

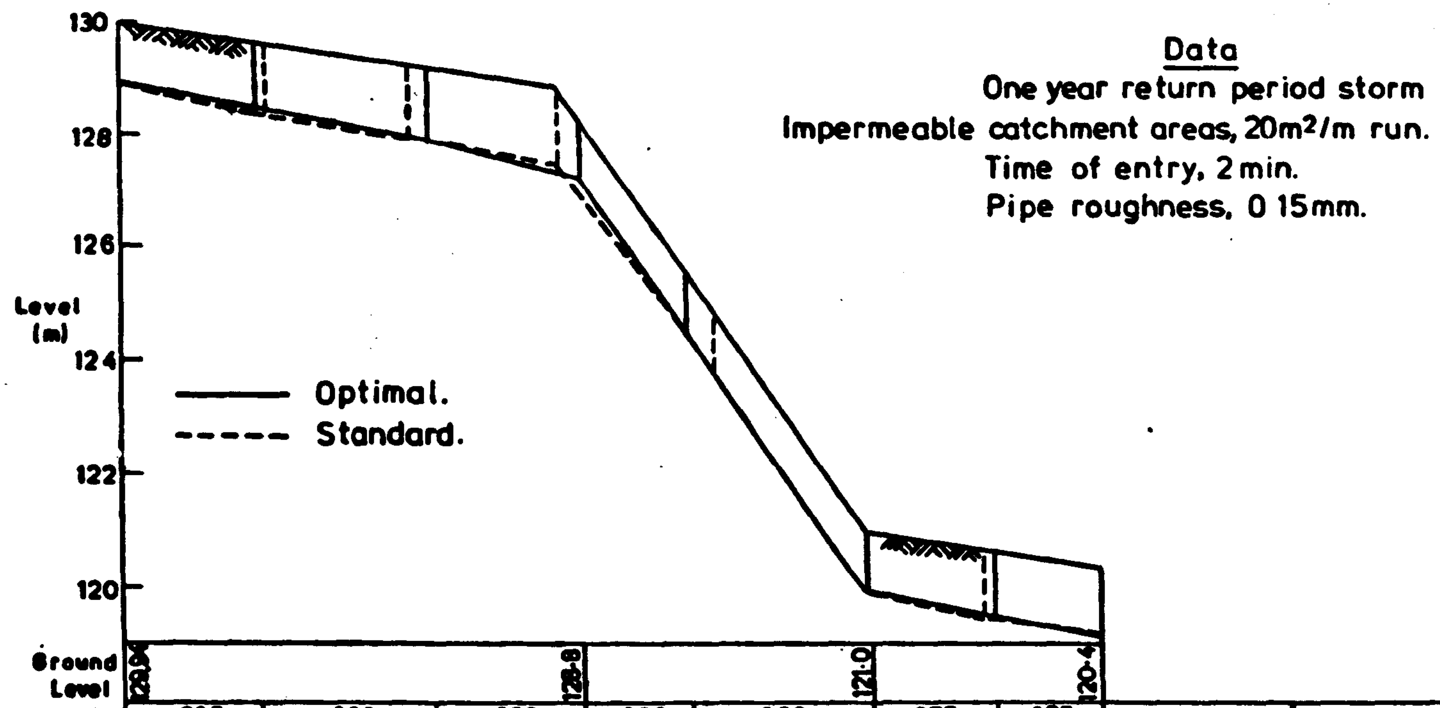
At present te is generally taken to be 2 minutes. Fig. 7.15(b) shows the effect on the network cost of taking te equal to 2,4,6,8 and 10 minutes, resulting in reductions in network cost of up to 7.5%.

7.14.5 Tests on other networks

MOD was used on rather more complicated road drainage networks similar to Network 3 but with one or two additional cross-drains. These runs were primarily to investigate the possibilities of variable cross-drain optimisation and will not be discussed here except to say that in all cases MOD produced sensible results, with rather greater cost savings than for Network 3.

In addition one further design network was used as an example. This consisted of a single run as shown in figure 7.16. The plan length of pipe run between the two fixed position manholes at the ends was 840 m. The ground profile has three regions each at a different linear slope. Possible manhole positions were defined at 10 m spacing along the pipe run with a maximum permissible distance of 150 m between manholes. A 1 m zone was used for the levels with 11 discrete pipe levels at 0.1 m spacing. Three pipe diameters were used at each location.

The optimal design uses 6 intermediate manholes placed at the chainages shown. The cost of this optimum design is 12493 units and took about 120 secs execution time on the Liverpool 1906S computer.



ø	225	300	300	300	300	375	375	Optimum Design	Cost = 12493
Slope	0·0040	0·0040	0·0051	0·0300	0·0300	0·040	0·0044		
Length	150	150	130	90	150	110	90		

ø	225	300	375	375	375	375	450	'Standard' Design	Cost = 14491
Slope	0·0042	0·0040	0·0040	0·0268	0·0300	0·0043	0·0040		
Length	126·67	126·67	126·67	130	130	100	100		

0 500 1000 Chainage (m)

DESIGN EXAMPLE

FIGURE 7.16

In order to compare it with a traditional manual design a standard design was produced and costed by hand. The definition of such a standard design is, as in all variable plan problems, rather subjective. However, the following logical stages were taken:

a) place manholes at both ground discontinuities;

b) place manholes at equal spacings in each sloping portion subject to a maximum spacing of 150 m;

c) design pipes at minimum cover subject to a minimum gradient requirement.

Using the same cost function as for the optimal design the standard design cost was 14491 units, i.e. 16% more expensive than the computed optimal.

This example demonstrates that considerable savings can be made by making relatively small adjustments to manhole positions and gradients. Most of the saving in this example is effected by reducing the pipe diameter in the central region from 375 mm to 300 mm, and by reducing the final pipe in the run from 450 mm to 375 mm.

7.15 Conclusions from using MOD

Using MOD on the large and realistic road drainage example of Network 3 showed two important differences to the results obtained from using DPO on preliminary examples.

These were firstly that the savings likely to be achieved over a sensible minimum gradient design were substantially less than expected. They were approximately 3% to 4% as opposed to the 5% to 15% expected.

Secondly the optimal solution could only be obtained by taking a range of depths of about 1.5 m or more, as opposed to the expected range of about 0.6 m.

The first point could be explained by the long lengths of gently sloping ground profiles, typical of a road carriageway, and the small number of pipe intersections. In fact in examples involving additional cross-drains MOD produced rather larger cost savings. Hence it would seem likely that the most spectacular results are achieved with rapidly varying ground levels and complicated pipe networks.

The second point is again probably explained by the long lengths of the runs as opposed to the relatively short lengths in previous examples, or perhaps just to the overall larger size of the network.

It may make sense to select the range of depth considered according to the maximum drainage path through the network, or have a variable range, the range at a manhole being a function of the length of the longest drainage path upstream of that manhole. This could be achieved by having a variable number of discrete levels, but the data handling routines would be rather more complicated than in MOD, and extra data would be required to define the number of levels at each point in the network. Nevertheless it remains a sensible approach for further investigation.

The choice of parameters to obtain the best solution for a reasonable outlay in computer resources is not obvious, and varies depending on the example. It seems reasonably clear that only 3 pipe diameters need be considered provided that the available diameters are in the standard 75 mm nominal increments, and that the original minimum gradient design produced does not result in artificially large diameters due to the use of a very small minimum gradient (less than, say 1 in 300).

Based on the information to date, a range of depths of at least 1.5 m is needed to ensure that the optimal design is not excluded. As 0.15 m is about the maximum spacing of discrete levels allowable this requires 11 discrete levels to ensure the solution is reasonably close to the optimal. This conclusion will need checking in the light of further experience in using MOD.

Having fixed the number of pipe diameters and suggested the range of levels and their spacing, it remains only to fix the spacing of intermediate manholes. The cost of the solution will continue to decrease as the manhole spacing decreases towards zero, whilst the computer costs involved rise rapidly. It is probably not worth while decreasing the spacing below 10 m, both on economic grounds and because of the desirability of keeping manholes at convenient chainages (see section 7.7), and spacings of 30 m or even 60 m may give reasonable answers with more acceptable computer execution times. Note that these figures relate to a maximum manhole spacing of 120 m, and are convenient fractions of 120 m. For another maximum manhole spacing other similar fractions of distance would be more appropriate.

The most important conclusion relating to the choice of parameters must however be that whatever computer program is developed for the MANVAR system, it must be flexible enough to incorporate changes in these parameters as experience is built up of their use.

It remains very clear that even though savings may not be as great as previously expected, large sums of money on construction costs can be saved by a very small outlay in computer time. This will always be worthwhile. Greater outlay on computing will save larger amounts on the construction costs. The extent of investment in computing time to obtain more savings in construction is then a matter of policy for those in charge of the design procedure, and may be controlled by careful selection of the optimising parameters.

7.16 A commercial program

7.16.1 Introduction

The funding that enabled the bulk of this research to take place was provided by the Department of Transport, Highway Engineering Computer Branch.

Their principal requirement was the production of a fully commercial optimal drainage design program for roads based on their existing DAPHNE highway drainage design program (ref. 56).

For reasons described in Chapter 8, it was decided that this program should be based on the MANVAR model and not on the CROSSVAR (variable cross drain) model described in that chapter.

For convenience the program will be referred to here as DAPHOP, although when released it will probably be as a user-selected option of a new version of DAPHNE.

7.16.2 The existing program DAPHNE

DAPHNE consists of approximately 8500 lines of Standard Fortran. Much of the coding is required for handling, interpreting and checking input data and outputting results and messages.

DAPHNE uses data which is already available in the form of computer files to define all the road geometry (alignment, crossfalls etc.), this information being available from running the BIPS suite of programs (ref. 59) for the design of highways. The DAPHNE user then defines the drainage network he requires, including all manhole positions, and the design parameters he wishes to use. DAPHNE then calculates all catchment areas, calculates design flows according to the Rational method and designs all pipe diameters with pipes at minimum possible cover.

7.16.3 The optimising version: DAPHOP

DAPHOP is structured on the MANVAR model, the details being shown in figure 7.17. Basically DAPHOP and the joint ASSEMB and MOD programs, are very similar except that DAPHOP uses the existing DAPHNE routines to establish a minimum cover design and to perform the final design of pipe gradients. The efficiency of some of the MOD routines and the data handling were improved before incorporation into DAPHOP, resulting in a generally more compact and efficient program than would otherwise have been possible.

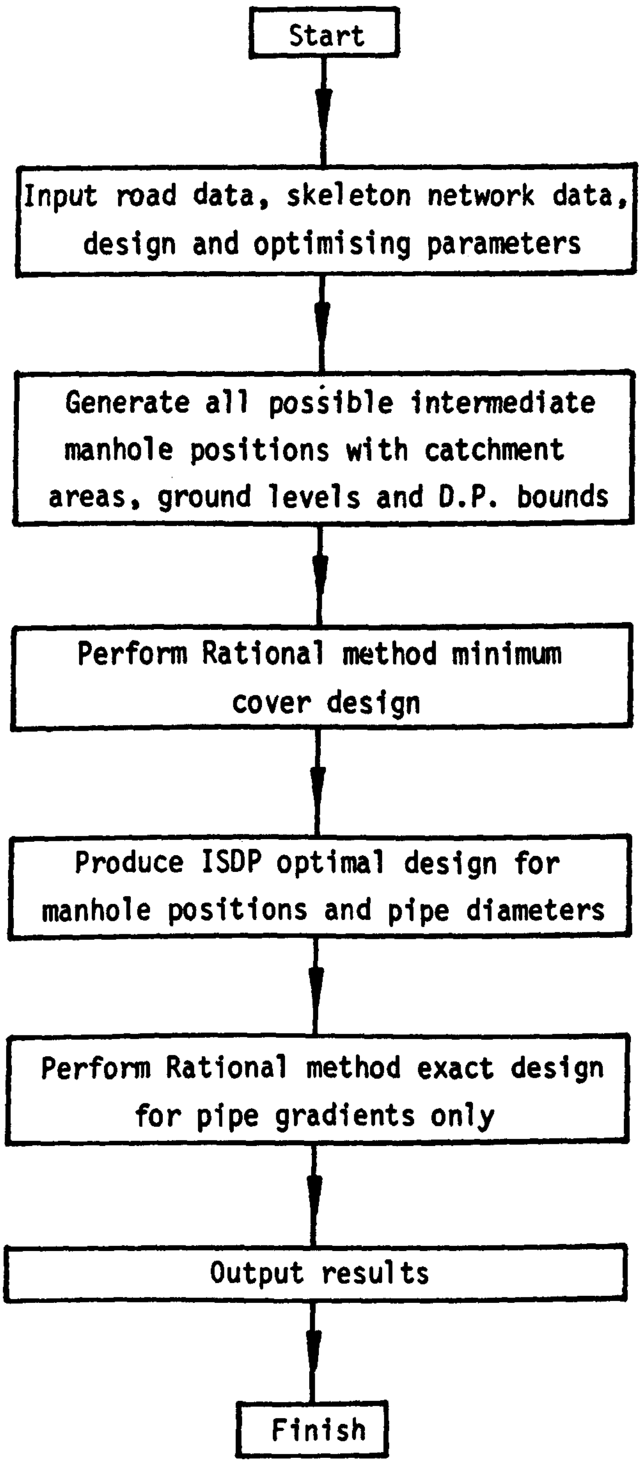
DAPHOP is at present undergoing trials with the DOT before being released for general use.

7.17 Conclusions on the MANVAR model

Experience has shown that the MANVAR model is fully practicable as a storm drainage design program for networks in which there are branches along which unknown numbers of intermediate manholes are to be placed. This type of network is typical of highway drainage.

The extent to which the MANVAR produced design approaches the true optimal solution is dependent on the choice of parameters in the optimising routines. The choice of parameters determines the cost of running MANVAR and may be limited by the size of available computing memory.

Significant savings can always be made by adopting very coarse parameters (e.g. a zone depth of 0.6 m and a manhole spacing of 60 m) at minimal computing costs. Larger investment in computing will result in larger cost savings, there being a practical rather than a theoretical limitation on this.



STRUCTURE OF DAPHOP

FIGURE 7.17

CHAPTER 8

THE VARIABLE CROSSDRAIN POSITION MODEL : CROSSVAR

8.1 Introduction

8.2 Defining the Problem

8.3 Methods of Approach

8.4 Fibonacci Search

8.4.1 Trial using Fibonaci search

8.4.2 Results

8.4.3 Conclusions

8.5 Polytope Search

8.5.1 Introduction

8.5.2 Results using a polytope search

8.5.3 Conclusions

8.6 Dynamic Programming Approach

8.7 Variable Cross-drains and the Modified Serial System

8.8 Applying ISDP to the Variable Cross-drain Problem

8.9 The Design of a Stage

8.10 Establishing the Ranges of Value for the State Variables

8.11 Design Flows

8.12 Cross-drain Sets with Networks Upstream

8.13 Cross-drain Sets Sharing a Common Base Cross-drain

8.14 A Practicable Model : CROSSVAR

8.14.1 Introduction

8.14.2 Structure of CROSSVAR

8.15 Program MODEX

8.16 Optimising Parameters for CROSSVAR

8.17 Program of Testing for CROSSVAR

8.18 Results Using the CROSSVAR Model

8.18.1 Checking CROSSVAR with previous results

8.18.2 Finding typical cross-drain spacings

8.18.3 Stability of cross-drain positions to variation of parameters

8.18.4 The effect of cross-drain resolution on the optimality of the solution

8.18.5 Runs using other networks

8.19 Choice of Values of the Optimising Parameters

8.20 Conclusions on the Use of CROSSVAR

Chapter 8. The Variable Cross-Drain Position Model - CROSSVAR

8.1 Introduction

In this chapter the second of the variable plan optimisation models for road drainage design is presented. This involves the determination of the number and position of cross-drains in a storm water drainage network for highways. As outlined in Section 6.3, this essentially completes the optimal design process for such networks.

8.2 Defining the Problem.

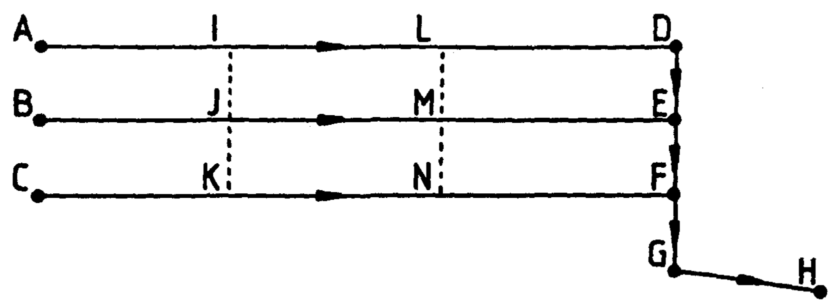
A typical network (Fig. 8.1) consists of carriageway drains (AD, BE, CF) connected to a base cross-drain (DEF) connected in turn to carrier drains (FG, GH).

However a number of additional cross-drains could be added (eg.IJK,LMN) thus diverting the flow from AD to AIJKF and ILMNF. Drains IL, LD will have zero flow at their upstream ends but will collect flow along their lengths.

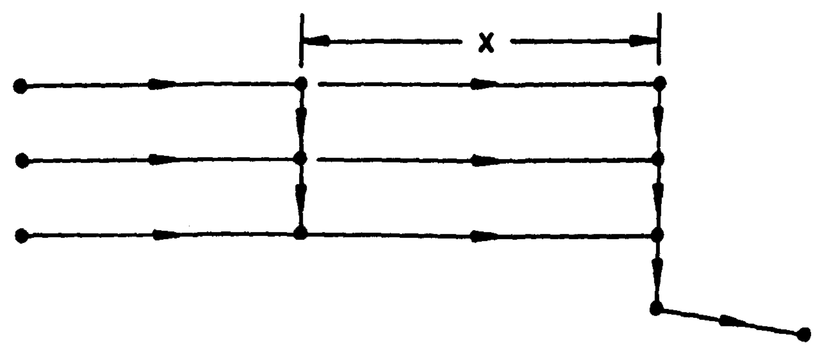
Flow along BE is similarly diverted. The overall result is for the drains KN, NF to be substantially larger than CF was, and for IL, LD, JM, ME to be smaller. This may well be cheaper to construct than the original basic layout.

The problem is thus to find the number of cross-drains and their positions that will result in the network of minimum construction cost.

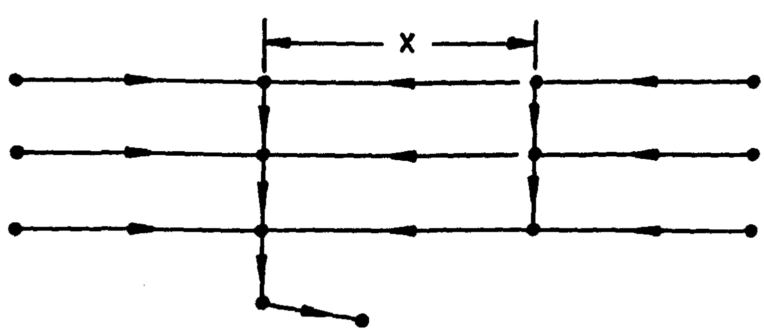
To make the model complete, it is necessary to find the number and positions of all intermediate manholes for each pipe run such as AI and the diameters and slopes of all pipes. This can be accomplished by incorporating the MANVAR model (see Chapter 7) into the present model.



(a) TYPICAL NETWORK



(b) POSITION OF SINGLE CROSS-DRAIN



(c) FIBONACCI SEARCH: NETWORK 3

FIGURE 8.1

8.3 Methods of approach.

Several different approaches were investigated in preliminary studies of the problem before D.P. was again selected as being the most likely contender. These preliminary studies used the MANVAR model as a step in a search procedure for the optimal solution and are described in the following sections.

8.4 Fibonacci Search.

If the problem is simplified greatly by assuming that only one additional cross-drain is required for the optimal solution, the well-known Fibonacci Search method (ref. 60) can be employed.

Defining the distance of the additional cross-drain from the base cross-drain as x (Fig. 8.1) the total network cost can be expressed as f(x) where f(x) can be evaluated for any feasible x by running the MANVAR model. It is then necessary to adopt a one-dimensional search technique to find the position of x that makes f(x) a minimum. It is necessary to make two assumptions, firstly, that there is a value of x that minimises f(x) within the range of x considered, and secondly that f(x) is unimodal within this range. It can then be shown (ref.61) that no grid search technique can be guaranteed to find the minimum in less function evaluations than the Fibonacci method.

Essentially the method consists of the following steps:

i) consider a set of positions x covering the range of interest.

ii) evaluate f(x) at a specified pair of points x_1, x_2.

iii) as f(x) is unimodal, from the values of $f(x_1)$ and $f(x_2)$ determine whether the value of x that minimises f(x) is $> x_1$ or $< x_2$. This narrows the range of x that need be considered.

iv) with reduced range, and knowing one value of f(x) within this range (either $f(x_1)$ or $f(x_2)$) determine f(x) at a specified point x_3 and repeat the process.

The points x_1, x_2, x_3 etc. are determined by reference to the Fibonacci series of numbers 0, 1, 1, 2, 3, 5, 8, 13, 21,34 etc.

In this way the minimum value of f(x) can be found for a large number of possible positions of x with the minimum number of evaluations. e.g. for 88 positions of x, only 9 function evaluations need be made.

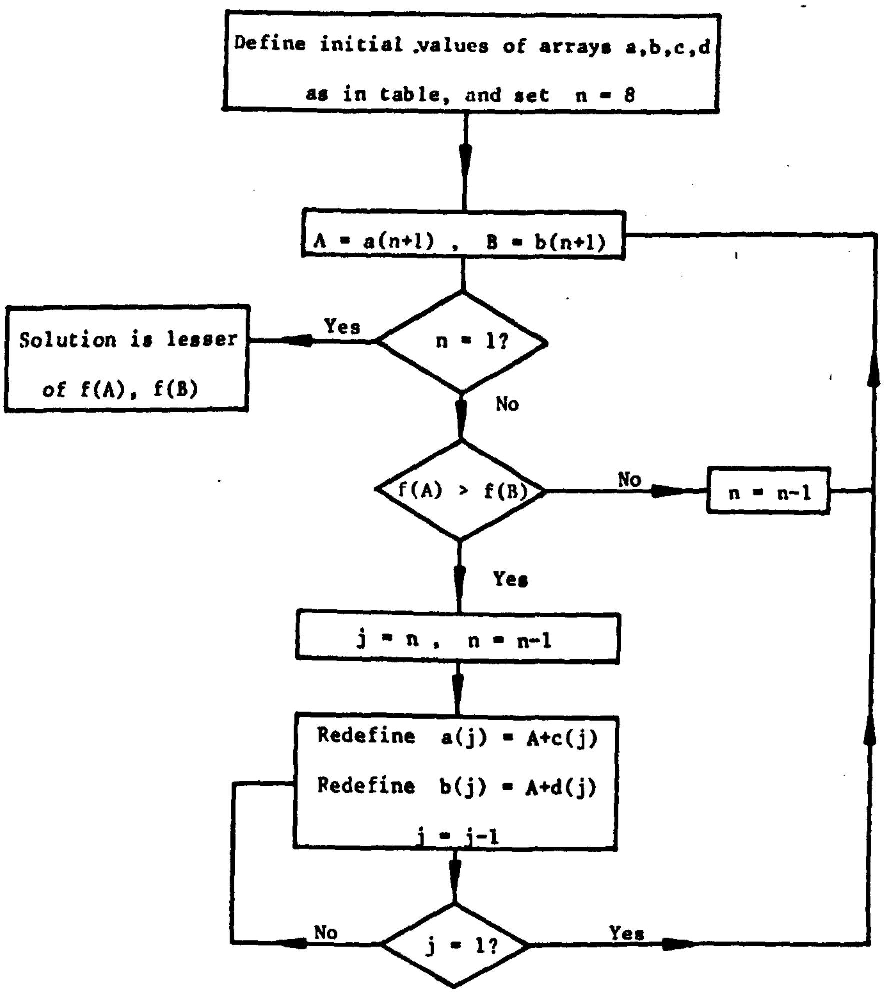
8.41 Trial Using Fibonacci Search.

An attempt was made to apply this technique to finding the optimum position of an additional cross-drain for Network 3 (fig. 7.11). 88 points were used, thus requiring a total of 9 evaluations. For this preliminary work, the MANVAR programs were used, with each evaluation requiring a new run using a manually altered set of input data.

Defining x as in Fig.8.1c, the problem can be stated as follows: Find the value of x that minimises the construction cost of the network, where $0 < x < 2000$.

It was originally thought that x would be about half way along the 2000 m long parallel drainage runs. Hence the search was restricted to the region $560 \leq x \leq 1430$ Using a 10 m grid interval allows 88 possible positions for x, varying from point (1), x = 560 m to point (88), x = 1430 m. In general for point n, $x = 550 + 10\ n$.

Assuming that the cost function is unimodal and has a minimum value within this range, the optimum value of x should then be obtained in 9 function evaluations, i.e. 9 runs of the MANVAR model, using networks defined by different positions of x. The choice of points at which to evaluate f(x) is shown in Fig. 8.2.



	Initial Value for Element								
Array	1	2	3	4	5	6	7	8	9
a	1	1	2	3	5	8	13	21	34
b	-	2	3	5	8	13	21	34	55
c	-	1	2	3	5	8	13	21	34
d	-	2	3	5	8	13	21	34	55

FIBONACCI SEARCH OVER 88 INTERNAL POINTS

FIGURE 8.2

8.4.2 Results.

The results of applying Fibonacci search to the region $560 \leq x \leq 1430$ are shown in Table 8.1.

After just four evaluations it became clear by plotting a graph of the points (Fig.8.3) that the optimal solutions could well lie outside the range being investigated, thus invalidating the search technique.

Table 8.1 Fibonacci Search.

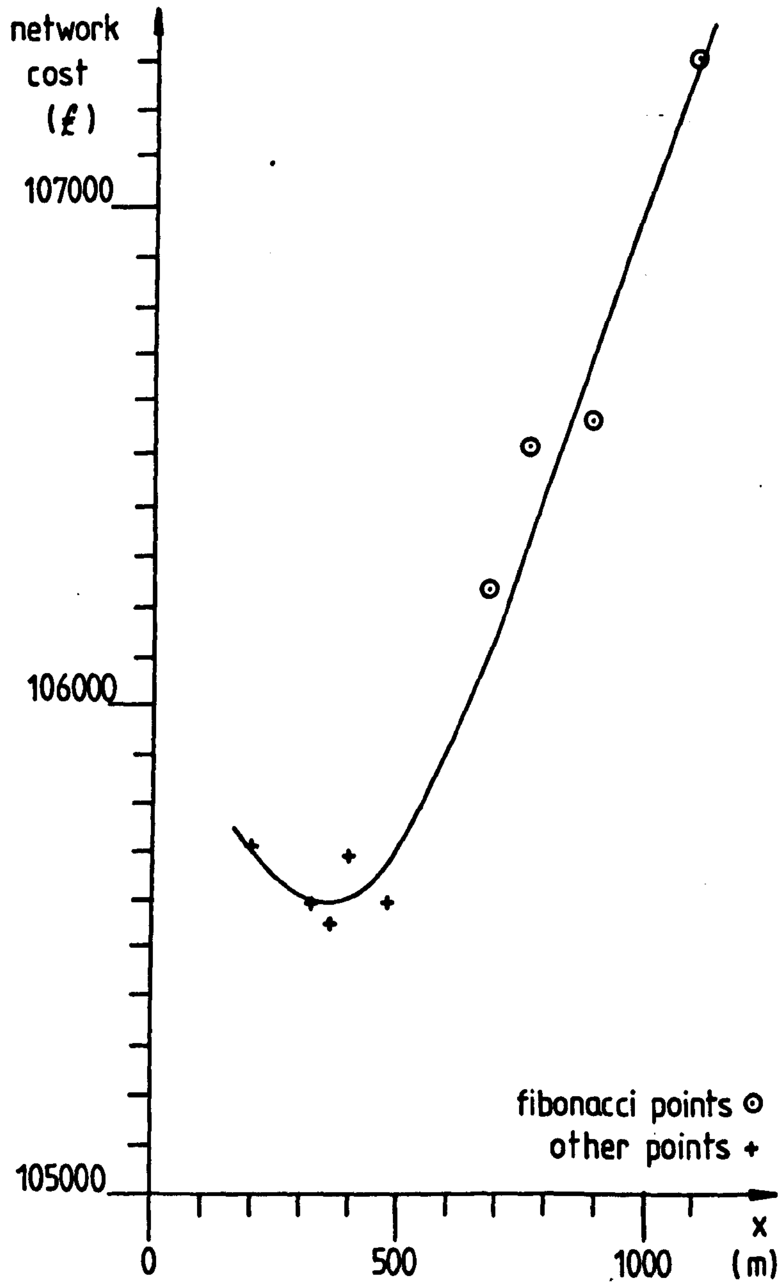
Pt.	x	Cost	
55	1100	£107297	} Solution in range 1 to 55
34	890	£106561	}
21	760	£106512	∴ Solution in range 1 to 34
13	680	£106237	∴ Solution in range 1 to 21

Consequently further values of x were used outside the range initially considered. These are tabulated in Table 8.2 in chronoligical order and plotted also on fig. 8.3.

Table 8.2

x	Cost
400	£105689
200	£105710
320	£105598
480	£105602
360	£105555

Two results are of immediate interest. Firstly the assumption of the likely position of the cross-drain was erroneous, and secondly the assumption of unimodal behaviour is also invalid, at least in close proximity to the optimal solution.



OPTIMAL POSITION OF A SINGLE CROSS-DRAIN

FIGURE 8.3

A comparison with two other solutions is also informative. Firstly if no additional cross-drain is used, the optimal solution using the same parameters as for the results above would cost £106 220 (£665 or 0·6% more expensive than the cheapest single cross-drain solution)

Secondly a manual, minimum cover design, with one additional cross-drain placed at x = 960 would cost £111 533 (£5978 or 5·7% more expensive). This represents a typical solution using current design practice.

8.4.3 Conclusions.

Had the Fibonacci search covered the whole of the range 0 < x < 2000 it would probably have attained the optimal solution with a similar number of function evaluations as were actually employed using the graphical plot as a guide.

However as the function cannot be relied on to be unimodal, the technique may have foundered, and it is felt that it is therefore insuff iciently robust to be of general use in this application.

The optimal position of the cross-drain, being only 320 m upstream of the base cross-drain, suggests that several more cross-drains at similar spacings may be required for a truly optimal solution. Hence a simple univariable search is probably not appropriate.

8.5 Polytope Search.

8.5.1 Intrdouction.

As the Fibonacci method was limited to the possibility of a single cross-drain it was clearly of limited use, some more general method being desirable.

The nature of the objective function rules out almost all the recognised multivariable optimisation algorithms with the exception of the polytope search technique - sometimes known as the simplex method (ref. 60)

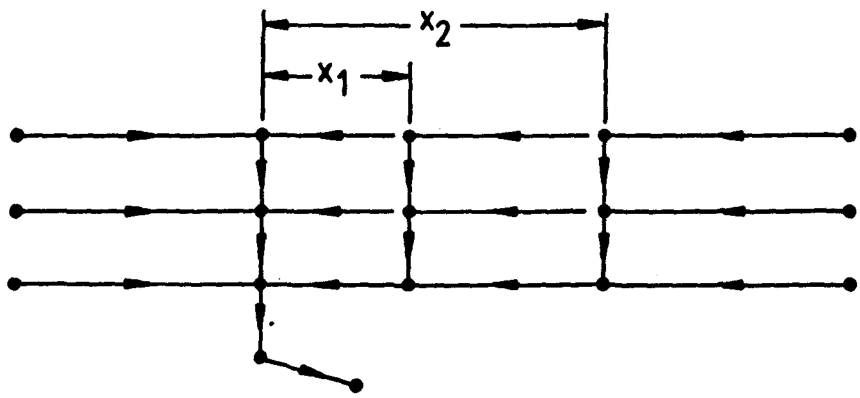
A polytope consists of a pattern of at least (n+1) points defining a non-zero volume in n-dimensional space. Hence if there are n variables in the problem, the first step is to evaluate the function at (n+1) appropriate points. The highest of these values is then discarded, and the function is evaluated at a new point, this being the reflection of the discarded point about the centroid of the remaining points. A new polytope is thus formed, the highest value of the function again being discarded and the process repeated.

Various rules can be applied to increase or decrease the size of the polytope and to deal with constraints on the position of the vertices.

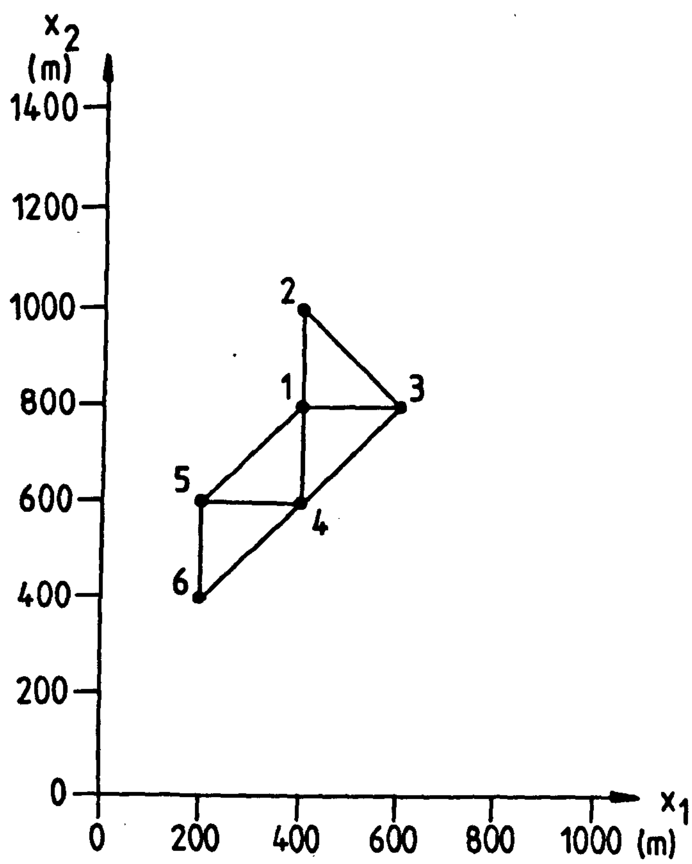
8.5.2 Results using a Polytope search.

This technique was applied to finding the optimum positions of two cross-drains for Network 3 (fig. 8.4a). A simple two-dimensional simplex was used. As it was felt desirable to keep cross-drains to sensible chainages, (e.g. multiples of 100 m) a right-angled isosceles triangle was used (fig. 8.4b). The method was applied manually using MANVAR to evaluate the function at the chosen vertices.

Vertices (1) (2) and (3) (fig.8.4b) were evaluated initially. The results of all evaluations are given in Table 8.3. As point (2) gave the highest cost this vertex should have been reflected about the mid-point of (1)-(3) to give the new vertex. However this would have created a vertex at pt.(600,600), giving coincident positions for the 2 cross-drains. Hence the nearest position which preserved the shape of the polytope was chosen for the new vertex, this being point 4.



(a)



POLYTOPE VERTICES

(b)

<u>POLYTOPE SEARCH FOR TWO CROSS-DRAINS</u>

<u>FIGURE 8.4</u>

Evaluation of vertex (4) identified vertex (3) as having the current highest cost. This vertex was then reflected about the centre of (1)-(4) and the new vertex (5) was identified and evaluated.

Of the current vertices (1, 4 and 5), (1) was the most expensive. Hence a vertex at (6) was identified and evaluated.

Of the current vertices ((4) (5) and (6)), (5) was the most expensive. As the next vertex would have been point (400,400) giving coincident cross-drains, and there was no other new point available for a vertex whilst preserving the same size and shape of polytope, the logical next step would have been to decrease the polytope size and to continue the search. The search was, however, terminated here as it was felt that sufficient information had been gained about the method. At termination the cheapest vertex was (4) having cross-drains at 400 and 600 m from the base crossdrain.

Table 8.3 Polytope Search

Vertex	x_1	x_2	Cost
1	400	800	106271
2	400	1000	106841
3	600	800	106367
4	400	600	105926
5	200	600	106249
6	200	400	105944

8.5.3 Conclusions.

The results confirmed one finding of the one dimensional search, namely that the optimal number of cross-drains was not obvious and could be quite large, due to the close spacing of the optimal cross-drains in the cases investigated.

Thus for a complete optimisation, polytope searches for 2, 3, 4 etc. variable cross-drain positions would have to be implemented. Although such a procedure is possible for up to about 6 variables using the Polytope method it would require a large number of function evaluations and be very inefficient.

Hence it was felt that although a Polytope search could be applied if the number of cross drains were known, the technique was not suited for the present case.

8.6 Dynamic Programming Approach.

It had become clear that any method which relied on the number of cross-drains being predetermined was of little practicable use, as results indicated that several cross-drains were normally economical, rather than zero, one or two.

The research done on the variable manhole position problem (Chapter 7) had shown that I.S.D.P. was capable of handling a similar situation and it was decided to investigate whether I.S.D.P. could be applied again.

8.7 Variable cross-drains and the modified serial system.

Section 7.6.2 introduces the concept of a modified serial system, essential for the implementation of I.S.D.P. Consider now a typical length of highway drainage consisting of a base cross-drain fixed in position into which run three roughly parallel carriageway drains. (Fig.8.5a)

It is possible to define a modified serial system in the following way.

