THE SYNTHESIS OF HEAT EXCHANGER NETWORKS SUBJECT TO VARIABLE SETS OF CONSTRAINTS

by

KEITH WILLIAM HANSON, BSc (Eng), A.C.G.I.

A Thesis submitted for the Degree of
Doctor of Philosophy of the
University of London
and for the
Diploma of Imperial College

Department of Chemical Engineering and Chemical Technology
Imperial College of Science and Technology
London

September 1984

THIS THESIS CONTAINS
MICROFILMS THAT COULD
NOT BE DIGITISED.

PLEASE CONTACT THE
UNIVERSITY IF YOU WISH TO
SEE THIS MATERIAL.

ABSTRACT OF THESIS

The thesis develops the basic theory of heat exchanger network synthesis and continues the philosophy, introduced by previous workers, of establishing design aids to be used during network synthesis. Procedures are developed which provide sets of possible solutions to problems which are encountered at various stages of the synthesis. The methods used have been chosen to allow process constraints to be specified in addition to those imposed by the laws of thermodynamics. In particular, the thesis addresses two important problems which have not been studied extensively before.

Most importantly, the splitting of process streams into parallel branches, disliked because of the extra control problems it entails, is sometimes essential if high energy recovery is to be obtained. Methods exist which indicate the need for stream splits but not for their effective introduction. An algorithm is developed which considers the regions where stream splits are most likely to be required, the pinch regions, and determines the complete set of exchanger configurations with the minimum number of stream branches. The algorithm is based on the formulation of the problem as a type of transportation problem from linear programming.

Secondly, the use of heat exchangers that exhibit pure counter-current flow has been assumed by nearly every previous worker and the use of other heat exchanger configurations has not been studied extensively. A branch-and-bound tree-search method is developed which synthesizes networks away from problem pinch points and which admits the use of non-counter-current heat exchangers. An implicit enumeration technique is used to reduce the search space size. This technique incorporates a method of updating the required bound on energy recovery from one level of the search tree to another.

Both algorithms are applied to several problems of academic and industrial importance.

ACKNOWLEDGEMENTS

I would like to thank Dr. A.R.H. Cornish, my supervisor during the period of this research, for his help and guidance.

I am endebted to the Science and Engineering Research Council for providing the bulk of my financial support during the last three years and to British Petroleum International plc for a supplementary bursary.

The extensive literature survey I have carried out would not have been possible without the assistance of the staff of the following libraries, to whom I am grateful:

(i) Chemical Engineering Library, Imperial College, London

(ii) Lyon Playfair Library, Imperial College, London

(iii) Science Museum Library, London

(iv) British Library, Boston Spa, Yorkshire

(v) W. Sussex County Library, Crawley, W. Sussex

(vi) Science Reference Library, London

TABLE OF CONTENTS PAGE

CHAPTER 1 INTRODUCTION TO HEAT EXCHANGER NETWORK SYNTHESIS

1.1 An overview of the past and present work on the subject 31

1.2 A formal description of the synthesis and detailed design of heat exchanger networks 33

1.3 The pinch point and composite curves 36

1.4 An outline of the following chapters 38

CHAPTER 2 HEAT EXCHANGER NETWORK SYNTHESIS LITERATURE REVIEW

2.1 Introduction 40

2.2 Pre-synthesis analysis 40

2.2.1 Minimum utility consumption target 41

2.2.2 Minimum number of matches target 44

2.2.2.1 Minimum number of matches 45

2.2.2.2 Minimum number of heat exchangers 46

2.2.3 Minimum area target 47

2.2.4 Conclusions on pre-synthesis analysis 47

2.3 Heat exchanger network synthesis 48

2.3.1 Mathematical methods 49

2.3.1.1 The adaptive random search method 50

2.3.1.2 Linear programming synthesis methods 50

2.3.1.2.1 Separable programming 51

2.3.1.2.2 The assignment problem 52

2.3.1.2.3 The transportation problem 54

2.3.1.2.4 The trans-shipment problem 55

2.3.1.3 Structural parameter optimisation methods 56

2.3.1.4 Branch-and-bound methods 58

2.3.1.5 The optimal control method 64

		PAGE
2.3.2	Heuristic methods	65
2.3.2.1	The heuristic structuring method	65
2.3.2.2	Temperature interval methods	66
2.3.2.3	Evolutionary methods	68
2.3.2.4	Rule-based methods	69
2.3.3	Thermodynamic methods	71
2.3.3.1	Exergy methods	71
2.3.3.2	General thermodynamic argument methods	72
2.3.3.3	The pinch design method	73
2.3.4	Graphical methods	73
2.3.4.1	Heat content diagram methods	74
2.3.4.2	Heat availability diagram methods	78
2.3.5	Summary	82
2.3.6	Conclusions on heat exchanger network synthesis	83
2.4	Post-synthesis network evolution	84
2.4.1	Structured evolutionary methods	85
2.4.2	Conclusions on post-synthesis evolution	89
2.5	Summary of conclusions	90

CHAPTER 3 INCORPORATION OF NON-COUNTER-CURRENT HEAT EXCHANGERS INTO A SYNTHESIS PROCEDURE

3.1	Introduction to the work in the following chapters	91
3.2	Introduction to non-counter-current heat exchangers	92
3.3	Counter-current heat exchangers	93
3.4	Mean temperature differences in heat exchangers	94
3.5	Introduction to the use of LMTD correction factors in synthesis	96
3.6	Dimensionless numbers	97

		PAGE
3.7	The synthesis (or design) of heat exchangers	98
3.8	The simulation of heat exchangers	99
3.9	Derivation of the LMTD correction factor	100
3.9.1	Single shell-pass exchangers	101
3.9.2	Multiple shell-pass exchangers	101
3.10	Heat exchangers with fixed LMTD correction factors	102
3.11	Bounding values on LMTD correction factors	102
3.12	Introduction of the temperature gap ratio into the bounding values on LMTD correction factors	104
3.13	Numerical results	105
3.14	The relationship between the temperature gap ratio and the lower bound on LMTD correction factors	107
3.15	Heat exchangers with a fixed temperature gap ratio	107
3.16	Fixed-area exchangers	109
3.17	Conclusions	110

CHAPTER 4 DEPTH-FIRST TREE-SEARCHING METHODS FOR THE SOLUTION OF INLP's, MINLP's AND MILP's

4.1	Introduction	111
4.2	Tree-searching using depth-first branch-and-bound methods	111
4.3	The upper bound on the objective function	113
4.4	The lower bound on additional 'cost'	114
4.5	The branching strategy	115
4.6	The stopping criteria	116
4.7	The heuristic to determine the value of the continuous variables	116
4.8	Application of tree-searching methods	117

CHAPTER 5 THE AUTOMATIC GENERATION OF ESSENTIAL PARALLEL STREAM SPLITS IN MAXIMUM ENERGY RECOVERY HEAT EXCHANGER NETWORKS

		PAGE
5.1	Introduction	118
5.2	A formal description of the problem	119
5.3	The occurrence of bypasses	120
5.4	The defining equations	121
5.5	Conditions for maximum energy recovery	122
5.5.1	Above the pinch point	122
5.5.2	Below the pinch point	123
5.6	An illustrative example	123
5.7	A pictorial representation	127
5.8	Reduction of the problem	130
5.9	Significance of the bottom row	131
5.10	A mathematical formulation	133
5.11	The problem viewed as a mixed integer linear programme	135
5.12	The problem viewed as a transportation problem	136
5.13	A new solution algorithm	137
5.13.1	Introduction to the new solution algorithm	137
5.13.2	The bounding procedure	139
5.13.2.1	The row bound	139
5.13.2.2	The column bound	140
5.13.2.3	A tighter column bound	141
5.13.3	The branching strategy	142
5.13.4	Unique solutions	143
5.13.4.1	Problem solutions with only one permutation of the disjoint elements	144
5.13.4.2	'Strings' of non-zero elements in the LP tableau	145
5.13.4.3	An amended procedure to decrease the solution permutations	147

		PAGE
5.13.5	Initial value of the upper bound on the objective function	148
5.14	Solution to the illustrative example	149
5.15	Comparison between MPOS and the new method	149
5.16	Conclusions	149
5.17	Possible future developments	150

CHAPTER 6 A SET OF ALGORITHMS TO SYNTHESIZE HEAT EXCHANGER NETWORKS

6.1	Introduction	153
6.2	Heat transfer across the pinch point	155
6.3	Introduction to the heuristic for setting the continuous variables	155
6.3.1	The match generator	156
6.3.2	Options allowed in the match generator	158
6.4	An introduction to the bounding algorithm	160
6.4.1	Stream match types	161
6.4.2	Outline of the procedure	163
6.4.3	Stream types	163
6.4.4	Transfer of match information	165
6.4.5	Matched stream status	166
6.4.6	A procedure to locate the first new delimiter	171
6.4.7	Treatment of subsequent delimiters and heat contents	175
6.5	The branching strategy	179
6.5.1	Determination of the ordered lists for the branching strategy	180
6.5.2	Unique solutions	180
6.6	Initial value of the upper bound on the objective function	181
6.7	Summary	182

CHAPTER 7 THE APPLICATION OF THE HEAT EXCHANGER NETWORK SYNTHESIS ALGORITHMS TO SOME PRACTICAL PROBLEMS — PAGE

7.1 Introduction 183
7.2 Problem One 184
7.3 Problem Two 189
7.4 Problem Three 194
7.5 Problem Four 199
7.6 Problem Five 202
7.7 Problem Six 206
7.8 Problem Seven 211
7.9 The cut-off temperature range 216
7.10 The interaction between the branching strategy and the procedure to ensure one-to-one correspondence of solutions and networks 216
7.11 Suitable values for the cut-off temperature range 219
7.12 Efficient use of HERNE 219
7.13 The effectiveness of the matching strategy 219
7.14 Summary of results for the literature problems 221
7.15 A new unpinched problem 223
7.16 A new pinched problem 229
7.17 Summary of results for the new problems 235

CHAPTER 8 CONCLUSIONS AND FUTURE WORK

8.1 Conclusions 237
8.2 Future work 239

APPENDICES

		PAGE
A	The heat exchanger network synthesis problem formulated as an MINLP	240
B	An expression for the maximum value of temperature efficiency given the heat capacity flowrate ratio	241
C	The minimum value for 'P/Pmax' for fixed values of F	241
D	The graphs plotting 'P/Pmax' against R suggesting monotonicity	244
E	A program to calculate temperature efficiency (P) given the LMTD correction factor (F) and heat capacity flowrate ratio (R) (including microfiche)	247
F	A program to calculate temperature gap ratio (G) given the LMTD correction factor (F) and heat capacity flowrate ratio (R) (including microfiche)	247
G	The minimum number of branches versus the minimum number of exchangers	248
H	The problem formulation BBMIP is designed to solve	249
I	The test problem data and solutions	251
J	The fixed-cost transportation problem and loops in the LP tableau	252
K	COMPASS - a computer program to generate heat exchanger configurations with the minimum number of parallel stream splits (including microfiche)	254
L	An example of the application of the heat exchanger network synthesis bounding algorithm	256
M	The basic problem data	274
N	The effect of cut-off temperature range on the problem-tree searches	278

PAGE

O Reproduction of a paper given at the First UK National Heat Transfer Conference, Leeds, July 1984 281

References 302

PAGE

LIST OF FIGURES

CHAPTER 1

1.1 A temperature/relative enthalpy graph with composite hot and cold streams. 37

1.2 Maximum energy recovery resulting in complete satisfaction of a stream. 37

1.3 Maximum energy recovery resulting in a pinch point. 37

1.4 An unpinched problem. 39

1.5 A threshold problem. 39

CHAPTER 2

2.1 A branch-and-bound problem tree. 59

2.2 A standard heat content diagram. 77

2.3 Nishimura's heat content diagram. 77

2.4 Heat availability diagrams. 81

2.5 Heat load 'loops'. 88

CHAPTERS 3 AND 4

None.

PAGE

CHAPTER 5

5.1 A pinch point of the illustrative example. 126

5.2 A pictorial representation for an above-pinch-point design. 129

5.3 An improved pictorial representation. 132

5.4 Movement of 'slack' using row splits. 132

5.5 A tableau fragment. 146

5.6 Solution one to the illustrative example. 151

5.7 Solution two to the illustrative example. 151

5.8 Solution three to the illustrative example. 152

5.9 A pinch point of the reduced problem. 152

CHAPTER 6

6.1 The partitioning of a heat exchanger at the points where heat capacity flowrate changes. 159

6.2 The transformation of bounding information from one level of the search tree to the next. 162

6.3 The four amendment phases of the data associated with a process stream exhibiting a temperature range. 168

PAGE

6.4 The three amendment phases of the data associated with a stream not exhibiting a temperature range. 168

6.5 The three amendment phases of the data associated with a utility stream exhibiting a temperature range. 170

6.6 A procedure to locate the first new delimiter. 172

6.7 No streams are left in the interval diagram. 173

6.8 An old delimiter becomes the new top delimiter. 173

6.9 The new hot stream end becomes the new top delimiter. 173

6.10 The old hot steam end becomes the new top delimiter. 174

6.11 The new cold stream end becomes the new top delimiter. 174

6.12 The old cold stream end becomes the new top delimiter. 174

6.13 The treatment of subsequent delimiters and heat contents. 176

6.14 A new stream end comes before the next old delimiter and marks the end of the current new interval. 177

PAGE

6.15 The next old delimiter marks the end of the current new interval. 177

6.16 The next old delimiter does not mark the end of the current new interval because a change in the heat capacity flowrate of a matched stream is associated with it. 178

6.17 The next old delimiter does not mark the end of the current interval because a matched stream old end is associated with it. 178

CHAPTER 7

7.1 A near-optimal solution to 4SP2 using counter-current exchangers and no stream splits. 186

7.2 A near-optimal solution to 4SP2 using counter-current exchangers and a three-way stream split. 186

7.3 The solution given in figure 7.1 using 1/2 exchangers. 186

7.4 The solution given in figure 7.2 using 1/2 exchangers. 187

7.5 The first ten-shell solution to 4SP2. 187

7.6 The second ten-shell solution to 4SP2. 187

7.7 The first sample solution to 4SP2 with paired exchangers. 188

PAGE

7.8 The second sample solution to 4SP2 with paired exchangers. 188

7.9 The Zachoval and Konecny solution to 4SP2. 188

7.10 The optimal solution to 5SP1 using counter-current exchangers. 190

7.11 The solution given in figure 7.10 using 1/2 exchangers. 190

7.12 The first eleven-shell sample solution to 5SP1. 190

7.13 The second eleven-shell sample solution to 5SP1. 191

7.14 The third eleven-shell sample solution to 5SP1. 191

7.15 The fourth eleven-shell sample solution to 5SP1. 191

7.16 The fifth eleven-shell sample solution to 5SP1. 192

7.17 The sixth eleven-shell sample solution to 5SP1. 192

7.18 The seventh eleven-shell sample solution to 5SP1. 193

7.19 The eighth eleven-shell sample solution to 5SP1. 193

PAGE

7.20 The ninth eleven-shell sample solution to 5SP1. 193

7.21 The optimal solution to 7SP2 using counter-current exchangers. 195

7.22 The solution given in figure 7.21 using 1/2 exchangers. 195

7.23 The first eleven-shell sample solution to 7SP2. 195

7.24 The second eleven-shell sample solution to 7SP2. 196

7.25 The third eleven-shell sample solution to 7SP2. 196

7.26 The fourth eleven-shell sample solution to 7SP2. 196

7.27 The fifth eleven-shell sample solution to 7SP2. 197

7.28 The sixth eleven-shell sample solution to 7SP2. 197

7.29 The seventh eleven-shell sample solution to 7SP2. 198

7.30 The eighth eleven-shell sample solution to 7SP2. 198

7.31 The optimal solution to 4SP1 using counter-current exchangers. 200

PAGE

7.32 The solution given in figure 7.31 using 1/2 exchangers. 200

7.33 The second nine-shell solution to 4SP1. 201

7.34 The third nine-shell solution to 4SP1. 201

7.35 The optimal solution to TC1 using counter-current exchangers. 203

7.36 The solution given in figure 7.35 using 1/2 exchangers. 203

7.37 The first seventeen-shell solution to TC1. 203

7.38 The second seventeen-shell solution to TC1. 204

7.39 The third seventeen-shell solution to TC1. 204

7.40 The fourth seventeen-shell solution to TC1. 204

7.41 The fifth seventeen-shell solution to TC1. 205

7.42 A twelve-shell solution to TC1. 205

7.43 A near-optimal solution to the six-stream problem using counter-current exchangers. 207

7.44 The solution given in figure 7.43 using 1/2 exchangers. 207

7.45 The first eight-shell above-pinch solution to the six-stream problem. 207

PAGE

7.46 The second eight-shell above-pinch solution to the six-stream problem. 208

7.47 The third eight-shell above-pinch solution to the six-stream problem. 208

7.48 The fourth eight-shell above-pinch solution to the six-stream problem. 208

7.49 The first eight-shell below-pinch solution to the six-stream problem. 209

7.50 The second eight-shell below-pinch solution to the six-stream problem. 209

7.51 The third eight-shell below-pinch solution to the six-stream problem. 210

7.52 The fourth eight-shell below-pinch solution to the six-stream problem. 210

7.53 A near-optimal solution to 7SP4 using counter-current exchangers. 212

7.54 The solution given in figure 7.53 using 1/2 exchangers. 212

7.55 The first eleven-shell above-pinch solution to 7SP4. 212

7.56 The second eleven-shell above-pinch solution to 7SP4. 213

7.57 The third eleven-shell above-pinch solution to 7SP4. 213

PAGE

7.58 The fourth eleven-shell above-pinch solution to 7SP4. 213

7.59 The first eleven-shell below-pinch solution to 7SP4. 214

7.60 The second eleven-shell below-pinch solution to 7SP4. 214

7.61 The third eleven-shell below-pinch solution to 7SP4. 215

7.62 The fourth eleven-shell below-pinch solution to 7SP4. 215

7.63 Two exchangers which show a possible clash in the requirements of two of the tree-search procedures. 218

7.64 The first grid diagram for the new unpinched problem. 226

7.65 The second grid diagram for the new unpinched problem. 226

7.66 A graph of the heat cascade versus temperature for the new unpinched problem. 227

7.67 The third grid diagram for the new unpinched problem. 228

7.68 A fourteen-shell solution to the new unpinched problem. 228

PAGE

7.69 The first grid diagram for the above-pinch section of the new pinched problem. 232

7.70 The second grid diagram for the above-pinch section of the new pinched problem. 232

7.71 The grid diagram for the below-pinch section of the new pinched problem. 232

7.72 A solution to the above-pinch section of the new pinched problem. 233

7.73 The first partial solution to the below-pinch section of the new pinched problem. 233

7.74 The second partial solution to the below pinch section of the new pinched problem. 233

7.75 A sixteen shell sample solution to the below pinch section of the new pinched problem. 234

7.76 A seventeen shell sample solution to the below-pinch section of the new pinched problem. 234

7.77 An eighteen shell sample solution to the below-pinch section of the new pinched problem. 234

CHAPTER 8

None.

PAGE

APPENDICES

D.1 A graph of 'P/Pmax' against R for various values of F. 245

D.2 An expansion of a section of graph D.1. 246

J.1 Removal of a loop from an LP tableau. 253

J.2 Another way to remove a loop from an LP tableau. 254

L.1 A pictorial representation of the typical solution to problem 5SP1 used in this example. 258

LIST OF TABLES

CHAPTERS 1 AND 2

None.

CHAPTER 3

3.1 Number of iterations required to obtain convergence using C05AZF. 106

3.2 Values of G for given values of R and F. 108

3.3 Values of F for given values of R with G equal to zero. 109

PAGE

CHAPTER 4

None.

CHAPTER 5

5.1 Permutation of a solution. 146

5.2 Reduced permutation of a solution. 147

CHAPTER 6

None.

CHAPTER 7

7.1 Number of nodes expanded to obtain the first optimal solution compared with the total number of nodes expanded for a cut-off temperature of 75K. 220

7.2 Results summary. 222

7.3 Results summary. 222

7.4 Data for the new unpinched problem. 224

7.5 Data for the new pinched problem. 230

PAGE

CHAPTER 8

None.

APPENDICES

I.1 The test problem data. 251

I.2 The test problem solutions. 252

L.1 The stream data for problem 5SP1. 257

L.2 The ordered list before any matches are made. 261

L.3 The ordered list after match one. 265

L.4 The ordered list after match two. 268

L.5 The ordered list after match three. 271

L.6 The ordered list after match four. 274

M.1 The data for problem 4SP2. 274

M.2 The data for problem 5SP1. 275

M.3 The data for problem 7SP2. 275

M.4 The data for problem 4SP1. 276

M.5 The data for problem TC1. 276

M.6 The data for the six-stream problem. 277

		PAGE
M.7	The data for problem 7SP4.	277
N.1	The effect of cut-off temperature range on the search tree of problem 4SP2.	278
N.2	The effect of cut-off temperature range on the search tree of problem 5SP1.	278
N.3	The effect of cut-off temperature range on the search tree of problem 7SP2.	279
N.4	The effect of cut-off temperature range on the search tree of problem 4SP1.	279
N.5	The effect of cut-off temperature range on the search tree of problem TC1.	279
N.6	The effect of cut-off temperature range on the search tree of the six-stream problem.	280
N.7	The effect of cut-off temperature range on the search tree of problem 7SP4.	280

NOMENCLATURE

A	Surface area of a heat exchanger.
B_c^j	Number of branches cold stream j is split into.
$B_c^j max$	Maximum number of branches cold stream j can be split into.
B_h^i	Number of branches hot stream i is split into.
$B_h^i max$	Maximum number of branches hot stream i can be split into.
C_n	Column number in a matrix used at level n of a tree search.
F	LMTD correction factor.
G	Temperature gap ratio.
Gmin	Minimum possible temperature gap ratio for a given value of R.
K_n	Unique number for matrix entries used at level n of a tree search.
L	Number of 'loops' in a network.
$\dot{m}Cp$	Heat capacity flowrate of a stream.
$\dot{m}Cp_c^j$	Heat capacity flowrate of cold stream j.

$\dot{m}Cp_c^{ij}$ Heat capacity flowrate of the branch of cold stream j that matches with a branch of hot stream i.

$\dot{m}Cp_h^i$ Heat capacity flowrate of hot stream i.

$\dot{m}Cp_h^{ij}$ Heat capacity flowrate of the branch of hot stream i that matches with a branch of cold stream j.

$\dot{m}\lambda$ Latent heat flowrate of a stream.

N_c Number of cold streams.

N_h Number of hot streams.

N_M Number of matches.

N_{ST} Number of streams including utilities.

NTU Number of heat transfer units.

P Temperature efficiency.

Pmax Maximum possible temperature efficiency for a given value of R.

R_n Row number in a matrix used at level n of a tree search.

R^{ij} Heat capacity flowrate ratio ($\dot{m}Cp_h^{ij}/\dot{m}Cp_c^{ij}$).

S Number of separate parts in a network.

T Temperature.

U Over-all heat transfer coefficient.

X^{ij} Binary integer indicating match existence or otherwise between hot stream i and cold stream j.

Y^{i} Binary integer indicating if hot stream i bypasses or not.

Y^{j} Binary integer indicating if cold stream j bypasses or not.

Δ Difference between the sum of the heat capacity flowrates for the hot and cold streams.

ΔT_{lm} Log-mean temperature difference.

ΔT_{m} Mean temperature difference.

ΔT_{min} Minimum temperature difference.

Superscripts:

in Condition at the inlet of an exchanger.

out Condition at the outlet of an exchanger.

Subscripts:

hot Conditions on a hot stream.

cold Conditions on a cold stream.

CHAPTER ONE

INTRODUCTION TO HEAT EXCHANGER NETWORK SYNTHESIS

1.1 AN OVERVIEW OF THE PAST AND PRESENT WORK ON THE SUBJECT

The methods for the systematic determination of suitable flowsheets for heat recovery, usually known as heat exchanger network synthesis procedures, have received much attention over the last two decades. Previously, sets of network configurations were generated in an ad hoc manner and the most suitable selected, sometimes after optimisation of the continuous variables in each network.

In the 1960's, studies into the design of heat exchanger trains were carried out in which one stream was to be heated by several streams to be cooled or vice versa (e.g. Westbrook (1961), Fan et al (1965)). These works were the first to handle the existence, or otherwise, of pieces of equipment in a systematic manner. Subsequently, Hwa (1965) started the investigation into the general problem of synthesis of heat exchanger networks. Although this work did not address the problem in full, the existence and scope of the problem were fully indicated and Hwa's paper can be considered to be the forerunner of the work carried out on two of the solution methods described in the literature.

The rapid rise in fuel costs during the 1970's added impetus to the study of the synthesis of high energy recovery networks and many different approaches were developed. In retrospect, it can now be seen that the most important developments were methods which are used to calculate

design targets that low-cost networks can approach closely and procedures that reveal the presence of energy 'bottle-necks', which have become known as pinch points. The design targets are of such importance because they can be calculated before any network synthesis takes place and hence used to screen candidate networks.

Latterly, the importance of generating networks within extra process constraints, in addition to those imposed by the laws of thermodynamics, has been recognised. This has also led to the realisation that networks cannot be selected on the basis of the sum of utility and discounted heat exchanger costs alone since other process factors must be accounted for such as flexibility, controllability and switchability. Thus some of the latest work has been aimed at giving sets of low, but not necessarily minimal, cost configurations on which further analysis can be carried out before the selection of the final network.

Problems still remain to be resolved in the field of heat exchanger network synthesis. Most importantly; the splitting of process streams into parallel branches, disliked because of the extra control problems it entails, is sometimes essential if high energy recovery is to be obtained. Methods exist which establish the need for stream splits but not for their effective introduction. The economic penalties incurred in some problems are so great if stream splits are not used that a procedure to determine the minimum number of stream splits necessary to obtain maximum energy recovery is developed in this thesis. Secondly, the use of heat exchangers that exhibit pure counter-current flow has been assumed by nearly all previous workers and the use of other heat exchanger configurations has not been studied extensively. Non-counter-current heat exchangers are considered in this work and utilised in the networks created.

The regions of the network in which process stream splits are most likely to be needed are centred around the problem pinch points. The algorithm developed here considers these pinch points and determines the set of surrounding configurations that exhibit the minimum number of stream branches within the constraints imposed. The solution technique is based on the formulation of the problem as a variant of the transportation problem from linear programming and is applied successfully to a number of problems.

The synthesis of networks away from the pinch points is also considered. A tree-searching method, which admits the use of non-counter-current heat exchangers, is developed which synthesizes networks which recover a given amount of process heat with the smallest possible number of heat exchanger shells. An implicit enumeration technique is used to reduce the search space size. This technique incorporates a method for updating the required bound on energy recovery from one level of the search tree to another. The complete procedure is applied to several problems of academic and industrial importance.

1.2 A FORMAL DESCRIPTION OF THE SYNTHESIS AND DETAILED DESIGN OF HEAT EXCHANGER NETWORKS

The problem considered in this thesis is that of the generation of a set of relatively low-cost heat exchanger networks composed of units which exchange process and utility heat between given sets of streams within any constraints imposed by the laws of thermodynamics and by the designer.

A set of process streams of known temperature range and heat capacity or latent heat flowrate is to be serviced

by heat exchangers that match process streams to be heated with process streams to be cooled or that match process streams with utility streams. Each member of the set of utility streams has a known inlet temperature and heat capacity or latent heat but unknown flowrate. In addition, the outlet temperatures of the utility streams that exhibit a temperature range are not fixed but merely bounded. This problem is usually simplified by representing the heat capacity and multi-component latent heat flowrates as piece-wise linear segments, i.e. the streams are partitioned into sequential temperature ranges each with a constant heat capacity or latent heat flowrate.

The streams to be heated have become known in the literature as cold streams and those to be cooled as hot streams irrespective of their absolute temperature.

The cost of a network is calculated on an annual basis. Associated with each utility stream is a price per unit mass supplied, which is used to calculate the network running cost. The capital cost of the heat exchangers is based on the heat transfer surface area of each shell raised to a positive power less than or equal to unity. This capital cost is written off over a fixed number of years to give a discounted capital cost. This is added to the running cost to give the total annual cost of each network.

The process conditions may limit the types of heat exchangers that can be used. In particular, heat exchangers exhibiting counter-current flow may not be suitable and configurations which include co-current flow may have to be introduced into the synthesized networks. If this is the case, extra precautions must be taken to ensure good design.

The parallel splitting of process streams is normally avoided, if at all possible, in order to simplify the network control scheme. However, in some cases, parallel splitting is essential if high energy recovery is to be obtained. In these cases the total number of stream branches is normally kept to a minimum.

Further constraints, such as disallowed matches between certain streams or limits on the number of matches or stream splits on any particular stream, may be imposed by the designer in addition to the constraints imposed by the laws of thermodynamics.

The low-cost networks synthesized to satisfy the above constraints should ideally be of sufficient variety to allow designers to distinguish between them in subsequent analysis stages. In addition, the number of solutions should be large enough to leave a sufficient number of candidate networks after unsuitable configurations have been discarded. It is the synthesis of this set of low-cost initial networks that this thesis considers in detail.

After the synthesis of a number of networks, the configurations are usually improved by parametric optimisation. In this stage of the design the over-all structure of the networks is fixed and only fairly minor alterations are made to the exchanger areas. The number of candidate networks is then reduced by suitable analysis techniques to give a few networks on which a detailed design and simulation can be carried out, leading to the selection of the final network. The network analysis would normally investigate the flexibility, controllability and switchability of each configuration.

The process of parametric optimisation, analysis, detailed design and simulation is not considered here as these

operations do not form a significant part of network synthesis. The procedures needed to carry out these parts of the design are well documented in the literature.

1.3 THE PINCH POINT AND COMPOSITE CURVES

The concept of the problem pinch point is crucial in the development of effective synthesis methods and so a brief description is given below. An extensive explanation of how to determine the position of these points and the significance of them is given in the I.Chem.E. user guide (1982).

The existence of pinch points can be determined by plotting the problem data on a graph of temperature versus relative enthalpy (see figure 1.1). Two composite streams are plotted on the graph, one representing all the hot process streams and the other all the cold process streams. Since relative, rather than absolute, enthalpy is plotted along the abscissa, the curves can be moved relative to one another. A match between the hot and cold composite streams is denoted by placing the relevant part of the hot curve above the cold curve leaving the unmatched stream sections, which project either side, to be satisfied by utilities (the laws of thermodynamics preclude the cold curve being placed above the hot curve). Maximum energy recovery is obviously obtained if as much as possible of the hot curve is placed above the cold curve. This occurs when either one composite stream is completely satisfied by the other (see figure 1.2) or when the two curves touch at some point; the pinch point (see figure 1.3). If the two curves touch a zero temperature difference occurs at this point and the heat transfer surface area required becomes infinite.

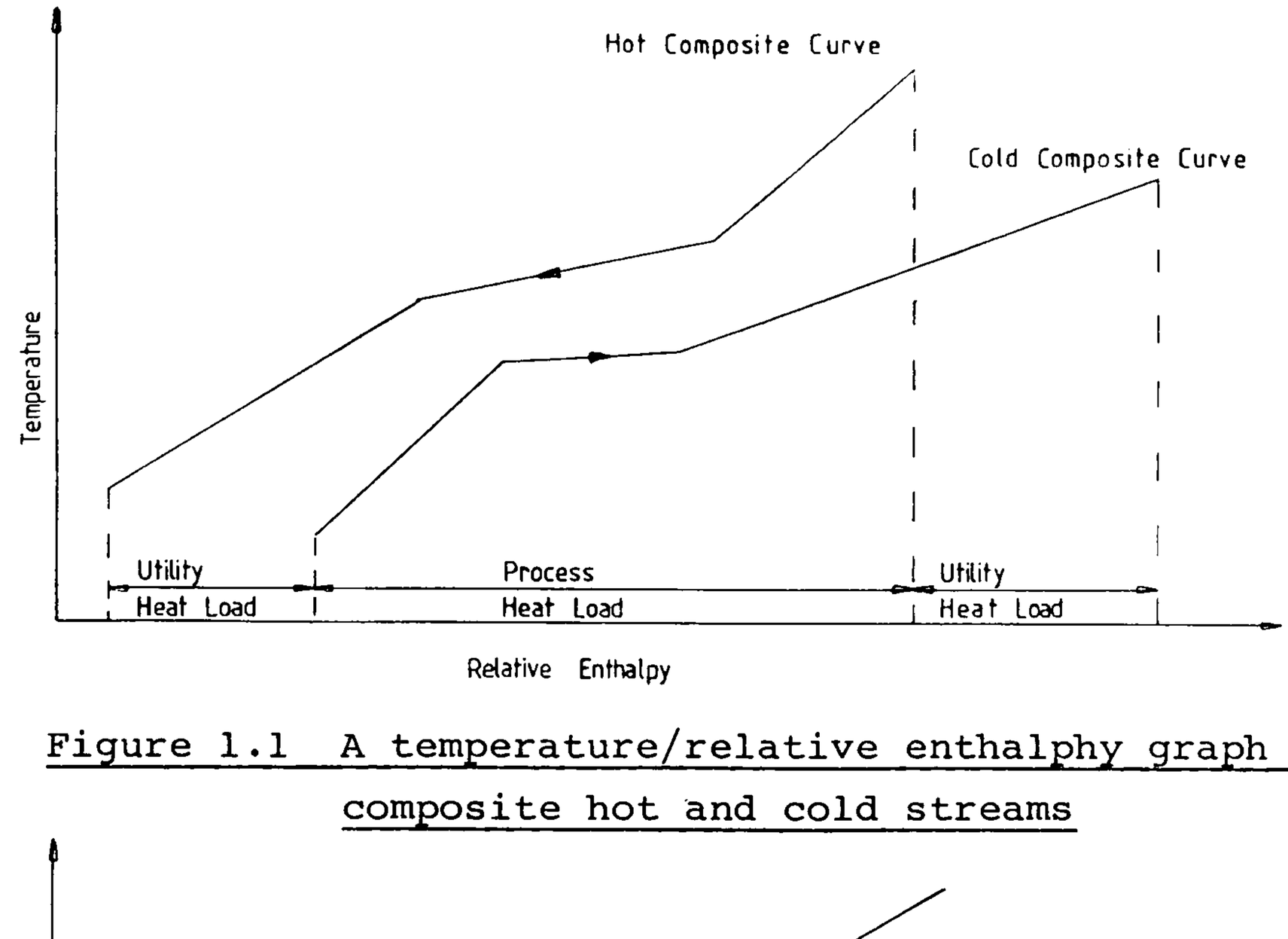


Figure 1.1 A temperature/relative enthalphy graph with composite hot and cold streams

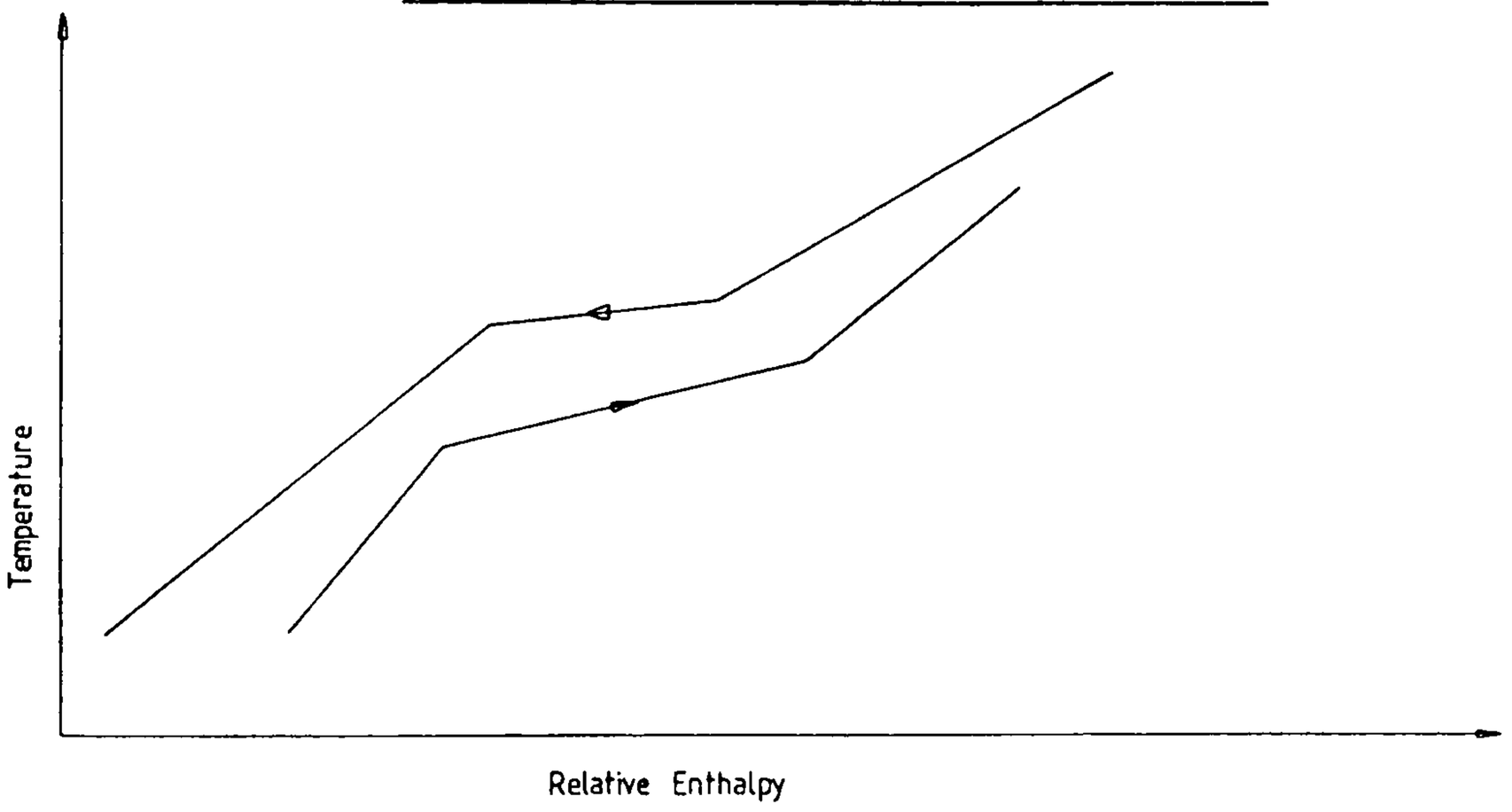


Figure 1.2 Maximum energy recovery resulting in complete satisfaction of a stream

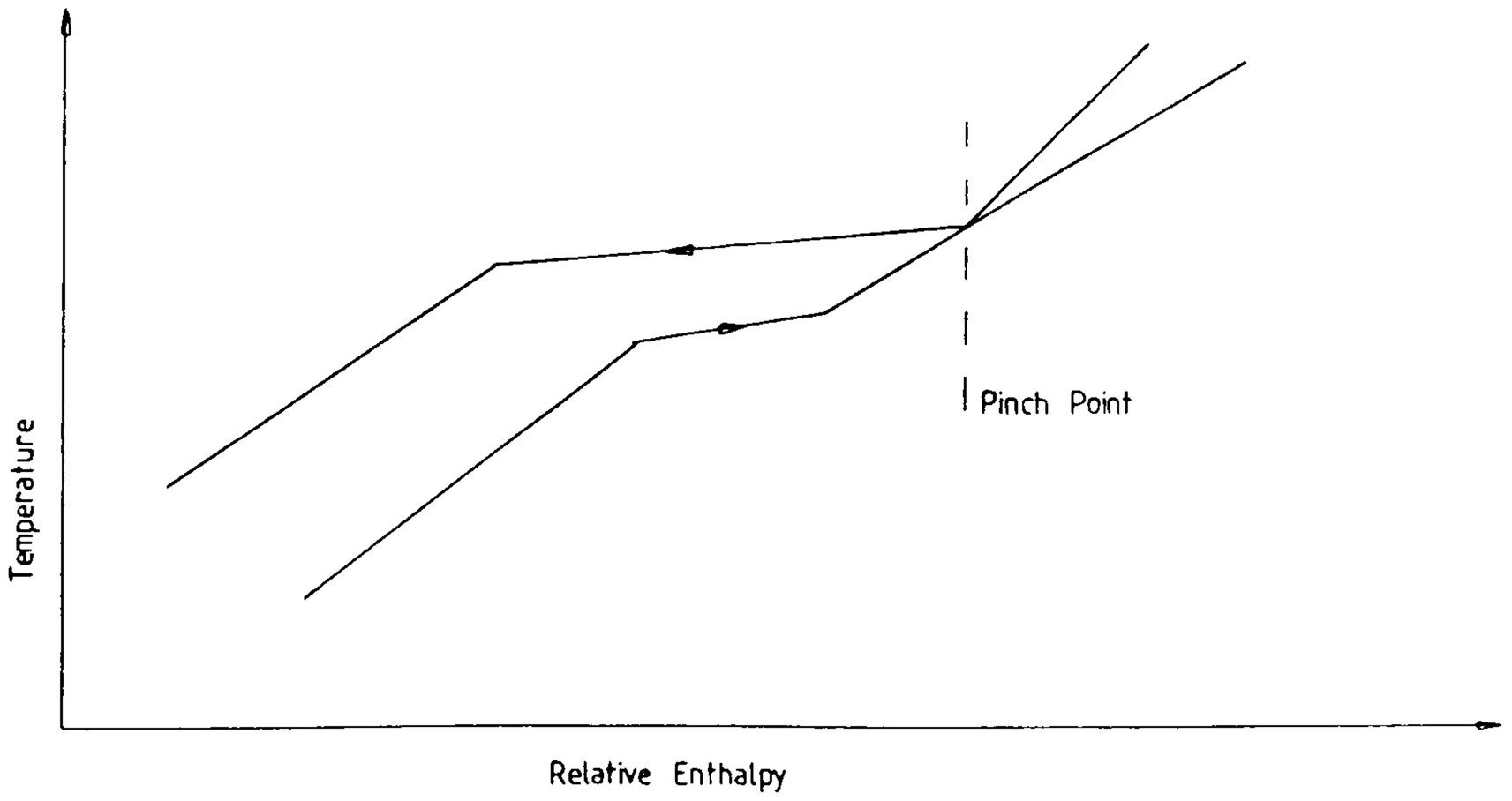


Figure 1.3 Maximum energy recovery resulting in a pinch point

A finite heat transfer area is normally ensured by imposing a constraint on the minimum temperature difference (ΔT_{min}), i.e. the minimum gap between the two composite curves. Thus the pinch point becomes identified by two temperatures; one for the composite hot stream and one for the cold.

Problems in which the requirements of neither of the composite curves can be completely satisfied by the other, irrespective of the value of the minimum temperature difference, are known as pinched problems (see figure 1.3). On the other hand, problems in which one of the composite curves can be completely satisfied by the other with a sufficient gap between the composite curves being maintained are known as unpinched problems (see figure 1.4). These problems require either hot or cold utilities but not both. As the minimum temperature difference is increased in an unpinched problem, a point is reached at which the minimum gap between the composite curves is equal to the minimum allowable and the problem is known as a threshold problem (see figure 1.5). Further increases in the minimum temperature difference force the problem to be pinched.

1.4 AN OUTLINE OF THE FOLLOWING CHAPTERS

In chapter two a literature review is carried out which highlights several areas of heat exchanger network synthesis that require further investigation. The most important of these topics are considered in chapters three to six and new techniques are developed to solve the associated problems. Finally, these algorithms are applied to several problems of industrial and academic significance and the results discussed in chapters seven and eight.

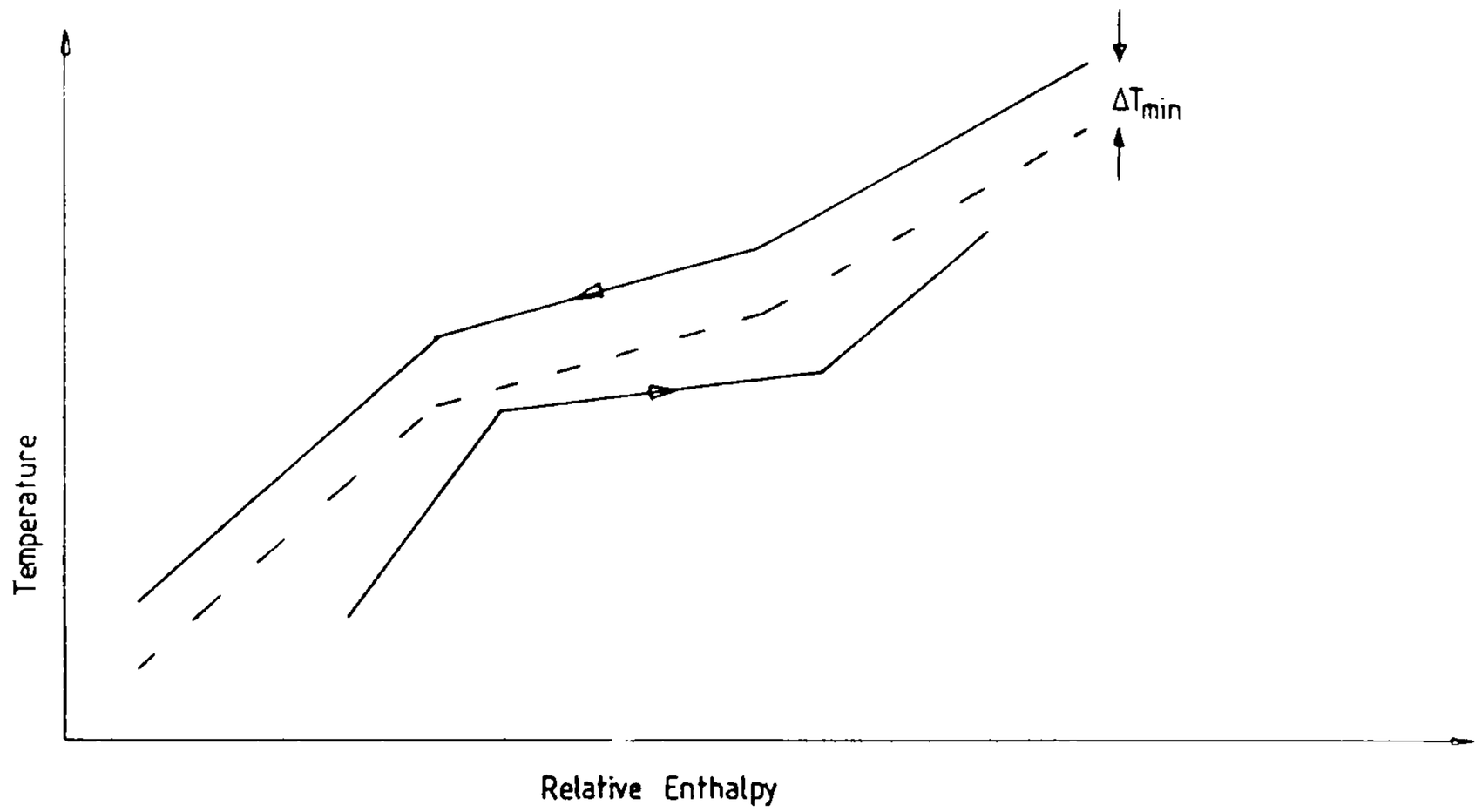


Figure 1.4 An unpinched problem

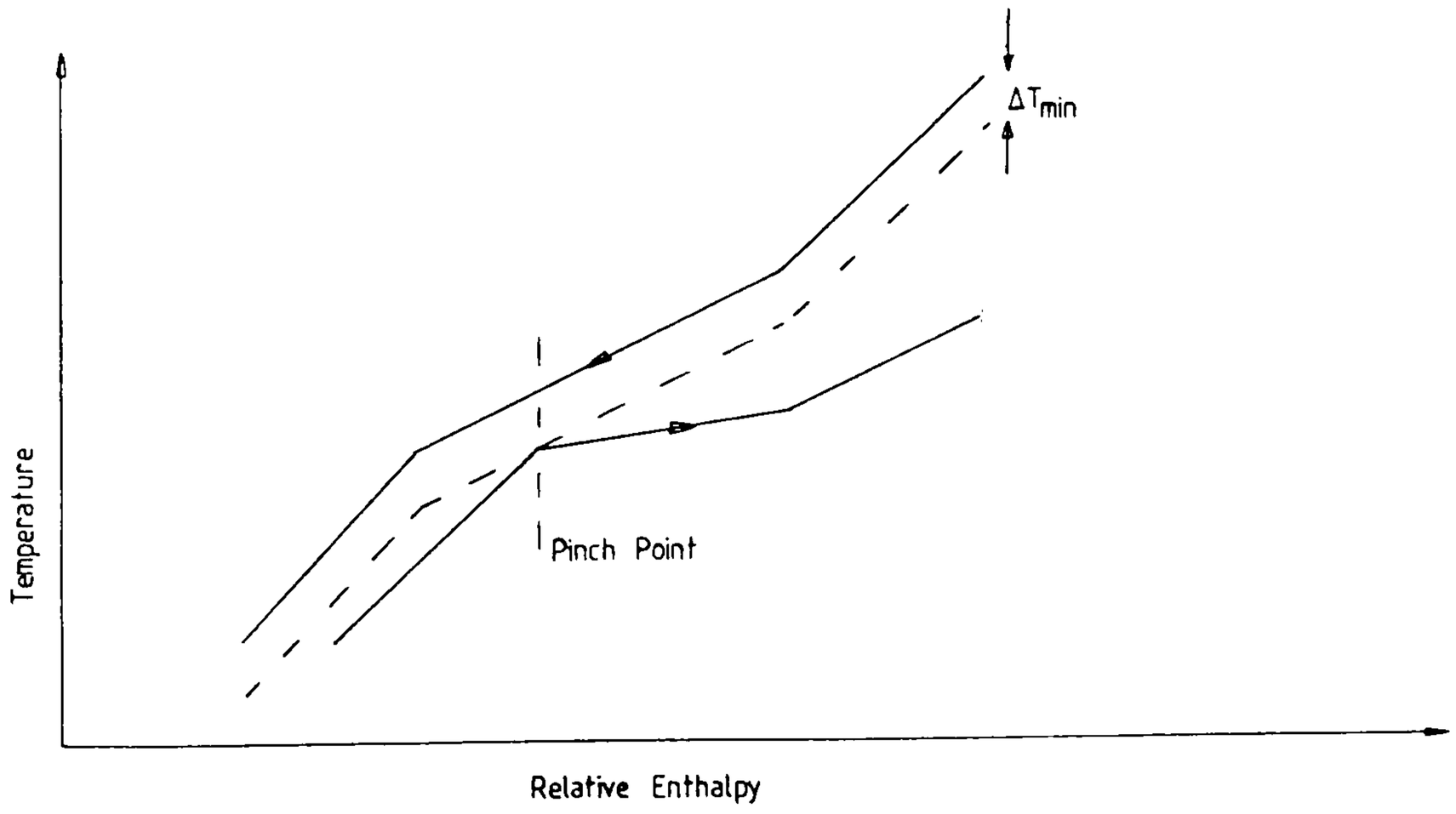


Figure 1.5 A threshold problem

CHAPTER TWO

HEAT EXCHANGER NETWORK SYNTHESIS LITERATURE REVIEW

2.1 INTRODUCTION

The synthesis of heat exchanger networks has developed considerably over the last two decades. Most procedures are composed of three stages:

a) A pre-synthesis analysis, or target-setting, stage.
b) A network synthesis stage.
c) A post-synthesis, network evolutionary stage.

The first stage is clearly defined and autonomous, whereas the other two are inter-dependent. Indeed, in the most extreme cases, part (c) can be omitted because network evolution is an integral part of the synthesis procedure in part (b). Hence a somewhat arbitrary division of the evolutionary methods is made in the following review.

2.2 PRE-SYNTHESIS ANALYSIS

Targets, which aid the synthesis of heat exchanger networks, can be calculated before any network is generated. There are three targets of importance:

a) The minimum utility consumption needed for given minimum temperature difference.

b) The minimum number of matches needed to satisfy all streams.

c) The minimum heat transfer surface area needed to satisfy all streams.

The first two targets are very important and many near-optimal networks achieve them. The third is usually approached but not obtained. Near-optimal networks have heat transfer surface areas within a few percent of the minimum but further decreases can only be made at the expense of increasing the numbers of heat exchangers which is very uneconomical.

2.2.1 Minimum utility consumption target

This target has been proved by many workers to be of utmost importance in the synthesis of heat exchanger networks. There are four methods of calculating a lower bound on utility consumption, described here in order of increasing tightness and calculation complexity.

The first method estimates the utility consumption as the difference of the sums of the heat contents of the hot and the cold streams. This value follows directly from the heat balance and has been used widely for many years. It is very easy to apply but often gives values far from the real thermodynamic limit and can never indicate the need for both hot and cold utilities.

The second method, suggested by Rathore and Powers (1975), admits the use of a minimum temperature approach constraint and starts to account for the thermodynamic constraints. All the sections of hot streams that are cooled below the coldest cold stream inlet temperature corrected for minimum temperature difference and all the sections of cold streams that are heated above the hottest hot stream inlet temperature corrected for minimum temperature dif-

ference are excluded from the heat balance since these must be satisfied by utilities. This method can indicate the need for both hot and cold utilities but is often quite slack.

The third method, suggested by Hohmann (1971), gives a strict, achievable bound on minimum utility usage assuming, like the two methods above, all matches are allowed. This uses the following procedure:

a) Stream temperature 'delimiters' are sorted into numerical order, after taking account of any minimum temperature difference, to form a series of intervals (a 'delimiter' being a stream inlet, outlet or intermediate point where the heat capacity flowrate changes).

b) The heat capacity flowrates of the streams in each interval are summed, with hot streams positive and cold streams negative. Each of these totals is then multiplied by the temperature difference across the relevant interval to give the net interval heat content. Single-component phase changes are accounted for by including intervals with zero temperature difference and a net heat content equal to the latent heat flowrate with the relevant sign for that stream (Su (1979) and Su and Motard (1984)).

c) The net heat contents of each interval in turn are summed from the hot to the cold end and the value of the running total at each interval assigned to that interval.

d) The most negative of the running totals is located. If no negative running total exists no heating utility is required, otherwise the magnitude of the

most negative running total gives the heating requirement. The cooling requirement then follows from the heat balance.

This method often gives significantly superior results to the two previous methods at the cost of increased computational load. There is a large amount of sorting, addition and searching required such that each interval involves about the same workload as the whole problem in the other two methods.

The last method accounts for disallowed matches and match dependent minimum temperature differences by using linear programming models. Cerda (1980) and Cerda et al (1983) use a transportation model and Papoulias and Grossmann (1983a) a trans-shipment model to partition the problem into intervals similar to Hohmann. However, in these methods only potential pinch points are used as interval separators instead of all the stream 'delimiters'. When no matches are disallowed and all minimum temperature differences are equal, the methods reduce to that of Hohmann (1971) and the optimal solution is easily found. However, if the minimum temperature differences are variable, the number of intervals increases very quickly indeed and the solution becomes difficult. Even in the case where only the disallowed match options are imposed, the linear programme is not amenable to frequent solution by hand. The application of this method is efficient only in a few, difficult cases rather than after each match within a synthesis procedure.

The last three methods rely on a value of the minimum temperature difference being set prior to the calculation of the utility usage bounds. Approximate values for this figure have been obtained in the past by assuming that networks exhibiting minimum utility usage, minimum number

of heat exchangers (see section 2.2.2) and minimum heat transfer surface area (see section 2.2.3) can be synthesized. An approximate cost can then be calculated for these imaginary networks, at various values of minimum temperature difference, by assuming the surface area is evenly distributed among the heat exchangers. Thus the optimal value of the minimum temperature difference can be estimated. As will be seen, this aproach is not valid in the general case and a more suitable technique is required.

2.2.2 Minimum number of matches target

Linnhoff and Turner (1981) show that the division of a heat exchanger network problem into sub-problems, by partitioning at the pinch points, applies equally to the calculation of the minimum number of matches in a maximum energy recovery design. However, rare exceptions do occur when matches can cross a pinch point. This happens when the heat capacity flowrate ratios of matches equate exactly to unity or when the heat capacity flowrates of streams change suddenly at the pinch points. However, a heat capacity flowrate ratio of unity is to be avoided, if at all possible, as the minimum temperature difference will occur along the whole length of the match making the heat transfer surface area very large and the exchanger difficult to control. In addition, large changes in heat capacity flowrate of a stream are usually due to the transition from a single-to a two-phase system or vice versa. Since it is good practice to separate superheating and subcooling from condensation and evaporation, multiple heat exchangers shells would normally be employed for this match and, hence, increase the exchanger count back to that indicated.

Unfortunately, in problems with large heat capacity flow-

rates, area limitations on heat exchanger sizes result in the use of multiple shells. Similarly, the use of non-counter-current exchangers often results in more than one shell being used to satisfy a particular duty because of temperture cross considerations. Thus, in these two cases, the number of heat exchangers and matches are not equal. Finally, in problems where energy integration is difficult away from pinch points, more than the indicated minimum number of matches and hence heat exchangers may be required. In all these cases, the minimum number of heat exchangers does not equate to the minimum number of matches. No simple, pre-synthesis calculation procedure to obtain the minimum number of exchangers is likely to be forthcoming in these situations.

2.2.2.1 Minimum number of matches

Hohmann (1971) and Hohmann and Lockhart (1976) develop a target for the minimum number of matches. They show that, unless a subset of the hot streams has exactly the same heat content as a subset of the cold streams, the minimum number of matches (N_M) is given by the total number of streams (N_{ST}), including utilities, less one, i.e.:

$$N_M = N_{ST} - 1 \qquad (2.1)$$

In addition, if more than the minimum number of matches are used heat load 'loops' must occur (see section 2.4.1).

Linnhoff et al (1979) stated the problem in the terms of Euler's network relationship, which relates the number of matches to the network structure. This takes the form:

$$N_M = N_{ST} + L - S \qquad (2.2)$$

where L is the number of independent heat load 'loops' and S is the number of separate parts of the network. This reduces to equation (2.1) when the network cannot be separated into subsets (i.e. S = 1) and no 'loops' exist (i.e. L = 0).

Cerda (1980) and Cerda and Westerberg (1983) consider the minimum number of matches through their linear programming synthesis method (see section 2.3.1.2.3) and show it to be equal to the number of linearly independent equality constraints at any vertex. Since upper bound secondary constraints are present, due to thermodynamic considerations, it follows that heat load 'loops' may occur in the optimal solution.

2.2.2.2 Minimum number of heat exchangers

As has already been discussed, a one-to-one correspondence between the number of matches and the number of heat exchangers does not always occur. Euler's network relationship, equation (2.2), can be applied equally well to give the number of heat exchangers but in this case the value of L is unlikely to be zero. Although the objective of minimising the number of heat exchangers, while maintaining maximum energy recovery, has been shown by many workers to be worthwhile, the calculation of this number before the synthesis step is not possible in the general case. A good lower bound is given by equation (2.1).

2.2.3 Minimum area target

Assuming an over-all heat transfer coefficient (OHTC) that is equal for all matches and that all matches are allowed, the calculation of the minimum heat transfer surface area is reasonably simple. The problem is considered one of matching the hot and the cold composite streams (see section 1.3), as if they are ordinary process streams, in a counter-current exchanger. This method forms the basis of the heat content diagram synthesis methods (see section 2.3.4.1), the solutions to which always exhibit large numbers of heat exchangers and stream splits.

However, many networks utilise heat transfer areas within a few percent of the minimum, including configurations with far fewer exchangers, often the minimum possible. These networks cost much less than the minimum area networks because a given surface area is much cheaper to buy in a few large units. Thus, the objective of strictly minimising surface area is not worthwhile in most cases.

2.2.4 Conclusions on pre-synthesis analysis

The aim of minimising the number of heat exchangers is a valid one but only in special circumstances can the actual figure be calculated.