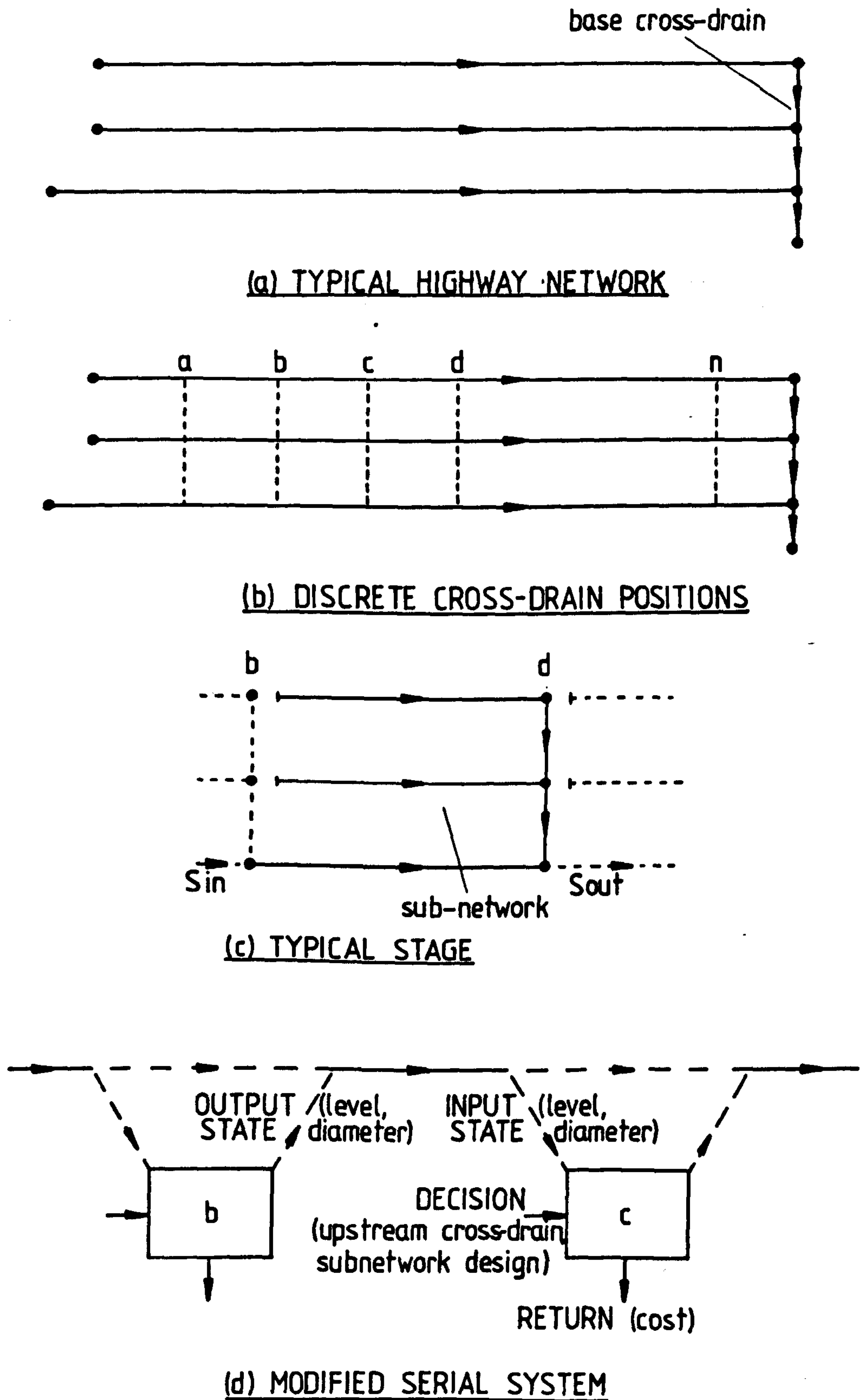

base cross-drain
(a) TYPICAL HIGHWAY NETWORK
a
b
c
d
n
(b) DISCRETE CROSS-DRAIN POSITIONS
b
d
Sin
Sout
sub-network
(c) TYPICAL STAGE
OUTPUT STATE
(level, diameter)
INPUT STATE
(level, diameter)
b
c
DECISION
(upstream cross-drain subnetwork design)
RETURN (cost)
(d) MODIFIED SERIAL SYSTEM

FIGURE 8.5

Define a set of possible discrete cross-drain positions a, b, c, etc. along the length of the drainage network. (Fig. 8.5b)

Let each of these correspond to the downstream end of a stage in the modified serial system. The output from a stage is then the lower pipe level and larger pipe diameter of the pipes that meet at the downstream end of the cross-drain. The input to a stage is the output from one of the upstream stages. Figure 8.5c shows a typical stage and Figure 8.5d the modified serial system.

8.8 Applying I.S.D.P. to the variable cross-drain problem.

The I.S.D.P. can now be structured as follows.

For cross-drain (a) obtain minimum costs for a set of output states for the network consisting of the cross-drain at (a) plus the length of carriageway drains upstream of the cross-drain.

For cross-drain (b) there is either no cross-drain upstream, or a cross-drain at (a). Hence obtain minimum costs for a set of output states at (b) considering both of the following possible networks.

i) cross-drain (b) plus all carriageway drains upstream.

ii) cross-drain (b), plus carriageway drains from (a) to (b) plus the set of minimum costs corresponding to input states at (a).

In general a cross-drain stage may have any of the possible upstream cross-drains forming its input state , or none at all.

In this way the I.S.D.P. proceeds downstream to the base cross-drain, giving a set of minimum costs corresponding to a range of depths and diameters. The process may then be continued by conventional D.P. through to the downstream end of the network.

When the overall minimum network cost is identified the solution can be traced back up the network and the optimal plan identified.

8.9 The design of a stage

The design of a cross-drain stage consists of finding the least cost solutions for all possible networks upstream for a range of output states. As outlined in Section 8.8, this is accomplished by considering the nearest upstream cross-drain to be in each possible position, or there to be no upstream cross-drain. Considering just one position of the upstream cross-drain, Fig.8.5c, the problem can now be stated as follows:

Given a set of input costs corresponding to input states S(in) find the set of designs for the network between the cross-drains that minimises the costs corresponding to output states S(out).

This can be done conveniently using the MANVAR model, thus giving optimal intermediate manhole positions and optimal pipe slopes and diameters.

8.10 Establishing the ranges of value for the state variables

For any D.P. process the ranges of values of the state variables need to be defined, ie. in Fig.8.5c the limits on S(in) and S(out) need to be fixed.

The range of the pipe level could be fixed in relation to the ground level, but this is rather inefficient. It is far better to fix the range in relation to the maximum possible pipe level, which will generally coincide with the "minimum cover" design for a given upstream network.

Here, however, the network upstream is not predetermined, and hence there is no unique minimum gradient design from which to obtain an upper limit on pipe level.

Hence all possible upstream networks should strictly be considered in obtaining the upper limit on pipe level at a given cross-drain position.

If pipe diameter is also a state variable (i.e. if diameters are constrained not to decrease in a downstream direction) then the maximum pipe diameter may be obtained from the same "minimum cover " design procedure.

8.11 Design flows

So far in this chapter it has been assumed that design flows can readily be calculated for the individual components of the design stages and that flows at the main stages are independent of the networks upstream.

Taking the former point first, provided the inlet points for flows into all the carriageway pipes are defined (e.g. gullies) or are continuous along the pipe lengths (e.g. French drain) then for each subnetwork, design flows can readily be calculated as for the MANVAR model.

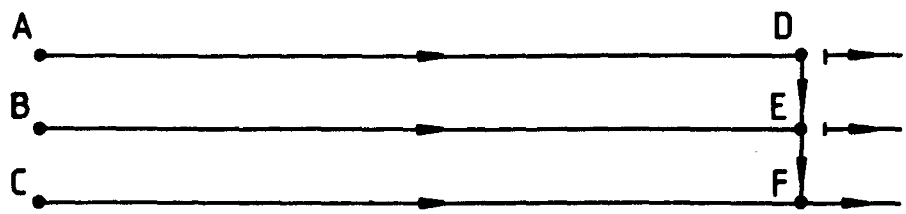
The latter point requires rather more attention. Consider the flow just downstream of F in two extreme cases (Fig. 8.6a and 8.6b)

Using the Rational method as an illustration, the design flow at F depends on the time to concentration t and the catchment area A. A will be equal for the two cases. However the time for case (a), t(a), will depend on the full flow velocity for pipes AD, DE, EF, being largely dependent on the velocity of flow in AD. For case (b), t(b), will depend largely on the velocity of flow in IF, which, being necessarily a larger pipe than AD, will usually have a significantly higher flow velocity.

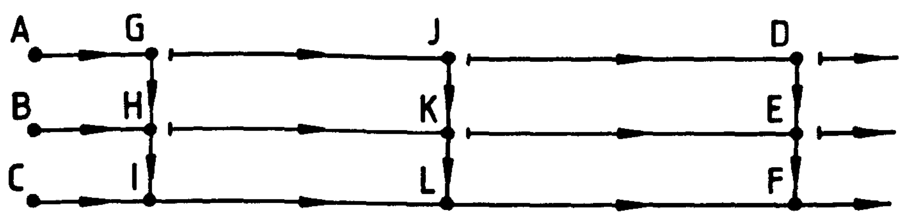
As an example, if AD is of diameter 150 mm throughout its length, if IF is 225 mm throughout its length, if AD = 1000m and if IF = 900m and all pipes are at a gradient of 1 in 250, then t(a) = 1550 secs. and t(b) = 1260 secs. for typical pipes. For a 1 year storm, design flow Q(b) is then 18% higher than design flow Q(a).

8.12 Cross-drain sets with networks upstream

It is possible to have branches joining into the component drains of a cross-drain set. Such an arrangement is shown in Figure 8.7a. This



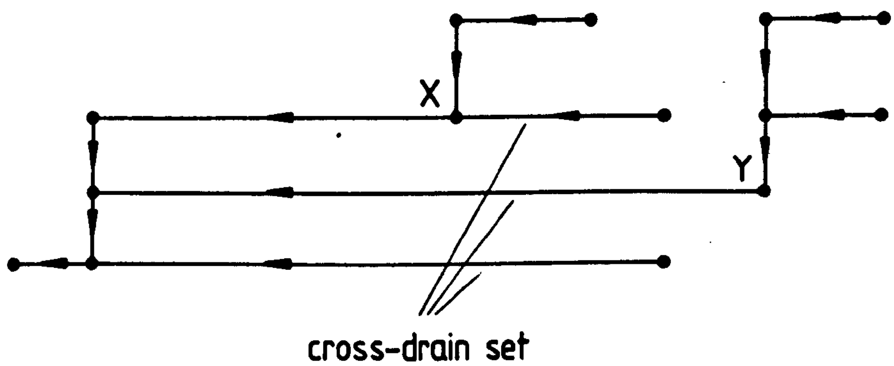
(a)



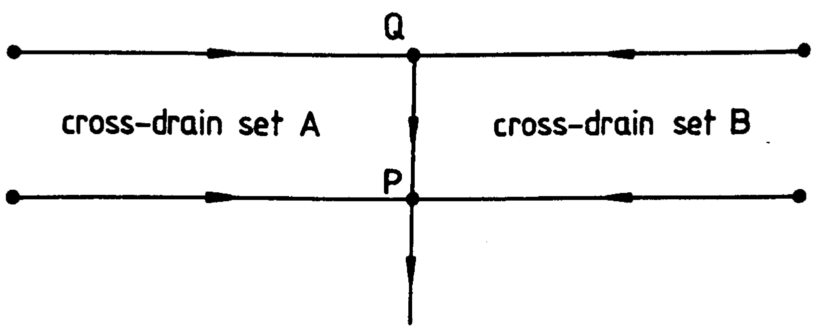
(b)

DESIGN FLOWS WITH VARIABLE NETWORK

FIGURE 8.6



(a) CROSS-DRAIN SET WITH BRANCHES JOINING



(b) CROSS-DRAIN SETS WITH A COMMON BASE

FIGURE 8.7

presents no theoretical difficulty, as the situation can be handled as follows:

1) Identify all such branches, and identify the manholes (e.g. X, Y) at which they enter the cross-drain set.

2) Define range of discrete output states for the branches at these manholes.

3) Obtain by MANVAR the set of minimum costs for these discrete output states.

The main cross-drain set I.S.D.P. design may then proceed, incorporating the sets of branch costs.

8.13 Cross-drain sets sharing a common base cross-drain

Frequently two sets of parallel drains enter a common base drain, one from either side as in Figure 8.7b. This raises a theoretical difficulty with the proposed method, as it has so far been assumed that a cross-drain set can be designed in isolation from any other cross-drain set.

Imagine performing the I.S.D.P. process on set A. The final cross-drain position considered is the base cross-drain QP.

A range of states at P is considered, and the minimum costs of arriving at those states is obtained, considering only cross-drain set A, i.e. excluding the effect on the pipe level at Q of the drain entering Q from cross-drain set B.

These costs include the cost of drain QP.

Imagine now the I.S.D.P. process on cross-drain set B. A new set of minimum costs will be obtained for the states at P, again including the cost of the drain QP.

These two sets of costs at P can then be combined to form a single set of optimal upstream costs over the range of states at P. These will however include drain QP twice. The D.P. will then proceed to the network outfall. The final network trace back will identify one state at P from which the optimal set of drains for A and B will be identified. However, base drain QP may well have two different designs for the two different cross-drain sets. Indeed the pipe levels at Q may well be quite incompatible.

Theoretically, then, it is necessary to consider the two cross-drain sets simultaneously in the I.S.D.P. process. Such a procedure would involve severe computational penalties even if a sound method could be evolved, and so was pursued no further.

In practice, therefore, cross-drain sets sharing a common base cross-drain are designed as if they had separate base cross-drains. It is in fact unlikely that the cost of the base cross-drain will exceed about 1% of the cost of the upstream drains in any normal network. Hence its effect on the positioning of the cross-drains is likely to be minimal. The problem of two differing trace-back solutions for the base cross-drain is overcome by using the procedure described in the following section which describes a practical model.

8.14 A practicable model - CROSSVAR

8.14.1 Introduction

A model based on the I.S.D.P. approach was developed for the variable cross-drain problem. The model, CROSSVAR, breaks the optimisation problem into three parts.

Firstly, cross-drain positions are determined. Next intermediate manhole positions and pipe diameters are found. Finally pipe slopes are determined. The last two stages are equivalent to the MANVAR model.

8.14.2 Structure of CROSSVAR

CROSSVAR is implemented by two programs, SORT and MODEX linked as shown in Fig.8.8. Both are generally run twice.

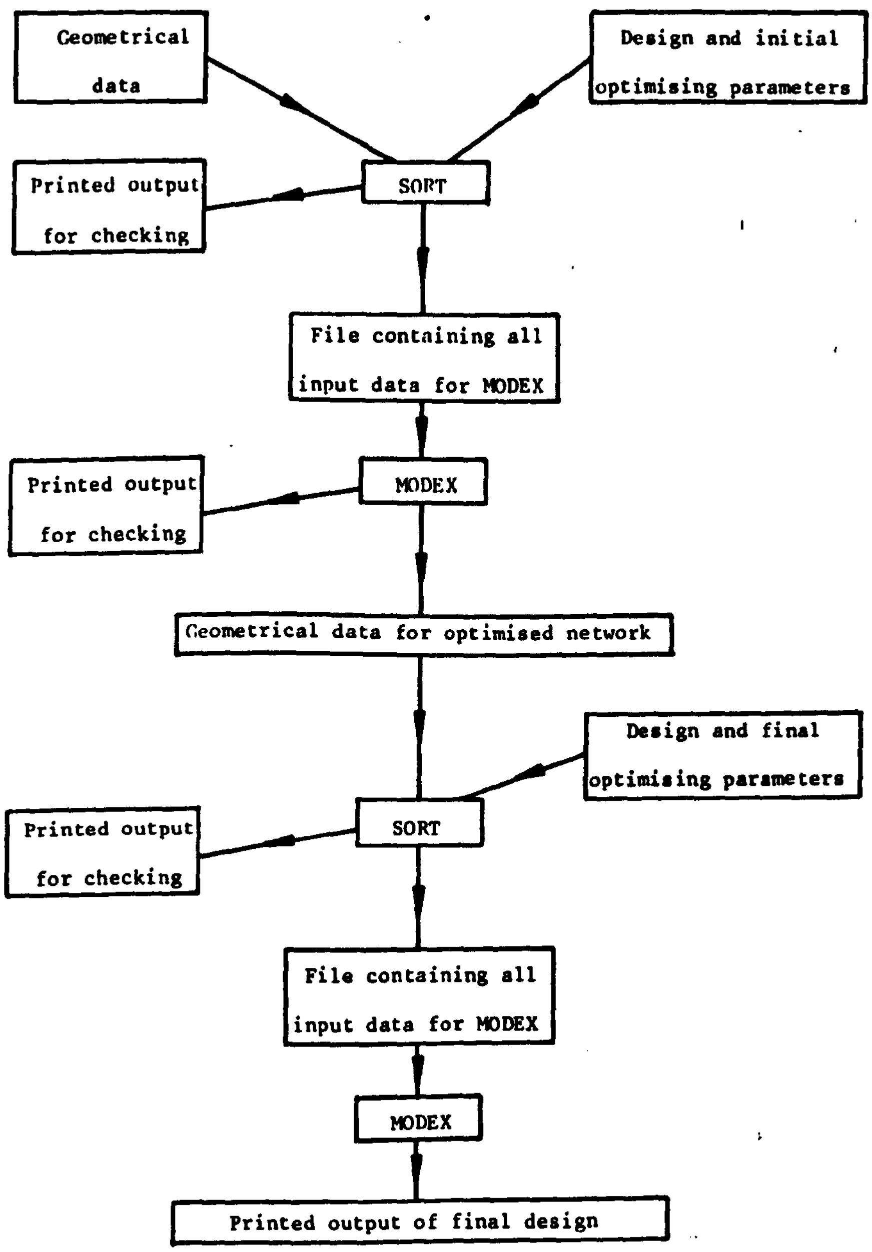
SORT accepts in card format input data describing the road geometry and network layout, and a set of design parameters. It outputs on magnetic tape a complete set of input data for MODEX.

A flow chart for SORT is given in Fig 8.9 and a program listing is given in Appendix F.

MODEX operates in one of two modes. If there are any cross-drain positions to determine, it will do so and output in card format the geometry corresponding to the resultant pipe network. This is then processed by SORT, and returned to MODEX which, because there are now no cross-drain positions to find, will operate in its second mode. In this mode it will perform the same function as program MOD in MANVAR (see Section 7.13) producing first a set of optimal manhole positions and pipe diameters and then a set of pipe gradients.

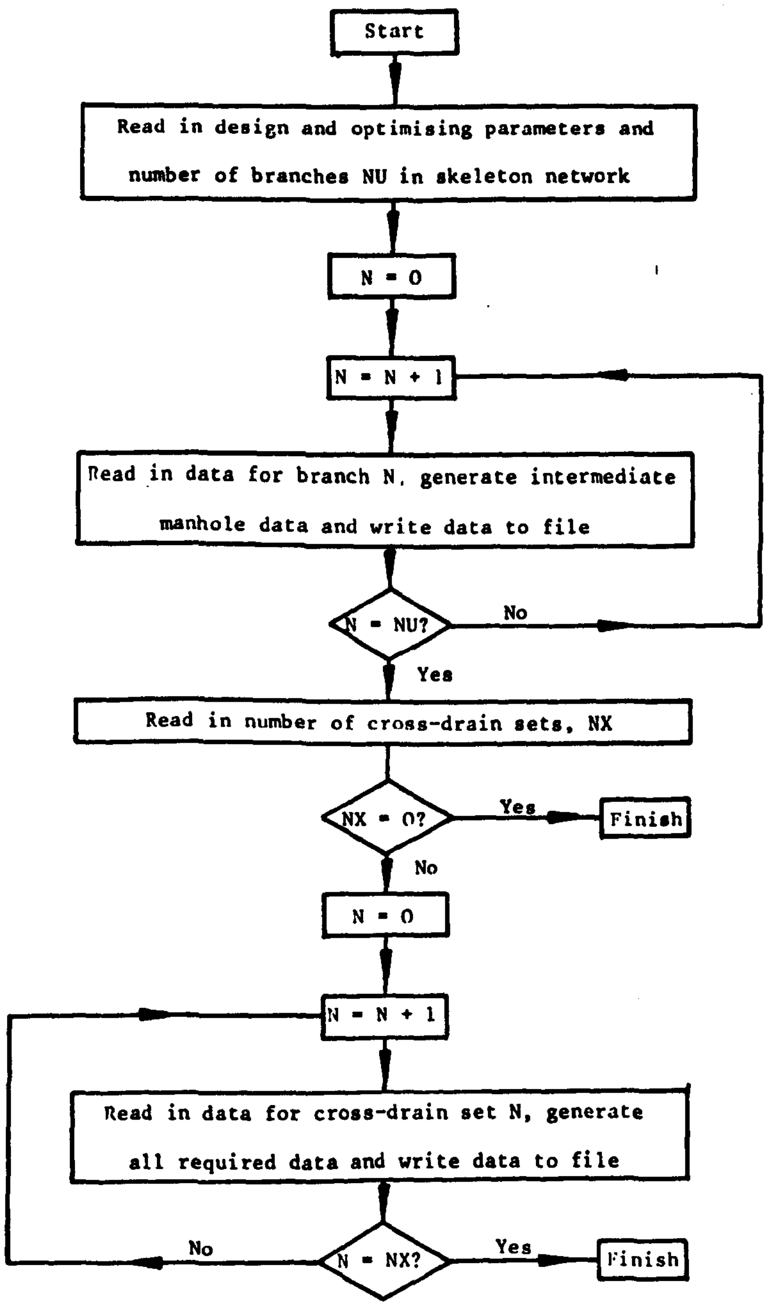
CROSSVAR was so structured for two main reasons. Firstly, as explained in Section 8.13, it is not possible to obtain a theoretically correct solution for the common case of two sets of cross-drains sharing a common base. Hence some approximation is necessary, the most satisfactory being to assume two base cross-drains independent of each other for the purpose of establishing cross-drain positions. Having established these, the optimal solution for the resulting network can then be obtained assuming a joint base cross-drain by re-running MODEX with no variable cross-drains.

The second reason is that of program efficiency. The time taken to produce a design with a single program run is an order of magnitude greater than that taken to run the program twice, once to establish cross-drain positions, and then to complete the design, whilst the results obtained are usually identical. The disadvantage is that the solution could be suboptimal in the latter case, if the cross-drain positions are sensitive to the choice of grid spacings for the manhole positions and pipe levels. (See Section 8.19)



IMPLEMENTING THE CROSSVAR MODEL

FIGURE 8.8



FLOW CHART FOR PROGRAM SORT

FIGURE 8.9

8.15 Program MODEX

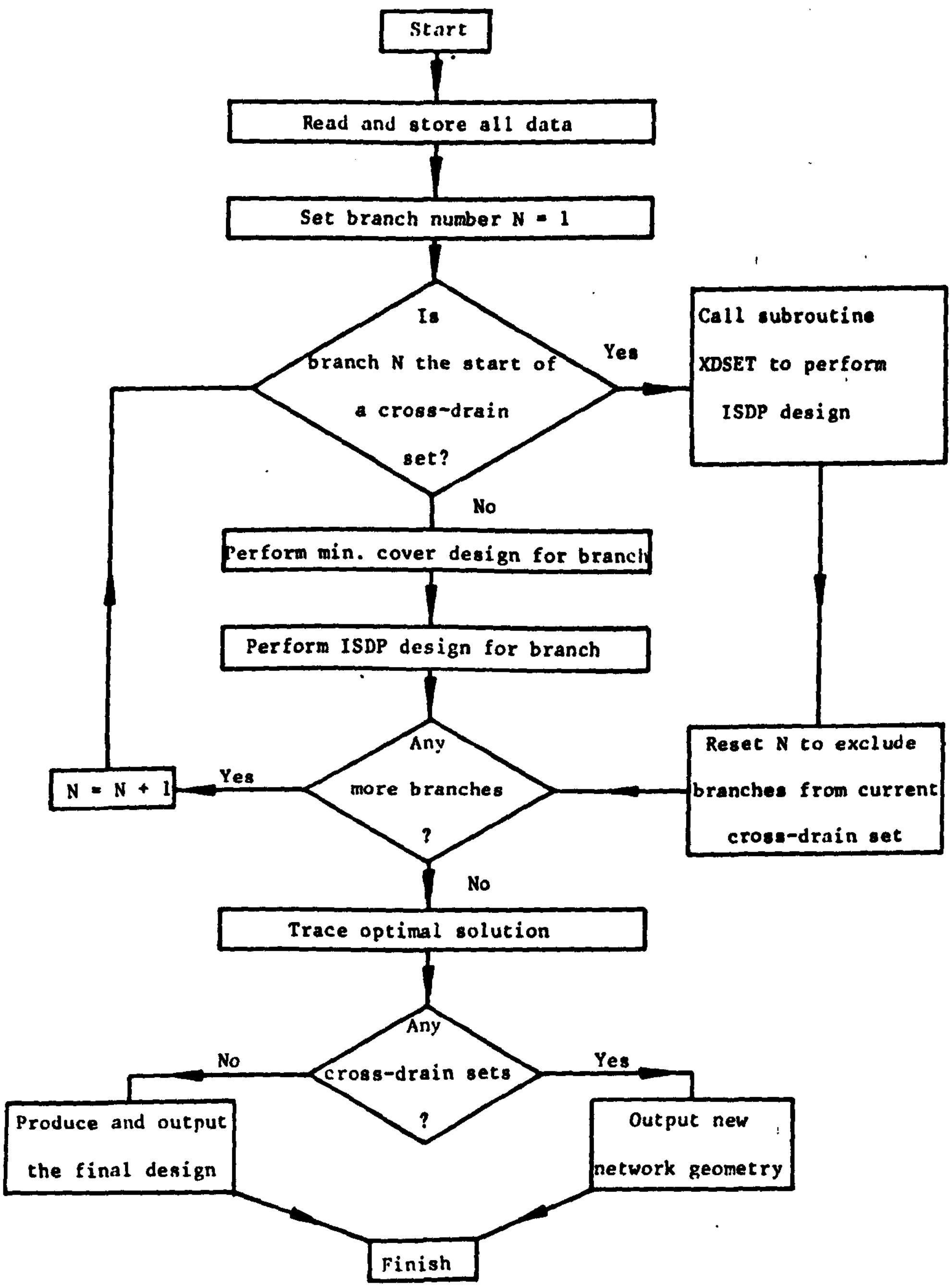
Flow charts for program MODEX are given in Figures 8.10 and 8.11, and a program listing in Appendix G.

Essentially MODEX is a version of MOD, extended to include a variable cross-drain facility. The program takes each branch of the network sequentially from the upstream ends, and if the branch is not a member of a cross-drain set, performs an ISDP optimisation for a range of downstream states.

When a member of a cross-drain set is identified, control is switched to a subroutine XDSET. This subroutine controls the I.S.D.P. optimisation for the cross-drain set, identifying subnetworks between cross-drains, which are then optimised by calling SUBNET.

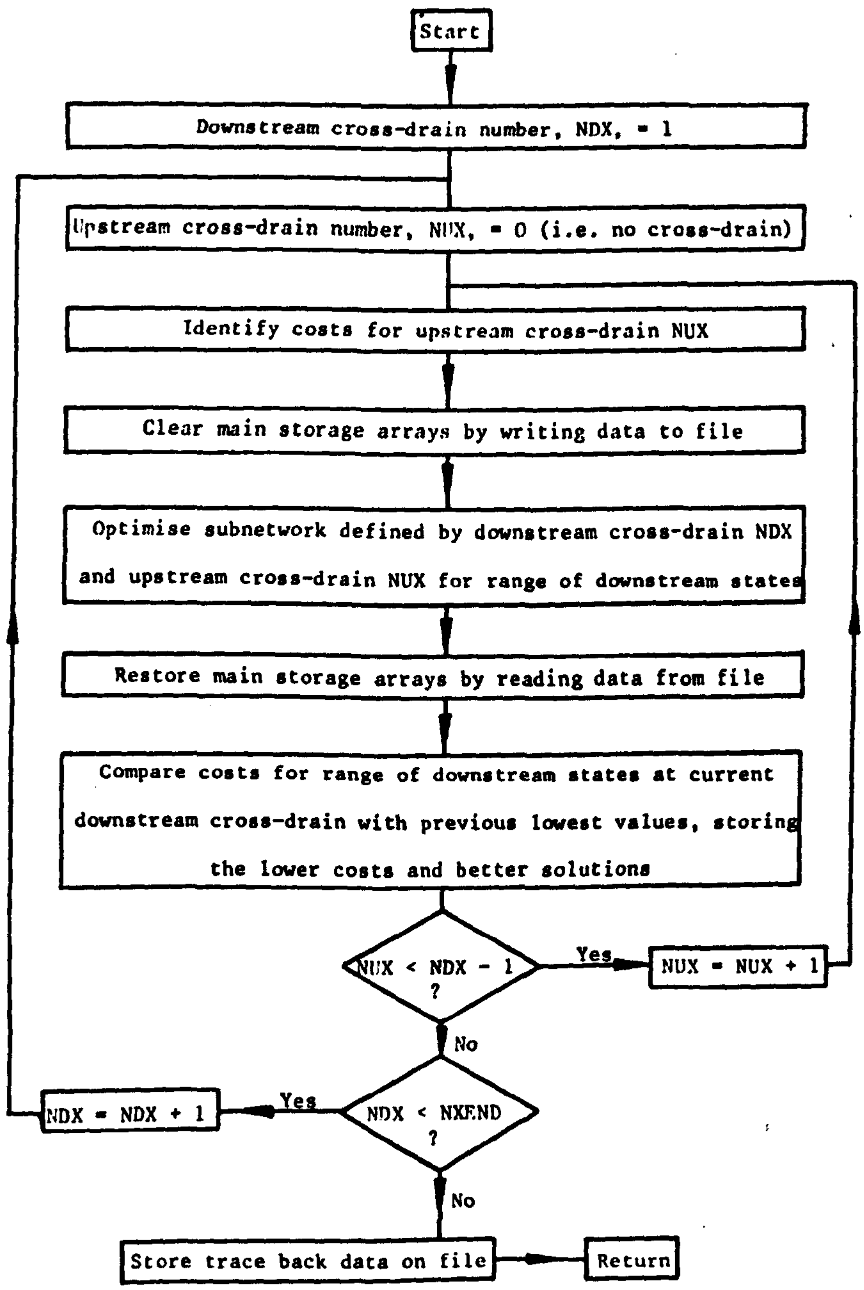
MODEX has an important refinement over MOD in that it was realised it was not essential to perform a minimum cover design at the start of the program. This was done in MOD to establish the upper limits on the state variables throughout the network. In MODEX, the minimum cover design is done for a branch at a time, just before that branch design is optimised. This overcomes the problem of the network not being defined initially.

MODEX consists of about 1900 lines of National Computer Centre Standard Fortran and is hence largely machine independent. Data is stored partly in two main arrays, these being dynamically addressed to minimise core storage requirements, and partly on two magnetic tape workfiles for bulk storage of data that is not being currently used by the program.



FLOW CHART FOR PROGRAM MODEX - Main Program

FIGURE 8.10



FLOW CHART FOR PROGRAM MODEX - Subroutine XDSET

FIGURE 8.11

8.16 Optimising Parameters for CROSSVAR

The CROSSVAR programs were written to cope with any reasonable combination of optimising parameters.

These parameters are as follows :

1) Cross-drain resolution, i.e. the spacing between possible cross-drain positions.
2) Manhole resolution, i.e. the spacing between possible manhole positions.
3) Number of pipe diameters considered at each manhole.
4) Width of the pipe level zone
5) Number of discrete levels within pipe level zone.

Rather than generating cross-drain positions at specific intervals of chainage along the road, it was thought better to specify the manholes that they would connect. Hence the spacing of possible cross-drain positions is in fact a multiple of the spacing of possible manhole positions.

The possible manhole positions are generated along each branch of the cross-drain set and numbered. Possible cross-drain positions are generated from the base cross-drain upstream and defined by the corresponding manhole numbers.

Using this system all possible manhole positions can be generated initially.

Parameter 1 is used only in the first run of SORT and MODEX. Parameters 2, 3, 4 and 5 are specified independently for the first and second runs of the program. Hence a coarse initial run can be used to establish cross-drain positions, followed by a finer process for establishing manhole positions and pipe diameters.

8.17 Program of Testing for CROSSVAR

A program of testing was devised for CROSSVAR, to check its validity and to choose suitable optimising parameters. The program was as follows:

(1) Check compatibility with MANVAR

(2) Find typical cross-drain spacing

(3) Test whether I.S.D.P. can be truncated by considering only a certain number of possible upstream cross-drain positions

(4) Test the stability of the solution (for cross-drain positions) to variations in the optimising parameters 2, 4 and 5

(5) Find suitable values of optimising parameters 2 to 5

(6) Find the effect of cross-drain resolution (parameter 1) on overall optimal cost

(7) Choose suitable values for cross-drain resolution

(8) Run using other networks

8.18 Results using the CROSSVAR model

8.18.1 Checking CROSSVAR with previous results

The first runs of the CROSSVAR model were to check that it was fully consistent with the MANVAR model. Hence these test runs were performed using CROSSVAR without its cross-drain optimising capability.

Three test runs were performed on two networks and compared to runs using MANVAR.

The first of these was on the network shown in Fig. 7.16 and described in Section 7.14.5. The results from the two models were identical.

The other two tests were on network 3 (Figure 7.11) using two sets of design parameters. The results differed very slightly from MANVAR. This was traced to a minor error in the MANVAR programs. However the results were substantially identical.

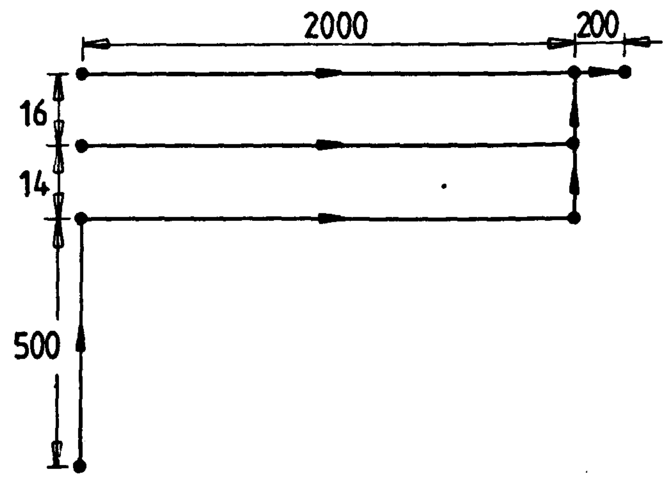
8.18.2 Finding Typical Cross-drain Spacings

The second set of runs were aimed at finding typical cross-drain spacings. Five runs were performed on network 4 (Fig 8.12a) using first of all a coarse spacing of possible cross-drain positions, and then gradually finer spacings.

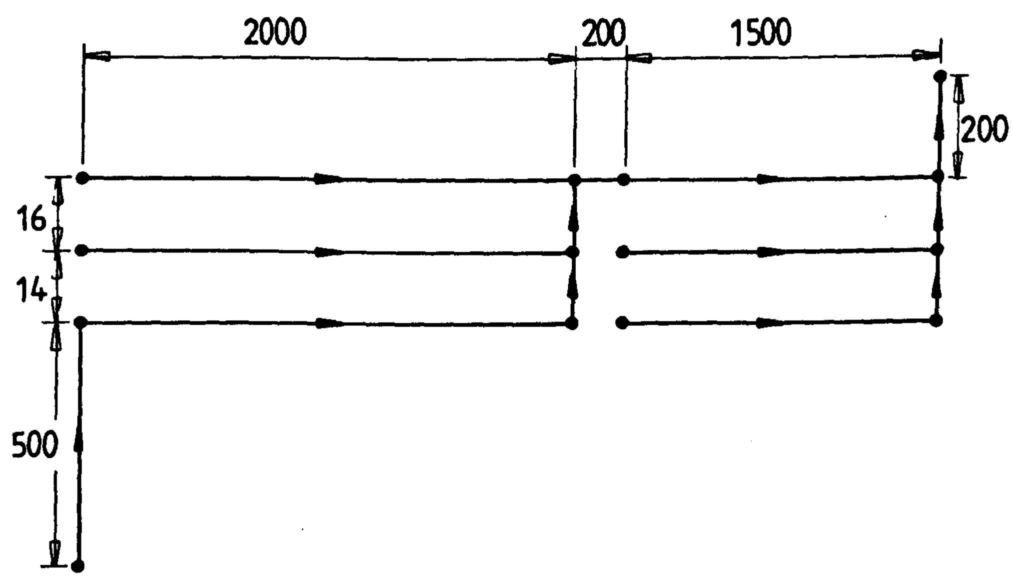
The results of these runs are illustrated in Figure 8.13. For run 5, the spacing between cross-drains was limited to 250 m to reduce program execution time. The result is therefore somewhat artificial.

In general the runs show that optimal cross-drain spacings were between 150 m and 500 m for this network.

It would seem reasonable therefore to limit the maximum cross-drain spacing in order to truncate the I.S.D.P. process, thereby considerably reducing computation.



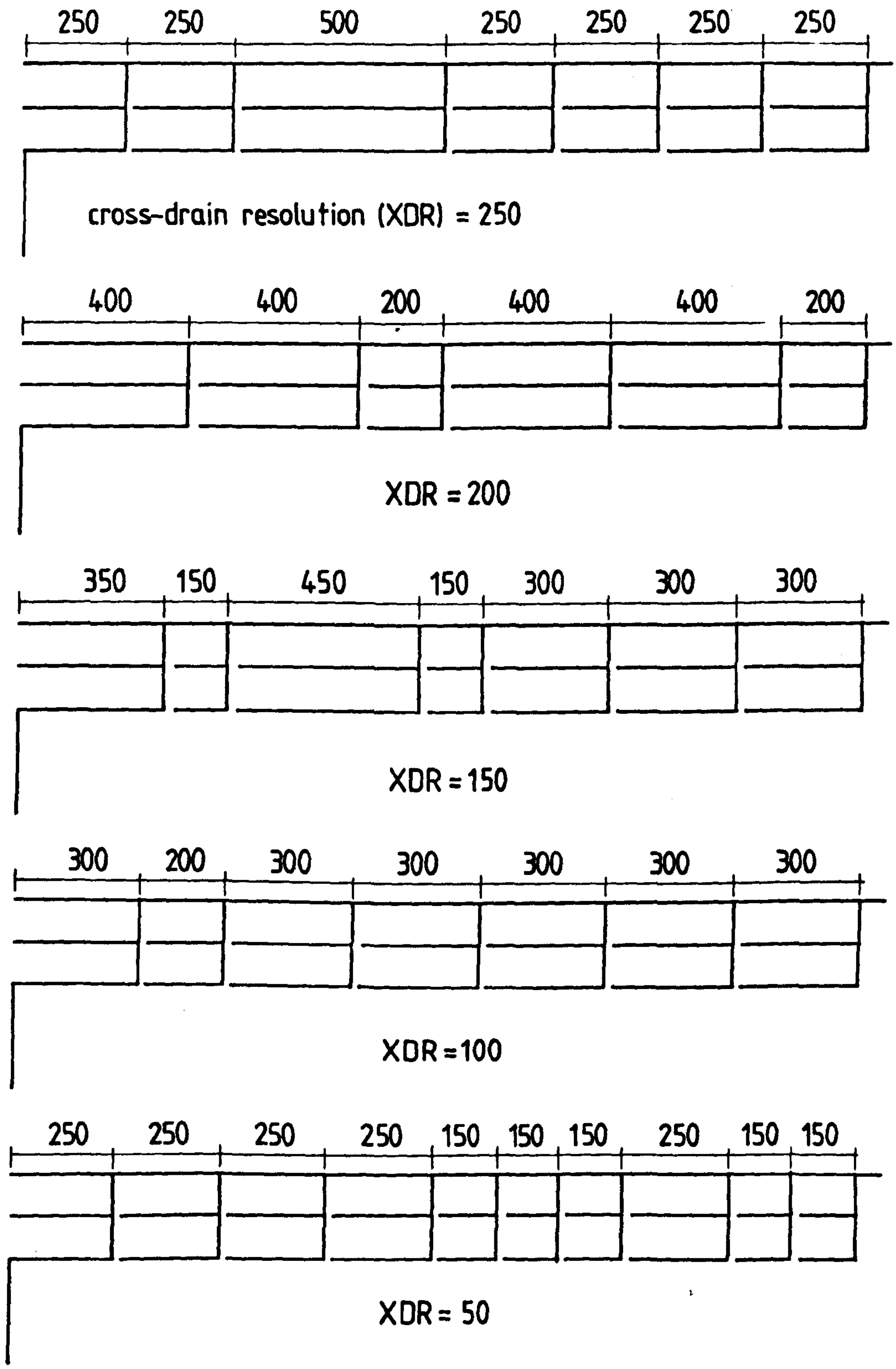
(a) NETWORK 4



(b) NETWORK 5

note: all dimensions in m.

FIGURE 8.12



CROSS-DRAIN POSITIONS FOR NETWORK 4 USING MODEX

FIGURE 8.13

8.18.3 Stability of Cross-Drain Position to Variation of Parameters.

The third set of runs were to examine the stability of the optimal cross-drains to variations in three of the optimising parameters. All runs used Network 4 (Fig. 8.12)

a) Manhole Resolution

The first parameter investigated was the manhole resolution, which was allowed to vary from 25 m to 150 m (the maximum spacing permitted), whilst the cross-drain resolution was held at 150 m. Having established cross-drain positions, the second optimisation used a new set of optimising parameters which were identical for all runs.