The minimum utility consumption target is very important. However, the calculation load of the more exact methods is heavy if used within an automatic synthesis method. More efficient calculation procedures are required. The linear programming methods are complex, in the general case, and are not suitable as sub-procedures of a synthesis algorithm. A procedure is required to calculate the optimal minimum temperature difference so that the utility

consumption target can be calculated. The minimum area target is not a worthwhile objective.

2.3 HEAT EXCHANGER NETWORK SYNTHESIS

Many reviews of the previous work in the heat exchanger network synthesis field have appeared in the literature. The more extensive of these works are listed below, in chronological order:

a) Hendry et al (1973)

b) Hlavacek (1978)

c) Polley (1980a)

d) Westerberg (1980)

e) Nishida et al (1981)

f) Umeda (1983)

However, these reviewers have considered the problem of isolating a single, optimum solution to the synthesis problem containing only counter-current heat exchangers. This review outlines the suitability of previous work for the generation of sets of near-optimal networks which can include heat exchanger formats of a more general nature.

The methods applied by previous workers can be divided into four broad classes:

a) The mathematical methods.

b) The heuristic methods.

c) The thermodynamic methods.

d) The graphical methods.

As the subject has become more deeply researched, discoveries in each class have been brought together resulting in hybrid approaches which do not fit snugly into any of the above sections. These methods are classified under their general approach rather than their specific details.

2.3.1 Mathematical methods

The synthesis of heat exchanger networks, when formulated as a mathematical problem, takes the form of a mixed integer non-linear programme (MINLP, see appendix A). This MINLP can be addressed directly or simplifications can be made to transform the problem into a more amenable form. The resulting methods can be divided into five classes:

a) The adaptive random search method (MINLP).
b) Linear programming methods (MILP/ILP/LP).
c) Structural parameter methods (NLP).
d) Branch-and-bound methods (MILP/INLP/ILP).
e) The optimal control method (INLP).

The MINLP is very difficult to solve directly and the method that has been applied is very inefficient. The other methods solve simplifications of the original problem and all have drawbacks because of this. The results obtained are only as good as the realism of the formulation permits.

2.3.1.1 The adaptive random search method

Kelahan and Gaddy (1977) present a method which attempts to solve the MINLP directly using an adaptive random search procedure. This procedure is dependent on two parameters over which careful control must be kept; increasing them speeds the search but also makes the probability of locating only a local, rather than global, optimum greater. Unfortunately, no method is given to determine these parameters. Since the method is random in nature, the speed of convergence is very variable but the average figures quoted show it to be very inefficient. The search region of even the smallest problems is very poorly behaved resulting in the procedure carrying out an almost exhaustive search. The addition of a heuristic which sets the heat content of any selected match is shown to improve the procedure's efficiency dramatically, but this effectively changes the problem from an MINLP to an INLP for which better solution methods exist (see section 2.3.1.4).

2.3.1.2 Linear programming synthesis methods

The general heat exchanger network synthesis problem is non-linear and so the problem must be forced into a linear format. In addition, since mixed integer linear programme (MILP) solution methods often involve a large computational load, the problem is often developed further into a linear programme (LP) or an integer linear programme (ILP) for which efficient solution methods exist. However, independent of which method is used, there is always the subsequent difficulty of the interpretation of the optimum result obtained and its translation into a realistic network.

The linear programming models used, in chronological order of first use, are:-

a) Separable programming.

b) The assignment problem.

c) The transportation problem.

d) The trans-shipment problem.

In all these methods, the heat exchangers are created simultaneously rather than sequentially i.e. there is always a complete network under development rather than a partial network being added to. Thus all the interactions between heat exchangers in the network are accounted for, unlike the sequential matching methods where only the interactions between the current partial network and the new match can be considered and interactions with future matches, which will complete the network, must be ignored.

2.3.1.2.1 Separable programming

Hwa (1965) applies separable programming techniques to the heat exchanger network synthesis problem. This involves the introduction of piece-wise linear approximations to the non-linear functions to obtain an MILP. A composite network containing an amalgamation of likely solutions is created by the engineer, fixing the integer variables and leaving an LP. This composite network is then optimised in an attempt to remove any unnecessary exchangers.

The major drawbacks of this approach are the lack of an algorithm to develop the initial good solutions for the composite network, the inability to ensure the isolation

of the global optimum due to the hyperbolic nature of the non-linearities and the excessive increase in computation load with problem size. This approach has since been developed into two distinct areas, the structural parameter methods (see section 2.3.1.3) and the true linear programming solution methods, described below.

2.3.1.2.2 The assignment problem

Many workers utilise the assignment problem formulation from linear programming. Each stream is divided sequentially, and sometimes also in a parallel manner, into sections so that all stream sections have the same heat content. The heat content of each section is set to a value small enough to model each stream accurately and to linearise the objective function. This leaves an assignment problem (an ILP) for which very efficient solution techniques exist.

There are two major drawbacks to this approach. Firstly, the solutions obtained are composed of a large number of very small exchangers. Adjacent matches which join the same two streams have to be merged to make larger, composite matches but no satisfactory automatic methods exist to do this. Additionally, the manual methods suggested often result in a large number of unnecessary stream splits. Secondly, a cost must be assigned to each possible match between hot and cold segments. Initially, each can be set to the cost of the small exchangers serving each segment match. However, as merging is introduced the cost per unit area drops, due to the non-linear function of cost against area, forcing merged segments to increase even more and not allowing new matches to enter the simplex tableau. Thus a potentially good match may be excluded because the initial entry, to which adjacent

matches between the same two streams could be added and hence merged, is too expensive.

Kestler and Parker (1969) are the first to apply LP methods to the general synthesis problem. Each stream is divided along its length into a series of segments. An MILP is then formulated which matches hot and cold stream sections in a feasible way. This MILP is solved by using an assignment problem algorithm, used to ensure feasibility, coupled with a modified LP algorithm through an adjacent exchanger merging procedure. The MILP circumvents the match cost problem, discussed above, by considering only a given set of possible stream matches (usually a very small subset of the total).

Cena et al (1977) use an assignment problem formulation, an extended form of Kesler and Parker's, within a simple iteration loop adjusting the match costs as merging takes place. The need for sophisticated pattern recognition heuristics and a clearly defined approach to parallel stream splitting is discussed and the growth of merged matches at the expense of others is noted.

Kobayashi et al (1971) allow streams to be segmented both sequentially and into parallel branches. An assignment problem formulation, used to obtain structures, is coupled with a fixed configeration network optimiser in an iterative loop converging on utility load. The placement of utility exchangers is fixed before the synthesis takes place and only the loads are altered. No precise method is suggested for the best stream segmentation strategy, which is a major pitfall as this procedure determines the number of stream splits in the solution. The problem of merging similar, adjacent exchangers is not addressed.

Kafarov et al (1980) and Kafarov and Shmidt (1981) consider the theoretical aspects of solving the assignment problem but no networks are synthesized and the implicit assumptions in the formulation are not corrected.

Jezowski and Pruchnicki (1981), Jezowski (1982) and Jezowski et al (1983) use the problem formulation of Cena et al (1977) but set the utility flows to the minimum required, as indicated by Linnhoff and Flower (1978a). In an attempt to overcome the problem of too many exchangers in the solution, the ability to have a set of essential matches and a set of disallowed matches is introduced. However, no method is presented to establish the content of these sets automatically, so a further attempt is made to ensure minimum match solutions in Jezowski (1982) and Jezowski et al (1983). This involves using a technique similar to Linnhoff and Hindmarsh's (1983) 'ticking-off heuristic' coupled with an assignment problem solver to choose the streams, or residual streams, matched. This approach can result in solutions far from optimum.

2.3.1.2.3 The transportation problem

Cerda (1980) and Cerda and Westerberg (1983) develop a synthesis method based on the transportation problem in linear programming. The first step in the procedure is the calculation of the minimum utility requirement described in Cerda (1980) and Cerda et al (1983), as already discussed (see section 2.2.1). This linear programme is developed into an MILP for the network synthesis step and is relaxed in several ways to obtain two LP's which are more amenable to solution. The objective of the MILP, and its subordinate LP's, is to find networks with the minimum number of pairings between streams. However, the solutions obtained are of the form

of small elements of heat to be exchanged, similar to the assignment problem elements but not as numerous. These elements are merged, as far as possible, but in a fair proportion of cases complete amalgamation is not possible and more than one heat exchanger is required. Thus a major drawback of the method is the inequality between the number of heat exchangers required and the objective function value. Another drawback is the implicit assumption that stream splits are allowed, and can be introduced in any number, to ensure the minimum number of stream pairings. The need to introduce stream splits is often an essential part of the MILP solution merging step and cannot be circumvented. Thus, in the heirarchy of objectives, the number of exchangers utilised is placed totally above the number of stream splits used.

There are strong similarities between this approach to the problem and Linnhoff's temperature interval method (see section 2.3.2.2) where solutions are sought to each interval and then merged together.

2.3.1.2.4 The trans-shipment problem

Papoulias and Grossmann (1983b) apply a trans-shipment model solver to the same formulation used by Cerda and Westerberg (1983). This results in a large decrease in the LP size and hence decrease in the computational load. The models are developed, in a similar fashion to Cerda and Westerberg's treatment, from a minimum utility requirement LP problem (see section 2.2.1) to the complete MILP synthesis problem. However, Papoulias and Grossmann solve the MILP direct, rather than relaxing the problem and solving an LP. This approach has the same drawbacks as the transportation problem techniques but is computationally more efficient.

2.3.1.3 Structural parameter optimisation methods

The structural parameter based methods remove the integer variables from the general MINLP formulation to leave an NLP which is more amenable to solution. This is achieved by modelling an amalgam formed by including all the likely heat exchanger and stream configurations into one 'super-network'. Thus, the integer variables denoting existence or otherwise of units are replaced by continuous variables denoting flow or split fraction. This approach is conceptually very powerful but fraught with problems. A large number of variables are required and often only a subset of the required 'supernetwork' can be considered. This subset must be formed very carefully to avoid making the objective function discontinuous and, hence, much more difficult to solve. In addition, the non-convexity of the objective function and feasible space leads to heat exchanger network synthesis problems having many local optima. Thus a branch-and-bound procedure must be incorporated to ensure global optimality of the solution obtained (see, for example, Westerberg and Shah (1978)). The main benefit of this method is that, if an optimal solution is obtained, all the split fractions and exchanger sizes are fully optimised.

The solution of the general flowsheeting synthesis problem is considered by Umeda et al (1972), Osakada and Fan (1973) and Ichikawa and Fan (1973) among others. Unfortunately, these workers have not considered the specific problem of heat exchanger network synthesis but have noted the drawback of the large problem size, caused by the high redundancy of subsystems required in the initial 'supernetwork'.

Grossmann (1977) discusses the problems of non-convexities and the resulting multiple local optima in depth. Various

relaxations of the problem are also considered but all are shown to be impractical.

Shah and Westerberg (1980) describe an equation-orientated flowsheeting package designed specifically for heat exchanger networks. This package can also undertake parametric optimisation if any degrees of freedom exist and, therefore, be used for synthesis. The quoted solutions are for small problems (up to seven process streams) and the one problem described at any length considers a very small subset of the complete 'supernetwork'.

The occurrence of inequality constraints, such as a minimum temperature difference, can cause optimisation procedures to find sub-optimal answers if the problem is not properly structured. In particular, even though heat exchanger bypasses may not be allowed in the final solution they must be included in the formulation to preserve the continuity of the objective function. They are used as a vehicle for extra mass-flow when small exchangers are in the process of being removed from the solution. This results in a large increase in the dimensionality of the problem rendering it impractical. This problem is discussed further by Shah and Westerberg (1977) and Heydweiller and Fan (1982).

2.3.1.4 Branch-and-bound methods

Branch-and-bound techniques are used to solve problems for which no direct satisfactory solution methods exist. Simpler problems are devised which bound the original problem and can be branched to. These simpler problems may be solvable in themselves or further branching may be required to the point where the original problem is simplified into one which can be solved.

The bounding methods that have been used take two forms. The first bounding method allows simultaneous matching of a stream with more than one other stream. This allows the processing of each stream to be considered in isolation, which is a much easier problem to solve. The cheapest solution that satisfies the original problem is then extracted from an ordered list of the solutions to the bounding problem. The second bounding method, which can be added to the first or used independently, replaces the original problem by a set of problems which jointly bound the original. Each member of this set takes the form of the original problem but with one heat exchanger match known to exist. A full set is thus obtained when each of the possible heat exchanger matches is contained in one member of the set. This branching strategy can be applied repeatedly to decrease the combinatorial problem size and, when taken to its extreme where all matches are defined, results in a problem tree (see figure 2.1).

A problem tree takes the form of a root node, the level zero node, representing the original problem. A set of nodes, the level one nodes, are connected to this root node representing the problems containing one specific match. The level two nodes are connected to these, representing problems containing two specific matches. This

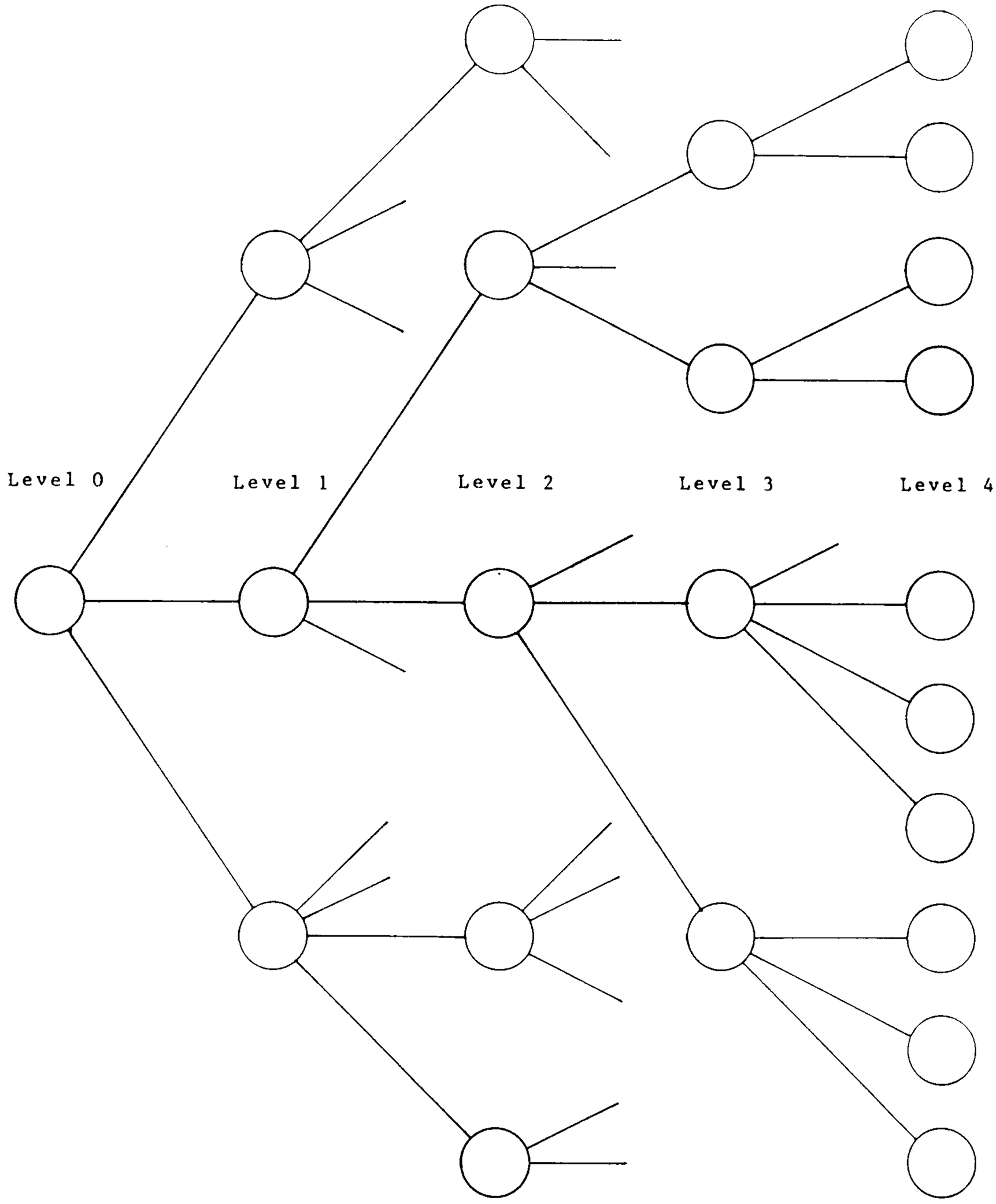


Figure 2.1 A branch-and-bound problem tree

tree branches downwards, continuing in a similar fashion, until all the branches terminate at the goal nodes where each problem has a full set of matches and the heat exchanger network is fully defined. Each level also contains a node where no match is made, which often used in practice to denote the remaining stream duties are to be serviced by utilities.

The simplest solution technique used in tree searches expands each level in turn. Thus all the nodes at level one are enumerated before any at level two, leading to the description 'breadth-first' procedure. This requires a large amount of data storage and is impractical in all but the smallest problems. Another type of algorithm, the 'depth-first' procedure, expands down into the tree in an attempt to isolate a solution as quickly as possible. The storage required is much less using this approach since the only nodes stored are the ones in the current partial solution being considered.

Improvements to increase the efficiency of the basic search algorithms can be made if more is known about the tree. Knowledge of where nearly optimum solutions are likely to be found can be used to direct the search rather than searching blind. Knowledge of a numerical lower bound on the 'cost' of further processing of any node can be used to formulate an implicit enumeration algorithm, removing the need to enumerate all nodes explicitly. The closer, or tighter, the lower bounds are to the actual 'cost' the fewer the number of nodes to be expanded becomes, improving the efficiency of the procedure.

Branch-and-bound techniques do not, in theory, involve any limiting assumptions but their practical applications do. All the methods used rely on branch-and-bound procedures to define the integer variables and further heuristics

must be added to determine the magnitude of the continuous variables. Therefore, the methods are only as good as the heuristics used. In addition, the combinatorial problem cannot be fully overcome and a total search of the problem trees created from large problems is infeasible, even with ordered searches and partial enumeration. Thus procedures must rely on only partial searches in the areas of the problem tree where the best solutions are likely to be found. Partial enumeration is complicated by the need to strike a balance between the tightness of the lower bound on further 'cost' and the computational load involved in calculating the lower bound. A very tight bound may involve such a large computational load that it can be more efficient to use a slacker, more easily calculated bound, even though the expansion of more nodes takes place, because the over-all computational load decreases. Finally, all these methods require an algorithm to introduce stream splits where necessary because this cannot be done efficiently as part of the tree-searching procedure.

Lee et al (1970) apply a branch-and-bound technique in a list-processing procedure to find optimal heat exchanger networks. It is assumed that cyclic networks cannot be optimal and so as much heat as possible is exchanged in each match without resorting to stream splitting. This allows a matrix to be formed with all the columns representing either entire hot streams or residual hot stream segments which are left after matching with cold streams and all the rows representing either whole or residual cold streams. Each element of the matrix has a figure associated with it representing the modified cost for that match. A list giving all the ways the duty of stream can

be satisfied is extracted from the matrix for each process streamand a list-processing procedure is used to locate the cheapest network. The approach uses both of the bounding methods described above to translate the problem into a tractable form. Unfortunately, the original matching matrix grows exceedingly quickly with problem size making the technique unmanageable except for the very smallest problems.

Menzies and Johnson (1972) use a similar method to Lee at al (1970) but add heuristics to decrease the number of possible matches in each list. The design of a utility system is also outlined based on a price-based decomposition algorithm. Although the problem size is decreased considerably, no guarantee can be made of the suitability of the network achieved due to the fallibility of the extra heuristics.

Pho and Lapidus(1973) suggest a way to circumvent the problem of the growth of the number of stream matching possibilities and, hence, the size of the cost matrix. The bounding problem allowing simultaneous matching of streams is dispensed with, leaving a tree-search problem. A compact matrix representation is used which assumes a pair of streams can match only once and that they then transfer as much heat as thermodynamically possible without stream splitting. The combinatorial nature of the problem is found to be overwhelming and a look-ahead procedure is implemented for partial enumeration of the tree. This provides inferior results to those that could have been achieved with implicit enumeration. The matrix representation makes the inclusion of non-counter-current exchangers impossible in the general case.

Rathore and Powers (1975) use a depth-first tree search with implicit enumeration. Two important features are

included; the ability to rematch pairs of streams (cyclicity) and the use of a maximum energy recovery (minimum utility usage) bound to determine network completion costs. The explicit statement of this energy bound leads directly to the conclusion that only networks with maximum energy recovery should be synthesized.

Raghavan (1977) and Greenkorn et al (1978) revert to the matrix representation of Pho and Lapidus (1973) modifying it to allow utility exchangers to be placed anywhere in the network. An implicit enumeration scheme is incorporated, based on maximum energy recovery targets and minimum number of heat exchangers, and an ordering heuristic, based on a fallible upper bound on network cost. The procedure partitions the problem into two sections. The first section sets the utility flowrates and the second synthesizes a network with these given flowrates. Reasonable results are achieved but cyclic networks are still excluded.

Liapis and Walter (1977) outline a method to synthesize networks based on minimising the lost work of the system. A tree search is carried out where exchangers with the lowest entropy change are added at each level, assuming no stream splits. The combinatorial problem is decreased by considering only matches between the two hot streams with the highest inlet temperatures and the two cold streams with the highest outlet temperatures at any level. However, this excludes a large amount of the solution space and cannot be regarded as satisfactory.

Grossmann (1977) and Grossmann and Sargent (1978) apply a two-stage procedure to synthesize heat exchanger networks. The first stage comprises of a depth-first tree search with implicit enumeration and the second carries out a continuous optimisation of the network found in the first

stage. An initial network is obtained by using the heuristic rule of Ponton and Donaldson (1974), see section 2.3.2.4, which works very effectively within the implicit enumeration scheme to remove redundant branches of the tree. Maximum energy recovery, within the constraints of the match generator, is ensured by including this as a problem constraint. Any stream splits must be determined before the synthesis begins.

Jezowski (1981) and Jezowski et al (1983) abandon the 'depth-first' solving techniques in favour of an ordered search method, suggested by Nilsson (1971), to solve the problem as formulated by Grossmann and Sargent (1978). The lower bound on future cost is improved by adding a term for heat exchanger costs to the utility costs. However, the method is not effective because, like the 'breadth-first' methods, data storage requirements are unrealistic.

2.3.1.5 The optimal control method

Grossmann (1975) formulates the heat exchanger network synthesis problem as one from optimal control. Initially, the network is modelled as a multi-stream heat exchanger, divided along its length into a number of switching lengths, in which sets of pairs of streams can be matched. The Kuhn-Tucker conditions for optimality are applied to formulate a problem in which only the number of switching lengths must be specified. Unfortunately, the numerical solution is unobtainable in practice due to the large number of variables, the complexity of the equations and the ill-conditioning of the problem. A simplification is described in which a group of multi-stream heat exchangers, each with only one switching length, is used with utility exchangers inbetween. This reintroduces the com-

binatorial problem and makes the determination of the gradient of the objective function difficult. Neither of these methods has been demonstrated in practice and their use of exchangers of fixed length at each stage is questionable.

2.3.2 Heuristic methods

Heuristic methods rely on procedures, found, by experience, to cope with most problems. They can be divided into the following groups:

a) The heuristic structuring method.

b) Temperature interval methods.

c) Evolutionary methods.

d) Rule based methods.

These methods work well on a fair proportion of problems and, indeed, they are often used as a starting point for other methods. However, there is no guarantee of optimality.

2.3.2.1 The heuristic structuring method

Masso and Rudd (1969) suggest a set of five heuristics, to which adjustable weights are assigned, that are used to determine which pair of streams should be matched at any point during the synthesis. Heat exchanger networks are built up sequentially with these matched pairs until the networks are complete. At every stage the weights given to each heuristic are varied in an attempt to find the set

that gives the best over-all strategy. This strategy clearly only determines the value of the integer variables and an extra heuristic, usually to maximise heat transfer in any particular match, must be applied for the continuous variables.

The convergence on a stable set of weights is a combinatorial problem, as the weights at one level depend strongly on those at other levels, making the method unsuitable for large problems. In addition, no satisfactory method is given to establish convergence. The hope that a strategy could be transported from one problem to many others is not realised and the need to include a random factor in the set of heuristics indicates a lack of a comprehensive set of heuristics. Indeed, the method can only be as good as the heuristics employed.

2.3.2.2 Temperature interval methods

These methods partition the synthesis problem into a series of sub-problems delimited by the stream inlet and outlet temperatures corrected for any minimum temperature difference. Each sub-problem is then solved independently and the resulting sub-networks joined together to form a network on which considerable evolutionary work must be carried out.

The design of sub-networks is all important since continuity between adjacent sub-networks simplifies the following evolutionary process. Two methods have been suggested which achieve similar results but one is simpler to apply, with the drawback of producing networks with more exchangers than the other.

The number of exchangers in the solutions obtained is

large, leaving a non-trivial combinatorial problem of finding a realistic network from the synthesized network.

Linnhoff and Flower (1978a), Linnhoff (1979), Boland and Linnhoff (1979), Linnhoff et al (1979) and Polley (1980b) describe a sub-problem design procedure which orders the hot and the cold streams into two lists, ordered according to heat capacity flowrate. The corresponding streams in each list are then matched (i.e. the n'th stream in the hot list is matched with the n'th stream in the cold list) until all the streams in one of the lists are matched. The remaining unmatched steams, if any, are then matched with the unsatisfied sections of already matched streams until all streams have at least one match.

Sub-problems which are difficult to solve are subdivided into yet more intervals until a satisfactory level of heat integration is obtained. Utility exchangers are then added to complete the sub-networks and the intervals joined together. Pairing of hot and cold utility heat exchangers is then attempted, across sub-problem boundaries, to create further process exchangers. de Mateo (1979) offers a slight modification to this method in which the lists are ordered according to heat content and are updated after each match. However, extra complications are introduced by having to ensure that at least one of the matched streams has not been paired previously.

Su (1979), Challand et al (1981), Colbert (1982) and Su and Motard (1984) use a similar technique for sub-problem design but exclude the matching of streams from the longer ordered list with residuals of already matched streams. This removes the need to ensure the thermodynamic feasibility of these matches but makes the need for the subdivision of sub-problems more likely. Thus more exchangers are included in the design and an even larger

evolutionary stage is required. This effect is counteracted, in part, in the later works by the use of heat exchangers with minimum temperature differences less than that indicated by the level of energy recovery, i.e. maximum energy recovery is not achieved.

2.3.2.3 Evolutionary methods

Evolutionary methods offer aids to enable the development of an existing network. This is carried out by making small changes in the network structure and considering the resulting total cost alteration. Examples of possible network changes are:

a) Reversing the order of two heat exchangers on one stream.

b) Adding or deleting a heat exchanger.

c) Adding or deleting a stream split.

d) Changing the temperature range of a stream split.

e) Merging of heat exchangers matching the same two streams.

All of these methods require a large manual input and rely on the engineers experience. A large calculation load is often involved as each configuration, once accepted, must usually be optimised before proceeding to the next modification. This approach to synthesis follows the standard path of process improvement used for many years and, hence, is not widely reported.

M^{c}Galliard and Westerberg (1972) develop a bounding method

based on Lasdon's two-level algorithm in an attempt to remove the need for parametric optimisation of networks. All the structural changes are suggested by the user and the procedure calculates an upper and lower cost-bound for the new configuration. An iterative process is carried out in an attempt to make these bounds converge and enable a choice to be made about the suitability of the modification. However, the heat exchanger network problem is non-convex and so this iterative procedure cannot be guaranteed to converge.

Shah and Westerberg (1975) suggest a procedure based on points (a) to (d) above. The general properties - completeness, reversibility and efficiency - that any evolutionary method should feature are described along with an approximate optimisation method which takes advantage of the structure of heat exchanger networks to decrease the apparent problem size. No algorithm is suggested which will determine when a particular alteration should be carried out and only the highly combinatorial process of checking as many of the networks derivable from the present solution as time or money will allow is proposed.

2.3.2.4 Rule-based methods

The following methods are based on applying a rule or small set of rules to generate a network in a sequential manner. The methods are extremely quick to use and in a good number of cases are effective but they can go drastically wrong.

Ponton and Donaldson (1974) show that the heuristic of matching the hot stream with the hottest inlet temperature with the cold stream with the hottest outlet temperature and transferring as much heat as possible is very effec-

tive. Repeated application of this rule leads often to good networks at very little computational cost. Additionally, if a tree structure is set up where the hottest hot stream can only be matched with the hottest or second hottest cold stream, the results are improved further. Unfortunately, when this heuristic was put forward, the significance of problem pinch points and utility placement was not fully understood. Thus, in its original form, the heuristic places hot utility exchangers in positions far from optimal and thus has a tendancy to use too many exchangers.

Wells and Hodgkinson (1977) use the heuristic of Ponton and Donaldson (1974) to obtain an initial network and suggest heuristic guidelines to evolve it into a better one. No procedure is supplied to indicate when and where to apply these additional heuristics.

Grimes (1980) and Grimes et al (1982) develop a synthesis method based on a sequential generation of a network where, at each stage, the exchanger with the smallest mean driving force is selected, assuming that each match considered transfers the maximum amount of heat and completes at least one stream's requirements. This results in very badly distributed driving forces which must be extensively evolved to obtain realistic networks.

Zachoval and Konecny (1982) list three heuristic rules and four 'suggestions', based on the heuristic of Ponton and Donaldson (1974), to be used in the generation of a small set of networks from which the best is chosen. A novel feature is the inclusion of non-counter-current exchangers and a positive attempt to equate the number of matches with the number of heat exchangers. Minimum temperature differences are calculated for each match by using the results obtained from the analysis of a one-hot-stream,

one-cold-stream problem rather than considering the complete network.

Rathore (1982) suggests ordering all the hot and all the cold streams according to heat capacity flowrate and matching the corresponding elements in each list. Then the residuals are matched with any remaining streams and the network completed with utility exchangers. This procedure is very fallible and of questionable use.

Muraki and Hayakawa (1982) use an algorithm based on the heuristic of matching, at each stage, the cold stream with the highest outlet temperature with the hot stream giving the largest value of heat transfer divided by cost.

2.3.3 Thermodynamic methods

These methods fall into three classes:

a) Exergy methods.

b) General thermodynamic argument methods.

c) The pinch design method.

2.3.3.1 Exergy methods

Many synthesis methods have been suggested which attempt to minimise the dissipation of available energy of which nearly all use a graphical representation of the problem to aid solution. These are described in section 2.3.4. However, one group of workers has attempted, unsuccessfully, to solve the problem more directly.

Nishitani et al (1982) consider the problem of network synthesis by attempting to obtain an optimum balance between total heat transfer area and exergy loss. The calculation of the differential of total heat transfer area with hot stream temperature is found to be impossible and so a procedure is presented to minimise the surface area for a given exergy loss based on a heat content diagram (see section 2.3.4.1).

2.3.3.2 General thermodynamic argument methods

Linnhoff (1979), Boland and Linhoff (1979), Linnhoff et al (1979) and Flower and Linnhoff (1980) describe a method which uses basic thermodynamic and topological arguments to decrease the combinatorial aspect of network synthesis problems. Three basic feasibility criteria are proposed based on each stream meeting its target temperature, each network having the theoretical minimum number of matches and each heat exchanger having a positive heat load. These criteria are used to screen all the possible network combinations using a stated amount of utilities and exhibiting no stream splits. Extra, problem dependent, arguments are outlined to decrease the problem space even more when included in a manual method. Finally, when all these criteria have been applied, each of the remaining feasible networks is generated and considered.

Unfortunately, if no networks exist which solve the problem within the stated constraints, no solution will be forthcoming. In particular, the objective of the ensuring the theoretical minimum number of matches is always used is very limiting, expecially when applied with the constraint of no stream splitting. This problem is exacerbated when non-counter-current heat exchangers are considered.

2.3.3.3 The pinch design method

The pinch design method is more of a philosophy than a complete solution to network synthesis. Linnhoff and Turner (1981), the I.Chem.E. user guide (1982), Wood (1982), Dyson and Kenny (1982), Linnhoff and Senior (1983), Linnhoff and Hindmarsh (1983) and Linnhoff (1983) develop a procedure which partitions the problem at its pinch point(s) (see section 1.3) and starts designing away from these points. A method is given which indicates when stream splitting is required, based on the population of hot streams versus cold streams and their relative heat capacity flowrates. This method also indicates which streams can and cannot be matched adjacent to the pinch point, if maximum energy recovery is to be achieved. Networks with the minimum number of counter-current exchangers are aimed for, by ensuring each match brings at least one stream to one of its terminal temperatures. Once the initial matches have been made surrounding the pinch point, no strategy is given to synthesize the rest of the network nor is an algorithmic procedure given to choose between the various designs surrounding the pinch points, if more than one exists. Additionally, if no pinch exists, there is no starting point for the synthesis process. However, as a philosophy indicating how to start network synthesis, it is very successful.

2.3.4 Graphical methods

Graphical methods can be divided into two main groups:

a) Heat content diagram methods.

b) Heat availability diagram methods.

Both methods, as is common with many graphical techniques, are not easy to put into algorithmic form. The heuristic rules suggested must not be applied rigorously but up to some ill-defined limit. Thus these methods cannot be described as fully quantitative as they retain a considerable qualitative element. Indeed, moving from the graphical representation, with its human interaction, to more algorithmic approaches always removes the strength of these methods, namely, their ability to describe concepts easily to the user.

2.3.4.1 Heat content diagram methods

Heat content diagrams are very useful pictorial representations formed by drawing blocks denoting each individual stream on a graph where the abscissa denotes relative heat capacity flowrate and the ordinate denotes temperature (see figure 2.2). The hot and cold streams are plotted separately; the hot streams above the horizontal axis and the cold streams below. Single-component phase changes are not easily represented but, since most methods using the diagram are manual, some symbol can be used to signify their presence.

These diagrams can be used to give a graphical picture of any proposed or existing network and to suggest where improvements might be made. In addition, they can be used much more extensively as the basis of a synthesis method. Generally, these synthesis methods aim to minimise the total heat transfer surface area and in consequence lead to excessive numbers of exchangers and stream splits. Thus an evolutionary step must follow in order to achieve realistic solutions with all the problems this entails (see section 2.4). Problems also arise if heat transfer coefficients between different pairs of streams span a

large numerical range because the minimum area algorithms fail under these conditions. Additionally, the algorithms cannot cope with disallowed matches between streams.

Nishida et al (1971) use a heat content diagram synthesis method within an iteration loop in an attempt to find the optimal utility heat load. Initially, a utility heat load is assumed, then a process heat exchanger network is devised, an external utility network added and the resulting fixed structure optimised. The resulting utility heat load is then used to control the iteration loop.

The synthesis is based on the application of a set of rules which divide the blocks in the heat content diagram into segments and match them. The extent to which these rules should be used is problem dependent and not defined. The rules are derived from five theorems based on the assumption of fixed over-all heat transfer coefficient between all streams and a linear relation between heat transfer surface area and cost.

Wells and Hodgkinson (1977) outline a method similar to that described by Nishida et al (1971) with an extra stage grouping the blocks within the heat content diagram in an attempt to obtain more realistic networks. However, an evolutionary stage is still required.

Nishida et al (1977) use a minimum area algorithm similar to that described in Nishida et al (1971). However, the iteration loop is no longer present and utility exchangers can be placed anywhere in the network. The utility usage is set to the minimum possible for a given minimum temperature difference. Evolutionary rules are given (see section 2.4.1) to enable the minimum area solution to be converted to a realistic solution but cannot guarantee optimality.

Nishimura (1980) uses a variant of the heat content diagram where temperature is plotted along the abscissa and relative heat capacity flowrate is plotted along the ordinate. Hot streams are still drawn as blocks above the horizontal axis and cold streams below (see figure 2.3). These diagrams are used to consider the matching of one hot and one cold stream, one stream (say cold) against a set of other streams (say hot) and, finally, a set of hot streams against a set of cold streams. The maximum principal of Pontryagin is used to attempt to find the conditions under which minimum surface area results. Unfortunately, the problem of matching several hot streams against several cold streams could not be solved in the general case; the other two cases being of little interest. However, the method does account for variable over-all heat transfer coefficients.

Nishitani et al (1982) use a method similar to that used by Nishida et al (1977) in a procedure balancing loss of available energy (exergy) against total heat transfer surface area (see section 2.3.3.1).

Hama and Matsumura (1982) use Nishimura's solution (1980) to the problem of the matching of one stream against a set of other streams to devise networks by a dynamic programming method. Heat content diagrams are drawn representing these networks and the diagrams used to suggest simplifications that might be made. However, no algorithms are suggested to indicate automatically the alterations to be made.

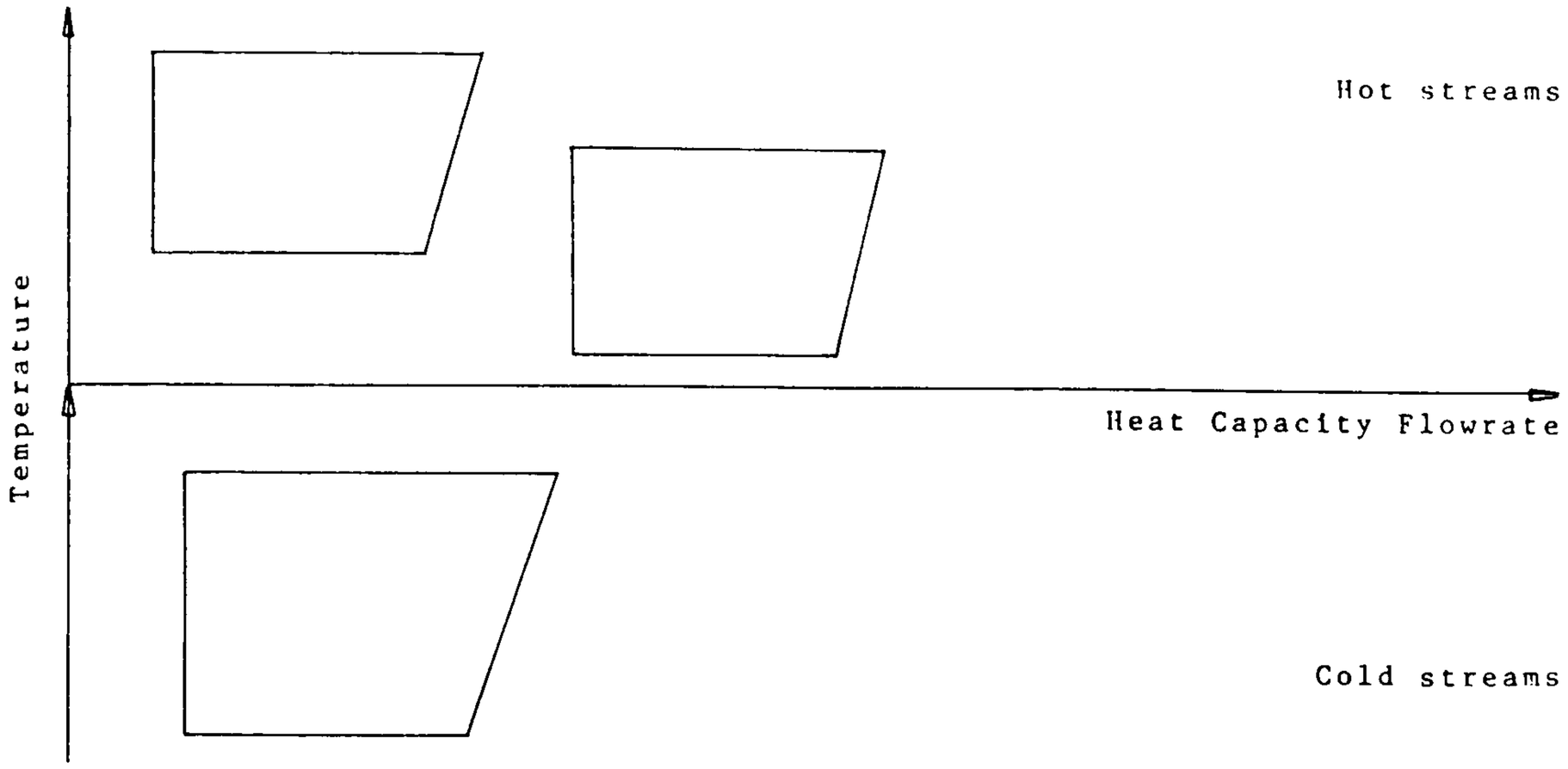


Figure 2.2 A standard heat content diagram

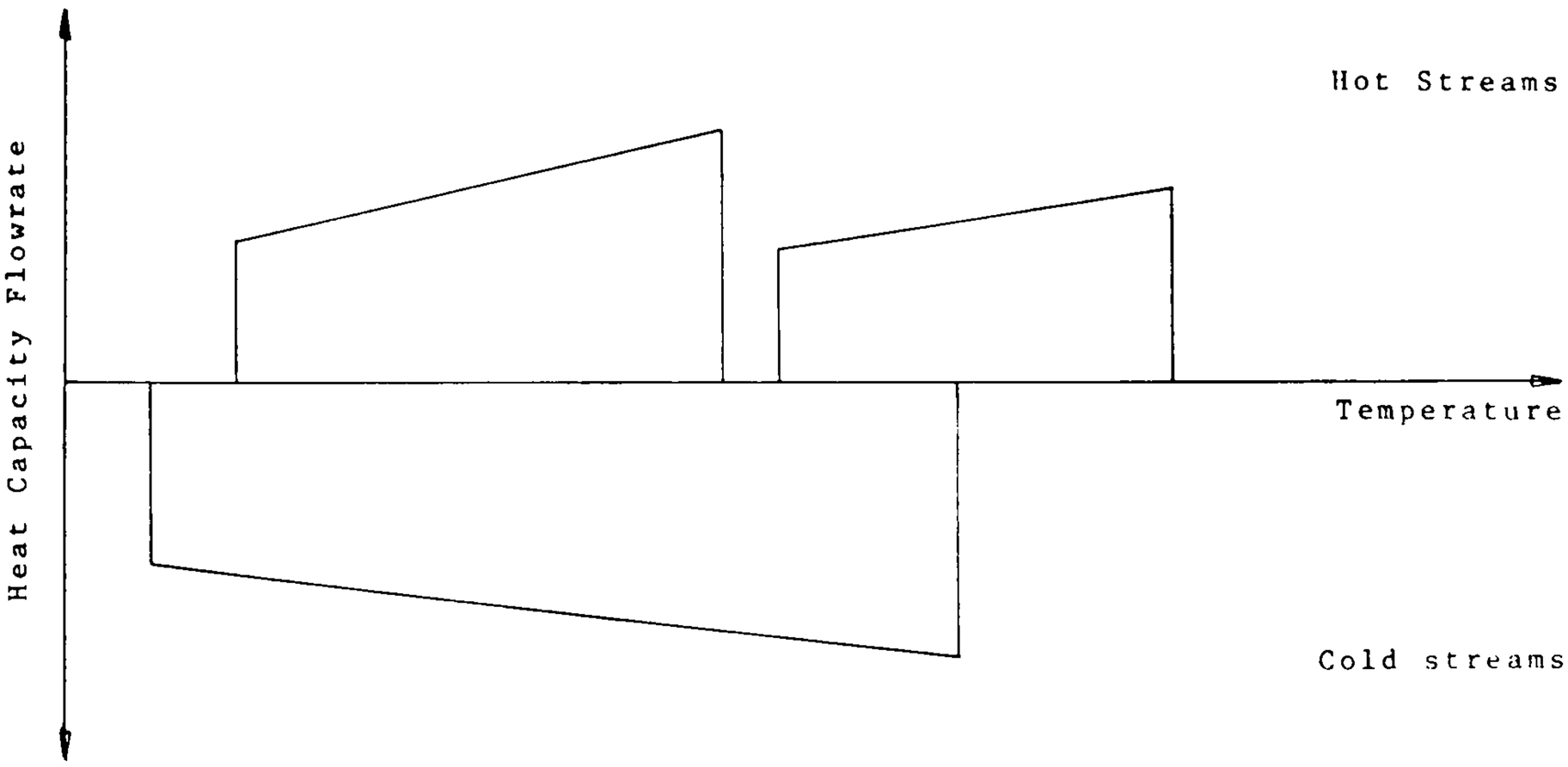


Figure 2.3 Nishimura's heat content diagram

2.3.4.2 Heat availability diagram methods

Heat availability diagrams have been used for many years and appear in three forms. All forms have relative enthalpy plotted along the abscissa and temperature, in some guise, plotted along the ordinate (see figures 2.4a-2.4c). This gives rise to the diagram's alternate names - the temperature/enthalpy diagram or T/Q diagram.

Heat availability diagrams can be considered as the integral of the heat content diagram discussed above (section 2.3.4.1) since the heat content of a stream is equal to the integral of its heat capacity flowrate between its inlet and outlet temperatures, i.e. in standard notation:

$$Q = \int_{T_1}^{T_2} \dot{m} C_p dT.$$

Many of the synthesis methods using heat availability diagrams are based on the same fundamental assumptions as the heat content diagram methods because of this connection and, similarly, have the same failings.

The first version of the diagram, attributed to Whistler (1948), plots each stream individually on the diagram (see figure 2.4a). A match between two streams is denoted by drawing one stream above another. Stream splits are denoted by replacing single lines for the relevant streams by a line for each branch. Thermodynamic feasibility is maintained by ensuring the hot streams are always above the cold streams on the diagram. A complete network must obviously have each stream serviced, i.e. each line paired along its whole length.

The second version of the diagram plots the composite streams (see figure 2.4b), giving an over-all view of the

synthesis task. This representation is often used to achieve minimum area solutions similar to the heat content diagram method.

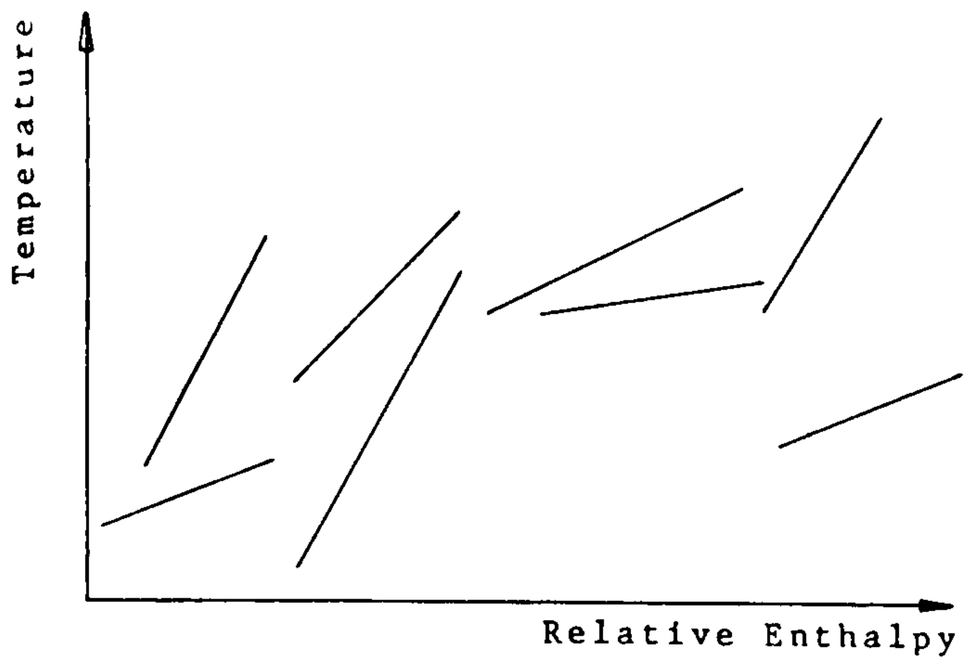
The final version of the diagram does not plot temperature directly on the ordinate but Carnot efficiency (see figure 2.4c). This representation is used in the methods that create networks which minimise the available heat (exergy) loss.

Hohmann (1971), Hohmann and Lockhart (1976) and Nash et al (1978) describe the use of the temperature/enthalpy diagram as a conceptual aid for the manual design of heat exchanger networks. Targets for minimum utility usage, minimum number of exchangers and minimum total heat transfer area are used to measure the performance of any particular network. The suggestions which indicate the streams to match are generally qualitative. Two algorithms are given to obtain minimum area networks, one for equal OHTC's and the other for differing OHTC's, but the results exhibit the usual excess of heat exchangers.

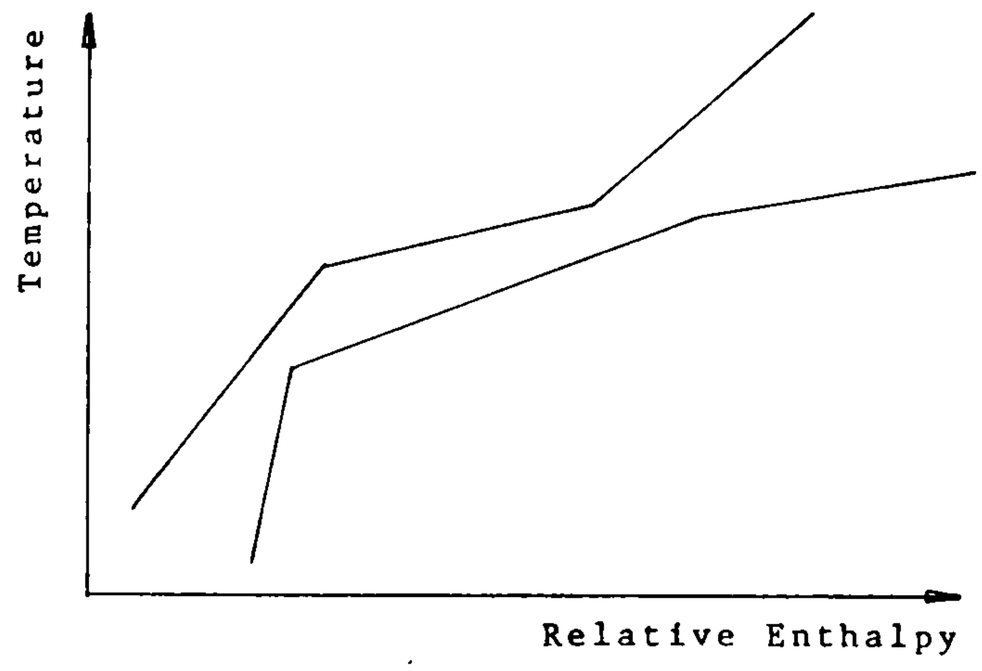
Umeda et al (1978), Umeda et al (1979) and Shiroko and Umeda (1981) use a similar, equal OHTC, minimum-area algorithm to which an evolutionary procedure is added. The optimal result derived by Nishimura (1980) for the one-to-several problem is applied to the synthesis problem with differing OHTC's for which it is not generally suited.

Rev and Fonyo (1982) study a three-process-stream problem using a Carnot efficiency diagram and attempt to draw general conclusions from the results. The results cannot be successfully applied universally and, in the cases for which solutions are found, stream splits are actively promoted rather than used when essential.

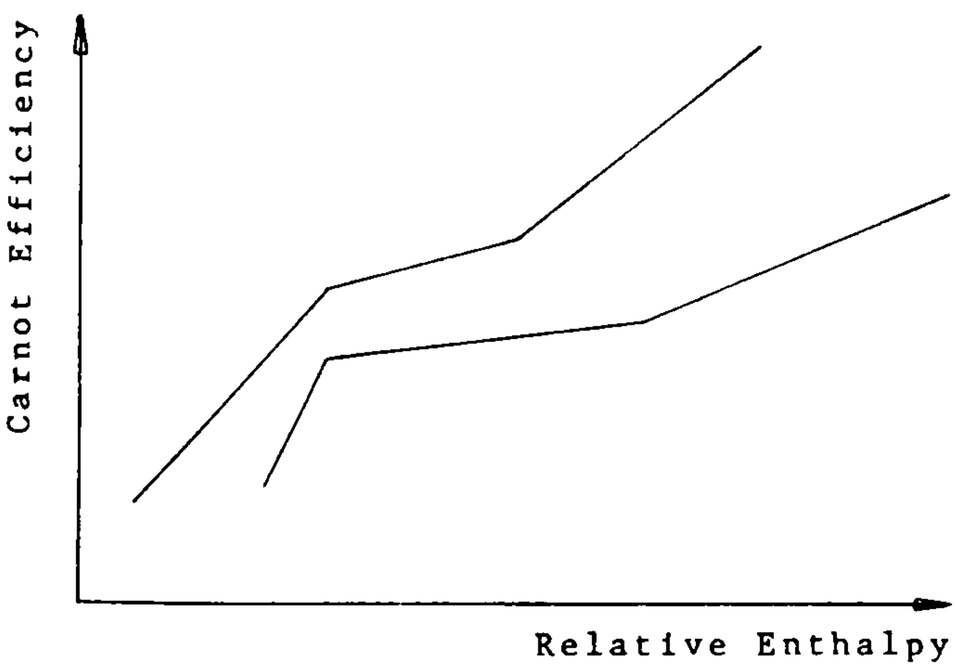
Naka and Takamatsu (1982) use the Carnot efficiency diagram to partition the synthesis problem into intervals. Vertical lines are drawn on the diagram at each discontinuity in the gradients of the two composite lines and the areas inbetween taken as sub-problems. A procedure for designing a network for each sub-problem is then followed similar to the temperature interval method (see section 2.3.2.2). Note, however, that the total heat content of the hot streams is equal to the total for the cold streams in each interval, unlike the temperature interval method, which leads to fewer utility exchangers. The method suffers from the same problems as the temperature interval method; i.e. networks with too many exchangers are synthesized and network evolution is required.



a : Individual streams.



b : Composite streams.



c : Carnot efficiency.

Figure 2.4 Heat availability diagrams

2.3.5 Summary

The mathematical methods, in general, are the most amenable to automisation and the most promising of these are the linear programming and tree-searching approaches. However, both have drawbacks; the linear programming algorithm results must be evolved into realistic networks by merging, which often creates a plethora of stream splits; the tree-searching algorithms require a procedure to introduce stream splits and a heuristic to determine the amount of heat transfer in a match. The adaptive random search method and the structural parameter methods, although theoretically very powerful, are inefficient in practice and the optimal control method is unrealistic and unproven.

The heuristic methods, by their very nature, are fallible and bring no guarantee of optimality. The heuristic structuring method is very dependent on the basic rules used and synthesis 'know-how' has not been shown to be transferable from one problem to another. The temperature interval methods are good for finding maximum energy recovery designs but utilise too many exchangers and require an evolutionary stage. Evolutionary methods are generally undirected and satisfactory procedures to determine which evolutionary modification to apply at any stage are not likely to be forthcoming. Rule-based methods are very efficient at obtaining initial estimates of suitable networks but do not readily offer alternative networks.

The thermodynamic methods are not readily described as a class. The optimal area/available energy loss formulation remains unsolved. The thermodynamic argument methods often fail to locate a solution because the constraints applied decrease the search space so dramatically a solution may not exist. The pinch design method is suitable

as a design philosophy but is not fully developed into an algorithm.

The graphical methods lose much of their conceptual power when automated and the results obtained require extensive evolution to obtain realistic networks.

2.3.6 Conclusions on heat exchanger network synthesis

The linear programming and tree-searching methods are the most suitable for automatic synthesis of heat exchanger networks because they are reasonably efficient and minimise the need for evolutionary stages.

The linear programming methods cannot incorporate the number of physical elements in the design directly into the objective function and no real control can be kept on the number of stream splits as these appear automatically as part of the post-synthesis merging step. However, a solution is guaranteed and the implicit incorporation of stream splits, which is a drawback, might also be used to indicate essential steam splits.

The tree-search methods can include the number of physical elements in the design into the objective function and stream splits can be kept under strict control. Indeed, an extra procedure to introduce them is essential. Unfortunately, a solution cannot be guaranteed because essential stream splits are not necessarily included and network interactions are not fully accounted for as matches are made sequentially rather then simultaneously.

The need to minimise primarily the number of streams splits and, less importantly, the number of heat exchanger shells indicates a tree-searching procedure must be used.

This approach also permits the pinch design philosophy, which has shown itself to be applicable, to be incorporated. The solution of this approach requires the development of an algorithm to introduce stream splits in an efficient manner.

2.4 POST-SYNTHESIS NETWORK EVOLUTION

Most synthesis methods produce networks which achieve very high energy recoveries but many do so with networks containing a large number of exchangers. Since the cost per unit area of heat exchangers drops with increasing size, it is more economical to utilise a given heat transfer surface area among a small number of exchangers rather than a larger number of smaller exchangers. Thus, many post-synthesis evolutionary development methods aim to decrease the number of heat exchangers while maintaining the heat transfer surface area approximately constant.

Evolutionary methods cover a wide range of applications from a fairly minor 'tidying-up' of a network with amalgamation or deletion of one or two exchangers through to procedures to merge tens, even hundreds, of very small exchangers. In practice, present methods are efficient at performing the former task but the large combinatorial nature of the latter task results in a problem as large, if not larger, than the original synthesis problem.

Most network evolution is carried out in an ad hoc manner and without a strong link between alterations attempted and the objective. This results in the area being poorly documented. The more structured evolutionary methods are outlined below.

2.4.1 Structured evolutionary methods

Hohmann (1971) and Hohmann and Lockhart (1976) describe the significance of heat exchanger load 'loops' in relation to the number of heat exchangers. A heat load 'loop' is a path drawn on a process flowsheet along streams, using heat exchangers as means to transfer the path from one stream to another, in such a way that a closed circuit is obtained (see figure 2.5). Any heat load 'loops' that exist can often be broken by transferring heat around the 'loop' until the load of one exchanger goes to zero and the heat exchanger ceases to exist. Thermodynamic checks must be made to ensure feasibility but this is often maintained, leaving a network with one less exchanger. If the heat transfer surface area has not been increased by a large factor, this new network will often exhibit a much lower cost than the original.

Nishida et al (1977) suggest three evolutionary rules to supplement their minimum area algorithm. Such algorithms result in networks with many small exchangers and a surfeit of stream splits and these must go through an evolution step in order to obtain realistic solutions. A hierarchy is suggested for these rules but the extent to which any of them is applied is problem dependent. Umeda et al (1978) outline a similar evolutionary procedure to improve the results of their minimum area algorithm.

Linnhoff and Flower (1978b) outline a method introducing the concept of 'freedom' of a heat exchanger. Ten rules are supplied to facilitate the maintenance of thermodynamic feasibility during configuration alterations with the aim of decreasing the number of heat exchangers. However, no clearly defined strategy is given to determine how to apply these rules. Linnhoff et al (1979) show the link

between the network relation of Euler and heat load 'loops' and, hence, the importance of breaking these 'loops'.

de Mateo (1979) uses the rules suggested by Linnhoff and Flower (1978b) to develop a procedure to remove heat load 'loops' which contain either two process heat exchangers or two process and two utility heat exchangers. Heat load 'loops' composed of a larger number of heat exchangers are not considered and so the procedure can be considered of limited use.

Su (1979) and Su and Motard (1984) develop an algorithmic method for the isolation of heat load 'loops' and for the determination of an order in which they should be broken, based on the number of exchangers in the 'loop'. This method has been used extensively by Challand et al (1981) and Colbert (1982) to improve networks obtained from a temperature interval method (see section 2.3.2.2).

Grimes (1980) and Grimes et al (1982) propose an evolutionary method based on heat load 'loop' breaking to obtain a minimum match solution and then an alternate creation and deletion of a 'loop' to create further minimum match solutions. This approach to obtaining alternate near-optimal networks is inefficient at locating networks with widely different topologies because each new network has just one new heat exchanger added and one deleted.

The I.Chem.E. user guide (1982) shows that thermodynamic feasibility can often be restored, if lost by heat load 'loop' breaking, by tracing a path from the infeasible exchanger to a utility cooler and heater to form a heat load 'path'. A heat load can then be added to the utility exchangers to decrease the load on the offending process exchanger and re-establish its feasibility with the penal-

ty of decreased energy recovery.

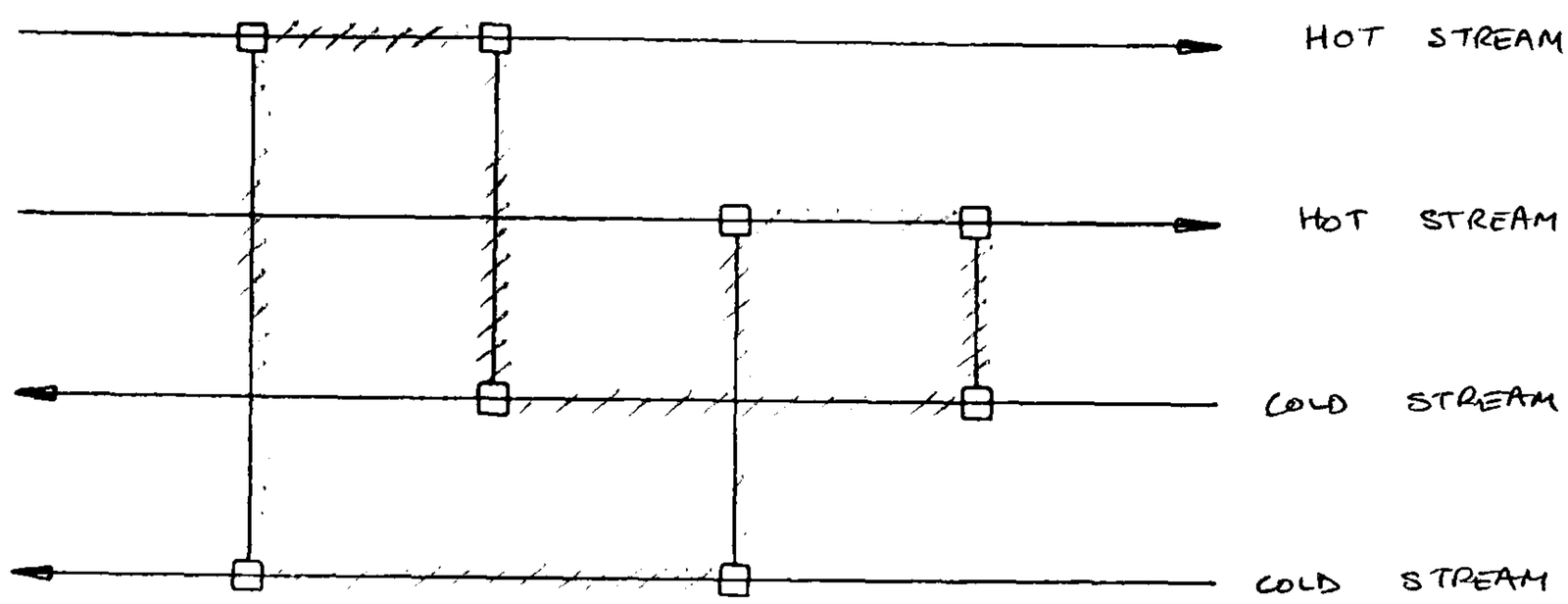


Figure 2.5 Heat load 'loops'

2.4.2 Conclusions on post-synthesis evolution

Evolutionary methods need a clearly defined objective that can be easily formulated and a procedure which will apply itself directly to obtaining that objective.

The methods that are most successful use the number of heat exchangers as an indication of optimality (i.e. the objective is to minimise the number of heat exchangers). The optimisation is carried out by identifying heat load 'loops' and breaking them.

Methods that use cost as an indication of optimality are much less efficient because the effect on the objective function of each possible network alteration is not known until it has been carried out.

Networks containing a large number of heat exchangers and, hence, heat load 'loops' can be evolved in an enormous number of ways. The solution of these problems is combinatorial and difficult to direct. The available evolutionary methods are most efficient when dealing with modifications to near-optimal networks or when the networks have been carefully structured to facilitate evolution.

2.5 SUMMARY OF CONCLUSIONS

A minimum utility usage target can be calculated before synthesis starts and most good networks achieve this target. The procedures given in the literature to calculate this target are not efficient if they are applied at every step in an automatic synthesis method and so a more efficient algorithm is required. A method for calculating the optimal minimum temperature difference is needed to enable the minimum utility usage to be found.

The target of minimising the number of heat exchangers in a network is a valid one but often cannot be calculated before synthesis takes place. However, tree-search methods can incorporate this aim very easily and exhibit the correct hierarchy of objectives by, first, maintaining control over the number of stream splits and, second, minimising the number of heat exchangers. Indeed, these methods are usually formulated so as to ignore the possibility of stream splitting unless it is specifically indicated and thus a procedure is required to identify automatically the need for stream splits and indicate this to the tree-searching methods.

Tree-searching methods can cope with different types of exchangers, which the other main method cannot, and can incorporate the pinch design method philosophy which has proved itself applicable. The results obtained are well developed and need little evolution. The existing evolutionary methods based on 'loop' breaking are capable of dealing with such networks.

CHAPTER THREE

INCORPORATION OF NON-COUNTER-CURRENT HEAT EXCHANGERS INTO A SYNTHESIS PROCEDURE

3.1 INTRODUCTION TO THE WORK IN THE FOLLOWING CHAPTERS

In the previous chapter it was shown there are several areas in the field of heat exchanger network synthesis that require further investigation. The first of these, the incorporation of heat exchangers that do not exhibit counter-current flow, is considered in this chapter. This study leads to the development of two procedures that aid the synthesis of heat exchanger networks composed of such exchangers. These procedures form part of a complete synthesis algorithm which is developed later on in the thesis.

Two other topics for consideration, the automatic generation of essential parallel stream splits when maximum energy recovery is required and the traditional synthesis of heat exchanger networks, are described in the subsequent chapters. Both problems are solved by using depth-first branch-and-bound tree searches. Chapter four expands on the description of these methods given in the above literature review and discusses the problem in general terms by outlining the fundamental parts of the problem.

Chapter five describes in detail the solution of the parallel stream-splitting problem and chapter six develops the algorithms required to solve the network synthesis problem. The procedures that are developed immediately below are incorporated into these synthesis algorithms.

Chapter seven demonstrates the practical value of these developments by applying the algorithms to several numerical examples.

3.2 INTRODUCTION TO NON-COUNTER-CURRENT HEAT EXCHANGERS

Previously, most workers in the field of heat exchanger network synthesis have assumed that all exchangers in the networks synthesized are counter-current. However, in most installed networks this is not the case. Furthermore, the use of heat exchangers with non-counter-current flow schemes often results in networks which contain many more heat exchanger shells than is indicated by the provisional flowsheets created by synthesis procedures. This is particularly so when exchangers with small temperature differences are required, as is often the case in high energy recovery networks. To enable better networks to be synthesized that include these types of exchangers, a study is made in this chapter of the mathematical properties of their describing equations. However, before this study, a summary is given of the types of equipment that approximate counter-current flow and the reasons why such exchangers cannot be used in many cases.

3.3 COUNTER-CURRENT HEAT EXCHANGERS

Counter-current flow is approximated in the following types of heat exchangers:

a) Double pipe exchangers - suitable only for small duties

b) Gasketed plate exchangers - upper temperature limit ≈ 450K

c) Welded plate exchangers - not easily cleaned

d) Shell-and-tube exchangers - not easily cleaned and prone to internal leakage on the shell-side
   [designs with an equal number of multiple shell and tube-passes]

e) Shell-and-tube exchangers - often long, thin and uneconomical
   [designs with a single shell- and tube-pass]

f) Shell-and-tube exchangers - most common but expensive for high temperature efficiencies
   [designs composed of a train of non-counter-current exchangers]

g) Spiral exchangers - not easily cleaned and of limited pressure range

h) Lamella exchangers - not easily cleaned

(For further information see, for instance, I.Chem.E. user guide (1982)).

It can be seen that many duties can only be dealt with in an approximately counter-current manner by using a train of several shell-and-tube non-counter-current exchangers, especially when regular cleaning is required. In this situation, each shell usually has one shell-pass and two or more tube-passes (i.e. far removed from a counter-current flow scheme) and counter-current flow is approximated by the use of multiple shells. This has a marked

effect on the frequently used objective in synthesis procedures of minimising the number of heat exchangers since a single match often requires more than one piece of equipment.

The use of a large number of shells in the heat exchanger trains described above can prove to be impractical and so the aim of obtaining counter-current flow is discarded. Thus non-counter-current exchangers, similar to those used in the exchanger trains, can be used singly to build up a network.

Irrespective of how networks are built up, the use of non counter-current exchangers is of practical importance.

3.4 MEAN TEMPERATURE DIFFERENCES IN HEAT EXCHANGERS

The preliminary design of heat exchangers is made simpler by the use of an over-all mean temperature difference (ΔTm). This is dependent on the temperatures of the matched streams in the exchanger and the physical layout of the exchanger and, along with the over-all heat transfer coefficient (OHTC or U), enables the heat transfer surface area (A) to be calculated from:

$$A = Q/U\,\Delta Tm \qquad (3.1)$$

where Q is the heat load to be transferred.

The value of ΔTm is frequently found by applying a correction factor (F) to the mean temperature difference calculated for the counter-current exchanger carrying out the same duty. This temperature difference is given by the counter-current log-mean temperature difference (LMTD or ΔTlm):

$$\Delta T_{lm} = \frac{(T_{hot}^{in} - T_{cold}^{out}) - (T_{hot}^{out} - T_{cold}^{in})}{\ln\left[\frac{(T_{hot}^{in} - T_{cold}^{out})}{(T_{hot}^{out} - T_{cold}^{in})}\right]} \qquad (3.2a)$$

or, when $T_{hot}^{in} - T_{cold}^{out} = T_{hot}^{out} - T_{cold}^{in}$:

$$\Delta T_{lm} = T_{hot}^{in} - T_{cold}^{out} \qquad (3.2b)$$

and so:

$$\Delta T_m = F . \Delta T_{lm} \qquad (3.3)$$

The precise relationship between the correction factor, F, and the temperature of the streams in a heat exchanger depends on the type of exchanger but in all non-counter-current exchangers the trend is for F to decrease as the temperature difference between the matched streams decreases. Indeed, as the temperature difference decreases the value of F decreases increasingly rapidly to the point where the exchanger becomes difficult to control. Thus many designers impose a lower limit on the value of F; usually in the range 0.75 to 0.8 for shell-and-tube heat exchangers.

3.5 INTRODUCTION TO THE USE OF LMTD CORRECTION FACTORS IN SYNTHESIS

The objective of this work is to produce networks which incorporate as few heat exchangers as possible in order to minimise cost. Thus the need to specify exchangers with F fixed at its lower bound (i.e. with the amount of heat exchanged in any particular exchanger maximised) is certain to occur.

The calculation of the unknown exchanger inlet and outlet temperatures, when given a fixed value of F, is an iterative process which, when coupled with the combinatorial nature of process synthesis, makes the study of these equations potentially fruitful. Any decrease in the number of iterations to determine the performance of an exchanger will lead to a large decrease in the over-all calculation burden. This has been achieved by developing a procedure which will find a pair of good initial estimates which bound the unknown temperatures.

However, further study of the lower bound on F shows that the figure 0.8 corresponds closely to the onset of a temperature cross. Thus, if the design of heat exchangers to such a value is required, the iterative process can be replaced by simply setting the two outlet temperatures equal.

3.6 DIMENSIONLESS NUMBERS

The study of shell-and tube heat exchangers is usually performed using a set of dimensionless numbers. This set comprises the LMTD correction factor (F), already mentioned, the temperature gap ratio (G), the heat capacity flowrate ratio (R), the temperature efficiency (P) and the number of heat transfer units (NTU). F and G are defined as follows:

$$F = \frac{Q}{UA\ \Delta T_{lm}} \qquad (3.4)$$

$$\text{and} \quad G = \frac{(T_{hot}^{out} - T_{cold}^{out})}{(T_{hot}^{in} - T_{cold}^{in})} \qquad (3.5)$$

R,P and NTU can be defined in two ways, forming two consistent sets, as follows:

$$R = \frac{(T_{hot}^{in} - T_{hot}^{out})}{(T_{cold}^{out} - T_{cold}^{in})} \qquad (3.6a)$$

$$P = \frac{(T_{cold}^{out} - T_{cold}^{in})}{(T_{hot}^{in} - T_{cold}^{in})} \qquad (3.7a)$$

$$NTU = \frac{UA}{\dot{m}Cp_{cold}} \qquad (3.8a)$$

$$R = \frac{(T_{cold}^{out} - T_{cold}^{in})}{(T_{hot}^{in} - T_{hot}^{out})} \qquad (3.6b)$$

$$P = \frac{(T_{hot}^{in} - T_{hot}^{out})}{(T_{hot}^{in} - T_{cold}^{in})} \qquad (3.7b)$$

$$NTU = \frac{UA}{\dot{m}Cp_{hot}} \qquad (3.8b)$$

Either of the above sets (3.6a to 3.8a or 3.6b to 3.8b) can be used in most circumstances. However, difficulties do arise when single-component phase changes occur in a

heat exchanger. In these cases all heat exchangers behave as counter-current exchangers and a set of simplifying modifications can be made to the following equations. Alternatively, if just one of the matched streams undergoes a single-component phase change, the equation set that gives a value of zero for R can be used, which leads to finite values of P and NTU.

The value of G can be obtained easily from the following relationship:

$$G = 1 - P(1+R) \qquad (3.9)$$

However, G has been employed rarely in the past with P used in preference. Nevertheless, the use of G is important in the understanding of the significance of the lower limit on F, as will be seen below.

3.7 THE SYNTHESIS (OR DESIGN) OF HEAT EXCHANGERS

The first stage of the synthesis of a heat exchanger involves the determination of the unknown inlet and outlet temperatures. Once these have been found, the heat transfer surface area can be determined with a knowledge of the heat capacity flowrates and the OHTC (i.e. a design rather than a simulation takes place). This can be done in two ways, either by calculating the number of NTU's or F. The expressions giving F as a function of R and P or R and G are easily converted to give NTU's or vice versa but, since F must be checked subsequently against its lower limit, the use of equations involving F is preferred. In the case where the value of F or A proves to be impractical, the duty must be dealt with in more shells of the same type or in a different configuration of exchanger. This problem can be overcome by either repeatedly adding

exchangers with F at its lower limit or A at its upper limit, which ever is relevant, until the duty can be satisfied or by ensuring the duty specified originally never results in exchangers which violate these limits. Heat exchangers that match the same two streams and which are synthesized on the basis of fixed values of F each have the same heat transfer surface area if the heat capacity flowrate ratio remains constant. Thus a realistic train results from such a strategy.

3.8 THE SIMULATION OF HEAT EXCHANGERS

Heat exchangers with fixed heat transfer surface areas are sometimes used in network synthesis because of the limits imposed by exchanger manufacture. In this case, in addition to the surface area, one inlet or outlet temperature on each stream is known and it is necessary to find the other two inlet or outlet temperatures. Thus expressions giving P and G as a functions of R and NTU are required. Expressions giving F are of little use in this problem, the simulation case, because an iterative procedure would be required to solve them.

An extensive description of this type of equation is given by Dodd (1980)

3.9 DERIVATION OF THE LMTD CORRECTION FACTOR

The derivation of LMTD correction factors depends on the following assumptions:

a) The heat exchanger is in a steady state.

b) The OHTC is constant.

c) The temperature versus enthalpy relationship is linear for both fluids i.e. the heat capacity flow-rates are constant.

d) Each fluid is well mixed i.e. the temperature is isothermal in any plane perpendicular to the over-all stream direction.

e) Each pass contains the same number of tubes.

f) The exchanger is adequately insulated from its surroundings.

Each of these items is true or approximated under the synthesis conditions, with the possible exception of (e). However, the streams under consideration can always be modelled by the use of sufficient temperature intervals within which the temperature versus enthalpy relationships are linear and between which the changes are small. As has been already described (see section 2.2.2), the problem of phase change in addition to single-phase heating or cooling of a stream is dealt with by partitioning the stream into completely separate parts and is therefore not relevant.

3.9.1 Single shell-pass exchangers

The value of F for a 1/2 shell-and-tube heat exchanger was given by Underwood (1934):

$$F = \frac{\sqrt{R^2+1}\ \ln[(1-PR)/(1-P)]}{(1-R)\ \ln\left[\frac{2-P(R+1-\sqrt{R^2+1})}{2-P(R+1+\sqrt{R^2+1})}\right]} \qquad (3.10a)$$

or, when R is equal to unity:

$$F = \frac{\sqrt{2}P}{(1-P)\ \ln\left[\frac{2-P(2-\sqrt{2})}{2-P(2+\sqrt{2})}\right]} \qquad (3.10b)$$

This also applies very closely for 1/n exchangers, where n is any even number (to be precise, the maximum relative error is 0.39% if $F \geq 0.75$, Dodd (1982)). Heat exchangers with an odd number of tube passes are much less common and will not be considered.

3.9.2 Multiple shell-pass exchangers

If shell partitions are allowed it may be possible to achieve counter-current flow (e.g. by using a 2/2 heat exchanger) and so the value of F becomes unity under all conditions, assuming no leakage. However, if the number of tube passes is greater than the number of shell passes a non-counter-current design results. Normally the number of shell passes will divide exactly into the number of tube passes (e.g. a 2/4 or 2/8 heat exchanger) resulting in no thermodynamic difference between the multi-shell exchanger and multiple 1/n exchangers. Heat exchangers with more shell passes then tube passes are rare and will

not be considered. It will be assumed here that multiple shell-pass exchangers will be built up from individual shell passes modelled as 1/n exchangers.

3.10 HEAT EXCHANGERS WITH FIXED LMTD CORRECTION FACTORS

Since the aim is to minimise the number of heat exchangers and hence cost when synthesizing a network and since each exchanger must have a lower limit applied to its LMTD correction factor, F, the design of heat exchangers with F fixed at this lower limit becomes important. The calculation of P given F and R from equation (3.10) is an iterative process. This process can be speeded up if a good estimate of the final solution can be calculated or, preferably, if two good estimates which bound the final solution can be found.

3.11 BOUNDING VALUES ON LMTD CORRECTION FACTORS

The maximum value that P can take for any given value of R is given by Pmax:

$$Pmax = 2/(R + 1 + \sqrt{R^2 + 1}) \qquad (3.11)$$

which occurs when F becomes zero (see appendix B).

Equation (3.10) can also be manipulated to show that the minimum value of P/Pmax, for a given value of F, occurs when R has a value of unity (see appendix C).

Therefore

$$1 > (P/Pmax) \geq (P/Pmax)\Big|_{R=1} \qquad (3.12)$$

These two observations lead to a simple procedure for the calculation of the actual value of P/Pmax.

a) Given a lower limit to be placed on F, use a suitable iterative method to calculate P from equation (3.10) for a value of R equal to unity.

b) Calculate Pmax for R equal to unity from equation (3.11) and hence $P/Pmax\big|_{R=1}$.

c) The results from (a) and (b) can now be used repeatedly to give bounds on the value of P/Pmax, and hence P, for any other value of R. Any suitable iteration technique can be used to find P to the required accuracy.

N.B. parts (a) and (b) are only carried out once. The calculation load decrease results from the repeated application of the results of these initial calculations to the solution of the problem for other values of R.

Experience has shown that the value of P/Pmax increases monotonically from R equal to unity towards R equal to zero and R equal to positive infinity (see appendix D). This property allows an even better lower bound to be calculated for P/Pmax. A table is created containing values of P/Pmax for various values of R and, on deciding which streams are to be paired in the current match, the value of P/Pmax is extracted for a value of R as close as possible to the one for the current match but always closer to R equal to unity. The value then replaces the bound given by $P/Pmax\big|_{R=1}$.

i.e.

$$1 > P/Pmax\big|_{R=A} \geq P/Pmax\big|_{R=B} \qquad (3.13)$$

where, if	$A > 1$	$A \geqslant B \geqslant 1$
else if	$A = 1$	$A = B = 1$
else	$A < 1$	$A \leqslant B \leqslant 1$

3.12 INTRODUCTION OF THE TEMPERATURE GAP RATIO INTO THE BOUNDING VALUES ON LMTD CORRECTION FACTORS

Equation (3.10) can be rearranged to give F as a function of G and R.

$$F = \frac{\sqrt{R^2+1}\ \ln\left[(R+G)/(1+GR)\right]}{(R-1)\ \ln\left[\dfrac{(R+1)(1+G)+(1-G)\sqrt{R^2+1}}{(R+1)(1+G)-(1-G)\sqrt{R^2+1}}\right]} \qquad (3.14a)$$

or, when R is equal to unity:

$$F = \frac{\sqrt{2}}{(1+G)\ \ln\left[\dfrac{2(1+G)+(1-G)\sqrt{2}}{2(1+G)-(1-G)\sqrt{2}}\right]} \qquad (3.14b)$$

A similar process to that applied to F as a function of P and R can be applied to give a minimum value of G, Gmin:

$$Gmin = \frac{\sqrt{R^2+1}-1-R}{\sqrt{R^2+1}+1+R} \qquad (3.15)$$

and a similar set of bounds to those given in (3.12) can be put on G by using the identity:

$$P/Pmax = (1-G)/(1-Gmin) \qquad (3.16)$$

to give:

$$1 > (G-1)/(Gmin-1) \gtrsim (G-1)/(Gmin-1)\Big|_{R=1} \qquad (3.17)$$

Thus, a similar procedure to that given above to find P can be written to find G, given F and R.

3.13 NUMERICAL RESULTS

The above bounding procedures, which give limits on the value of P or G for fixed values of F and R, were used to give initial ranges to a subroutine which locates simple zeros of functions. The subroutine used was C05AZF from the Numerical Algorithm Group's (NAG) FORTRAN library (see NAG(1983)). This uses a combination of linear interpolation, linear extrapolation and bisection to reduce the given range, in a series of iterations, to one which lies within a pre-set tolerance. The increase in the speed of convergence was considerable with the worst case being some 17% faster and with a typical improvement of about 30% (see table 3.1). The procedure to find P was found to out-perform that to find G.

A listing of the computer programs used is given in appendices E and F.

F →	0.75				0.8		
Tol →	0.001			0.0001	0.001		0.0001
R ↓							
10	7	6	5	6	7	5	7
5	6	6	4	6	8	4	6
3	8	6	5	6	5	6	6
2	6	5	4	5	7	4	5
1.5	7	4	4	4	7	4	4
1.1	7	2	2	4	8	2	4
0.9	7	3	2	4	8	2	4
0.7	7	4	2	4	7	4	4
0.5	7	5	4	6	7	5	6
0.3	8	6	5	7	7	6	7
0.2	8	6	6	6	8	5	6
0.1	9	6	6	7	9	7	7
Procedure	1	2	3		1	3	

1 Using the standard procedure for P, $0<P\leqslant Pmax$

2 Using the new procedure for G

3 Using the new procedure for P

Tol = Tolerance, defined such that the zero of the function is known to within at tolerance of $\pm$ Tol

Table 3.1 Number of iterations required to obtain covergence using C05AZF

3.14 THE RELATIONSHIP BETWEEN THE TEMPERATURE GAP RATIO AND THE LOWER BOUND ON LMTD CORRECTION FACTORS

Wales (1981) considers the use of the temperature gap ratio with reference to the LMTD correction factor equation. The graphs plotted of F versus G for various R show that when F is in the range 0.75 to 0.8 the temperature gap ratio is almost independent of R (see table 3.2) and that this region corresponds to the onset of an outlet temperature cross (i.e. G starting to go negative). Thus, an alternative to designing exchangers with fixed F may be to design exchangers with fixed values of G where G is approximately zero.

3.15 HEAT EXCHANGERS WITH A FIXED TEMPERATURE GAP RATIO

Using equations (3.9) and (3.11) an expression for P/Pmax can be derived.

$$P/Pmax = (1-G)(1+R+\sqrt{R^2+1})/2(1+R) \qquad (3.18)$$

On considering the value of P/Pmax as R goes to either of its two limits it is found that:

$$\text{as } R \doteq 0, \quad P/Pmax \doteq 1-G$$

$$\text{and as } R \doteq \infty, \quad P/Pmax \doteq 1-G \qquad (3.19)$$

This shows that any negative value of G must be used with bounds on the value of R because in the limit P/Pmax becomes greater than one. This is thermodynamically impossible and indicates that the value of F drops to zero at finite and non-zero values of R. Non-negative values of G can be used unbounded because F always remains at a finite and practical value. (In fact, the lowest value of

F under these conditions occurs when R is equal to unity and G equal to zero. F then takes the value 0.8023).

Table 3.3 shows that when G equal to zero is used the value of F is close to 0.8 and is therefore a practical alternative to designing with F fixed at this value

R	F=0.8	F=0.75
10	-0.01087	-0.02317
5	-0.01022	-0.03305
3	-0.00752	-0.03961
2	-0.00482	-0.04347
1.5	-0.00326	-0.04522
1.1	-0.00238	-0.04609
0.9	-0.00240	-0.04608
0.7	-0.00305	-0.04543
0.5	-0.00482	-0.04347
0.3	-0.00819	-0.03839
0.2	-0.01022	-0.03305
0.1	-0.01087	-0.02317

Table 3.2 Values of G for given values of R and F

R	F
10	0.8299
5	0.8170
3	0.8095
2	0.8052
1.5	0.8033
1.1	0.8023
0.9	0.8023
0.7	0.8031
0.5	0.8052
0.3	0.8109
0.2	0.8170
0.1	0.8299

Table 3.3 Values of F for given values of R with G equal to zero

3.16 FIXED-AREA EXCHANGERS

Kays and London (1964) rearranged equation (3.10) to obtain an equation explicit in P:

$$P = \frac{2}{1+R+\sqrt{R^2+1}\ \coth\left[NTU\ \sqrt{R^2+1}/2\right]} \qquad (3.20)$$

which, when combined with equation (3.9), gives an expression for G:

$$G = \frac{\sqrt{R^2+1}\ \coth\left[NTU\ \sqrt{R^2+1}/2\right] -1-R}{\sqrt{R^2+1}\ \coth\left[NTU\ \sqrt{R^2+1}/2\right] +1+R} \qquad (3.21)$$

These can be used directly to obtain the unknown temperatures given the heat transfer surface area.

3.17 CONCLUSIONS

Exchangers exhibiting counter-current flow cannot always be used in networks and hence other types of exchangers must be used for which a lower limit on the LMTD correction factor must be applied.

In the case where the lower limit on the LMTD correction factor is 0.8 or larger, the limit can be replaced by designing the exchangers with a fixed temperature gap ratio. A figure of 0.8 for the LTMD correction factor corresponds closely to a temperature gap ratio of zero.

When the lower limit for the LMTD correction factor is below 0.8 (usually down to 0.75) the use of a fixed temperature gap ratio is not straightforword and an iterative procedure to find the temperature efficiency is recommended instead. A new bound on the solution is developed which improves on the standard bound and hence improves the iterative procedure.

Fixed-area heat exchangers are best introduced during synthesis by using relationships based on NTU and R since these do not require iterative methods to obtain solutions.

CHAPTER FOUR

DEPTH-FIRST TREE-SEARCHING METHODS FOR THE SOLUTION OF INLP's, MINLP's AND MILP's

4.1 INTRODUCTION

The application of a depth-first branch-and-bound tree-searching method was shown in chapter two to be one of the best ways to find solutions to the heat exchanger network synthesis problem. Indeed, such methods are used effectively to solve a wide variety of problems in engineering, mathematics and management science.

This chapter expands on the general description of branch-and-bound methods given in chapter two and outlines the fundamental sections into which such methods can be divided. The integer variables that arise in the problems in this thesis are all binary in nature and so the extra problems associated with the determination of the integer variables when a wider range of values is possible will not be considered here.

4.2 TREE-SEARCHING USING DEPTH-FIRST BRANCH-AND-BOUND METHODS

Depth-first branch-and-bound tree-searching methods are used widely to solve integer non-linear programmes (INLP) and mixed integer linear and non-linear programmes (MILP and MINLP). In mixed integer programmes an extra optimisation procedure is required to supplement the basic tree search so that the values of the continuous variables can be determined.

Branch-and-bound methods can be split into four fundamental parts:

a) The upper bound on the objective function.
This bound is used to curtail the development of partial networks which are unlikely to lead to optimal solutions.

b) The lower bound on the additional 'cost' to complete a network.
This bound is added to the 'cost' for a partial network to give a lower bound on the total 'cost'. This value is then compared with the bound outlined in (a) to determine whether the partial network should be developed further.

c) The branching strategy.
This determines the order in which descendants of nodes in the search tree are expanded and, sometimes, how many of the nodes are expanded.

d) The stopping criteria.
These indicate when a solution, a goal node, has been found. The stopping criteria are an essential requirement of the solution procedure.

Calculation schemes to satisfy parts (a), (b) and (c) can be replaced by others with no change to the over-all final results as long as the bounds are strict (i.e. they cannot be violated) and the branching strategy is exhaustive (i.e. all nodes are considered). However, the efficiency with which the optimal solutions are isolated can vary widely from method to method.

Many of the problems to be solved by these methods have

very large search trees and thus an exhaustive search may not be feasible. In these cases it is crucial that a branching strategy is used which develops the most likely areas of the search tree first, in order to increase the probability of finding optimal solutions. Even when an exhaustive search is made, such a branching strategy is desirable because it leads to a rapid decrease in the upper bound on the objective function.

The use of a lower bound on the additional 'cost' to complete a network that is not strict is not to be recommended as no optimal solution may be found.

The four fundamental parts to branch-and-bound tree-searching methods will now be considered in more detail along with the determination of the optimal value of continuous variables in MILP's and MINLP's.

4.3 THE UPPER BOUND ON THE OBJECTIVE FUNCTION

The initial value of this bound can be supplied by the user, either as a number or as a heuristic to obtain a first solution. Alternatively, it can be set automatically to some suitably high number below which the solution almost certainly occurs. The starting value is important, especially if the branching strategy does not isolate a near-optimal solution quickly, because a large amount of effort may spent developing solutions with objective function values far greater than optimal.

Once a solution has been found with an objective function value less that the current upper bound, the bound is replaced by this new value. Thus the maximum depth of search decreases as the search progresses with the rate of decrease being dependent on the branching strategy.

The tree-searching procedures described in this thesis use both user-supplied and automatically generated initial values of the upper bound.

4.4 THE LOWER BOUND ON ADDITIONAL 'COST'

The procedure used to obtain the lower bound on additional 'cost' must be as accurate as possible in order to minimise the number of nodes of the search tree enumerated explicitly but must also be simple to calculate. Thus there is a balance to be struck between the tightness of the lower bound and the calculation complexity; if too lax, more nodes than neccessary will be enumerated and, if too complex, the computational load will be biased towards the bound calculation and the tree search will progress slowly.

The lower bounds used in the tree searches in this thesis use algorithms which depend crucially on the availability of ordered lists. Normally the generation of ordered lists is computationally expensive but extensive use can be made of the way information alters from one level of a search tree to another to lessen greatly the computional load. Previous workers in the field of heat exchanger synthesis have failed to take advantage of this fact.

The integer variables in the problems described here determine the matching of the items in two lists (a list of hot and a list of cold streams). At each node just one item from each list is matched leaving all the other items unchanged. Thus all the information at one level of the search tree can be transferred to the next with a fairly minor modification for the match just carried out. This can be achieved with a once-through pass of the data for

any particular node rather than creating afresh the ordered lists at each node, which would require a multi-pass procedure.

4.5 THE BRANCHING STRATEGY

Most nodes in a branch-and-bound search tree have more than one immediate descendant and so an enumeration order must be determined to control the manner in which nodes are expanded. The procedure used to obtain these ordered enumeration lists is called the branching strategy.

A good branching strategy will always create lists within which nodes that feature in near-optimal solutions are close to the top. A near-optimal solution can then be isolated quickly. Thus the upper bound on the objective function can be decreased rapidly and the amount of the search tree enumerated explicitly cut to a minimum.

The branching strategies used in the tree searches described in this thesis use ordered lists created for the procedure which calculates the lower bound on the additional 'cost'. The structure of the problems is such that this is an effective strategy for isolating near-optimal solutions rapidly with the added bonus that it is computionally very efficient.

4.6 THE STOPPING CRITERIA

In some problems each node must be carefully checked to ascertain whether it is a goal node (i.e. if it represents a complete solution to the problem). This is especially the case when several criteria must be applied in conjunction with each other. However, in the problems considered here, only one criterion needs to be applied and the goal nodes are almost self-evident. No complicated procedures are required.

4.7 THE HEURISTIC TO DETERMINE THE VALUE OF THE CONTINUOUS VARIABLES

Mixed integer programmes contain continuous variables as well as integer variables which must be optimised. This optimisation can be achieved during the solution of an MILP by applying a standard linear programming method at each node of the search tree. However, this is computationally expensive and can sometimes be circumvented by applying heuristics to determine the value of the continuous variables. These heuristic rules are obtained by considering the physical model used and are thus very problem dependent. No general guidance is available to aid the development of a suitable strategy.

MINLP's are much harder to solve because many local optima are often present. This makes the complete solution very tedious. Thus heuristic methods are used frequently to aid solution, similar to those described above for MILP's.

Heuristic methods for determining the continuous variables will be used in the tree searches carried out in this thesis.

4.8 APPLICATION OF TREE-SEARCHING METHODS

The development of two depth-first branch-and-bound tree-searching algorithms is described in the following chapters. The first algorithm is used to solve the problem of how to minimise the number of parallel stream splits at the pinch point of any heat exchanger network problem. This has not been considered to any depth by previous workers although it is of considerable industrial importance.

The second algorithm is used to solve the traditional heat exchanger network synthesis problem. A very efficient procedure for calculating the utility usage requirements is developed which forms part of the lower bound on the additional 'cost' to complete a network. A new branching strategy is also developed which increases the efficiency of the tree-searching process.

CHAPTER FIVE

THE AUTOMATIC GENERATION OF ESSENTIAL PARALLEL STREAM SPLITS IN MAXIMUM ENERGY RECOVERY HEAT EXCHANGER NETWORKS

5.1 INTRODUCTION

Many heat exchanger network synthesis problems can be solved with configurations which do not contain process streams split into parallel branches. Such configurations can be found readily by using a branch-and-bound tree-searching technique. In other cases, stream splits are necessary but their automatic generation from within a tree-searching synthesis program is difficult because the possibility of stream splitting increases greatly the size of the combinatorial problem. This dilemma can be circumvented by deciding which streams are to be split prior to synthesizing a complete network. This information is then introduced into the synthesis procedure as a fait accompli and networks are designed under these new conditions.

In some cases there exist one or two streams of large heat capacity flowrate which require heat exchange with a larger number of streams of much smaller heat capacity flowrate so that the choice of which stream to split may be obvious. However, there is another type of problem in which parallel splits are necessary but their placement is far from obvious. It is these problems that the following algorithm addresses. In particular, the regions where stream splitting is most likely to be needed are considered : the pinch points.

Linnhoff and Hindmarsh (1983) have stated the conditions which must be satisfied in the vicinity of pinch points in heat exchanger networks in order to achieve maximum

energy recovery. However, the conditions themselves do not provide guidance as to which streams should be split in order to minimise the number of splits. An algorithm to solve this problem, a mixed integer linear programme, is developed below which provides the complete set of suitable networks in the vicinity of the pinch point. This algorithm is similar in over-all concept to the algorithms used by other workers to solve the heat exchanger network synthesis problem.

5.2 A FORMAL DESCRIPTION OF THE PROBLEM

At a problem pinch point a number (N_h) of process streams to be cooled are to be matched with a number (N_c) of process streams to be heated. Conventionally, these are referred to as the hot streams and the cold streams respectively.

Each stream has a known heat capacity flowrate. These are denoted thus:

$\dot{m}Cp_h^i$ heat capacity flowrate of hot stream i

$\dot{m}Cp_c^j$ heat capacity flowrate of cold stream j.

Each stream may split up into a large number of parallel-branches. The number of branches on hot stream i is given by B_h^i and on cold stream j by B_c^j. In any practical network only a small number of all the possible matches between branches of hot and cold streams exist. The existence of a match between a branch of hot stream i and a branch of cold stream j is indicated by the binary integer X^{ij} which is assigned a value of unity when a match occurs and a value of zero otherwise.

The parallel nature of any stream split necessitates a designation for the heat capacity flowrate in each branch. These are denoted thus:

$\dot{m}Cp_h^{ij}$ heat capacity flowrate of the branch of hot stream i that matches with a branch of cold stream j

$\dot{m}Cp_c^{ij}$ heat capacity flowrate of the branch of cold stream j that matches with a branch of hot stream i.

Both $\dot{m}Cp_h^{ij}$ and $\dot{m}Cp_c^{ij}$ are defined only when X^{ij} is equal to unity.

Finally, the heat capacity flowrate ratio, R^{ij}, of any match is defined, when X^{ij} is equal to unity, as:

$$R^{ij} = \frac{\dot{m}Cp_h^{ij}}{\dot{m}Cp_c^{ij}} . \qquad (5.1)$$

The problem is to determine which streams should be split, if any, and the matches required to minimise the total number of branches. The use of more branches than is absolutely necessary contradicts present-day engineering practice.

5.3 THE OCCURRENCE OF BYPASSES

It is possible for streams, or branches of streams, not to be matched at a pinch point. Thus the occurrence of bypasses must be allowed. Bypass heat capacity flowrates are denoted by matches with pseudostreams $N_c + 1$ for hot streams and $N_h + 1$ for cold. Thus variables:

X^{i,N_c+1} and $\dot{m}Cp_h^{i,N_c+1}$ are defined for hot streams and

$X^{N_h+1,j}$ and $\dot{m}Cp_c^{N_h+1,j}$ are defined for cold.

Maximum energy recovery is, of course, achievable only if there is no bypassing of hot streams immediately above a pinch point and of cold streams immediately below (Linnhoff and Hindmarsh, 1983).

5.4 THE DEFINING EQUATIONS

By definition:

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c+1} \dot{m}Cp_h^{ij} \qquad (5.2)$$

and

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h+1} \dot{m}Cp_c^{ij} \qquad (5.3)$$

Further, the number of branches on a hot stream can easily be found from:

$$B_h^i = \sum_{j=1}^{N_c+1} X^{ij} \qquad (5.4)$$

and on a cold stream:

$$B_c^j = \sum_{i=1}^{N_h+1} X^{ij} \qquad (5.5)$$

5.5 CONDITIONS FOR MAXIMUM ENERGY RECOVERY

The definition of a pinch point and the conditions, given below, that must be satisfied in its immediate vicinity were derived from the observations of Linnhoff and Hindmarsh (1983).

5.5.1 Above a pinch point

From the definition of a pinch point:

$$\sum_{i=1}^{N_h} \dot{m}Cp_h^i < \sum_{j=1}^{N_c} \dot{m}Cp_c^j \qquad (5.6a)$$

To ensure all hot stream branches are serviced:

$$\sum_{i=1}^{N_h+1} B_h^i \leqslant \sum_{j=1}^{N_c+1} B_c^j \qquad (5.7a)$$

$$\text{with } B_c^{N_c+1} = 0$$

To ensure the minimum temperature difference constraint is not violated:

$$R^{ij} \leqslant 1 \text{ if } X^{ij} = 1 \qquad (5.8a)$$

5.5.2 Below the pinch point

Similarly below the pinch point:

$$\sum_{i=1}^{N_h} \dot{m}Cp_h^i \quad > \quad \sum_{j=1}^{N_c} \dot{m}Cp_c^j \qquad (5.6b)$$

$$\sum_{i=1}^{N_h+1} B_h^i \quad \geqslant \quad \sum_{j=1}^{N_c+1} B_c^j \qquad (5.7b)$$

with $B_h^{N_h+1} = 0$

and $R^{ij} \geqslant 1$ if $X^{ij} = 1$. (5.8b)

5.6 AN ILLUSTRATIVE EXAMPLE

Consider a pinch region of a problem which involves eleven process streams, five hot and six cold, as shown in figure 5.1 . Each has a specified heat capacity flowrate which is assumed to be constant in the vicinity of this pinch point.

Since the heat capacity flowrates are constant, the pinch point must be positioned at the inlet temperature of one of the streams, in this case: cold stream six. Application of equations (5.6a) and (5.6b) confirms that this point is a potential pinch point.

Above the pinch point

$$\sum_{i=1}^{5} \dot{m}Cp_h^i = 0.6354, \quad \sum_{j=1}^{6} \dot{m}Cp_c^j = 0.6755$$

$$\therefore \quad \sum_{i=1}^{5} \dot{m}Cp_h^i < \sum_{j=1}^{6} \dot{m}Cp_c^j \quad \text{as required by equation (5.6a)}$$

Below the pinch point

$$\sum_{i=1}^{5} \dot{m}Cp_h^i = 0.6354, \quad \sum_{j=1}^{5} \dot{m}Cp_c^j = 0.6137$$

$$\therefore \quad \sum_{i=1}^{5} \dot{m}Cp_h^i > \sum_{j=1}^{5} \dot{m}Cp_c^j \quad \text{as required by equation (5.6b)}$$

Consider the above problem further by applying equation (5.7a) to the unsplit streams. It can be seen that splitting is not required just because of the numbers of streams since:

$$\sum_{i=1}^{6} B_h^i = 5, \quad \sum_{j=1}^{7} B_c^j = 6$$

$$\therefore \quad \sum_{i=1}^{6} B_h^i \leq \sum_{j=1}^{7} B_c^j \quad \text{as required by (5.7a)}$$

However, the heat capacity flowrate ratios show that hot stream one must be split into at least two branches in order to satisfy equation (5.8a). This increases the sum

of B_h^i to six, the upper limit. If the problem could be solved with this single split the example would be simple. However, this is not the case, and a further hot stream must be split, violating equation (5.8a). Thus a cold stream must be split also. The problem is fast becoming unmanageable and the need for an algorithm to isolate minimum branch solutions is clearly shown.

Below the pinch point the problem is much simpler. As before, splitting is not required just because of the numbers of streams, since:

$$\sum_{i=1}^{6} B_h^i = 5, \quad \sum_{j=1}^{6} B_c^j = 5$$

$$\therefore \quad \sum_{i=1}^{6} B_h^i \geq \sum_{j=1}^{6} B_c^j \quad \text{as required by equation (5.7b)}$$

However, in this case, it is possible to match the hot and cold streams consecutively (1 to 1, 2 to 2, 3 to 3, etc.) without violating equation (5.8b). Thus no branching is required below the pinch point.

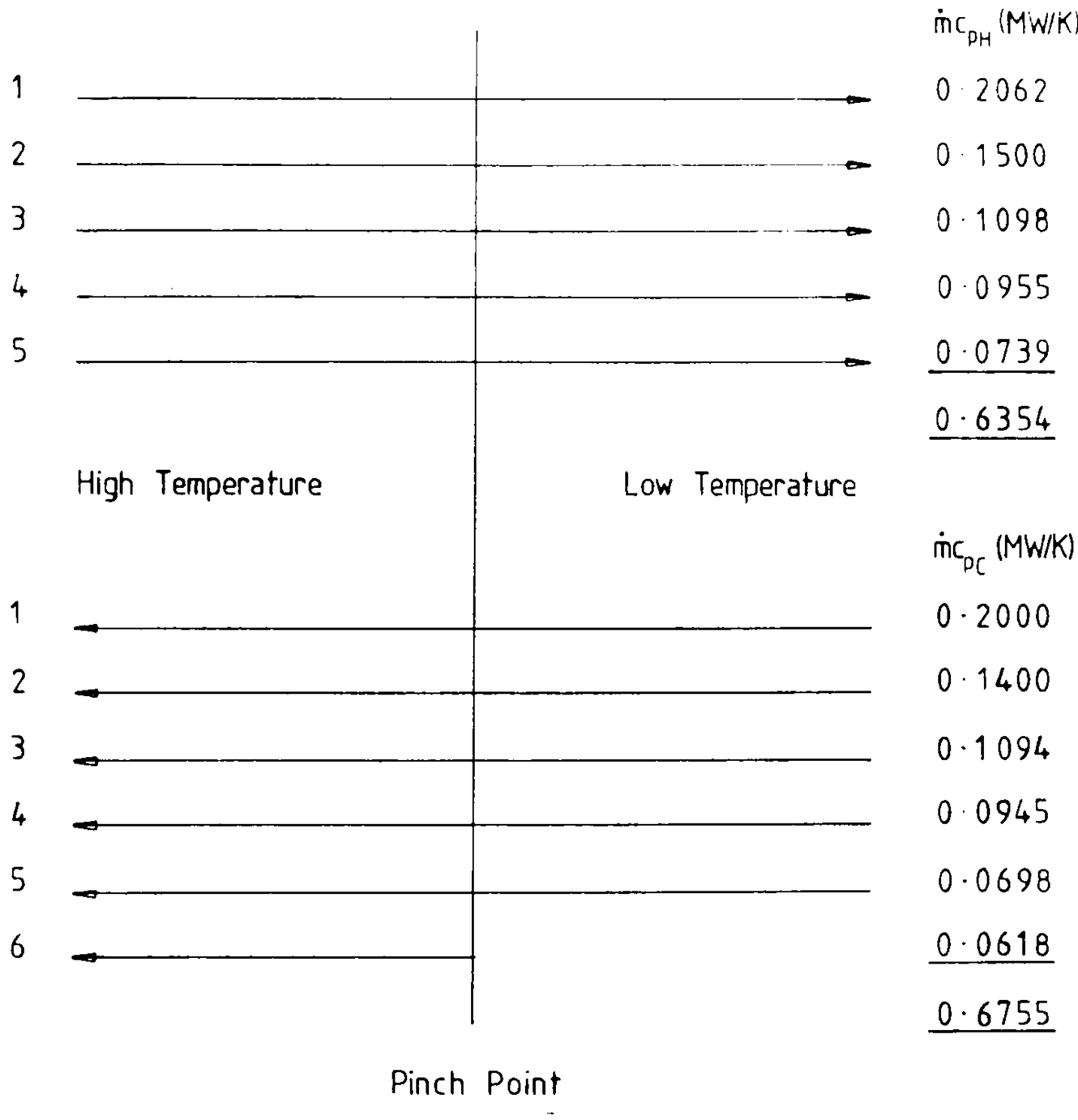


Figure 5.1 A pinch point of the illustrative example

5.7 A PICTORIAL REPRESENTATION

The problem can be considered pictorially (see figure 5.2). The streams with the smaller total heat capacity flowrate are listed in a column on the left hand side of a matrix (hot streams in an above pinch design) and the streams with the larger total heat capacity flowrate are listed in a row along the top.

Associated with each element of the matrix are three numbers. The first is X^{ij}, indicating the existence or otherwise of a match, the second and third $\dot{m}Cp_h^{ij}$ and $\dot{m}Cp_c^{ij}$, the heat capacity flowrates associated with the match. Multiple non-zero X^{ij} entries in any row or column denote the splitting of that stream.

An extra row is now added to the bottom of the matrix headed by the difference between the two total heat capacity flowrates (denoted by $\dot{m}Cp_h^{N_h+1}$ in the example). Associated with each element in this row are two numbers. The first is X^{ij} ($X^{N_h+1,j}$ in the example), indicating the existence or otherwise of a stream bypass, the second $\dot{m}Cp^{ij}$ ($\dot{m}Cp_c^{N_h+1,j}$ in the example), the heat capacity flowrate associated with this bypass. Note that the third heat capacity flowrate, which is associated with the elements of other rows, does not exist, since it would be meaningless. In order to ensure maximum energy recovery, a similar column for bypasses associated with the row streams does not exist.

The objective of minimising the number of branches can now be seen as the minimisation of the sum of all the X^{ij} within the constraints of equations (5.2), (5.3) and (5.8a) or (5.8b).

The heat capacity flowrate data supplied determines

whether an above or below pinch design is being considered. By construction, the limitation on the number of branches, equation (5.7a) or (5.7b), is automatically satisfied since the total number of stream branches associated with the rows must be less that the total number associated with the columns by the number of stream branches in the row headed $\dot{m}Cp_h^N h^{+1}$.

COLD / HOT	$\dot{m}c_{PC}^{1}$		$\dot{m}c_{PC}^{j}$		$\dot{m}c_{PC}^{Nc}$
$\dot{m}c_{PH}^{1}$	X^{11} $\dot{m}c_{PH}^{11}$ $\dot{m}c_{PC}^{11}$				
$\dot{m}c_{PH}^{i}$			X^{ij} $\dot{m}c_{PH}^{ij}$ $\dot{m}c_{PC}^{ij}$		
$\dot{m}c_{PH}^{Nh}$					
$\dot{m}c_{PH}^{Nh+1}$	$X^{Nh+1,1}$ $\dot{m}c_{PC}^{Nh+1,1}$				

Figure 5.2 A pictorial representation for an above-pinch-point design

5.8 REDUCTION OF THE PROBLEM

A branch, rather than a complete stream, which bypasses matches immediately adjacent to a pinch point brings little practical benefit to standard network designs. It forces the heat capacity flowrate ratios of the branches that <u>are</u> matched closer to unity and hence reduces the mean temperature differences of the corresponding exchangers.

Reasons for branch bypassing could be for flexibility or controllability purposes or to increase the area of a very small exchanger up to some minimum manufacturable size. However, bypassing of branches will be discounted and only total stream bypassing allowed.

The effect of this restriction is that any heat capacity flowrate in the bottom row of the matrix can be regarded as being 'spare' and available for distribution among the matches in the column. Only if no other entries are in a column does an entry in the bottom row of that column take on any physical significance.

Since partial bypassing is no longer allowed, the objective function must now be changed to take account of this restriction by removing the X^{ij} in the bottom row of any column which has another non-zero entry. The X^{ij} of the bottom row could be removed altogether but in certain, fairly unusual, cases this can cause problems since the modified objective function would then minimise the number of exchangers at the pinch rather than the number of branches (see appendix G).

A large proportion of the variables can now be removed from the problem. Preliminary matches can be made so that $R^{ij} = 1$ for each match (satisfying equations (5.7a) and

(5.7b)) thus allowing the heat capacity flowrates associated with the column streams to be removed from all elements except those in the bottom row. The bottom row now contains the difference of the row heat capacity flowrate entries in that column and the total column heat capacity flowrate (see figure 5.3).

Feasible solutions cannot be ruled out by this approach since each match utilises the minimum amount of heat capacity flowrate of the column streams and leaves the greatest flexibility for succeeding matches.

5.9 SIGNIFICANCE OF THE BOTTOM ROW

The amount of 'slack' in each column, given by the entry in the bottom row, has important physical consequences. In general, a constant ratio between those entries and the total column heat capacity flowrates is beneficial since each match can diverge away from the pinch at an equal rate. Thus the problem solutions where some of the bottom row entries are very small are likely to be uneconomical.

'Slack' heat capacity flowrate can be moved from one column to another only if a row stream is split. This allows a path to be drawn from the bottom row of one column, through the branches of the split row stream, down to the bottom row of another column. 'Slack' heat capacity flowrate can then be transferred from one column to the other by successively subtracting and adding a fixed amount from each element on the path. Complicated paths of many elements can be built up for this purpose. An example is shown in figure 5.4 where the matrix in figure 5.4a is converted to that in figure 5.4b . The third row is split between columns three and four enabling 'slack' to be transferred between these two columns.

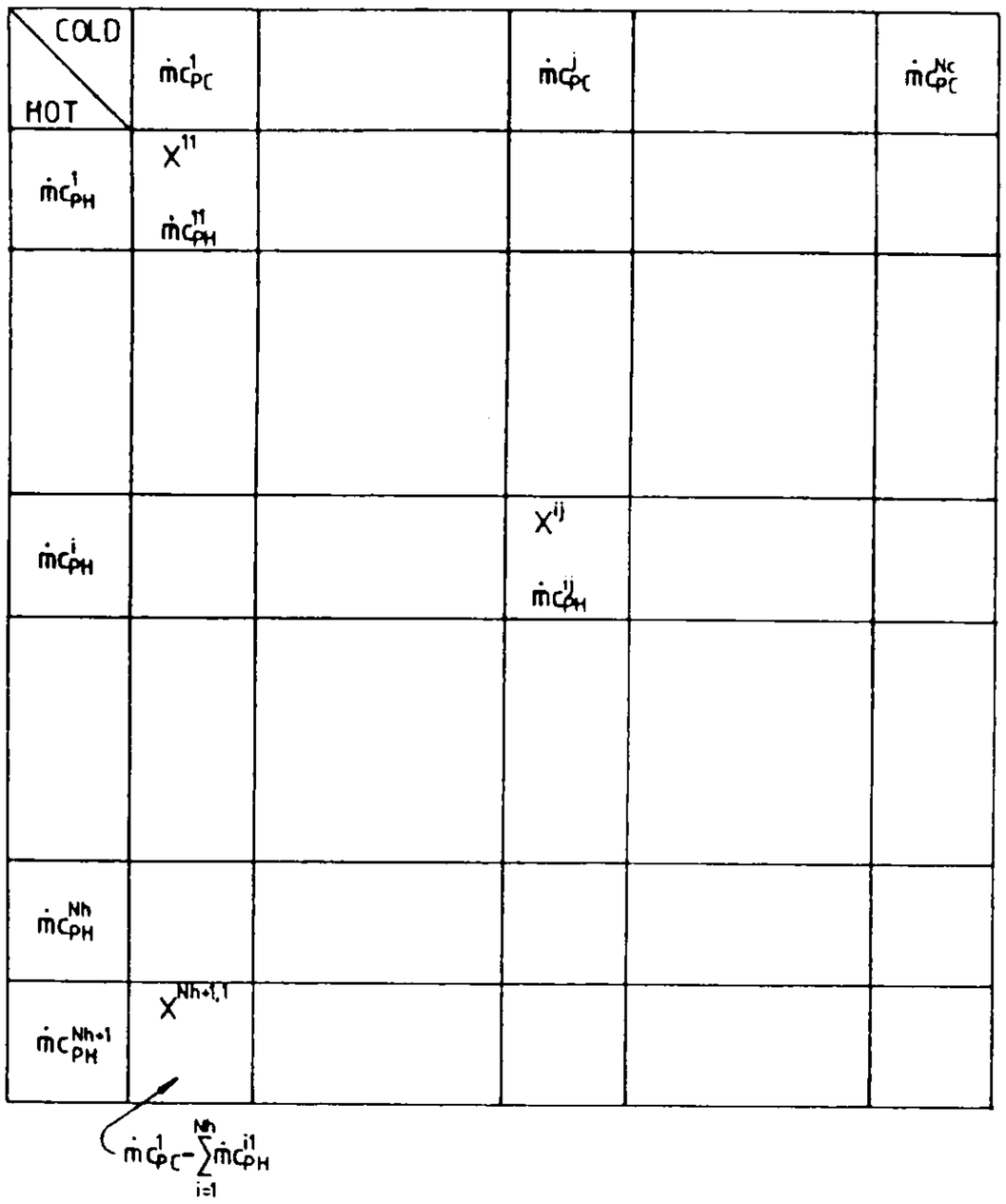


Figure 5.3 An improved pictorial representation.

	6	6	2	2
5	1 5	0 0	0 0	0 0
4	0 0	1 4	0 0	0 0
3	0 0	0 0	1 2	1 1
×4	1 1	1 2	0 0	1 1

5.4a Before

	6	6	2	2
5	1 5	0 0	0 0	0 0
4	0 0	1 4	0 0	0 0
3	0 0	0 0	1 1.5	1 1.5
×4	1 1	1 2	1 0.5	1 0.5

5.4b After

× The 'slack' row

Transfer of a heat capacity flowrate of 0.5 MW/K is shown in figure 5.4 .

Figure 5.4 Movement of 'slack' using row splits

5.10 A MATHEMATICAL FORMULATION

The problem can be written in mathematical notation as:

Above pinch

$$\min_{X^{ij}} F = \sum_{i=1}^{N_h} \sum_{j=1}^{N_c} X^{ij} + \sum_{j=1}^{N_c} Y^{j}$$

where

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c} \dot{m}Cp_h^{ij} \qquad i = 1 \ \ldots \ N_h+1$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h+1} \dot{m}Cp_h^{ij} \qquad j = 1 \ \ldots \ N_c$$

$$\dot{m}Cp_h^{N_h+1} = \sum_{j=1}^{N_c} \dot{m}Cp_c^j - \sum_{i=1}^{N_h} \dot{m}Cp_h^i$$

and

$$X^{ij} = 0 \text{ if } \dot{m}Cp_h^{ij} = 0$$

$$X^{ij} = 1 \text{ if } \dot{m}Cp_h^{ij} > 0$$

$$Y^j \text{ binary integer}$$

$$\sum_{i=1}^{N_h} X^{ij} + Y^j \geq 1 \qquad j = 1 \ \ldots \ N_c$$

Below pinch

$$\min_{X^{ij}} F = \sum_{i=1}^{N_h} \sum_{j=1}^{N_c} X^{ij} + \sum_{i=1}^{N_h} Y^i$$

where

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c+1} \dot{m}Cp_c^{ij} \qquad i = 1 \quad \ldots \quad N_h$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h} \dot{m}Cp_c^{ij} \qquad j = 1 \quad \ldots \quad N_c+1$$

$$\dot{m}Cp_c^{N_c+1} = \sum_{i=1}^{N_h} \dot{m}Cp_h^i - \sum_{j=1}^{N_c} \dot{m}Cp_c^j$$

and

$X^{ij} = 0$ if $\dot{m}Cp_c^{ij} = 0$

$X^{ij} = 1$ if $\dot{m}Cp_c^{ij} > 0$

Y^i binary integer

$$\sum_{j=1}^{N_c} X^{ij} + Y^i \geq 1 \qquad i = 1 \quad \ldots \quad N_h$$

The variable Y caters for the completely bypassed stream, taking the value of unity only when all the other elements in the column are zero.

Added constraints

Further constraints can be added to limit the numbers of stream branches or to disallow matches. In this case a solution cannot be guaranteed. To limit the number of branches on any one stream, say hot stream i, a constraint of the form:

$$\sum_{j=1}^{N_c} X^{ij} \leqslant B_h^i \max$$

must be imposed. To disallow a match, say between hot stream i and cold stream j, a constraint of the form:

$$X^{ij} = 0$$

must be imposed.

5.11 THE PROBLEM VIEWED AS A MIXED INTEGER LINEAR PROGRAMME

In the form stated, the problem is a mixed integer linear programme (MILP). Computer codes are readily available to solve such problems. Initially software developed at North-Western University was used (MPOS). In particular, a branch-and-bound technique (BBMIP) within the package was found to be applicable (see appendix H).

BBMIP initially solves the problem with all the integer variables replaced by continuous variables bounded within the same limits. If a feasible solution can be found the package successively establishes values for the integer variables and solves the reduced linear programme in order to obtain bounds and determine the search strategy.

The results achieved for a number of test problems (see appendix I) supported the validity of the approach. However, only problems up to about nine streams could be solved because of computer running time limitations. The running time was strongly dependent on the upper bound placed on the total number of branches. As problems became larger it became more difficult to estimate this bound and the running times became unrealistic. Additionally, not all the solutions with the optimum objective function value were obtained. The algorithm only finds a verified optimum solution and does not supply all the alternatives.

5.12 THE PROBLEM VIEWED AS A TRANSPORTATION PROBLEM

The problem can also be seen as a variant of the transportation problem in linear programming. Unlike the classical problem ('Hitchcock problem') in which the transportation costs are based on a unit cost, in this formulation the costs are fixed, i.e. independent of the amount transported. Many codes exist to solve the classical problem and some to solve the fixed- and unit-cost problem (where the cost of transportation is a mixture of a fixed cost and a unit cost). However, codes that take advantage of the simplifications that arise when there are only fixed costs seem difficult to obtain. In addition, the need to find all the solutions with the optimum function value indicated the need to develop a new approach.

5.13 A NEW SOLUTION ALGORITHM

The use of branch-and-bound tree-searching techniques to solve ILP's, MILP's and MINLP's is widespread and the success of BBMIP in solving test problems showed that the development of such a procedure was likely to be fruitful. This led to the development of a new solution algorithm described below.