Results for these runs are shown in Table 8.4

It can be seen that **alteration** of the manhole resolution from 25 m to 150 m produces just one change in the cross-drain positions. This change leads to a very slight (0·1%) increase in the cost of the final optimal network. Overall execution time for the program decreases by a factor of between 6 and 7.

In these circumstances it would seem reasonable to consider only a manhole resolution equal to the maximum specified spacing of manholes, assuming this to be not greatly in excess of 150 m.

b) Width of Pipe Level Zone

The second parameter investigated was the width of the pipe level zone in the initial stage of the model.

As a preliminary to this, one run was performed using a zero width of zone for this stage, i.e. a minimum cover design.

This produced a design which at £118 124 was about 1% more expensive than the optimum. The number of cross-drains generated was larger than

Run Number	Manhole Resolution (m)	Cross-drain location (m) (Base cross-drain at chainage 2000) 200	350	500	650	800	950	1100	1250	1400	1550	1700	1850	1900	Network Cost (£)	Run Time (secs)
21	25		X	X			X	X		X		X			117143	911
13	50		X	X			X	X		X		X			117143	323
23	75		X	X		X		X		X		X			117267	212
22	150		X	X		X		X		X		X			117267	147
	Zone Depth (m)															
(23)	1.0		X	X		X		X		X		X			117267	147
42	2.0		X	X		X		X		X		X			117267	148
43	3.0		X	X		X		X		X		X			117267	149
44	4.0		X	X		X	X	X	X	X			X		118058	146
	Number of Levels															
(44)	11		X	X		X	X	X	X	X			X		118058	146
51	21		X	X		X		X		X		X			117267	?
52	31		X	X		X		X		X		X			117267	494
53	41		X	X		X		X		X		X			117267	776

X = cross-drain position

SENSITIVITY OF CROSS-DRAIN POSITIONS TO OPTIMISING PARAMETERS

TABLE 8.4

that for any previous run, probably due to larger pipe sizes being generated by the low gradients.

As the manhole resolution was quite fine, (25 m), the run involved a moderate execution time of 232 secs. The idea of using a minimum gradient design to establish the cross-drain pattern was rejected as it was felt that better solutions could be obtained in less execution time by a suitable choice of parameters.

The main series of runs varied the width of zone from 1·0 m to 4·0 m, the top of the zone being at a minimum cover design level. The remaining optimising parameters were held constant, the values of manhole resolution being 150 m, cross-drain resolution being 150 m, number of levels being 11 and number of diameters being 3.

The results are shown in Table 8.4. These indicate that there is no advantage in having a wider zone whilst keeping the number of discrete levels the same. In fact, if the spacing between discrete levels becomes excessive, the solution may well become sub-optimal as in run 44.

c) Number of levels in zone

The last series of runs in this section investigated the effect of varying the number of levels within a zone of fixed width.

Four runs were performed using a standard zone width of 4·0 m, manhole and cross-drain resolution of 150 m and 3 pipe diameters. The number of levels were varied from 11 to 41, giving spacings of 0·4 m to 0·1 m between levels. The results are shown in Table 8.4.

Only the first section of CROSSVAR was implemented for these runs. This, however, was sufficient to show that the cross-drain positions remained stable for spacings up to 0·2 m.

8.18.4 The effect of cross-drain resolution on the optimality of the solution

The fourth set of runs were designed to find how the cost of the optimised network varied with the choice of cross-drain resolution.

Network 4 was again used, for cross-drain resolutions of from 100 m to 500 m. Manhole resolution and maximum manhole spacing were taken as 100 m. A zone width of 1 m with 11 discrete levels was adopted together with 3 possible pipe diameters.

The results are plotted on the graph of Figure 8.14.

As one would expect, the general trend is for the solution to decrease in cost as the spacing decreases. There is not however, a great increase in execution time and it would seem sensible to adopt a cross-drain resolution equal to maximum manhole spacing.

8.18.5 Runs using other Networks

Two runs were performed using other, more complicated, networks. These were primarily to test that the program was capable of handling such networks and to see whether the results it gave were sensible.

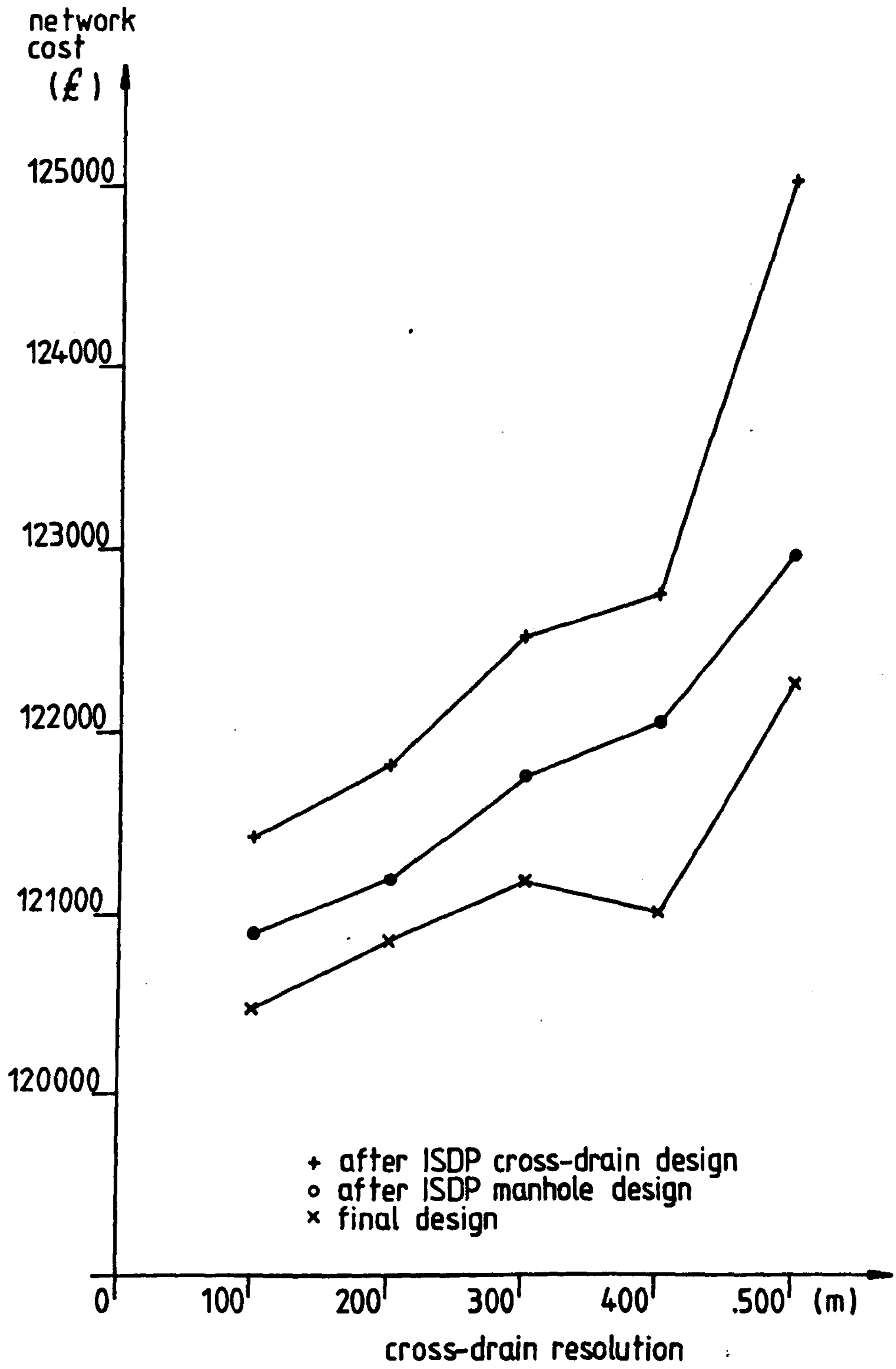
The networks used were Network 5, Figure 8.12b, and Network 3, Figure 7.11.

The resulting designs are shown in Figure 8.15(a) and (b).

8.19 Choice of values for the Optimising Parameters

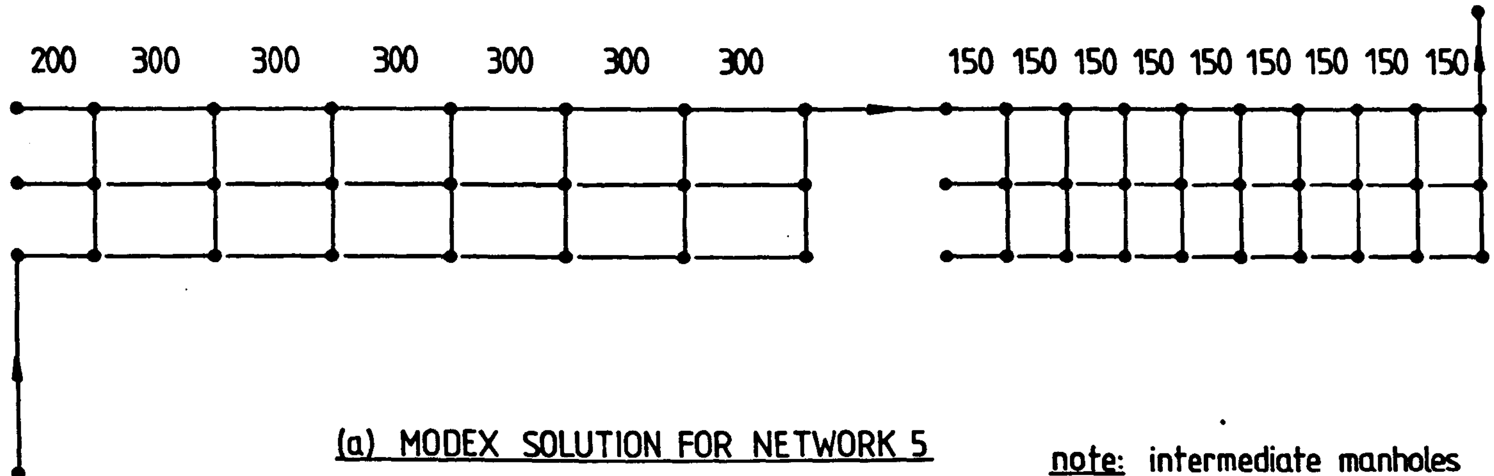
The choice of values for the optimising parameters is not obvious as it depends on a trade-off between computer resources and the degree of optimality of the design.

However, it seems from the examples used in this research that the first part of the program can be run with the following values of parameters and



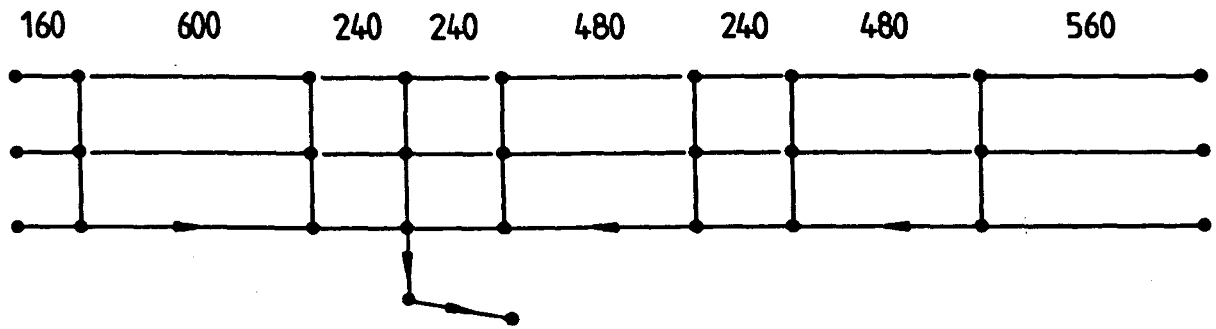
SENSITIVITY OF NETWORK COST TO CROSS-DRAIN RESOLUTION

FIGURE 8.14



(a) MODEX SOLUTION FOR NETWORK 5

note: intermediate manholes not shown



(b) MODEX SOLUTION FOR NETWORK 3

FIGURE 8.15

still, in most cases, achieve the optimal set of cross-drain positions for a given cross-drain resolution.

Manhole resolution = maximum manhole spacing

Pipe level zone = 1·0 m

Discrete pipe levels = 6

Discrete pipe sizes = 3

The choice of cross-drain resolution is less obvious. It would not seem sensible to decrease the spacing below the maximum manhole spacing. A value of cross-drain resolution equal to the maximum manhole spacing is thus suggested.

For the second part of the program, it is necessary to alter the manhole resolution to 25 m (or a convenient factor of the cross-drain resolution close to this value), and alter the number of pipe levels considered to 11. This will give a set of manhole positions and pipe diameters close to the optimal set, if not actually optimal.

8.20 Conclusions on the use of CROSSVAR

This chapter shows that an optimal drainage design model capable of dealing simultaneously with variable cross-drain positions, variable intermediate manhole positions, variable pipe diameters and gradients can be implemented using the I.S.D.P. approach.

Using a sensible choice of optimising parameters the execution times on a large computer are reasonable, the costs involved being only a small proportion of the likely saving on construction costs.

On the examples tested, the savings made by using the CROSSVAR model instead of the MANVAR model with fixed cross-drain positions are not large. As execution times for CROSSVAR are up to an order of magnitude larger than for MANVAR, it was decided not to proceed with a commercial version of CROSSVAR at this stage. It was felt that the extra program length, execution cost and documentation would have discouraged engineers from using it.

CHAPTER 9

A FURTHER VARIABLE PLAN OPTIMISATION PROBLEM

9.1 Introduction

9.2 Connecting several sources of flow to a single main drain

9.3 A DP approach - MULTICON

9.4 Comments on the method

Chapter 9 A further variable plan optimisation problem

9.1 Introduction

In this chapter a further problem involving variable plan optimisation is examined and a method of solution is proposed. The proposed method has not been tested.

9.2 Connecting several sources of flow to a single main drain

A typical variable plan drainage problem might be posed as follows:

Given sources of known drainage flow at manholes A,B,C,D,E,F in Fig. 9.1a connect them in the least cost way to the outfall manhole O, whilst satisfying all the usual drainage design constraints (see Section 2.3). Such problems may indeed be applicable to other forms of network, for instance water supply, roads and gas.

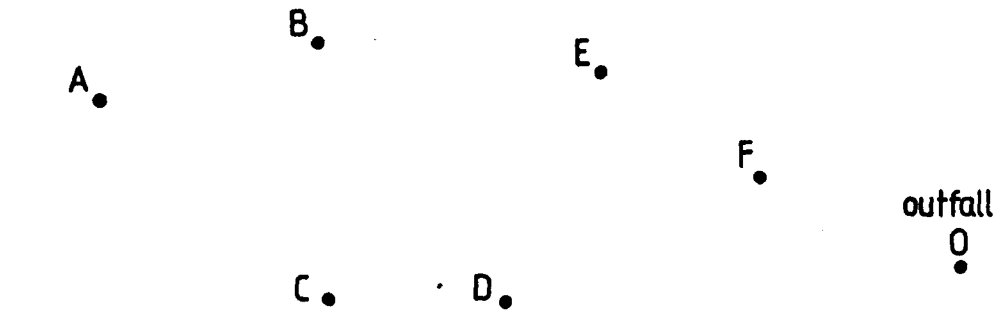
Wilson (ref. 25) attempted to form a model based on Geometric Programming, with the simplifying assumption that there was a straight main drain into which each of the others connected (eg AO in Fig.9.1b) However, he met with severe problems due to optimal solutions involving manholes coinciding and a large number of equality constraints.

9.3 A DP approach - MULTICON

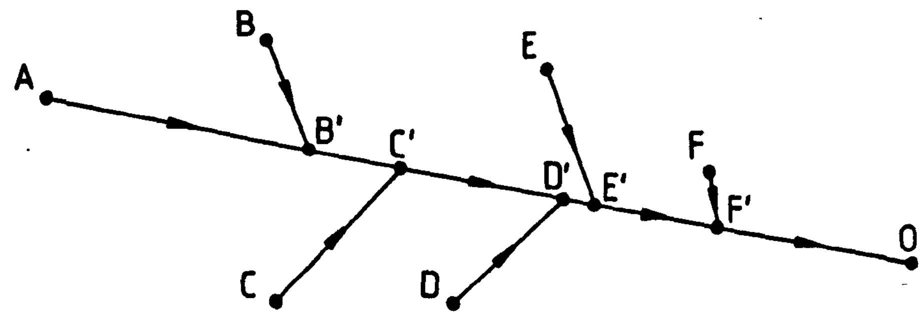
It is necessary to make two simplifying assumptions. The first is that each manhole is connected to a single main drain. This excludes the possibility of a branch drain linking several manholes before connecting to a main drain. Secondly it is necessary to predefine the order in which the manholes are connected along the length of the main drain. In the example it is assumed they are connected in the order ABCDEF. Note that it is not necessary for the main drain to be straight.

For simplicity it is assumed that pipe diameters may increase down the network, although a restriction on pipe diameters can readily be incorporated.

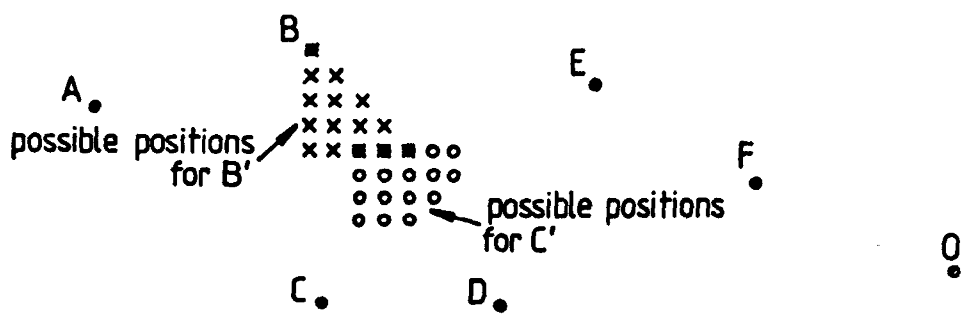
Working from the upstream end of the network consider first manhole A. This can be considered as the start of the main drain. Consider now manhole B. A drain will run from manhole B to join the main drain at an unknown position, say B', or possibly the main drain could run through B. Consider a grid of points representing possible positions for this junction B'. This grid may include point B. (see Fig. 9.1c)



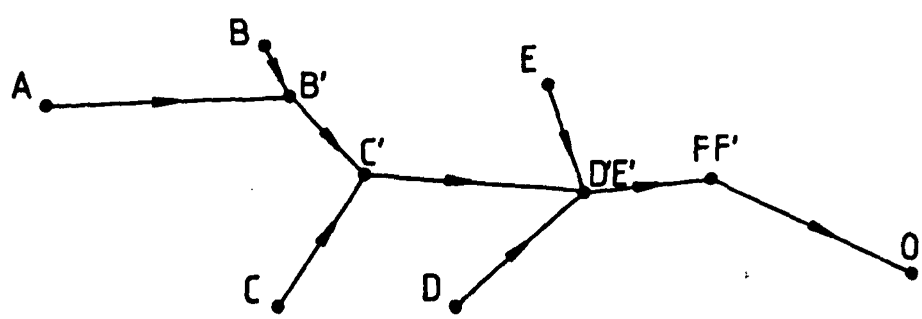
(a) BASIC LAYOUT



(b) TYPICAL SOLUTION USING STRAIGHT MAIN DRAIN



(c) TYPICAL GRID POSITIONS FOR JUNCTION MANHOLES



(d) POSSIBLE MULTICON SOLUTION

CONNECTING SOURCES TO SINGLE MAIN DRAIN

FIGURE 9.1

For each possible grid position B' calculate the minimum total cost of connections AB', BB', for each of a range of discrete pipe depths at B'. This could be done using the MANFIX model. Moving on to manhole C, consider a grid of possible junction points C' and obtain for each grid position and discrete depth the minimum total upstream cost. This consists of the cost of CC' plus the cost of B'C' plus the previously obtained cost of the network upstream of B', and may be obtained by using MANFIX on the subnetwork consisting of CC' and B'C' for every feasible position of junction B'.

In this way the design proceeds downstream and the minimum cost of the network can be found for a range of depths at the outfall manhole. Hence the overall minimum cost can be selected and the solution traced back up the network. A typical solution is shown in Fig. 9.1d. A flow chart for this procedure is shown in Fig. 9.2.

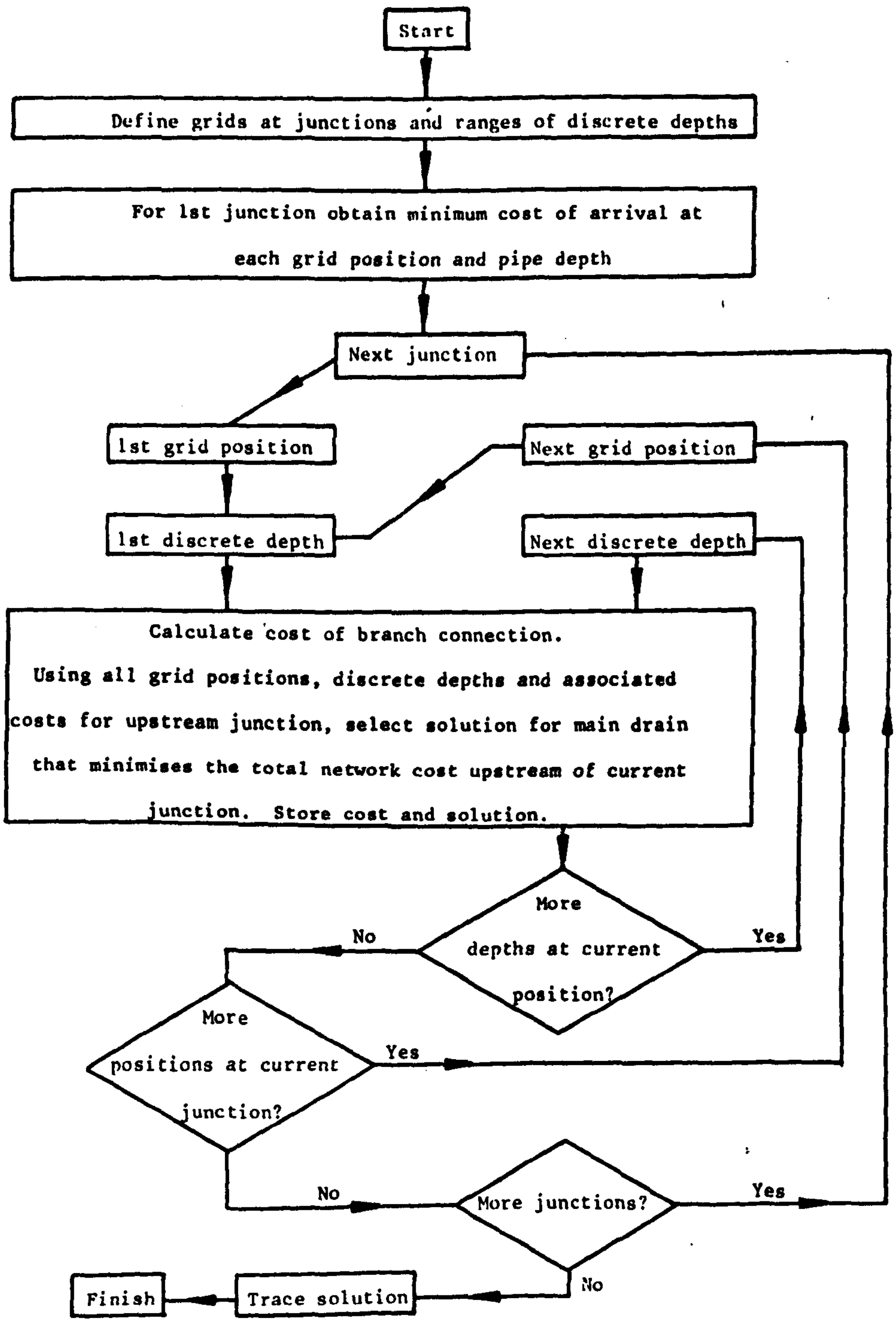
9.4 Comments on the method

The main disadvantage with the model is that the order of connection must be predetermined. For example, a solution in which C was connected into the main drain before B would not be considered. However it can allow B and C to be connected at the same point thus overcoming a problem that Wilson found (see Section 9.2) and indeed can allow the main drain to pass through the sources of flow.

If lengths between junctions, or lengths from source manholes to the main drain justify the inclusion of intermediate manholes, then MANVAR can be used to establish the minimum subnetwork costs.

A computer program has not yet been written to implement the method so no practical problems such as storage requirements or execution time can be discussed.

Applications in sewer system design could be considerable wherever some freedom of choice exists in the positioning of manholes. The discrete nature of the possible solutions may itself be valuable in accommodating problems in which manholes are limited to a small number of possible discrete positions. For example, this can arise if manholes are to be at street corners or in road verges.



FLOW CHART FOR MULTICON

FIGURE 9.2

CHAPTER 10

CONCLUSIONS AND AREAS FOR FURTHER STUDY

10.1 MANFIX

10.2 MANVAR

10.3 CROSSVAR

10.4 MULTICON

10.5 General Conclusions

10.6 Areas for Further Study

Chapter 10. Conclusions and Areas for Further Study

10.1 MANFIX

A study of previous work on the optimisation of fixed plan drainage networks provided a good basis for the development of a new and efficient Dynamic Programming model called MANFIX . This correctly handles the constraint that pipe diameters should not decrease in a downstream direction by the use of two state variables in the D.P. process. Although the number of elemental designs at each stage is thereby greatly enlarged, the actual computation time is not unduly increased due to each elemental design being greatly simplified. The final solution produced is fully consistent with the method chosen for determining design flows. This is accomplished by using the D.P. to establish the set of optimal pipe diameters, and then using these diameters in a final fully consistent design process to establish pipe gradients.

Results show that the set of optimal pipe diameters can be reliably found by the use of a very coarse D.P. grid, using an initial "minimum cover" design to establish a set of approximate flows and bounds on the D.P. process. Hence the process is comparable to DDDP in computer time and storage requirements with the possible advantage of simplicity.

A separate computer program was not developed for MANFIX, but the model was an essential foundation for MANVAR, the variable manhole position model.

10.2. MANVAR

The problem of optimising the number and position of manholes along the line of a non-branching drainage run required the introduction of a new type of D.P.

This was termed Indeterminate Stage Dynamic Programming (ISDP) as the number of stages in the final solution is not predetermined. The concept of a set of discrete feasible positions for intermediate manholes provides the key to the problem, enabling ISDP to be used in an elegant procedure.The choice of a suitable set of optimising parameters leads to an efficient and fully practicable computer program. Whilst savings in construction cost over non-optimised schemes are not as great for road drainage networks as for other forms of drainage, it is clear that for a very modest computer running cost, large sums of money can still be saved. The model is flexible enough to allow a great deal of freedom in the choice of optimising parameters so that they can be altered at will to allow for varying computer costs.

Schemes designed using MANVAR produced solutions with levels close to the minimum possible cover but with generally smaller pipe sizes. This was accomplished by minor changes in pipe gradients and better positioning of manholes. The resultant designs are usually better from an engineering viewpoint in that less of the network is at minimum gradient and hence is less likely to suffer from siltation and blockage.

A fully commercial version of MANVAR for use in the design of new road drainage schemes (DAPHOP) was produced and is undergoing trials by the Department of Transport.

10.3. CROSSVAR

The ISDP process was used for the more complicated problem of finding the number and position of drains crossing the road to link parallel carriageway drains.A set of feasible cross-drain positions is first identified and a coarse ISDP design performed to establish the set of optimal cross-drains.

The resultant network is then optimised using the MANVAR procedure. In this way a near optimal design of cross-drains, manhole positions, pipe diameters and gradients can be achieved for a typical road drainage problem.

CROSSVAR is not fully optimal for design flows that are dependent on the network, but gives a solution which is probably very close to the optimum.

A practical computer program was written and tested successfully, but a fully commercial version was not produced.

Experience in the use of CROSSVAR showed that the optimal number of cross-drains in the typical road drainage program was not at all obvious, and that the overall network cost was not very sensitive to cross-drain positions.

10.4. MULTICON

MULTICON demonstrates the adaptability of the general D.P. approach to drainage network problems. A variable layout problem, completely different from those tackled by MANVAR and CROSSVAR, is solved by D.P. This is the case of a number of manholes connected by drains to a single main drain. The method involves defining a set of discrete possible positions for each junction manhole, and using manhole position, in effect, as a state variable. This idea developed from the use of ISDP in the MANVAR and CROSSVAR models, but as yet MULTICON has not been implemented.

10.5. General Conclusions

Due to the complexity of optimal layout problems for storm drainage systems no single "black box" algorithm is possible at present or likely to be developed in the foreseeable future.

However the general D.P. approach has been shown to be highly effective if correctly tailored to the individual type of network considered.

Road drainage is one such type of network and alternative network optimisation models have been developed and tested using ISDP. These are both fully practicable and show worthwhile savings over non-optimised solutions , (see also Ref. 62).

To demonstrate the suitability of the general DP approach one further type of network was examined and a DP model formulated for its optimal solution.

10.6. Areas for Further Study

The possibilities for further work in the general area of storm water drainage optimisation are numerous. These possibilities include the development of optimisation models for other typical types of network. It is likely that any successful work in this area will involve some form of D.P. due to the serial nature of drainage systems.

The MULTICON model requires the writing and testing of a computer program to test its validity and efficiency. Problems could arise when using network dependent design flows and some approximate procedure may be required as in the MANFIX, MANVAR and CROSSVAR models. It may indeed be of greater use for foul sewerage networks, or as the basis of an optimisation model for other distribution systems (e.g. water supply aqueducts).

The concept of ISDP needs further exploration to see whether other engineering optimisation applications exist.

MANFIX, MANVAR and CROSSVAR have been implemented using the Rational design method for calculating design flows. The practical difficulties of using, say, the TRRL hydrograph method should be explored and the resulting designs compared with Rational designs.

The models developed have assumed no possibility of detaining flood waters by ponding in special tanks, ponds or oversized pipes. As this is likely to be the object of considerable attention in future years, the incorporation of such storage items within an optimal design model should be investigated.

The existing models should be further tested to investigate their sensitivity to differing forms of cost function, as this is the factor of greatest uncertainty in any optimisation process. If the "optimal" design is very sensitive to the form of cost function then further work is required on establishing the most accurate cost functions possible. However if the designs are relatively insensitive to the form of the cost function then the optimisation models are valid without further research on costs.

MANFIX could be used as a tool to investigate the effect on the cost of a network of alterations to the design parameters. Such parameters are at present selected on the basis of judgement and experience with little knowledge of the cost penalties for being over-conservative.

A version of MANFIX for use on a mini-computer should be developed. This would enable the design of small housing estate and industrial drainage networks to be performed optimally in the smallest design office.

REFERENCES

(1) P.J.Colyer, R.W.Pethick — "Storm drainage design methods- a literature review", Hydraulics Research Station, Wallingford, 1976.

(2) Local Government Operational Research Unit — "Economics of sewerage design", Report no. C218, 1975.

(3) British Standards Institution — "British standard code of practice for sewerage", CP 2005, 1968, London.

(4) British Standards Institution — "British standards code of practice for building drainage", CP 301, 1971, London.

(5) Department of the Environment — "A guide for engineers to the design of storm sewer systems", Road Note 35, 2nd. edition, HMSO, 1976.

(6) L.H.Watkins, C.P.Young — "Developments in urban hydrology in Great Britain", Road Research Laboratory, LN/885,1965.

(7) British Standards Institution — "Clay drain and sewer pipes including surface water pipes and fittings", BS 65 and 540, Part 1, 1971, London.

(8) British Standards Institution — "Asbestos-cement pipes, joints and fittings for sewerage and drainage", BS 3656, 1973, London.

(9) British Standards Institution — "Concrete unreinforced tubes and fittings with Ogee joints for surface water drainage", BS 4101, 1967, London.

(10) British Standards Institution — "Prestressed concrete pipes for drainage and sewerage", BS 5178, 1975, London.

(11) British Standards Institution — "Pitch impregnated fibre pipes and fittings for below and above ground drainage", BS 2760, 1973, London.

(12) British Standards Institution — "Unplasticized PVC underground drain pipe and fittings", BS 4660, 1973, London.

(13) British Standards Institution — "Unplasticized PVC pipe and fittings for gravity sewers", BS 5481, 1977, London.

(14) A.D.Haith — " Vertical alignment of sewer and drainage systems by Dynamic Programming", PhD. thesis, M.I.T., Boston, 1976.

(15) R.E.Bellman — "Dynamic Programming", Princeton University Press, Princeton, New Jersey, 1957.

(16) J.S.Dajani, R.S.Gemmell, E.K.Morlok — "Optimal design of urban wastewater collection networks", Proc. ASCE, SA6, 1972, pp853-67.

(17) J.S.Dajani, Y.Hasit — "Capital cost minimisation of drainage networks", Proc. ASCE, EE2, 1974, pp 325-337.

(18) P.Yletyinen — "Optimal design of water supply and sewerage networks", Helsinki University of Technology Research Paper 48, 1973, Otaniemi, Finland.

(19) L.Von Dobschutz — "Mathematical models for the optimisation of pipe networks", Die Wasserwirtschaft, Vol.65, no.6, 1975, pp 160-164.

(20) E.H.Iman, J.A.McCorquodale, J.K.Bewtra — "Damage and cost simulation in pumped storm sewers", Canadian Journal of Civil Engineering, Vol.6, 1979, pp 129-138

(21) P.F.Lemieux, Y.Zech, R.Delarue — "Design of a stormwater sewer by nonlinear pr programming", Canadian Journal of Civil Engineering, Vol 3, 1976, pp 83-89.

(22) J.B.Rosen — "The gradient projection method for non-linear programming, part 1, linear constraints", Journal Soc. Ind.Appl. Math., Vol 8, 1960, pp 181-217.

(23) R.K.Price — "Design of storm sewers for minimum construction cost", Proc. Int. Conf. on Urban Storm Drainage, University of Southampton, 1978. Pentech Press, pp 636-47

(24) R.J.Duffin, E.L.Peterson, C.Zener — "Geometric programming- theory and application", Wiley, New York, 1967.

(25) A.J.Wilson — "Optimization in computer aided building design", PhD. thesis, University of Liverpool, 1973.

(26) D.D.Meredith — "Dynamic programming with case study on planning and design of urban water facilities", Treatise on urban water systems, Colorado State University, Fort Collins, 1971.

(27) L.B.Merrit R.H.Bogan — "Computer based optimal design of sewer systems", Proc. ASCE, EE1, 1973, pp35-53

(28) S.Walsh L.C.Brown — "Least cost method for sewer design", Proc ASCE, EE3, 1973, pp333-45

(29) S.Froise, S.J.Burges — "Least cost design of urban drainage networks", Proc. ASCE, WR1, 1978, pp75-92

(30) T.Liang — "Design of conduit system by dynamic programming", Proc. ASCE, HY3, 1971, pp383-393.

(31) G.A.Walters, A.B.Templeman — "Non-optimal dynamic programming algorithms in the design of minimum cost drainage systems", Engineering Optimisation, Vol 4, 1 79

(32) M.Heidari, V.T.Chow, P.V.Kokotovich, D.D.Meredith — "Discrete differential dynamic programming approach to water resources systems optimisation", Water Resources Research, Vol 7, 1971, pp273-82.

(33) L.W.Mays — "Optimal layout and design of storm sewer systems", PhD. thesis, University of Illinois, 1976.

(34) L.W.Mays, B.C.Yen — "Optimal cost design of branched sewer systems", Water Resources Research, Vol 11, 1975, pp 37-47.

(35) B.C.Yen, W.H.Tang, L.W.Mays — "Designing storm sewers using the rational method", Water and Sewage Works, Oct. 1974, pp92-95, and Nov. 1974, pp84-85.

(36) W.H.Tang, L.W.Mays, B.C.Yen — "Optimal risk based design of storm sewer networks", Proc. ASCE, EE3, 1975, pp 381-98.

(37) L.W.Mays, H.G.Wenzel — "Optimal design of multilevel branching sewer systems", Water Resources Research, Vol 12, 1976, pp 913-917.

(38) Y.Argaman, U.Shamir, E.Spivak — "Design of optimal sewerage systems", Proc. ASCE, EE5, 1973, pp 703-716.

(39) P.Nopmongcol, A.J.Askew — "Multilevel incremental dynamic programming", Water Resources Research, Vol 12, 1976, pp 1291-7

(40) V.T.Chow, D.R.Maidment, G.W.Tauxe — "Computer time and memory requirements for DP and DDDP in water resources systems analysis", Water Resources Research, Vol11, 1975, pp 621-28.

(41) J.C.Liebman — "A heuristic aid for the design of sewer networks", Proc. ASCE, SA4, 1967, pp 81-90.

(42) J.F.Barlow — "Cost optimisation of pipe sewerage systems", Proc. Inst. Civil Engineers, Pt 2, Vol 53, 1972, pp 57-64.

(43) I.H.Lowsley "An implicit enumeration algorithm for optimal sewer layout", PhD. thesis, John Hopkins University, Baltimore, 1973.

(44) L.W.Mays, H.G.Wenzel, J.S.Liebman "Model for layout and design of sewer systems", Proc. ASCE, Vol 102, WR2, 1976, pp 385-405.

(45) D.M.Farrar "A procedure for calculating the cost of laying rigid sewer pipes", SR 333, Transport and Road Research Laboratory, Crowthorne, Berkshire, 1977.

(46) Davis, Belfield and Everest "Architects and builders price book", 102nd edition, Spon Ltd., London, 1977.

(47) A.J.M.Harrison "An analysis of sewer costs", Internal report, Hydraulics Research Station, Wallingford, 1974.

(48) Department of the Environment "Notes for guidance, rural motorways, drainage", 1973.

(49) R.Hooke, T.A.Jeeves "Direct search solution of numerical and statistical problems", Journal A.C.M., Vol 8, 1961, pp212-29.

(50) H.H.Rosenbrock "An automatic method for finding the greatest or least value of a function", The Computer Journal, Vol 3, 1960, pp175-84.

(51) W.H.Swann "Report on the development of a new direct search method of optimisation", ICI Ltd., Note 64/3, 1964.

(52) W.C.Davidson "Variable metric method for minimisation", Argonne Nat. Lab., 1959, ANL-5990 Rev.

(53) R.Fletcher, M.J.D.Powell "A rapidly convergent descent method for minimisation", Computer Journal, Vol 6, 1963, p 163.

(54) R.Fletcher, C.M.Reeves "Function minimisation by conjugate gradients", Computer Journal, Vol 7, 1964, p149.

(55) P.E.Gill, W.Murray (eds) "Numerical methods for constrained optimisation", Academic Press, 1974.

(56) Department of the Environment "Drainage analysis program for highway networks, DAPHNE", Highway Engineering Computer Laboratory, D.o.E., London.

(57) National Computing Centre "Standard Fortran programming manual", National Computing Centre, 1972.

(58) Department of the Environment "HECB programming instruction manual, vol 3, HECB subset of NCC Fortran standards", Highway Engineering Computer Branch, D.o.E., London.

(59) Department of the Environment — "British integrated program system for highway design", D.o.E., London.

(60) L.C.W.Dixon — "Nonlinear optimisation", English Universities Press, London, 1972.

(61) D.J.Wilde, C.S.Beightler — "Foundations of optimisation", Prentice-Hall, 1967.

(62) A.B.Templeman, G.A.Walters — "Optimal design of stormwater drainage networks for roads", Proc. Inst. Civil Engineers, Pt 2, Vol 67, 1979.

APPENDIX A

Cost Calculations Based On Spon's Architects and Builders Price Book

A1 General

Cost of a pipe run between two manholes = length × cost per m run + cost of upstream manhole.

Costs are adjusted where necessary to March 1977 prices.

A2 Cost per m run.

This consists of the following items:

(1) pipe supply
(2) excavation of trench by machine
(3) layering and compaction of backfill
(4) removal of surplus backfill
(5) support for trench excavation
(6) smoothing the trench bottom by hand
(7) supply and placing of bedding and haunching material
(8) laying and jointing of pipes

The costs of these items are taken as follows:

(1) Pipe supply

Manufacturers' quotes x 1.05 for wastage.

Diameter (mm)	150	225	300	375	450	525	600	675
Cost (£ per m)	1.86	4.03	5.80	8.35	12.55	15.22	19.83	25.21
Diameter (mm)	750	825	900					
Cost (£ per m)	30.30	34.90	42.60					

(2),(3),(4) Excavation, Compaction etc.

Machine and operator for 0.11 hours + 1.26 hours labour attendance on machine per cubic metre excavated = £2.69 / m^3

(5) Trench support

Trench depth (m)	Cost per m^2 of trench wall
$y < 1.0$	zero
$1.0 \leq y < 1.5$	0.22 hours labour + 0.00165m^3 timber = £0.46
$1.5 \leq y < 3.0$	0.32 hours labour + 0.00335m^3 timber = £0.73
$3.0 \leq y < 4.5$	0.43 hours labour + 0.00335m^3 timber = £0.91

(6) Smoothing trench bottom

0.39 hours labour per m^2 = £0.56 / m^2

(7) Bedding and haunching

Cost of supply of bedding material + 2.6 hours labour per m^3

= £5.94 + £3.74 = £9.68 / m^3

(8) Laying of pipes

Diameter (mm)	150	225	300	375	450	525	600	675
Labour (hrs/m)	0.25	0.32	0.40	0.46	0.53	0.60	0.67	0.75
Cost (£/m)	0.36	0.46	0.58	0.66	0.77	0.86	0.96	1.08

Diameter (mm)	750	825	900
Labour (hrs/m)	0.82	0.89	0.95
Cost (£/m)	1.18	1.28	1.37

A3 Cost of a manhole

This consists of the following items:

(1) Excavation by machine
(2) Support of excavation walls
(3) Smoothing the bottom of the excavation by hand
(4) Placing in situ concrete base
(5) Supply and placing of precast concrete manhole rings
(6) Placing concrete benching
(7) Backfilling around manhole
(8) Removal of surplus material
(9) Supply and placing of concrete cover slab
(10) Supply and placing of brickwork, access cover and frame
(11) Supply and fitting of step irons
(12) Supply and placing of tapered ring sections (if required)

The costs of these items are taken as follows:

(1) Excavation

depth (m)	$0 < y \leq 1.5$	$1.5 < y \leq 3.0$	$3.0 < y \leq 4.5$
cost (£/m^3)	5.38	7.34	9.28

(2) Wall support

depth (m)	$0 < y \leq 1.5$	$1.5 < y \leq 3.0$	$3.0 < y \leq 4.5$
cost (£/m^3)	0.46	0.73	0.91

(3) Smoothing bottom of excavation

£0.10/m^2

(4) Placing concrete insitu base

£3.62/m^2

(5) Manhole rings

Manhole diameter (mm)	900	1050	1200	1500
Cost (£/m)	35.43	44.41	56.47	90.93

(6) Benching

£45.2/m^3

(7) Backfilling

£1.64

(8) Removal of surplus

£1.87/m^3

(9) Concrete cover slabs

Manhole diameter	900	1050	1200	1500
Cost (£)	23.11	29.26	39.47	63.24

(10) Access cover, frame and brickwork

£30.51 (lump sum)

(11) Step irons

£3.30 each

(12) Tapered ring sections

Special sections tapering to 685 mm diameter

Diameter	900	1050	1200	1500
Cost (£)	24.59	31.23	39.51	85.29

A4 Cost Functions

Using quantities taken from typical detail drawings (ref. 48), the cost functions quoted in section 4.3 have been developed.

APPENDIX B

Cost Calculations Based on Farrar (ref. 45)

B1 General

Cost of a pipe run between two manholes = length x cost per m. run + cost of upstream manhole.

Four basic cost coefficients are defined, C1, C2, C3, C4 where

cost of pipe supply = $C1(0.025+D^2)$ (D in metres)

cost of a wheeled excavator = C2 £/hr

cost of labour (general operative) =C3 £/hr

cost of granular bedding material = C4 £/m^3

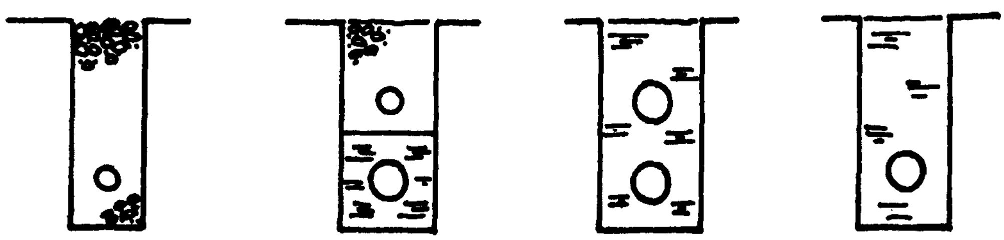
Default values of C1, C2, C3, C4 are 40, 3.5, 1.6, and 3.0 respectively.

All other costs are expressed as factors of these 4 basic rates.

An excavation factor F1 is defined for excavation in hard or difficult ground conditions. For normal conditions F1 = 1.0.

B2 Cost per m run.

Four types of drain are considered as show in the sketch below:



French (A) | French + Gully or Carrier (B) | Gully + Carrier (C) | Gully or Carrier (D)

The operations involved are:

(1) Excavation
(2) Trench support
(3) Pipe supply
(4) Distribute and lay pipes
(5) Place bedding material
(6) Place backfill or free draining mater al
(7) Compact backfill or free draining material
(8) Supply granular bedding or free draining material
(9) Remove surplus soil

The costs of these operations are as follows (all in £/m run of drain) :

(1) Excavation

Cost = b x y x α_1 x F1 where b= trench width, y =trench depth,

and α_1 = 0.097C2 + 0.130C3

(2) Trench support

$y \leq 1.5m$	Cost = 0
$1.5 < y \leq 3.0$	Cost = 0.893 α_1
$3.0 < y$	Cost = 5.03 α_1

(3) Pipe supply

Drains type A and D Cost = C1(0.025 + D^2) where D = pipe diameter (m)

Drains type B and C Cost = C1(0.05 + D^2 + $D_2^{\,2}$) where D_2 = upper pipe diameter.

(4) Distribute and lay pipes

Drains type A and D

$D < 0.3$ Cost = 0.0455C2 + 0.201C3

$D \geq 0.3$ Cost = 0.217C2 + 0.389C3

Drains type B and C (assuming $D_2 < 0.3$)

$D < 0.3$ Cost = 0.091C2 + 0.402C3

$D \geq 0.3$ Cost = 0.263C2 + 0.590C3

(5) Place bedding material

Drains type A and D Cost = 0.05C2 + 0.321C3

Drains type B and C Cost = 0.1C2 + 0.642C3

(6) Place backfill or free draining material

Cost = α_2 x v_2 where v_2 = volume of backfill or free draining material excluding bedding per m run,

and α_2 = 0.0781C2 + 0.104C3

(7) Compact backfill or free draining material

Cost α_3 x v_2 where α_3 = 0.057C2 + 0.222C3

(8) Supply granular bedding or free draining material

Cost = C4 x v_3 where v_3 = total volume of granular or free draining material per m run.

(9) Remove surplus soil from site

Cost = α_1 x v_4 x 0.912 where v_4 = volume of spoil

B3 Cost of a manhole

This consists of :

(1) Excavation
(2) Supports for excavation
(3) Place concrete base
(4) Place rings
(5) Benching
(6) Place concrete road slab
(7) Backfill
(8) Remove spoil
(9) Supply concrete
(10) Supply rings
(11) Supply slab
(12) Supply fittings
(13) Place brickwork and fittings

The costs of these operations are as follows (all in £)

(1) Excavation

Cost = α_1 F1 x Volex where Volex = volume of excavation

(2) Supports

$y \leq 1.5$	Cost = 0
$1.5 < y \leq 3.0$	Cost = length x 1.79 α_1 where length = length
$3.0 < y$	Cost = length x 10.1 α_1 of a side of square hole

(3) Place concrete base

Cost = 0.105C2 + 2.22C3

(4) Place manhole ring and joint

Cost = 0.643C2 + C3 (per ring)

(5) Place benching

Cost = 0.105C2 + 6.67C3

(6) Place concrete road slab

Cost = 0.157C2 + 0.667C3

(7) Place backfill and compact

Cost = (0.135C2 + 0.326C3) x Volback where Volback = volume of backfill.

(8) Remove surplus fill

Cost = 0.912 x α_1 x Volspoil where Volspoil = volume of spoil

(9) Supply concrete for benching and base slab

Cost = 3.67C4 x Volconc where Volconc = volume of concrete

(10) Supply precast concrete manhole rings

Cost = $0.391 \times C1 \times (diam)^2$ per m   where diam = manhole diameter (m)

(11) Supply road slab

Cost = $C1(1.04 \times diam - 0.68)$

(12) Supply fittings

ie. frame, cover, step irons, bricks

Cost = C1

(13) Place brickwork and fittings

Cost = $0.105C2 + 5.33C3$

B4 Costing Routine

Using the above unit rates the cost per m run and manhole costs are calculated by the subroutine COSTIT, which then gives the total cost of the pipe run.

COSTIT identifies the pipe type, calculates volumes for the mean pipe depth along a run and hence calculates the costs.

APPENDIX C

PROGRAM DPO

```
C     CURRENT EDITION: 26/5/78  ALTERATIONS TO GROUND CALCS,COSTS,ARRAY SIZES
      COMMON/DP/D(20),SMIN,SMAX,DMIN,DMAX,SPMIN,SPMAX,MEND,JEND,T,ND,RK
      COMMON/WHERE/IN(50,4),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIP(8000),PIP(8000),NIT(4000),PIT(4000)
      LOGICAL OK
C-----SPECIFY MAXIMUM ARRAY SIZES
      IMAX=4000
      JMAX=IMAX
      KMAX=2*IMAX
      LMAX=KMAX
C-----READ DESIGN PARAMETERS
      CALL DATA1
C-----READ SYSTEM GEOMETRY AND STORE IN PERMANENT ARRAYS
      CALL DATA2(NIP,PIP,NIT,PIT,IMAX,JMAX,KMAX,LMAX)
      LO=NIP(2)
      NO=NIP(1)
C-----SET INITIAL FLOW VALUES TO ZERO
      DO 1 I=LP1,LP2
    1 PIP(I)=0.0
C-----DEFINE FLOWS BY ASSUMING 1.5 M/S VEL IN PIPES FOR TIME TO CONC.
      N=NO-1
      DO 5 I=1,N
    5 PIT(I)=1.5
    6 OK=.TRUE.
      NCOSTS=0
      IJ=LN1+1
      CALL FLOWS(NIP,PIP,KMAX,LMAX,PIT,JMAX,T,OK)
      IF(OK) GO TO 30
C-----PROCEED WITH DESIGN
      DO 20 I=1,LO
      CALL COMB (NIP,PIP,NIT,PIT,KMAX,LMAX,IMAX,JMAX,I,NCOSTS)
      CALL NBRUN (NIP,PIP,NIT,PIT,IMAX,JMAX,KMAX,LMAX,I)
      CALL TRAIL (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,I,IJ)
   20 CONTINUE
      CALL TRACE (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX)
      GO TO 6
   30 CALL PRINT(NIP,PIP,KMAX,LMAX)
      STOP
      END
```

END OF SEGMENT, LENGTH   128, NAME DVPROG

```
      SUBROUTINE DATA1
      COMMON/DP/D(20),SMIN,SMAX,DMIN,DMAX,SPMIN,SPMAX,MEND,JEND,T,ND,RK
C-----READ IN NUMBER OF VERTICAL ZONES AND PIPE CHOICES
      READ (5,1002) MEND,JEND
      WRITE (6,2005) MEND,JEND
C-----READ IN NUMBER OF PIPE SIZES AVAILABLE AND THEIR DIAMETERS
      READ (5,1000) NP,(D(I),I=1,NP)
      WRITE (6,2000) (D(I),I=1,NP)
C-----READ IN PIPE ROUGHNESS IN MM.
      READ (5,1001) RK
      WRITE(6,2006) RK
      RK=RK/1000.0
C-----READ IN TIME OF ENTRY
      READ (5,1001) TIME
      WRITE (6,2004) TIME
      T=TIME*60.0
C-----READ IN MIN AND MAX PIPE SLOPES
      READ (5,1001) SMIN,SMAX
      WRITE (6,2001) SMIN,SMAX
C-----READ IN MIN AND MAX DEPTH OF COVER
      READ (5,1001) DMIN,DMAX
      WRITE (6,2002) DMIN,DMAX
C-----READ IN MIN AND MAX MANHOLE SPACING
      READ (5,1001) SPMIN,SPMAX
      WRITE (6,2003) SPMIN,SPMAX
C-----READ IN DIAGNOSTICS LEVEL
      READ (5,1002) ND
      RETURN
 1000 FORMAT (I6,2(/10F8.3))
 1001 FORMAT (2F8.3)
 1002 FORMAT (2I6)
 2000 FORMAT (5X,16HPIPE DIAMETERS,21X,10F8.3)
 2001 FORMAT (5X,23HMIN AND MAX PIPE SLOPES,12X,2F8.3)
 2002 FORMAT (5X,17HMIN AND MAX COVER,18X,2F8.3)
 2003 FORMAT (5X,23HMIN AND MAX M/H SPACING,2F8.3)
 2004 FORMAT (5X,13HTIME OF ENTRY,F6.1,5H MINS)
 2005 FORMAT (5X,I6,15H VERTICAL ZONES,I6,11H PIPE ZONES)
 2006 FORMAT (5X,15HPIPE ROUGHNESS=,F6.3,4H MM.)
      END
```

END OF SEGMENT, LENGTH   161, NAME DATA1

```
      SUBROUTINE DATA2 (NIP,PIP,NIT,PIT,IMAX,JMAX,KMAX,LMAX)
C-----SUBROUTINE CALLS GEOM AND SORTS OUT ADDRESSES FOR PERM.ARRAYS
      COMMON/DP/D(20),SMIN,SMAX,DMIN,DMAX,SPMIN,SPMAX,MEND,JEND,T,ND,RK
      COMMON/WHERE/IN(50,4),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX)
      CALL GEOM (NIP,PIP,KMAX,LMAX)
      LO=NIP(2)
C-----DEFINE ADDRESSES FOR ARRAYS NIP AND PIP
      NTOT=0
      LTOT=0
      MTOT=0
      DO 10 I=1,LO
      IN(I,1)=2+2*NTOT+LTOT+3*I
      IN(I,4)=1+4*NTOT+2*MTOT
      J=IN(I,1)
      NTOT=NTOT+NIP(J)
      IN(I,2)=3+2*NTOT+LTOT+3*I
      IN(I,3)=IN(I,2)+1
      J=IN(I,3)
      LTOT=LTOT+NIP(J)
      J=IN(I,2)
      MTOT=MTOT+NIP(J)
   10 CONTINUE
      LN1=2+2*NTOT+LTOT+3*(LO+1)
      LN2=LN1+MEND*JEND*NIP(4)
      LN3=LN2+MEND*JEND*LO+1
      LP1=1+4*NTOT+2*MTOT
      LP2=LP1+MEND*JEND*5
      LP3=LP2+NIP(1)+1
      WRITE (6,1000)
      WRITE (6,2000) ((IN(I,J),J=1,4),I=1,LO)
      WRITE (6,2001) LN1,LN2,LN3,LP1,LP2,LP3
      RETURN
 1000 FORMAT (1H0/10X,34HADDRESSES STORED IN ARRAY IN(LO,4))
 2000 FORMAT (4I10)
 2001 FORMAT (1H0/5X,4HLN1=,I4,5X,4HLN2=,I4,5X,4HLN3=,I4,5X,4HLP1=,I4,
     15X,4HLP2=,I4,5X,4HLP3=,I4)
      END

END OF SEGMENT, LENGTH   277, NAME  DATA2
```

```
      SUBROUTINE GEOM (NIP,PIP,KMAX,LMAX)
      DIMENSION NIP(KMAX),PIP(LMAX)
C-----PROBLEM SIZE
      READ (5,1000)  NIP(2)
C-----SET COUNTERS
      K=6
      L=0
C-----NUMBER OF BRANCHES
      LO=NIP(2)
C-----READ IN DATA FOR EACH BRANCH
      DO 60 J=1,LO
      K=K+1
      READ (5,1000) NN
      NIP(K)=NN
C-----READ IN CHAINAGES,AREAS,TOP AND BOTTOM ZONE LEVELS
      DO 10 M=1,4
      L=L+1
      N=L+NN-1
      READ(5,2000) (PIP(I),I=L,N)
      L=N
   10 CONTINUE
C-----READ IN FIRST AND LAST MANHOLES
      DO 20 M=1,2
      K=K+1
      N=K+NN-1
      READ (5,1000) (NIP(I),I=K,N)
      K=N
   20 CONTINUE
C-------READ IN GROUND LEVEL DATA
      K=K+1
      READ (5,1000) NG
      NIP(K)=NG
      DO 30 M=1,2
      L=L+1
      N=L+NG-1
      READ (5,2000) (PIP(I),I=L,N)
      L=N
   30 CONTINUE
C-----READ IN CONNECTIONS UPSTREAM
      K=K+1
      READ (5,1000) NR
      NIP(K)=NR
      IF  (NR.EQ.0) GO TO 60
      K=K+1
      N=K+NR-1
      READ (5,1000) (NIP(I),I=K,N)
      K=N
   60 CONTINUE
      READ (5,1000) NIP(1),NIP(3),NIP(4)
C-----PRINT OUT DATA AS STORED
      WRITE (6,5000)
      WRITE (6,3000) (NIP(I),I=1,K)
      WRITE (6,4000) (PIP(I),I=1,L)
      RETURN
 1000 FORMAT (16I5)
 2000 FORMAT (10F8.3)
 3000 FORMAT (20I5)
 4000 FORMAT (10F10.3)
 5000 FORMAT (1H0/10X,45HDATA STORED IN ARRAYS NIP AND PIP AS FOLLOWS:)
      END

END OF SEGMENT, LENGTH   257, NAME  GEOM
```

```
      SUBROUTINE FLOWS(NIP,PIP,KMAX,LMAX,PIT,JMAX,T,OK)
      COMMON/WHERE/IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIP(KMAX),PIP(LMAX),PIT(JMAX)
      LOGICAL OK
C-----DEFINE RETURN PERIOD
      RP=1.0
      LO =NIP(2)
      NA=0
C-----FOR EACH RUN
      DO 20 I=1,LO
      TUP=T
C-----IDENTIFY NUMBER OF U/S BRANCHES
      K=IN(I,3)
      N=NIP(K)
      IF (N.EQ.0) GO TO 5
      DO 3 J=1,N
      K=IN(I,3)+J
      K=JMAX-NIP(K)
      WRITE(6,1001) K,PIT(K)
      IF (PIT(K).GT.TUP) TUP=PIT(K)
    3 CONTINUE
    5 WRITE (6,1001) I,TUP
      K=IN(I,1)
      N=NIP(K)
      DIST1=0.0
      DO 10 J=2,N
C-----IDENTIFY DISTANCE ALONG RUN FROM LAST POSSIBLE M/H POSITION
      K=IN(I,6)+J-1
      DIST=PIP(K)-DIST1
      DIST1=PIP(K)
C------CALCULATE TIME
      NA=NA+1
      TIME=DIST/PIT(NA)+TUP
      TUP=TIME
C-----IDENTIFY AREA
      L=IN(I,1)
      K=IN(I,6)+NIP(L)+J-1
      AREA=PIP(K)
C-----CALCULATE FLOW FROM BILHAM-HOLLAND RAINFALL FORMULA
      CALL RAIN (RP,TIME/60.0,RI)
      K=LP2-1+NA
      FLOW=AREA*RI/3.6E6
      IF(ABS(FLOW-PIP(K))/FLOW.GT..01) OK=.FALSE.
      WRITE (6,1000) J,NA,DIST,PIT(NA),TIME,AREA,PIP(K),FLOW
      PIP(K)=FLOW
   10 CONTINUE
      K=JMAX-I
      PIT(K)=TIME
      WRITE (6,1000) I,K,PIT(K)
   20 CONTINUE
      RETURN
 1000 FORMAT (10X,2I5,3F9.3,F12.3,2F10.6)
 1001 FORMAT (10X,I5,F9.3)
      END

END OF SEGMENT, LENGTH   278, NAME  FLOWS
```

```
      SUBROUTINE COMB (NIP,PIP,NIT,PIT,KMAX,LMAX,IMAX,JMAX,I,NCOSTS)
      COMMON/OP/D(20),SMIN,SMAX,DMIN,DMAX,SPMIN,SPMAX,NEND,JEND,T,ND,RK
      COMMON/WHERE/IN(50,4),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIP(KMAX),PIP(LMAX),NIT(IMAX),PIT(JMAX)
      MJ=NEND*JEND
C-----HOW MANY U/S PIPES?
      K=IN(I,3)
      N=NIP(K)
      IF (N.GT.0) GO TO 30
C-----NO U/S PIPES; SET COST OF ARRIVAL AT 1ST MANHOLE IN RUN TO ZERO
      DO 10 J=1,MJ
   10 PIT(J)=0.0
      IF (I.EQ.1) GO TO 70
C-----STORE D/S COSTS FOR LAST RUN IN NEXT SECTION OF D/S COSTS ARRAY
      K=IN(I-1,1)
      NEND=NIP(K)
      J1=(NEND-1)*MJ+1
      J2=NEND*MJ
      NCOSTS=NCOSTS+1
      K=LP1+(NCOSTS-1)*MJ-1
      K1=K+1
      DO 20 J=J1,J2
      K=K+1
   20 PIP(K)=PIT(J)
      WRITE (6,1001) K1,K
      WRITE (6,1002) (PIP(IK),IK=K1,K)
      GO TO 70
C----ONE U/S PIPE; SET U/S COSTS =D/S COSTS FOR LAST RUN
   30 K=IN(I-1,1)
      NEND=NIP(K)
      J1=(NEND-1)*MJ+1
      J2=NEND*MJ
      K=0
      DO 40 J=J1,J2
      K=K+1
   40 PIT(K)=PIT(J)
      IF (N.EQ.1) GO TO 70
C-----TWO OR THREE U/S PIPES; U/S COSTS=SUM OF D/S COSTS
      M=N-1
      DO 60 L=1,M
      J=LP1+MJ*(NCOSTS-1)-1
      DO 50 K=1,MJ
      J=J+1
      PIT(K)=PIT(K)+PIP(J)
      IF (PIT(K).GT.999999.9) PIT(K)=999999.9
   50 CONTINUE
   60 NCOSTS=NCOSTS-1
   70 WRITE (6,1003) I,N
      WRITE (6,1002) (PIT(K),K=1,MJ)
      RETURN
 1001 FORMAT (5X,37HD/S COSTS,LAST RUN,STORED IN PIP FROM,I6,3H TO,I6)
 1002 FORMAT (10F12.3)
 1003 FORMAT (5X,17HU/S COSTS FOR RUN,I6,19H (NO. OF U/S PIPES=,I4,1H))
      END

END OF SEGMENT, LENGTH   317, NAME  COMB

      SUBROUTINE PRINT (NIP,PIP,KMAX,LMAX)
      DIMENSION NIP(KMAX),PIP(LMAX)
      RETURN
      END

END OF SEGMENT, LENGTH    32, NAME  PRINT
```

```
      SUBROUTINE NARUN (NIP,PIP,NIT,PIT,IMAX,JMAX,KMAX,LMAX,NRUN)
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX),LEV(20),
     1LL1(20),LL2(20),DIST(20),AREA(20)
      COMMON/WHERE/IN(50,4),LN1,LN2,LN3,LP1,LP2,LP3
      COMMON/DP/D(20),SMIN,SMAX,DMIN,DMAX,SPMIN,SPMAX,NEND,JEND,T,ND,RK
      LOGICAL NEWN
      IN1=IN(NRUN,1)
      IN2=IN(NRUN,2)
      IN3=IN(NRUN,3)
      IN6=IN(NRUN,4)-1
      LEND=NIP(IN2)
      NEND=NIP(IN1)
C-----DEFINE D/S STATE
      N=1
   20 N=N+1
C-----DEFINE PARAMETERS DEPENDENT ON "N"
      NEWN=.TRUE.
      KXN=IN6+N
      KLAN=IN1+N
      KLBN=KLAN+NEND
      XN=PIP(KXN)
      KZTOP=IN6+2*NEND+N
      KZBOT=KZTOP+NEND
      ZTOP=PIP(KZTOP)
      ZBOT=PIP(KZBOT)
      KQN=0
      IF (NRUN.EQ.1) GO TO 10
      NB=NRUN-1
      DO 5 II=1,NB
      K=IN(II,1)
    5 KQN=KQN+NIP(K)-1
   10 KQN=KQN+LP2+N-2
      J=0
   30 J=J+1
      M=0
   40 M=M+1
      MJN=((N-1)*JEND*MEND+(J-1)*MEND+M)
      PIT(MJN) =999999.9
      ZDS=ZTOP-FLOAT(M-1)*(ZTOP-ZBOT)/FLOAT(MEND-1)
C-----DEFINE UPSTREAM STATE (MANHOLE)
      NN=NIP(KLAN)-1
   44 NN=NN+1
      KXNN=IN6+NN
      XNN=PIP(KXNN)
      KZT=IN6+2*NEND+NN
      ZT=PIP(KZT)
      KZB=KZT+NEND
      ZB=PIP(KZB)
      IF (.NOT.NEWN) GO TO 150
C-----DEFINE INTERMEDIATE GROUND LEVELS
   45 DO 50 L=2,LEND
      KGXL=IN6+4*NEND+LEND+L
      IF (PIP(KGXL).GT.PIP(KXNN)+0.01)   GO TO 60
   50 CONTINUE
   60 L1=L
      KGXL1=KGXL
      KGZL1=KGXL1-LEND
      L2=L1-1
      A=0.0
      LEVEL=1
      IF (L1.EQ.LEND) GO TO 140
      IF (PIP(KGXL1+1).GT.XN+0.01) GO TO 140
      L3=L1+1
      DO 70 L=L3,LEND
      KGXL=IN6+4*NEND+LEND+L
      IF (PIP(KGXL).GT.XN-0.01) GO TO 80
   70 CONTINUE
   80 L2=L-1
      KGZL2=IN6+4*NEND+L2
      KGXL2=KGZL2+LEND
C-----DEFINE AREA OF LONG SECTION ABOVE STRAIGHT LINE
      A=-PIP(KGXL1-1)*(PIP(KGZL1)-PIP(KGZL2+1))*0.5
      DO 85 LL=L1,L2
      KGZLL=IN6+4*NEND+LL
      KGXLL=KGZLL+LEND
   85 A=A-PIP(KGXLL)*(PIP(KGZLL+1)-PIP(KGZLL-1))*0.5
      A=A-PIP(KGXL2+1)*(PIP(KGZL1-1)-PIP(KGZL2))*0.5
C-----IS GROUND LEVEL,CONCAVE,CONVEX OR VARIABLE
      GSLOPE=(PIP(KGZL2+1)-PIP(KGZL1-1))/(XN-XNN)
      DO 130 L=L1,L2
      KGZL=IN6+4*NEND+L
      KGXL=KGZL+LEND
      PSLOPE=(PIP(KGZL)-PIP(KGZL1-1))/(PIP(KGXL)-XNN)
      IF (PSLOPE.LT.GSLOPE-0.00001) GO TO 90
      IF (PSLOPE.LT.GSLOPE+0.00001) GO TO 130
      GO TO (110,100,130),LEVEL
   90 GO TO (120,130,100),LEVEL
  100 LEVEL=4
      GO TO 140
  110 LEVEL=3
      GO TO 130
  120 LEVEL=2
  130 CONTINUE
```

```
C-----DEFINE GROUND CONDITIONS AND DISTANCE BETWEEN MANHOLES
  140 NTH=NN-NIP(KLAN)+1
      LEV(NTH)=LEVEL
      LL1(NTH)=L1
      LL2(NTH)=L2
      DIST(NTH)=XN-XNN
      AREA(NTH)=A
      WRITE (6,2002) N,NN,NTH,LEVEL,L1,L2,DIST(NTH),A
C-----DEFINE UPSTREAM STATE (U/S PIPE DIAMETER AND CROWN LEVEL)
  150 NTH=NN-NIP(KLAN)+1
      JJ=0
  160 JJ=JJ+1
      MM=0
  170 MM=MM+1
      MMJJNN=(NN-1)*JEND*MEND+(JJ-1)*MEND+MM
C-----CHECK FEASIBILITY OF SOLUTION
C-----(U/S STATE FEASIBLE?)
      IF (PIT(MMJJNN).GT.999999.0) GO TO 250
C-----(PIPE SLOPE WITHIN RESTRAINTS?)
      ZUS=ZT-FLOAT(MM-1)*(ZT-ZB)/FLOAT(MEND-1)
      SLOPE=(ZUS-ZDS)/DIST(NTH)
      IF (SLOPE.LT.SMIN-0.00001) GO TO 260
      IF (SLOPE.GT.SMAX+0.00001) GO TO 250
C-----(PIPE CAPACITY SUFFICIENT? COLEBROOK-WHITE FORMULA)
      SQ=SQRT(SLOPE*D(J))
      QFULL=-6.927*D(J)*D(J)*SQ*ALOG10(RK/3.7/D(J)+0.6671E-6/SQ/D(J))
      IF (QFULL.LT.PIP(KQN)) GO TO 250
C-----(DEPTH OF COVER RESTRAINTS VIOLATED?)
      LEVEL=LEV(NTH)
      L1=LL1(NTH)
      L2=LL2(NTH)
      GO TO (240,180,220,180),LEVEL
  180 DO 200 L=L1,L2
      KGXL=IN4+4*NEND*LEND+L
      KGZL=KGXL-LEND
      IF (ZUS-SLOPE*(PIP(KGXL)-XNN).GT.PIP(KGZL)-DMIN+0.01) GO TO 250
  200 CONTINUE
  210 IF (LEVEL.EQ.2) GO TO 240
  220 DO 230 L=L1,L2
      KGXL=IN4+4*NEND*LEND+L
      KGZL=KGXL-LEND
      IF (ZUS-SLOPE*(PIP(KGXL)-XNN).LT.PIP(KGZL)-DMAX) GO TO 260
  230 CONTINUE
C-----SOLUTION IS FEASIBLE SO COST AND COMPARE WITH PREVIOUS CHEAPEST
  240 KGZL1=IN4+4*NEND+L1
      KGZL2=KGZL1-L1+L2
      CALL CORTIT (J,AREA(NTH),PIP(KGZL1+1)-ZUS,PIP(KGZL2+1)-ZDS,
     1DIST(NTH),C)
      C=C+PIT(MMJJNN)
      IF(C.GT.PIT(MJN)-0.001) GO TO 250
      PIT(MJN)=C
      NIT(MJN)=(NN-1)*JEND*MEND+(JJ-1)*MEND+MM
      IF (ND.EQ.2)      WRITE(6,2005) N,J,M,NN,JJ,MM,NIT(MJN),C
C-----MOVE ON TO NEXT U/S STATE
  250 IF (MM.LT.MEND) GO TO 170
  260 IF (JJ.LT.J) GO TO 160
      IF (NN.LT.NIP(KLBN)) GO TO 66
C-----MOVE ON TO NEXT D/S STATE
      NEWNN=.FALSE.
      IF (M.LT.MEND) GO TO 40
      IF(J.LT.JEND) GO TO 30
      IF (N.LT.NEND) GO TO 20
      RETURN
 2002 FORMAT (1H0,15HGROUND FROM M/H,I4,6HTO M/H,5I4,2(F9.3))
 2005 FORMAT (10X,3HD/S,3I5,10X,3HU/S,6I5,F15.3)
      END

END OF SEGMENT, LENGTH   942, NAME  NRRUN
```

```
      SUBROUTINE COSTIT (J,AREA,DUS,DDS,DIST,COST)
      DEPTH=(DUS+DDS)/2.0+AREA/DIST
      GO TO (10,20,30,40,50,60,70) ,J
   10 COST=DIST*(2.8+6.1*DEPTH)+30.0+70.0*DUS
      RETURN
   20 COST=DIST*(5.7+6.1*DEPTH)+30.0+70.0*DUS
      RETURN
   30 COST=DIST*(8.9+6.1*DEPTH)+30.0+75.0*DUS
      RETURN
   40 COST=DIST*(12.3+6.6*DEPTH)+30.0+80.0*DUS
      RETURN
   50 COST=DIST*(15.9+6.7*DEPTH)+30.0+85.0*DUS
      RETURN
   60 COST=DIST*(19.7+5.0*DEPTH)+30.0+90.0*DUS
      RETURN
   70 COST=DIST*(23.7+5.3*DEPTH)+30.0+95.0*DUS
      RETURN
      END
```

END OF SEGMENT, LENGTH 136, NAME COSTIT

```
      SUBROUTINE TRAIL (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,NRUN,IJ)
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX)
      COMMON/DP/D(20),SMIN,SMAX,DMIN,DMAX,SPMIN,SPMAX,MEND,JEND,T,ND,RK
      COMMON/WHERE/IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
      IN1=IN(NRUN,1)
      NEND=NIP(IN1)
      K=LN2-1+MEND*JEND*(NRUN-1)
      MJNEND=(NEND-1)*JEND*MEND
      N=NEND*MEND*JEND
      DO 60 J=1,JEND
      DO 60 M=1,MEND
      K=K+1
C-----NIP(NIP(K)) IS 1ST U/S REFERENCE NUMBER IN TRACE BACK FROM D/S
C-----STATE (J,M)
      N=N+1
C-----NIT(N)=REFERENCE BACK ACROSS M/H FROM D/S STATE(J,M)
      MJNEND=MJNEND+1
C-----PIT(MJNEND)=COST OF SOLUTION TO D/S STATE(J,M)
C-----ASSUME NO CHANGE OF REFERENCE ACROSS THE M/H
      NIT(N)=MJNEND
C-----IF COST OF ARRIVAL AT A HIGHER LEVEL IS CHEAPER ADOPT THIS COST
C-----AND ALTER REFERENCE
      IF (M.LE.1) GO TO 10
      IF (PIT(MJNEND).LT.PIT(MJNEND-1)) GO TO 10
      PIT(MJNEND)=PIT(MJNEND-1)
      NIT(N)=NIT(N-1)
C-----IF COST OF ARRIVAL WITH A SMALLER DIAMETER IS CHEAPER,ADOPT THIS
C-----COST AND ALTER REFERENCE
   10 IF (J.LE.1) GO TO 20
      I=MJNEND-MEND
      IF (PIT(MJNEND).LT.PIT(I)) GO TO 20
      PIT(MJNEND)=PIT(I)
      I=N-NEND
      NIT(N)=NIT(I)
   20 CONTINUE
C---1ST U/S REF.NO. IN TRACE BACK FROM D/S STATE(J,M) IS REF. ACROSS M/H
      IJ=IJ+1
      NIP(K)=IJ
      NIP(IJ)=NIT(N)
      IF(PIT(MJNEND).GT.999999.0) NIP(IJ)=0
C-----ESTABLISH THE REST OF THE TRACE BACK
   30 MAJANA=NIP(IJ)
      IJ=IJ+1
      IF(MAJANA.LT.MEND*JEND) GO TO 40
      NIP(IJ)=NIT(MAJANA)
      GO TO 30
C-----DEFINE START AND END POINTS IN ARRAY NIP FOR TRACE BACK FROM D/S
C-----STATE(J,M)
C-----IF DIAGNOSTICS=1 OR 2,PRINT OUT TRACE BACK FOR D/S STATE
   40 IJ=IJ-1
      IF (ND.EQ.0) GO TO 60
      IK=NIP(K)
      WRITE(6,2003) J,M,PIT(MJNEND)
      DO 50 II=IK,IJ
      IDM=(NIP(II)-1)/(MEND*JEND)+1
      IDD=(NIP(II)-1-(IDM-1)*MEND*JEND)/MEND+1
      IDZ=NIP(II)-(IDM-1)*MEND*JEND-(IDD-1)*MEND
   50 WRITE (6,2006) IDM,IDD,IDZ
   60 CONTINUE
      RETURN
 2003 FORMAT (1H0,10X,3HJ= ,I3,5X,3HM= ,I3,10X,19HM/H   U/S D   LEVEL,
     15X,5HCOST=,F12.3)
 2006 FORMAT (32X,3I8)
      END.
```