5.13.1 Introduction to the new solution algorithm

BBMIP was designed to solve the general formulation of the MILP and therefore cannot be expected to take advantage of the special characteristics of each problem. In this formulation, the values of the continuous variables, which do not appear in the objective function, are very closely linked to the values of the integer variables. Indeed, when the value of an integer variable becomes fixed at some level of the branch-and-bound search tree, the value of the associated continuous variable can be fixed immediately. This very special property is the key to a far more efficient solution procedure.

Optimal solutions to fixed-cost transportation problems can never have loops of non-zero entries contained in their problem tableaux unless secondary upper-bound constraints are imposed on the transfer of 'goods' from particular sources to particular destinations (see appendix J). There are no such upper bound constraints in this problem.

Transportation problems are concerned with moving fixed amounts of 'goods' from sources to destinations at the lowest cost. In the fixed-cost transportation problem, the optimal amount to 'transport', once a particular match

between source and destination has been selected, is simply the maximum allowable. This can only make the remaining problem easier to solve because the amount of 'goods' left to 'transport' is minimised. This also ensures that loops can never occur in the tableau because at least one row or column is always satisfied at each level of the search tree.

BBMIP uses the solution of a linear programme at each node of the search tree to obtain a lower bound on the further objective function 'cost' which will be incurred while completing the problem and to determine the search strategy for the remaining nodes of the tree. The lower bound on further 'cost' can be calculated by considering in turn each row of the tableau against a list of all the columns ordered into decreasing size to give a minimum 'cost' for the rows. A similar process can be carried out for the columns by considering them against a list of all the rows ordered into decreasing size. The larger of the column 'cost' and the row 'cost' is then taken as the lower bound on the future 'cost'. A good branching heuristic can also be obtained from the ordered lists of rows and columns at minimal computational expense thus rendering the need to solve a linear programme at each node of the search tree unnecessary.

5.13.2 The bounding procedure

A lower bound for both the rows and the columns can be calculated and the larger taken as the over-all lower bound. This is carried out using structured lists of the row and column residual heat capacity flowrates which are sorted into descending order. (The residual heat capacity flowrate being defined as the amount of heat capacity flowrate still not satisfied by the matches already existing.) In the case where no matches have yet been placed on a stream the 'residual' equals the original heat capacity flowrate. A useful by-product of this bounding procedure is that the branching strategy can use these lists to determine the order in which the tree is searched at negligible computational cost.

5.13.2.1 The row bound

Each row in the MILP tableau for a particular node is considered in turn and compared with the ordered list of columns starting with the one with largest residual heat capacity flowrate. A running total is kept of the heat capacity flowrates of the columns with which the row is allowed to match until the sum is greater than or equal to the row heat capacity flowrate. The number of columns contributing to this sum is then the minimum number of matches required to satisfy this row. The number of columns for each row is then summed to give the lower bound required.

In the case where a row cannot be satisfied by all the allowable column matches (or where it cannot be satisfied by the maximum number of allowable column matches when a limit is imposed) the MILP is unsolvable and the node and its descendants should be discarded.

5.13.2.2 The column bound

A similar approach to that outlined for the row bound can be used for the column bound with an extra allowance for the total heat capacity flowrate difference (called $\dot{m}Cp_h^{N_h+1}$ or $\dot{m}Cp_c^{N_c+1}$ above, depending on whether an above or a below pinch design is being carried out; denoted here by Δ). A part of each column, up to heat capacity flowrate Δ , can remain unmatched and so Δ is added initially to the running total before considering matches with the ordered list of rows. However, if no matches are indicated for a column because of this allowance and no matches already exist in that column the number of matches required is set to unity to denote a completely bypassed stream (i.e. Y^i or Y^j in the mathematical formulation becomes non-zero).

The ability to impose a limit on the number of branches into which a stream may be split enables the tableaux to be kept as small as possible. A column that has its full complement of matches can be deleted from the tableaux of the succeeding nodes even though it is still not completely satisfied. Thus the value of Δ changes from level to level in the search tree because the unsatisfied residual heat capacity flowrate of a deleted column, the 'slack', can be subtracted from the value of Δ for the relevant node to give the value of Δ for the next level down in the search tree. This results in the column bound becoming tighter as the search tree is fathomed further. (As stated above, subsequent nodes are not developed if, at some node where a column obtains a full set of matches, the unsatisfied residual heat capacity flowrate of that column is greater than the relevant Δ. This is because Δ must always remain positive to maintain feasibility).

5.13.2.3 A tighter column bound

The procedure described above uses the value of Δ in the calculation of the bound for each column rather than just once in the complete problem. In the case of large values of Δ, the bound will thus be considerably slacker than necessary.

A tighter bound can be envisaged when, instead of considering each column in turn, the column order list is used more extensively. Each time a match is made the residual heat capacity flowrate of that column could be placed back in the list in its correct, new position. The new column at the top of the list would then be processed. This procedure would then be continued until the sum of the remaining residuals was less than or equal to Δ. Thus the bound would be tightened considerably at the expense of a considerable amount of extra sorting.

In problems where Δ is large, the problem of which streams to match is likely to be easily solved and therefore the efficiency of the bounds is not of key importance. The number of matching options is likely to be large and matches around the pinch can be selected to minimise some cost function. At the other extreme, where Δ is small, the above tightening of the bounds will have little effect on the number of nodes expanded and increase the computational load considerably. The procedure was therefore not implemented because the benefits appear to be outweighed by the drawbacks.

5.13.3 The branching strategy

The branching strategy used here creates an ordered list composed, in itself, of two ordered lists. These are the two lists used by the bounding procedure; the ordered lists of the residual heat capacity flowrates of the columns and of the rows. The first expansion of each node of the search tree entails the matching of the first element on the column list with the first element in the row list. On back-tracking to this node, the first column list entry is matched against the second row list entry; on the second back-track the first column list entry is matched against the third row list entry. This continues until there are no more entries on the row list. The process continues by matching the second column list entry with each of the row list entries followed by the third column list entry and so on until every combination has been considered.

This form of branching strategy was chosen for several reasons. Firstly, an optimal solution to a problem which requires no splitting is found very quickly; frequently the first solution found is optimal. This is because the largest entries in the ordered lists are the first to be matched at any given node so, unless a very big residual is left by a node close to the root node, the streams will be matched consecutively from the ordered lists for the root node (i.e. all match residuals go to back into the ordered lists below the original streams still unmatched). Secondly, matches involving small residuals created from matches with the column streams are the last to be considered. These matches are unlikly to appear in an optimal solution since the heat capacity flowrate involved is likely to be absorbed into the 'slack' without adding to the objective function 'cost'. Finally, the ordered lists are created as part of the bounding procedure and thus the

branching strategy can be carried out at negligible computational cost.

5.13.4 Unique solutions

The branching strategy described above will produce all the possible combinations of matches that result in the optimal value of the objective function. However, different sequences of matches can often represent the same physical network. Thus a procedure is required to assure one-to-one correspondence between the solutions obtained and the physical networks or, at least, to decrease the number of variants obtained.

The possibility of more than one problem solution describing the same physical network can be demonstrated in the following way. Let M_n represent the n'th match in the sequence of matches forming a solution where M_n denotes a match between the column C_n and the row R_n. Similarly, the next match in the sequence, M_{n+1}, represents a match between C_{n+1} and R_{n+1}. Now, if M_n and M_{n+1} are disjoint (i.e. $C_n \neq C_{n+1}$ and $R_n \neq R_{n+1}$), the two matches can be exchanged to produce another optimal solution which has the same physical structure. This type of match swopping is also possible when disjoint matches are not adjacent so long as the intervening matches are disjoint with the matches to be moved.

5.13.4.1 Problem solutions with only one permutation of the disjoint elements

A procedure, introduced by Pho and Lapidus (1973) and developed by Grossmann and Sargent (1978), to ensure that just one of the permutations of the disjoint elements is produced during the solution of heat exchanger network synthesis problems can also be used during the solution of this stream-splitting problem. In essence, this procedure assigns a unique number (k) to each element of the MILP tableau. Each potential new match is then checked against all the existing matches, created higher up in the search tree, starting with the most recent match and working back to the root node. If the potential match has a row or column in common with the existing match being considered, the match is immediately allowed. Otherwise, if the value of k for the potential match is less than the value of k for the existing match being considered, the match is rejected. If neither of these is true, (i.e. k for the potential match is greater then the existing match) the procedure continues onto the next level up in the search tree. Eventually, if the potential match is neither accepted or rejected on the way up through the search tree, the root node is reached and the match is accepted by default.

The unique numbering system used in this work is given by:

$$k = R + (C-1).(\text{Maximum numbers of rows})$$

where R and C denote the row and column numbers, respectively.

This procedure removes many of the problem solutions that correspond to the same physical network but does not give one-to-one correspondence. A study of the nature of the

problem solutions, particularly the effect of joint tableau elements (i.e. elements with a row or column in common), leads to an amended procedure. This decreases the number of permutations obtained which describe a single physical network.

5.13.4.2 'Strings' of non-zero elements in the LP tableau

The above procedure decreases drastically the number of allowable permutations of a single physical solution when the majority of the non-zero elements in the tableau are disjoint. However, many permutations of joint elements are still permissible. These permutations would normally represent different solutions of an MILP or an ILP but in this particular problem they can often represent the same physical network. There are two reasons why this should be so. Firstly, the non-zero elements in the bottom row of a tableau usually have no physical significance (they represent the 'slack') and so all the solutions with the same non-zero integer variables in the main body of the tableau represent the same network irrespective of the position of the non-zero elements in the bottom row. Secondly, streams which are split into several branches in a particular solution will give rise to many permutations because the non-zero elements in the relevant row or column will appear in many different sequences but always have the same value.

These problems can be illustrated by considering the following tableau fragment which shows a particular solution, marked by X's, which satisfies the given rows and columns:

ṁCp	8	3	2
6	1 X	4 X	7 X
6	2 X	5	8
Δ = 1	3	6	9

Figure 5.5

A tableau fragment

TOP LEFT HAND OF EACH MATRIX ELEMENT CONTAINS THE K NUMBER

The above procedure to remove permutations of disjoint elements still allows the solution in figure 5.5 to be built up in the following eight ways:

Permutation Number	Matching Order	Placement of 'slack'	Pair Number
1	2,1,4,7	9	1
2	2,4,1,7	9	
3	2,7,4,1	3	2
4	2,4,7,1	3	
5	2,7,1,4	6	3
6	2,1,7,4	6	
7	4,7,1,2	3	4
8	7,4,1,2	3	

Table 5.1
Permutation of a solution

Permutations one and two can be seen to be exactly the same except for the order of the matches in elements one and four. Similarly, permutations three and four differ only in the order of the matches in elements four and seven, permutations five and six differ in the order of the matches in elements one and seven and permutations seven and eight differ in the order of the matches in elements four and seven. The exchange of matches within these pairs of permutations makes no difference to the final solution and the reason for the abundance of permutations can be seen to be due to the shuffling of all the matches on the split stream (matches in elements one, four and seven in row one) except for the last element made on

that stream. In other words, the elements representing a split stream are order independent except for the last one created.

The continuous variables associated with the matches are different for each pair of permutations but any solution given can be obtained from any other solution by the movement of 'slack' from one element in the bottom row to another.

5.13.4.3 An amended procedure to decrease the solution permutations

Since the matches on a split stream are order independent except for the last one made, an extention can be made to the above procedure to decrease the number of permutations. This involves the rejection of any potential match if it has a smaller k number than any already existing joint match and if it does not completely satisfy the relevant row or column. Thus branches on a split stream are built up by creating matches in elements of increasing k number. This makes it necessary only to compare the most recent match on any row or column with the present potential match because that tableau element will have the largest k value for the relevant row or column. Using this procedure on the above example results in the number of permutations being cut down to three:

Permutation Number	Matching Order	Placement of 'slack'
1	2,1,4,7	9
2	2,1,7,4	6
3	4,7,1,2	3

Table 5.2 Reduced permutation of a solution

5.13.5 Initial value of the upper bound on the objective function

A computer program, COMPASS (COde to Minimise PArallel Stream Splitting), has been developed containing the above procedures. The initial value of the upper bound on the objective function is set automatically to the maximum that the software is designed to handle until the first solution is obtained. This then becomes the new upper bound in the normal way. A brief outline and listing of this computer program are given in appendix K.

5.14 SOLUTION TO THE ILLUSTRATIVE EXAMPLE

A limitation of a maximum of two branches per stream was imposed. The solutions obtained are shown in figures 5.6 , 5.7 and 5.8 . These three configurations were obtained in a computer running time of 34.89 s (CDC Cyber 170/720). All that is left for the process engineer to carry out is the placement of the 'slack' described earlier to give the final design. Due regard must be made of the relative heat transfer coefficients and the size of each exchanger when allocating the 'slack' since this determines the rate at which the streams diverge from the minimum temperature difference as heat is transferred.

5.15 COMPARISON BETWEEN MPOS AND THE NEW METHOD

A reduced version of the problem given above was solved using both MPOS and the new algorithm (see figure 5.9). Because of machine memory limitations the MPOS solution was obtained using the simplified formulation given in the appendix G.

When an upper limit of two branches per stream was imposed BBMIP found 3 solutions in a run time of 293 s. The new method found 8 solutions in a run time of 2.55 s.

Additionally,when an upper limit of three branches per stream was imposed the BBMIP run time increased to 428 s and the new method's run time increased to 3.48 s.

5.16 CONCLUSIONS

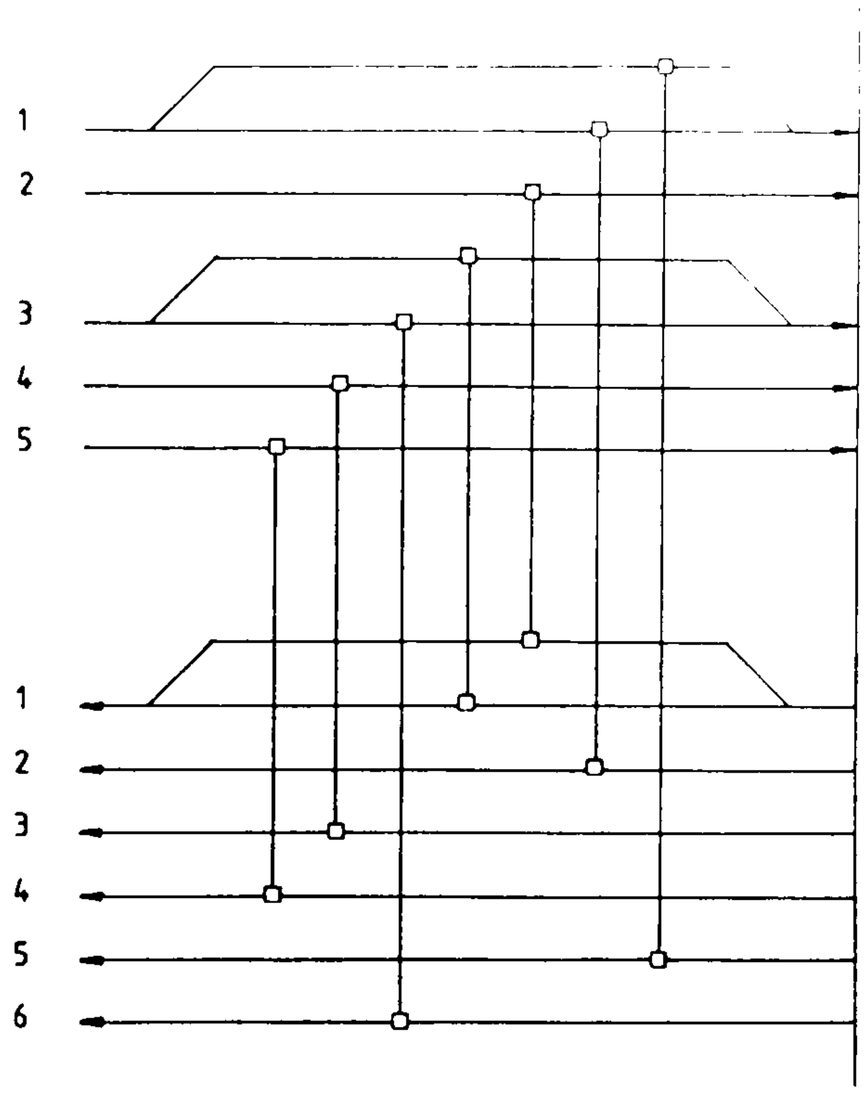
An algorithm has been developed to determine all the configurations of heat exchangers adjacent to problem

pinch points that utilise the minimum number of branches. The algorithm completes its search in a time which is two orders of magnitude quicker than a general purpose optimisation code. Additionally, the process engineer can be certain that all such solutions have been isolated and is free to choose between them on grounds such as flexibility and operability.

5.17 POSSIBLE FUTURE DEVELOPMENTS

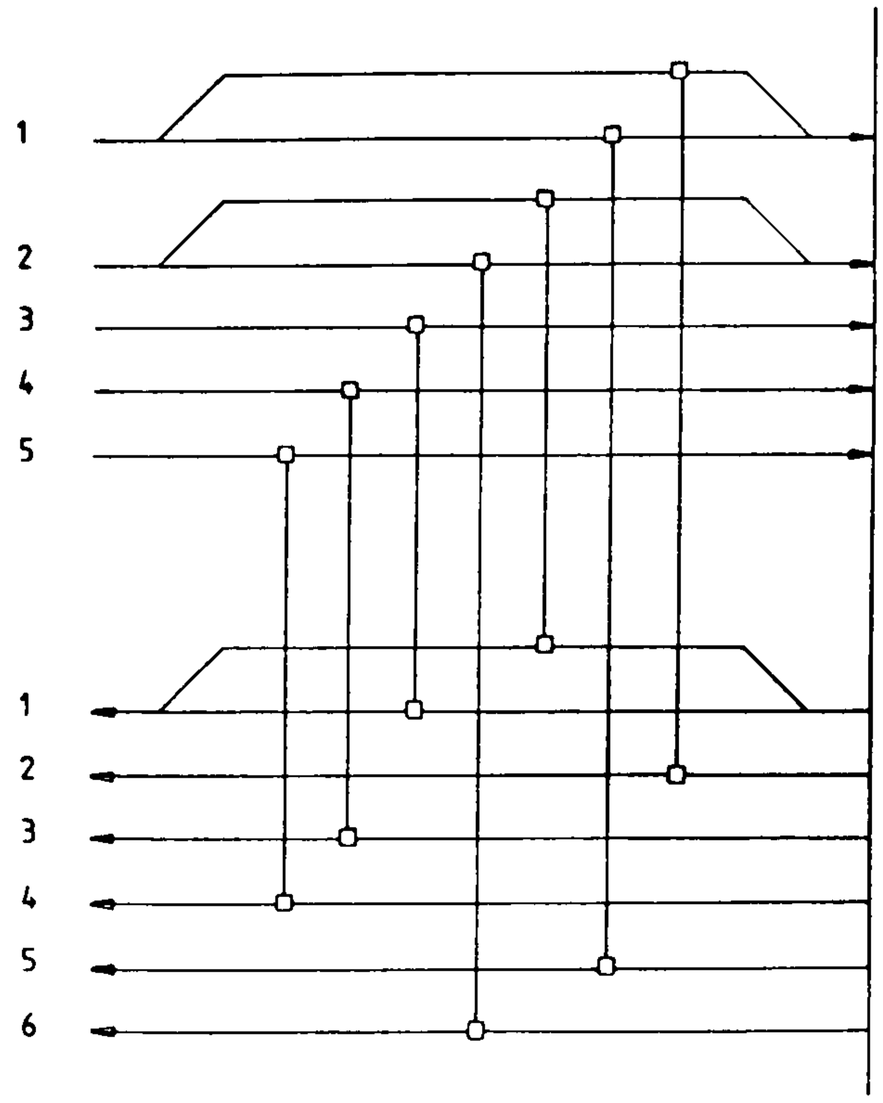
In the cases where the number of branches are constrained or matches are disallowed a solution is not guaranteed. Maximum energy recovery may not be possible and some alternative approach is required.

The problem can be viewed as the servicing of all the rows of the matrix without recourse to using utilities if at all possible. In the case where no solution can be found an extra column must be added to represent utility usage. A suitably high cost must be put on matches with this column to minimise its use.



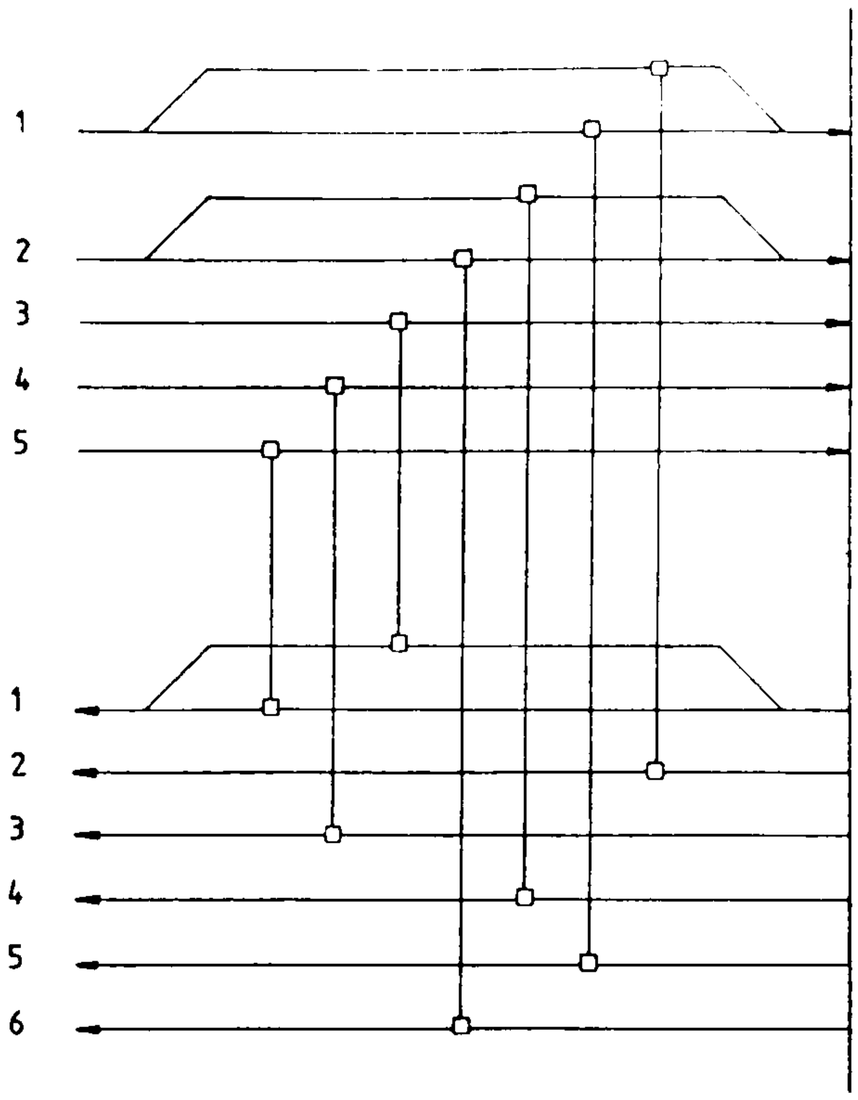
(For m̀Cp data see figure 5.1)

Figure 5.6 Solution one to the illustrative example



(For m̀Cp data see figure 5.1)

Figure 5.7 Solution two to the illustrative example



(For ṁCp data see figure 5.1)

Figure 5.8 Solution three to the illustrative example

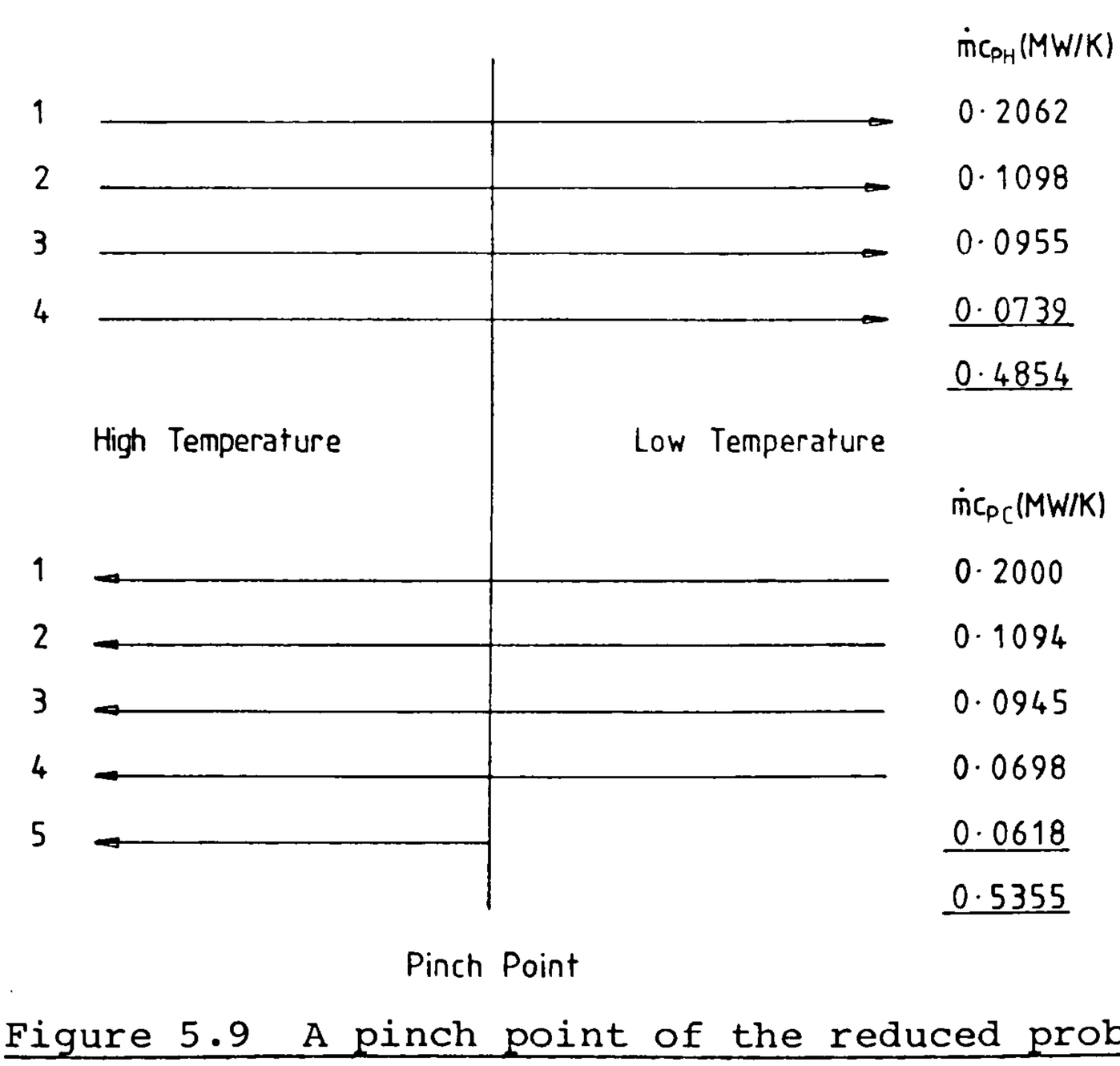


Figure 5.9 A pinch point of the reduced problem

CHAPTER SIX

A SET OF ALGORITHMS TO SYNTHESIZE HEAT EXCHANGER NETWORKS

6.1 INTRODUCTION

A tree-searching method is developed in this chapter which solves the heat exchanger network synthesis problem in a manner very similar in over-all concept to the stream splitting algorithm described in chapter five. The bounding algorithm re-orders information associated with two streams at each node; the branching strategy uses two ordered lists; permutations of the same solution are avoided by applying an amended form of the procedure suggested by Pho and Lapidus (1973).

The philosophy of the pinch point design method (Linnhoff and Hindmarsh (1983)) is used to partition pinched problems into sections, which are then solved independently. Design starts at the pinch point and extends outwards but, unlike the pinch point design method, heat exchangers are not added so as to always satisfy the heat content of one stream in a sub-problem. This difference is a consequence of the introduction of non-counter-current exchangers.

Temperature interval tables, introduced by Hohmann (1971) and amended by Linnhoff and Flower (1978a) and Su (1979), are used to check the utility requirements at each level of the search tree. The data associated with these temperature interval tables form the bounding information which are amended at each level of the search tree for the streams just matched to give the tables for the nodes in the next level down.

The tree-search algorithm is formulated to find a given number of heat exchanger networks, both above and below the problem pinch point if it exists, which have the minimum number of heat exchanger shells and which are within the constraints imposed by the laws of thermodynamics and by the user. If no pinch point is present, the synthesis starts at either the high or the low temperature end of the network and works consistently towards the other end.

The utility stream heat loads are fixed by the user thus enabling process and utility streams to be treated in a similar manner. This leads to a restructuring of the utility usage bound algorithm so that it indicates if extra utilities are required, above the level stated, rather than the total required. Since all streams are treated in a like manner, only one pair of ordered lists is required for the branching strategy and matches between utility and process streams are treated in the same way as matches between two process streams. Thus matches exchanging utility heat are not constrained, as in many synthesis methods, to stream ends.

6.2 HEAT TRANSFER ACROSS THE PINCH POINT

Networks designed to obtain maximum energy recovery sometimes contain too many stream splits or heat exchangers to be practical. In these cases, more suitable networks can be obtained by reducing the amount of heat recovered. This is usually achieved by transferring heat across the pinch point.

Heat transfer across the pinch point is not readily dealt with if the heat exchanger network synthesis problem is completely partitioned into two sub-problems at this point. The procedure used in this algorithm adds two dummy single-component phase-change streams at the pinch point, one to each sub-problem, with a heat content equal to the heat transfer required across the pinch point. The partitioned problems are then solved individually leaving the user to amalgamate the solutions by allocating the matches stradling the pinch point and hence removing the dummy streams.

6.3 INTRODUCTION TO THE HEURISTIC FOR SETTING THE CONTINUOUS VARIABLES

The setting of the continuous variables in the parallel stream-splitting problem in chapter five was straightforward because a direct relationship was found between the value of the integer variables and the continuous variables. A similar heuristic is used in the solution of the heat exchanger network synthesis problem described in this chapter although the reasoning behind its application is somewhat less tenable.

In the network synthesis problem there are interactions between the continuous variables and so they cannot be set

completely independently of one another. However, the use of non-counter-current heat exchangers lessens this interdependence to the point where, in practice, the interactions can often be ignored.

Since the heuristic that sets the continuous variables determines the size of the heat exchanger used in each match, the heuristic is referred to as the match generator in the following sections.

6.3.1 The match generator

At any given node of the search tree a pair of streams is designated as the streams to be matched in the next heat exchanger. The amount of heat to be transferred and hence the area and cost of the heat exchanger must then be determined. The heuristic used by many workers in the past maximised the heat transfer in each exchanger within the constraints imposed by the laws of thermodynamics and any minimum temperature difference limitation. A similar heuristic is used in this work with an extra constraint added to ensure that non-counter-current heat exchangers can be used effectively.

Chapter three showed that the maximum amount of heat that can be transferred in a single non-counter-current heat exchanger shell is often limited by the need to keep the exchanger controllable. The heuristic which is used widely to avoid controllability problems was shown to be equivalent to ensuring that the heat exchanger is designed so that a cross does not occur in the outlet temperatures. Therefore temperature crosses are forbidden in the non-counter-current heat exchangers specified by the match generator used in this work. Thus the maximum heat duty allowed in a heat exchanger becomes the smallest of the

following three heat duties:

a) The unsatisfied heat duty, within the current sub-problem, of the cold stream in the exchanger.

b) The unsatisfied heat duty, within the current sub-problem, of the hot stream in the exchanger.

c) The heat duty obtained when the hot and cold stream outlet temperatures from the exchanger are equal.

(A sub-problem being either that above or below the complete problem pinch point.)

An extra difficulty occurs when utility streams which exhibit a temperature range are matched because limits, rather than fixed values, are specified for the outlet temperatures of these streams. Thus heat exchangers matching these utilities can often be made larger by decreasing the utility stream outlet temperature and hence allowing more heat to be transferred before the onset of a temperature cross. However, since minimum utility consumption is thought generally to be more important than minimising the number of heat exchangers, the temperature range of a utility stream is maximised within the constraints imposed by the process stream temperatures.

The use of multiple heat capacity/latent heat flowrate ranges for streams complicates the process of calculating stream temperatures within heat exchangers and ensuring that the minimum temperature difference is maintained. The procedure used in this algorithm works from the end of the exchanger for which the temperatures are known towards the other end. Each heat capacity/latent heat flowrate range delimiter encountered on the way is used to partition the heat exchanger into sections (see figure 6.1).

At each partition a check is made on the temperature difference; if the difference is large enough the calculations continue into the next section, otherwise the duty at which the minimum difference occurs in the current section is calculated and the exchanger terminated at that point.

6.3.2 Options allowed in the match generator

The way in which the heat content of a designated match is calculated has been made as flexible as possible.

The main option carries out the procedure described above by defining one heat exchanger shell at a time. This makes a very large number of networks possible.

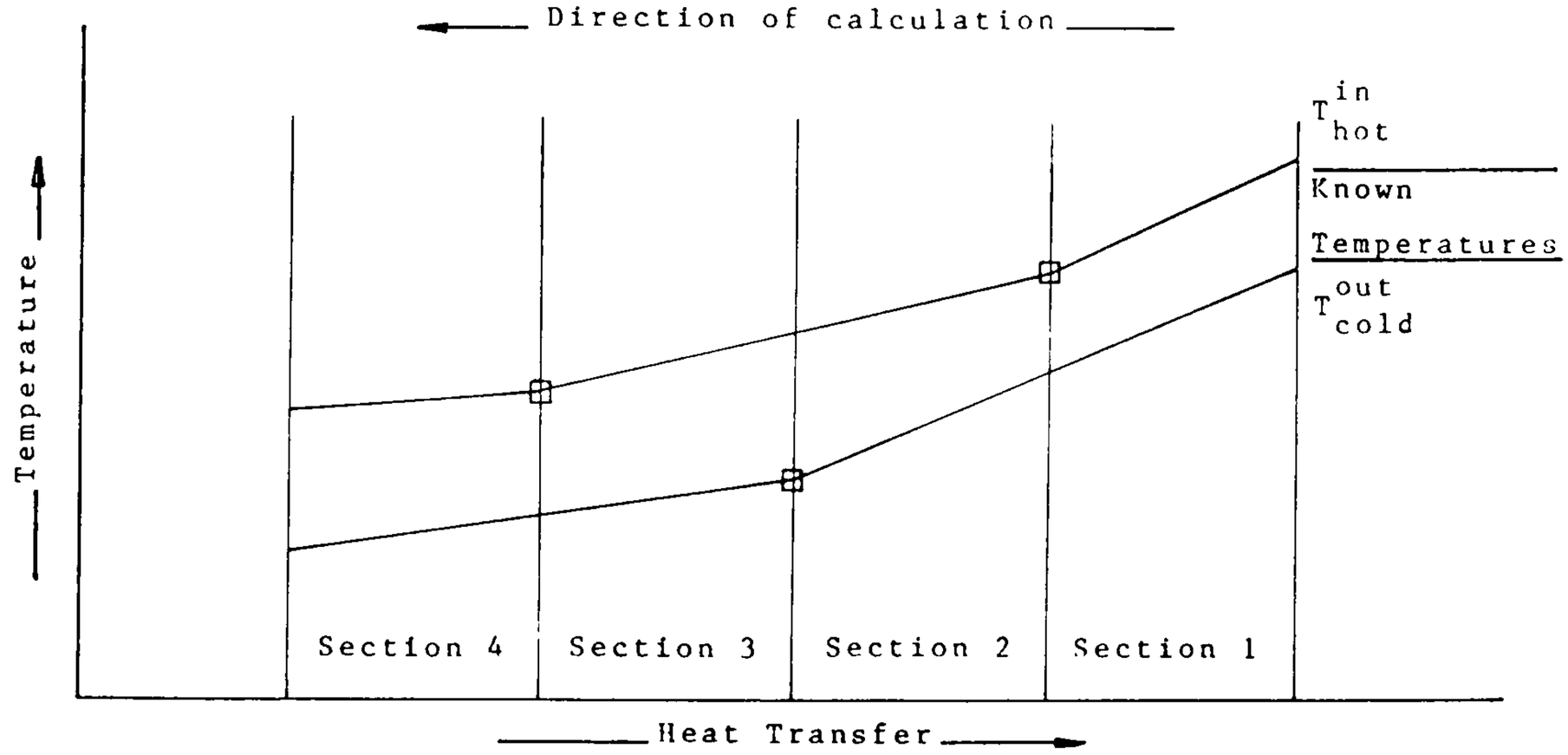
In problems where heat recovery is not too difficult, a further option is included which reduces the number of possible networks by synthesizing heat exchanger trains, wherever possible, with up to a stated maximum number of shells. Thus, once a match has been decided upon, heat exchanger shells are repeatedly added which match the same two streams until one of the following occurs:

a) The heat duty, in the current sub-problem, of the cold stream becomes fully satisfied

b) The heat duty, in the current sub-problem, of the hot stream becomes fully satisfied

c) Further matches between the same two streams are infeasible

d) The maximum limit on the number of heat exchanger shells allowed in a train is reached.



□ denotes a change in heat capacity flowrate of a stream

Figure 6.1 The partitioning of a heat exchanger at the points where heat capacity flowrate changes

The networks obtained from such a procedure achieve a measure of 'modularity' leading to a largely unquantifiable saving in cost. However, the number of shells required is often increased.

Finally, for networks in which counter-current exchangers can be used, the limitation on temperature crosses can be removed in order to obtain networks similar to those based in the pinch design method and the 'ticking-off' heuristic (Linnhoff and Hindmarsh (1983)).

6.4 AN INTRODUCTION TO THE BOUNDING ALGORITHM

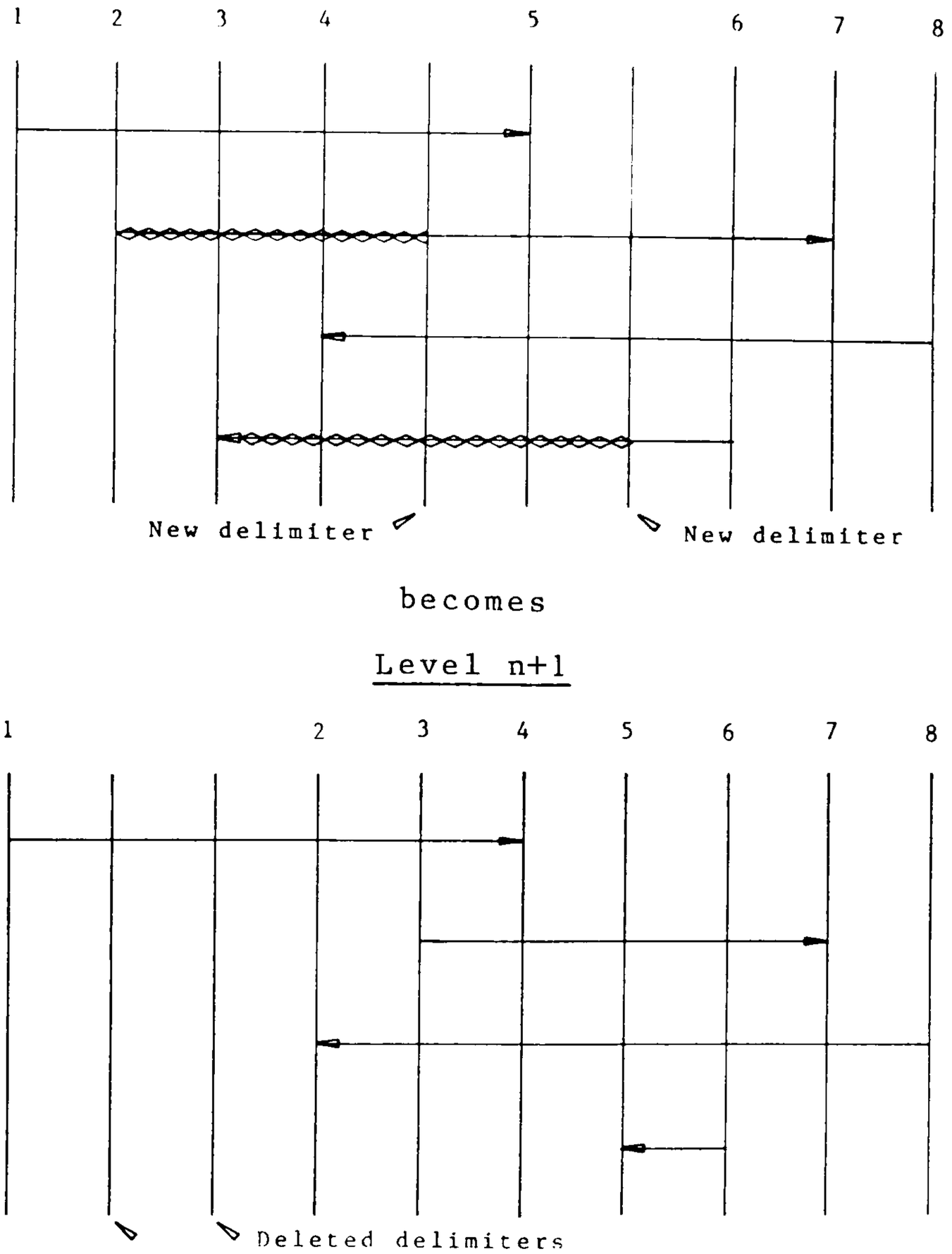
The information required to calculate the utility requirement bound is composed of a set of interval delimiters, ordered by temperature, and the associated interval heat contents. Each interval delimiter marks a temperature at which a stream enters or leaves the network or alters heat capacity or latent heat flowrate. Since only one stream is associated with each delimiter, several adjacent delimiters of equal temperature can occur if more than one stream enters, leaves or alters heat data at a given temperature. The interval heat contents are the net result of subtracting the sum of the cold stream heat contents from the sum of the hot stream heat contents for the unmatched stream sections in each interval.

At each node in the search tree a match is made between two streams leaving all the other streams unaltered. Thus the bounding information which has been calculated for any level of the search tree can be amended to give the bounding information for subsequent levels by amending the data for the two matched streams; the data concerning the other streams are not altered (see figure 6.2). This amendment

can be achieved in a once-through pass of the bounding information resulting in a large reduction in the calculation load compared with the multi-pass ordering procedures required when no previous knowledge is available. A significant improvement in efficiency is therefore obtained because all the computationally expensive ordering takes place only once, at the root node.

6.4.1 Stream match types

Streams can be matched so that heat exchangers are sequential, in parallel or form a combination of the two. Sequential matching results in a decrease in the temperature ranges of the unmatched sections of streams but leaves the heat content of the remaining sections unaltered. Parallel matching results in a decrease in the interval heat contents through which the streams pass but does not change the temperature ranges of the unmatched sections of streams until the stream is completely satisfied and deletion occurs.



◇◇ denotes the matched section of a stream

Figure 6.2 The transformation of bounding information from one level of the search tree to the next

6.4.2 Outline of the procedure

The procedure described below considers in turn the information associated with each element in the ordered list of interval delimiters for a particular level of the search tree, starting at either the highest or lowest delimiter. The data for each element in the old list are transferred to a new list, representing the new level of the tree, and amended before considering the next old element. Amendments take the form of the inclusion of new delimiters (i.e. the splitting of an old interval into two or more new intervals), the deletion of old delimiters (i.e. the amalgamation of two or more old intervals into one new one) and the updating of interval heat contents.

The need to add and delete delimiters arises from the reduction in the temperature range of the unsatisfied sections of streams as heat exchangers are added sequentially. Thus, when a new section of a stream becomes matched,the associated delimiters must be deleted and a new delimiter inserted to mark the new end of the unsatisfied section of the stream. The need to alter interval heat contents arises from both the removal of matched sections of sequentially matched streams and the use of heat exchangers matched in parallel.

6.4.3 Stream types

There are four types of stream allowed in this formulation of the problem. These are:

a) Process streams which exhibit a temperature range, i.e. streams which transfer energy by exchanging sensible heat or latent heat from a multi-component phase change

b) Process streams which do not exhibit a temperature range, i.e. streams which transfer energy by exchanging latent heat from a single-component phase change

c) Utility streams which exhibit a temperature range, i.e. streams which transfer energy by exchanging sensible heat or latent heat from a multi-component phase change

d) Utility streams which do not exhibit a temperature range, i.e. streams which transfer energy by exchanging latent heat from a single-component phase change

The inclusion of streams which undergo phase change and sensible heat change is achieved by modelling the three regions of liquid phase, two-phase and gas phase as separate streams. This ensures that the heat exchanger network design will be realistic by not allowing the occurrence of single- and two-phase regions in a single exchanger shell.

All process and utility streams which do not exhibit a temperature range are dealt with in a similar fashion. They are characterised by equal inlet and outlet temperatures and a latent heat flowrate, which results in two adjacent delimiters of equal temperature in the ordered list. In the case of two or more streams changing phase at the same temperature, each stream has its own pair of delimiters resulting in a string of paired delimiters with indentical temperatures. Associated with the interval inbetween each pair of delimiters is the interval heat content equal to the relevant stream latent heat flowrate. Since streams transferring energy by sensible heat or by multi-component phase change exchange no heat when a zero

temperature change occurs, they do not contribute to the interval heat content between the paired delimiters. Other single-component phase-change streams are limited to their own paired delimiters and, similarly, do not contribute.

Process streams which exhibit a temperature range are modelled as a set of sequential temperature ranges within which the heat capacity or latent heat flowrate is constant.

Utility streams which exhibit a temperature range are modelled in a different manner to that of the corresponding process streams. Process streams are generally matched in a sequential manner with only infrequent and very carefully controlled use of branching to allow parallel matching. This restriction does not apply to utility streams where widespread use is made of both serial and parallel matching. This makes the good design of utility systems a major exercise in its own right. This problem is circumvented by modelling these utility streams as a single-component phase change at the stream inlet temperature with all the energy exchange transferred to this point. In addition, a delimiter is placed in the temperature interval table at the stream exit temperature bound. This has no influence on the bounding procedure but marks the outlet temperature for the ordering heuristic which follows directly from it.

6.4.4 Transfer of match information

The match generator ascertains the heat to be transferred between the two designated streams and, in the case of process streams exhibiting a temperature range, the range satisfied. In addition, to prevent the computer repre-

sentation round-off problems that can occur because of the repetitive nature of the calculations carried out during tree searching, the matched streams are explicitly flagged as either satisfied (i.e. to be totally removed) or as still having unsatisfied ranges (i.e. only to be amended). Thus the bounding procedure receives, in addition to the bounding information from the level above, the following:

a) The identities of the two matched streams.

b) The heat to be transferred between them.

c) An indication if either or both of the streams becomes completely satisfied by the match.

d) The new temperature ranges of the matched streams if they are not completely satisfied and are process streams which exhibit a temperature range.

6.4.5 Matched stream status

The amendment algorithm starts at one end of the ordered list and works consistently towards the other. During this process the amendment of the data associated with the two matched streams passes through several phases. The exact nature of these phases is dependent on the stream type.

The amendment of the data associated with matched process streams that exhibit a temperature range passes through four phases (see figure 6.3). Since the match generator consistently matches from one end of the ordered list, the list can always be structured so that the portions of the matched streams to be removed are the highest parts of the streams on the list. Therefore, the four amendment phases

always appear in the following order:

Phase 1 — The matched stream does not exist at the current temperature. An end delimiter of the stream has not yet been encountered.

Phase 2 — The old end of the matched stream has been located but the new end has not. The delimiters associated with this stream are in the process of being deleted and the heat contents of the intervals are being corrected for the removal of the matched stream.

Phase 3 — The position of the new end of the matched stream has been determined. A new delimiter has been introduced to mark the new end and the heat content of the old interval has been distributed accordingly. No further amendment is required.

Phase 4 — The other end of the matched stream has been located and the stream no longer exists at the current temperature.

In the case where a stream has all of its remaining unsatisfied heat duty fulfilled in a match, phase three does not occur and the procedure passes directly from phase two to phase four.

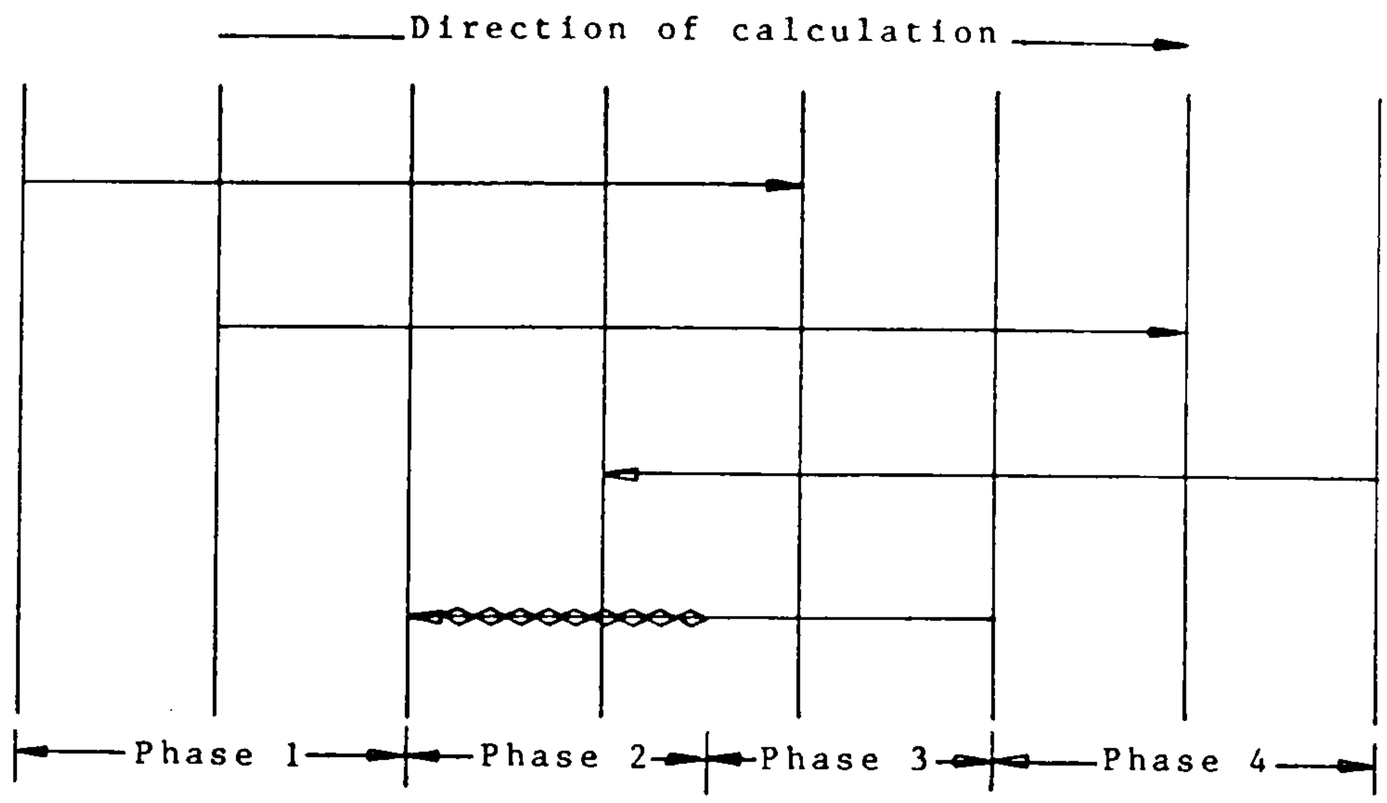


Figure 6.3 The four amendment phases of the data associated with a process stream exhibiting a temperature range

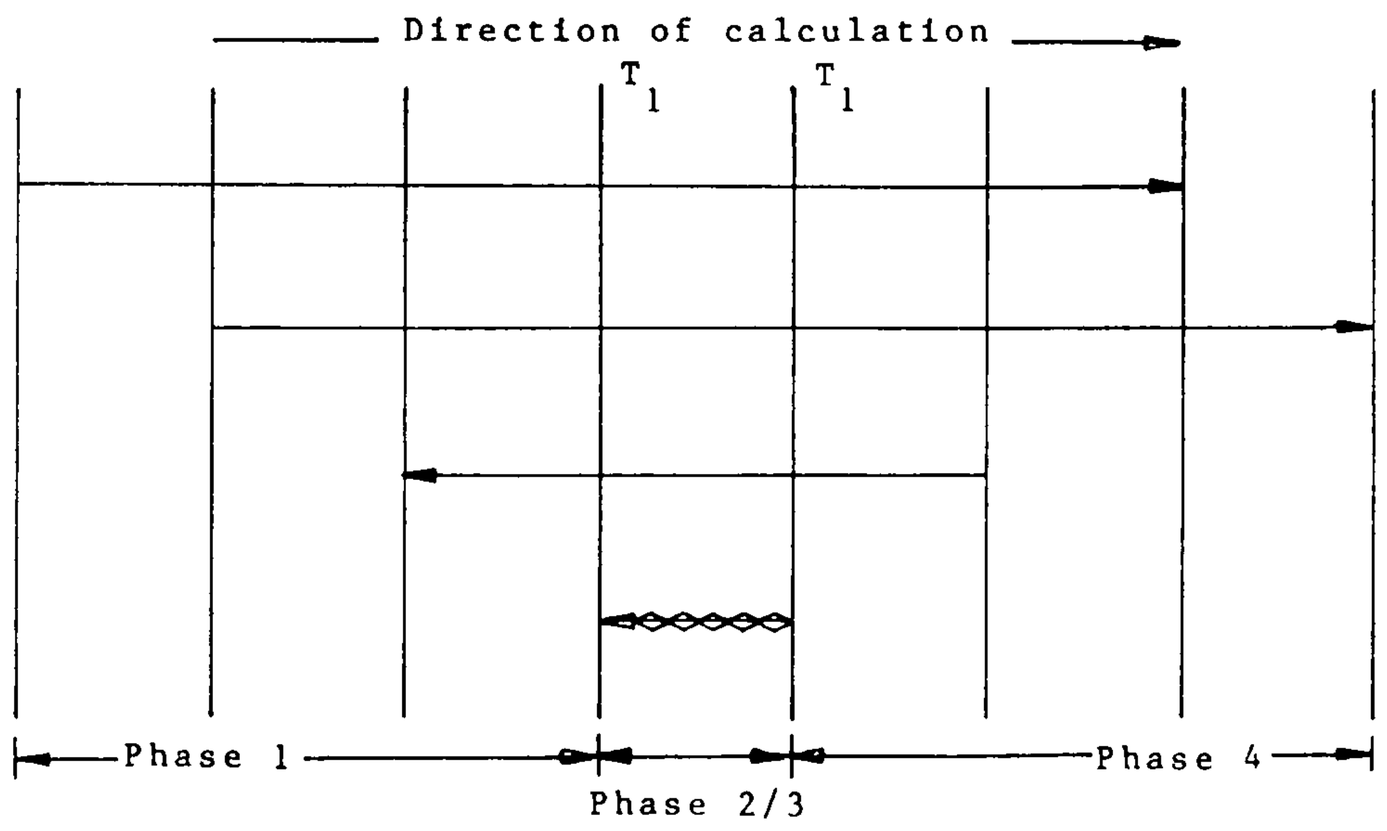


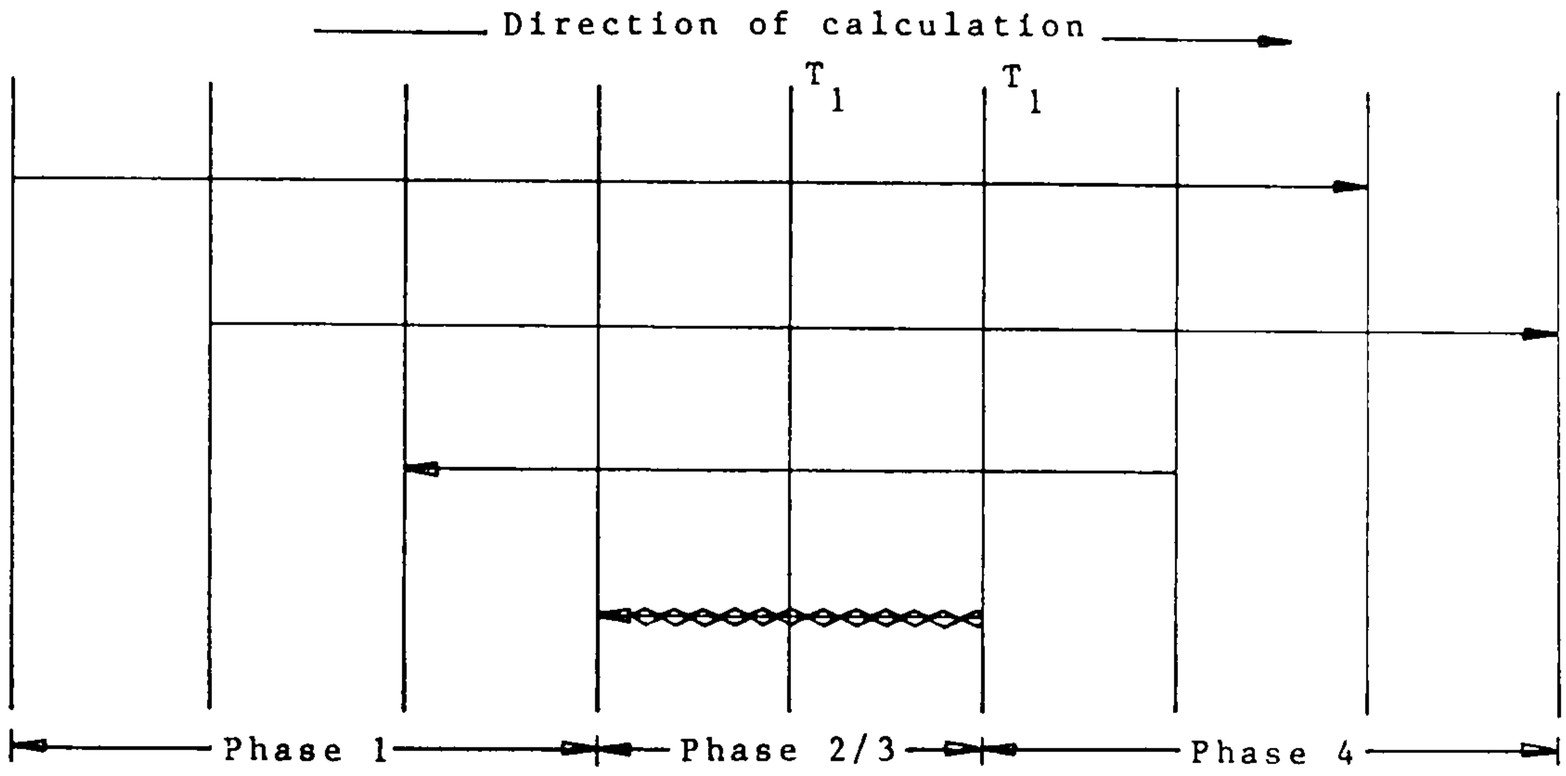
Figure 6.4 The three amendment phases of the data associated with a stream not exhibiting a temperature range

◇◇ denotes the matched section of a stream

The amendment of data associated with streams that do not exhibit a temperature range passes through three phases (see figure 6.4). The action taken depends on whether the stream has all of its remaining unsatisfied heat duty fulfilled or not; if so, both delimiters and the intermediate interval heat content must be deleted leading to an amalgamation of the old intervals immediately above and below; if not, the interval heat content merely requires amendment. Thus phases two and three outlined above must be replaced by:

Phase 2/3 An end of the matched stream has been located. If the stream is to be deleted the next two delimiters and the intermediate heat content are in the process of being removed, otherwise the heat content is being corrected.

Finally, utility streams that exhibit a temperature range are treated in a similar manner to the streams not exhibiting a temperature range but with an allowance for the use of three delimiters rather than two. The entire heat content of these streams is located in the intervals bounded by a pair of delimiters at each stream inlet temperature, similar to a stream not exhibiting a temperature range. The dummy delimiter marking the outlet temperature bound is ignored unless deletion of the stream is required (see figure 6.5).



◇◇denotes the matched section of a stream

The stream may, or may not, be deleted depending on the heat transferred in the match.

The interval containing the matched stream heat content can be either the first or last interval on the stream

Figure 6.5 The three amendment phases of the data associated with a utility stream exhibiting a temperature range

6.4.6 A procedure to locate the first new delimiter

The process of amending the bounding information calculated for one level of the search tree in order to provide the bounding information for the next level down starts with the determination of the position of the first new delimiter. Each of the delimiters on the old list is considered in turn until the first suitable candidate for the new list is found (see flowsheet given in figure 6.6). There are four possibilities:

a) There is no first new delimiter; the network is complete (see figure 6.7).

b) The first new delimiter is that associated with the start or the end of one of the unmatched streams. (see figure 6.8).

c) The first new delimiter is that associated with the, possibly shifted, end of the hot stream just matched (see figures 6.9 and 6.10).

d) The first new delimiter is that associated with the, possibly shifted, end of the cold stream just matched (see figures 6.11 and 6.12).

The process is complicated by the need to cope with various types of streams. In particular, the movement of the ends of process streams exhibiting a temperature range compared with the fixed ends of the other streams necessitates special action.

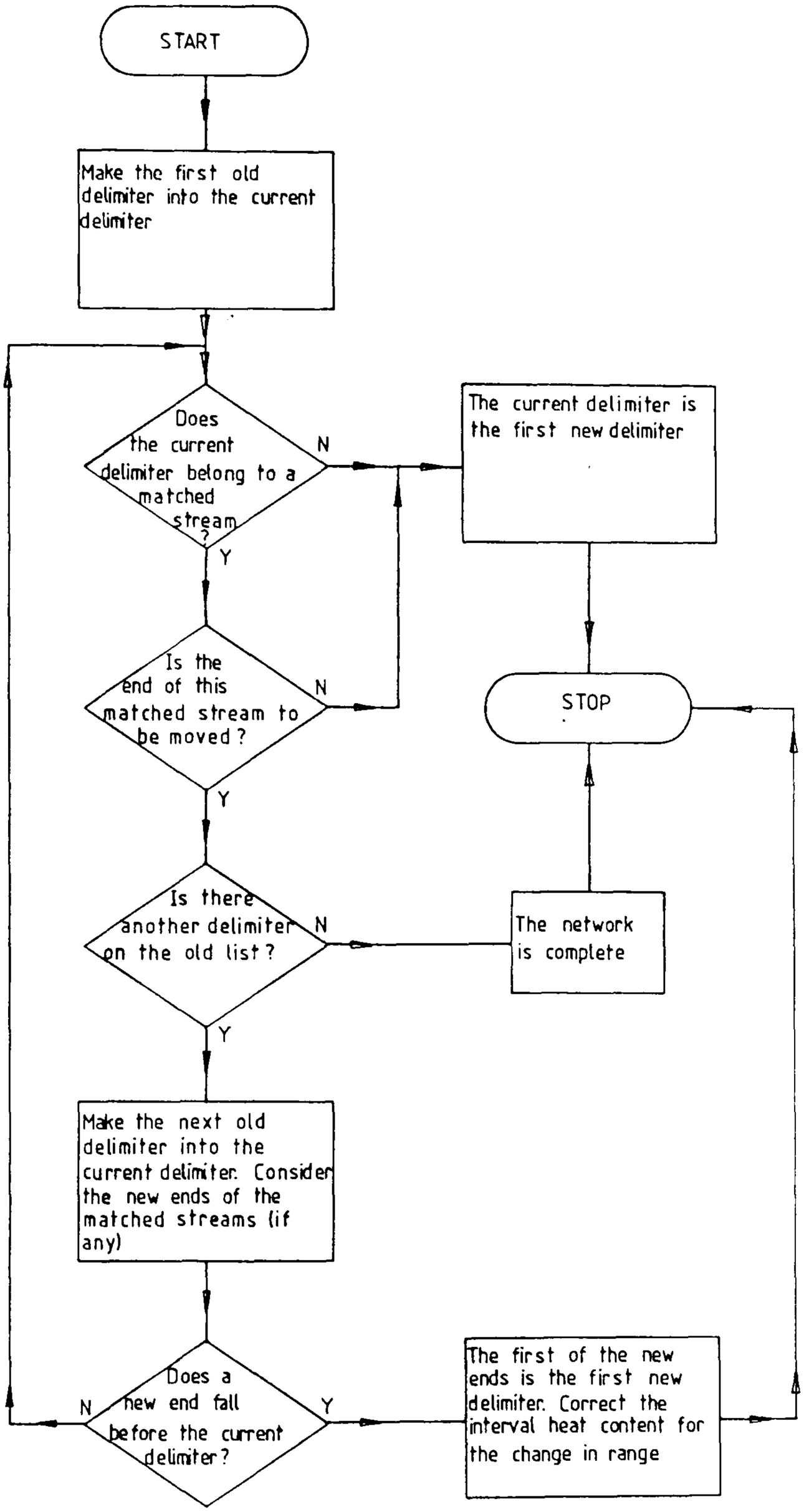


Figure 6.6 A procedure to locate the first new delimiter

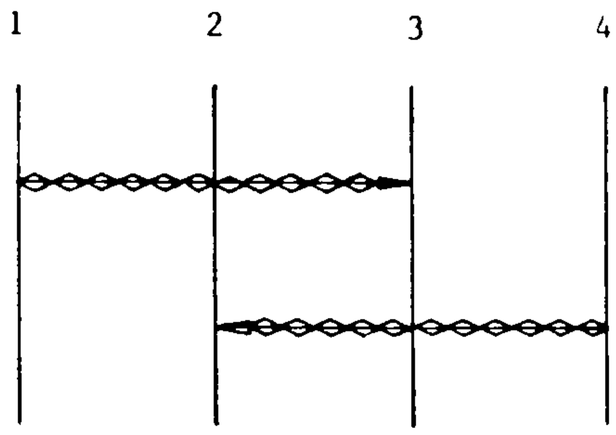


Figure 6.7 No streams are left in the interval diagram

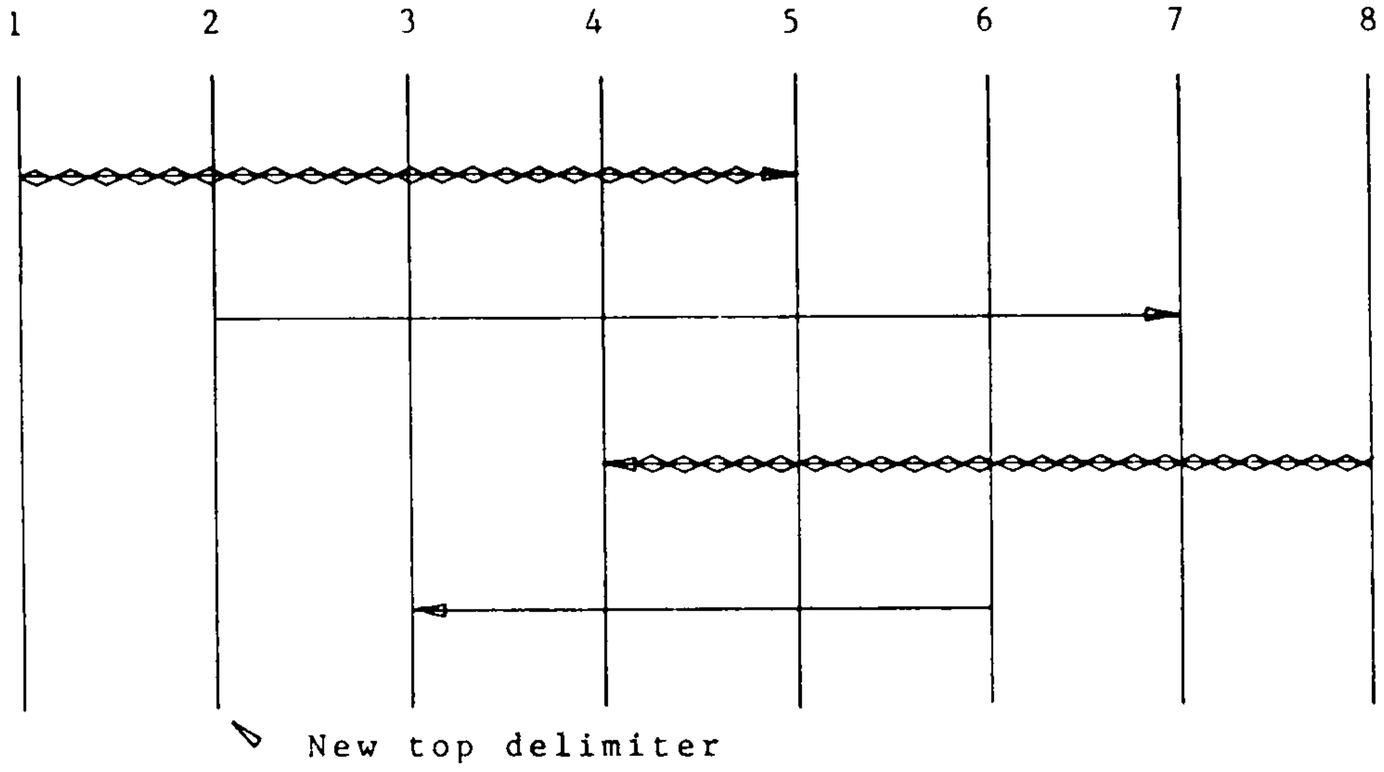


Figure 6.8 An old delimiter becomes the new top delimiter

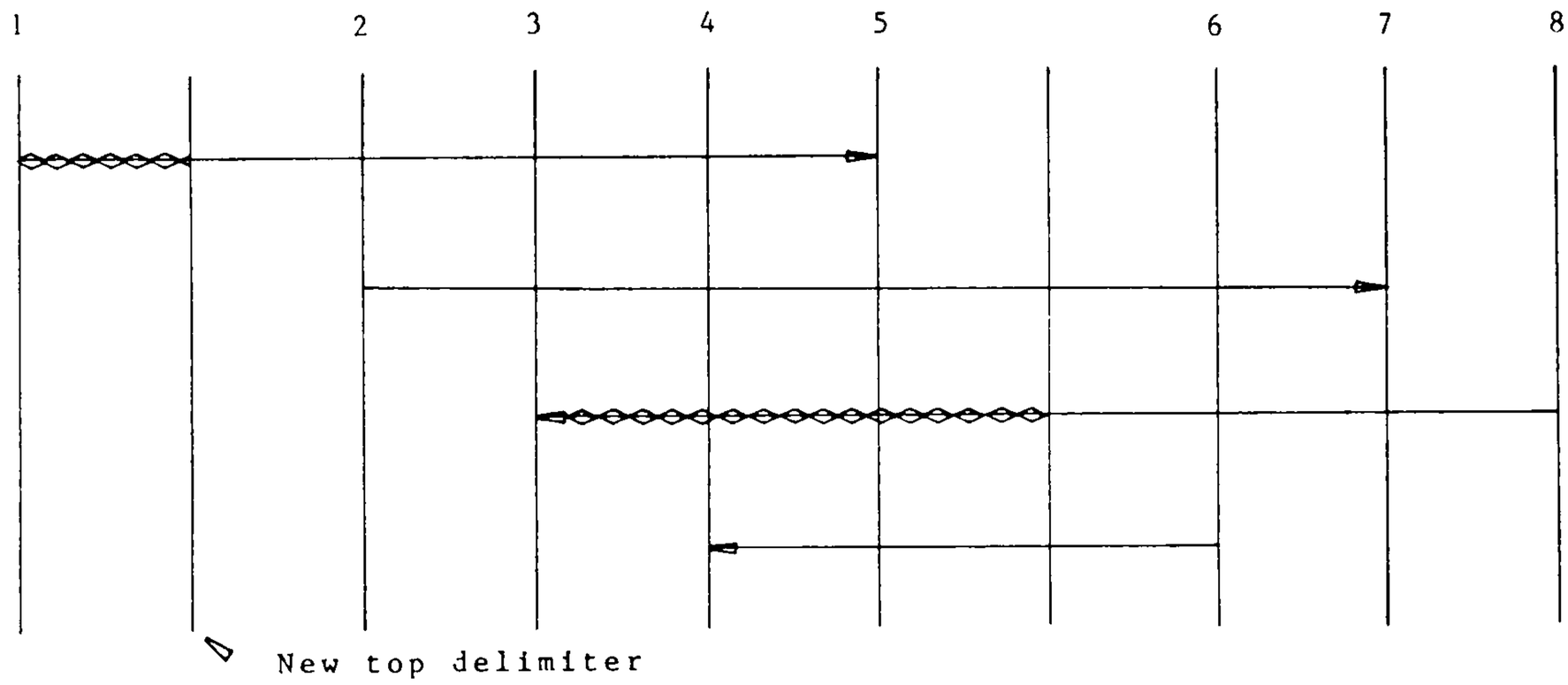


Figure 6.9 The new hot stream end delimiter becomes the new top delimiter

⋈ denotes the matched section of a stream

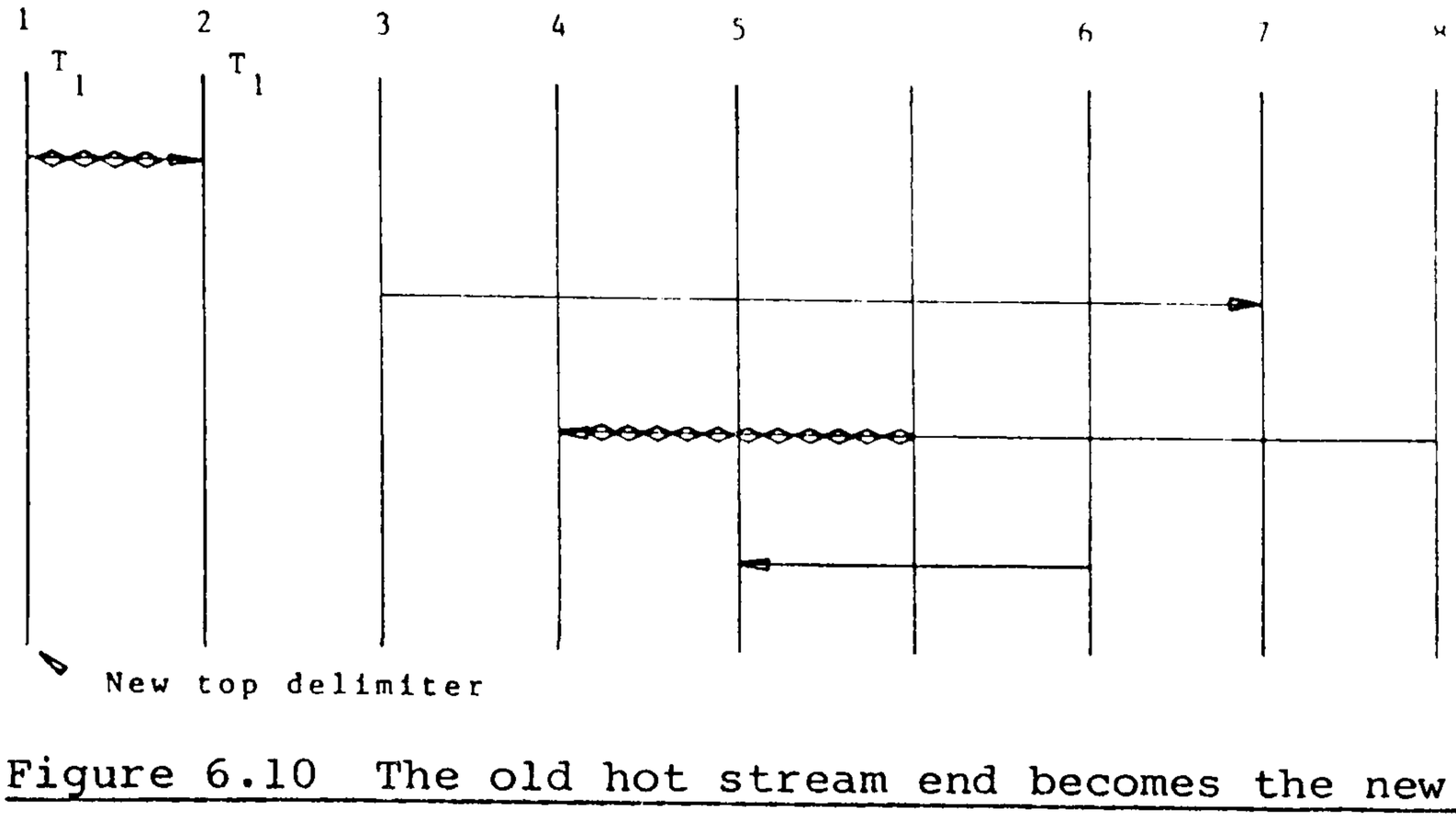


Figure 6.10 The old hot stream end becomes the new top delimiter

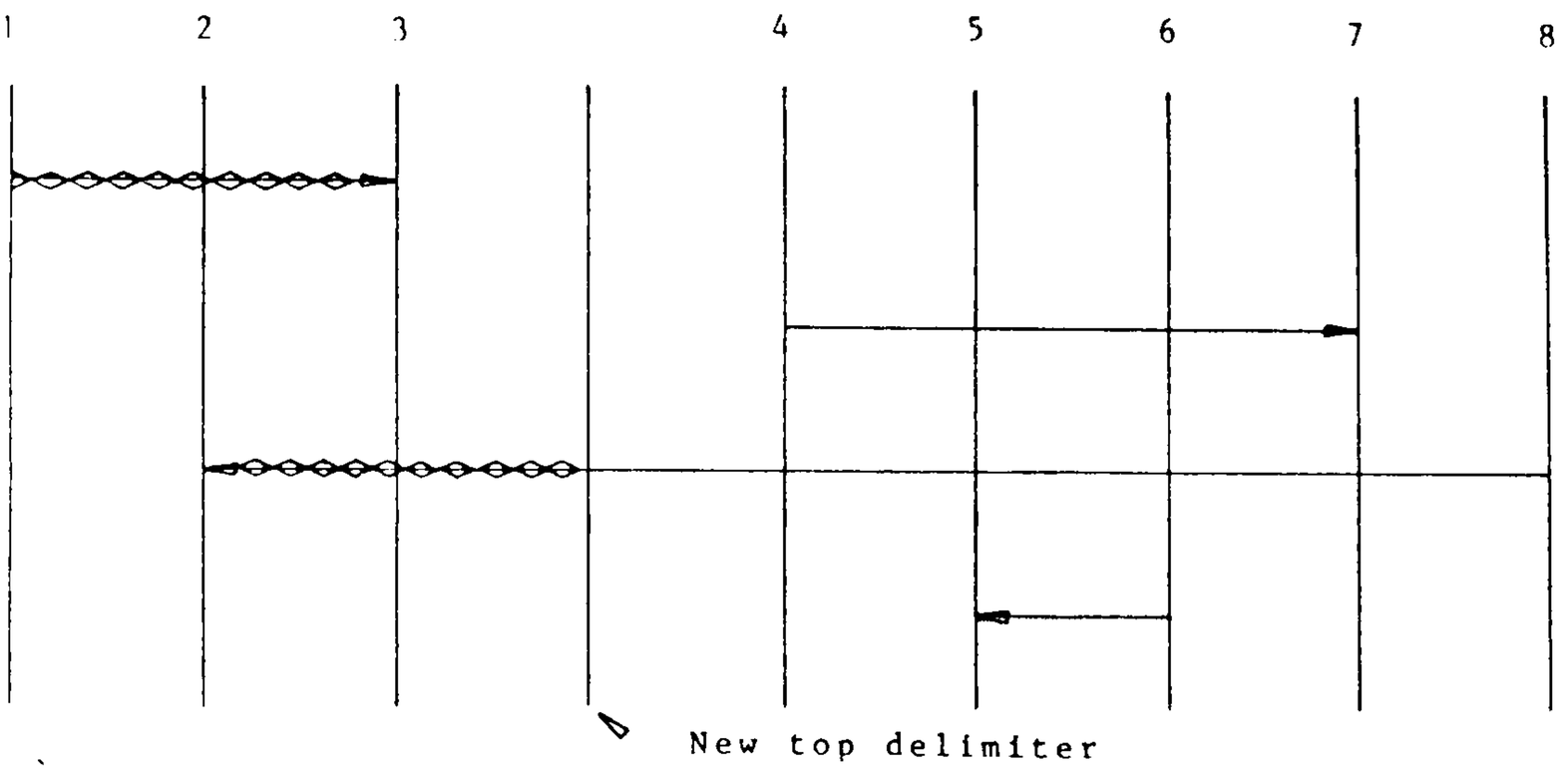


Figure 6.11 The new cold stream end becomes the new top delimiter

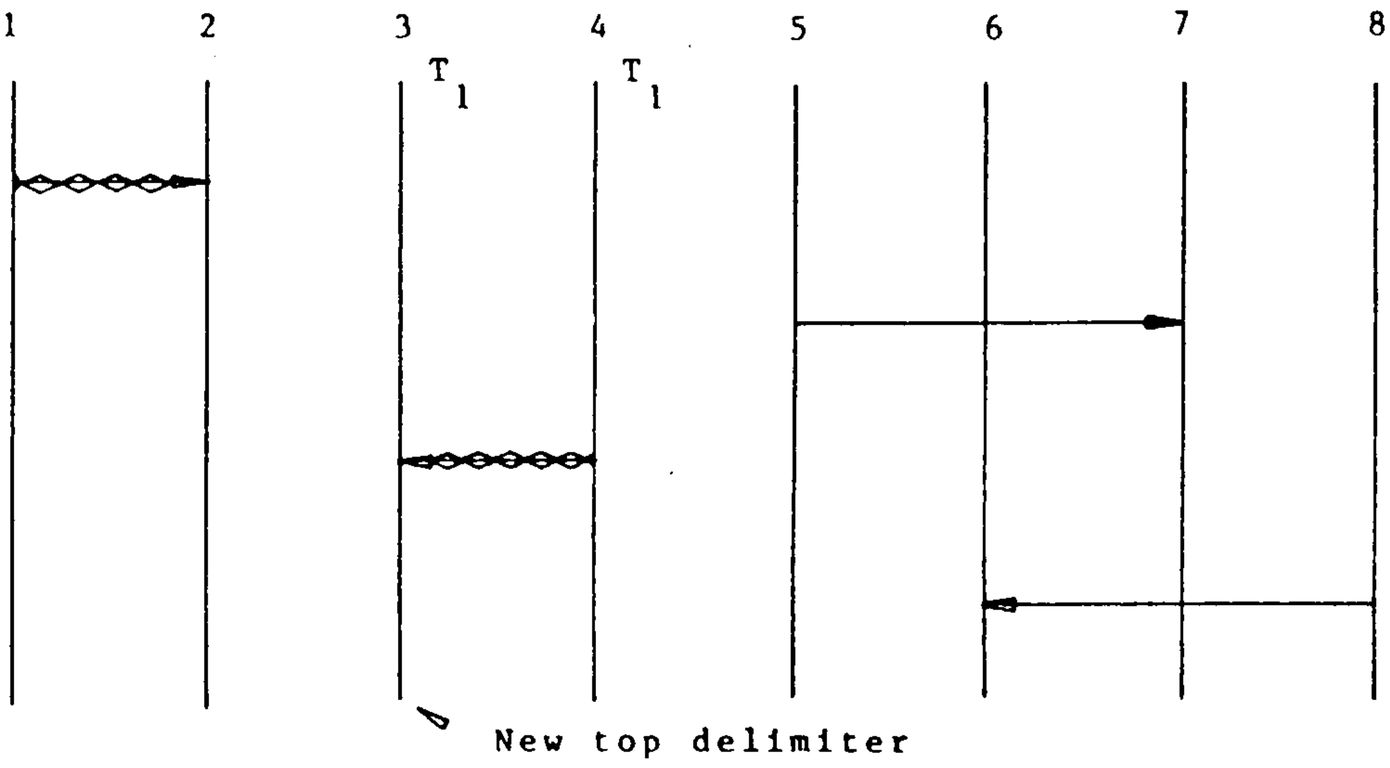


Figure 6.12 The old cold stream end becomes the new top delimiter

◇◇ denotes the matched section of a stream

6.4.7 Treatment of subsequent delimiters and heat contents

Once one of the delimiters of a new interval has been fixed, the next delimiter on the old list is considered in an attempt to find the range of the new interval (see flowsheet given in figure 6.13). There are four possibilities for each delimiter on the old list:

a) There are no more delimiters; the amendment is complete.

b) A new delimiter is to be added to the new list before the processing of the next delimiter on the old list (see figure 6.14).

c) The next delimiter on the old list is to be transferred to the new list (see figure 6.15).

d) The next delimiter on the old list is not to be transferred to the new list (i.e. it is to be deleted, see figures 6.16 and 6.17).

Up to two new delimiters, the possible new ends of the matched streams, may have to be considered for addition to the new list during the amendment procedure and the correct order must be maintained. On introducing a new delimiter the old heat content must be split into two parts, one of which completes the current new interval and the other starts the next new interval.

Each old interval is considered in turn; its heat content is amended and added to the heat content of the current interval on the new list until the next new delimiter is found. This is then entered on the new list and a new search started for the next delimiter whilst summing the

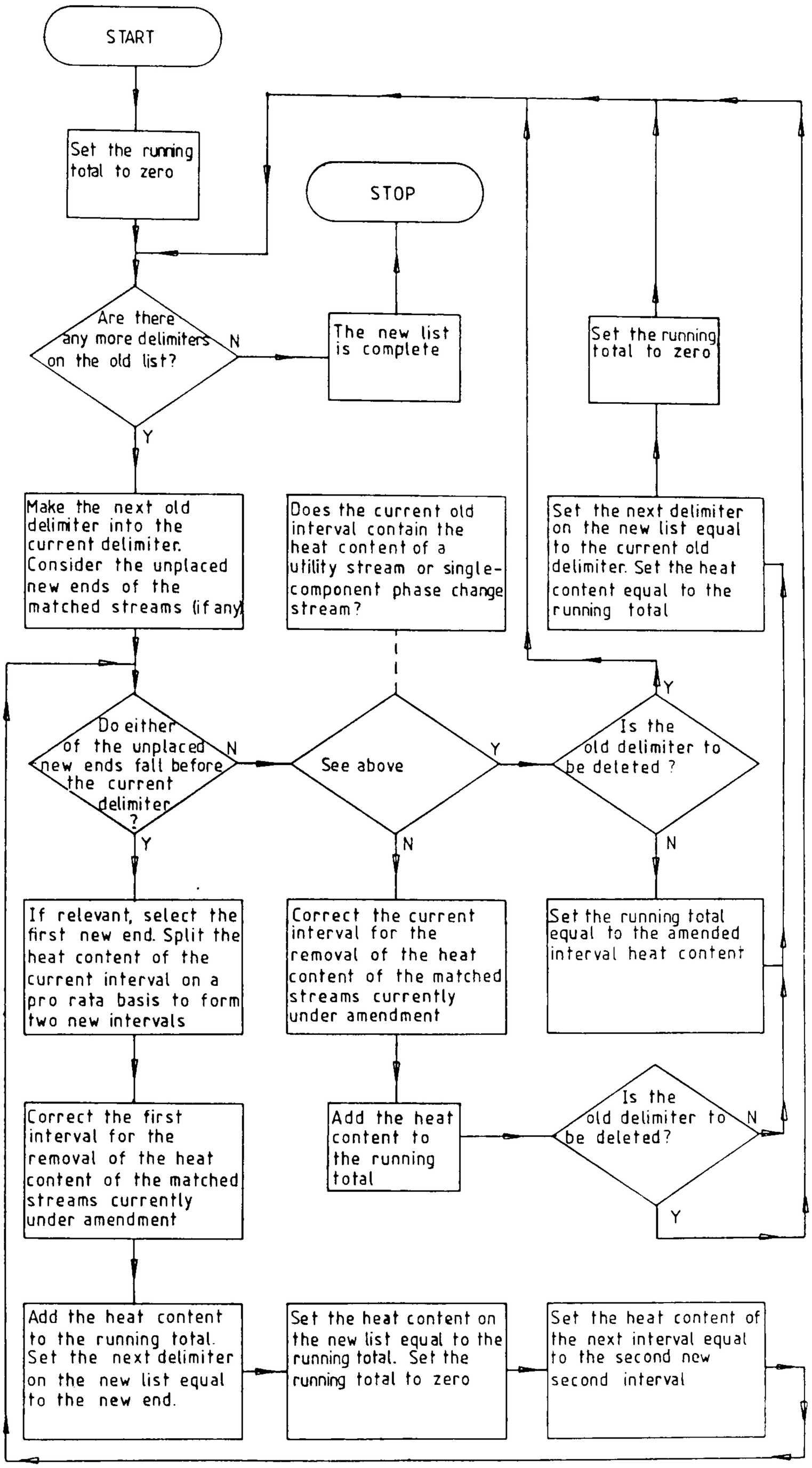


Figure 6.13 The treatment of subsequent delimiters and heat contents

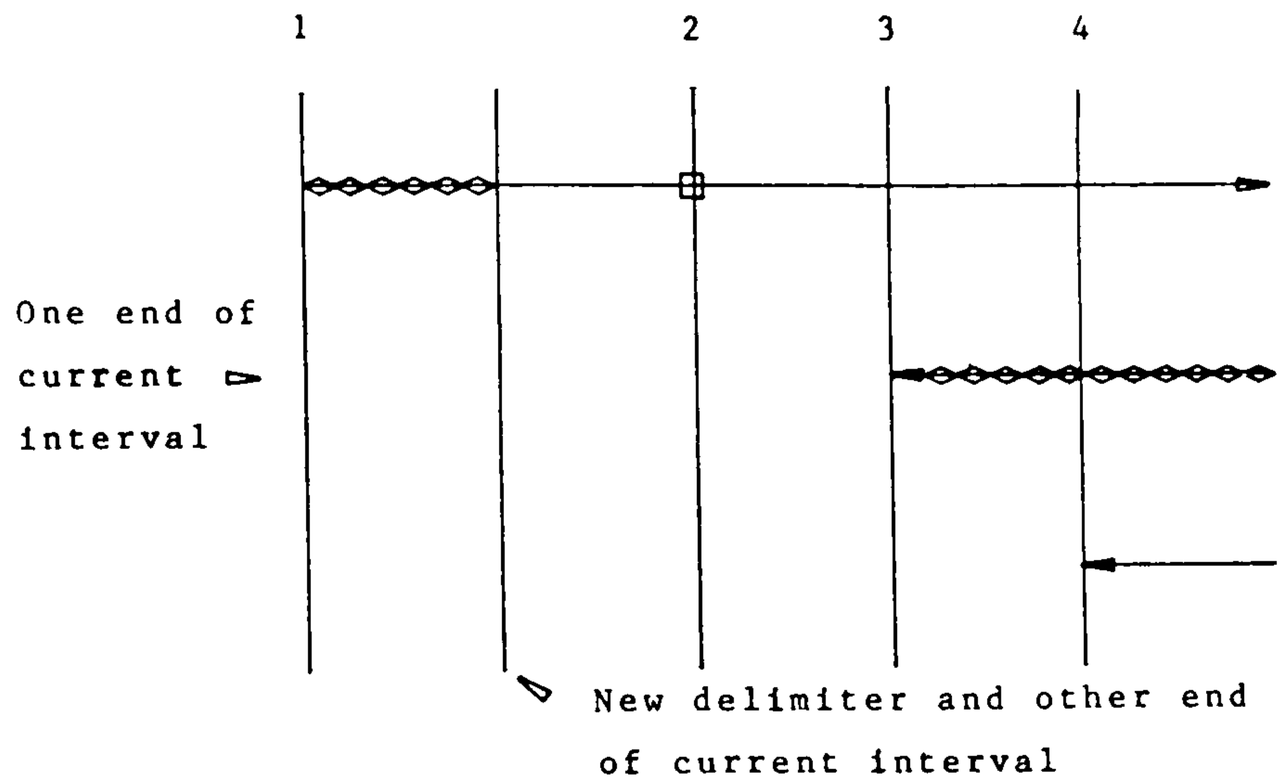


Figure 6.14 A new stream end comes before the next old delimiter and marks the end of the current new interval

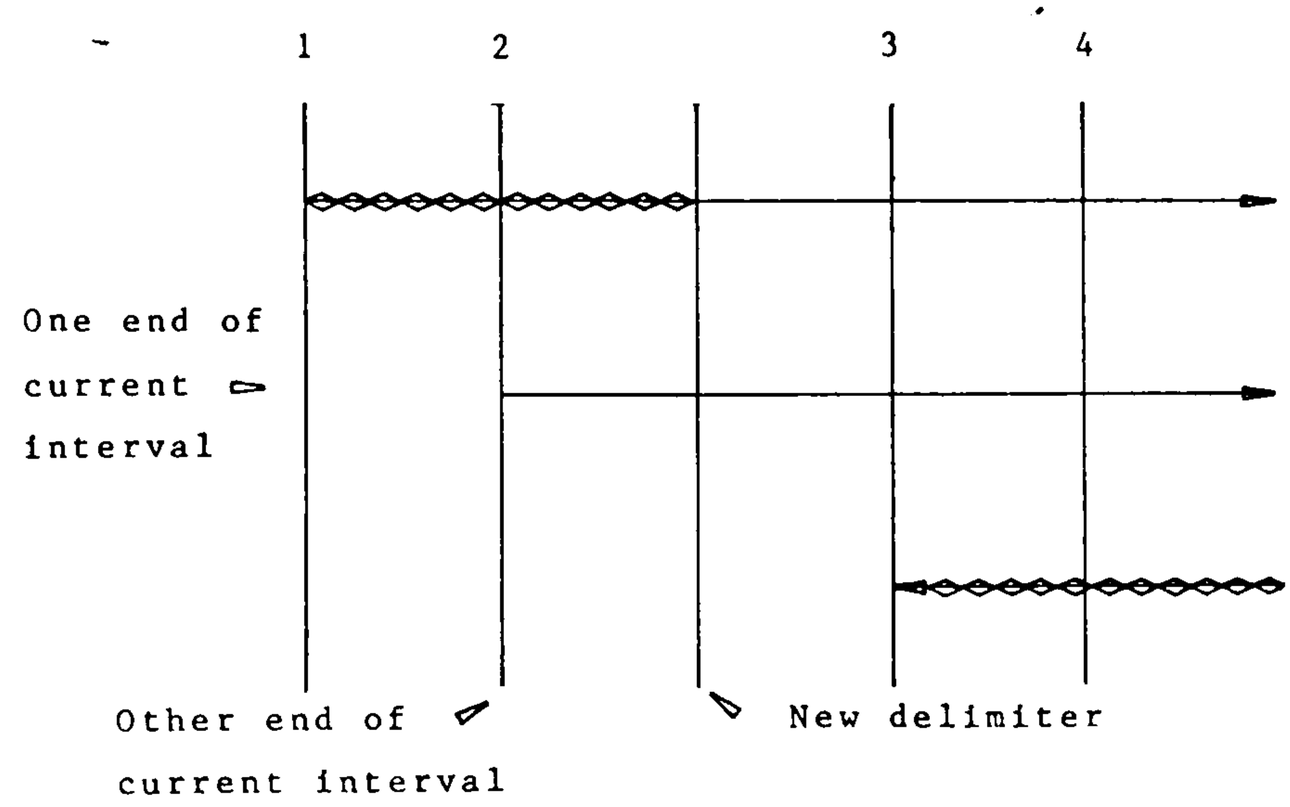


Figure 6.15 The next old delimiter marks the end of the current new interval

◇◇ denotes the matched section of a stream

□ denotes a change in heat capacity flowrate of a stream

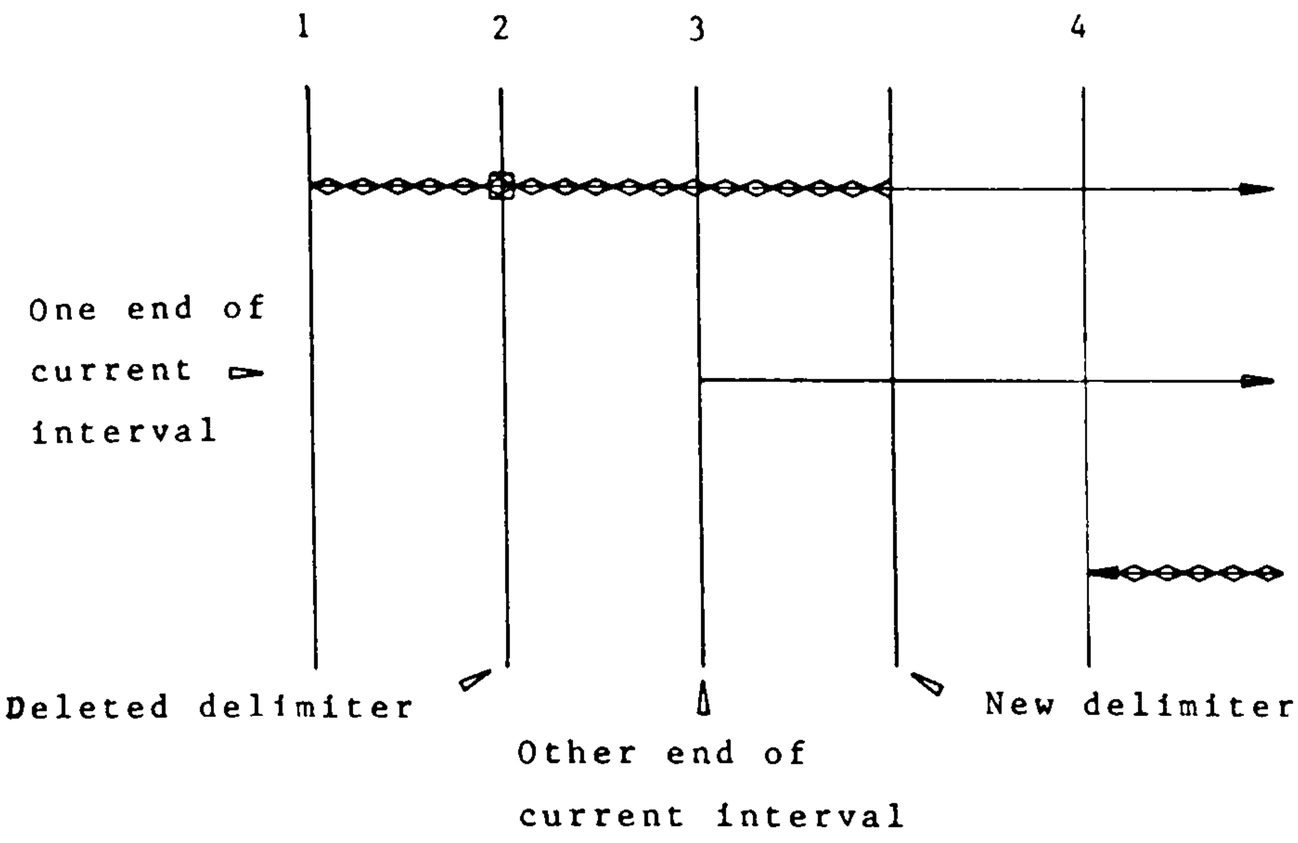


Figure 6.16 The next old delimiter does not mark the end of the current new interval because a change in the heat capacity flowrate of a matched stream is associated with it

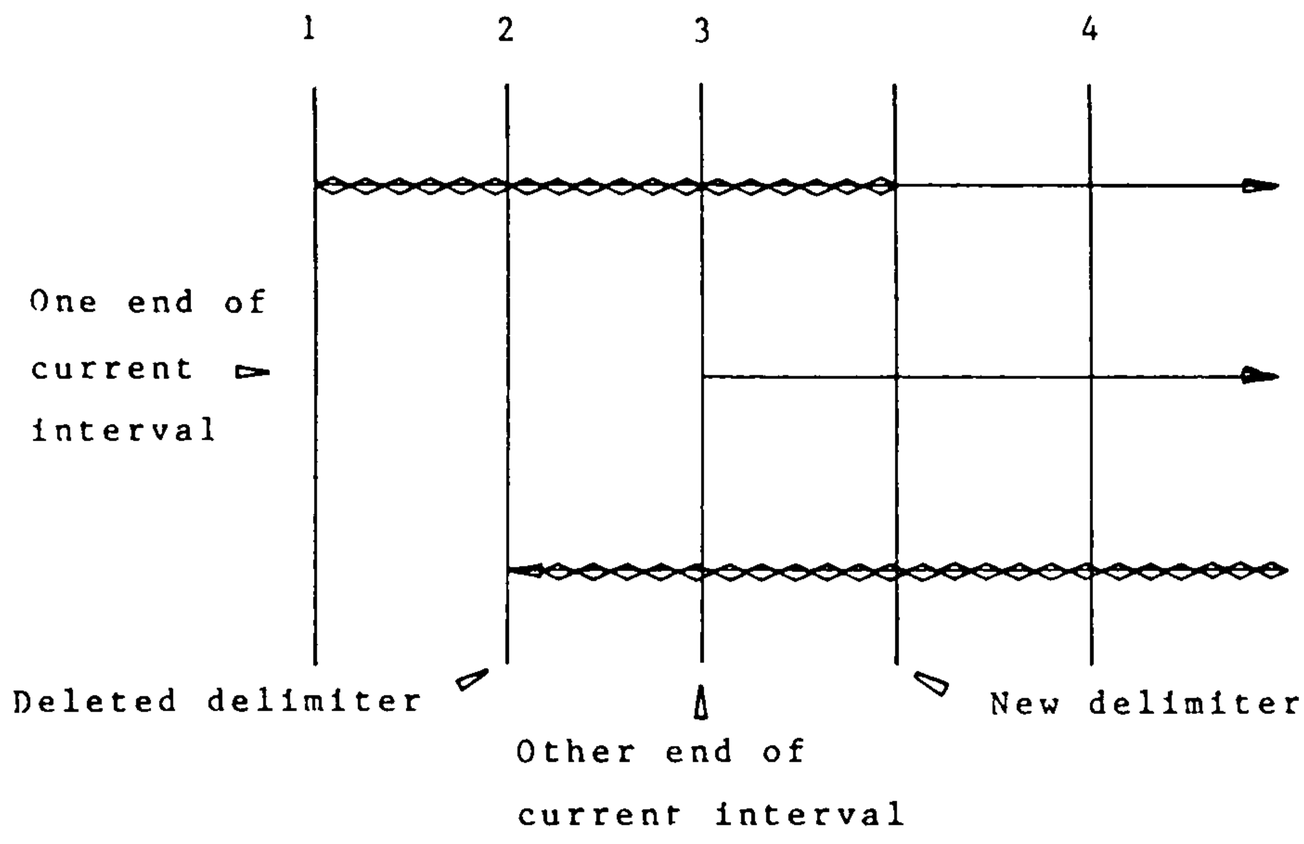


Figure 6.17 The next old delimiter does not mark the end of the current interval because a matched stream old end is associated with it

◇◇ denotes the matched section of a stream

□ denotes a change in heat capacity flowrate of a stream

old amended heat contents in a similar fashion. The amendment of heat contents is only necessary when a matched stream is in its second phase.

6.5 THE BRANCHING STRATEGY

The branching strategy used in this network synthesis problem is similar to that used in the stream-splitting problem. The ordered list which controls the enumeration of the immediate successors to a node is composed, in itself, of two ordered lists, one for the hot streams and one for the cold streams. However, since an exhaustive search of the solution space is not always required when considering this particular problem, the ordered lists generated do not necessarily contain a comprehensive collection of all the possible matches. A heuristic is included in order to limit the number of members on the ordered lists of streams so that each entry lies within a certain temperature range of the first member of the list (see section 6.5.1). These entries are the nodes which are most likely to lead to an optimal solution. The value of the temperature range which precludes extra streams being added to the ordered lists is a user specified parameter. A further constraint is also imposed on the expansion of the search tree by allowing an upper limit to be placed on the number of solutions to be found at any level of the tree. Once this limit has been reached the upper bound on the objective function is decremented by one and the search continued (i.e. the tree is pruned when a stated number of solutions have been found at any one level).

Both the optimal value for the temperature cut-off range and the number of solutions at which pruning occurs are problem dependent but suitable values are suggested in the

next chapter.

6.5.1 Determination of the ordered lists for the branching strategy

The ordered lists for the branching strategy at each level can be obtained at negligable computational cost from the relevant temperature interval table. After a match is made the temperature interval table for the relevant node is amended by considering each of the delimiters in the table in turn. Each time the first delimiter of a stream is placed into the new table its stream number is placed onto the list of ordered streams. Since the temperature interval table is a set of ordered delimiters, the correct order is established in the stream-ordering list for the branching strategy. The heuristic to limit the number of members on the ordered lists using a temperature range is catered for easily by comparing the temperature of the stream which is about to be added with the first member of the list. Only if the temperature difference is smaller than or equal to the stated temperature range is the stream added to the list.

6.5.2 Unique solutions

The same initial procedure as that used in the stream splitting algorithm to ensure only one permutation of the disjoint elements occurs can be used in the solution of this problem (see section 5.13.4.1). However, the amendments made to screen out permutations of multiple row or column entries in the stream-splitting problem are not always applicable because the equivalent matches in this problem are generally order dependent. There are two exceptions to this generalisation. These involve the

matching of streams which undergo a single-component phase change and the matching of utility streams.

A stream that undergoes only a single-component phase change obviously maintains a constant temperature throughout the network. Thus matches with other streams are order independent except for the last match on the single-component phase-change stream. This means that the amended procedure developed for the stream-splitting problems can be applied to these phase-change streams in the heat exchanger network synthesis problem.

Matches involving utility streams can usually be considered to be order independent because utility streams are normally split into a sufficient number of parallel branches to service each utility exchanger individually. Thus the amended procedure can again be used to minimise the number of solution permutations.

6.6 INITIAL VALUE OF THE UPPER BOUND ON THE OBJECTIVE FUNCTION

A computer program, HERNE the Hunter (HEat Recovery NEtwork Hunter), has been developed which contains the above procedures. The initial value of the upper bound on the objective function can either be entered as a number or set automatically to the maximum that the software is designed to handle. The conditions under which each of the two initial upper bounds should be used is discussed in the next chapter.

Herne the Hunter is a phantom hunter who is supposed to haunt Windsor Great Park and who was impersonated by Falstaff in 'The Merry Wives of Windsor'. Herne is probably a manifestation of the wild hunstman myth known throughout the world as a person excessively fond of the chase who makes a rash pledge with a stranger, the devil, and is thus doomed to hunt forever. It is hoped that this software is more efficient at catching its prey.

6.7 SUMMARY

A complete depth-first branch-and-bound tree-searching procedure has been developed to solve the heat exchanger network synthesis problem. A sophisticated match generator has been created to allow non-counter-current exchangers of various types, as well as counter-current exchangers, to be incorporated in a variety of ways. An efficient bounding algorithm has been developed which decreases the computation required at each level of the search tree by using the bounding information calculated for the level above. Thus the usual, computationally expensive, algorithm for calculating the bound need only be applied at the root node. As a result of applying this particular bound the ordered lists required for the branching strategy are obtained at negligible computational cost.

Application of this solution method to several practical problems is shown in the next chapter.

CHAPTER SEVEN

THE APPLICATION OF THE HEAT EXCHANGER NETWORK SYNTHESIS ALGORITHMS TO SOME PRACTICAL PROBLEMS

7.1 INTRODUCTION

In the following chapter seven problems that have appeared already in the literature (4SP1, 4SP2, 5SP1, 7SP2, 7SP4, TC1 and an unnamed, six process stream problem) are reinvestigated with a view to finding solutions which use only 1/2 shell-and-tube heat exchangers. Three of these problems (4SP2, 5SP1 and 7SP2) are unpinched and the other four pinched.

Two new problems are also introduced, one of which is pinched and the other unpinched. These involve more process streams than the other examples and are real processes for which heat exchanger networks already exist. These new problems highlight the strengths and deficiencies of the new methods to a greater extent than the literature problems mentioned above and are derived from data supplied by B.P. International plc. The heat capacity /latent heat flowrates and temperatures have been altered to retain confidentiality.

Optimal and near-optimal solutions containing counter-current exchangers which are given in the literature for the standard problems are modified for use with 1/2 shell-and-tube exchangers. Each counter-current match is replaced by the minimum possible number of 1/2 exchangers that is compatible with disallowing temperature crosses in individual shells. This follows directly from the results in chapter three. These networks then form the basis with

which to compare the new flowsheets generated.

The cost of each network is based on the sum of the UA values of each exchanger in the network raised to the power 0.6 (U being the over-all heat transfer coefficient and A the heat transfer surface area). The calculation of the UA values takes account of the log-mean temperature difference correction factor required for 1/2 heat exchangers. The costs can be considered to provide only a very rough guide, accurate to possibly ± 20%, and are used merely as an indication. Indeed, the costs of networks which contain equal numbers of exchangers have been found to be remarkably constant with the largest range exhibited by a set of solutions being no larger than the possible inaccuracy in any one cost. Thus network costs are not used to screen candidate configurations.

In all seven problems maximum energy recovery for the stated minimum temperature difference is specified and the networks generated on that basis.

7.2 PROBLEM ONE

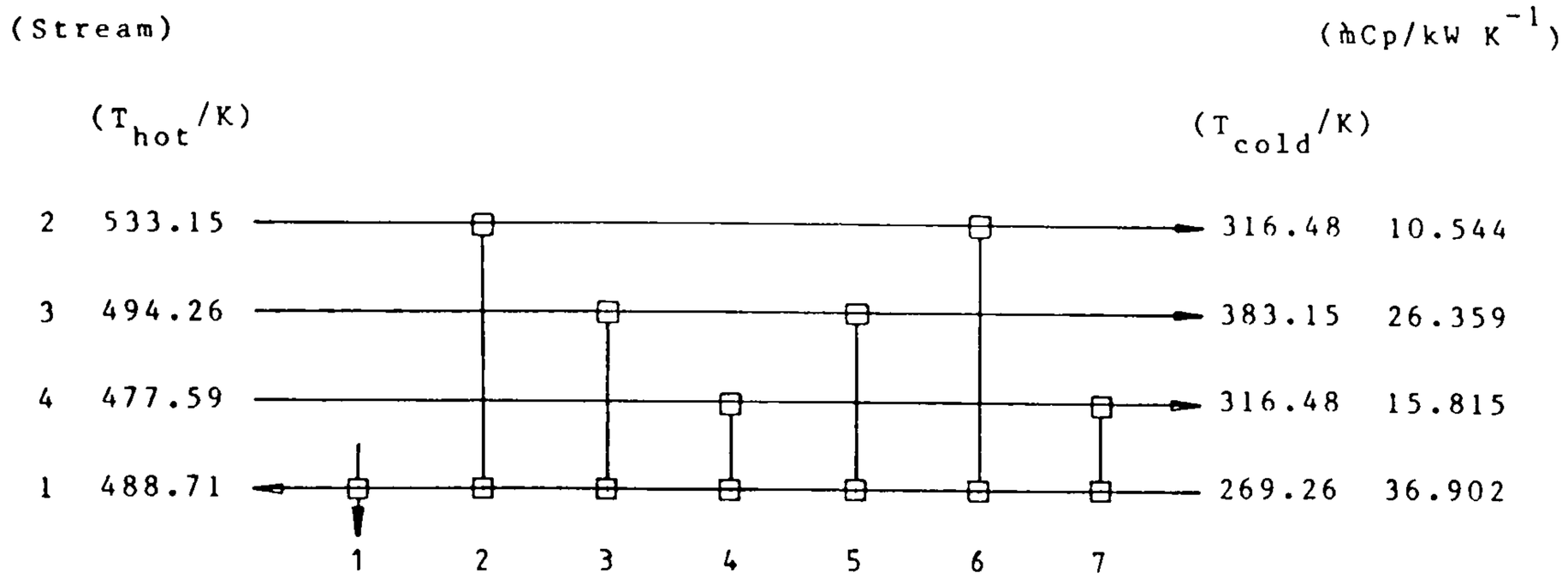
The first problem to be discussed here was introduced by Ponton and Donaldson (1970) and is known conventionally as 4SP2. A summary of the relevant data is given in appendix M. The construction of the temperature interval table for the root node shows that the problem is unpinched and that only hot utilities are required.

Two solutions that have become widely accepted as near-optimal, if counter-current exchangers are permitted, were first given by Linnhoff and Flower (1978b) and are reproduced in figures 7.1 and 7.2. The first of these solutions contains no stream splits whereas the second solu-

tion features a three-way branch of the cold stream. However, if the counter-current heat exchangers in these solutions are replaced by the minimum feasible number of 1/2 shell-and-tube exchangers, as an example of non-counter-current exchangers, the networks become much more complex (see figures 7.3 and 7.4). The unsplit solution now contains fourteen shells and the solution with the stream split eighteen shells. The need for parallel stream splits is therefore brought into serious question if maximum energy recovery can be achieved by other means.

The new computer program, HERNE, was applied to this problem and a complete tree search carried out. Two solutions were found which contain just ten shells. These are given in figures 7.5 and 7.6. In an attempt to increase the 'modularity' of the networks (i.e. to increase the number shells with the same heat transfer area) HERNE was applied for a second time to the problem. On this occasion pairs of 1/2 heat exchangers were included wherever possible (see section 6.3.1). A total of eleven networks were obtained, each composed of thirteen shells, with cost indicators ranging from 76.835 to 81.167. Two example solutions are given in figure 7.7 and 7.8. In this particular example the pairing up of heat exchangers is obviously not worthwile because the number of shells is increased by an unacceptable factor.

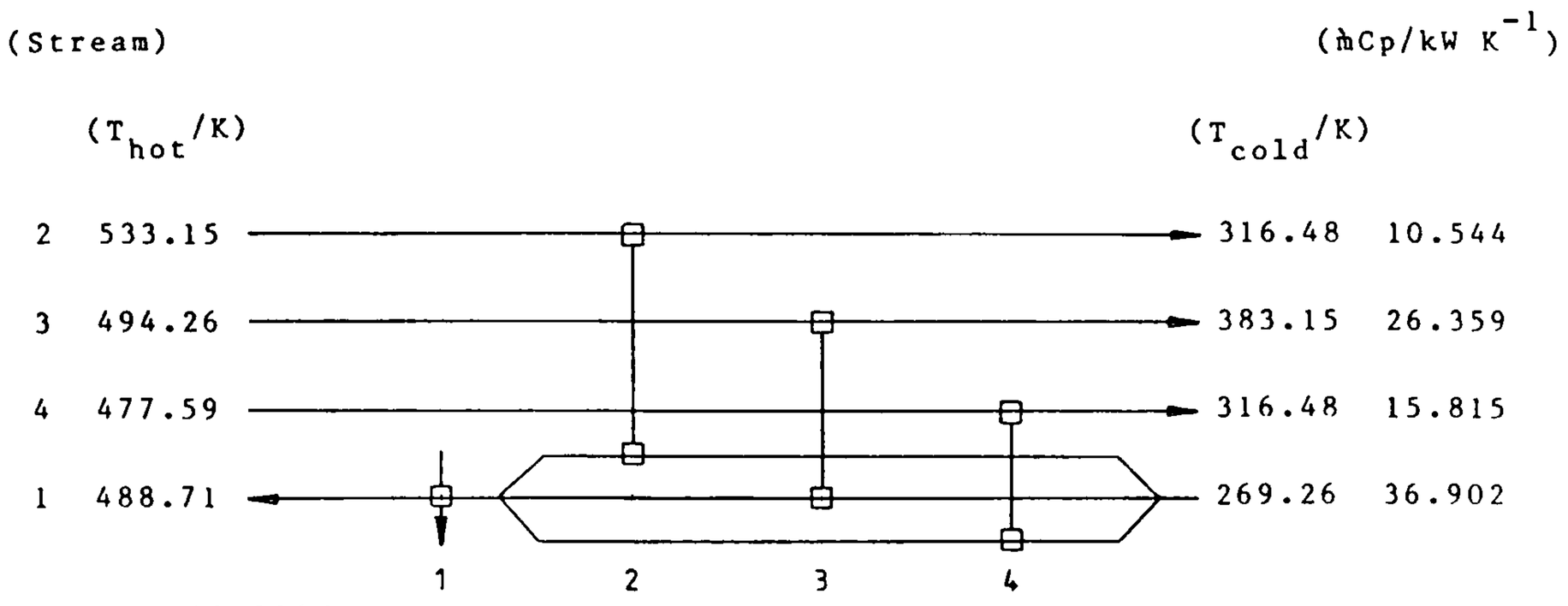
Problem 4SP2 was also considered by Zachoval and Konecny (1982) in the only known application of their approach to the synthesis of networks containing 1/2 exchangers. The solution they obtained is given in figure 7.9. It contains one more shell than the two best networks obtained in the current work.