END OF SEGMENT, LENGTH 384, NAME TRAIL

```
      SUBROUTINE TRACE (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX)
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX),JDS(10),MDS(10)
      COMMON/DP/ D(20),SMIN,SMAX,DMIN,DMAX,SPMIN,SPMAX,MEND,JEND,T,ND,RK
      COMMON/WHERE/ IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
C-----TOTAL NUMBER OF BRANCHES AND MANHOLES
      LO=NIP(2)
      NO=NIP(1)
C-----NUMBER OF MANHOLES IN LAST RUN
      K=IN(LO,1)
      NEND=NIP(K)
C-----START AND END ELEMENTS FOR FINAL COSTS IN ARRAY PIT
      MJ=MEND*JEND
      I1=(NEND-1)*MJ+1
      I2=NEND*MJ
C------WHICH IS CHEAPEST
      ICOST=0
      COST=999999.8
      DO 10 I=I1,I2
      IF (PIT(I).GE.COST) GO TO 10
      COST=PIT(I)
      ICOST=I
   10 CONTINUE
C-----NO FEASIBLE SOLUTION?
      IF (ICOST.EQ.0) STOP
C------IDENTIFY THE DOWNSTREAM STATE
      I=ICOST-I1+1
      J=(I-1)/MEND+1
      M=I-(J-1)*MEND
      NRUN=LO
      L=0
      LL=0
C-----IDENTIFY START AND END ELEMENTS K1,K2 IN ARRAY NIP FOR TRACE BACK
C-----UP BRANCH GIVEN RUN NUMBER ,J AND M
   15 K=LN2-1+(NRUN-1)*JEND*MEND+(J-1)*MEND+M
      K1=NIP(K)
      K2=NIP(K+1)-1
      K=IN(NRUN,1)
      NEND=NIP(K)
      N3=0
      DO 16 NN=1,NRUN
      K=IN(NN,1)
   16 N3=N3+NIP(K)+1
C-----TRACE BACK ALONG BRANCH NRUN FROM (J,M)
      DO 20 K=K1,K2
      L=L+1
      NIT(L)=NRUN
      NA=(NIP(K)-1)/(MEND*JEND)+1
      L=L+1
      NIT(L)=NA
      JA=(NIP(K)-1-(NA-1)*MEND*JEND)/MEND+1
      L=L+1
      NIT(L)=JA
      MA=NIP(K)-(NA-1)*MEND*JEND-(JA-1)*MEND
      L=L+1
      NIT(L)=MA
C-----IDENTIFY SOFFIT LEVEL
      KK1=IN(NRUN,6)+2*NEND-1+NA
      KK2=KK1+NEND
      ZUS=PIP(KK2)+(PIP(KK1)-PIP(KK2))*FLOAT(MEND-MA)/FLOAT(MEND-1)
C-----IDENTIFY DIAMETER AND M/H NUMBER
      DIAMUS=D(JA)
      MHUS=NA
C------IDENTIFY CHAINAGE
      KK1=IN(NRUN,6)-1+NA
      XUS=PIP(KK1)
      IF (K.EQ.K1) GO TO 31
C-----CALCULATE PIPE SLOPE
      SLOPE=(ZUS-ZDS)/(XDS-XUS)
C-----CALCULATE VELOCITY FROM COLEBROOK-WHITE FORMULA
      SQ=SQRT(SLOPE*DIAM)
      VEL=-8.018*SQ*ALOG10(RK/3.7/DIAM+0.6671E-6/SQ/DIAM)
C-----STORE VELOCITY
      N1=N3-NEND+MHUS+1
      N2=N1-MHUS+MHDS-1
      DO 30 N=N1,N2
   30 PIT(N)=VEL
C-----MOVE ON TO NEXT MANHOLE
   31 ZDS=ZUS
      XDS=XUS
      DIAM=DIAMUS
      MHDS=MHUS
   20 CONTINUE
```

```
C-----NOW AT UPSTREAM END OF RUN
C-----HOW MANY UPSTREAM RUNS?
      K=IN(NRUN,3)
      K=NIP(K)+1
      GO TO (70,60,50,40),K
   40 LL=LL+1
      JDS(LL)=JA
      MDS(LL)=MA
   50 LL=LL+1
      JDS(LL)=JA
      MDS(LL)=MA
   60 NRUN=NRUN-1
      J=JA
      M=MA
      GO TO 15
   70 IF (NRUN.EQ.1) GO TO 80
      NRUN=NRUN-1
      J=JDS(LL)
      M=MDS(LL)
      LL=LL-1
      GO TO 15
   80 WRITE (6,1003) COST
      WRITE (6,1001) (NIT(II),II=1,L)
      WRITE (6,1006)
      WRITE (6,1002) (PIT(II),II=1,NO)
C-----IF DIAGNOSTIC(S=1 OR 2,PRINT OUT STORAGE ARRAYS
      IF (ND.EQ.0) RETURN
      WRITE (6,5002) (NIT(I),I=1,IMAX)
      WRITE (6,5002) (NIP(I),I=1,KMAX)
      WRITE (6,5003) (PIT(I),I=1,JMAX)
      WRITE (6,5003) (PIP(I),I=1,LMAX)
      RETURN
 1001 FORMAT (10X,3HRUN,I3,5X,3HM/H,I3,5X,8HU/S DIAM,I3,5X,5HLEVEL,I3)
 1002 FORMAT (10F12.3)
 1003 FORMAT (1H0/10X,31HTRACE BACK OF CHEAPEST SOLUTION,5X,5HCOST=,
     1F12.3)
 1006 FORMAT (1H0//5X,29HNEW FULL FLOW PIPE VELOCITIES/)
 5002 FORMAT (20I6)
 5003 FORMAT (10E12.5)
      END

END OF SEGMENT, LENGTH   610, NAME  TRACE
```

```
      SUBROUTINE RAIN (Y,T,RI)
C-----Y=RETURN PERIOD(YRS),T=TIME(MINS),RI=INTENSITY (MM/HR)
      RI=60./T*((Y*T*202.26)**.28169-2.56)
      IF (RI-33.) 10,10,40
   10 IF(RI) 20,30,30
   20 RI=0.0
   30 RETURN
   40 NI=0
C-----ITERATION LOOP FOR HOLLAND FORMULA
   50 F=ALOG(15260.0/Y/RI/T*(RI*T/1526.0+0.1)**3.55)-1.0+1.0316*RI
      DF=-1.0/RI+3.55/(RI+152.6/T)+.0316
      X=F/DF
      RI=RI-X
      IF (ABS(X)-0.01) 10,10,60
   60 IF (NI-10) 80,80,70
C-----ERROR
   70 STOP
   80 NI=NI+1
      GO TO 50
      END

END OF SEGMENT, LENGTH    94, NAME  RAIN

      FINISH
```

APPENDIX D

PROGRAM ASSEMB

```
C-----READS DATA IN SIMPLIFIED FORM AND OUTPUTS TO CARD PUNCH FILE FOR
C      PROGRAMS DPO AND MOD.     SUITABLE FOR INTERACTIVE USE
      DIMENSION D(20),GL(300),GX(300),DIST(300),KA(300),KB(300),X(300),
     1Z(300),GROUND(300),OBLEV(20),OBDIST(20),NBR(10),AREADS(30),
     2ZTOPDS(30),ZTOP(300),ZBOT(300),AREA(300)
      E=0.000001
C-----READ IN BASIC DESIGN PARAMETERS
      WRITE (6,1001)
      READ (5,2001) SMIN
      IF (SMIN.LE.E) SMIN=0.004
      SMAX=0.1
      WRITE (6,1002)
      READ (5,2001) VELMIN
      IF (VELMIN.LE.E) VELMIN=0.7
      WRITE (6,1003)
      READ (5,2001) VELMAX
      IF (VELMAX.LE.E) VELMAX=6.0
      WRITE (6,1004)
      READ (5,2001) DMIN
      IF (DMIN.LE.E) DMIN=1.0
      WRITE (6,1005)
      READ (5,2001) DMAX
      IF (DMAX.LE.E) DMAX=6.0
      WRITE (6,1006)
      READ (5,2001) TIME
      IF (TIME.LE.E) TIME=2.0
      WRITE (6,1007)
      READ (5,2001) RK
      IF (RK.LE.E) RK=0.15
      WRITE (6,1008)
      READ (5,2001) SPMIN
      IF (SPMIN.LE.E) SPMIN=30.0
      WRITE (6,1009)
      READ (5,2001) SPMAX
      IF (SPMAX.LE.E) SPMAX=150.0
C-----READ IN PIPE DATA
      WRITE (6,1010)
      READ (5,2002) NP
      IF (NP.EQ.0) GO TO 20
      DO 10 I=1,NP
      WRITE (6,1011)
      READ (5,2001) SIZE
      D(I)=SIZE/1000.0
   10 CONTINUE
      GO TO 30
   20 NP=7
      D(1)=.150
      D(2)=.225
      D(3)=.300
      D(4)=.375
      D(5)=.450
      D(6)=.525
      D(7)=.600
C-----READ IN PROGRAM CONTROL VALUES
   30 WRITE (6,1012)
      READ (5,2001) DZ
      IF (DZ.LE.E) DZ=0.5
      WRITE (6,1013)
      READ (5,2002)  MEND
      IF (MEND.EQ.0) MEND=5
      WRITE (6,1014)
      READ (5,2002) JEND
      IF (JEND.EQ.0) JEND=4
      WRITE (6,1015)
      READ (5,2002) ND
      WRITE (6,1016)
      READ (5,2001) SPMH
      IF (SPMH.LE.E)   SPMH=10.0
C-----READ IN NO. OF BRANCHES IN DESIGN PROBLEM
      WRITE (6,1017)
      READ (5,2002) LO
C-----WRITE OUT DATA TO FORMATTED FILE
      WRITE (1,3001) MEND,JEND,NP
      WRITE (1,3002) (D(I),I=1,NP)
      WRITE (1,3002) RK
      WRITE (1,3002) TIME
      WRITE (1,3002) SMIN,SMAX
      WRITE (1,3002) DMIN,DMAX
      WRITE (1,3002) SPMIN,SPMAX
      WRITE (1,3001) ND
      WRITE (1,3003) LO
      DISTOT=0.0
      KNTOT=0
      KLTOT=0
C-----READ IN DATA FOR EACH BRANCH
      DO 290 I=1,LO
      WRITE (6,1018) I
      READ (5,2003) ITYPE,Y,DW
C------READ IN GROUND LEVEL DATA
      J=0
   40 J=J+1
      WRITE (6,1019)
      READ (5,2001) GL(J)
      WRITE (6,1020)
      READ (5,2001) GX(J)
      IF(GX(J).LT.Y-0.1)      GO TO 40
```

```
C-----GENERATE POSITIONS OF MANHOLES
      IF (ITYPE.EQ.0) GO TO 70
   60 DIST(1)=0.0
      DIST(2)=Y
      NN=2
      GO TO 90
C-----IS LENGTH LESS THAN TWICE MINIMUM SPACING?
   70 IF (Y.LT.2.0*SPMIN-0.1) GO TO 60
      DIST(1)=0.0
      DIST(2)=AMAX1(SPMIN,SPMH)
      NN=3+INT((Y-DIST(2)-SPMIN-0.1)/SPMH)
      DO 80 K=3,NN
      DIST(K)=DIST(K-1)+SPMH
      IF (K.EQ.NN) DIST(K)=Y
   80 CONTINUE
C-----GENERATE PERMISSIBLE MANHOLE CONNECTIONS
      KA(1)=1
      KB(1)=1
      DO 340 M=2,NN
      DO 300 L=1,M
      IF (DIST(M)-DIST(L).GT.SPMAX+0.1) GO TO 300
      KA(M)=L
      GO TO 310
  300 CONTINUE
  310 DO 320 L=1,M
      IF (DIST(M)-DIST(L).LT.SPMIN-0.1) GO TO 330
  320 CONTINUE
  330 KB(M)=L-1
  340 CONTINUE
C-----CALCULATE GROUND LEVELS
   90 GROUND(1)=GL(1)
      IF(NN.LT.3) GO TO 130
      KK=NN-1
      DO 120 K=2,KK
      DO 100 M=1,J
      IF (GX(M).GT.DIST(K)+0.01) GO TO 110
  100 CONTINUE
  110 GROUND(K)=GL(M-1)+(GL(M)-GL(M-1))/(GX(M)-GX(M-1))*
     1(DIST(K)-GX(M-1))
  120 CONTINUE
  130 GROUND(NN)=GL(J)
      KM=1
      KN=1
      KL=0
  140 KL=KL+1
      IF (DIST(KM)-GX(KN).GT.0.01) GO TO 145
      X(KL)=DIST(KM)
      Z(KL)=GROUND(KM)
      IF (KM.EQ.NN) GO TO 146
      IF (DIST(KM)-GX(KN).GT.-0.01) KN=KN+1
      KM=KM+1
      GO TO 140
  145 X(KL)=GX(KN)
      Z(KL)=GL(KN)
      KN=KN+1
      GO TO 140
C-----READ IN DETAILS OF OBSTRUCTIONS
  146 NOB=0
  150 NOB=NOB+1
      WRITE (6,1021)
      READ (5,2001) OBLEV(NOB)
      IF (OBLEV(NOB).LE.-998.9) GO TO 160
      WRITE (6,1020)
      READ (5,2001) OBDIST(NOB)
      GO TO 150
  160 NOB=NOB-1
C-----READ IN UPSTREAM CONNECTIONS
      NUB=0
  170 NUB=NUB+1
      WRITE (6,1022)
      READ (5,2002) NBR(NUB)
      IF (NBR(NUB).EQ.0) GO TO 180
      GO TO 170
  180 NUB=NUB-1
C-----IDENTIFY TOP OF ZONE AT UPSTREAM END OF RUN
      ZTOPUS=GROUND(1)-DMIN
      AREAUS=0.0
      IF (NUB.EQ.0) GO TO 200
      DO 190 JJ=1,NUB
      K=NBR(JJ)
      AREAUS=AREAUS+AREADS(K)
      IF (ZTOPDS(K).LT.ZTOPUS) ZTOPUS=ZTOPDS(K)
  190 CONTINUE
```

```
C-----DEFINE TOP OF ZONE ALONG THE BRANCH
  200 ZIN=ZTOPUS
      ZOB=999999.9
      ZGL=999999.9
      DO 270 JJ=1,NN
      ZOUT=GROUND(JJ)-DMIN
      ZTOP(JJ)=AMIN1(ZIN,ZOB,ZGL,ZOUT)
      IF (JJ.EQ.NN) GO TO 270
      ZIN=ZTOP(JJ)-SMIN*(DIST(JJ+1)-DIST(JJ))
      IF(NOB.EQ.0) GO TO 220
      DO 210 K=1,NOB
      IF (OBDIST(K).GT.DIST(JJ).AND.OBDIST(K).LT.DIST(JJ+1)) GO TO 230
  210 CONTINUE
  220 ZOB=999999.9
      GO TO 240
  230 ZOB=OBLEV(K)-SMIN*(DIST(JJ+1)-OBDIST(K))
  240 DO 250 K=1,J
      IF (GX(K).GT.DIST(JJ).AND.GX(K).LT.DIST(JJ+1)) GO TO 260
  250 CONTINUE
      ZGL=999999.9
      GO TO 270
  260 ZGL=GL(K)-SMIN*(DIST(JJ+1)-GX(K))
  270 CONTINUE
      ZTOPDS(I)=ZTOP(NN)
      IF ((ZTOPUS-ZTOPDS(I))/Y.LT.SMAX-0.00001) GO TO 275
      ZTOPUS=ZTOPDS(I)+(SMAX-0.00002)*Y
      GO TO 200
  275 DO 280 JJ=1,NN
      ZBOT(JJ)=ZTOP(JJ)-DZ
  280 AREA(JJ)=AREAUS+DIST(JJ)*DW
      AREADS(I)=AREA(NN)
      NNTOT=NNTOT+NN-1
      KLTOT=KLTOT+KL-1
      DISTOT=DISTOT+Y
C-----WRITE DATA TO FILE
      WRITE (1,3003) NN
      WRITE (1,3005) (DIST(K),K=1,NN)
      WRITE (1,3004) (AREA(K),K=1,NN)
      WRITE (1,3002) (ZTOP(K),K=1,NN)
      WRITE (1,3002) (ZBOT(K),K=1,NN)
      WRITE (1,3003) (KA(K),K=1,NN)
      WRITE (1,3003) (KB(K),K=1,NN)
      WRITE (1,3003)  KL
      WRITE (1,3002) (Z(K),K=1,KL)
      WRITE (1,3005) (X(K),K=1,KL)
      WRITE (1,3003) NUB
      IF (NUB.EQ.0) GO TO 290
      WRITE (1,3003) (NBR(K),K=1,NUB)
  290 CONTINUE
C-----PROBLEM SIZE
      NOMHAV=LO+INT(DISTOT/50.0)+1
      NNTOT=NNTOT+1
      KLTOT=KLTOT+1
      WRITE (1,3003) NNTOT,KLTOT,NOMHAV
      WRITE (6,1023)
      STOP
 1001 FORMAT (10X,36H****FOR DEFAULT VALUE INPUT ZERO****/5X,
     117HMINIMUM GRADIENT=)
 1002 FORMAT (5X,17HMINIMUM VELOCITY=)
 1003 FORMAT (5X,17HMAXIMUM VELOCITY=)
 1004 FORMAT (5X,14HMINIMUM COVER=)
 1005 FORMAT (5X,14HMAXIMUM COVER=)
 1006 FORMAT (5X,20HTIME OF ENTRY(MINS)=)
 1007 FORMAT (5X,15HPIPE ROUGHNESS=)
 1008 FORMAT (5X,24HMINIMUM MANHOLE SPACING=)
 1009 FORMAT (5X,24HMAXIMUM MANHOLE SPACING=)
 1010 FORMAT (5X,61HFOR LIBRARY PIPE SIZES ENTER ZERO;OTHERWISE ENTER NO
     1.OF PIPES)
 1011 FORMAT (5X,14HPIPE SIZE(MM)=)
 1012 FORMAT (5X,14HDEPTH OF ZONE=)
 1013 FORMAT (5X,17HNUMBER OF LEVELS=)
 1014 FORMAT (5X,16HNUMBER OF PIPES=)
 1015 FORMAT (5X,18HDIAGNOSTICS LEVEL=)
 1016 FORMAT (5X,29HSPACING OF POSSIBLE MANHOLES=)
 1017 FORMAT (5X,19HNUMBER OF BRANCHES=)
 1018 FORMAT (5X,10HBRANCH NO.,I6,5X,43HENTER TYPE(0 OR 1),LENGTH AND DR
     1AINED WIDTH)
 1019 FORMAT (5X,18HENTER GROUND LEVEL)
 1020 FORMAT (5X,36HENTER DISTANCE FROM UPSTREAM MANHOLE)
 1021 FORMAT (5X,42HENTER OBSTRUCTION LEVEL(-999 TO TERMINATE))
 1022 FORMAT (5X,48HENTER UPSTREAM BRANCH NUMBER (ZERO TO TERMINATE))
 1023 FORMAT (5X,10(1H*),18HEXECUTION FINISHED,10(1H*))
 2001 FORMAT (F0.0)
 2002 FORMAT (I0)
 2003 FORMAT(I0,2F0.0)
 3001 FORMAT (2I6)
 3002 FORMAT (10F8.3)
 3003 FORMAT (16I5)
 3004 FORMAT (10F8.0)
 3005 FORMAT (10F8.1)
      END
```

APPENDIX E

PROGRAM MOD

```
CURRENT EDITION:24.6.76, LARGE SIZE VERSION
      COMMON/DP/D(20),GMIN(20),GMAX(20),DMIN,DMAX,MEND,JEND,T,ND,RK,RP
      COMMON/UHERE/IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIP(20000),PIP(10000),NIT(13000),PIT(13000)
      LOGICAL OK
C-----SPECIFY MAXIMUM ARRAY SIZES
      IMAX=13000
      JMAX=13000
      KMAX=20000
      LMAX=10000
C-----READ DESIGN PARAMETERS
      CALL DATA1
C-----READ SYSTEM GEOMETRY AND STORE IN PERMANENT ARRAYS
      CALL DATA2(NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX)
      LO=NIP(2)
      NO=NIP(1)
C-----SET INITIAL FLOW VALUES TO ZERO
      DO 1 I=LP1,LP2
    1 PIP(I)=0.0
      NCOSTS=0
      IJ=LN1-1
C-----PRODUCE A MINIMUM GRADIENT DESIGN
      CALL MGRAD (NIP,PIP,KMAX,LMAX)
C-----PRODUCE OPTIMUM DESIGN BASED ON MINIMUM GRADIENT FLOWS
      DO 20 I=1,LO
      CALL COMB (NIT,PIT,NIP,PIP,KMAX,LMAX,IMAX,JMAX,I,NCOSTS)
      CALL NBRUN (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,I)
      CALL TRAIL (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,I,IJ)
   20 CONTINUE
      CALL TRACE (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,L)
C-----PRODUCE FINAL DESIGN BY ALTERING GRADIENTS
      CALL LEVELS (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,L)
      STOP
      END
```

```
      SUBROUTINE DATA1
      COMMON/DP/D(20),GMIN(20),GMAX(20),DMIN,DMAX,MEND,JEND,T,ND,RK,RP
C-----READ IN NUMBER OF VERTICAL ZONES AND PIPE CHOICES
      READ (5,1002) MEND,JEND
      WRITE (6,2005) MEND,JEND
C-----READ IN NUMBER OF PIPE SIZES AVAILABLE AND THEIR DIAMETERS
      READ (5,1000) NP,(D(I),I=1,NP)
      WRITE (6,2000) (D(I),I=1,NP)
C-----READ IN PIPE ROUGHNESS IN MM.
      READ (5,1001) RK
      WRITE(6,2006) RK
      RK=RK/1000.0
C-----READ IN TIME OF ENTRY
      READ (5,1001) TIME
      WRITE (6,2004) TIME
      T=TIME*60.0
C-----READ IN MIN AND MAX PIPE SLOPES
      READ (5,1001) SMIN,SMAX
      WRITE (6,2001) SMIN,SMAX
C-----FIX MIN AND MAX GRADIENTS FOR EACH PIPE SIZE
      DO 10 I=1,NP
      GMIN(I)=SMIN
   10 GMAX(I)=SMAX
C-----READ IN MIN AND MAX DEPTH OF COVER
      READ (5,1001) DMIN,DMAX
      WRITE (6,2002) DMIN,DMAX
C-----READ IN MIN AND MAX MANHOLE SPACING
      READ (5,1001) SPMIN,SPMAX
      WRITE (6,2003) SPMIN,SPMAX
C-----READ IN DIAGNOSTICS LEVEL
      READ (5,1002) ND
      RETURN
 1000 FORMAT (I6,2(/10F8.3))
 1001 FORMAT (2F8.3)
 1002 FORMAT (2I6)
 2000 FORMAT (5X,16HPIPE DIAMETERS,21X,10F8.3)
 2001 FORMAT (5X,23HMIN AND MAX PIPE SLOPES,12X,2F8.3)
 2002 FORMAT (5X,17HMIN AND MAX COVER,18X,2F8.3)
 2003 FORMAT (5X,23HMIN AND MAX M/H SPACING,2F8.3)
 2004 FORMAT (5X,13HTIME OF ENTRY,F6.1,5H MINS)
 2005 FORMAT (5X,I6,15H VERTICAL ZONES,I6,11H PIPE ZONES)
 2006 FORMAT (5X,15HPIPE ROUGHNESS=,F6.3,4H MM.)
      END
```

```
      SUBROUTINE DATA2 (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX)
C-----SUBROUTINE CALLS GEOM AND SORTS OUT ADDRESSES FOR PERM.ARRAYS
      COMMON/RD/D(20),SMIN(20),SMAX(20),DMIN,DMAX,HEND,JEND,T,ND,NK,NP
      COMMON/UNFRE/IN(50,4),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX)
      CALL GEOM (NIP,PIP,KMAX,LMAX)
      LO=NIP(2)
C-----DEFINE ADDRESSES FOR ARRAYS NIP AND PIP
      NTOT=0
      LTOT=0
      MTOT=0
      DO 10 I=1,LO
      IN(I,1)=2+2*NTOT+LTOT+3*I
      IN(I,4)=1+4*NTOT+2*MTOT
      J=IN(I,1)
      NTOT=NTOT+NIP(J)
      IN(I,2)=3+2*NTOT+LTOT+3*I
      IN(I,3)=IN(I,2)+1
      J=IN(I,3)
      LTOT=LTOT+NIP(J)
      J=IN(I,2)
      MTOT=MTOT+NIP(J)
   10 CONTINUE
      LN1=2+2*NTOT+LTOT+3*(LO+1)
      LN2=LN1+HEND*JEND*NIP(4)
      LN3=LN2+HEND*JEND*LO
      LP1=1+4*NTOT+2*MTOT
      LP2=LP1+HEND*JEND*5
      LP3=LP2+NIP(1)-1
      IF (ND.EQ.0) RETURN
      WRITE (6,1000)
      WRITE (6,2000) ((IN(I,J),J=1,4),I=1,LO)
      WRITE (6,2001) LN1,LN2,LN3,LP1,LP2,LP3
      RETURN
 1000 FORMAT (1H0/10X,34HADDRESSES STORED IN ARRAY IN(LO,4))
 2000 FORMAT (4I10)
 2001 FORMAT (1H0/5X,4HLN1=,I4,5X,4HLN2=,I4,5X,4HLN3=,I4,5X,4HLP1=,I4,
     15X,4HLP2=,I4,5X,4HLP3=,I4)
      END
```

```
      SUBROUTINE GEOM (NIP,PIP,KMAX,LMAX)
      COMMON/DP/D(20),SMIN(20),SMAX(20),DMIN,DMAX,MEND,JEND,T,ND,RK,RP
      DIMENSION NIP(KMAX),PIP(LMAX)
C-----PROBLEM SIZE
      READ (5,1000)  NIP(2)
C-----SET COUNTERS
      K=6
      L=0
C-----NUMBER OF BRANCHES
      LO=NIP(2)
C-----READ IN DATA FOR EACH BRANCH
      DO 40 J=1,LO
      K=K+1
      READ (5,1000) NN
      NIP(K)=NN
C-----READ IN CHAINAGES,AREAS,TOP AND BOTTOM ZONE LEVELS
      DO 10 M=1,4
      L=L+1
      N=L+NN-1
      READ(5,2000) (PIP(I),I=L,N)
      L=N
   10 CONTINUE
C-----READ IN FIRST AND LAST MANHOLES
      DO 20 M=1,2
      K=K+1
      N=K+NN-1
      READ (5,1000) (NIP(I),I=K,N)
      K=N
   20 CONTINUE
C-----READ IN GROUND LEVEL DATA
      K=K+1
      READ (5,1000) NG
      NIP(K)=NG
      DO 30 M=1,2
      L=L+1
      N=L+NG-1
      READ (5,2000) (PIP(I),I=L,N)
      L=N
   30 CONTINUE
C-----READ IN CONNECTIONS UPSTREAM
      K=K+1
      READ (5,1000) NR
      NIP(K)=NR
      IF  (NR.EQ.0) GO TO 40
      K=K+1
      N=K+NR-1
      READ (5,1000) (NIP(I),I=K,N)
      K=N
   40 CONTINUE
      READ (5,1000) NIP(1),NIP(3),NIP(4)
C-----PRINT OUT DATA AS STORED
      IF (ND.EQ.0) RETURN
      WRITE (6,5000)
      WRITE (6,3000) (NIP(I),I=1,K)
      WRITE (6,4000) (PIP(I),I=1,L)
      RETURN
 1000 FORMAT (16I5)
 2000 FORMAT (10F8.3)
 3000 FORMAT (20I5)
 4000 FORMAT (10F10.3)
 5000 FORMAT (1H0/10X,45HDATA STORED IN ARRAYS NIP AND PIP AS FOLLOWS:)
      END
```

```
      SUBROUTINE COMB (NIT,PIT,NIP,PIP,KMAX,LMAX,IMAX,JMAX,I,NCOSTS)
      COMMON/DP/D(20),GMIN(20),GMAX(20),DMIN,DMAX,MEND,JEND,T,NP,RR,R
      COMMON/WHERE/IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIP(KMAX),PIP(LMAX),NIT(IMAX),PIT(JMAX)
      MJ=MEND*JEND
C-----HOW MANY U/S PIPES?
      KK=IN(I,3)
      N=NIP(KK)
C-----SET COST OF ARRIVAL AT 1ST MANHOLE IN RUN TO ZERO
      DO 10 J=1,MJ
   10 PIT(J)=0.0
      IF (I.EQ.1) GO TO 21
C-----STORE D/S COSTS FOR LAST RUN IN NEXT SECTION OF D/S COSTS ARRAY
      K=IN(I-1,1)
      NEND=NIP(K)
      J1=(NEND-1)*MJ+1
      J2=NEND*MJ
      NCOSTS=NCOSTS+1
      K=LP1+(NCOSTS-1)*MJ-1
      K1=K+1
      DO 20 J=J1,J2
      K=K+1
   20 PIP(K)=PIT(J)
      WRITE (6,1001) K1,K
      WRITE (6,1002) (PIP(IK),IK=K1,K)
   21 IF (N.NE.0)  GO TO 25
C-----DEFINE INFEASIBLE PIPE ZONES AT UPSTREAM ENDS OF NETWORK
      IF (JEND.EQ.1) GO TO 70
      J2=JEND-1
      K=0
      DO 22 J=1,J2
      DO 22 M=1,MEND
      K=K+1
   22 PIT(K)=999999.9
      GO TO 70
C-----ONE,TWO OR THREE U/S PIPES
C-----FIND D/S TOP OF ZONE
   25 K=IN(I,1)
      NENDDS=NIP(K)
      L=IN(I,6)+2*NENDDS
      ZTDS=PIP(L)
C-----FIND D/S MAX DIAMETER
      II=I-1
      K=LN3
      DO 30 L=1,II
      LL=IN(L,1)
   30 K=K+NIP(LL)
      NDDS=NIP(K)
      DO 40 LK=1,N
C-----FIND NO OF U/S RUN
      K=KK+N-LK+1
      NR=NIP(K)
C-----FIND U/S TOP AND BOTTOM OF ZONE
      K=IN(NR,1)
      NENDUS=NIP(K)
      L=IN(NR,6)+3*NENDUS-1
      ZTUS=PIP(L)
      L=L+NENDUS
      ZBUS=PIP(L)
C-----FIND U/S MAX DIAMETER
      K=LN3
      DO 60 L=1,NR
      LL=IN(L,1)
   60 K=K+NIP(LL)
      NDUS=NIP(K+1)
C-----FIND CORRESPONDENCE BETWEEN U/S AND D/S STATES
      DO 41 M=2,MEND
      ZUS=ZTUS-FLOAT(M-1)*(ZTUS-ZBUS)/FLOAT(MEND-1)
      IF (ZUS.LT.ZTDS-0.001) GO TO 42
   41 CONTINUE
      M=MEND+1
   42 M1=M-2
      DO 43 J=1,JEND
      NUS=NDUS-JEND+J
      IF (NUS.EQ.NDDS-JEND+1) GO TO 44
   43 CONTINUE
      J=JEND
   44 J1=J-1
      K=0
      LREF=LN2+MJ*(NR-1)-1
      DO 50 J=1,JEND
      JUS=MIN0(JEND,J+J1)
      DO 50 M=1,MEND
      K=K+1
      MUS=MIN0(MEND,M+M1)
```

```
C-----DEFINE COST OF U/S STATE
      L=LP1+MJ*(NCOSTS-1)+(JUS-1)*MEND+MUS-1
      COST=PIP(L)
C-----ADD COST TO COST AT START OF D/S RUN
      PIT(K)=AMIN1(PIT(K)+COST,999999.9)
C-----IDENTIFY TRACE BACK REFERENCE ACROSS MANHOLE FROM STATE (JUS,MUS),
C-----AND ASSIGN THIS REFERENCE TO STATE (J,M)
      L=LN2+MJ*(NA-1)+(JUS-1)*MEND+MUS-1
      LREF=LREF+1
      NIP(LREF)=NIP(L)
   50 CONTINUE
   60 NCOSTS=NCOSTS-1
   70 WRITE (6,1003) I,N
      WRITE (6,1002) (PIT(K),K=1,MJ)
      RETURN
 1001 FORMAT (1H0/5X,37HD/S COSTS,LAST RUN,STORED IN PIP FROM,I6,3H TO,
     1I6)
 1002 FORMAT (10F12.3)
 1003 FORMAT (1H0//5X,17HU/S COSTS FOR RUN,I6,19H (NO. OF U/S PIPES=,I6,
     11H))
      END
```

```
      SUBROUTINE NBRUN (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,NRUN)
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX),LEV(20),
     1LL1(20),LL2(20),DIST(20),AREA(20)
      COMMON/WHERE/IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
      COMMON/MD/D(20),GMIN(20),GMAX(20),DMIN,DMAX,MEND,JEND,T,ND,PK,R7
      LOGICAL NEWN
      IN1=IN(NRUN,1)
      IN2=IN(NRUN,2)
      IN3=IN(NRUN,3)
      IN6=IN(NRUN,6)-1
      LEND=NIP(IN2)
      NEND=NIP(IN1)
C-----DEFINE D/S STATE
      N=1
   20 N=N+1
C-----DEFINE PARAMETERS DEPENDENT ON "N"
      NEWN=.TRUE.
      KXN=IN6+N
      KLAN=IN1+N
      KLAN=KLAN+NEND
      XN=PIP(KXN)
      KZTOP=IN6+2*NEND+N
      KZBOT=KZTOP+NEND
      ZTOP=PIP(KZTOP)
      ZBOT=PIP(KZBOT)
      KON=0
      IF (NRUN.EQ.1) GO TO 10
      NB=NRUN-1
      DO 5 II=1,NB
      K=IN(II,1)
    5 KON=KON+NIP(K)-1
   10 KJMG=KON+LN3+N+NRUN-2
      KON=KON+LP2+N-2
      J=0
   30 J=J+1
      JPIPE=NIP(KJMG)-JEND+J
      M=0
   40 M=M+1
      MJN=((N-1)*JEND*MEND+(J-1)*MEND+M)
      PIT(MJN) =99999.9
      IF (JPIPE.LT.1) GO TO 270
      ZDS=ZTOP-FLOAT(M-1)*(ZTOP-ZBOT)/FLOAT(MEND-1)
C-----DEFINE UPSTREAM STATE (MANHOLE)
      NN=NIP(KLAN)-1
   44 NN=NN+1
      KXNN=IN6+NN
      XNN=PIP(KXNN)
      KZT=IN6+2*NEND+NN
      ZT=PIP(KZT)
      KZB=KZT+NEND
      ZB=PIP(KZB)
      IF (.NOT.NEWN) GO TO 150
C-----DEFINE INTERMEDIATE GROUND LEVELS
   45 DO 50 L=2,LEND
      KGXL=IN6+4*NEND+LEND+L
      IF (PIP(KGXL).GT.PIP(KXN)+0.01)   GO TO 60
   50 CONTINUE
   60 L1=L
      KGXL1=KGXL
      KGZL1=KGXL1-LEND
      L2=L1-1
      A=0.0
      LEVEL=1
      IF (L1.EQ.LEND) GO TO 140
      IF (PIP(KGXL1+1).GT.XN+0.01) GO TO 140
      L3=L1+1
      DO 70 L=L3,LEND
      KGXL=IN6+4*NEND+LEND+L
      IF (PIP(KGXL).GT.XN-0.01) GO TO 80
   70 CONTINUE
   80 L2=L-1
      KGZL2=IN6+4*NEND+L2
      KGXL2=KGZL2+LEND
C-----DEFINE AREA OF LONG SECTION ABOVE STRAIGHT LINE
      A=-PIP(KGXL1-1)*(PIP(KGZL1)-PIP(KGZL2+1))*0.5
      DO 85 LL=L1,L2
      KGZLL=IN6+4*NEND+LL
      KGXLL=KGZLL+LEND
   85 A=A-PIP(KGXLL)*(PIP(KGZLL+1)-PIP(KGZLL-1))*0.5
      A=A-PIP(KGXL2+1)*(PIP(KGZL1-1)-PIP(KGZL2))*0.5
```

```
C-----IS GROUND LEVEL,CONCAVE,CONVEX OR VARIABLE
      GSLOPE=(PIP(KGZL2+1)-PIP(KGZL1-1))/(XN-XNN)
      DO 130 L=L1,L2
      KGZL=IN4+4*NEND+L
      KGXL=KGZL+LEND
      PSLOPE=(PIP(KGZL)-PIP(KGZL1-1))/(PIP(KGXL)-XNN)
      IF (PSLOPE.LT.GSLOPE-0.00001) GO TO 90
      IF (PSLOPE.LT.GSLOPE+0.00001) GO TO 130
      GO TO (110,100,130),LEVEL
   90 GO TO (120,130,100),LEVEL
  100 LEVEL=4
      GO TO 140
  110 LEVEL=3
      GO TO 130
  120 LEVEL=2
  130 CONTINUE
C-----DEFINE GROUND CONDITIONS AND DISTANCE BETWEEN MANHOLES
  140 NTH=NN-NIP(KLAN)+1
      LEV(NTH)=LEVEL
      LL1(NTH)=L1
      LL2(NTH)=L2
      DIST(NTH)=XN-XNN
      AREA(NTH)=A
      IF (ND.GT.0) WRITE (6,2002) N,NN,NTH,LEVEL,L1,L2,DIST(NTH),A
C-----DEFINE UPSTREAM STATE (U/S PIPE DIAMETER AND CROWN LEVEL)
  150 NTH=NN-NIP(KLAN)+1
      JJ=0
  160 JJ=JJ+1
      MM=0
  170 MM=MM+1
      MMJJNN=(NN-1)*JEND*MEND+(JJ-1)*MEND+MM
C-----CHECK FEASIBILITY OF SOLUTION
C-----(U/S STATE FEASIBLE?)
      IF (PIT(MMJJNN).GT.999999.0) GO TO 250
C-----(PIPE SLOPE WITHIN RESTRAINTS?)
      ZUS=ZT-FLOAT(MM-1)*(ZT-ZH)/FLOAT(MEND-1)
      SLOPE=(ZUS-ZDS)/DIST(NTH)
      IF (SLOPE.LT.GMIN(JPIPE)-0.00001) GO TO 260
      IF (SLOPE.GT.GMAX(JPIPE)+0.00001) GO TO 250
C-----(PIPE CAPACITY SUFFICIENT?)
      CALL VELOC (SLOPE,D(JPIPE),RK,VFL,QFULL)
      IF (QFULL.LT.PIP(KQN)) GO TO 250
C-----(DEPTH OF COVER RESTRAINTS VIOLATED?)
      LEVEL=LEV(NTH)
      L1=LL1(NTH)
      L2=LL2(NTH)
      GO TO (260,180,220,180),LEVEL
  180 DO 200 L=L1,L2
      KGXL=IN4+4*NEND+LEND+L
      KGZL=KGXL-LEND
      IF (ZUS-SLOPE*(PIP(KGXL)-XNN).GT.PIP(KGZL)-DMIN+0.01) GO TO 250
  200 CONTINUE
  210 IF (LEVEL.EQ.2) GO TO 260
  220 DO 230 L=L1,L2
      KGXL=IN4+4*NEND+LEND+L
      KGZL=KGXL-LEND
      IF (ZUS-SLOPE*(PIP(KGXL)-XNN).LT.PIP(KGZL)-DMAX) GO TO 260
  230 CONTINUE
C-----SOLUTION IS FEASIBLE SO COST AND COMPARE WITH PREVIOUS CHEAPEST
  240 KGZL1=IN4+4*NEND+L1
      KGZL2=KGZL1-L1+L2
      CALL COSTIT (JPIPE,AREA(NTH),PIP(KGZL1-1)-ZUS,PIP(KGZL2+1)-ZDS,
     1DIST(NTH),C)
      C=C+PIT(MMJJNN)
      IF(C.GT.PIT(MJN)-0.001) GO TO 250
      PIT(MJN)=C
      NIT(MJN)=(NN-1)*JEND*MEND+(JJ-1)*MEND+MM
C-----MOVE ON TO NEXT U/S STATE
  250 IF (MM.LT.MEND) GO TO 170
  260 KJJNG=KJNG-N+NN
      JJPIPE=NIP(KJJNG)-JEND+JJ
      IF (JJPIPE.LT.JPIPE.AND.JJ.LT.JEND) GO TO 160
      IF (NN.LT.NIP(KLHN)) GO TO 44
C-----MOVE ON TO NEXT D/S STATE
      NEWN=.FALSE.
  270 IF (M.LT.MEND) GO TO 40
      IF(J.LT.JEND) GO TO 30
      IF (N.LT.NEND) GO TO 20
      RETURN
 2002 FORMAT (10X,15HGROUND FROM M/H,I6,6HTO M/H,5I4,2(F9.3))
      END
```

```
      SUBROUTINE COSTIT (J,AREA,DUS,DDS,DIST,COST)
      DEPTH=(DUS+DDS)/2.0+AREA/DIST
      GO TO (10,20,30,40,50,60,70) ,J
   10 COST=DIST*(2.8+6.1*DEPTH)+30.0+70.0*DUS
      RETURN
   20 COST=DIST*(5.7+6.1*DEPTH)+30.0+70.0*DUS
      RETURN
   30 COST=DIST*(8.9+6.1*DEPTH)+30.0+75.0*DUS
      RETURN
   40 COST=DIST*(12.5+6.6*DEPTH)+30.0+80.0*DUS
      RETURN
   50 COST=DIST*(15.9+6.7*DEPTH)+30.0+85.0*DUS
      RETURN
   60 COST=DIST*(19.7+5.0*DEPTH)+30.0+90.0*DUS
      RETURN
   70 COST=DIST*(23.7+5.3*DEPTH)+30.0+95.0*DUS
      RETURN
      END
```

```
      SUBROUTINE TRAIL (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,NRUN,IJ)
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX)
      COMMON/RP/D(20),GMIN(20),GMAX(20),DMIN,DMAX,MEND,JEND,T,NP,RK,RP
      COMMON/LNFRF/IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
      IN1=IN(NRUN,1)
      NEND=NIP(IN1)
      K=LN2-1+MEND*JEND*(NRUN-1)
      MJNEND=(NEND-1)*JEND*MEND
      N=NEND*MEND*JEND
      DO 40 J=1,JEND
      DO 40 M=1,MEND
      K=K+1
C-----NIP(NIP(K)) IS 1ST U/S REFERENCE NUMBER IN TRACE BACK FROM D/S
C-----STATE (J,M)
      N=N+1
C-----NIT(N)=REFERENCE BACK ACROSS M/H FROM D/S STATE(J,M)
      MJNEND=MJNEND+1
C-----PIT(MJNEND)=COST OF SOLUTION TO D/S STATE(J,M)
C-----ASSUME NO CHANGE OF REFFRENCE ACROSS THE M/H
      NIT(N)=MJNEND
C-----IF COST OF ARRIVAL AT A HIGHER LEVEL IS CHEAPER ADOPT THIS COST
C-----AND ALTER REFERENCE
      IF (M.LE.1) GO TO 10
      IF (PIT(MJNEND).LT.PIT(MJNEND-1)) GO TO 10
      PIT(MJNEND)=PIT(MJNEND-1)
      NIT(N)=NIT(N-1)
C-----IF COST OF ARRIVAL WITH A SMALLER DIAMETER IS CHEAPER,ADOPT THIS
C-----COST AND ALTER REFERENCE
   10 IF (J.LE.1) GO TO 20
      I=MJNEND-MEND
      IF (PIT(MJNEND).LT.PIT(I)) GO TO 20
      PIT(MJNEND)=PIT(I)
      I=N-MEND
      NIT(N)=NIT(I)
   20 CONTINUE
C---1ST U/S REF.NO. IN TRACE BACK FROM D/S STATE(J,M) IS REF. ACROSS M/H
      IJ=IJ+1
      NIP(K)=IJ
      NIP(IJ)=NIT(N)
      IF(PIT(MJNEND).GT.999999.0) NIP(IJ)=0
C-----ESTABLISH THE REST OF THE TRACE BACK
   30 MAJANA=NIP(IJ)
      IJ=IJ+1
      IF(MAJANA.LT.MEND*JEND) GO TO 40
      NIP(IJ)=NIT(MAJANA)
      GO TO 30
   40 IJ=IJ-1
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRACE (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,L)
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX),JDS(10),MDS(10)
      COMMON/PP/ D(20),GMIN(20),GMAX(20),DMIN,DMAX,MEND,JEND,T,ND,RK,RP
      COMMON/WHERE/ IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
C-----TOTAL NUMBER OF BRANCHES AND MANHOLES
      LO=NIP(2)
      NO=NIP(1)
C-----NUMBER OF MANHOLES IN LAST RUN
      K=IN(LO,1)
      NEND=NIP(K)
C-----START AND END ELEMENTS FOR FINAL COSTS IN ARRAY PIT
      MJ=MEND*JEND
      I1=(NEND-1)*MJ+1
      I2=NEND*MJ
C------WHICH IS CHEAPEST
      ICOST=0
      COST=999999.8
      DO 10 I=I1,I2
      IF (PIT(I).GE.COST) GO TO 10
      COST=PIT(I)
      ICOST=I
   10 CONTINUE
C------NO FEASIBLE SOLUTION?
      IF (ICOST.EQ.0) STOP
C------IDENTIFY THE DOWNSTREAM STATE
      I=ICOST-I1+1
      J=(I-1)/MEND+1
      M=I-(J-1)*MEND
      NRUN=LO
      L=0
      LL=0
C-----IDENTIFY START AND END ELEMENTS IN ARRAY NIP FOR TRACE BACK
C-----UP BRANCH GIVEN RUN NUMBER, J AND M
   15 K=LN2-1+(NRUN-1)*JEND*MEND+(J-1)*MEND+M
      K=NIP(K)-1
C-----TRACE BACK ALONG BRANCH NRUN FROM (J,M)
   20 K=K+1
      L=L+1
      NIT(L)=NRUN
      NA=(NIP(K)-1)/(MEND*JEND)+1
      L=L+1
      NIT(L)=NA
      JA=(NIP(K)-1-(NA-1)*MEND*JEND)/MEND+1
      L=L+1
      NIT(L)=JA
      MA=NIP(K)-(NA-1)*MEND*JEND-(JA-1)*MEND
      L=L+1
      NIT(L)=MA
      IF(NA.GT.1) GO TO 20
C-----NOW AT UPSTREAM END OF RUN
C-----HOW MANY UPSTREAM RUNS?
      K=IN(NRUN,3)
      K=NIP(K)+1
      GO TO (70,60,50,40),K
   40 LL=LL+1
      JDS(LL)=JA
      MDS(LL)=MA
   50 LL=LL+1
      JDS(LL)=JA
      MDS(LL)=MA
   60 NRUN=NRUN-1
      J=JA
      M=MA
      GO TO 15
   70 IF (NRUN.EQ.1) GO TO 80
      NRUN=NRUN-1
      J=JDS(LL)
      M=MDS(LL)
      LL=LL-1
      GO TO 15
   80 WRITE (6,1003) COST
      WRITE (6,1001) (NIT(II),II=1,L)
      RETURN
 1001 FORMAT (10X,5HRUN,I3,5X,3HM/H,I3,5X,8HU/S DIAM,I3,5X,5HLEVEL,I3)
 1003 FORMAT (1H1///10X,31HTRACE BACK OF CHEAPEST SOLUTION,5X,5HCOST=,
     1F12.3)
      END
```

```
      SUBROUTINE RAIN (Y,T,RI)
C-----Y=RETURN PERIOD(YRS),T=TIME(MINS),RI=INTENSITY (MM/HR)
      RI=40./T*((Y*T+202.76)**.28169-2.56)
      IF (RI-33.) 10,10,40
   10 IF(RI) 20,30,30
   20 RI=0.0
   30 RETURN
   40 NI=0
C-----ITERATION LOOP FOR HOLLAND FORMULA
   50 F=ALOG(15240.0/Y/RI/T*(RI*T/1524.0+0.1)**3.55)-1.0+0.0316*RI
      DF=-1.0/RI+3.55/(RI+152.4/T)+.0316
      X=F/DF
      RI=RI-X
      IF (ABS(X)-0.01) 10,10,60
   60 IF (NI-10) 80,80,70
C-----ERROR
   70 STOP
   80 NI=NI+1
      GO TO 50
      END
```

```
      SUBROUTINE MGRAD(NIP,PIP,KMAX,LMAX)
C-----PRODUCES A MINIMUM GRADIENT 'DESIGN'
      COMMON/DP/ D(20),GMIN(20),GMAX(20),DMIN,DMAX,MEND,JEND,T,ND,RK,RP
      COMMON/NHERE/ IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIP(KMAX),PIP(LMAX),TDS(50),MD(50)
C-----SET INITIAL VALUES
      RP=1.0
      IA=LP2-1
      IB=LN3-1
      LO=NIP(2)
      IF (ND.NE.0)        WRITE (6,1001)
C-----FOR EACH BRANCH IN TURN
      DO 50 I=1,LO
C------IDENTIFY NUMBER OF MANHOLES
      J=IN(I,1)
      NN=NIP(J)
C-----IDENTIFY TIME TO UPSTREAM END OF BRANCH,AND MAX. UPSTREAM DIAM.
      J=IN(I,3)
      NBR=NIP(J)
      TUS=T
      NPIPE=1
      IF (NBR.EQ.0) GO TO 20
      DO 10 KK=1,NBR
      J=J+1
      K=NIP(J)
      IF (TDS(K).GT.TUS) TUS=TDS(K)
      IF (MD(K).GT.NPIPE) NPIPE=MD(K)
   10 CONTINUE
C-----IDENTIFY POSITIONS IN PIP FOR CHAINAGE,AREA,LEVEL FOR FIRST M/H
   20 J1=IN(I,6)
      J2=J1+NN
      J3=J2+NN
      ZUS=PIP(J3)
      DISTUS=0.0
      IB=IB+1
      NIP(IB)=NPIPE
C------FOR EACH MANHOLE POSITION DOWN THE BRANCH
      DO 45 K=2,NN
      J1=J1+1
      J2=J2+1
      J3=J3+1
      DISTDS=PIP(J1)
      ZDS=PIP(J3)
      DIST=DISTDS-DISTUS
      AREA=PIP(J2)
      SLOPE=(ZUS-ZDS)/DIST
C-----CALCULATE VELOCITY OF FLOW AND PIPE CAPACITY
   30 CALL VELOC (SLOPE,D(NPIPE),RK,VEL,CAP)
C------CALCULATE RAINFALL AND FLOW
      TIME=TUS+DIST/VEL
      CALL RAIN (RP,TIME/60.0,RI)
      FLOW=AREA*RI/3.6E6
      IF (FLOW.LT.CAP) GO TO 40
      NPIPE=NPIPE+1
      GO TO 30
C------STORE FLOW AND PIPE SIZE NUMBER
   40 IA=IA+1
      IB=IB+1
      PIP(IA)=FLOW
      NIP(IB)=NPIPE
      IF (ND.NE.0)        WRITE (6,1002) I,K,D(NPIPE),
     1SLOPE,DIST,AREA,VEL,TIME,RI,CAP,FLOW
      TUS=TIME
      ZUS=ZDS
      DISTUS=DISTDS
   45 CONTINUE
      TDS(I)=TIME
      MD(I)=NPIPE
   50 CONTINUE
      RETURN
 1001 FORMAT (1H1///,5X,91HBRANCH  D/S M/H      DIAM   SLOPE  LENGTH
     1AREA VELOCITY  TIME  RAINFALL CAPACITY    FLOW)
 1002 FORMAT (7X,I3,5X,I3,4X,F8.3,F8.4,F8.1,F8.0,F8.3,F8.1,F8.3,2F10.4)
      END


      SUBROUTINE VELOC (SLOPE,DIAM,RK,V,Q)
C-----CALCULATES FLOWS FROM COLEBROOK-WHITE FORMULA
      SQ=SQRT(SLOPE*DIAM)
      V=-8.818*SQ*ALOG10(RK/3.7/DIAM+0.6471E-6/SQ/DIAM)
      Q=V*DIAM*DIAM*0.7854
      RETURN
      END
```

```
      SUBROUTINE LEVELS (NIT,PIT,NIP,PIP,IMAX,JMAX,KMAX,LMAX,LNIT)
      COMMON/DP/D(20),GMIN(20),GMAX(20),DMIN,DMAX,MEND,JEND,T,NP,RK,RD
      COMMON/WHERE/IN(50,6),LN1,LN2,LN3,LP1,LP2,LP3
      DIMENSION NIT(IMAX),PIT(JMAX),NIP(KMAX),PIP(LMAX),TDS(50),ZD(50)
      WRITE (6,1002)
      COST=0.0
      LO=0
      LJ=LN3-1
   10 LO=LO+1
      K=IN(LO,1)
      NEND=NIP(K)
C---IDENTIFY THE UPSTREAM BRANCHES AND HENCE U/S TIME OF FLOW AND LEVEL
      K=IN(LO,3)
      NBR=NIP(K)
      TUS=T
      DISTUS=0.0
      L1=IN(LO,6)+6*NEND
      GLUS=PIP(L1)
      ZUS=GLUS-DMIN
      L=IN(LO,2)
      NG=NIP(L)
      IF (NBR.EQ.0) GO TO 60
      DO 30 I=1,NBR
      K=K+1
      L=NIP(K)
      TUS=AMAX1(TUS,TDS(L))
      ZUS=AMIN1(ZUS,ZD(L))
   30 CONTINUE
C-----DEFINE PIPE DIAMETER,D/S M/H,CATCHMENT AREA
   60 LNIT=LNIT-4
      N=NIT(LNIT-2)
      J=NIT(LNIT-1)
      LJ1=LJ+N
      JPIPE=MAX0(NIP(LJ1)-JEND+J,1)
      DIAM=D(JPIPE)
      L=IN(LO,6)-1+N
      DISTDS=PIP(L)
      DIST=DISTDS-DISTUS
      L=L+NEND
      AREA=PIP(L)
C------FIND REQUIRED GRADIENT
      CALL GRADE(JPIPE,DIST,AREA,TUS,SLOPE,Q,QMAX,VEL)
C------FIND D/S GROUND LEVEL
      L2=L1+NG
      DO 50 I=L1,L2
      K=I+NG
      IF (PIP(K).GT.DISTDS-0.1) GO TO 60
   50 CONTINUE
   60 GLDS=PIP(I)
      L2=I
      ZDS=GLDS-DMIN
C-----IS MIN SLOPE SOLUTION FEASIBLE?
      IF (ZUS-SLOPE*DIST.GT.ZDS+0.001) GO TO 65
      ZDS=ZUS-SLOPE*DIST
      GO TO 70
   65 SLOPE=(ZUS-ZDS)/DIST
      IF (SLOPE.LT.GMAX(JPIPE)+0.0001) GO TO 68
      ZUS=ZDS+GMAX(JPIPE)*DIST
      SLOPE=GMAX(JPIPE)
   68 CALL VELOC(SLOPE,DIAM,RK,VEL,QMAX)
      TIME=(TUS+DIST/VEL)/60.0
      CALL RAIN (RP,TIME,RI)
      Q=AREA*RI/3.6E6
C-----CHECK GROUND COVER EN ROUTE
   70 L3=L1+1
      L4=L2-1
      GAREA=0.0
      IF (L3.GT.L4) GO TO 90
      DZMAX=0.0
      DO 80 I=L3,L4
      K=I+NG
   80 DZMAX=AMAX1(ZUS-SLOPE*(PIP(K)-DISTUS)-PIP(I)+DMIN,DZMAX)
      ZUS=ZUS-DZMAX
      ZDS=ZDS-DZMAX
```

```
C-----CALCULATE AREA OF LONG SECTION ABOVE STRAIGHT LINE
      M1=L1+NG
      M2=L2+NG
      GAREA=-PIP(M1)*(PIP(L3)-PIP(L2))*0.5
      DO 100 I=L3,L4
      K=I+NG
  100 GAREA=GAREA-PIP(K)*(PIP(I+1)-PIP(I-1))*0.5
      GAREA=GAREA-PIP(M2)*(PIP(L1)-PIP(L4))*0.5
   90 CALL COSTIT (JPIPE,GAREA,GLUS-ZUS,GLDS-ZDS,DIST,C)
      COST=COST+C
      WRITE (6,1001) LO,DIST,DIAM,SLOPE,ZUS,ZDS,GLUS,GLDS,AREA,GAREA,Q,
     1QMAX,VEL,C,COST
      TUS=TUS+DIST/VEL
      ZUS=ZDS
      GLUS=GLDS
      DISTUS=DISTDS
      L1=L2
      IF(N.LT.NEND) GO TO 40
      LNIT=LNIT-6
      LJ=LJ1
      TDS(LO)=TUS
      ZD(LO)=ZUS
      IF (LO.LT.NIP(2)) GO TO 10
      RETURN
 1001 FORMAT (1H0,I6,F9.3,F7.1,F7.4,4F8.3,F8.0,F8.3,2F7.4,F8.1,2F10.1)
 1002 FORMAT (1H1//3X,116HBRANCH LENGTH  DIAM  SLOPE U/S SL  D/S SL  U/S
     1 GL  D/S GL  AREA  GROUND A.  FLOW  CAPACITY  VEL.     COST      S
     2UM)
      END
```

```
      SUBROUTINE GRADE (JPIPE,DIST,AREA,TUS,SLOPE,Q,QFULL,V)
C-----CALCULATES REQUIRED SLOPE OF A PIPE ACCORDING TO RATIONAL METHOD
      COMMON/DP/D(20),GMIN(20),GMAX(20),DMIN,DMAX,NEND,JEND,T,ND,RK,RP
      LOGICAL NPLUS,MINUS
      K=0
      NPLUS=.FALSE.
      MINUS=.FALSE.
      SLOPE=GMIN(JPIPE)
      DIAM=D(JPIPE)
    5 CALL VELOC(SLOPE,DIAM,RK,V,QFULL)
      TIME=TUS+DIST/V
      CALL RAIN(RP,TIME/60.0,RI)
      Q=AREA*RI/3.6E6
      IF (Q.LT.QFULL.AND.K.EQ.0) RETURN
      SLOPE=0.02065*Q*Q/DIAM**5*(ALOG10(RK/3.7/DIAM+6.1365/(Q/DIAM/
     11.141E-6)**0.89))**(-2)
      K=K+1
      IF (K.LE.2) GO TO 5
   10 K=K+1
      CALL VELOC(SLOPE,DIAM,RK,V,QFULL)
      TIME=TUS+DIST/V
      CALL RAIN (RP,TIME/60.0,RI)
      Q=AREA*RI/3.6E6
      IF (ABS((Q-QFULL)/QFULL).LT.0.001) RETURN
      IF (Q-QFULL) 30,20,20
   20 IF (MINUS) RETURN
      SLOPE=SLOPE*1.001
      NPLUS=.TRUE.
      MINUS=.FALSE.
      GO TO 10
   30 IF (NPLUS) RETURN
      SLOPE=SLOPE*.999
      MINUS=.TRUE.
      NPLUS=.FALSE.
      GO TO 10
      END
```