ΔT_{min} = 11.111K

Heat exchanger loads/kW
1 : 336.9 2 : 620.7 3 : 1856.5 4 : 1493.8 5 : 1072.2
6 : 1663.9 7 : 1054.2

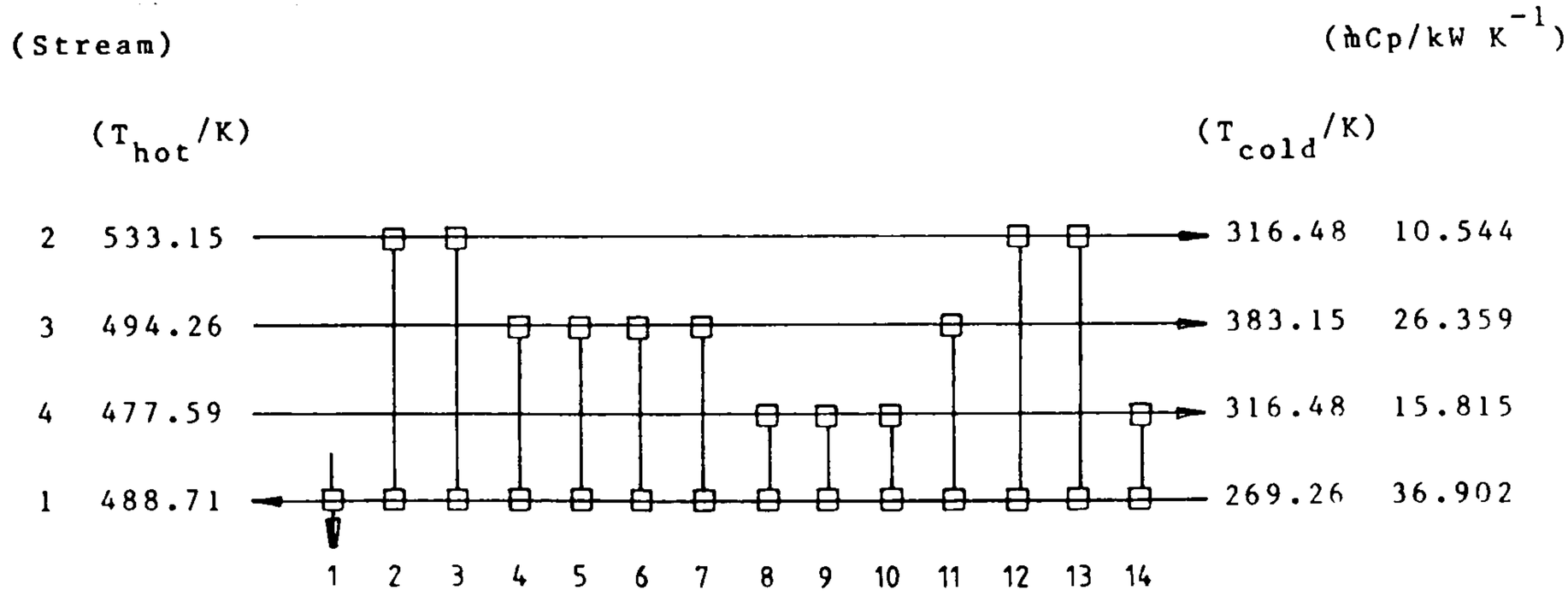
Figure 7.1 A near-optimal solution to 4SP2 using counter-current exchangers and no stream splits



ΔT_{min} = 11.111K

Heat exchanger loads/kW
1 : 336.9 2 : 2284.6 3 : 2928.7 4 : 2548.0

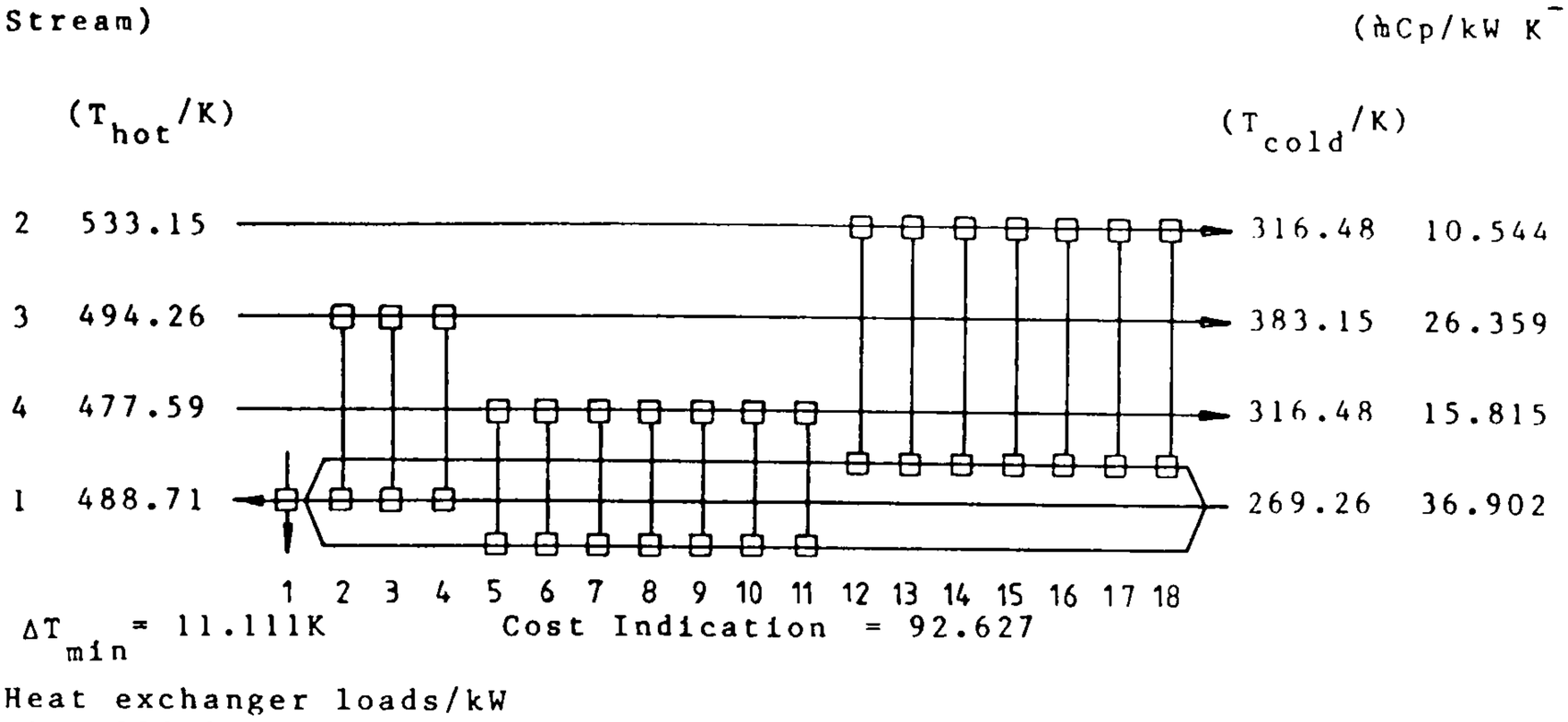
Figure 7.2 A near-optimal solution to 4SP2 using counter-current exchangers and a three-way stream split



ΔT_{min} = 11.111K Cost Indication = 81.531

Heat exchanger loads/kW
1 : 336.9 2 : 564.8 3 : 55.8 4 : 830.3 5 : 593.1
6 : 423.7 7 : 9.4 8 : 1030.3 9 : 441.7 10 : 21.8
11 : 1072.2 12 : 1385.3 13 : 278.7 14 : 1054.3

Figure 7.3 The solution given in figure 7.1 using 1/2 exchangers

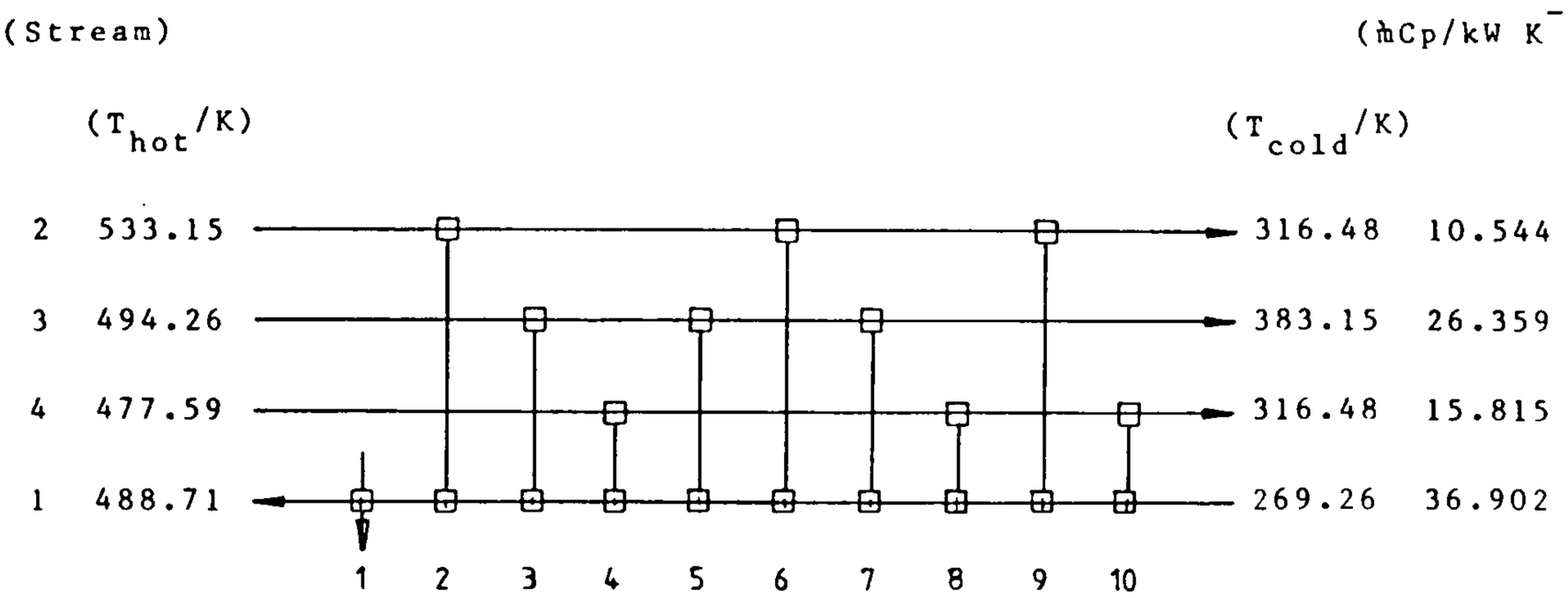


ΔT_{min} = 11.111K Cost Indication = 92.627

Heat exchanger loads/kW

1 : 336.9	2 : 507.7	3 : 940.1	4 : 1481.0	5 : 230.7
6 : 276.1	7 : 330.4	8 : 395.4	9 : 473.2	10 : 566.3
11 : 275.7	12 : 233.4	13 : 260.6	14 : 290.7	15 : 324.4
16 : 361.9	17 : 403.7	18 : 409.8		

Figure 7.4 The solution given in figure 7.2 using 1/2 exchangers

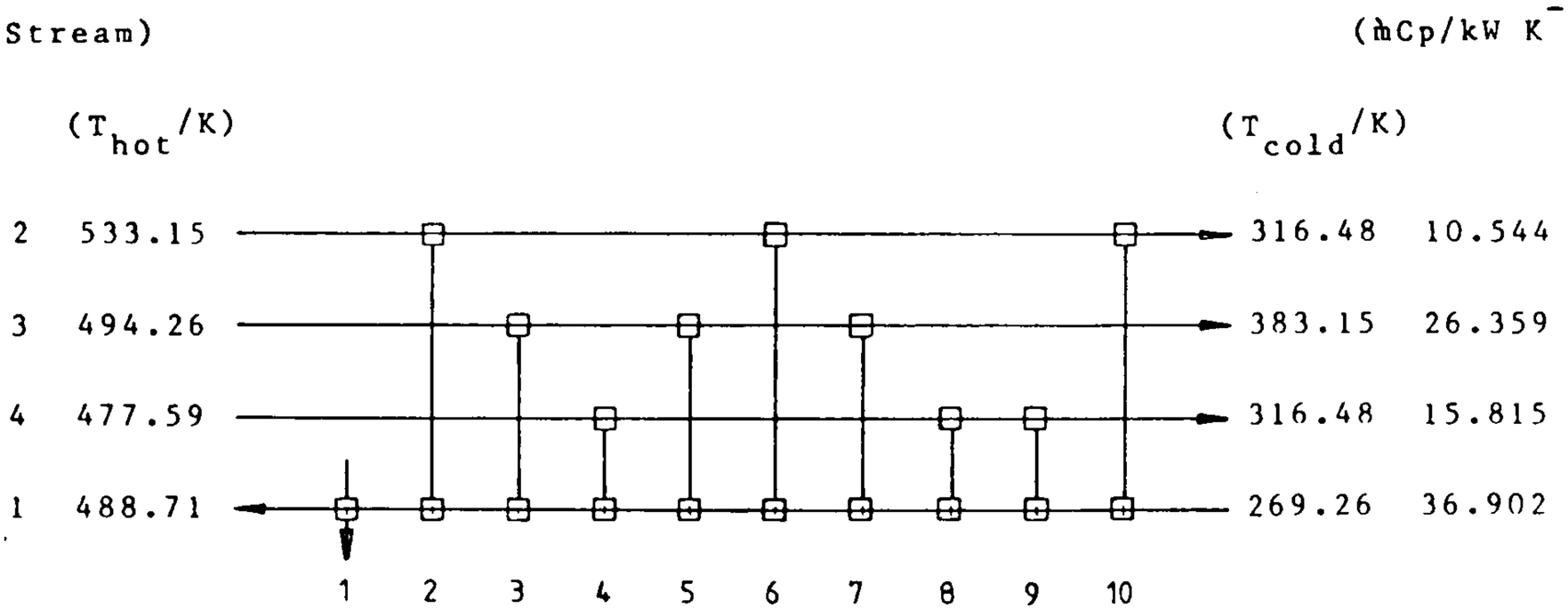


ΔT_{min} = 11.111K Cost Indication = 68.499

Heat exchanger loads/kW

1 : 336.9	2 : 564.6	3 : 790.4	4 : 549.5	5 : 956.9
6 : 817.7	7 : 1181.2	8 : 1502.3	9 : 902.3	10 : 496.3

Figure 7.5 The first ten-shell solution to 4SP2

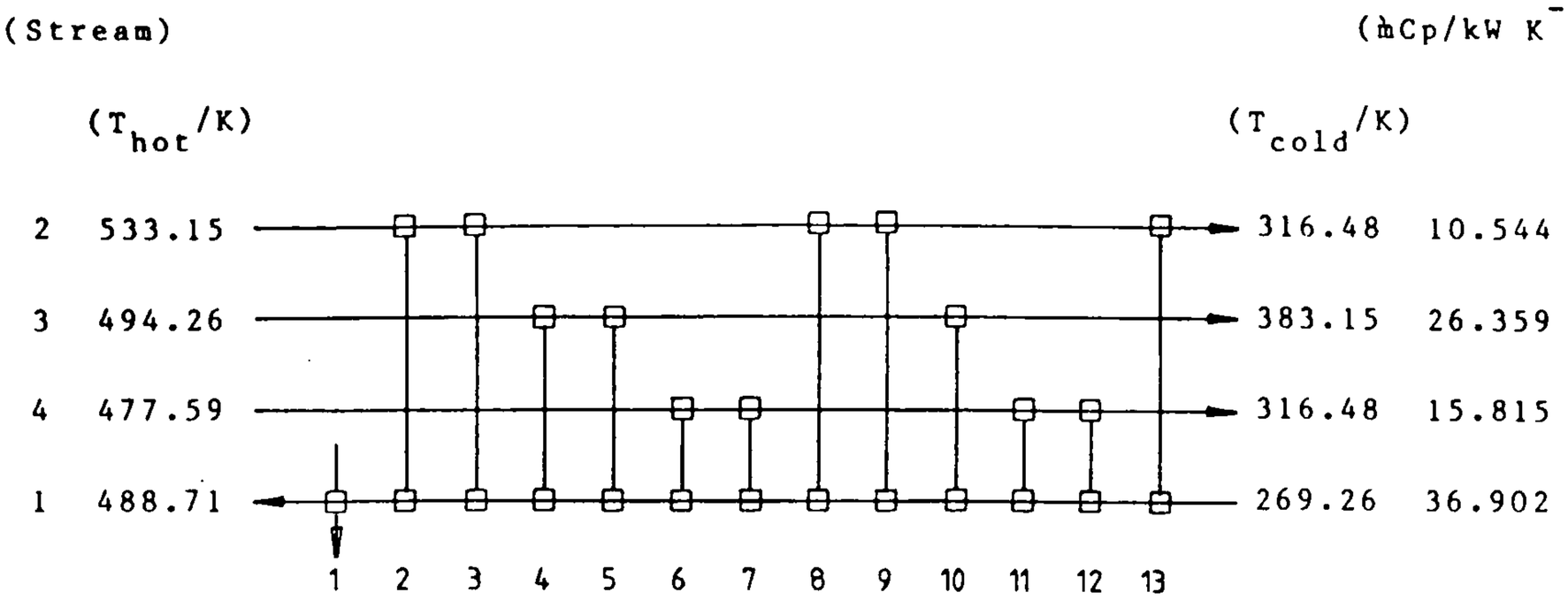


ΔT_{min} = 11.111K Cost Indication = 69.474

Heat exchanger loads/kW

1 : 336.9	2 : 564.6	3 : 790.4	4 : 549.5	5 : 956.9
6 : 817.7	7 : 1181.2	8 : 1502.3	9 : 496.3	10 : 902.3

Figure 7.6 The second ten-shell solution to 4SP2

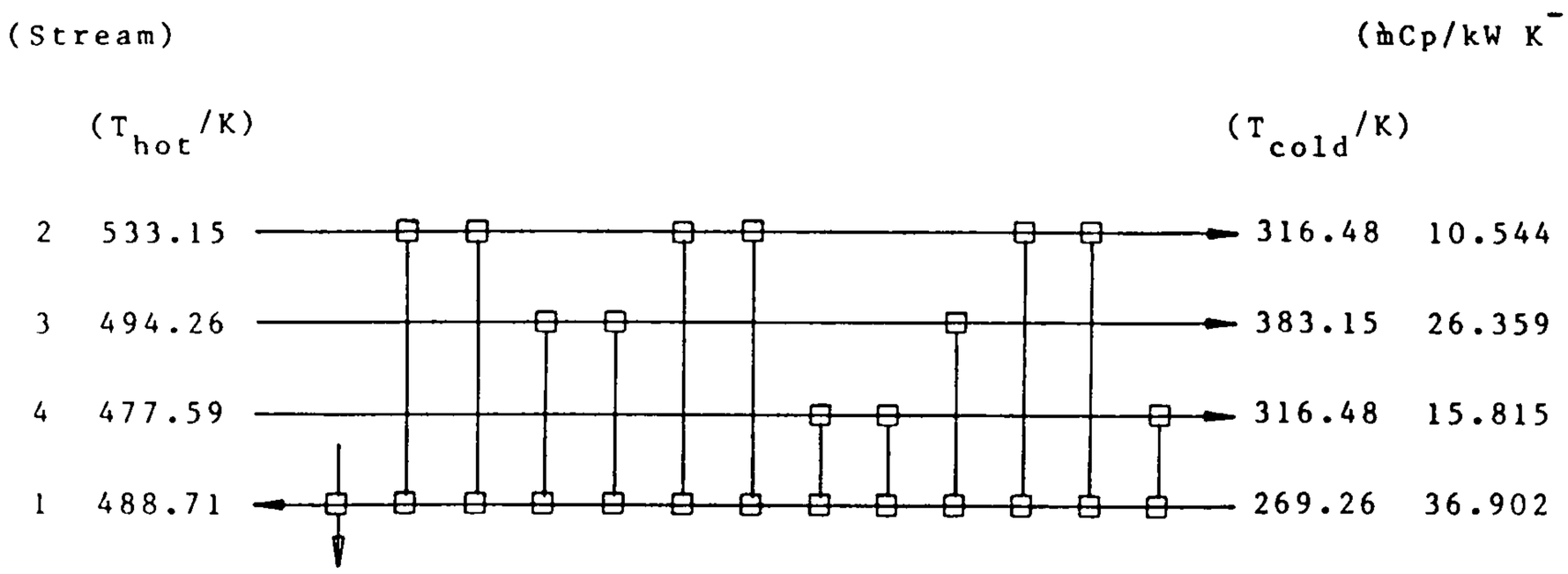


ΔT_{min} = 11.111K Cost Indication = 77.100

Heat exchanger loads/kW

1 : 336.9	2 : 564.6	3 : 62.0	4 : 834.7	5 : 596.0
6 : 850.6	7 : 329.9	8 : 863.5	9 : 181.2	10 : 1498.2
11 : 1265.4	12 : 101.8	13 : 613.3		

Figure 7.7 The first sample solution to 4SP2 with paired exchangers



ΔT_{min} = 11.111K Cost Indication = 76.835

Heat exchanger loads/kW

1 : 336.9	2 : 564.6	3 : 62.0	4 : 834.7	5 : 596.0
6 : 526.2	7 : 46.1	8 : 1095.6	9 : 469.8	10 : 1497.9
11 : 992.3	12 : 93.4	13 : 982.7		

Figure 7.8 The second sample solution to 4SP2 with paired exchangers

ΔT_{min} = 11.111K

Figure 7.9 The Zachoval and Konecny solution to 4SP2

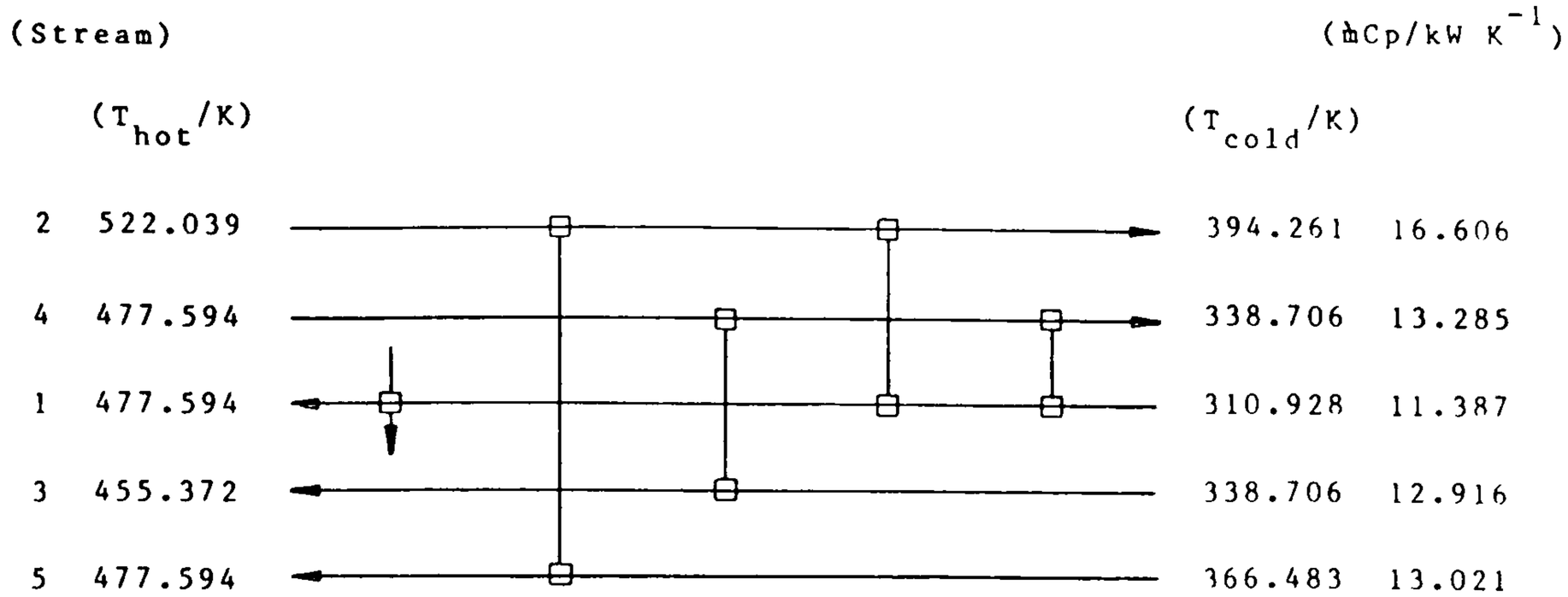
7.3 PROBLEM TWO

The second problem to be discussed here was introduced by Masso and Rudd (1969) and is known conventionally as 5SP1. A summary of the problem data is given in appendix M. The construction of the temperature interval table for the root node shows that the problem is unpinched and that only hot utilities are required.

The optimal solution when counter-current exchangers are permitted was first presented by Masso and Rudd (1969) and is reproduced here in figure 7.10. The same network, when constructed of 1/2 shell-and-tube heat exchangers, is shown in figure 7.11 and consists of twelve shells.

The new method isolates one hundred and thirty-one different networks composed of eleven shells with costs in the range 49.664 to 54.057. Examples are given in figures 7.12 to 7.17.

A casual inspection of these networks might lead to the conclusion that, although one less shell is required, there has been a loss of 'modularity' (i.e. most of the exchangers are of different sizes) and an increase in the complexity of the pipework required. However, contained within the set of solutions are fourteen networks that contain three pairs of adjacent exchangers matching the same streams, six networks that contain an adjacent pair and an adjacent triplet of exchangers matching the same streams and thirty that contain two pairs of adjacent exchangers matching the same streams. Examples of the networks exhibiting three pairs of adjacent exchangers matching the same streams are given in figures 7.18, 7.19 and 7.20.

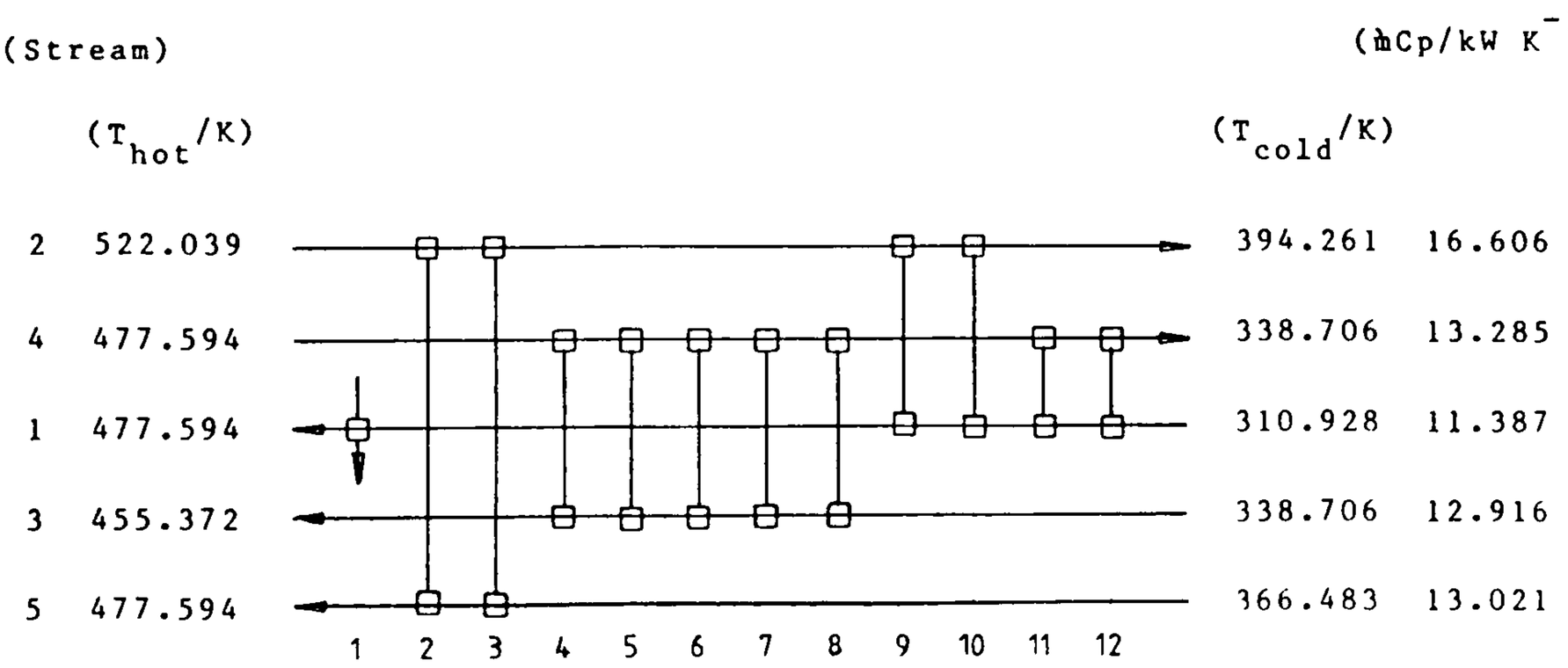


ΔT_{min} = 11.111K

Heat exchanger loads/kW

1 : 884.5 2 : 1446.8 3 : 1506.9 4 : 675.1 5 : 338.3

Figure 7.10 The optimal solution to 5SP1 using counter-current exchangers



ΔT_{min} = 11.111K Cost Indication = 54.138

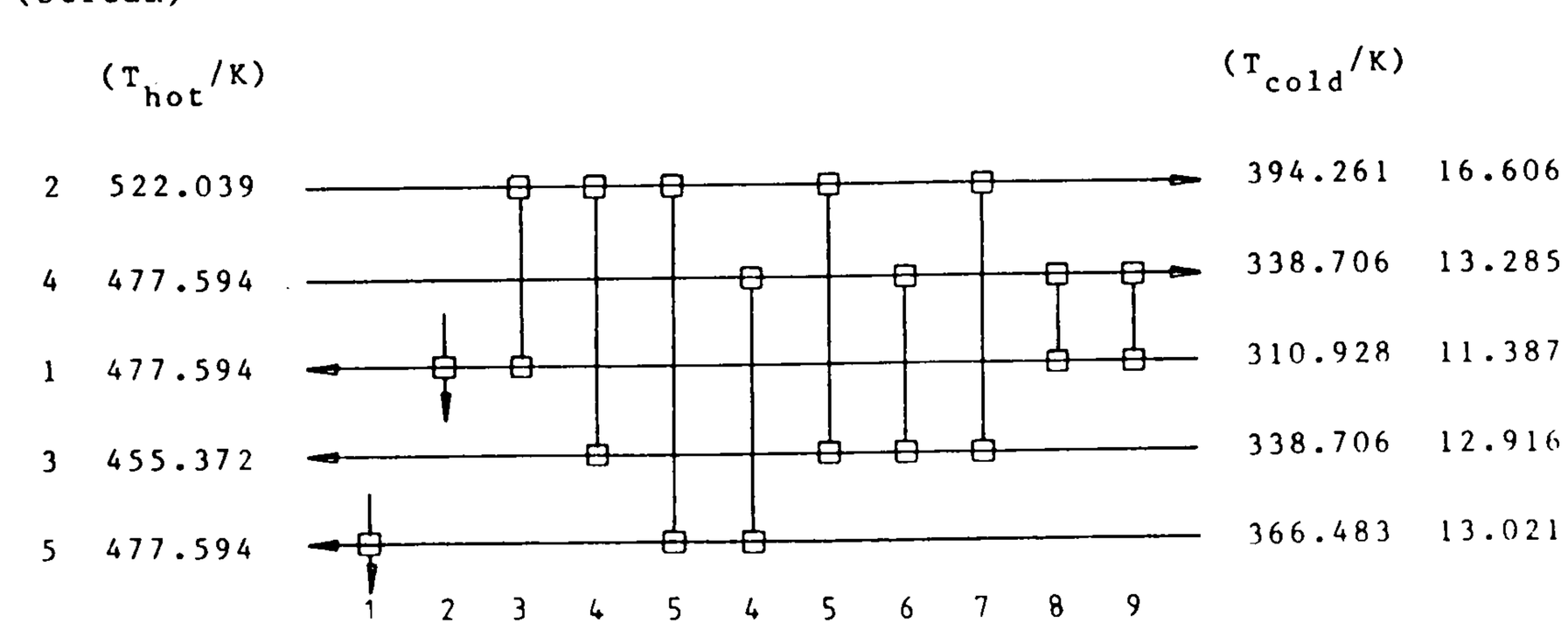
Heat exchanger loads/kW

1 : 884.5 2 : 555.7 3 : 891.1 4 : 245.2 5 : 302.2

6 : 310.9 7 : 319.7 8 : 328.9 9 : 64.5 10 : 610.6

11 : 22.0 12 : 316.3

Figure 7.11 The solution given in figure 7.10 using 1/2 exchangers



ΔT_{min} = 11.111K Cost Indication = 53.139

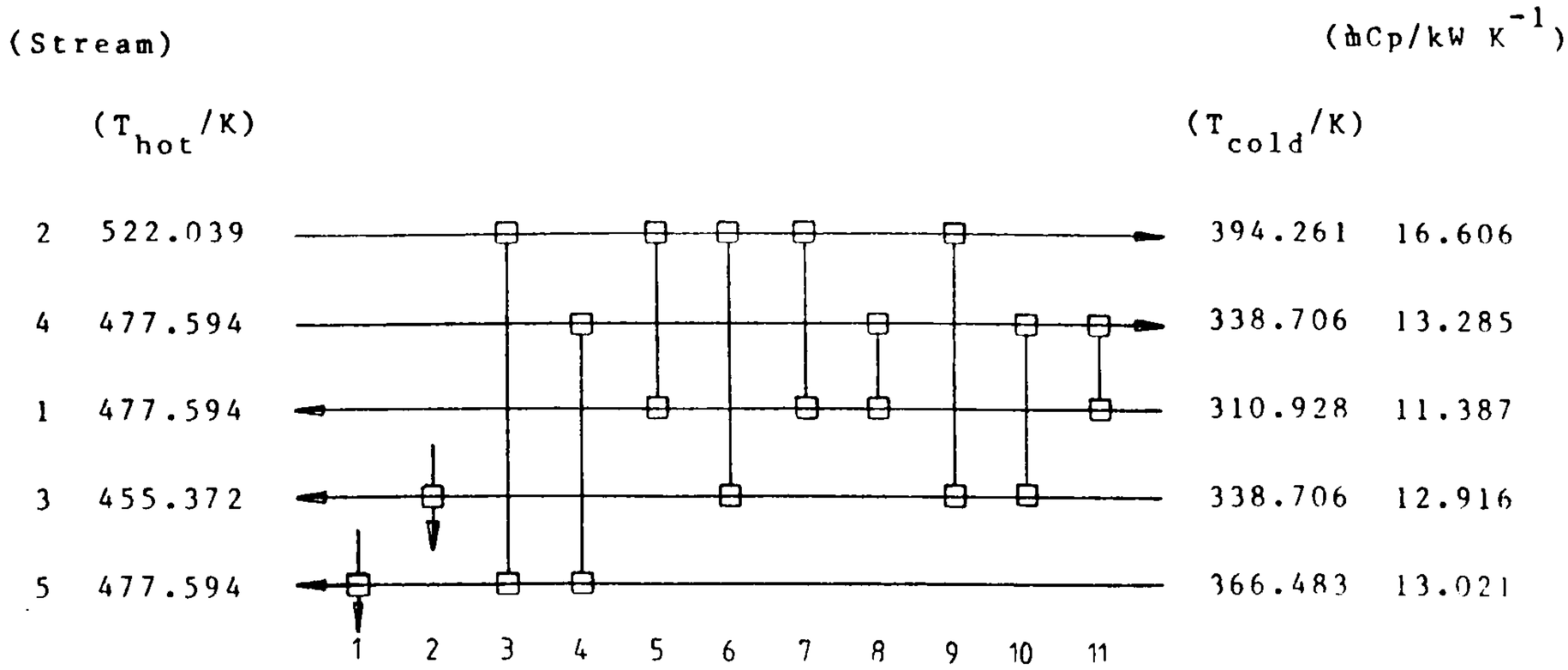
Heat exchanger loads/kW

1 : 446.3 2 : 418.1 3 : 530.9 4 : 508.1 5 : 293.9

6 : 686.6 7 : 427.6 8 : 571.1 9 : 361.4 10 : 271.1

11 : 316.4

Figure 7.12 The first eleven-shell sample solution to 5SP1

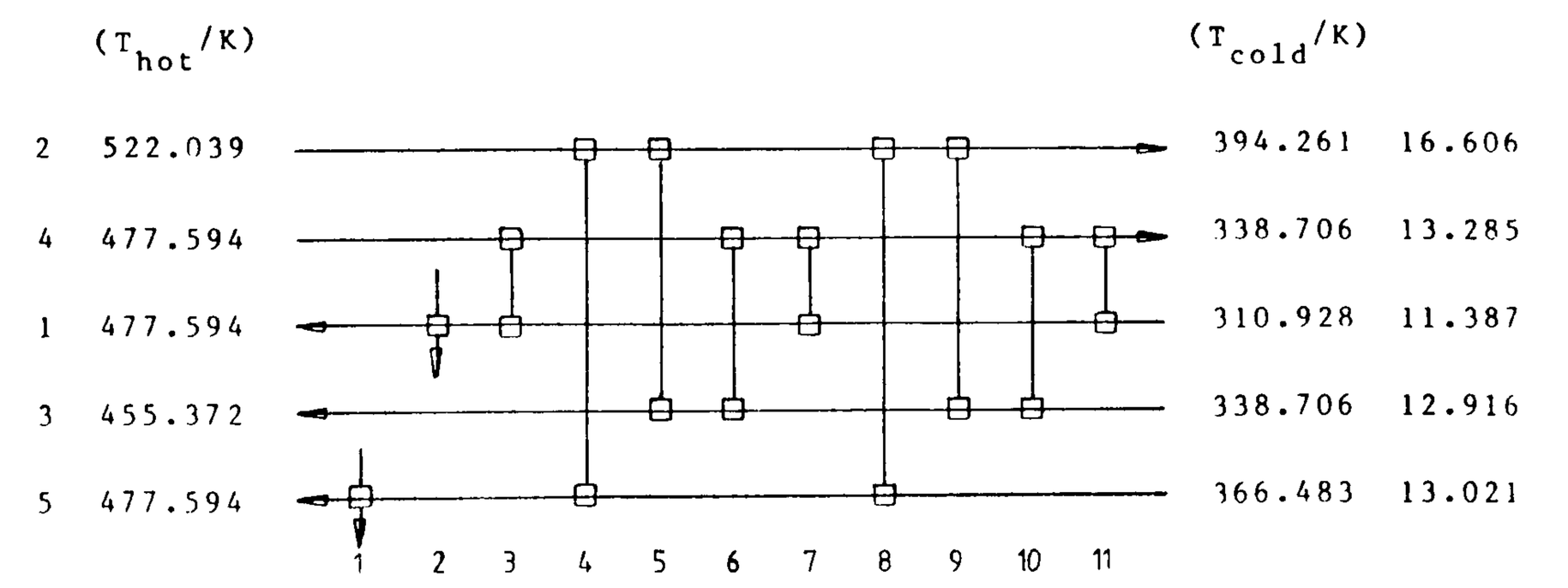


ΔT_{min} = 11.111K Cost Indication = 49.933

Heat exchanger loads/kW

1 : 708.9	2 : 175.6	3 : 51.3	4 : 686.6	5 : 667.8
6 : 613.8	7 : 379.1	8 : 534.6	9 : 410.0	10 : 307.6
11 : 316.4				

Figure 7.13 The second eleven-shell sample solution to 5SP1

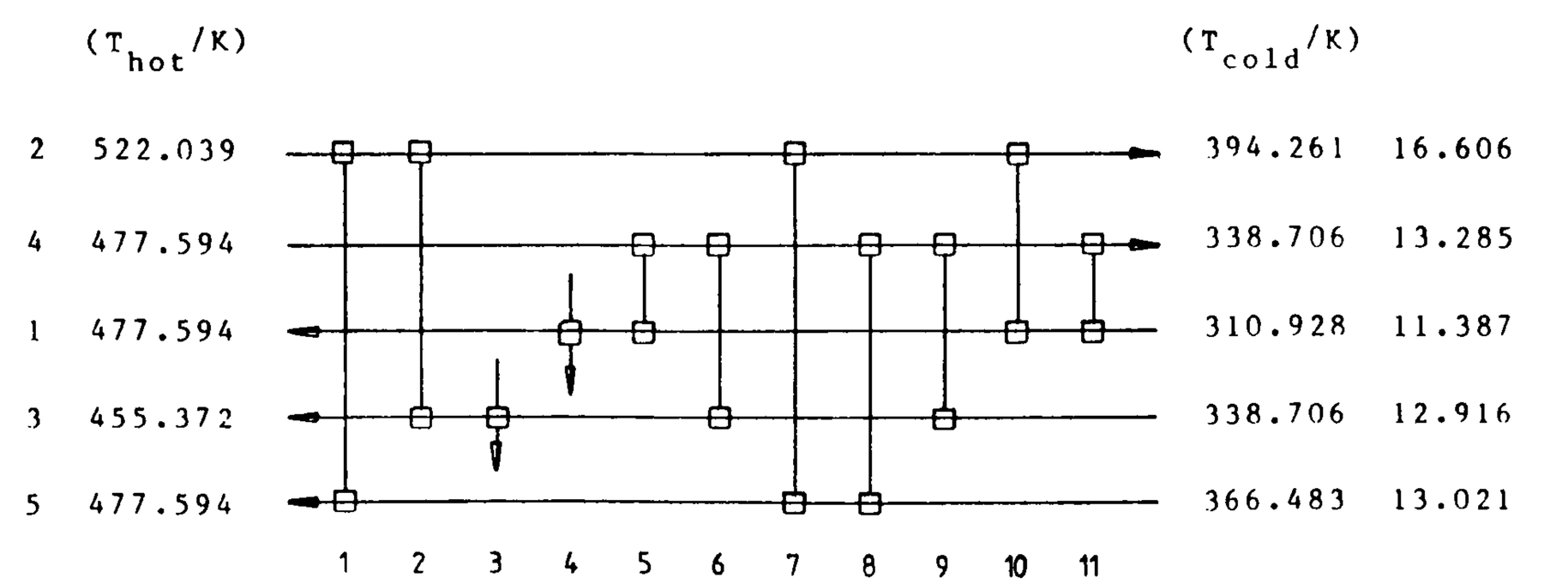


ΔT_{min} = 11.111K Cost Indication = 50.864

Heat exchanger loads/kW

1 : 115.4	2 : 769.0	3 : 277.8	4 : 648.2	5 : 380.5
6 : 408.8	7 : 534.6	8 : 683.2	9 : 410.0	10 : 307.6
11 : 316.4				

Figure 7.14 The third eleven-shell sample solution to 5SP1

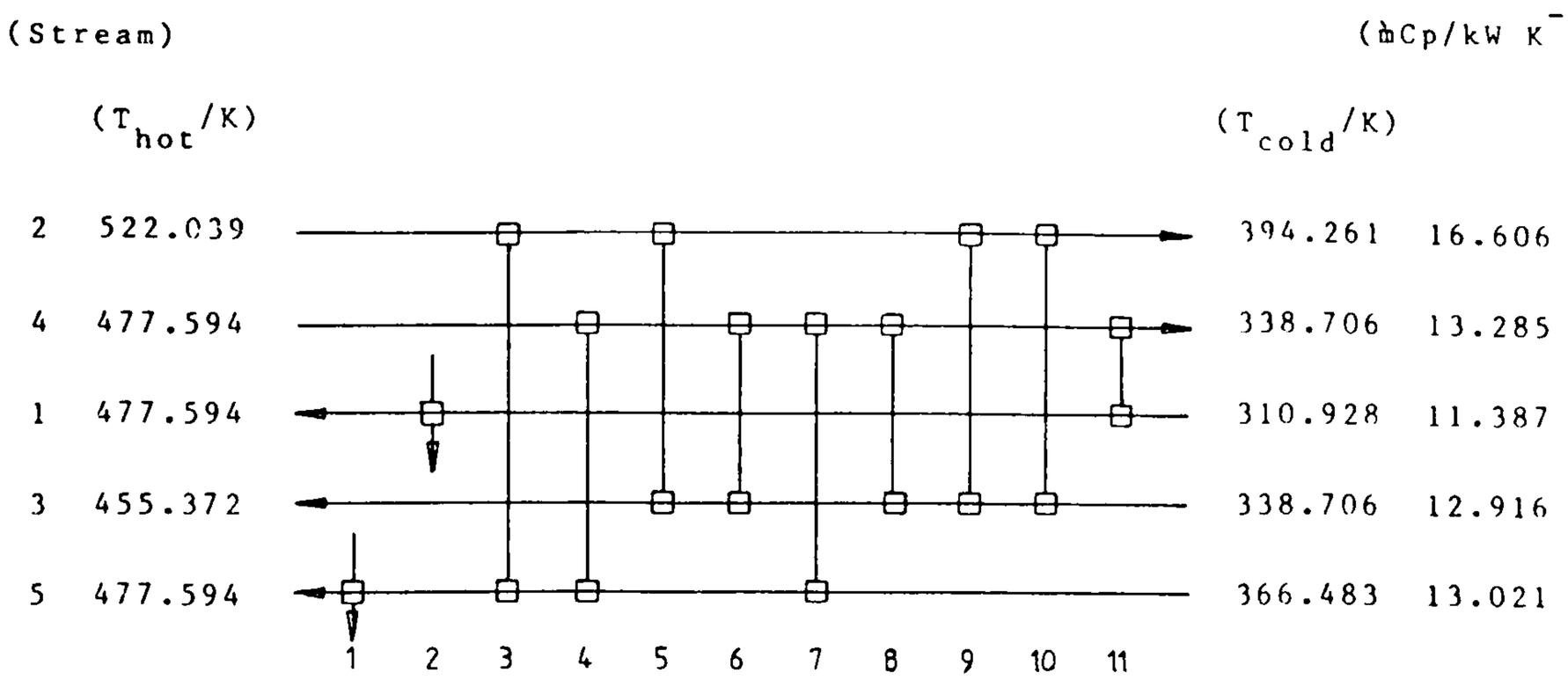


ΔT_{min} = 11.111K Cost Indication = 51.283

Heat exchanger loads/kW

1 : 589.0	2 : 292.2	3 : 365.4	4 : 519.2	5 : 429.7
6 : 541.7	7 : 608.1	8 : 249.7	9 : 307.6	10 : 632.5
11 : 316.4				

Figure 7.15 The fourth eleven-shell sample solution to 5SP1

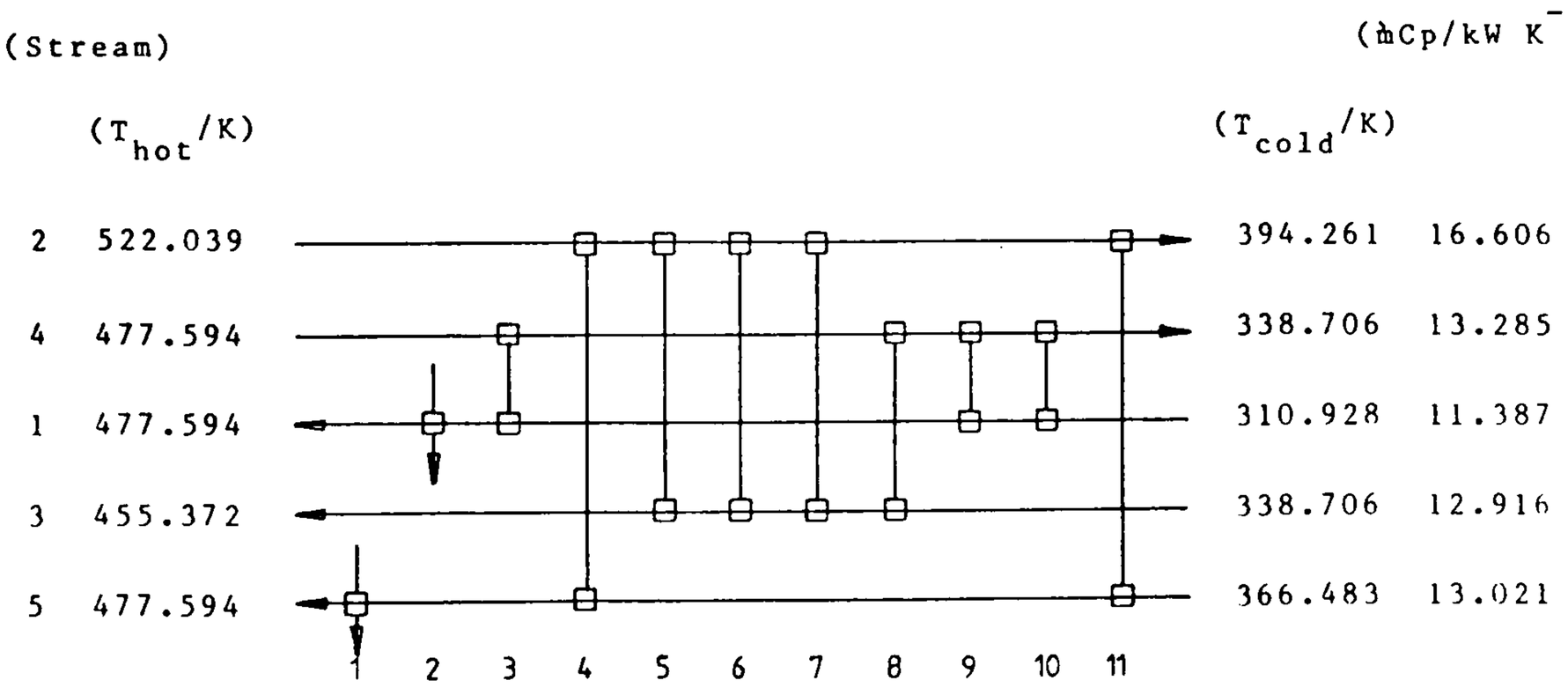


ΔT_{min} = 11.111K Cost Indication = 51.236

Heat exchanger loads/kW

1 : 369.3	2 : 515.1	3 : 397.9	4 : 429.8	5 : 657.6
6 : 541.7	7 : 249.7	8 : 307.6	9 : 433.8	10 : 632.5
11 : 316.4				

Figure 7.16 The fifth eleven-shell sample solution to 5SP1

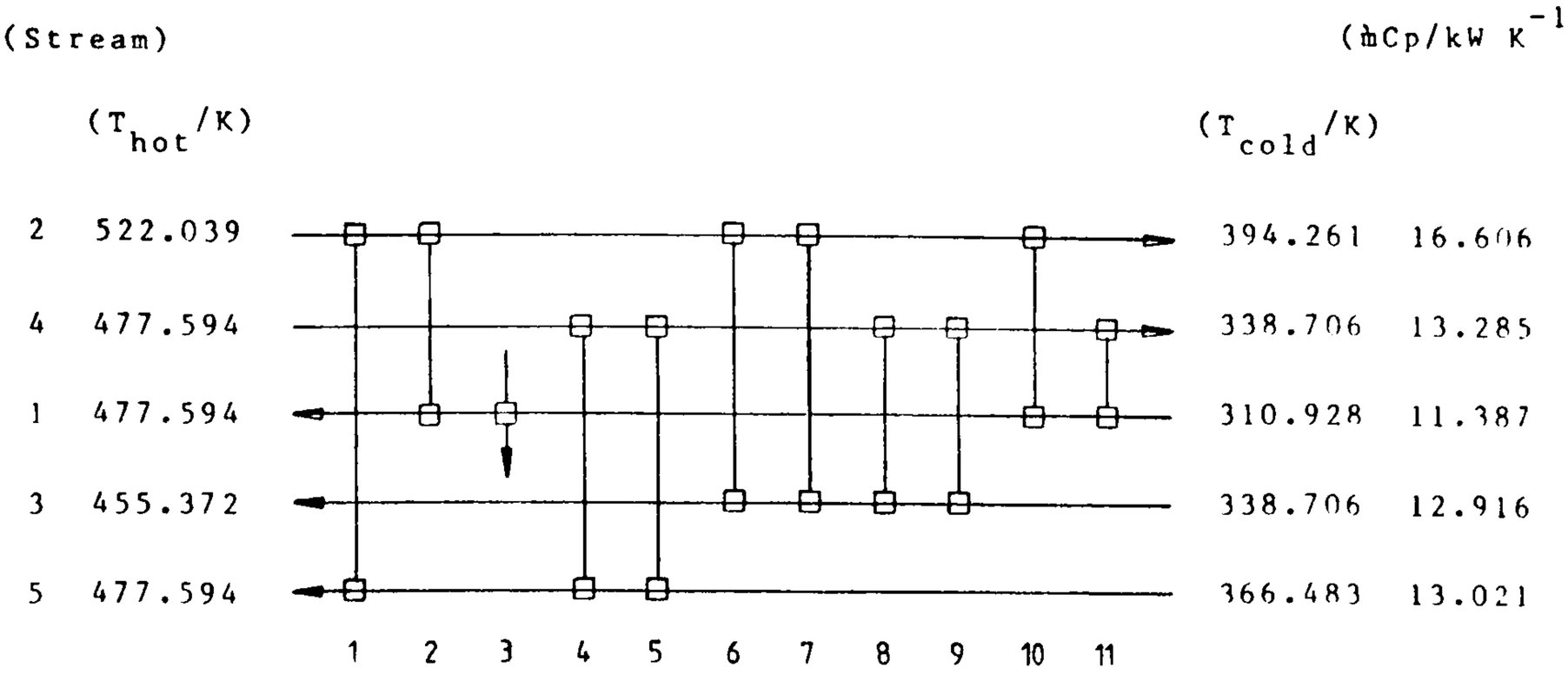


ΔT_{min} = 11.111K Cost Indication = 53.746

Heat exchanger loads/kW

1 : 260.6	2 : 623.8	3 : 686.5	4 : 824.5	5 : 175.3
6 : 332.7	7 : 427.8	8 : 571.1	9 : 271.1	10 : 316.4
11 : 361.7				

Figure 7.17 The sixth eleven-shell sample solution to 5SP1

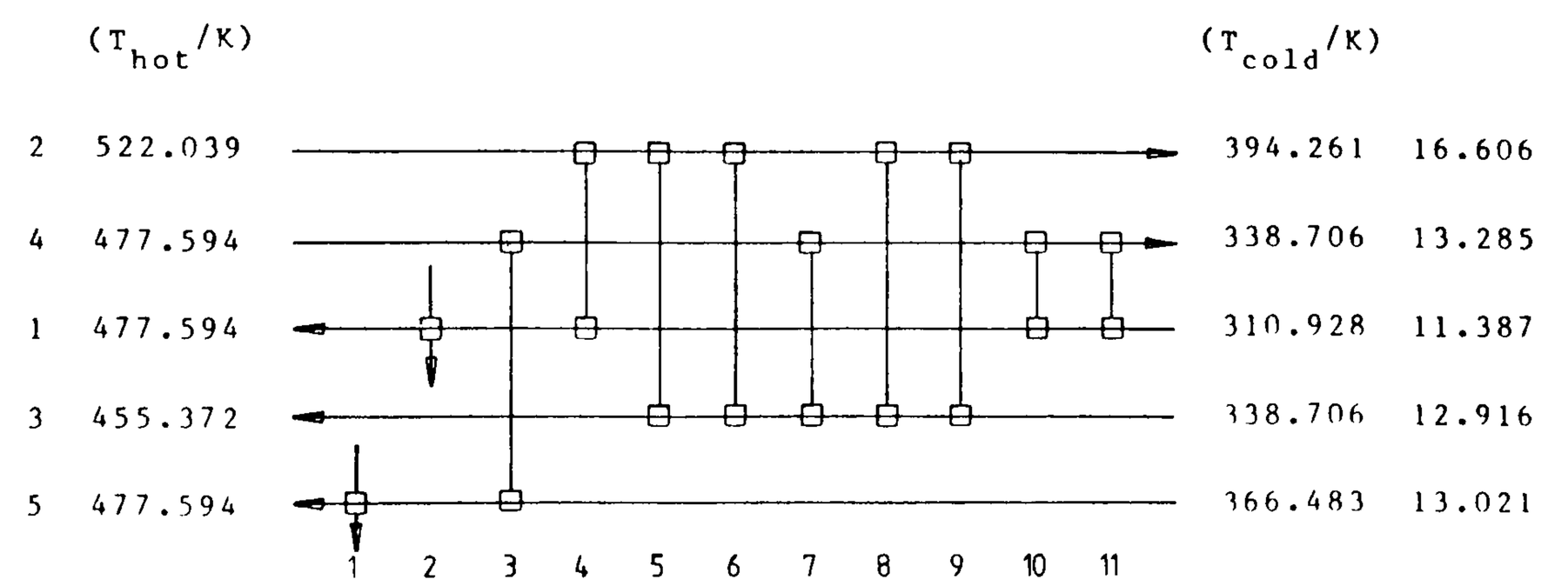


ΔT_{min} = 11.111K Cost Indication = 51.877

Heat exchanger loads/kW

1 : 524.4	2 : 64.5	3 : 884.4	4 : 379.6	5 : 542.8
6 : 279.2	7 : 603.2	8 : 298.9	9 : 307.6	10 : 632.5
11 : 316.4				

Figure 7.18 The seventh eleven-shell sample solution to 5SP1

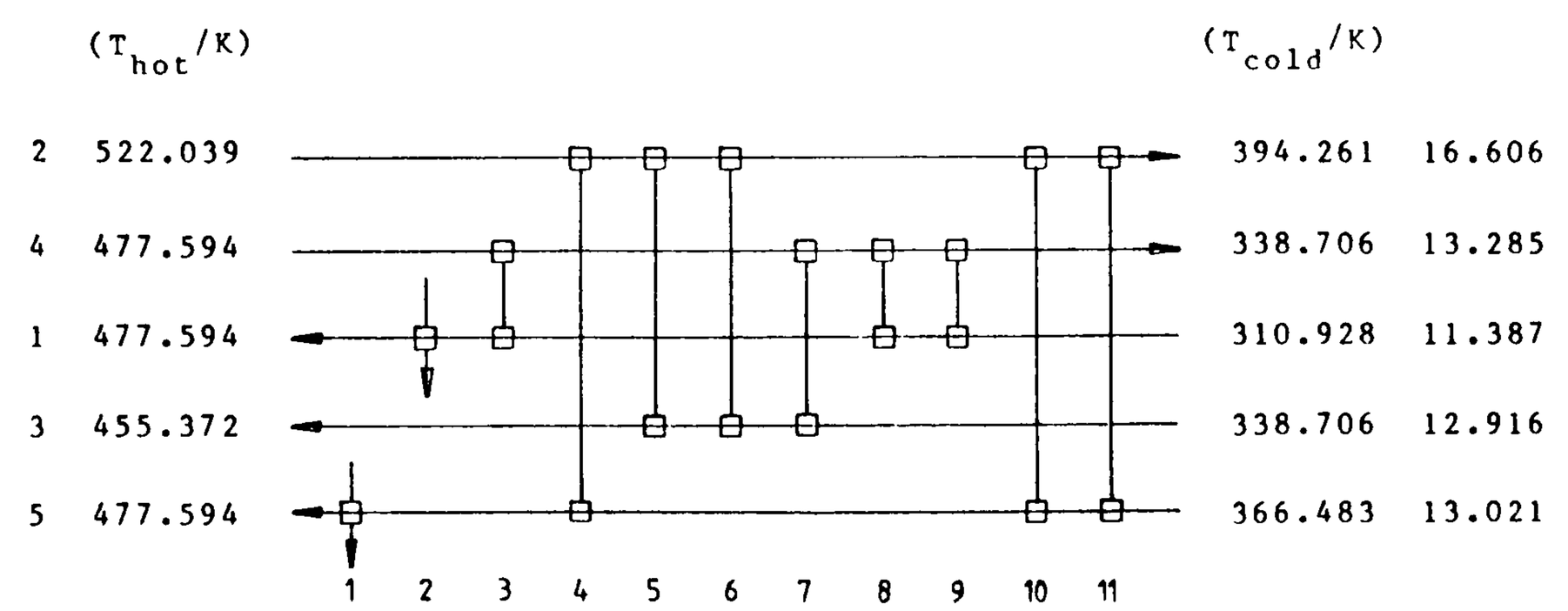


ΔT_{min} = 11.111K Cost Indication = 51.885

Heat exchanger loads/kW

1 : 760.2	2 : 124.3	3 : 686.6	4 : 576.8	5 : 315.3
6 : 620.5	7 : 571.1	8 : 247.9	9 : 361.4	10 : 271.1
11 : 316.4				

Figure 7.19 The eighth eleven-shell sample solution to 5SP1



ΔT_{min} = 11.111K Cost Indication = 52.148

Heat exchanger loads/kW

1 : 260.6	2 : 623.8	3 : 686.5	4 : 540.9	5 : 287.4
6 : 648.4	7 : 571.1	8 : 271.1	9 : 316.4	10 : 283.6
11 : 361.7				

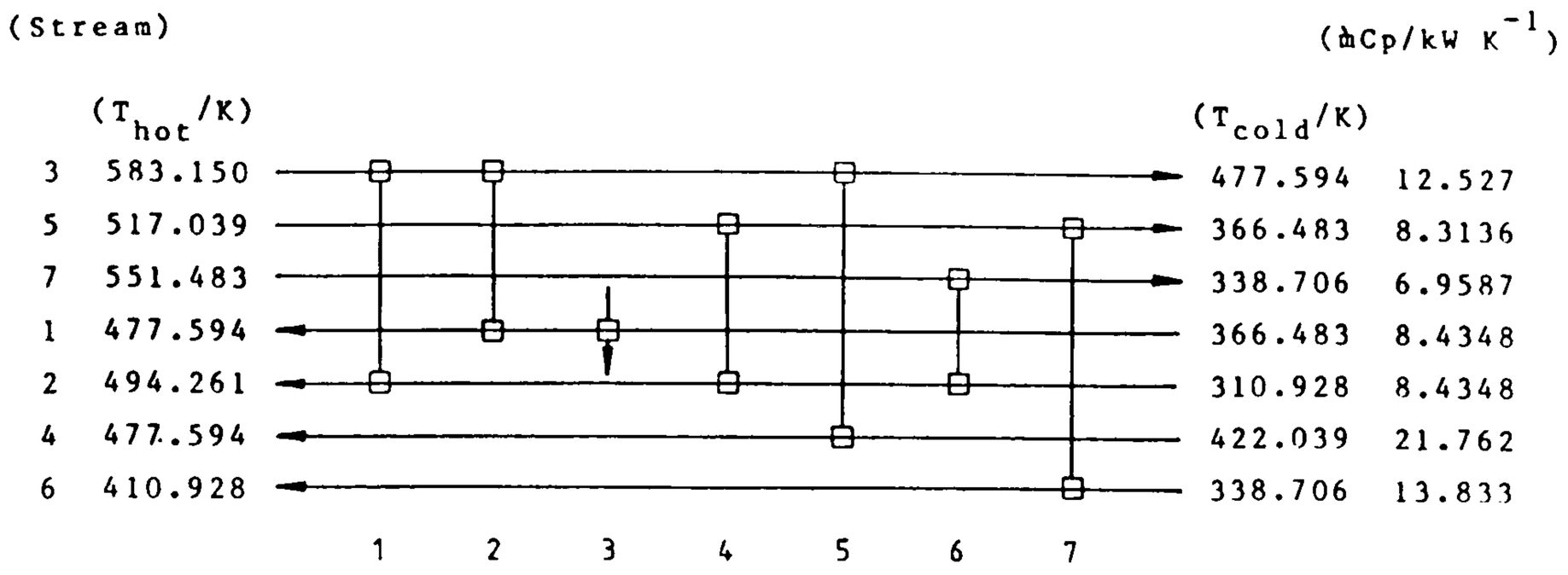
Figure 7.20 The ninth eleven-shell sample solution to 5SP1

7.4 PROBLEM THREE

The third problem to be discussed here was introduced by Masso and Rudd (1969) and is known conventionally as 7SP2. A summary of the relevant data is given in appendix M. The construction of the temperature interval table for the root node shows that the problem is unpinched and that only hot utilities are required.

The optimal solution when counter-current exchangers are permitted was first presented by Linnhoff and Flower (1978a) and is reproduced in figure 7.21. The same network, when constructed of 1/2 shell-and-tube heat exchangers, is shown in figure 7.22 and contains twelve shells.

The new method quickly isolates a solution with eleven shells, shown in figure 7.23, and proceeds to find many more with cost indicators ranging from 39.742 to 44.423. An upper limit of one hundred was placed on the number of solutions to be obtained and this constraint was quickly reached. A typical selection of these networks is given in figures 7.24 to 7.28. Obviously the problem is fairly easy to solve and the doubling, if not trebling, of shells at each match worth attempting (see section 6.3.1). Further applications of HERNE showed this to be the case. The lowest cost solutions are given in figures 7.29 and 7.30 for pairing and tripling respectively. These new solutions are marginally cheaper than the solutions found before only because the constraint of doubling or trebling exchangers in each match cuts out many branches of the search tree. Thus the tree search investigates other sections of the tree, where slightly cheaper solutions exist, which are not reached in the original search because of the enforcement of the upper bound in the number of solutions to be found.

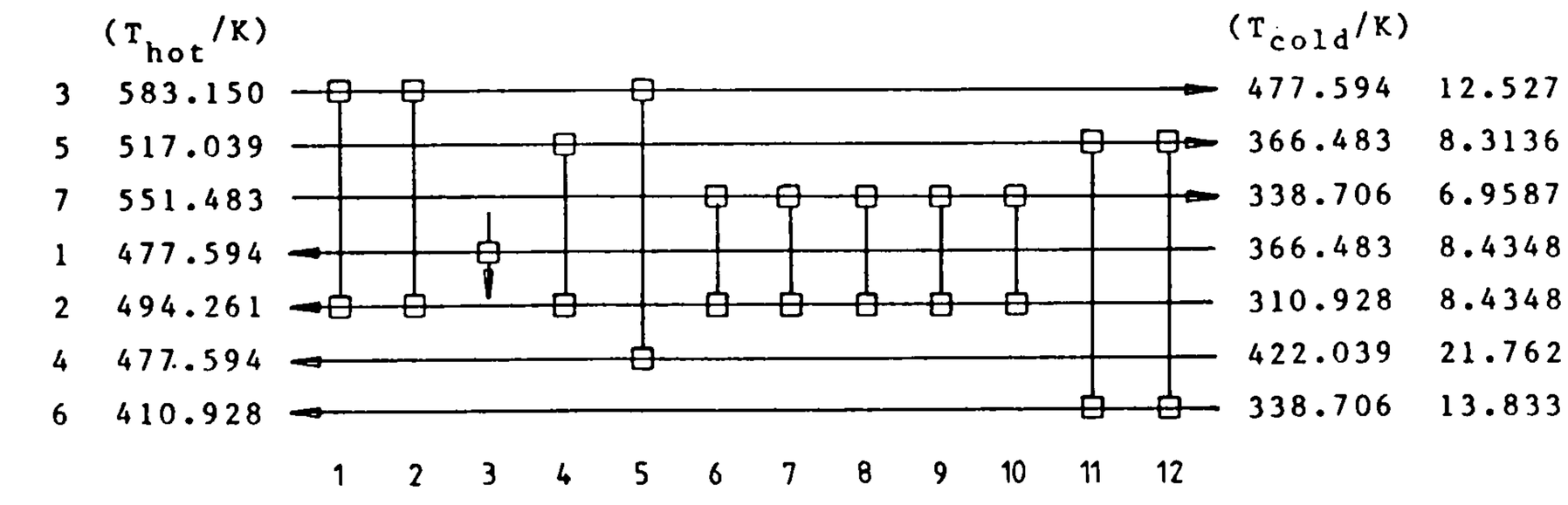


ΔT_{min}=11.111K

Heat exchanger loads/kW

1 : 65.7	2 : 47.6	3 : 637.0	4 : 252.6	5 : 1209.0
6 : 1480.7	7 : 999.0			

Figure 7.21 The optimal solution to 7SP2 using counter-current exchangers

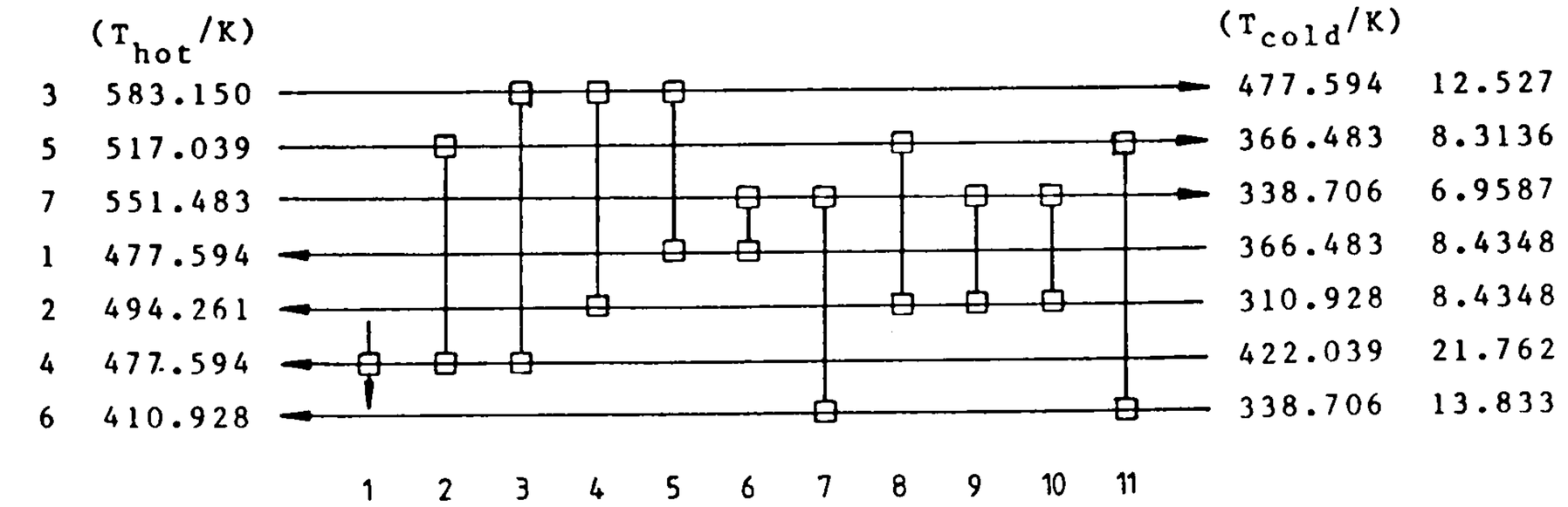


ΔT_{min}=11.111K Cost Indication = 39.747

Heat exchanger loads/kW

1 : 65.7	2 : 47.6	3 : 637.0	4 : 252.6	5 : 1209.0
6 : 200.8	7 : 417.3	8 : 344.3	9 : 284.0	10 : 234.3
11 : 614.8	12 : 384.2			

Figure 7.22 The solution given in figure 7.21 using 1/2 exchangers

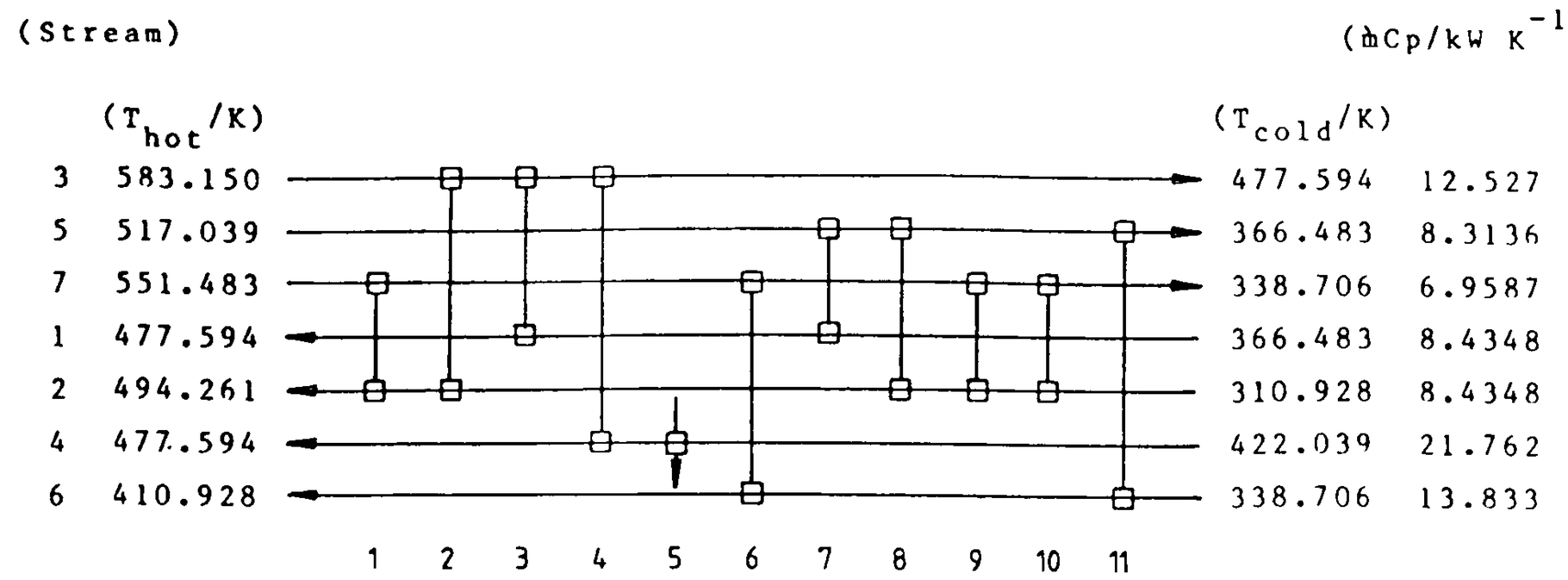


ΔT_{min}=11.111K Cost Indication = 40.666

Heat exchanger loads/kW

1 : 637.0	2 : 527.3	3 : 44.6	4 : 688.0	5 : 589.6
6 : 347.6	6 : 614.8	7 : 340.1	8 : 284.0	10 : 234.3
11 : 384.2				

Figure 7.23 The first eleven-shell sample solution to 7SP2

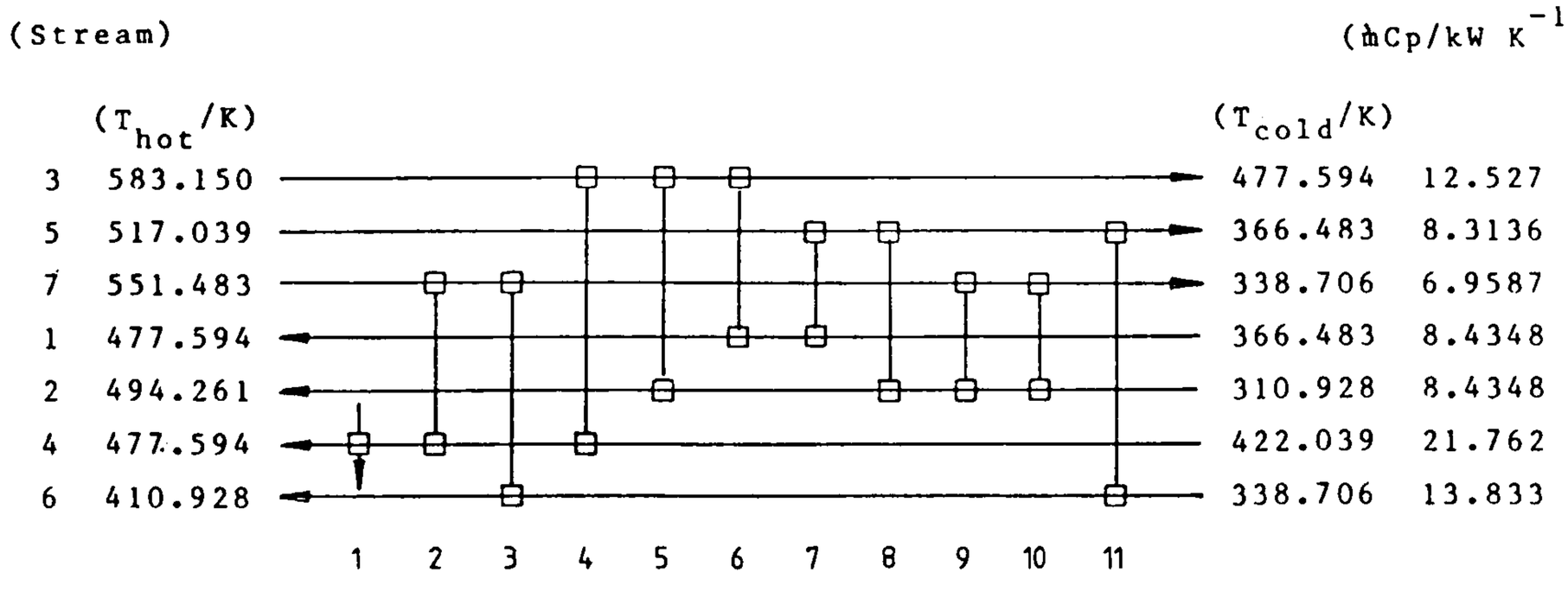


ΔT_{min}=11.111K Cost Indication = 41.466

Heat exchanger loads/kW

1 : 347.5	2 : 340.4	3 : 409.9	4 : 572.0	5 : 637.0
6 : 614.8	7 : 527.3	8 : 340.1	9 : 284.0	10 : 234.3
11 : 384.2				

Figure 7.24 The second eleven-shell sample solution to 7SP2

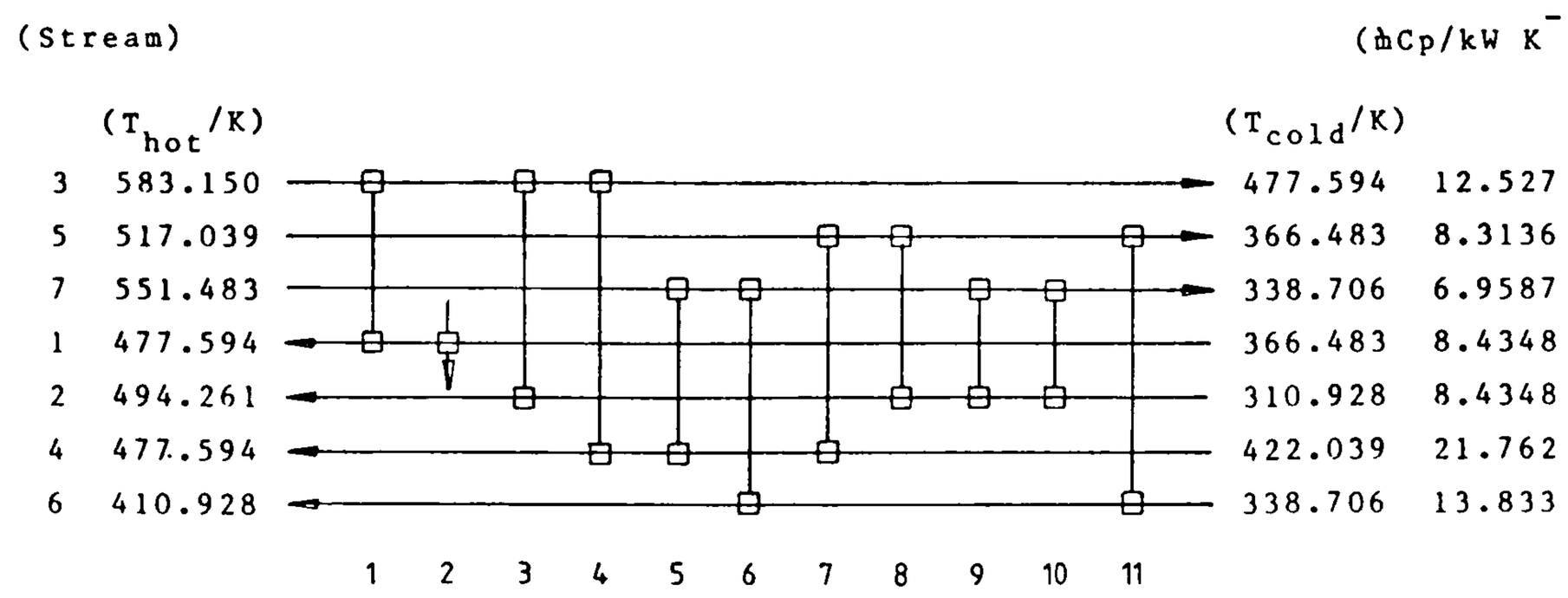


ΔT_{min}=11.111K Cost Indication = 40.932

Heat exchanger loads/kW

1 : 637.0	2 : 347.5	3 : 614.8	4 : 224.4	5 : 688.0
6 : 409.9	7 : 527.3	8 : 340.1	9 : 284.0	10 : 234.4
11 : 384.2				

Figure 7.25 The third eleven-shell sample solution to 7SP2

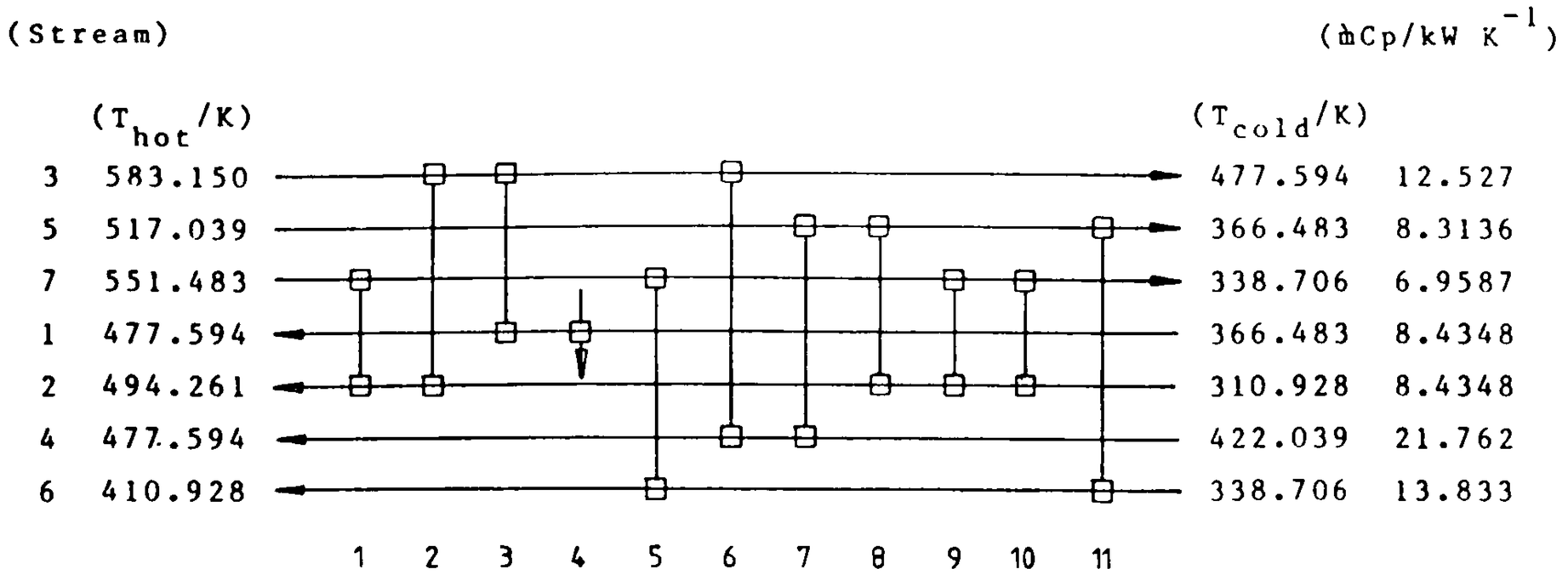


ΔT_{min}=11.111K Cost Indication = 42.985

Heat exchanger loads/kW

1 : 300.2	2 : 637.0	3 : 688.0	4 : 334.1	5 : 347.5
6 : 614.8	7 : 527.3	8 : 340.1	9 : 284.0	10 : 234.3
11 : 384.2				

Figure 7.26 The fourth eleven-shell sample solution to 7SP2

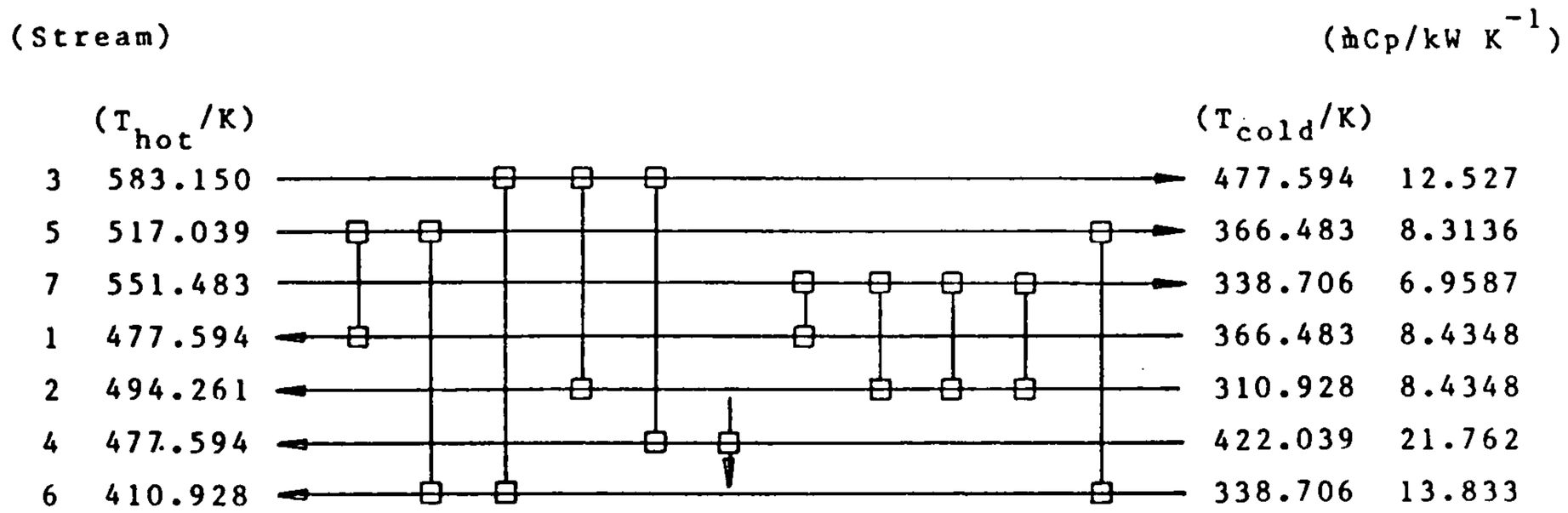


ΔT_{min}=11.111K Cost Indication = 41.763

Heat exchanger loads/kW

1 : 347.5	2 : 340.4	3 : 300.2	4 : 637.0	5 : 614.8
6 : 681.7	7 : 527.3	8 : 340.1	9 : 284.0	10 : 234.3
11 : 384.2				

Figure 7.27 The fifth eleven-shell sample solution to 7SP2

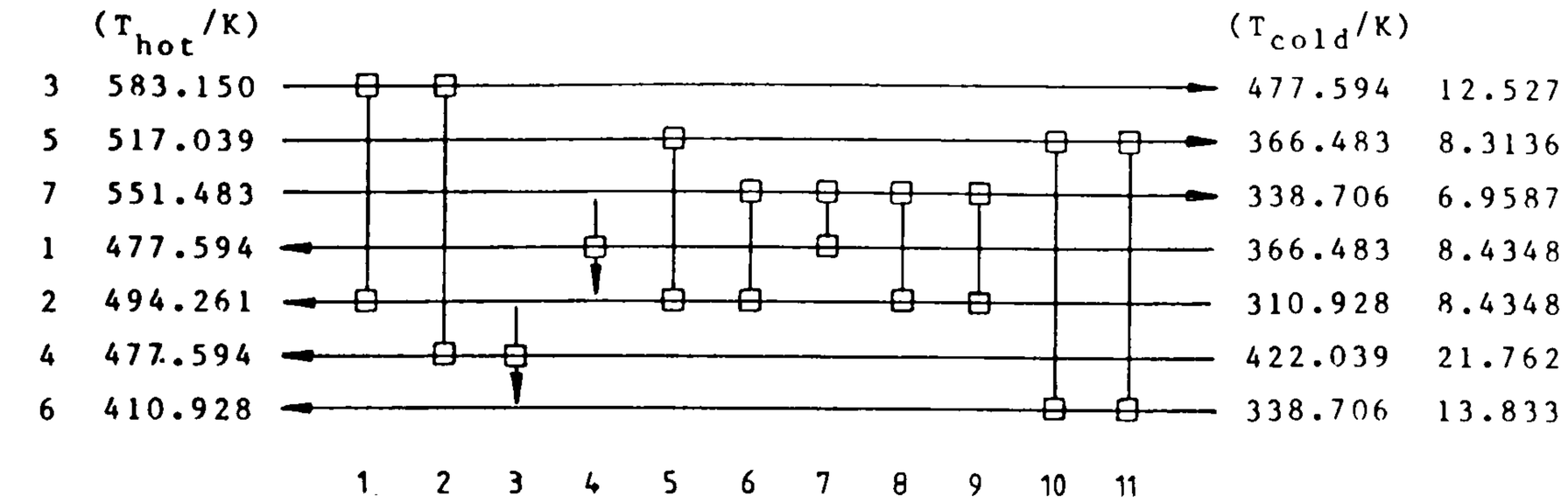


ΔT_{min}=11.111K Cost Indication = 41.699

Heat exchanger loads/kW

1 : 319.1	2 : 548.3	3 : 66.5	4 : 683.8	5 : 572.0
6 : 637.0	7 : 618.1	8 : 344.2	9 : 284.0	10 : 234.3
11 : 384.2				

Figure 7.28 The sixth eleven-shell sample solution to 7SP2

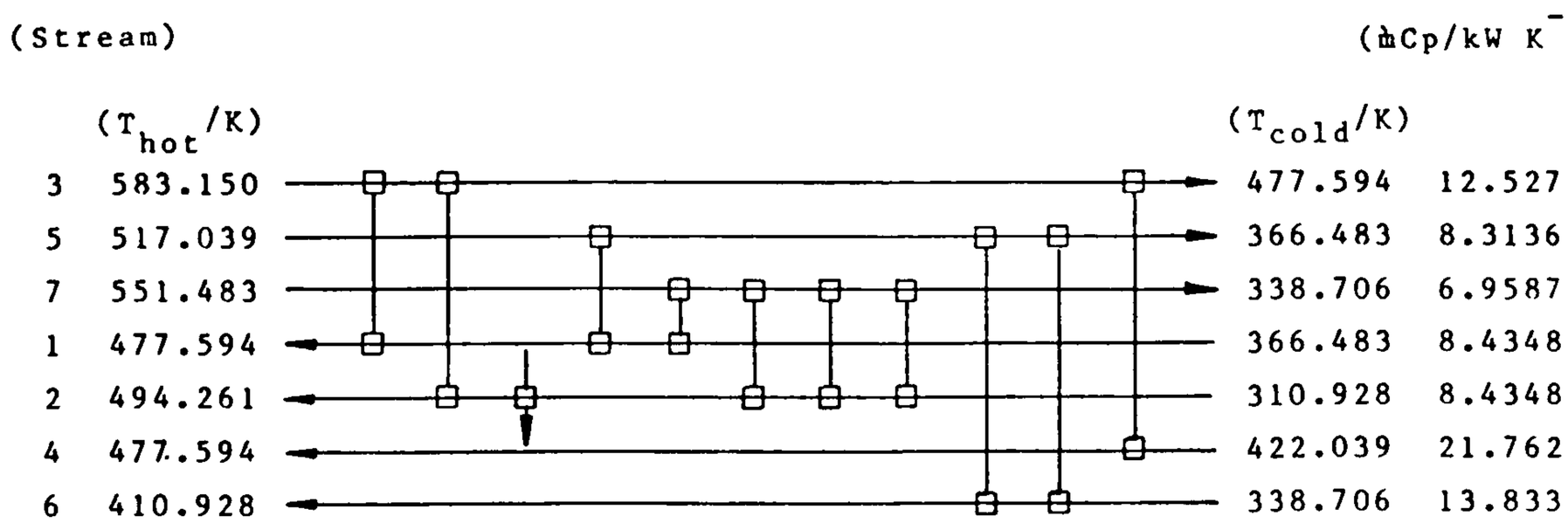


ΔT_{min}=11.111K Cost Indication = 39.283

Heat exchanger loads/kW

1 : 207.1	2 : 1115.2	3 : 93.8	4 : 543.2	5 : 252.6
6 : 568.3	7 : 394.0	8 : 284.0	9 : 234.3	10 : 614.8
11 : 384.2				

Figure 7.29 The seventh eleven-shell sample solution to 7SP2



ΔT_{min}=11.111K Cost Indication = 39.098

Heat exchanger loads/kW

1 : 66.5	2 : 46.8	3 : 637.0	4 : 252.6	5 : 618.1
6 : 344.2	7 : 284.0	8 : 234.3	9 : 614.8	10 : 384.2
11 : 1209.0				

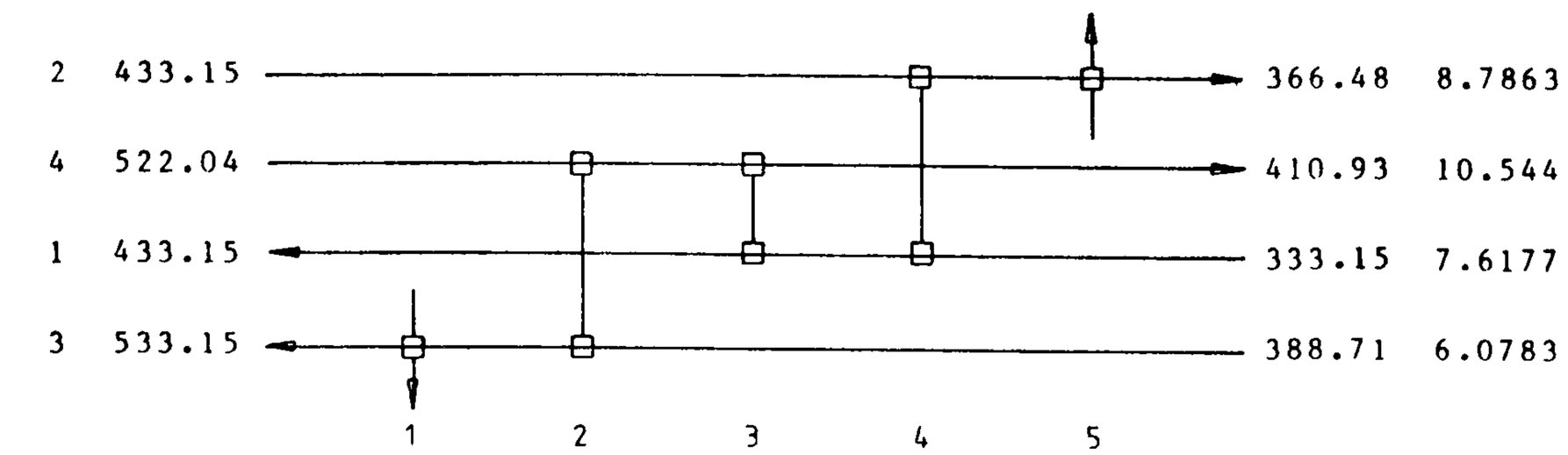
Figure 7.30 The eighth eleven-shell sample solution to 7SP2

7.5 PROBLEM FOUR

The fourth problem to be discussed here was introduced by Lee et al (1970) and is known conventionally as 4SP1. A summary of the relevant data is given in appendix M. The construction of the temperature interval table for the root node shows that the problem is pinched and so both hot and cold utilities are required.

The optimal solution when counter-current exchangers are permitted was first presented by Hohmann (1971) and is reproduced in figure 7.31. The same network when constructed of 1/2 shell-and-tube heat exchangers is shown in figure 7.32 and contains nine shells.

The new method isolates this solution plus two new networks composed of nine shells, shown in figures 7.33 and 7.34. These solutions are of very similar cost and so the choice of the final network would have to be made by other means (e.g. on the grounds of flexibility, controllability or switchability). The solution in figure 7.33 is very similar to that in figure 7.32 since it contains the same exchangers in a different order. However, the solution in figure 7.34 is different since a match between streams three and two is present. The usefulness, or otherwise, of this match would become apparent in further analysis of the problem.

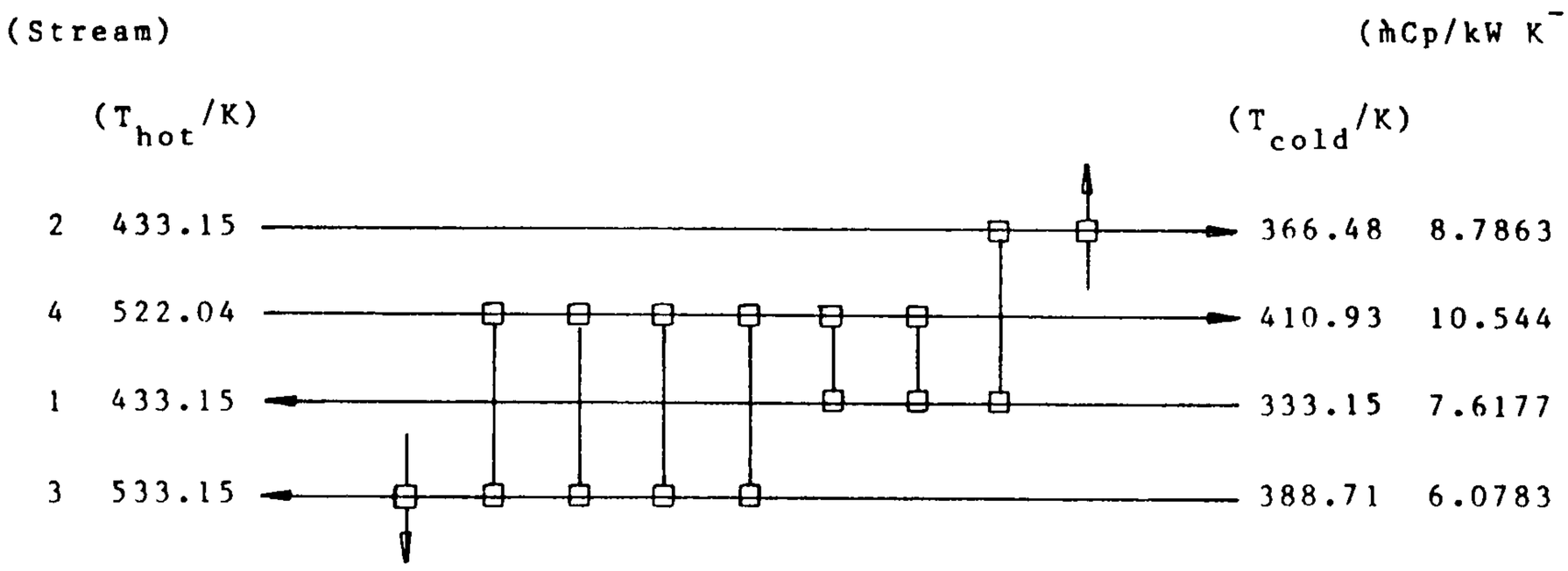


ΔT_{min} = 11.111K

Heat exchanger loads/kW

1 : 135.1 2 : 742.9 3 : 428.6 4 : 330.7 5 : 255.1

Figure 7.31 The optimal solution to 4SP1 using counter-current exchangers



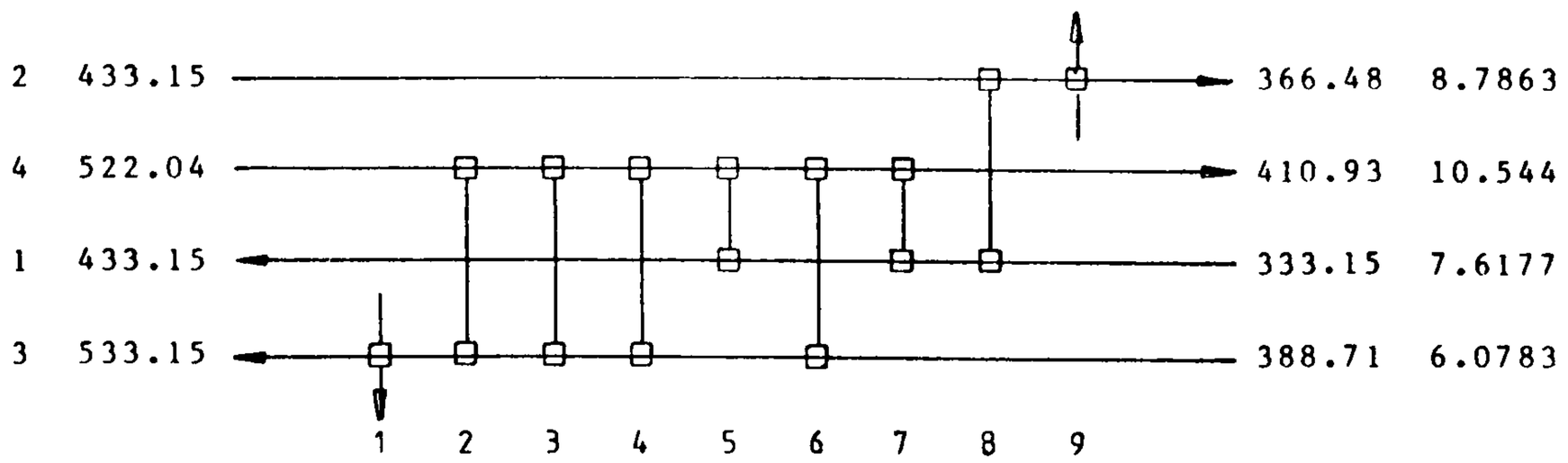
ΔT_{min} = 11.111K Cost Indication = 29.236

Heat exchanger loads/kW

1 : 135.1 2 : 117.2 3 : 203.3 4 : 352.4 5 : 70.0

6 : 194.3 7 : 234.2 8 : 330.7 9 : 255.1

Figure 7.32 The solution given in figure 7.31 using 1/2 exchangers

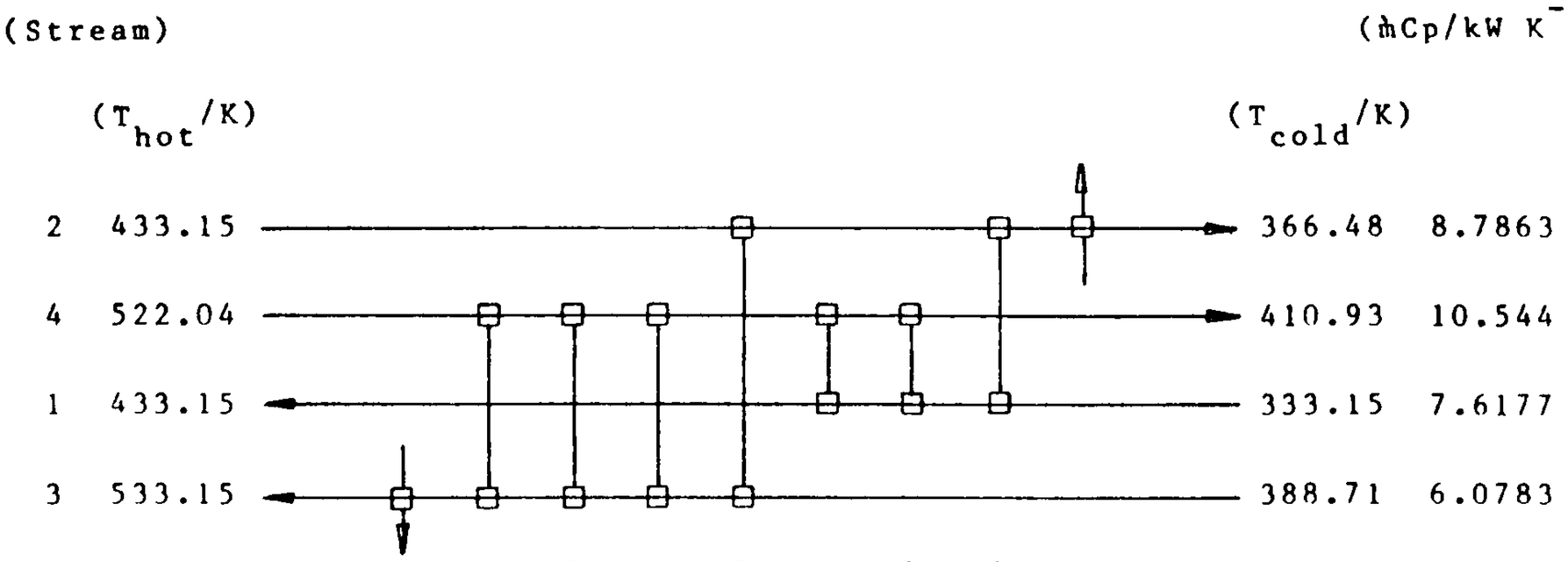


ΔT_{min} = 11.111K Cost Indication = 28.673

Heat exchanger loads/kW

1 : 135.1	2 : 117.2	3 : 203.3	4 : 352.4	5 : 264.3
6 : 70.0	7 : 164.2	8 : 330.7	9 : 255.1	

Figure 7.33 The second nine-shell solution to 4SP1



ΔT_{min} = 11.111K Cost Indication = 28.493

Heat exchanger loads/kW

1 : 135.1	2 : 117.2	3 : 203.3	4 : 352.4	5 : 70.0
6 : 264.3	7 : 234.3	8 : 260.6	9 : 255.1	

Figure 7.34 The third nine-shell solution to 4SP1

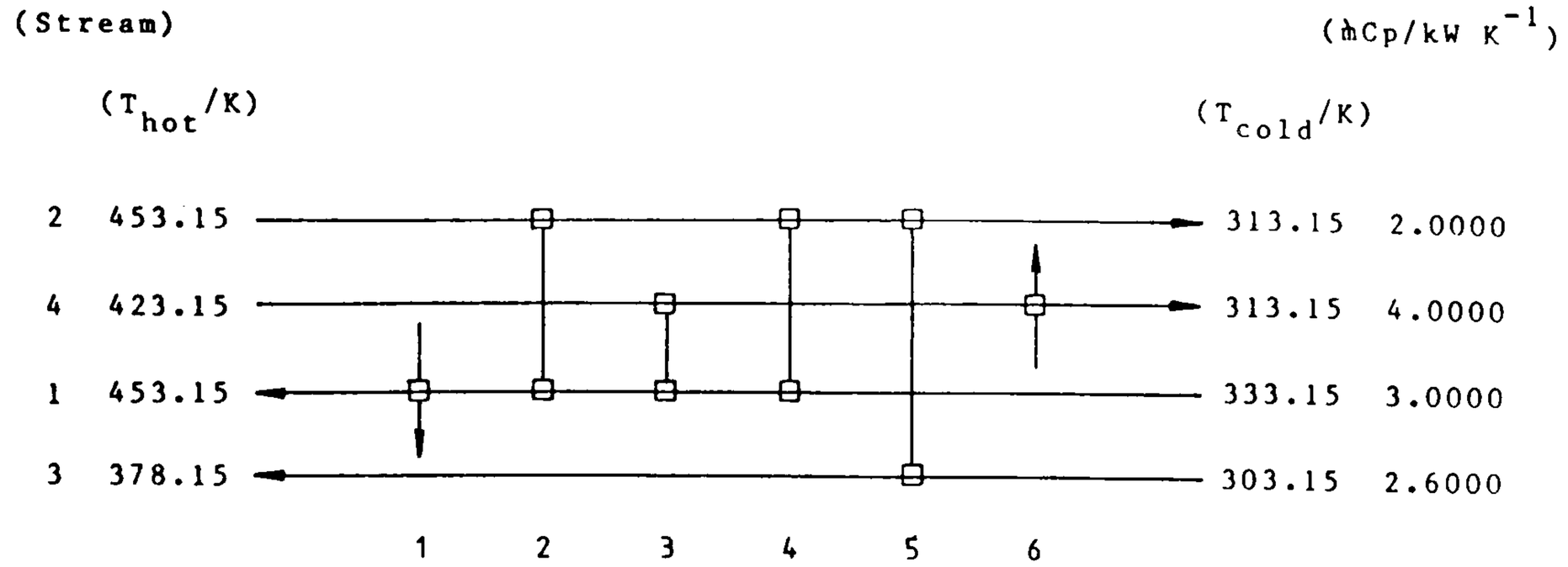
7.6 PROBLEM FIVE

The fifth problem to be discussed here was introduced by Linnhoff and Flower (1978a) and is named TC1. (N.B. Linnhoff et al have introduced different problems in other works also known as TC1). A summary of the relevant data is given in appendix M. The construction of the temperature interval table for the root node shows that the problem is pinched and so both hot and cold utilities are required.

The optimal solution when counter-current exchangers are permitted was first presented by Linnhoff and Flower (1978a) and is reproduced in figure 7.35. The same network, when constructed of 1/2 shell-and-tube heat exchangers, is shown in figure 7.36 and contains eighteen shells.

The new method isolates five solutions with seventeen shells, shown in figures 7.37 to 7.41. All the solutions feature a multi-shell match between process stream four and the cold utility stream.

The number of shells in this match could be considered as excessive. If so, two obvious amendments can be made to reduce the number; since a clean utility stream is involved a counter-current exchanger can probably be used for this duty, alternatively the cold utility stream outlet temperature can be lowered. If the outlet temperature of the cold utility stream is decreased, the reduction in the number of shells is offset by an increase in utility cost. An extreme case of this type of amendment is shown in figure 7.42 where the cold utility stream is modelled as a single-component phase change at 303.15K. Eighteen such networks with twelve shells were found by HERNE.

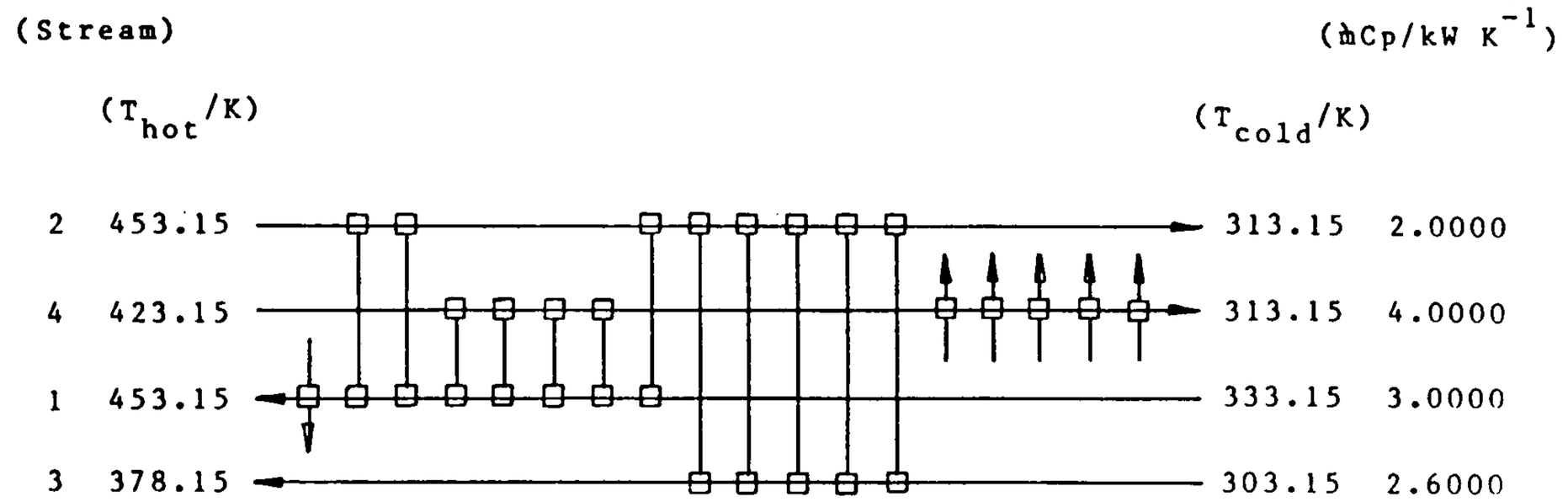


ΔT_{min} = 10.000K

Heat exchanger loads/kW

1 : 60.0	2 : 60.0	3 : 215.0	4 : 25.0	5 : 195.0
6 : 225.0				

Figure 7.35 The optimal solution to TC1 using counter-current exchangers

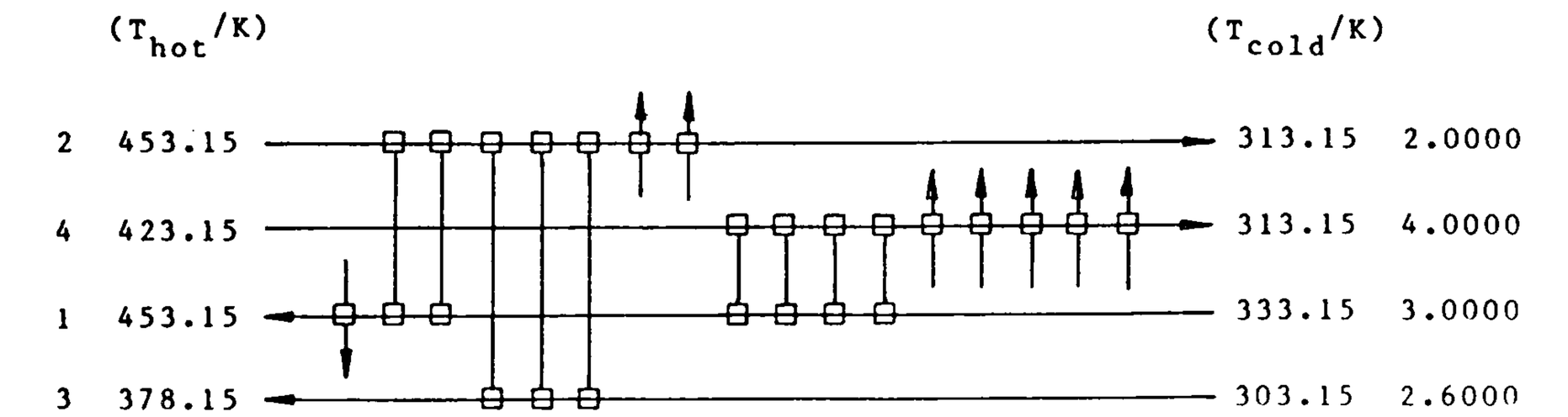


ΔT_{min} = 10.000K Cost Indication = 33.563

Heat exchanger loads/kW

1 : 60.0	2 : 30.0	3 : 30.0	4 : 21.3	5 : 47.1
6 : 62.8	7 : 83.8	8 : 25.0	9 : 34.1	10 : 57.1
11 : 43.9	12 : 33.8	13 : 26.0	14 : 8.3	15 : 64.1
16 : 57.0	17 : 50.6	18 : 45.0		

Figure 7.36 The solution given in figure 7.35 using 1/2 exchangers

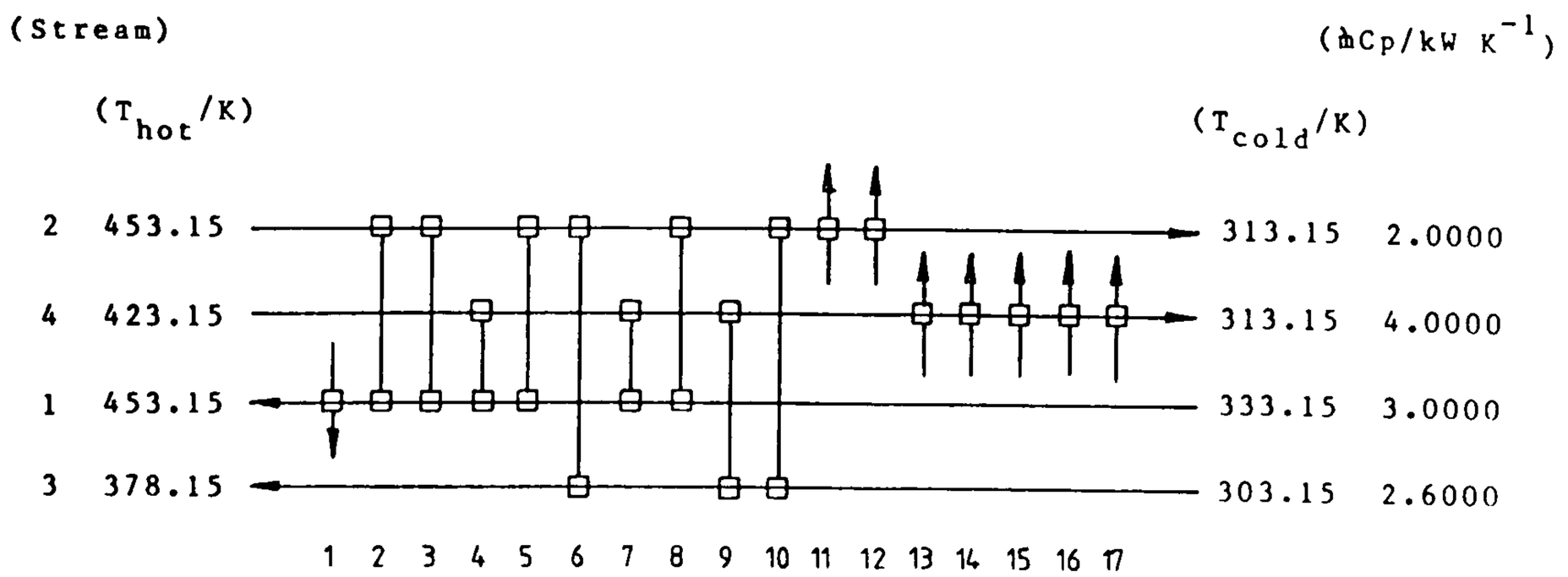


ΔT_{min} = 10.000K Cost Indication = 30.434

Heat exchanger loads/kW

1 : 60.0	2 : 30.0	3 : 30.0	4 : 90.0	5 : 69.2
6 : 35.8	7 : 20.0	8 : 5.0	9 : 40.0	10 : 53.3
11 : 71.1	12 : 75.6	13 : 40.0	14 : 40.0	15 : 40.0
16 : 40.0	17 : 40.0			

Figure 7.37 The first seventeen-shell solution to TC1

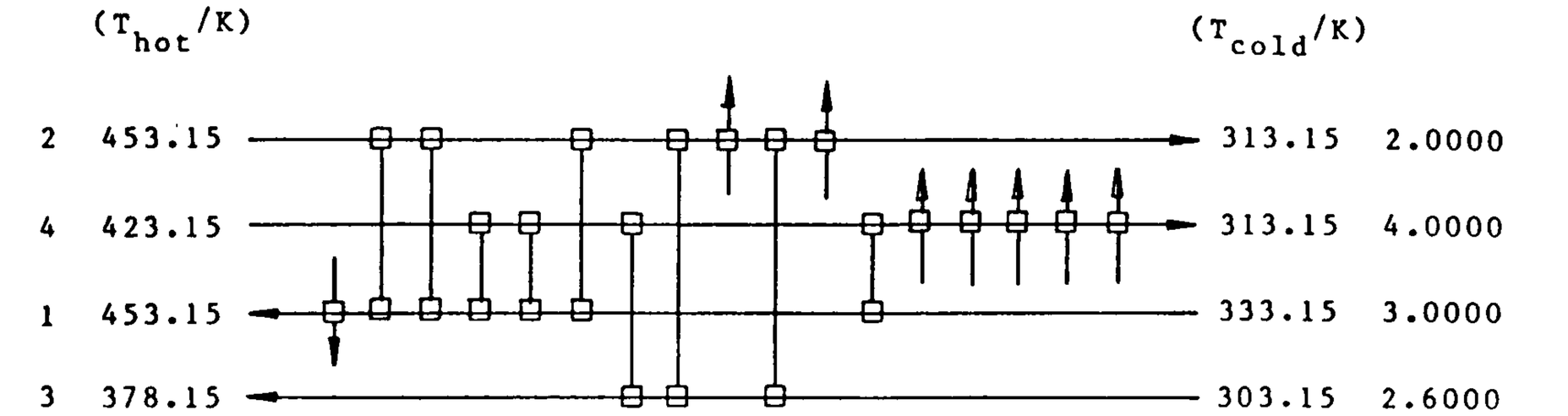


ΔT_{min} = 10.000K Cost Indication = 30.112

Heat exchanger loads/kW

1 : 60.0	2 : 30.0	3 : 30.0	4 : 40.0	5 : 46.7
6 : 43.3	7 : 115.6	8 : 37.8	9 : 91.1	10 : 60.6
11 : 20.0	12 : 11.7	13 : 40.0	14 : 40.0	15 : 40.0
16 : 40.0	17 : 33.3			

Figure 7.38 The second seventeen-shell solution to TC1

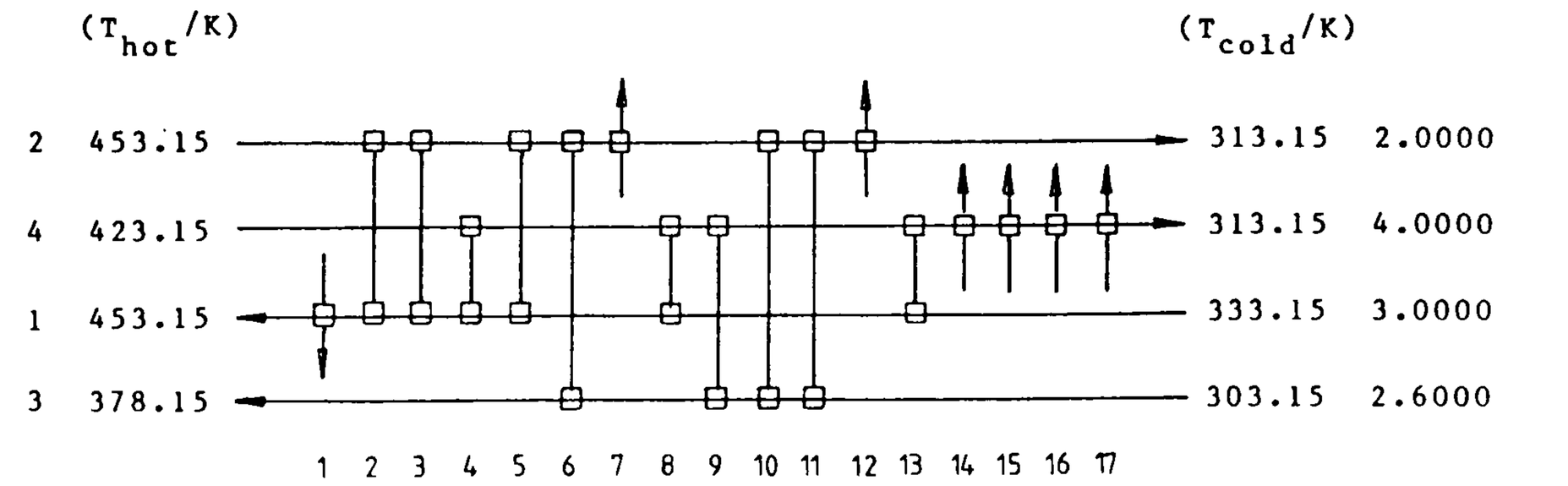


ΔT_{min} = 10.000K Cost Indication = 30.309

Heat exchanger loads/kW

1 : 60.0	2 : 30.0	3 : 30.0	4 : 40.0	5 : 53.3
6 : 82.2	7 : 86.7	8 : 74.5	9 : 20.0	10 : 33.9
11 : 9.4	12 : 64.4	13 : 40.0	14 : 40.0	15 : 40.0
16 : 40.0	17 : 35.6			

Figure 7.39 The third seventeen-shell solution to TC1

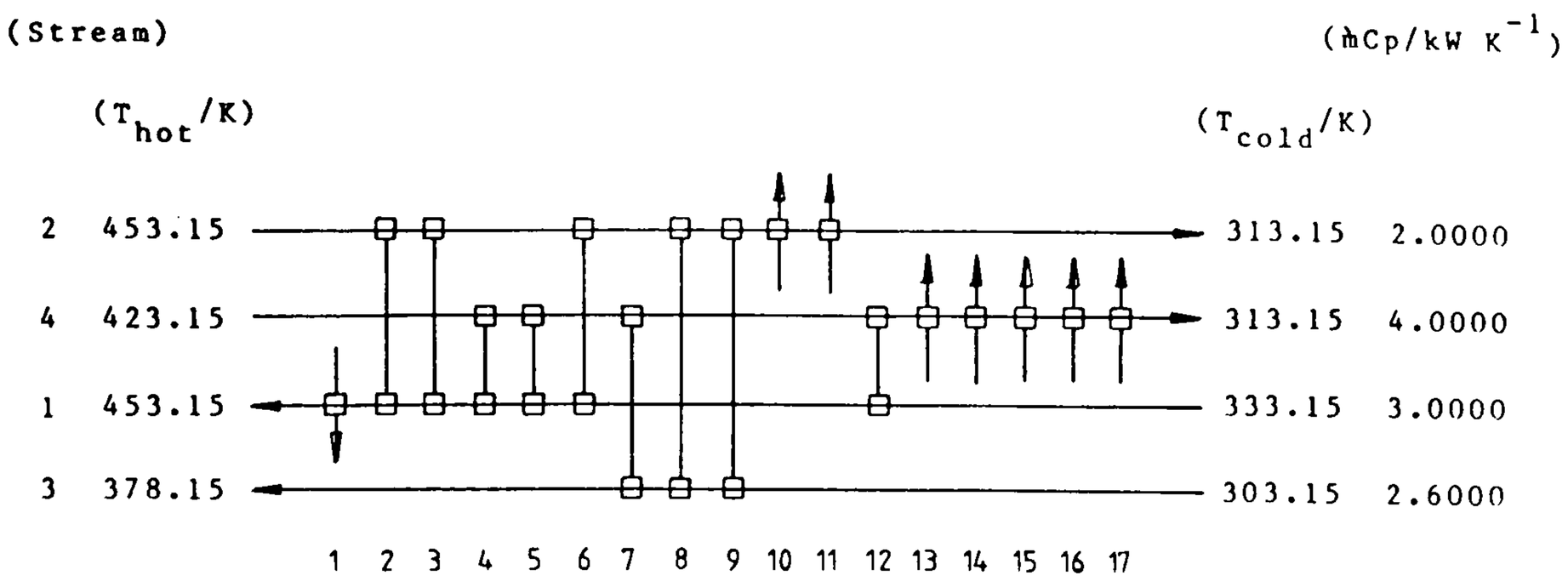


ΔT_{min} = 10.000K Cost Indication = 30.258

Heat exchanger loads/kW

1 : 60.0	2 : 30.0	3 : 30.0	4 : 40.0	5 : 46.7
6 : 43.3	7 : 50.0	8 : 115.6	9 : 91.1	10 : 53.4
11 : 7.1	12 : 19.4	13 : 37.8	14 : 40.0	15 : 40.0
16 : 40.0	17 : 35.6			

Figure 7.40 The fourth seventeen-shell solution to TC1

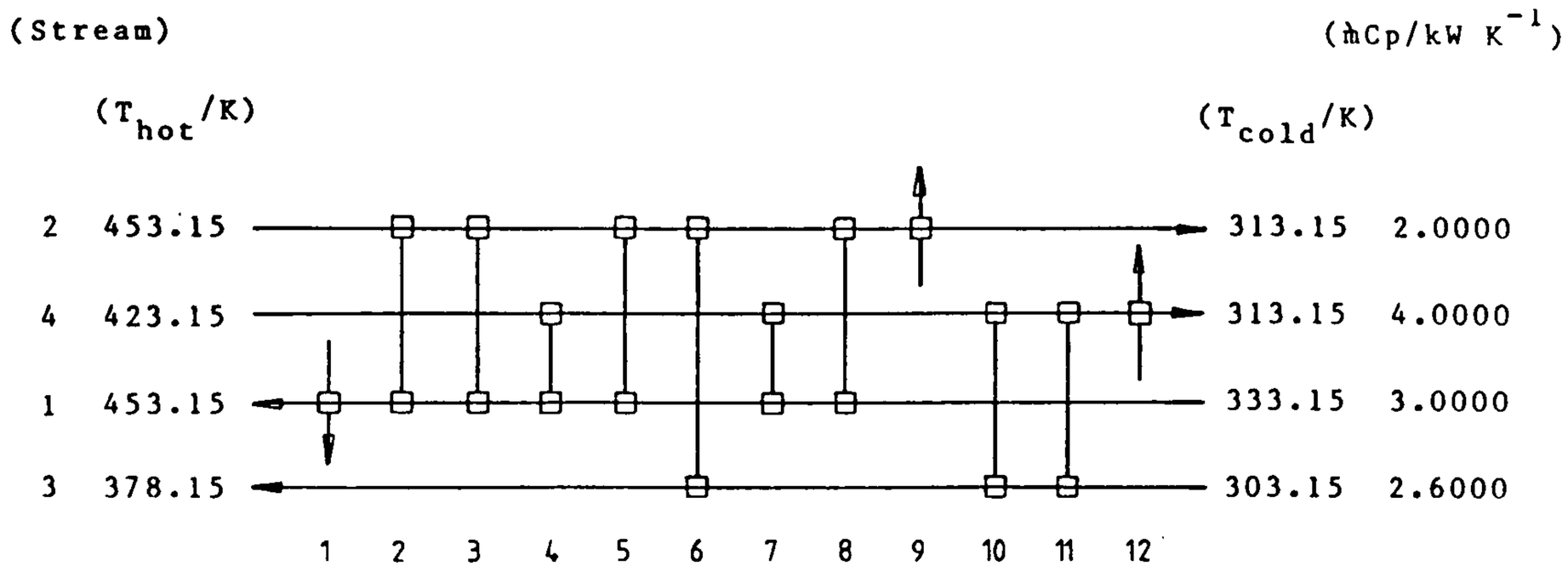


ΔT_{min} = 10.000K Cost Indication = 30.165

Heat exchanger loads/kW

1 : 60.0	2 : 30.0	3 : 30.0	4 : 40.0	5 : 53.3
6 : 82.2	7 : 86.7	8 : 74.5	9 : 33.9	10 : 20.0
11 : 9.4	12 : 64.4	13 : 40.0	14 : 40.0	15 : 40.0
16 : 40.0	17 : 35.6			

Figure 7.41 The fifth seventeen-shell solution to TC1



ΔT_{min} = 10.000K Cost Indication = 22.639

Heat exchanger loads/kW

1 : 60.0	2 : 30.0	3 : 30.0	4 : 40.0	5 : 46.7
6 : 43.3	7 : 115.6	8 : 37.8	9 : 92.2	10 : 91.1
11 : 60.6	12 : 132.8			

Figure 7.42 A twelve-shell solution to TC1

7.7 PROBLEM SIX

The sixth problem to be discussed here comes from an unpublished work by Linnhoff. A summary of the relevant data is given in appendix M. The construction of the temperature interval table for the root node shows that the problem is pinched and so both hot and cold utilities are required.