APPENDIX F

PROGRAM SORT

```
C-----PROGRAM READS DATA FOR MODEX IN SIMPLIFIED FORM AND OUTPUTS TO
C        MAGNETIC TAPE FILE
      DIMENSION D(20),GC(300),GX(300),DIST(300),KA(300),KI(300),K(300),
     1F(300),GROUND(300),NBR(10),AREA(300),NUMMH(300),MK(300),KRR(99),
     2AMSX(300),CFF(300)
      REWIND 7
      E=0.000001
C-----READ IN PIPE DATA, STARTING WITH NUMBER OF PIPE TYPES (0=DEFAULT)
      READ (5,1002) NPTYPE
      IF (NPTYPE.GT.0) GO TO 520
C-----(DEFINE DEFAULT VALUES)
      DMIN=1.0
      RK=0.00015
      VELMIN=0.7
      VELMAX=6.0
      SMIN=0.004
      DELTA=0.075
      DSTART=0.075
      DO 510 I=1,15
  510 D(I)=DSTART+FLOAT(I)*DELTA
      NPTYPE=1
      WRITE (6,2001) DMIN,RK,VELMIN,VELMAX,SMIN,(D(I),I=1,15)
      WRITE (7)      DMIN,RK,VELMIN,VELMAX,SMIN,(D(I),I=1,15)
      GO TO 560
  520 IF (NPTYPE.GT.9)  STOP
      DO 550 I=1,NPTYPE
      READ (5,1001) DMIN,RK,VELMIN,VELMAX,SMIN
      RK=RK/1000.0
      DO 530 J=1,15
  530 D(J)=0.0
      J=0
  540 J=J+1
      READ (5,1001)  D(J)
      D(J)=D(J)/1000.0
      IF (D(J).GT.E.AND.J.LT.15) GO TO 540
      WRITE (6,2001) DMIN,RK,VELMIN,VELMAX,SMIN,(D(II),II=1,15)
      WRITE (7)      DMIN,RK,VELMIN,VELMAX,SMIN,(D(II),II=1,15)
  550 CONTINUE
      IF (NPTYPE.EQ.9) GO TO 590
  560 DO 570 I=1,20
  570 D(I)=0.0
      NPTYPE=NPTYPE+1
      DO 580 I=NPTYPE,9
      WRITE (6,2001) (D(II),II=1,20)
  580 WRITE (7)      (D(II),II=1,20)
  590 CONTINUE
C-----READ IN TIME OF ENTRY AND STORM RETURN PERIOD
      READ (4,1001) T,RP
      IF (T.LE.E) T=2.0
      IF (RP.LE.E) RP=1.0
      T=60.0*T
C-----READ IN MINIMUM AND MAXIMUM MANHOLE SPACING
      READ (4,1001) SPMIN,SPMAX
      IF (SPMIN.LE.E) SPMIN=30.0
      IF (SPMAX.LE.E) SPMAX=150.0
      WRITE (6,2001) T,RP,SPMIN,SPMAX
      WRITE (7)      T,RP,SPMIN,SPMAX
C-----READ IN OPTIMISATION PERFORMANCE PARAMETERS
      READ (4,1001) DZ,RESMN
      READ (4,1002) IEND,JEND,KXD
      IF (DZ.LE.E) DZ=0.5
      IF (RESMN.LE.E) RESMN=50.0
      IF (KXD.LE.0) KXD=4
      IF (IEND.LE.0) IEND=6
      IF (JEND.LE.0) JEND=2
C-----READ IN DIAGNOSTICS LEVEL
      READ (4,1002) ND
C-----READ IN NUMBER OF BRANCHES IN BASIC LAYOUT PROBLEM
      READ (5,1002) LO
      WRITE (6,2002) IEND,JEND,ND,LO
      WRITE (7)      IEND,JEND,ND,LO
      WRITE (6,2001) DZ
      WRITE (7)      DZ
```

```
C-----READ IN DATA FOR EACH BRANCH
      DO 290 I=1,LB
      READ (5,1002) ITYPE
      READ (5,1001) Y,DH,OFFSET,STEM,DIR
C-----READ IN GROUND LEVEL DATA
      J=0
   40 J=J+1
      READ (5,1001) GL(J)
      READ (5,1001) GX(J)
      IF(GX(J).LT.Y-0.001)      GO TO 40
C-----GENERATE POSITIONS OF MANHOLES
      IF (ITYPE.LT.10.OR.ITYPE.GT.100)  GO TO 70
   60 DIST(1)=0.0
      DIST(2)=Y
      NN=2
      GO TO 90
C-----IS LENGTH LESS THAN TWICE MINIMUM SPACING?
   70 IF (Y.LT.2.0*SPMIN-0.1.OR.Y.LT.RESMH+0.1) GO TO 60
      DIST(1)=0.0
      DIS=AMAX1(SPMIN,RESMH)
      NN=3+INT((Y-DIS-SPMIN+0.1)/RESMH)
      DIST(NN)=Y
      DIST(NN-1)=Y-DIS
      IF (NN.EQ.3) GO TO 81
      DO 80 K=4,NN
      L=NN-K+2
      DIST(L)=DIST(L+1)-RESMH
   80 CONTINUE
C-----GENERATE PERMISSIBLE MANHOLE CONNECTIONS(****REDUNDANT****)
   81 KA(1)=1
      KB(1)=1
      DO 340 M=2,NN
      DO 300 L=1,M
      IF (DIST(M)-DIST(L).GT.SPMAX+0.1) GO TO 300
      KA(M)=L
      GO TO 310
  300 CONTINUE
  310 DO 320 L=1,M
      IF (DIST(M)-DIST(L).LT.SPMIN-0.1) GO TO 330
  320 CONTINUE
  330 KB(M)=L-1
  340 CONTINUE
C-----CALCULATE GROUND LEVELS
   90 GROUND(1)=GL(1)
      IF(NN.LT.3) GO TO 130
      KK=NN-1
      DO 120 K=2,KK
      DO 100 M=1,J
      IF (GX(M).GT.DIST(K)+0.001) GO TO 110
  100 CONTINUE
  110 GROUND(K)=GL(M-1)+(GL(M)-GL(M-1))/(GX(M)-GX(M-1))*
     1(DIST(K)-GX(M-1))
  120 CONTINUE
  130 GROUND(NN)=GL(J)
      KM=1
      KN=1
      KL=0
  140 KL=KL+1
      IF (DIST(KM)-GX(KN).GT.0.001) GO TO 145
      X(KL)=DIST(KM)
      Z(KL)=GROUND(KM)
      IF (KM.EQ.NN) GO TO 146
      IF (DIST(KM)-GX(KN).GT.-0.001) KN=KN+1
      KM=KM+1
      GO TO 140
  145 X(KL)=GX(KN)
      Z(KL)=GL(KN)
      KN=KN+1
      GO TO 140
C-----READ IN UPSTREAM CONNECTIONS
  146 NUB=0
  170 NUB=NUB+1
      READ (5,1002) NBR(NUB)
      IF (NBR(NUB).EQ.0) GO TO 180
      GO TO 170
  180 NUB=NUB-1
C-----DEFINE INCREMENTS OF AREA ALONG BRANCH
      DO 190 JJ=2,NN
  190 AREA(JJ)=DH*(DIST(JJ)-DIST(JJ-1))
      AREA(1)=0.0
C-----DEFINE OFFSETS
      DO 200 JJ=1,NN
  200 OFF(JJ)=OFFSET
```

```
C-----DEFINE ABSOLUTE CHAINAGE
      DO 210 JJ=1,KL
  210 ABSX(JJ)=STCH+DIR*DIST(JJ)
      WRITE (6,2002) ITYPE,NUB,NN,KL
      WRITE (7)      ITYPE,HUD,NN,KL
      NUMMN(I)=NN
      WRITE (6,2001) (OFF(K),K=1,NN)
      WRITE (7)      (OFF(K),K=1,NN)
      WRITE (6,2001) (ABSX(K),K=1,NN)
      WRITE (7)      (ABSX(K),K=1,NN)
      WRITE (6,2001) (DIST(K),K=1,NN)
      WRITE (7)      (DIST(K),K=1,NN)
      WRITE (6,2001) (AREA(K),K=1,NN)
      WRITE (7)      (AREA(K),K=1,NN)
      IF (NUB.EQ.0) GO TO 625
      WRITE (6,2002) (NBR(K),K=1,NUB)
      WRITE (7)      (NBR(K),K=1,NUB)
  625 CONTINUE
      WRITE (6,2001) (Z(K),K=1,KL)
      WRITE (7)      (Z(K),K=1,KL)
      WRITE (6,2001) (X(K),K=1,KL)
      WRITE (7)      (X(K),K=1,KL)
  290 CONTINUE
C-----READ IN NUMBER OF CROSS DRAIN SETS
      READ (5,1002) NXDSET
      WRITE (6,2002) NXDSET
      WRITE (7)      NXDSET
      IF (NXDSET.EQ.0) GO TO 620
C---- FOR EACH SET
      DO 410 I=1,NXDSET
C-----READ IN PARALLEL BRANCHES
      J=0
  400 J=J+1
      READ (5,1002) KBR(J)
      IF (KBR(J).NE.0) GO TO 400
      J=J-1
C-----WRITE OUT NUMBER OF BRANCHES IN THIS SET
      WRITE (6,2002) J
      WRITE (7)      J
C-----FOR EACH BRANCH IN THIS SET
      DO 410 K=1,J
      KBRA=KBR(K)
      M=NUMMN(KBRA)
      LA=MAX0((M-1)/KXD,1)
      L=LA
      MM(L)=M
      IF (L.EQ.1) GO TO 406
  405 L=L-1
      M=M-KXD
      MM(L)=M
      IF (L.GT.1) GO TO 405
  406 IF (K.NE.1) GO TO 407
      WRITE (6,2002) LA
      WRITE (7)      LA
  407 WRITE (6,2002) KBRA
      WRITE (7)      KBRA
      WRITE (6,2002) (MM(N),N=1,LA)
      WRITE (7)      (MM(N),K=1,LA)
  410 CONTINUE
  620 STOP
 1001 FORMAT (F0.0)
 1002 FORMAT (I0)
 2001 FORMAT (10F12.3)
 2002 FORMAT (10I12)
      END
      FINISH
```

APPENDIX G

PROGRAM MODEX

```
CURRENT EDITION:20.03.79
      DIMENSION KE(8500),GA(12500)
      COMMON/DATA/NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
C-----SPECIFY MAXIMUM ARRAY SIZES
      NKE=8500
      NGA=12500
C-----SET INITIAL VALUES
      DO 1 I=1,NKE
    1 KE(I)=0
      DO 2 I=1,NGA
    2 GA(I)=0.0
      DO 3 I=1,50
      DO 3 J=1,11
    3 ID(I,J)=0
      DO 4 I=1,10
      DO 4 J=1,12
    4 IDX(I,J)=0
      NITEM=0
      KITEM=0
      NSET=0
      NB=0
C-----READ IN DATA FROM MAGNETIC TAPE FILE
      CALL DATAMT (KE,GA,NKE,NGA)
C-----BRANCHES HAVE BEEN SEQUENTIALLY ORDERED:DESIGN EACH IN TURN,
C      EXCEPT FOR COMPONENTS OF X-DRAIN SETS WHICH ARE DESIGNED
C      BY CALLING SUBROUTINE XDSET
      NB=0
   10 NB=NB+1
C-----IS THE BRANCH THE START OF A CROSS DRAIN SET?
      IF (ID(NB,1).GT.100) GO TO 40
C-----DESIGN THIS BRANCH TO A MINIMUM GRADIENT
      IF (ND.LT.4) GO TO 15
      WRITE (6,1001) ((ID(I,J),J=1,11),I=1,50)
      WRITE (6,1002) (GA(I),I=1,NKE)
      WRITE (6,1001) IDA,IDL,IDT,IDK,IDG,IDD,IDJ,IDP,IDR
   15 CALL MGRAD (KE,GA,NKE,NGA)
C-----COMBINE UPSTREAM COSTS
      NN=(IDG+2*ID(NB,3))
      CALL COMB (KE,GA,NKE,NGA,NN,GA(IDG),KE(IDK))
C-----PRODUCE OPTIMAL DESIGNS FOR THIS BRANCH FOR RANGE OF D/S STATES
      CALL NBRUN (KE,GA,NKE,NGA)
C-----PRODUCE TRACE BACK UP BRANCH FOR RANGE OF D/S STATES
      CALL TRAIL (KE,GA,NKE,NGA)
   20 IF (NB.LT.NBRAN) GO TO 10
   30 CALL TRACE (KE,GA,NKE,NGA,NGG,NKK)
      IF (NXDSET.GT.0)          CALL OUTPUT (KE,GA,NKE,NGA)
      IF (NXDSET.EQ.0)          CALL LEVELS (KE,GA,NKE,NGA,NGG,NKK)
      STOP
C-----BRANCH IS START OF A X-DRAIN SET: DESIGN THIS SET
   40 NSET=NSET+1
      CALL XDSET (KE,GA,NKE,NGA)
      IF (ND.LT.4) GO TO 50
      WRITE (6,1001) NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM
      WRITE (6,1001) MEND,JEND,MJ,ND
      WRITE (6,1002) ((PDA(I,J),J=1,20),I=1,9)
      WRITE (6,1002) T,RP,ZD,DMAX,DMIN,DELTA
      WRITE (6,1001) IDA,IDL,IDT,IDK,IDG,IDD,IDJ,IDP,IDR,IDM,IDB
      WRITE (6,1001) ((ID(I,J),J=1,11),I=1,NBRAN)
      WRITE (6,1001) ((IDX(I,J),J=1,12),I=1,NXDSET)
      WRITE (6,1001) (KE(I),I=1,NKE)
      WRITE (6,1002) (GA(I),I=1,NGA)
C-----UPDATE THE CURRENT BRANCH NUMBER
   50 NB=NB+IDX(NSET,6)-1
      GO TO 20
 1001 FORMAT (20I6)
 1002 FORMAT (10F12.3)
      END

END OF SEGMENT, LENGTH   477, NAME  MODEX
```

```
      SUBROUTINE XDSET (KE,GA,NKE,NGA)
C-----THIS SUBROUTINE PRODUCES A SET OF OPTIMAL DESIGNS FOR A RANGE OF
C      STATE VARIABLES AT D/S END OF A X-DRAIN SET
      DIMENSION KE(NKE),GA(NGA)
      COMMON/DATA/NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      IF (ND.GT.0) WRITE (6,2000)  NSET
C-----IDENTIFY NUMBER OF X-DRAIN POSITIONS IN CURRENT X-DRAIN SET
      NXEND=IDX(NSET,7)
C-----DEFINE LAST ELEMENTS OF ARRAYS KE AND GA REQUIRED FOR STORING
C      TRACE AND COST DATA AT X-DRAIN POSITIONS
      LASTKE=IDK+NXEND*MJ-1
      LASTGA=IDG+NXEND*(MJ+4)-1
      IF (LASTKE.GT.NKE-MJ)  CALL MESAGE (6,0)
      IF (LASTGA.GE.IDP)      CALL MESAGE (7,0)
C      SET COSTS ARTIFICIALLY HIGH
      DO 1  I=IDG,LASTGA
    1 GA(I)=999999.9
C-----DEFINE START OF RUN PARAMETERS AND COSTS FOR EACH MEMBER OF SET
      K=NB
      J=IDX(NSET,6)
C-----(FOR EACH BRANCH IN THE SET)
      DO 10 I=1,J
C-----(IDENTIFY BRANCH NUMBER AND PIPE TYPE)
      NB=IDX(NSET,I)
      NPIPE=MOD(ID(NB,1),10)
      IF (I.EQ.1) NPIPE1=NPIPE
C-----(DEFINE START OF RUN VALUES)
      L=ID(NB,10)
      CALL UPVAL (KE,GA,NKE,NGA,TUS,AREAUS,ZUS,DUS,GA(L))
C-----(STORE START OF RUN VALUES)
      L=IDR+(I-1)*(MJ+4)
      GA(L)=TUS
      GA(L+1)=AREAUS
      GA(L+2)=ZUS
      GA(L+3)=DUS
C-----COMBINE AND STORE UPSTREAM COSTS
      CALL SIZED (PDA,NPIPE,JI,DUS)
      CALL COMB (KE,GA,NKE,NGA,L+4,ZUS,JI)
   10 CONTINUE
      NB=K
C-----CONSIDER EACH X-DRAIN POSITION IN TURN STARTING AT U/S END
      NX=0
   20 NX=NX+1
C-----CONSIDER FIVE NEAREST U/S X-DRAINS,(CAN START WITH NO U/S X-DRAIN)
      NUX=MAX0(NX-6,-1)
      NCROSS=0
   40 NUX=NUX+1
      NCROSS=NCROSS+1
C-----IDENTIFY UPSTREAM VALUES AND COSTS FOR CURRENT U/S X/D
      IF (NUX.EQ.0) GO TO 50
      K=IDG+(NUX-1)*(MJ+4)-1
      L=IDR-1
      LL=MJ+4
      DO 45 KK=1,LL
      K=K+1
      L=L+1
   45 GA(L)=GA(K)
C-----WRITE CONTENTS OF KE,GA,COMMONS "WHERE" AND "DATA" TO MAG TAPE
   50 CALL WRITMT (KE,GA,LASTKE,LASTGA)
C-----SET UP SUBNETWORK FOR OPTIMISING
      CALL SETUP (KE,GA,NKE,NGA,NX,NUX)
C-----OPTIMISE SUBNETWORK FOR RANGE OF STATE VARIABLES AT D/S END
      CALL SUBNET (KE,GA,NKE,NGA,NUX)
C-----RESTORE CONTENTS OF KE,GA,COMMONS "WHERE" AND "DATA"
      CALL READMT (KE,GA,LASTKE,LASTGA)
C-----STORE VALUES OF TIME,AREA,LEVEL AND DIAMETER AT X/D POINT NX
      K=IDG+(NX-1)*(MJ+4)
      IF (NCROSS.EQ.1)  GA(K+1)=GA(IDR+1)
      IF (NCROSS.EQ.1)  GA(K)=0
      GA(K)=(GA(K)*FLOAT(NCROSS-1)+GA(IDR))/FLOAT(NCROSS)
      IF (NCROSS.GT.1) GO TO 54
      ZOLD=-999999.9
      DOLD=0.0
      GO TO 55
   54 ZOLD=GA(K+2)
      DOLD=GA(K+3)
   55 ZLAST=GA(IDR+2)
      DLAST=GA(IDR+3)
      ZNEW=AMAX1(ZOLD,ZLAST)
      DNEW=AMAX1(DOLD,DLAST)
      GA(K+2)=ZNEW
      GA(K+3)=DNEW
      IF (NCROSS.GT.1)  GO TO 57
      MOLD=MEND
      GO TO 58
   57 MOLD=IFIX((ZNEW-ZOLD+DELTA-0.01)/DELTA)
   58 MLAST=IFIX((ZNEW-ZLAST+DELTA-0.01)/DELTA)
      CALL SIZED (PDA,NPIPE1,NOLD,DOLD)
      CALL SIZED (PDA,NPIPE1,NLAST,DLAST)
      NNEW=MAX0(NOLD,NLAST)
      JOLD=NNEW-NOLD
      JLAST=NNEW-NLAST
```

```
C-----ALTER OLD AND LAST COSTS TO NEW REFERENCE GRID AND SELECT CHEAPEST
      DO 60 J=1,JEND
      J1=MIN0(J+JOLD,JEND)
      J2=MIN0(J+JLAST,JEND)
      DO 60 I=1,MEND
      M=MEND+1-I
      IK=IDK+(NX-1)*MJ+(J-1)*MEND+M-1
      IG=IDG+(NX-1)*(MJ+4)+(J-1)*MEND+M+3
      COLD=999999.9
      M1=M-MOLD
      M2=M-MLAST
      IF (M1.LT.1) GO TO 601
      IGOLD=IDG+(NX-1)*(MJ+4)+(J1-1)*MEND+M1+3
      IKOLD=IDK+(NX-1)*MJ+(J1-1)*MEND+M1-1
      COLD=GA(IGOLD)
      KOLD=KE(IKOLD)
  601 CLAST=999999.9
      IF (M2.LT.1) GO TO 602
      IGLAST=IDR+(J2-1)*MEND+M2+3
      IKLAST=NKE-MJ+(J2-1)*MEND+M2
      CLAST=GA(IGLAST)
      KLAST=KE(IKLAST)
  602 KE(IK)=0
      GA(IG)=999999.9
      IF (CLAST.LT.COLD) GO TO 603
      IF (COLD.GT.999999.0) GO TO 60
      KE(IK)=KOLD
      GA(IG)=COLD
      GO TO 60
  603 KE(IK)=KLAST
      GA(IG)=CLAST
   60 CONTINUE
      IK1=IDK+(NX-1)*MJ
      IG1=IDG+(NX-1)*(MJ+4)+4
      IK2=IK1+MJ-1
      IG2=IG1+MJ-1
      K1=K+3
      IF (ND.LT.1) GO TO 65
      WRITE (6,2001) NX,IG1,IG2
      WRITE (6,2002) (GA(I),I=IG1,IG2)
      WRITE (6,2003) (GA(I),I=K,K1)
      WRITE (6,2004) IK1,IK2
      WRITE (6,2005) (KE(I),I=IK1,IK2)
C-----IS THERE ANOTHER POSSIBLE POSITION FOR U/S X-DRAIN?
   65 IF (NUX.LT.NX-1) GO TO 40
C-----IS THERE ANOTHER POSSIBLE POSITION FOR X-DRAIN?
      IF (NX.LT.NXEND) GO TO 20
C-----STORE FINAL VALUES OF TIME,AREA,LEVEL,DIAMETER AND COSTS
      J=IDJ+(NB-1)*MJ-1
      K=IDG+(NXEND-1)*(MJ+4)+3
      L=IDT+NB-1
      GA(L)=GA(K-3)
      L=IDA+NB-1
      GA(L)=GA(K-2)
      L=IDL+NB-1
      GA(L)=GA(K-1)
      L=IDD+NB-1
      GA(L)=GA(K)
      DO 70 I=1,MJ
      J=J+1
      K=K+1
   70 GA(J)=GA(K)
      IF (ND.LT.1) GO TO 75
      WRITE (6,2006)
      WRITE (6,2004) IDK,IK2
      WRITE (6,2005) (KE(I),I=IDK,IK2)
C-----STORE TRACE DATA FOR X/D SET IN MAG. TAPE FILE.
   75 REWIND 8
      IF (KITEM.EQ.0) GO TO 90
      DO 80 I=1,KITEM
   80 READ (8)
   90 DO 100 I=IDK,IK2
  100 WRITE (8) KE(I)
      WRITE (8) GA(IDP+2),GA(IDP+3)
      KITEM=KITEM+IK2-IDK+2
      RETURN
 2000 FORMAT (1H0/////5X,13HENTERED XDSET/5X,22HCROSSDRAIN SET NUMBER=,
     1I6)
 2001 FORMAT (1H0/5X,36HOPTIMAL SOLUTION DOWN TO CROSS DRAIN,I6/10X,
     129HCOSTS STORED IN ARRAY GA FROM,I6,4H TO ,I6)
 2002 FORMAT (1X,F11.3,9F12.3)
 2003 FORMAT (5X,5HTIME=,F12.3,6H AREA=,F12.3,11H MAX.LEVEL=,F12.3,
     110H MAX.DIAM=,F12.3)
 2004 FORMAT (5X,41HREFERENCE NUMBERS STORED IN ARRAY KE FROM,I6,
     14H TO ,I6)
 2005 FORMAT (1X,I5,19I6)
 2006 FORMAT (1H0/5X,27HDESIGN OF X/D SET COMPLETED)
      END
```

END OF SEGMENT, LENGTH 999, NAME XDSET

```
      SUBROUTINE WRITMT(KE,GA,NKE,NGA)
      DIMENSION KE(NKE),GA(NGA)
      COMMON/DATA/N(5),NITEM,KITEM,NK
      COMMON/WHERE/J(2331)
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      IF(ND.GT.0)  WRITE (6,2000)
      REWIND 8
      IF (KITEM.EQ.0) GO TO 5
      DO 3 I=1,KITEM
    3 READ  (8)
    5 WRITE (8)   (KE(I),I=1,NKE)
      WRITE (8)   (GA(I),I=1,NGA)
      WRITE (8)   (N(I),I=1,5)
      WRITE (8)   NITEM,KITEM,NK
      WRITE (8)   (J(I),I=1,2331)
      IF (NITEM.EQ.0) RETURN
      REWIND 9
      DO 10 I=1,NITEM
      READ  (9) K,R,S
   10 WRITE (8) K,R,S
 2000 FORMAT (5X,14HENTERED WRITMT)
      RETURN
      END

END OF SEGMENT, LENGTH   170, NAME  WRITMT
```

```
      SUBROUTINE READMT (KE,GA,NKE,NGA)
      DIMENSION KE(NKE),GA(NGA)
      COMMON/DATA/N(5),NITEM,KITEM,NK
      COMMON/WHERE/J(2331)
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      IF (ND.GT.0) WRITE (6,2000)
      REWIND 8
      IF (KITEM.EQ.0) GO TO 5
      DO 3 I=1,KITEM
    3 READ (8)
    5 READ (8) (KE(I),I=1,NKE)
      READ (8) (GA(I),I=1,NGA)
      READ (8) (N(I),I=1,5)
      READ (8) NITEM,KITEM,NK
      READ (8) (J(I),I=1,2331)
      IF (NITEM.EQ.0) RETURN
      REWIND 9
      DO 10 I=1,NITEM
      READ  (8) K,R,S
   10 WRITE (9) K,R,S
 2000 FORMAT (5X,14HENTERED READMT)
      RETURN
      END

END OF SEGMENT, LENGTH   170, NAME  READMT
```

```
      SUBROUTINE DATANT (KF,GA,NKE,NGA)
C-----THIS ROUTINE READS THE DESIGN PARAMETERS AND BASIC DATA FROM A
C-----MAG TAPE FILE AND DEFINES THE CONTENTS OF COMMON 'WHERE'
      DIMENSION KE(NKE),GA(NGA)
      COMMON/DATA/NSET,NBRAN,NB,NPIPE,NKOSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IOX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      NG=0
      NK=0
C-----READ IN PIPE TABLE DATA
      DO 1 I=1,9
    1 READ (7) (PDA(I,J),J=1,20)
C-----READ IN TIME OF ENTRY(SECS),RETURN PERIOD(YEARS),MIN AND MAX SPACING
      READ (7) T,RP,DMIN,DMAX
C-----READ IN STATE VARIABLE PARAMETERS,DIAG LEVEL AND NO. OF BRANCHES
      READ (7) MEND,JEND,ND,NBRAN
      IF (ND.GT.0) WRITE(6,2000)
      IF (NBRAN.GT.200)  CALL MESAGE(19,NBRAN)
      MJ=MEND*JEND
C-----READ IN DEPTH OF ZONE
      READ (7) ZD
C-----DEFINE VALUE OF DEPTH INCREMENT DELTA
      DELTA=999999.9
      IF (MEND.GT.1) DELTA=ZD/FLOAT(MEND-1)
C-----FOR EACH BRANCH:
      DO 50 NB=1,NBRAN
C-----READ IN PIPE TYPE,NUMBER OF U/S BRANCHES,NUMBER OF POSSIBLE M/HS,
C      NUMBER OF GROUND LEVELS
      READ (7) (ID(NB,I),I=1,4)
C-----READ IN OFFSETS AND ABSOLUTE CHAINAGES
      NN=ID(NB,3)
      IF (NGA.LT.NG+4*NN+2*ID(NB,4))  CALL MESAGE (3,0)
      ID(NB,6)=NG+1
      NG1=ID(NB,6)
      NG2=NG1+NN-1
      READ (7) (GA(N),N=NG1,NG2)
      ID(NB,7)=NG+1+NN
      NG1=ID(NB,7)
      NG2=NG1+NN-1
      READ (7) (GA(N),N=NG1,NG2)
      NG=NG2
C-----READ IN CHAINAGES AND INCREMENTS OF AREA
      ID(NB,8)=NG+1
      NG1=ID(NB,8)
      NG2=NG1+NN-1
      READ (7) (GA(N),N=NG1,NG2)
      ID(NB,9)=NG+1+NN
      NG1=ID(NB,9)
      NG2=NG1+NN-1
      READ (7) (GA(N),N=NG1,NG2)
      NG=NG2
C-----READ IN U/S BRANCH NUMBERS
      IF (ID(NB,2).EQ.0) GO TO 30
      ID(NB,5)=NK+1
      NK1=ID(NB,5)
      NK2=NK1+ID(NB,2)-1
      IF (NKE.LT.NK2)       CALL MESAGE (4,0)
      READ (7) (KE(N),N=NK1,NK2)
      NK=NK2
C-----READ IN GROUND LEVEL DATA
   30 ID(NB,10)=NG+1
      NG1=ID(NB,10)
      NG2=NG1+ID(NB,4)-1
      READ (7) (GA(N),N=NG1,NG2)
      ID(NB,11)=NG+1+ID(NB,4)
      NG1=ID(NB,11)
      NG2=NG1+ID(NB,4)-1
      READ (7) (GA(N),N=NG1,NG2)
      NG=NG2
   50 CONTINUE
```

```
C-----READ IN NUMBER OF X-DRAIN SETS
      READ (7) NXDSET
      IF (NXDSET)  90,90,60
C-----FOR EACH SET:
   60 DO 80 N=1,NXDSET
C-----READ IN NUMBER OF BRANCHES IN THIS SET
      READ (7) IDX(N,6)
      NUMB=IDX(N,6)
C-----READ IN NUMBER OF X-DRAINS
      READ (7) IDX(N,7)
      NUMXD=IDX(N,7)
      IF (NKE.LT.NK+NUMXD*NUMB)   CALL MESAGE (4,0)
C-----FOR EACH BRANCH:
      DO 70 M=1,NUMB
C-----READ IN BRANCH NUMBER
      READ (7) IDX(N,M)
      IDX(N,M+7)=NK+1
C-----READ IN THE NUMBERS OF THE M/HS WHICH HAVE X-DRAIN CONNECTIONS
      NK1=NK+1
      NK2=NK1+NUMXD-1
      READ (7) (KE(NI),NI=NK1,NK2)
      NK=NK2
   70 CONTINUE
   80 CONTINUE
   90 IDK=NK+1
      IDT=NG+1
      IDA=IDT+NBRAN
      IDL=IDA+NBRAN
      IDD=IDL+NBRAN
      IDJ=IDD+NBRAN
      IDG=IDJ+MJ*NBRAN
      IDR=NGA-MJ-3
      IDP=IDR-5*MJ-20
      IF (IDG.GT.IDP)     CALL MESAGE (5,0)
      WRITE (6,1001)
      DO 100 I=1,9
  100 WRITE (6,1002) I,(PDA(I,J),J=1,5)
      WRITE (6,1003)
      DO 110 I=1,9
  110 WRITE (6,1005) I,(PDA(I,J),J=6,20)
      TIME=T/60
      WRITE (6,1004) TIME,RP,DMIN,DMAX,MEND,JEND,ZD
      RETURN
 2000 FORMAT (5X,14HENTERED DATAMT)
 1001 FORMAT (50X,12HPIPE LIBRARY/50X,12(1H*)/10X,4HTYPE,15X,5HCOVER,
     111X,9HROUGHNESS,12X,8HMIN.VEL.,12X,8HMAX.VEL.,11X,9HMIN.GRAD./
     230X,3H(M),15X,4H(M) ,15X,5H(M/S),15X,5H(M/S))
 1002 FORMAT (8X,I6,F20.3,F20.6,2F20.3,F20.4)
 1003 FORMAT (1H0,10X,4HTYPE,50X,9HDIAMETERS)
 1004 FORMAT (1H0/10X,19HTIME OF ENTRY(MINS),F11.3,5X,
     118HRETURN PERIOD(YRS),F12.3/10X,22HMIN.MANHOLE SPACING(M),F8.3,5X,
     222HMAX.MANHOLE SPACING(M),F8.3/10X,I6,15H VERTICAL ZONES,10X,I6,
     311H PIPE ZONES,    14HZONE DEPTH(M) ,F6.3/1H1)

 1005 FORMAT (8X,I6,15F7.3)
      END
```

```
      SUBROUTINE SETUP (KE,GA,NKE,NGA,NX,NUX)
      DIMENSION KE(NKE),GA(NGA),CHAIN(100),AREA(100),XGL(100),ZGL(100),
     1DSOFF(5),DSGL(5),IDN(11)
      COMMON/DATA/NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      IF (ND.GT.0)   WRITE (6,2000)
      IF (ND.GT.-1)   WRITE (6,1001)  NUX,NX
      NITEM=0
      REWIND 9
C-----DEFINE NO. OF PARALLEL LINES IN THIS X/D SET
      NPARA=IDX(NSET,6)
      IGA=0
      IKE=0
C-----FOR EACH DRAINAGE LINE
      DO 120 I=1,NPARA
C-----IDENTIFY OLD NO. OF DRAINAGE LINE
      NUM=IDX(NSET,I)
C-----FIND OLD NUMBERS OF 1ST AND LAST MANHOLES
   20 IF (NUX.EQ.0) GO TO 30
      J=IDX(NSET,I+7)+NUX-1
      MH1=KE(J)
      GO TO 40
   30 MH1=1
   40 J=IDX(NSET,I+7)+NX-1
      MH2=KE(J)
C-----IDENTIFY OLD CHAINAGES OF FIRST AND LAST MANHOLES
      L=ID(NUM,8)+MH1-1
      CH1=GA(L)
      L=ID(NUM,8)+MH2-1
      CH2=GA(L)
C-----IDENTIFY CHAINAGES AND AREAS FOR EACH M/H
      K=0
      DO 50 J=MH1,MH2
      K=K+1
      L=ID(NUM,8)+J-1
      CHAIN(K)=GA(L)-CH1
      L=ID(NUM,9)+J-1
      AREA(K)=GA(L)
   50 CONTINUE
C------IDENTIFY GROUND LEVEL DATA
      NGL=ID(NUM,4)
      DO 60 J=1,NGL
      L=ID(NUM,11)+J-1
      IF (GA(L).GT.CH1-0.001) GO TO 70
   60 CONTINUE
   70 L=0
      DO 80 K=J,NGL
      L=L+1
      L1=ID(NUM,10)+K-1
      L2=ID(NUM,11)+K-1
      XGL(L)=GA(L2)-CH1
      ZGL(L)=GA(L1)
      IF (GA(L2).GT.CH2-0.001) GO TO 90
   80 CONTINUE
C-----IDENTIFY UPSTREAM BRANCH
   90 NUB=0
      IF (NUX.EQ.0.OR.I.EQ.1) NUB=1
      NUMB=2*NPARA+I
C-----IDENTIFY ID'S
      N=MH2-MH1+1
      IDN(1)=ID(NUM,1)
      IDN(2)=NUB
      IDN(3)=N
      IDN(4)=L
      IDN(5)=0
      IF (NUB.EQ.0) GO TO 95
      IKE=IKE+1
      IDN(5)=IKE
      KE(IKE)=NUMB
   95 IDN(6)=IGA+1
      IDN(7)=IDN(6)+N
      IDN(8)=IDN(7)+N
      IDN(9)=IDN(8)+N
      IDN(10)=IDN(9)+N
      IDN(11)=IDN(10)+L
C-----STORE G.L.AND OFFSET FOR D/S M/H
      DSGL(I)=ZGL(L)
      LL=ID(NUM,6)+MH2-1
      DSOFF(I)=GA(LL)
C-----STORE THE DATA FOR THIS BRANCH IN KE,GA AND COMMON WHERE
C-----(STORE CHAINAGES AND AREAS)
      IGA=IGA+2*N
      K=0
      DO 100 J=1,N
      IGA=IGA+1
      IGB=IGA+N
      K=K+1
      GA(IGA)=CHAIN(K)
      GA(IGB)=AREA(K)
  100 CONTINUE
```

```
      IGA=IGA+N
      K=0
      DO 110 J=1,L
      IGA=IGA+1
      IGB=IGA+L
      K=K+1
      GA(IGA)=ZGL(K)
      GA(IGB)=XGL(K)
  110 CONTINUE
      IGA=IGA+L
C-----SPECIFY ID'S
      DO 120 J=1,11
      ID(I,J)=IDN(J)
  120 CONTINUE
C-----CREATE BRANCH DATA FOR EACH SECTION OF CROSS DRAIN
      II=NPARA+1
      IJ=II+NPARA-2
      K=NPARA+1
      DO 130 I=II,IJ
      K=K-1
      ID(I,1)=11
      ID(I,2)=2
      IF (I.EQ.II) ID(I,2)=1
      ID(I,3)=2
      ID(I,4)=2
      ID(I,5)=IKE+1
      DO 125 JJ=6,11
      JK=1+2*(JJ-6)
  125 ID(I,JJ)=IGA+JK
C-----DEFINE CHAINAGES AND AREAS
      GA(IGA+5)=0.0
      GA(IGA+6)=ABS(DSOFF(K)-DSOFF(K-1))
      GA(IGA+7)=0.0
      GA(IGA+8)=0.0
      GA(IGA+9)=DSGL(K)
      GA(IGA+10)=DSGL(K-1)
      GA(IGA+11)=0.0
      GA(IGA+12)=GA(IGA+6)
      IGA=IGA+12
C-----DEFINE U/S BRANCHES
      KE(IKE+1)=K
      IF(I.EQ.II) GO TO 128
      KE(IKE+2)=I-1
  128 IKE=IKE+ID(I,2)
  130 CONTINUE
C-----CREATE UPSTREAM BRANCH DATA FOR DUMMY PIPE D/S OF X/D SET
      I=IJ+1
      ID(I,2)=2
      ID(I,5)=IKE+1
      KE(IKE+1)=1
      KE(IKE+2)=IJ
      IKE=IKE+2
      NBRAN=I-1
      NBRA=I+1
      IF (NUX.EQ.0) NBRA=I+NPARA
      NB=0
      IDT=IGA+1
      IDA=IDT+NBRA
      IDL=IDA+NBRA
      IDD=IDL+NBRA
      IDK=IKE+1
      IDJ=IDD+NBRA
      IDG=IDJ+MJ*NBRA
C-----CREATE DUMMY PIPE VALUES UPSTREAM OF X/D SET AS REQUIRED
      JJ=0
      NP=1
      L=IDR-1
      IF (NUX.NE.0) GO TO 140
      L=IDP-1
      NP=NPARA
  140 DO 160 II=1,NP
      I=I+1
      K=IDT+I-1
      GA(K)=GA(L+1)
      K=IDA+I-1
      GA(K)=GA(L+2)
      K=IDL+I-1
      GA(K)=GA(L+3)
      K=IDD+I-1
      GA(K)=GA(L+4)
      L=L+4
      K=IDJ-1+MJ*(I-1)
      DO 159 JI=1,MJ
      K=K+1
      L=L+1
  159 GA(K)=GA(L)
      JJ=JJ+1
      NUM=IDX(NSET,JJ)
      ID(I,1)=ID(NUM,1)
  160 CONTINUE
      RETURN
 1001 FORMAT (1H0,5X,20HUPSTREAM CROSSDRAIN=,I6,
     124H  DOWNSTREAM CROSSDRAIN=,I6)
 2000 FORMAT (1H0/////5X,13HENTERED SETUP)
      END

END OF SEGMENT, LENGTH   609, NAME  SETUP
```

```
      SUBROUTINE SUBNET (KE,GA,NKE,NGA,NUX)
C-----THIS SUBROUTINE PRODUCES A SET OF OPTIMAL DESIGNS FOR A NETWORK
C       OVER A RANGE OF D/S STATES
      DIMENSION KE(NKE),GA(NGA)
      COMMON/DATA/ NSET,NBRAN,NB,NPIPE,NXOSET,MITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,I,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      IF (ND.GT.0)     WRITE (6,2000)
C-----BRANCHES HAVE BEEN SEQUENTIALLY ORDERED:DESIGN EACH IN TURN
      DO 100 NB=1,NBRAN
C-----DESIGN BRANCH TO A MINIMUM GRADIENT
      CALL MGRAD (KE,GA,NKE,NGA)
C-----SORT OUT UPSTREAM COSTS
      NN=IDG+2*ID(NB,3)
      CALL COMB (KE,GA,NKE,NGA,NN,GA(IDG),KE(IDK))
C-----PRODUCE OPTIMAL DESIGNS FOR THIS BRANCH FOR RANGE OF D/S STATES
      CALL NBRUN (KE,GA,NKE,NGA)
C-----PRODUCE TRACE BACK UP BRANCH FOR RANGE OF D/S STATES
      IF (NB.EQ.1)  CALL TRAIL (KE,GA,NKE,NGA)
  100 CONTINUE
      NB=NBRAN+1
      NPIPE=MOD(ID(1,1),10)
      K=ID(NB,10)+ID(1,4)-1
      CALL UPVAL(KE,GA,NKE,NGA,TUS,A,Z,D,GA(K))
      GA(IDR)=TUS
      GA(IDR+1)=A
      GA(IDR+2)=Z
      GA(IDR+3)=D
      CALL SIZED (PDA,NPIPE,I,D)
      CALL COMB (KE,GA,NKE,NGA,IDR+4,Z,I)
C-----DEFINE MAX LEVEL AND MAX DIAM NUMBER FOR BRANCH 1
      ZMAX=GA(IDL)
      DMA=GA(IDD)
      CALL SIZED (PDA,NPIPE,JMAX,DMA)
C-----FOR EACH D/S STATE,TRACE BACK TO OBTAIN REF TO STATE U/S OF SUBNET
C-----READ TRAIL DATA FROM MAG TAPE
      REWIND 9
      NK=IDK-1
      DO 190 J=1,MJ
      NK=NK+1
  190 READ (9) KE(NK),REAL,REAL
      REWIND 9
      NK1=NKE-MJ
      NG1=IDR+3
C-----FOR EACH D/S STATE
      DO 200 J=1,JEND
      J1=I-JEND+J
      DDS=PDA(NPIPE,J1)
      DO 200 M=1,MEND
      NK1=NK1+1
      KE(NK1)=0
      NG1=NG1+1
      IF (GA(NG1).GT.999999.0) GO TO 200
      ZDS=Z-DELTA*FLOAT(M-1)
C-----IDENTIFY U/S M AND J
      CALL BRIDGE (GA,NGA,ZDS,DDS,ZMAX,JMAX,NPIPE,IDJ,M1,J1)
      IF (ND.GT.1) WRITE (6,3000) ZDS,DDS,ZMAX,JMAX,NPIPE,IDJ,M1,J1,M,J
      K=IDK+(J1-1)*MEND+M1-1
      KE(NK1)=NUX*MJ+KE(K)
  200 CONTINUE
      IF (ND.LT.2) RETURN
      NK1=NKE-MJ+1
      WRITE (6,1001) NK1,NKE
      WRITE (6,1002) (KE(I),I=NK1,NKE)
      RETURN
 2000 FORMAT (5X,14HENTERED SUBNET)
 1001 FORMAT (1H0/5X,47HREFERENCES ACROSS SUBNETWORK STORED IN KE FROM ,
     1I6,4H TO ,I6)
 1002 FORMAT (1X,I5,19I6)
 3000 FORMAT (1X,3F8.3,7I6)
      END

END OF SEGMENT, LENGTH   389, NAME  SUBNET
```

```
      SUBRØUTINE RAIN (Y,T,RI)
C-----Y=RETURN PERIØD(YRS),T=TIME(MINS),RI=INTENSITY (MM/HR)
      RI=60./T*((Y*T*202.26)**.28169-2.54)
      IF (RI-33.) 10,10,40
   10 RI=AMAX1(RI,0.0)
      RETURN
   40 NI=0
C-----ITERATIØN LØØP FØR HØLLAND FØRMULA
   50 NI=NI+1
      F=ALØG(15240.0/Y/RI/T*(RI*T/1524.0+0.1)**3.55)-1.0+0.0314*RI
      DF=-1.0/RI+3.55/(RI+152.4/T)+.0314
      X=F/DF
      RI=RI-X
      IF (ABS(X)-0.01) 10,10,60
   60 IF (NI.LT.10) GØ TØ 50
      CALL MESAGE(17,0)
      END
```

END ØF SEGMENT, LENGTH   98, NAME RAIN

```
      SUBRØUTINE VELØC (SLØPE,DIAM,RK,V,Q)
C-----CALCULATES FLØWS FRØM CØLEBRØØK-WHITE FØRMULA
      SQ=SQRT(SLØPE*DIAM)
      V=-8.818*SQ*ALØG10(RK/3.7/DIAM+0.6471E-6/SQ/DIAM)
      Q=V*DIAM*DIAM*0.7854
      RETURN
      END
```

END ØF SEGMENT, LENGTH   63, NAME VELØC

```
      SUBROUTINE MGRAD (KE,GA,NKE,NGA)
C-----THIS SUBROUTINE DESIGNS AN INDIVIDUAL BRANCH TO MINIMUM GRADIENT
      DIMENSION KE(NKE),GA(NGA)
      COMMON/DATA/NSET,NBRAN,NB,NPIPE,NXDSFT,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      IF (ND.GT.0)      WRITE (6,2000)
C-----DEFINE PIPE TYPE
      NPIPE=MOD(ID(NB,1),10)
C-----IDENTIFY GROUND LEVEL AT UPSTREAM MANHOLE
      N=ID(NB,10)
      GLUS=GA(N)
C-----OBTAIN START OF RUN VALUES
      CALL UPVAL (KE,GA,NKE,NGA,TUS,AREAUS,ZUS,DUS,GLUS)
      CALL RAIN (RP,TUS/60.0,RI)
C-----IDENTIFY NUMBER OF MANHOLES IN THIS BRANCH
      NMH=ID(NB,3)
      NVAL=IDK+NMH+(NMH-1)*MJ
      IF (NKE.LT.NVAL)    CALL MESAGE (8,NVAL)
      NVAL=IDG+NMH*(MJ+2)
      IF (NGA.LT.NVAL)    CALL MESAGE (9,NVAL)
      NG1=IDG
      NG2=NG1+NMH
      NK1=IDK
      GA(NG1)=ZUS
      GA(NG2)=AREAUS*RI/3.6E6
      JJ=KE(NK1)
      N=ID(NB,8)
      MUS=ID(NB,11)
      LUS=ID(NB,10)
      K=ID(NB,9)
      GMIN=PDA(NPIPE,5)
      COVER=PDA(NPIPE,1)
      CHUS=0.0
C-----FOR EACH MANHOLE POSITION
      DO 50 I=2,NMH
      N=N+1
      NG1=NG1+1
      NG2=NG2+1
      NK1=NK1+1
      CHDS=GA(N)
      DIST=CHDS-CHUS
      ZDS=ZUS-DIST*GMIN
      M=MUS
      L=LUS
   30 M=M+1
      X=GA(M)
      L=L+1
      Z=GA(L)
      ZDS=AMIN1(Z-COVER-(CHDS-X)*GMIN,ZDS)
      IF (CHDS.GT.X+0.001) GO TO 30
      SLOPE=(ZUS-ZDS)/DIST
      K=K+1
      AREADS=AREAUS+GA(K)
C-----CALCULATE REQUIRED DIAMETER
   35 DI=PDA(NPIPE,JJ)
      IF (JJ.GT.20.OR.DI.LT.0.01.OR.DI.GT.2.0) CALL MESAGE (18,JJ)
      CALL VELOC (SLOPE,DI,PDA(NPIPE,2),V,0)
      IF (V-PDA(NPIPE,3)) 38,36,36
   36 IF (V-PDA(NPIPE,4)) 42,42,43
   42 TF=TUS+DIST/V
      CALL RAIN (RP,TF/60.0,RI)
      FLOW=AREADS*RI/3.6E6
      IF (FLOW.LE.Q) GO TO 40
   37 JJ=JJ+1
      GO TO 35
   38 V=PDA(NPIPE,3)
      TF=TUS+DIST/V
      CALL RAIN (RP,TF/60.0,RI)
```

```
      FLOW=AREADS*RI/3.6E6
      Q=0.7854*DI*DI*V
      IF (Q-FLOW) 37,39,39
   39 CALL FINDG (DI,V,PDA(NPIPE,2),SLOPE)
      ZDS=ZUS-DIST*SLOPE
      M=MUS
      L=LUS
   41 M=M+1
      X=GA(M)
      L=L+1
      Z=GA(L)
      ZDS=AMIN1(Z-COVER-(CHDS-X)*SLOPE,ZDS)
      IF (CHDS.GT.X+0.001) GO TO 41
      GO TO 40
   43 V=PDA(NPIPE,4)
      TF=TUS+DIST/V
      CALL RAIN (RP,TF/60.0,RI)
      FLOW=AREADS*RI/3.6E6
      Q=0.7854*DI*DI*V
      IF (Q-FLOW) 37,44,44
   44 CALL FINDG (DI,V,PDA(NPIPE,2),SLOPE)
      ZUSNEW=ZDS+SLOPE*DIST+ZD
      IF (ZUSNEW.GT.ZUS) GO TO 40
      IF (NMH.GT.2) CALL MESAGE (19,I)
      GA(NG1-1)=ZUSNEW
      IF (NO.GT.0) WRITE (6,1001) ZUSNEW
C-----STORE DOWNSTREAM ZONE LEVEL,FLOW AND PIPE SIZE
   40 GA(NG1)=ZDS
      GA(NG2)=FLOW
      KE(NK1)=JJ
C-----REDEFINE UPSTREAM VALUES
      CHUS=CHDS
      ZUS=ZDS
      AREAUS=AREADS
      TUS=TF
      MUS=M
      LUS=L
   50 CONTINUE
C-----STORE END OF BRANCH VALUES
      M=IDT+NB-1
      GA(M)=TUS
      M=IDA+NB-1
      GA(M)=AREAUS
      M=IDL+NB-1
      GA(M)=ZUS
      M=IDD+NB-1
      GA(M)=PDA(NPIPE,JJ)
      IDZ=IDG+2*NMH-1
      IDY=IDK+NMH-1
      IF (NO.LT.3) RETURN
      WRITE (6,1002) IDT,IDZ,IDK,IDY
      WRITE (6,1001) (GA(I),I=IDT,IDZ)
      WRITE (6,1002) (KE(I),I=IDK,IDY)
      RETURN
 1001 FORMAT (1X,F11.3,9F12.3)
 1002 FORMAT (1X,I5,19I6)
 2000 FORMAT (5X,13HENTERED MGRAD)
      END
```

```
      SUBROUTINE UPVAL (KE,GA,NKE,NGA,TUS,AREAUS,ZUS,DUS,GLUS)
C-----FINDS THF START RF RUN VALUES RF TIME AREA LEVEL AND DIAMETER
C     FOR BRANCH NB
      DIMENSION KE(NKE),GA(NGA)
      COMMON/DATA/NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMRN/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      IF (ND.GT.0)   WRITE (6,2000)
      TUS=T
      AREAUS=0.0
      ZUS=999999.9
      DUS=0.0
C-----IDENTIFY NUMBER OF UPSTREAM BRANCHES
    5 NUB=ID(NB,2)
      IF (NUR.EQ.0) GO TO 20
      N=ID(NR,5)-1
C-----FOR EACH UPSTREAM BRANCH
      DO 10 I=1,NUB
      N=N+1
C-----IDENTIFY BRANCH NUMBER AND UPSTREAM VALUES
      NBRA=KE(N)
      M=IDT+NBRA-1
      TUS=AMAX1(TUS,GA(M))
      M=IDA+NBRA-1
      AREAUS=AREAUS+GA(M)
      M=IDL+NBRA-1
      ZUS=AMIN1(ZUS,GA(M))
      M=IDD+NBRA-1
      DUS=AMAX1(DUS,GA(M))
   10 CONTINUE
   20 ZUS=AMIN1(ZUS,GLUS-PDA(NPIPE,1))
      GA(IDG)=ZUS
      CALL SIZED (PDA,NPIPE,KE(IDK),DUS)
      IF (ND.LT.1) RETURN
      WRITE (6,2001) NB,NUB,TUS,AREAUS,ZUS,DUS
      RETURN
 2000 FORMAT (5X,13HENTERED UPVAL)
 2001 FORMAT (5X,6HBRANCH,I6,I8,23HBRANCHES UPSTREAM,TIME=,F12.3,
     15HAREA=,F12.3,6HLEVEL=,F12.3,5HDIAM=,F12.3)
      END

END OF SEGMENT, LENGTH   212, NAME  UPVAL
```

```
      SUBROUTINE COMB (KE,GA,NKE,NGA,NN,ZTDS,NDDS)
      DIMENSION KE(NKE),GA(NGA)
      COMMON/DATA/NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDR
      IF (ND.GT.0)  WRITE (6,2000)
C-----SET COST OF ARRIVAL AT FIRST MANHOLE IN RUN TO ZERO
      DO 10 I=1,MJ
      J=NN+I-1
   10 GA(J)=0.0
      NUB=ID(NB,2)
      IF (NUB.GT.0) GO TO 30
      IF (JEND.EQ.1) GO TO 100
C-----DEFINE INFEASIBLE PIPE ZONES AT THIS UPSTREAM END OF THE NETWORK
      J2=JEND-1
      K=NN-1
      DO 20 J=1,J2
      DO 20 M=1,MEND
      K=K+1
   20 GA(K)=999999.9
      GO TO 100
   30 K=ID(NB,5)-1
C-----FOR EACH UPSTREAM BRANCH
      DO 90 I=1,NUB
C-----(FIND U/S PIPE NUMBER,TYPE AND MAX.DIAM.NUMBER)
      K=K+1
      NR=KE(K)
      NPIPUS=MOD(ID(NR,1),10)
      L=IDD+NR-1
      CALL SIZED (PDA,NPIPUS,JI,GA(L))
      L=IDL+NR-1
      ZTUS=GA(L)
      DO 80 J=1,JEND
      JJ=NDDS-JEND+J
      IF (JJ.GE.6) GO TO 40
      IF (I.EQ.1) GO TO 110
      GO TO 80
   40 DDS=PDA(NPIPE,JJ)
      DO 70 M=1,MEND
      ZDS=ZTDS-FLOAT(M-1)*DELTA
      COSTA=999999.9
      DO 60 JUS=1,JEND
      JJUS=JI-JEND+JUS
      IF (JJUS.LT.6) GO TO 60
      DUS=PDA(NPIPUS,JJUS)
      IF (DUS.GT.DDS+0.001) GO TO 65
      DO 55 MUS=1,MEND
      ZUS=ZTUS-FLOAT(MUS-1)*DELTA
      IF (ZUS.LT.ZDS - 0.001)   GO TO 60
      L=IDJ-1+(NR-1)*MJ+(JUS-1)*MEND+MUS
      COSTA=AMIN1(COSTA,GA(L))
   55 CONTINUE
   60 CONTINUE
   65 L=NN-1+(J-1)*MEND+M
      GA(L)=AMIN1(GA(L)+COSTA,999999.9)
   70 CONTINUE
   80 CONTINUE
   90 CONTINUE
  100 IF (ND.LT.1) RETURN
      L=NN+MJ-1
      WRITE (6,1001) NB,NUB,NN,L
      WRITE (6,1002) (GA(I),I=NN,L)
      RETURN
  110 L=NN-1+(J-1)*MEND
      DO 120 KEW=1,MEND
      L=L+1
  120 GA(L)=999999.9
      GO TO 80
 1001 FORMAT (1H0,5X,20HU/S COSTS FOR BRANCH,I6,19H (NO. OF U/S PIPES=,
     1I6,1H),2I6)
 1002 FORMAT (1X,F11.3,9F12.3)
 2000 FORMAT (5X,12HENTERED COMB)
      END

END OF SEGMENT, LENGTH   407, NAME  COMB
```

```
      SUBRØUTINE NBRUN (KE,GA,NKE,NGA)
      DIMENSIØN KE(NKE),GA(NGA)
      CØMMØN/DATA/NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      CØMMØN/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      CØMMØN/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      LØGICAL NEWN
      IF (ND.GT.0) WRITE (6,2000)
      LEND=ID(NB,4)
      NEND=ID(NB,3)
      IDM=IDG+2*NEND
      IDB=IDK+NEND
      IDQ=IDM+NEND*MJ
      IDN=IDB+(NEND-1)*MJ
      KCØST=IDN+MJ-1
      KREF =IDB-1
      N=1
C-----DEFINE D/S STATE (MANHØLE N)
   10 N=N+1
C-----(DEFINE PARAMETERS DEPENDENT ØN "N")
      NEWN=.TRUE.
C-----(RELATIVE CHAINAGE)
      K=ID(NB,8)+N-1
      XDS=GA(K)
C-----(TØP ØF ZØNE)
      K=IDG+N-1
      ZTDS=GA(K)
C-----(FLØW)
      K=IDG+N-1+NEND
      QN=GA(K)
C-----(MAX.PIPE DIAM.NUMBER)
      K=IDK+N-1
      NDN=KE(K)
      J=0
C-----DEFINE D/S STATE (DIAMETER J)
   20 J=J+1
      JPIPE=NDN-JEND+J
      M=0
C-----DEFINE D/S STATE (LEVEL M)
   30 M=M+1
C-----D/S STATE (M,J,N) IS NØW DEFINED:SET CØST ØF ARRIVAL AT (M,J,N)
C     ARTIFICIALLY HIGH
      KCØST=KCØST+1
      KREF=KREF+1
      GA(KCØST)=999999.9
      KE(KREF)=0
C-----DØES JPIPE CØRRESPØND TØ A REAL PIPE?
      IF (JPIPE.LT.6) GØ TØ 300
      ZDS=ZTDS-FLØAT(M-1)*DELTA
      NN=0
C-----DEFINE UPSTREAM STATE (MANHØLE)
      NREF=0
   40 NN=NN+1
      K=ID(NB,8)+NN-1
      XUS=GA(K)
C-----IS DISTANCE BETWEEN MANHØLES PERMISIBLE?
      IF (XDS-XUS.GT.DMAX+0.001.AND.NEND.GT.2) GØ TØ 40
      IF (XDS-XUS.LT.DMIN-0.001.AND.NEND.GT.2) GØ TØ 290
      NREF=NREF+1
      K=IDG+NN-1
      IF (NKE.LT.IDM-1+3*NREF)    CALL MESAGE (10,0)
      IF (NGA.LT.IDQ-1+2*NREF)    CALL MESAGE (11,0)
      NK=IDM+3*(NREF-1)
      NG=IDQ+2*(NREF-1)
      ZTUS=GA(K)
      IF (.NØT.NEWN) GØ TØ 150
C-----DEFINE PARAMETERS DEPENDENT ØN CØMBINATIØN ØF N AND NN
C-----(DEFINE INTERMEDIATE GRØUND LEVELS)
      DØ 50 L=2,LEND
      K=ID(NB,11)+L-1
      IF (GA(K).GT.XUS+0.001)   GØ TØ 60
   50 CØNTINUE
   60 L1=L
      KGXL1=K
      KGZL1=K-LEND
      L2=L1-1
      A=0.0
      LEVEL=1
      IF (L1.EQ.LEND) GØ TØ 140
      IF (GA(KGXL1+1).GT.XDS+0.001) GØ TØ 140
      L3=L1+1
      DØ 70 L=L3,LEND
      K=ID(NB,11)+L-1
      IF (GA(K).GT.XDS-0.001) GØ TØ 80
   70 CØNTINUE
   80 L2=L-1
      KGXL2=ID(NB,11)+L2-1
      KGZL2=KGXL2-LEND
```

```
C-----DEFINE AREA OF LONG SECTION ABOVE STRAIGHT LINE FROM NN TO N
      A=-GA(KGXL1-1)*(GA(KGZL1)-GA(KGZL2+1))*0.5
      DO 85 LL=L1,L2
      KGXLL=ID(NB,11)+LL-1
      KGZLL=KGXLL-LEND
   85 A=A-GA(KGXLL)*(GA(KGZLL+1)-GA(KGZLL-1))*0.5
      A=A-GA(KGXL2+1)*(GA(KGZL1-1)-GA(KGZL2))*0.5
C-----IS GROUND LEVEL,CONCAVE,CONVEX OR VARIABLE
      GSLOPE=(GA(KGZL2+1)-GA(KGZL1-1))/(XDS-XUS)
      DO 130 L=L1,L2
      KGXL=ID(NB,11)+L-1
      KGZL=KGXL-LEND
      PSLOPE=(GA(KGZL)-GA(KGZL1-1))/(GA(KGXL)-XUS)
      IF (PSLOPE.LT.GSLOPE-0.00001) GO TO 90
      IF (PSLOPE.LT.GSLOPE+0.00001) GO TO 130
      GO TO (110,100,130),LEVEL
   90 GO TO (120,130,100),LEVEL
  100 LEVEL=4
      GO TO 140
  110 LEVEL=3
      GO TO 130
  120 LEVEL=2
  130 CONTINUE
C-----DEFINE AND STORE GROUND CONDITIONS AND DISTANCE BETWEEN MANHOLES
C     FOR THIS COMBINATION OF N AND NN
  140 KE(NK)=LEVEL
      KE(NK+1)=L1
      KE(NK+2)=L2
      GA(NG)=XDS-XUS
      GA(NG+1)=A
      IF (ND.GT.1) WRITE (6,2002) NN,N,NREF,LEVEL,L1,L2,GA(NG),A
C-----DEFINE UPSTREAM STATE (U/S PIPE DIAMETER)
  150 DIST=GA(NG)
      AREA=GA(NG+1)
      LEVEL=KE(NK)
      L1=KE(NK+1)
      L2=KE(NK+2)
      JJ=0
  160 JJ=JJ+1
      K=IDK+NN-1
      JJPIPE=KE(K)-JEND+JJ
      IF (JJPIPE.LT.6) GO TO 260
C-----DEFINE UPSTREAM STATE (CROWN LEVEL MM)
      MM=0
  170 MM=MM+1
      ICOST=IDM-1+(NN-1)*MJ+(JJ-1)*MEND+MM
C-----CHECK FEASIBILITY OF SOLUTION
C-----(U/S STATE FEASIBLE?)
      IF (GA(ICOST).GT.999999.0) GO TO 250
C-----(PIPE SLOPE WITHIN RESTRAINTS?)
      ZUS=ZTUS-FLOAT(MM-1)*DELTA
      SLOPE=(ZUS-ZDS)/DIST
      IF (SLOPE.LT.PDA(NPIPE,5)-0.00001) GO TO 260
C-----(PIPE CAPACITY SUFFICIENT?)
      CALL VELOC (SLOPE,PDA(NPIPE,JPIPE),PDA(NPIPE,2),VEL,QFULL)
      IF (QFULL.LT.QN) GO TO 250
C-----(VELOCITY ACCEPTABLE?)
      IF (VEL.LT.PDA(NPIPE,3)-0.001) GO TO 260
      IF (VEL.GT.PDA(NPIPE,4)+0.005) GO TO 250
C-----(DEPTH OF COVER RESTRAINT VIOLATED?)
      GO TO (240,180,240,180),LEVEL
  180 DO 200 L=L1,L2
      KGXL=ID(NB,11)+L-1
      KGZL=KGXL-LEND
      IF (ZUS-SLOPE*(GA(KGXL)-XUS).GT.GA(KGZL)-PDA(NPIPE,1)+0.001) GO
     1TO 250
  200 CONTINUE
C-----SOLUTION IS FEASIBLE SO COST AND COMPARE WITH PREVIOUS CHEAPEST
  240 KGZL1=ID(NB,10)+L1-1
      KGZL2=ID(NB,10)+L2-1
      CALL COSTIT (JPIPE-5,AREA,GA(KGZL1-1)-ZUS,GA(KGZL2+1)-ZDS,DIST,C)
      C=C+GA(ICOST)
      IF(C.GT.GA(KCOST)-0.001) GO TO 250
      GA(KCOST)=C
      KE(KREF)=(NN-1)*MJ+(JJ-1)*MEND+MM
      IF (ND.GT.1)  WRITE (6,2004) N,J,M,NN,JJ,MM,XDS,XUS,ZDS,ZUS,
     1PDA(NPIPE,JPIPE),PDA(NPIPE,JJPIPE),KREF,KCOST,KE(KREF),GA(KCOST)
C-----MOVE ON TO NEXT U/S STATE
  250 IF (MM.LT.MEND) GO TO 170
  260 IF (JJPIPE.LT.JPIPE.AND.JJ.LT.JEND) GO TO 160
      IF (NN.LT.N-1) GO TO 40
```

```
C-----MØVE ØN TØ NEXT D/S STATE
  290 NEWN=.FALSE.
  300 IF (M.LT.MEND) GØ TØ 30
      IF (J.LT.JEND) GØ TØ 20
      IF (N.LT.NEND) GØ TØ 10
C-----STØRE DØWNSTREAM CØSTS FØR THIS BRANCH
      I=IDH+(NEND-1)*MJ-1
      J=IDJ+(NB-1)*MJ-1
      JA=J+1
      JB=J+MJ
      DØ 310 K=1,MJ
      I=I+1
      J=J+1
      GA(J)=GA(I)
  310 CØNTINUE
      IF (ND.EQ.0) RETURN
      WRITE (6,2005) NB
      WRITE (6,2006) (GA(J),J=JA,JB)
      I=IDT+NB-1
      J=IDA+NB-1
      K=IDL+NB-1
      L=IDD+NB-1
      WRITE (6,2007) NB,GA(I),GA(J),GA(K),GA(L)
      RETURN
 2000 FØRMAT (5X,13HENTERED NBRUN)
 2002 FØRMAT (10X,15HGRØUND FRØM M/H,I4,6HTØ M/H,5I4,2(F9.3)/5X,
     1108HN     J     M     NN    JJ    MM     XDS        XUS        ZDS        ZUS
     2     DDS        DUS     KREF  KCØST  REF        CØST)
 2004 FØRMAT (1X,6I5,6F9.3,3I6,F12.3)
 2005 FØRMAT (1H0,5X,20HD/S CØSTS FØR BRANCH,I6)
 2006 FØRMAT (1X,F11.3,9F12.3)
 2007 FØRMAT (1H0,5X,7HBRANCH ,I6,26H  DØWNSTREAM VALUES: TIME=,F12.3,
     17H  AREA=,F12.3,8H  LEVEL=,F12.3,7H  DIAM=,F12.3//)
      END

END ØF SEGMENT, LENGTH  1083, NAME  NBRUN
```

```
      SUBRØUTINE CØSTIT (J,AREA,DUS,DDS,DIST,CØST)
      DEPTH=(DUS+DDS)/2.0+AREA/DIST
      GØ TØ (10,20,30,40,50,60,70) ,J
   10 CØST=DIST*(2.8+4.1*DEPTH)+30.0+70.0*DUS
      RETURN
   20 CØST=DIST*(5.7+4.1*DEPTH)+30.0+70.0*DUS
      RETURN
   30 CØST=DIST*(8.9+4.1*DEPTH)+30.0+75.0*DUS
      RETURN
   40 CØST=DIST*(12.3+4.4*DEPTH)+30.0+80.0*DUS
      RETURN
   50 CØST=DIST*(15.9+4.7*DEPTH)+30.0+85.0*DUS
      RETURN
   60 CØST=DIST*(19.7+5.0*DEPTH)+30.0+90.0*DUS
      RETURN
   70 CØST=DIST*(23.7+5.3*DEPTH)+30.0+95.0*DUS
      RETURN
      END

END ØF SEGMENT, LENGTH   139, NAME  CØSTIT
```

```
      SUBROUTINE TRAIL (KE,GA,NKF,NGA)
C-----TRACES BACK UP A BRANCH FOR EACH D/S STATE AND STORES THE TRACE ON
C     MAGNETIC TAPE FILE
      DIMENSION KE(NKE),GA(NGA),KTEMP(200),ZTEMP(200),DTEMP(200)
      COMMON/DATA/NSET,NRRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      IF (ND.GT.0) WRITE (6,2000)
      K=IDM-MJ-1
      NN=0
      DO 50 I=1,MJ
C-----IDENTIFY TRACE BACK REFERENCE FROM END MANHOLE
      K=K+1
      MJN=KE(K)
C-----WRITE REFERENCE TO TAPE
   10 IF (NXDSET.GT.0.AND.MJN.GT.MJ) GO TO 20
C-----IDENTIFY ACTUAL LEVEL AND DIAMETER
      IF (MJN.NE.0) GO TO 30
      Z=0.0
      D=0.0
      GO TO 40
   30 N=(MJN-1)/MJ+1
      J=(MJN-MJ*(N-1)-1)/MEND+1
      M=MJN-MJ*(N-1)-MEND*(J-1)
      L=IDK+N-1
      JMAX=KE(L)
      JPIPE=JMAX-JEND+J
      D=PDA(NPIPE,JPIPE)
      L=IDG+N-1
      Z=GA(L)-DELTA*FLOAT(M-1)
   40 WRITE (9) MJN,Z,D
      NN=NN+1
      IF (MJN.LE.MJ) GO TO 50
C-----IDENTIFY NEXT TRACE BACK REFERENCE
   20 L=IDB-MJ+MJN-1
      MJN=KE(L)
      GO TO 10
   50 CONTINUE
C-----CHECK ON CONTENTS OF MAG. TAPE
      IF (ND.LT.2) GO TO 80
      DO 60 I=1,NN
   60 BACKSPACE 9
      DO 70 I=1,NN
      READ (9) KTEMP(I),ZTEMP(I),DTEMP(I)
   70 CONTINUE
      WRITE (6,1000) (KTEMP(I),I=1,NN)
      WRITE (6,1001) (ZTEMP(I),I=1,NN)
      WRITE (6,1001) (DTEMP(I),I=1,NN)
   80 NITEM=NITEM+NN
      RETURN
 2000 FORMAT (5X,13HENTERED TRAIL)
 1000 FORMAT (1X,I5,19I6)
 1001 FORMAT (1X,F11.3,9F12.3)
      END

END OF SEGMENT, LENGTH   284, NAME  TRAIL
```

```
      SUBROUTINE BRIDGE (GA,NGA,ZDS,DDS,ZMAX,JMAX,NPIPE2,NN,M1,J1)
      DIMENSION GA(NGA)
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ED,DMAX,DMIN,DELTA
C-----FINDS THE VALUES OF M1,J1 FOR AN UPSTREAM PIPE AT A MANHOLE
C-----CORRESPONDING TO LEVELS AND DIAMETERS ZDS,DDS OF OUTGOING PIPE.
C-----SUCH THAT U/S LEVEL .GE. ZDS,AND U/S DIAM .LE. DDS AND U/S COST IS MIN.
      IF (ND.GT.1) WRITE (6,2000)
      M1=0
      J1=0
      COSTUS=999999.9
      DO 20 J=1,JEND
      JA=JMAX-JEND+J
      IF (JA.LT.6) GO TO 20
      DUS=PDA(NPIPE2,JA)
      IF (DUS.GT.DDS+0.001) GO TO 30
      DO 10 M=1,MEND
      ZUS=ZMAX-DELTA*FLOAT(M-1)
      IF (ZUS.LT.ZDS-0.001) GO TO 20
      K=NN+(J-1)*MEND+M-1
      COST=GA(K)
      IF (COST.GE.COSTUS-0.001) GO TO 10
      COSTUS=COST
      M1=M
      J1=J
   10 CONTINUE
   20 CONTINUE
   30 RETURN
 2000 FORMAT (5X,14HENTERED BRIDGE)
      END
```

END OF SEGMENT, LENGTH 165, NAME BRIDGE

```
      SUBROUTINE SIZED (PDA,NPIPE,J,D)
C-----FINDS THE PIPE NUMBER J CORRESPONDING TO OR GREATER THAN DIAM D
      DIMENSION PDA(9,20)
      DO 10 J=6,20
      IF (PDA(NPIPE,J).GE.D-0.001) RETURN
   10 CONTINUE
      CALL MESAGE (15,NPIPE)
      END
```

END OF SEGMENT, LENGTH 55, NAME SIZED

```
      SUBRØUTINE TRACE (KE,GA,NKE,NGA,IG,IK)
      DIMENSIØN KE(NKE),GA(NGA)
      LØGICAL LX
      CØMMØN/DATA/NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      CØMMØN/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      CØMMØN/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      IF (ND.GT.0) WRITE (6,2000)
      LX=.FALSE.
      NPIPE=MØD(ID(NBRAN,1),10)
      IF (NXDSET.EQ.0) LX=.TRUE.
      IF (LX) IG=NGA
      IF (LX) IK=NKE
      NK=NKE+1
C-----IDENTIFY CHEAPEST END STATE
      I=IDJ+(NB-1)*MJ-1
      KREF=0
      CØST=999999.9
      DØ 20 J=1,MJ
      I=I+1
      IF (GA(I).GT.CØST-0.001) GØ TØ 20
      CØST=GA(I)
      KREF=J
   20 CØNTINUE
      WRITE (6,2003) CØST,KREF
      IF (KREF.NE.0)  GØ TØ 30
      WRITE (6,2004)
      STØP
C-----IDENTIFY ØUTFALL LEVEL AND DIAMETER
   30 J=(KREF-1)/MEND+1
      M=KREF-(J-1)*MEND
      I=IDD+NBRAN-1
      DMA=GA(I)
      I=IDL+NBRAN-1
      ZMAX=GA(I)
      Z=ZMAX-DELTA*FLØAT(M-1)
      CALL SIZED (PDA,NPIPE,JJ,DMA)
      JJ=JJ-JEND+J
      D=PDA(NPIPE,JJ)
      WRITE (6,2009) Z,D
C-----SET DØWNSTREAM LEVELS AND DIAMETERS TØ -999999.9
      IG3=IDG
      IG4=IDG+2*NBRAN-1
      DØ 40 I=IG3,IG4
   40 GA(I)=-999999.9
C-----READ TRAIL DATA FØR BRANCH FRØM M.T.FILE INTØ ARRAYS GA AND KE
   41 IG1=IG4-1
      IF (LX) IK1=IDK-1
      NM=0
      LITEM=0
    4 NM=NM+1
    5 BACKSPACE 9
      IG1=IG1+2
      IF (LX) IK1=IK1+1
      IF (IG1.GT.NGA) CALL MESAGE (1,0)
      IF (LX.AND.IK1.GT.NKE)  CALL MESAGE (2,0)
      READ (9)  J,GA(IG1),GA(IG1+1)
      LITEM=LITEM+1
      IF (LX) KE(IK1)=J
      BACKSPACE 9
      IF (J-MJ)  10,10,5
   10 IF (MJ-NM)  11,11,4
   11 IG2=IG1+1
      IG1=IG4+1
      IF (ND.LT.2) GØ TØ 50
      WRITE (6,2002) IG1,IG2
      WRITE (6,2001) (GA(I),I=IG1,IG2)
   50 IF (LX) GØ TØ 140
C-----IDENTIFY UPSTREAM END ØF BRANCH LEVEL AND DIAMETER
      IGL=IG1+2*(MJ-KREF)
      ZUS=GA(IGL)
      DUS=GA(IGL+1)
   52 WRITE (6,2005) NB,ZUS,DUS
C-----SET D/S LEVEL AND DIAMETER FØR UPSTREAM PIPES
C-----IDENTIFY UPSTREAM PIPES
   51 N=ID(NB,2)
      IF (N.EQ.0) GØ TØ 60
      I=ID(NB,5)-1
      DØ 55 J=1,N
      I=I+1
      K=KE(I)
      L1=IG3+K-1
      L2=IG3+K-1+NBRAN
      GA(L1)=ZUS
      GA(L2)=DUS
   55 CØNTINUE
   60 IF (ND.GT.1)         WRITE (6,2001) (GA(I),I=IG3,IG4)
   70 NB=NB-1
      IF (NB.LE.0) RETURN
      I=IG3+NB-1
```

```
C-----IS BRANCH A MINOR MEMBER OF A X/D SET,(OR U/S OF A MINOR MEMBER)?
      IF (GA(I).LT.-999999.0.AND.ID(NB,1).GT.100) GO TO 90
      IF (GA(I).GT.-999999.0) GO TO 75
      DO 71 J=1,MJ
      BACKSPACE 9
   71 CONTINUE
      GO TO 70
C-----FIND D/S LEVEL AND DIAMETER
   75 ZDS=GA(I)
      I=I+NBRAN
      DDS=GA(I)
      NPIPE=MOD(ID(NB,1),10)
C-----FIND CORRESPONDENCE FROM D/S PIPE TO CURRENT PIPE
      I=IDL+NB-1
      ZMAX=GA(I)
      I=IDD+NB-1
      DMA=GA(I)
      CALL SIZED (PDA,NPIPE,J,DMA)
      NN=IDJ+MJ*(NB-1)
      CALL BRIDGE (GA,NGA,ZDS,DDS,ZMAX,J,NPIPE,NN,M1,J1)
      KREF=(J1-1)*MEND+M1
C-----IS BRANCH OTHER THAN THE MAIN MEMBER OF A CROSSDRAIN SET?
      IF (ID(NB,1).LT.101) GO TO 41
      NXEND=IDX(NSET,7)
      IF (IDK+NXEND*MJ.GE.NK)  CALL MESAGE (12,9)
C-----READ IN X/D SET TRACE DATA INTO KE,AND MAX LEVEL AND DIAM U/S OF SET
   80 REWIND 8
      KITEM =KITEM-NXEND*MJ-1
      IF (KITEM.EQ.0) GO TO 100
      DO 90 I=1,KITEM
   90 READ (8)
  100 J=NXEND*MJ
      I=IDK-1
      DO 110 K=1,J
      I=I+1
  110 READ (8) KE(I)
      READ (8) ZMAX,DMA
      NK=NK-1
      KE(NK)=NXEND
      IF (ND.GT.0)  WRITE (6,2006) NSET,NXEND,KREF
C-----IDENTIFY UPSTREAM REFERENCE
      I=IDK+(NXEND-1)*MJ+KREF-1
  120 KREF=KE(I)
      NXD=(KREF-1)/MJ
      KREF=KREF-MJ*NXD
      IF (ND.GT.0)   WRITE (6,2006) NSET,NXD,KREF
      KNMM=IDX(NSET,8)+NXD-1
      NMM=KE(KNMM)
      I=IDX(NSET,1)
      KABS=ID(I,7)+NMM-1
      IF (NXD.GT.0)  WRITE (6,2006) GA(KABS)
      NK=NK-1
      KE(NK)=NXD
      IF (NXD.EQ.0) GO TO 130
      I=IDK+(NXD-1)*MJ+KREF-1
      GO TO 120
C-----IDENTIFY LEVEL AND DIAMETER AT UPSTREAM END OF CROSSDRAIN SET
  130 J=(KREF-1)/MEND+1
      M=KREF-(J-1)*MEND
      ZUS=ZMAX-DELTA*FLOAT(M-1)
      CALL SIZED (PDA,NPIPE,JI,DMA)
      JJ=JI-JEND+J
      DUS=PDA(NPIPE,JJ)
      NSET=NSET-1
      GO TO 52
C-----FIND END MANHOLE NUMBER LEVEL AND DIAMETER
  140 J=(KREF-1)/MEND+1
      M=KREF-MEND*(J-1)
      N=ID(NB,3)
      I=IDL+NB-1
      ZEND=GA(I)-DELTA*FLOAT(M-1)
      I=IDD+NB-1
      CALL SIZED (PDA,NPIPE,JI,GA(I))
      JJ=JI-JEND+J
      DEND=PDA(NPIPE,JJ)
      WRITE(6,2007) N,J,M,DEND,ZEND
      GA(IG)=DEND
      KE(IK)=N
      IG=IG-1
      IK=IK-1
```

```
C-----FIND STARTS OF TRACE BACK SEQUENCES FOR D/S STATE (M,J)
      NA=IK1+1
      NJ=1
      DO 150 I=1,LITEM
      NA=NA-1
      IF (NJ.EQ.KREF) GO TO 160
      IF (KE(NA).GT.MJ) GO TO 150
      NJ=NJ+1
  150 CONTINUE
  160 NC=IG2-2*I+1
  170 N=(KE(NA)-1)/M3+1
      J=(KE(NA)-(N-1)*MJ-1)/MEND+1
      M=KE(NA)-(N-1)*MJ-MEND*(J-1)
      ZUS=GA(NC)
      DUS=GA(NC+1)
C-----WRITE OUT M/H POSITIONS DIAMS AND LEVELS: STORE M/HS AND DIAMS.
      WRITE (6,2007) N,J,M,DUS,ZUS
      IF (IG2.GT.IG)       CALL MESAGE (13,0)
      IF (IK1.GT.IK)       CALL MESAGE (14,0)
      KE(IK)=N
      GA(IG)=DUS
      IK=IK-1
      IG=IG-1
      IF (N.LE.1) GO TO 52
      NA=NA-1
      NC=NC-2
      GO TO 170
 2000 FORMAT (1H0/////1X,13HENTERED TRACE)
 2001 FORMAT (1X,F11.3,9F12.3)
 2002 FORMAT (1X,51HTRAIL DATA:  LEVELS AND DIAMETERS STORED IN GA FROM,
     1I6,4H TO ,I6)
 2003 FORMAT (1H0/1X,23HCHEAPEST SOLUTION COSTS,F12.3,5X,10HEND STATE=,
     1I6)
 2004 FORMAT (1X,20HNO FEASIBLE SOLUTION)
 2005 FORMAT (1X,6HBRANCH,I6,5X,15HUPSTREAM LEVEL=,F9.3,5X,
     118HUPSTREAM DIAMETER=,F6.3)
 2006 FORMAT (1X,14HCROSSDRAIN SET,I6,3X,17HCROSSDRAIN NUMBER,I5,
     13X,5HSTATE,I6)
 2007 FORMAT (1X,7HMANHOLE,I6,9H DIAM.NO.,I6,10H LEVEL NO.,I6,
     19H DIAMETER,F6.3,6H LEVEL,F9.3)
 2008 FORMAT (1X,26HNEW CROSSDRAIN AT CHAINAGE,F16.3)
 2009 FORMAT (1X,14HOUTFALL LEVEL=,F12.3,19H  OUTFALL DIAMETER=,F12.3)
      END

END OF SEGMENT, LENGTH  1105, NAME  TRACE
```

```
      SUBROUTINE OUTPUT (KE,GA,NKE,NGA)
      DIMENSION KE(NKE),GA(NGA),GL(5),OFFSET(5)
      COMMON/DATA/NSET,NBRAN,NB,NPIPE,NXDSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDO,IDJ,
     1IDP,IDR,IDM,IDB
      WRITE (6,2000)
C-----WRITE INTRODUCTORY BASIC DATA TO FORMATTED FILE
      DO 10 I=1,9
      IF (PDA(I,6).LT.0.001) GO TO 11
   10 CONTINUE
      I=10
   11 N=I-1
      WRITE (4,1000) N
      DO 13 I=1,N
      DO 12 J=1,20
      IF (J.EQ.2.OR.J.GE.6) PDA(I,J)=1000.0*PDA(I,J)
      WRITE (4,1001) PDA(I,J)
      IF (J.GE.6.AND.PDA(I,J).LT.0.001) GO TO 13
   12 CONTINUE
   13 CONTINUE
C-----HOW MANY BRANCHES IN NEW NETWORK?
      NSET=1
      K=0
      DO 40 I=NK,NKE
      IF (KE(I).EQ.IDX(NSET,7)) GO TO 20
      IF (KE(I).NE.0) K=K+2*IDX(NSET,6)-1
      GO TO 40
   20 K=K+IDX(NSET,6)-1
      NSET=NSET+1
   40 CONTINUE
      NBRA=NBRAN+K
      WRITE (4,1000) NBRA
      NSET=0
      NB=0
      NR=0
C-----MOVE TO NEXT BRANCH IN ORIGINAL NETWORK
   50 NB=NB+1
C-----ANY MORE BRANCHES?
      IF (NB.GT.NBRAN) GO TO  300
C-----MEMBER OF A CROSSDRAIN SET?
      IF (ID(NB,1).GT.100) GO TO 100
      NR=NR+1
C-----WRITE DETAILS OF THIS BRANCH TO FORMATTED FILE
      CALL DETAIL (KE,GA,NKE,NGA,NB,1,ID(NB,3),X,Y,AX,NR)
C-----SORT OUT UPSTREAM PIPES
      CALL UPBRAN (KE,NKE,NB)
C-----STORE NEW NUMBER FOR THIS BRANCH
      I=IDK+NB-1
      KE(I)=NR
      GO TO 50
C-----BRANCH IS A MEMBER OF A CROSS DRAIN SET
  100 NSET=NSET+1
      NRUN=IDX(NSET,6)
      NUX=0
      I=NK
  120 I=I+1
C-----FIND NUMBER OF DOWNSTREAM CROSSDRAIN
      NDX=KE(I)
C-----FOR EACH CROSSDRAIN SET MEMBER
      DO 200 N=1,NRUN
C-----FIND UPSTREAM AND DOWNSTREAM  MANHOLE NUMBERS
      IF (NUX.GT.0) GO TO 130
      MH1=1
      GO TO 140
  130 J=IDX(NSET,7+N)+NUX-1
      MH1=KE(J)
  140 J=IDX(NSET,7+N)+NDX-1
      MH2=KE(J)
C-----FIND OLD BRANCH NUMBER
      NUMB=IDX(NSET,N)
C-----WRITE DETAILS OF NEW BRANCH TO FILE
      NR=NR+1
      CALL DETAIL (KE,GA,NKE,NGA,NUMB,MH1,MH2,GL(N),OFFSET(N),AX,NR)
C-----SORT OUT UPSTREAM BRANCH DETAILS
      IF (MH1.EQ.1) GO TO 160
      IF (N.GT.1)   GO TO 150
      KK=2
      WRITE (6,1002) KK
      K=NR-2*NRUN+1
      WRITE (4,1000) K
      WRITE (6,1003) K
      K=NR-1
      WRITE (4,1000) K
      WRITE (6,1003) K
  150 K=0
      WRITE (4,1000) K
      IF (N.GT.1) WRITE (6,1002) K
      GO TO 200
```

```
C-----BRANCH IS UPSTREAM END OF X/D SET MEMBER: FIND U/S BRANCHES
  160 CALL UPBRAN (KE,NKE,NUMB)
  200 CONTINUE
C-----SORT OUT DETAILS OF CROSSDRAINS
      N=NRUN-1
      K=NRUN+1
      DO 250 J=1,N
      NR=NR+1
      NPIPE=11
      WRITE (4,1000) NPIPE
      WRITE (6,1004) NR,NPIPE
      K=K-1
      X=ABS(OFFSET(K)-OFFSET(K-1))
      WRITE (4,1001) X
      WRITE (6,1005) AX,OFFSET(K)
      WRITE (6,1006) AX,OFFSET(K-1)
      Y=0.0
      WRITE (4,1001) Y,Y,AX,Y,GL(K),Y,GL(K-1),X
      IF (J.EQ.1)   GO TO 230
      KK=2
      WRITE (6,1002) KK
      L=NR-2*J+1
      WRITE  (4,1000)   L
      WRITE (6,1003) L
  230 L=NR-1
      WRITE (4,1000)   L
      IF (J.EQ.1) WRITE (6,1002) J
      WRITE (6,1003) L
      L=0
      WRITE  (4,1000)   L
  250 CONTINUE
      NUX=NDX
      IF (NUX.LT.IDX(NSET,7)) GO TO 120
C-----SORT OUT NEW BRANCH NUMBER FOR CROSSDRAIN SET MAIN MEMBER
      NB1=NR-2*NRUN+2
      KOLD=IDX(NSET,1)
      KNEW=10000*NB1+NR
      J=IDK+KOLD-1
      KE(J)=KNEW
C-----UPDATE BRANCH COUNTER
      NB=NB+NRUN-1
      NK=I+1
      GO TO 50
  300 I=0
      WRITE (4,1000) I
      RETURN
 1000 FORMAT (1X,I10)
 1001 FORMAT (1X,F15.6)
 2000 FORMAT (1H0//20X,29H*****NEW NETWORK CREATED*****)
 1002 FORMAT (15X,I6,19H UPSTREAM BRANCHES)
 1003 FORMAT (15X,I12)
 1004 FORMAT (5X,6HBRANCH,I6,6H  TYPE,I6)
 1005 FORMAT (15X,20HUPSTREAM CHAINAGE  =,F16.3,5X,7HOFFSET=,F12.3)
 1006 FORMAT (15X,20HDOWNSTREAM CHAINAGE=,F16.3,5X,7HOFFSET=,F12.3)
      END

END OF SEGMENT, LENGTH   823, NAME  OUTPUT
```

```
      SUBROUTINE DETAIL (KE,GA,NKE,NGA,NB,MH1,MH2,GL,OFFSET,AX,NR)
      DIMENSION KE(NKE),GA(NGA)
      COMMON/WHERE/ID(200,11),IOX(10,12),IDA,IDL,IDT,IDK,IDS,IDD,IDJ,
     1IDP,IDR,IDM,IDB
C-----TYPE OF BRANCH
      I=MOD(ID(NB,1),100)
      WRITE (4,1000) I
      WRITE (6,1002) NR,I
C-----LENGTH OF BRANCH
      I=ID(NB,8)+MH1-1
      X1=GA(I)
      J=ID(NB,8)+MH2-1
      X2=GA(J)
      DIST=X2-X1
      WRITE (4,1001) DIST
C-----CATCHMENT WIDTH (ONLY WORKS FOR CONSTANT WIDTH)
      I=ID(NB,8)+1
      J=ID(NB,9)+1
      WIDTH=GA(J)/GA(I)
      WRITE (4,1001) WIDTH
C-----OFFSET
      I=ID(NB,6)
      OFFSET=GA(I)
      WRITE (4,1001) OFFSET
C-----ABSOLUTE CHAINAGE
      I=ID(NB,7)+MH1-1
      AX=GA(I)
      WRITE (4,1001) AX
      WRITE (6,1003) AX,OFFSET
      II=ID(NB,7)+MH2-1
      AX=GA(II)
      WRITE (6,1004) AX,OFFSET
C-----DIRECTION INDICATOR
      IND=1
      J=I+1
      IF (GA(J).LT.GA(I))              IND=-1
      IF (ABS(GA(J)-GA(I)).LT.0.001) IND=0
      WRITE (4,1000) IND
C-----GROUND LEVEL DATA
      I=ID(NB,10)-1
      J=ID(NB,11)-1
      K=ID(NB,4)
      DO 60 N=1,K
      I=I+1
      J=J+1
      IF (GA(J).LT.X1-0.001) GO TO 60
      IF (GA(J).GT.X2+0.001) GO TO 70
      X=GA(J)-X1
      WRITE (4,1001) GA(I),X
      GL=GA(I)
   60 CONTINUE
   70 RETURN
 1000 FORMAT (1X,I10)
 1001 FORMAT (1X,F15.6)
 1002 FORMAT (5X,6HBRANCH,I6,6H  TYPE,I6)
 1003 FORMAT (15X,20HUPSTREAM CHAINAGE  =,F16.3,5X,7HOFFSET=,F12.3)
 1004 FORMAT (15X,20HDOWNSTREAM CHAINAGE=,F16.3,5X,7HOFFSET=,F12.3)
      END

END OF SEGMENT, LENGTH   323, NAME  DETAIL
```

```
      SUBROUTINE UPBRAN (KE,NKE,NUMB)
      DIMENSION KE(NKE)
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDR
      NUB=ID(NUMB,2)
      WRITE (6,1001) NUB
      IF (NUB.EQ.0) GO TO 30
      J=ID(NUMB,5)-1
      DO 20 K=1,NUB
      J=J+1
C-----OLD U/S BRANCH NUMBER
      KOLD=KE(J)
C-----FIND CORRESPONDING NEW BRANCH NUMBER(S)
      KI=IDK+KOLD-1
      KNEW=KE(KI)
      IF (KNEW.GT.10000) GO TO 10
      WRITE (4,1000) KNEW
      WRITE (6,1002) KNEW
      GO TO 20
   10 KNE=KNEW/10000
      WRITE (4,1000) KNE
      WRITE (6,1002) KNE
      KNEW=KNEW-KNE*10000
      WRITE (4,1000) KNEW
      WRITE (6,1002) KNEW
   20 CONTINUE
   30 KNEW=0
      WRITE (4,1000) KNEW
      RETURN
 1000 FORMAT (1X,I10)
 1001 FORMAT (15X,I6,18H UPSTREAM BRANCHES)
 1002 FORMAT (15X,I12)
      END
```

END OF SEGMENT, LENGTH   128, NAME UPBRAN

```
      SUBROUTINE MESAGE (M,N)
      WRITE (6,1000) M,N
      STOP
 1000 FORMAT (40X,17H*****ERROR NUMBER,I6,12H  *****      ,I6)
      END
```

END OF SEGMENT, LENGTH   18, NAME MESAGE

```
      SUBROUTINE LEVELS (KE,GA,NKE,NGA,MG,MK)
      DIMENSION KE(NKE),GA(NGA),TDS(200),ZD(200)
      COMMON/DATA/NSET,NRRAN,NS,NPIPE,NKDSET,NITEM,KITEM,NK
      COMMON/PARAM/PDA(9,20),MEND,JEND,MJ,ND,T,RP,ZD,DMAX,DMIN,DELTA
      COMMON/WHERE/ID(200,11),IDX(10,12),IDA,IDL,IDT,IDK,IDG,IDD,IDJ,
     1IDP,IDR,IDM,IDB
      WRITE (6,1002)
      COST=0.0
      LO=0
   10 LO=LO+1
      NPIPE=MOD(ID(LO,1),10)
      CMIN=PDA(NPIPE,1)
      RK  =PDA(NPIPE,2)
      VMIN=PDA(NPIPE,3)
      VMAX=PDA(NPIPE,4)
      NEND=ID(LO,3)
      NUS=1
C---IDENTIFY THE UPSTREAM BRANCHES AND HENCE U/S TIME OF FLOW AND LEVEL
      NBR=ID(LO,2)
      TUS=T
      DISTUS=0.0
      L1=ID(LO,10)
      GLUS=GA(L1)
      ZUS=GLUS-CMIN
      AREAUS=0.0
      NGL=ID(LO,4)
      IF (NBR.EQ.0) GO TO 40
      K=ID(LO,5)-1
      DO 30 I=1,NBR
      K=K+1
      L=KE(K)
      TUS=AMAX1(TUS,TDS(L))
      ZUS=AMIN1(ZUS,ZD(L))
      J=IDA+L-1
      AREAUS=AREAUS+GA(J)
   30 CONTINUE
C-----DEFINE PIPE DIAMETER,D/S M/H,CATCHMENT AREA
   40 MG=MG+1
      DIAM=GA(MG+1)
      MK=MK+1
      NDS=KE(MK+1)
      NIG=NUS+1
      AREA=AREAUS
      DO 41 NIF=NIG,NDS
      J=ID(LO,9)+NIF-1
   41 AREA=AREA+GA(J)
      J=ID(LO,8)+NDS-1
      DISTDS=GA(J)
      DIST=DISTDS-DISTUS
C-----FIND MINIMUM GRADIENT CONSISTENT WITH MINIMUM VELOCITY
      GMIN=PDA(NPIPE,5)
      CALL VELOC (GMIN,DIAM,RK,V,Q)
      IF (V.GT.VMIN) GO TO 44
      CALL FINDG(DIAM,VMIN,RK,GMIN)
C------FIND REQUIRED GRADIENT
   44 CALL GRADE (DIAM,GMIN,RK,RP,DIST,AREA,TUS,SLOPE,Q,QMAX,VEL)
C------FIND D/S GROUND LEVEL
      L2=L1+NGL
      DO 50 I=L1,L2
      K=I+NGL
      IF (GA(K).GT.DISTDS-0.1) GO TO 60
   50 CONTINUE
   60 GLDS=GA(I)
      L2=I
      ZDS=GLDS-CMIN
C-----IS MIN SLOPE SOLUTION FEASIBLE?
      IF (ZUS-SLOPE*DIST.GT.ZDS+0.001) GO TO 65
      ZDS=ZUS-SLOPE*DIST
      GO TO 70
   65 SLOPE=(ZUS-ZDS)/DIST
      CALL VELOC (SLOPE,DIAM,RK,VEL,QMAX)
      IF (VEL.LT.VMAX+0.0001) GO TO 68
      CALL FINDG (DIAM,VMAX,RK,GMAX)
      ZUS=ZDS+GMAX*DIST
      SLOPE=GMAX
      CALL VELOC(SLOPE,DIAM,RK,VEL,QMAX)
   68 TIME=(TUS+DIST/VEL)/60.0
      CALL RAIN (RP,TIME,RI)
      Q=AREA*RI/3.6E6
```

```
C-----CHECK GROUND COVER EN ROUTE
   70 L3=L1+1
      L4=L2-1
      GAREA=0,0
      IF (L3.GT.L4) GO TO 90
      DZMAX=0,0
      DO 80 I=L3,L4
      K=I+NGL
   80 DZMAX=AMAX1(ZUS-SLOPE*(GA(K)-DISTUS)-GATT)+CMIN,DZMAX)
      ZUS=ZUS-DZMAX
      ZDS=ZDS-DZMAX
C-----CALCULATE AREA OF LONG SECTION ABOVE STRAIGHT LINE
      M1=L1+NGL
      M2=L2+NGL
      GAREA=-GA(M1)*(GA(L3)-GA(L2))*0,5
      DO 100 I=L3,L4
      K=I+NGL
  100 GAREA=GAREA-GA(K)*(GA(I+1)-GA(I-1))*0,5
      GAREA=GAREA-GA(M2)*(GA(L1)-GA(L4))*0,5
   90 CALL SIZED (PDA,NPIPE,JPIPE,DIAM)
      CALL COSTIT (JPIPE-5,GAREA,GLUS-ZUS,GLDS-ZDS,DIST,C)
      COST=COST+C
      WRITE (6,1001) LO,DIST,DIAM,SLOPE,ZUS,ZDS,GLUS,GLDS,AREA,GAREA,Q,
     1QMAX,VEL,C,COST
      TUS=TUS+DIST/VEL
      ZUS=ZDS
      GLUS=GLDS
      AREAUS=AREA
      DISTUS=DISTDS
      L1=L2
      NUS=NDS
      IF(NDS.LT.NEND) GO TO 40
      MG=MG+1
      MK=MK+1
      TDS(LO)=TUS
      ZD(LO)=ZUS
      IF (LO.LT.NBRAN) GO TO 10
      RETURN
 1001 FORMAT (1H0,I6,F9.3,F7.3,F7.4,4F8.3,F8.0,F8.3,2F7.4,F8.3,2F10.1)
 1002 FORMAT (1H1//4X,114HBRANCH LENGTH  DIAM  SLOPE U/S SL  D/S SL  U/S
     1 GL  D/S GL  AREA  GROUND A, FLOW  CAPACITY  VEL,     COST      S
     2UM)
      END

END OF SEGMENT, LENGTH   563, NAME  LEVELS
```

```
      SUBROUTINE GRADE (DIAM,GMIN,RK,RP,DIST,AREA,TUS,SLOPE,Q,QFULL,V)
C-----CALCULATES REQUIRED SLOPE OF A PIPE ACCORDING TO RATIONAL METHOD
      LOGICAL NPLUS,MINUS
      K=0
      NPLUS=.FALSE.
      MINUS=.FALSE.
      SLOPE=GMIN
    5 CALL VELOC(SLOPE,DIAM,RK,V,QFULL)
      TIME=TUS+DIST/V
      CALL RAIN(RP,TIME/60,0,RI)
      Q=AREA*RI/3,6E6
      IF (Q.LT.QFULL.AND.K.EQ.0) RETURN
      SLOPE=FFF(Q,DIAM,RK)
      K=K+1
      IF (K.LE.2) GO TO 5
   10 CALL VELOC(SLOPE,DIAM,RK,V,QFULL)
      TIME=TUS+DIST/V
      CALL RAIN (RP,TIME/60,0,RI)
      Q=AREA*RI/3,6E6
      IF (ABS((Q-QFULL)/QFULL).LT.0,001) RETURN
      IF (Q-QFULL) 30,20,20
   20 IF (MINUS) RETURN
      SLOPE=SLOPE*1,001
      NPLUS=.TRUE.
      GO TO 10
   30 IF (NPLUS) RETURN
      SLOPE=SLOPE*,999
      MINUS=.TRUE.
      GO TO 10
      END

END OF SEGMENT, LENGTH   176, NAME  GRADE
```

```
                   FUNCTION FFF(Q,DIAM,RK)
                   FFF=0,02065*Q*Q/DIAM**5*(ALØG10(RK/3,7/DIAM+4,1365/(Q/DIAM/
                  11,141E-6)**0,89))**(-2)
                   RETURN
                   END

END ØF SEGMENT, LENGTH    51, NAME  FFF
```

```
                   SUBRØUTINE FINDG(DIAM,VV,RK,SLØPE)
                   LØGICAL NPLUS,MINUS
                   Q=0,7854*DIAM*DIAM*VV
                   SLØPE=FFF(Q,DIAM,RK)
                   NPLUS=,FALSE,
                   MINUS=,FALSE,
                10 CALL VELØC (SLØPE,DIAM,RK,V,Q)
                   IF (ABS(VV-V),LT,,001) RETURN
                   IF (VV-V)  20,20,30
                20 IF (NPLUS)  RETURN
                   SLØPE=SLØPE*0,999
                   MINUS=,TRUE,
                   GØ TØ 10
                30 IF (MINUS) RETURN
                   SLØPE=SLØPE*1,001
                   NPLUS=,TRUE,
                   GØ TØ 10
                   END

END ØF SEGMENT, LENGTH    80, NAME  FINDG
```