A good solution containing counter-current exchangers was given by Linnhoff and is reproduced in figure 7.43. The same network when constructed of 1/2 shell-and-tube heat exchangers is shown in figure 7.44 and contains nineteen shells, eight above the pinch point and eleven below.

An analysis of the problem around the pinch point shows that stream number two must be split if maximum energy recovery is to be obtained. A two-way split with equal flowrates was chosen in order to simplify the design.

The new method isolates fifty-two sub-networks above the pinch point containing eight shells each, with costs ranging from 6.2813 to 6.7868. Examples are given in figures 7.45 to 7.48. These costs are higher than that of the original solution because of the different split fraction used. The optimal split ratio could be easily found by multiple runs of HERNE.

Below the pinch point eight networks, each composed of eight shells, were found with costs in the range 8.7426 to 8.8627, with no stream splits. Examples are given in figures 7.49 to 7.52. Again the use of stream splits when not essential is shown to be questionable.

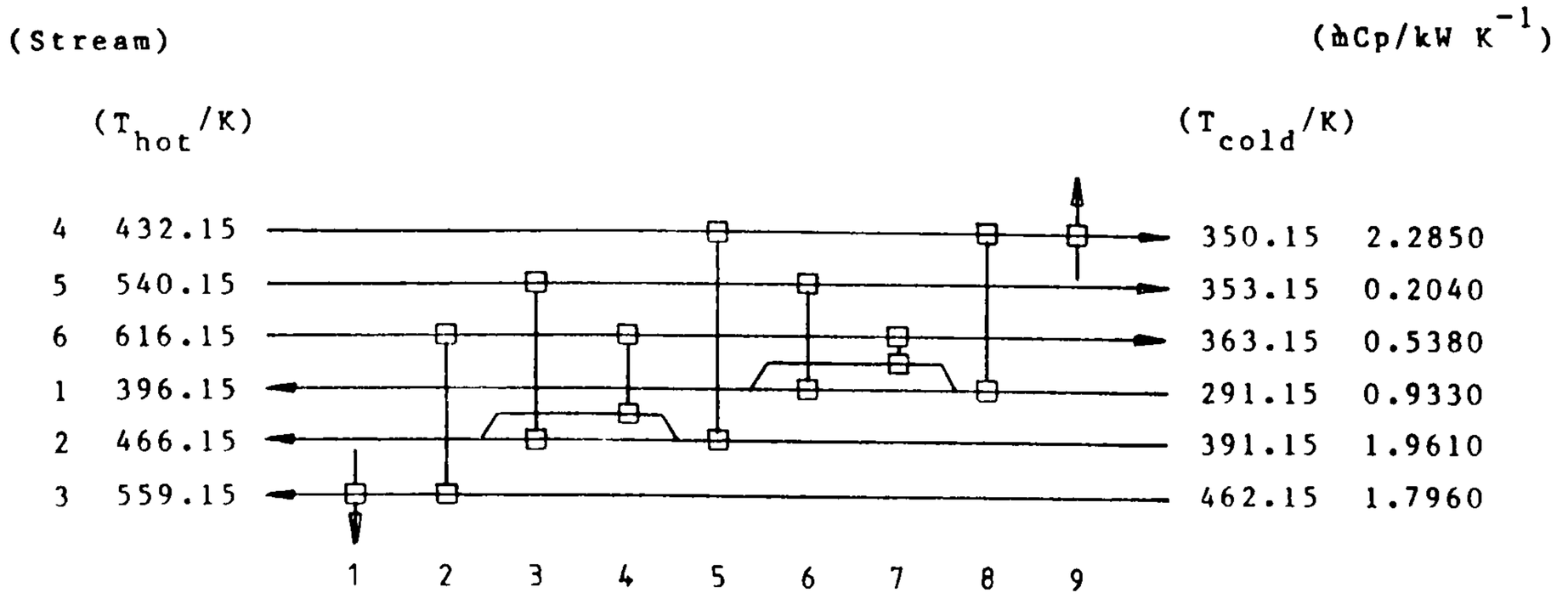


Figure 7.43 A near-optimal solution to the six-stream problem using counter-current exchangers

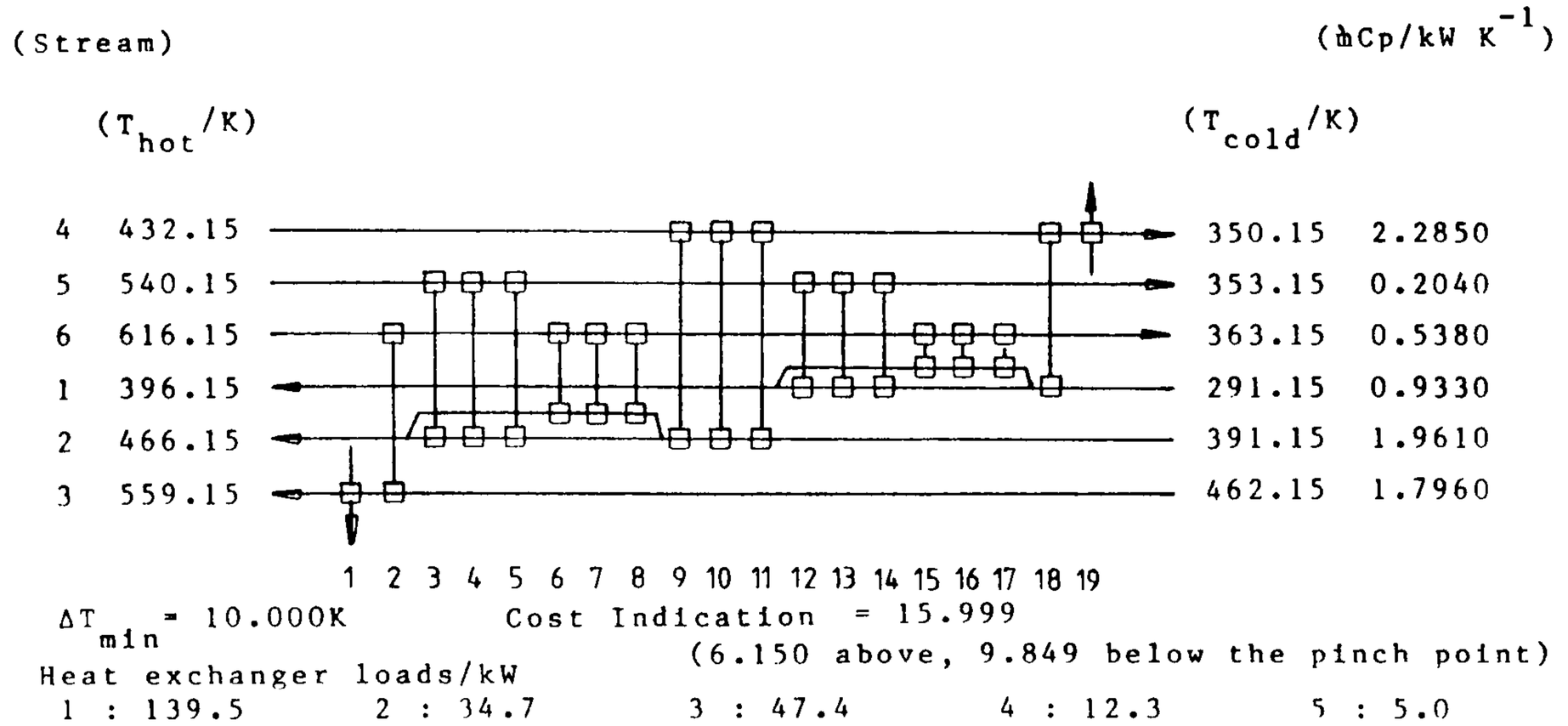


Figure 7.44 The solution given in figure 7.43 using 1/2 exchangers

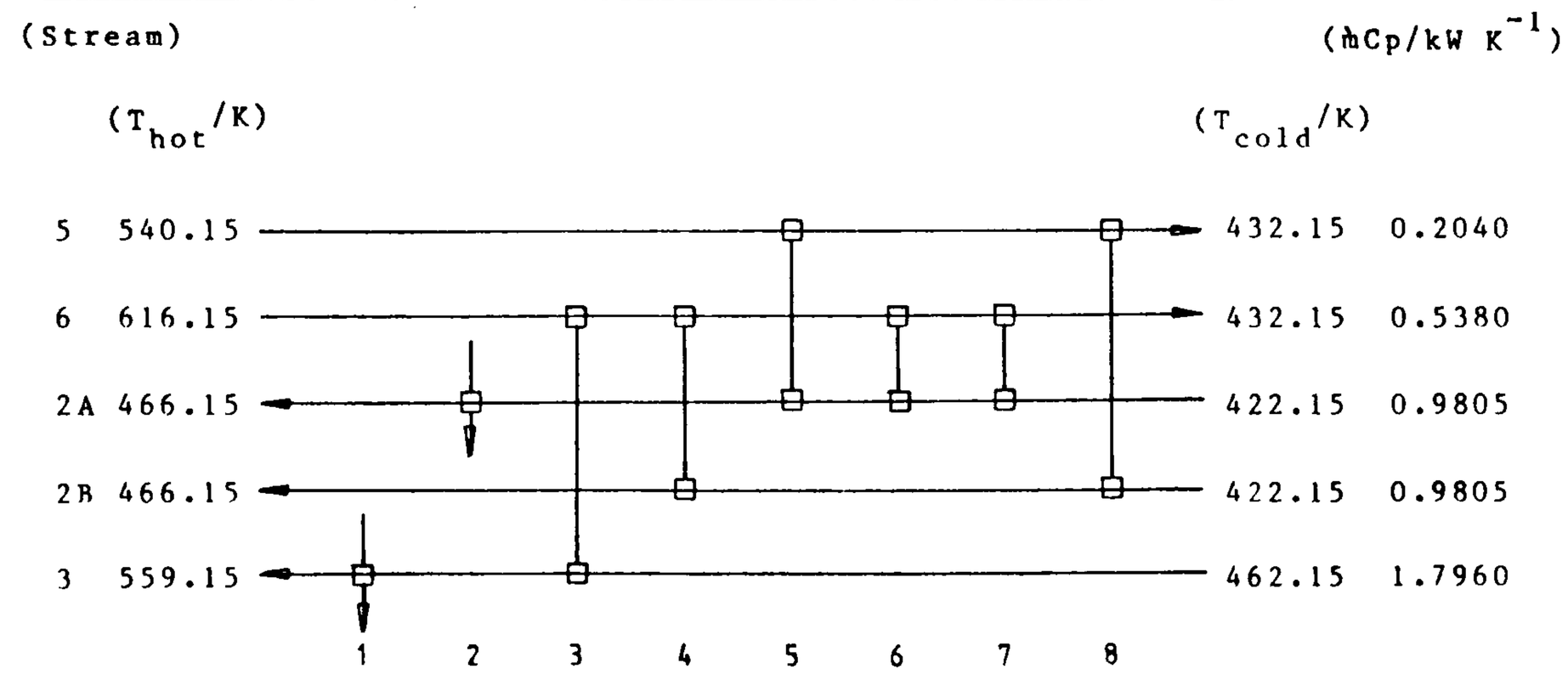
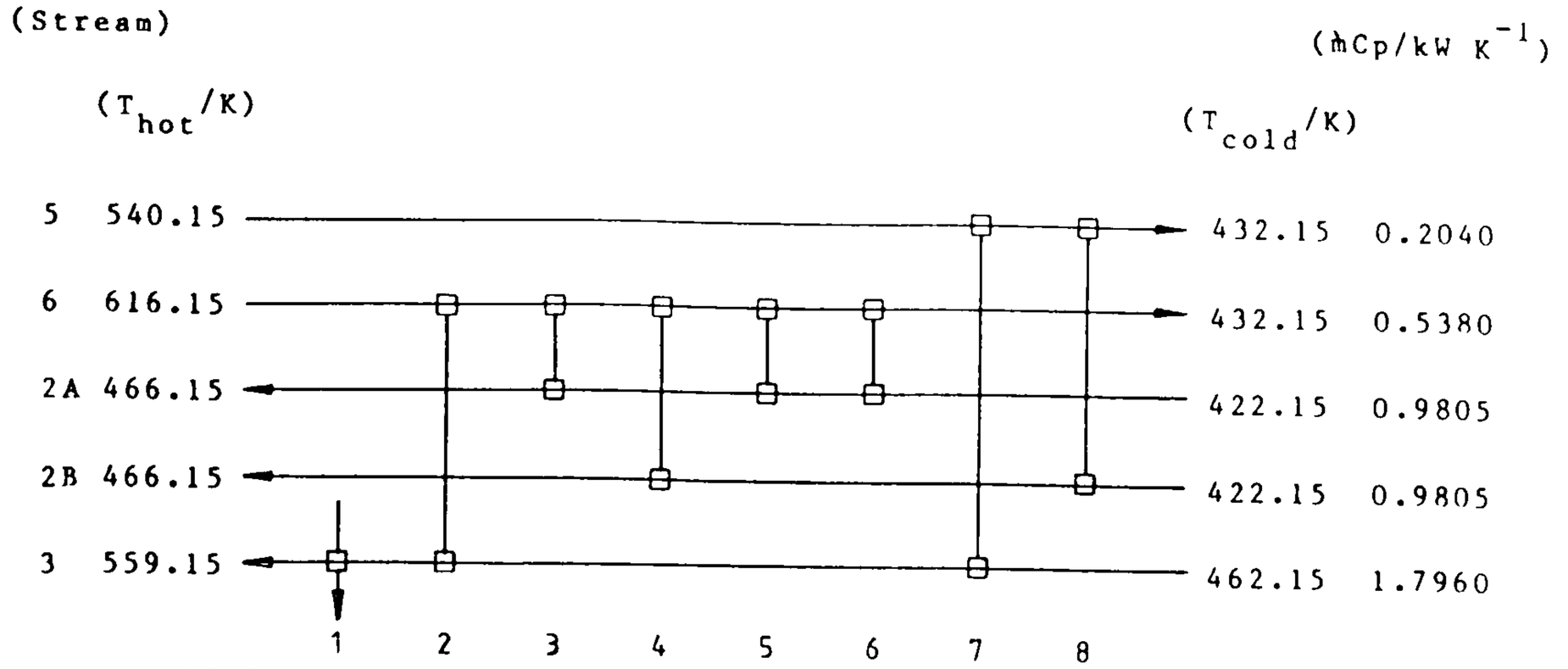


Figure 7.45 The first eight-shell above-pinch solution to the six-stream problem

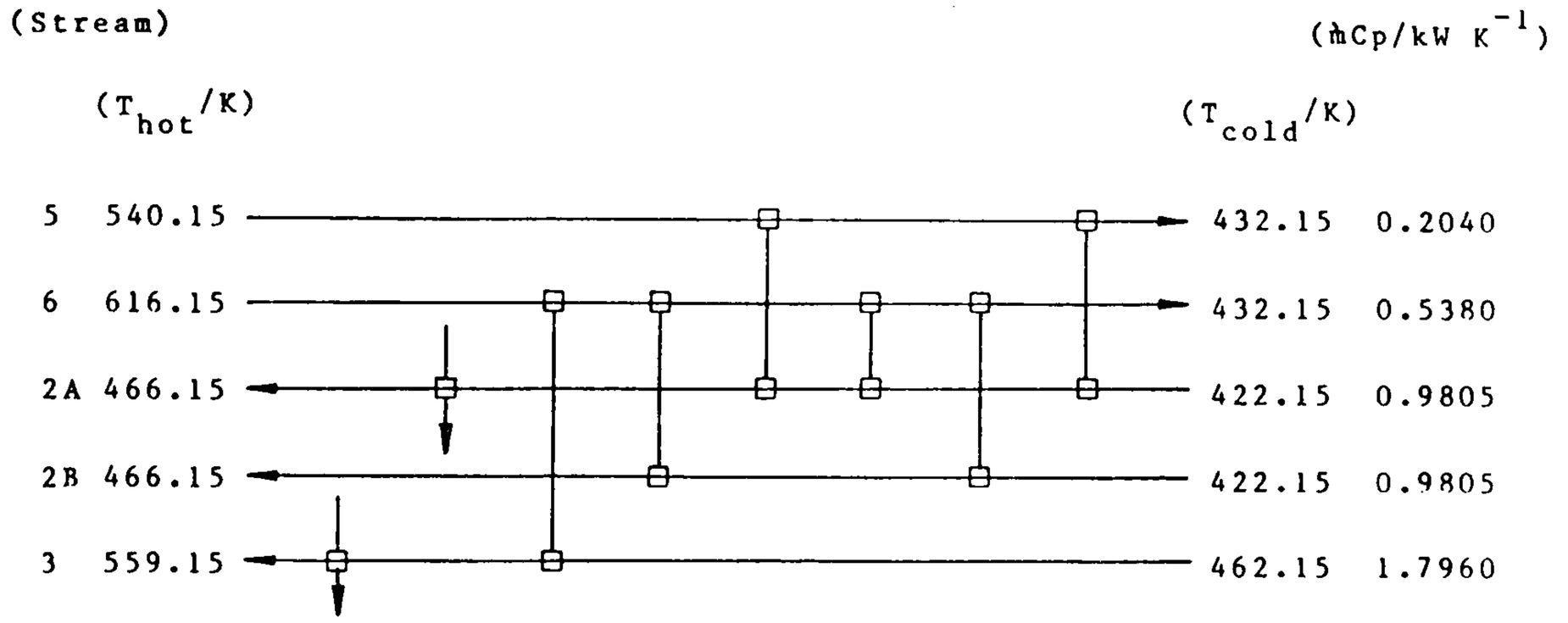


ΔT_{min} = 10.000K Cost Indication = 6.4003

Heat exchanger loads/kW

1 : 139.5	2 : 22.5	3 : 15.5	4 : 33.3	5 : 17.9
6 : 9.8	7 : 12.2	8 : 9.8		

Figure 7.46 The second eight-shell above-pinch solution to the six-stream problem

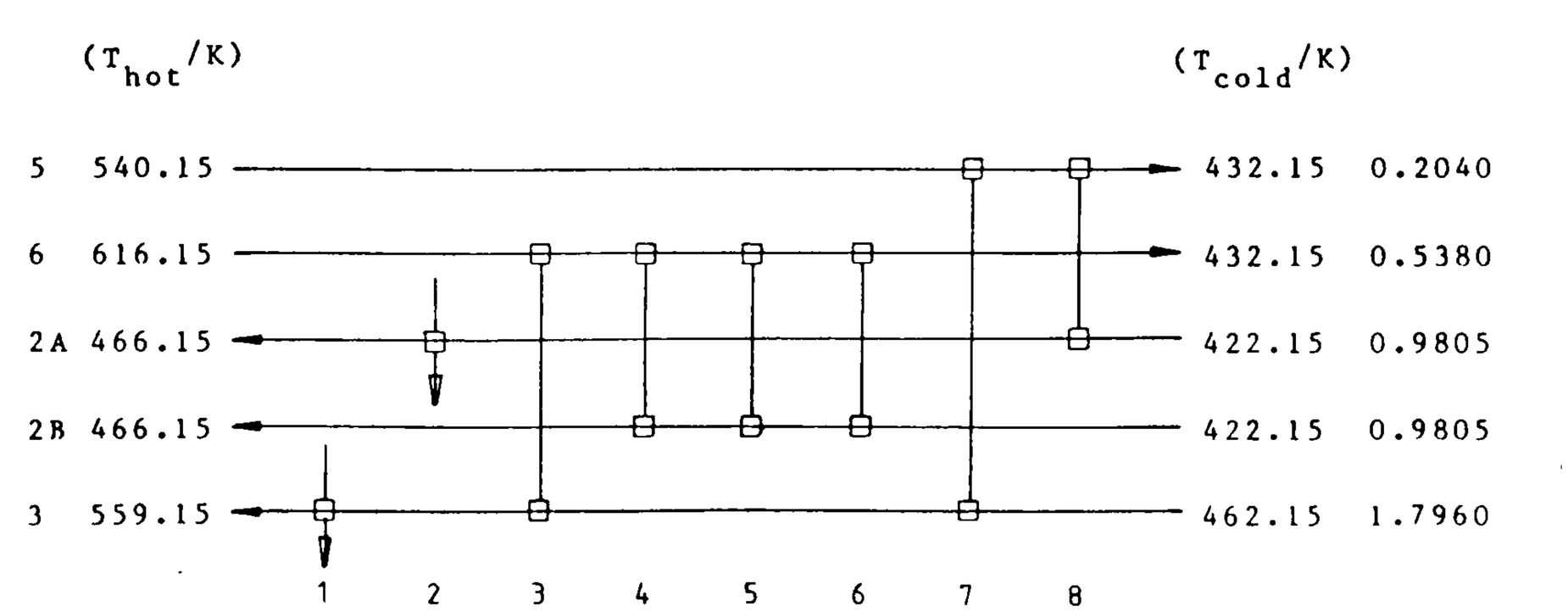


ΔT_{min} = 10.000K Cost Indication = 6.2879

Heat exchanger loads/kW

1 : 136.2	2 : 3.2	3 : 38.0	4 : 33.3	5 : 12.2
6 : 17.9	7 : 9.8	8 : 9.8		

Figure 7.47 The third eight-shell above-pinch solution to the six-stream problem



ΔT_{min} = 10.000K Cost Indication = 6.5557

Heat exchanger loads/kW

1 : 106.1	2 : 33.3	3 : 55.8	4 : 15.5	5 : 17.9
6 : 9.8	7 : 12.2	8 : 9.8		

Figure 7.48 The fourth eight-shell above-pinch solution to the six-stream problem

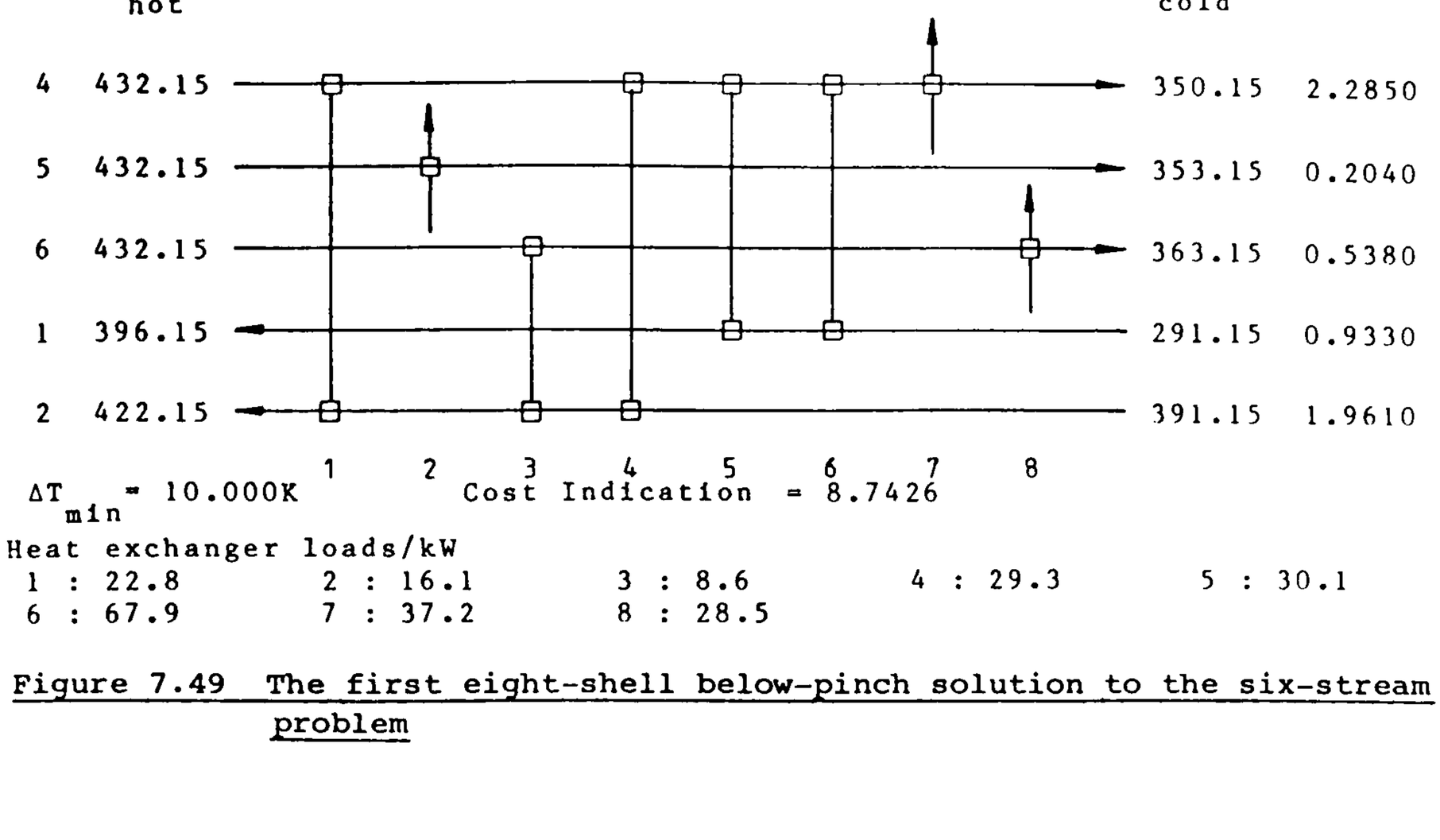


Figure 7.49 The first eight-shell below-pinch solution to the six-stream problem

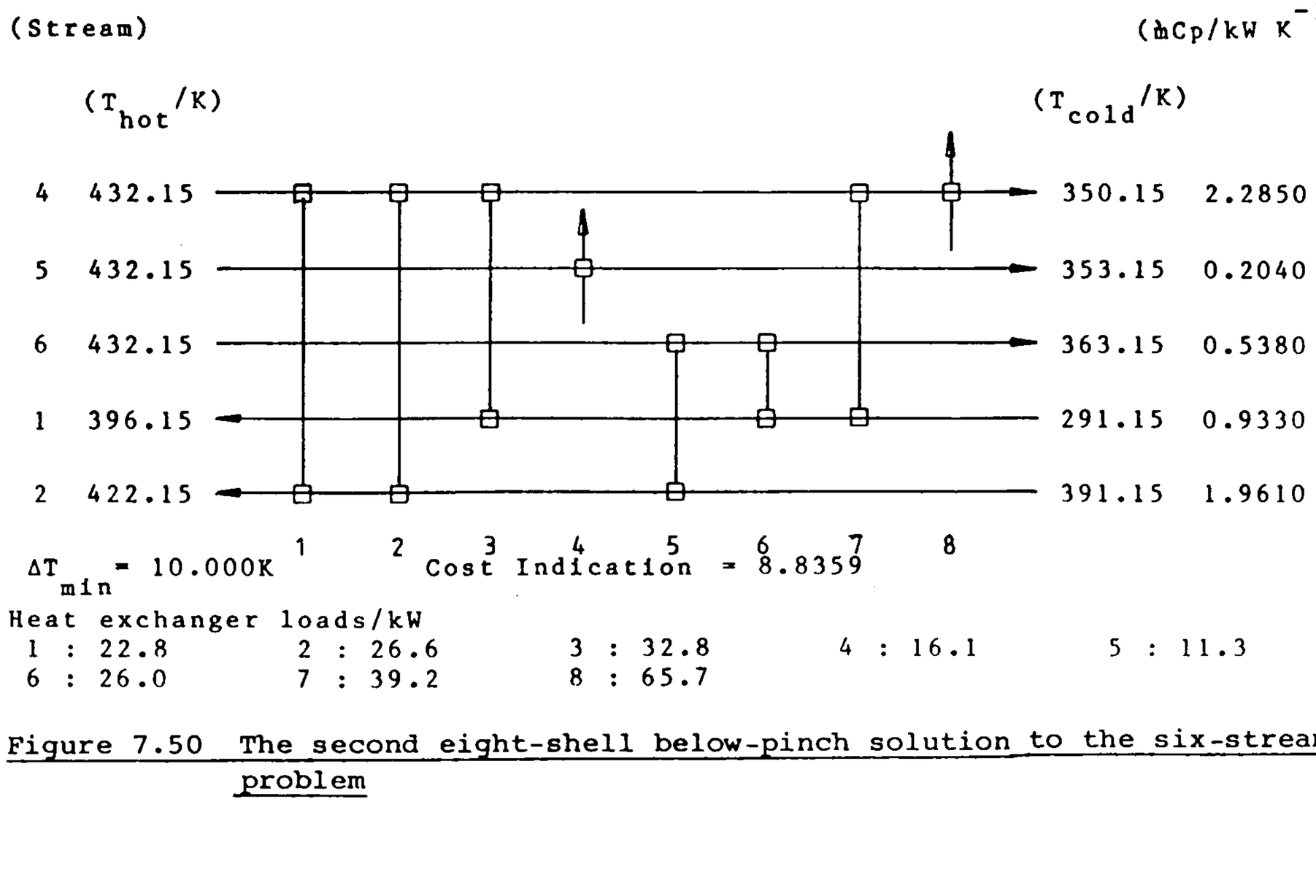
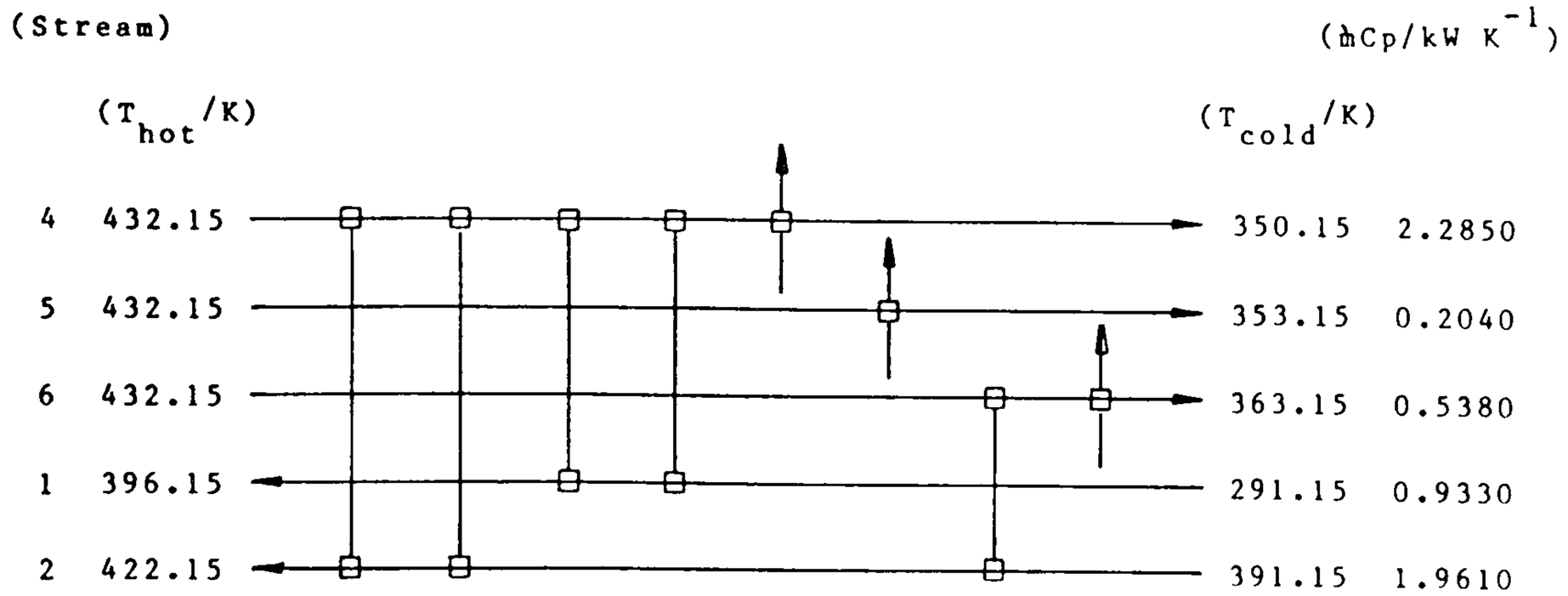


Figure 7.50 The second eight-shell below-pinch solution to the six-stream problem



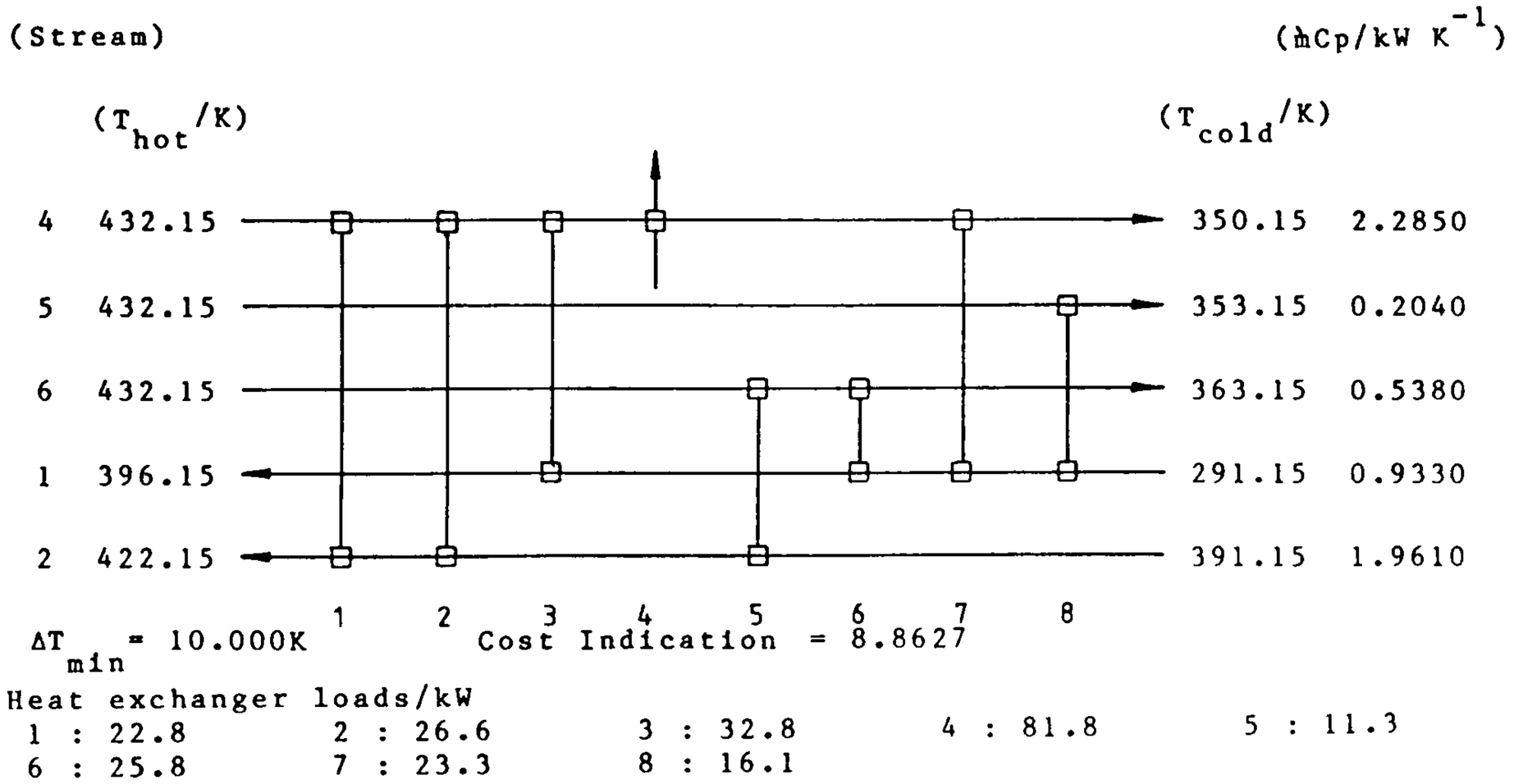
ΔT_{min} = 10.000K Cost Indication = 8.8177

Heat exchanger loads/kW

1 : 22.8 2 : 26.6 3 : 32.8 4 : 65.2 5 : 40.0

6 : 16.1 7 : 11.3 8 : 25.8

Figure 7.51 The third eight-shell below-pinch solution to the six-stream problem



ΔT_{min} = 10.000K Cost Indication = 8.8627

Heat exchanger loads/kW

1 : 22.8 2 : 26.6 3 : 32.8 4 : 81.8 5 : 11.3

6 : 25.8 7 : 23.3 8 : 16.1

Figure 7.52 The fourth eight-shell below-pinch solution to the six-stream problem

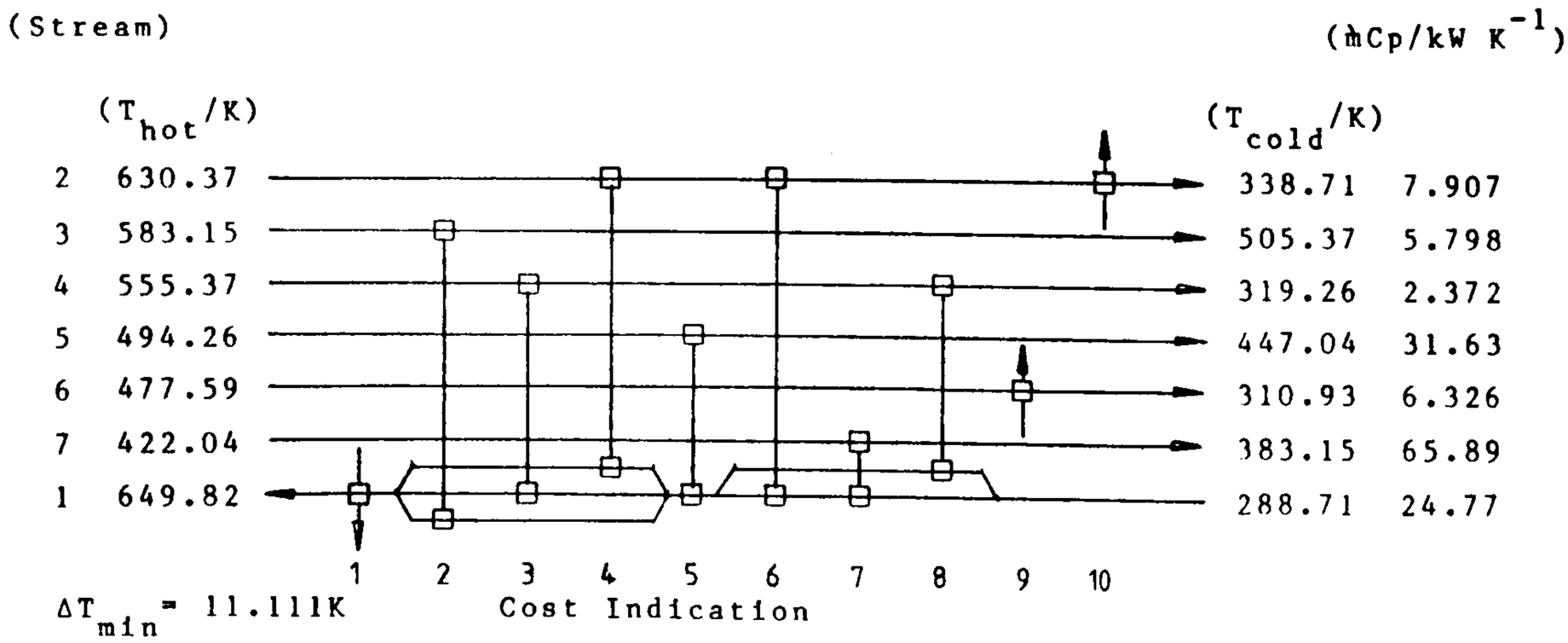
7.8 PROBLEM SEVEN

The seventh problem to be discussed here was introduced by Papoulias and Grossmann (1983b) and is known conventionally as 7SP4. A summary of the relevant data is given in appendix M. The construction of the temperature interval table for the root node shows that the problem is pinched and so both hot and cold utilities are required.

The solution given by Papoulias and Grossmann (1983b) is reproduced in figure 7.53. The same network constructed with 1/2 shell-and-tube heat exchangers is given in figure 7.54 and contains twenty-six shells, eleven above the pinch point and fifteen below.

As in problem six, an analysis of the problem around the pinch point shows that a cold stream must be split into two branches immediately above the pinch point. An equal split was chosen again.

The new method quickly isolates one hundred solutions, the upper limit that is imposed, with eleven shells, for the above pinch point sub-problem, with costs ranging from 38.210 to 40.225. Examples are given in figures 7.55 to 7.58. This is achieved without recourse to a three-way split which was featured in the original solution. Below the pinch point ten solutions containing eleven shells each were found with costs in the range 63.500 to 66.821. Examples are given in figures 7.59 to 7.62. No stream split is required in this sub-problem.

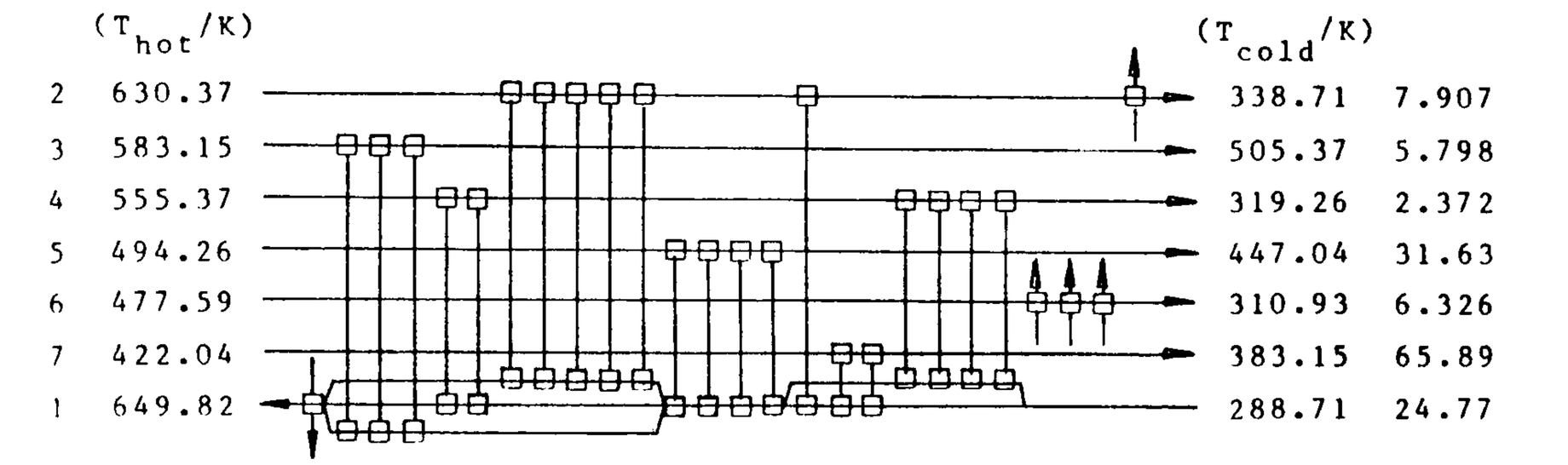


ΔT_{min} = 11.111K

Heat exchanger loads/kW

1 : 2457.5	2 : 451.0	3 : 145.0	4 : 1076.2	5 : 1493.6
6 : 346.0	7 : 2562.7	8 : 415.1	9 : 1054.3	10 : 883.8

Figure 7.53 A near-optimal solution to 7SP4 using counter-current exchangers



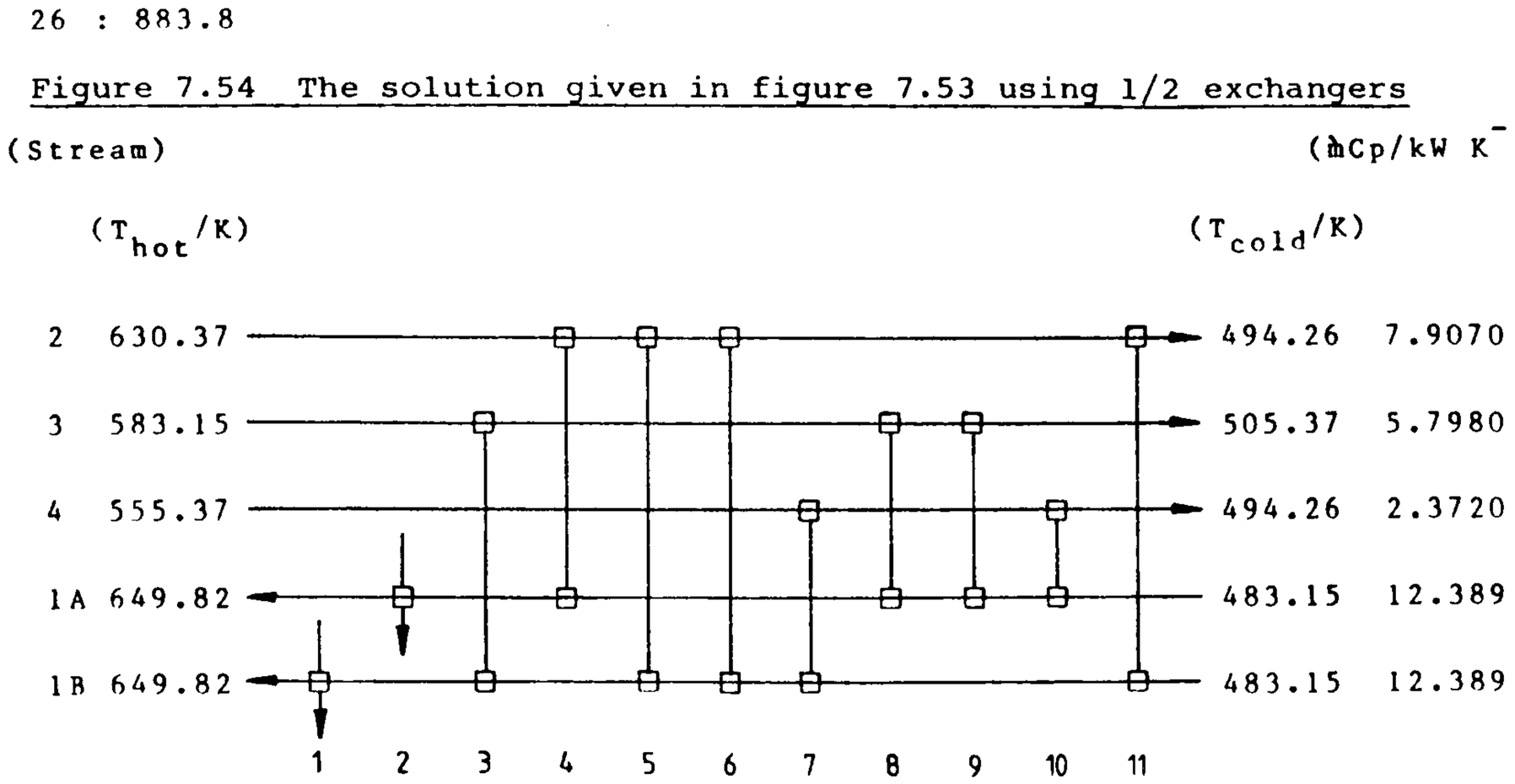
ΔT_{min} = 11.111K Cost Indication = 109.069

(38.437 above, 70.632 below the pinch point)

Heat exchanger loads/kW

1 : 2457.5	2 : 35.6	3 : 239.7	4 : 175.7	5 : 86.4
6 : 58.6	7 : 74.4	8 : 406.7	9 : 277.3	10 : 189.0
11 : 128.9	12 : 351.4	13 : 448.6	14 : 572.7	15 : 120.9
16 : 346.0	17 : 999.5	18 : 1563.2	19 : 169.3	20 : 129.8
21 : 99.5	22 : 16.4	23 : 913.7	24 : 70.3	25 : 70.3
26 : 883.8				

Figure 7.54 The solution given in figure 7.53 using 1/2 exchangers

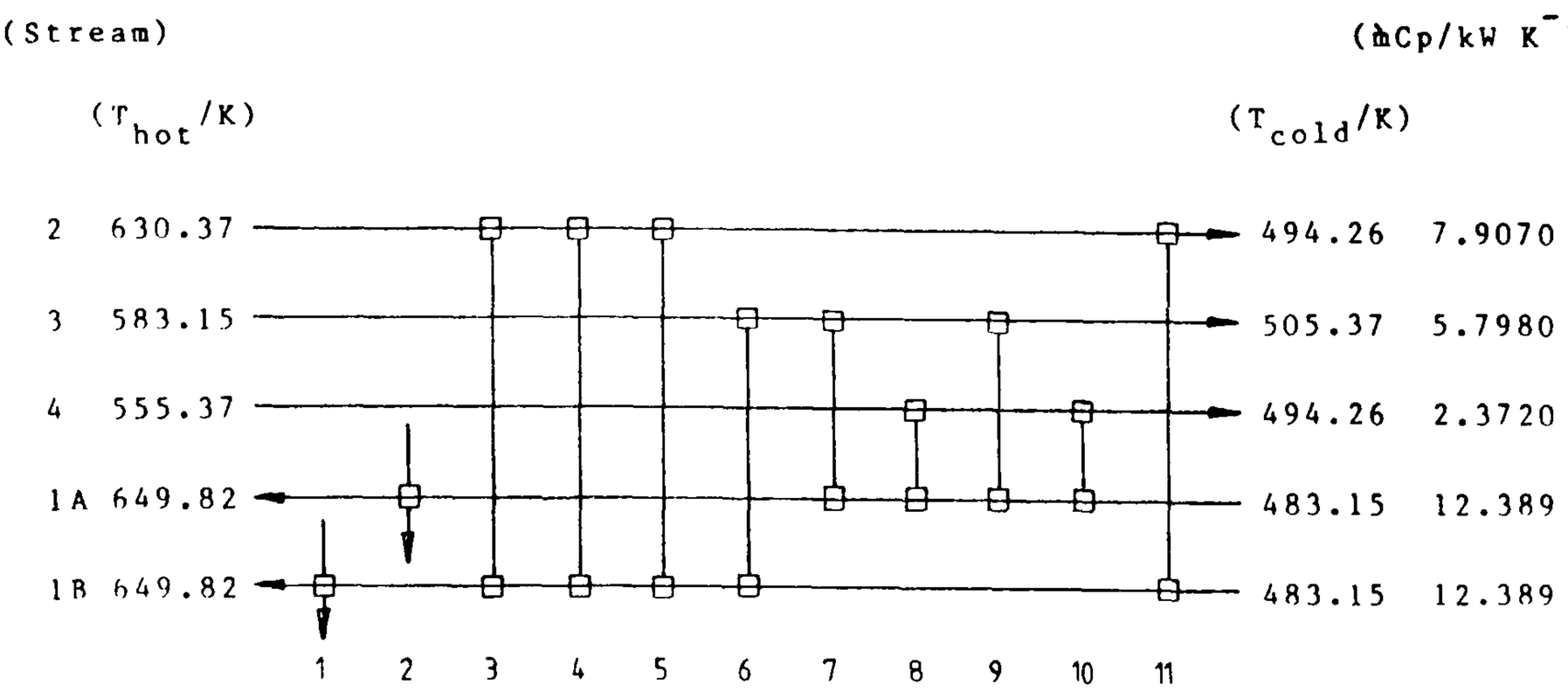


ΔT_{min} = 11.111K Cost Indication = 38.250

Heat exchanger loads/kW

1 : 1366.0	2 : 1091.6	3 : 19.1	4 : 403.8	5 : 326.5
6 : 208.4	7 : 7.3	8 : 294.2	9 : 137.6	10 : 137.6
11 : 137.6				

Figure 7.55 The first eleven-shell above-pinch solution to 7SP4

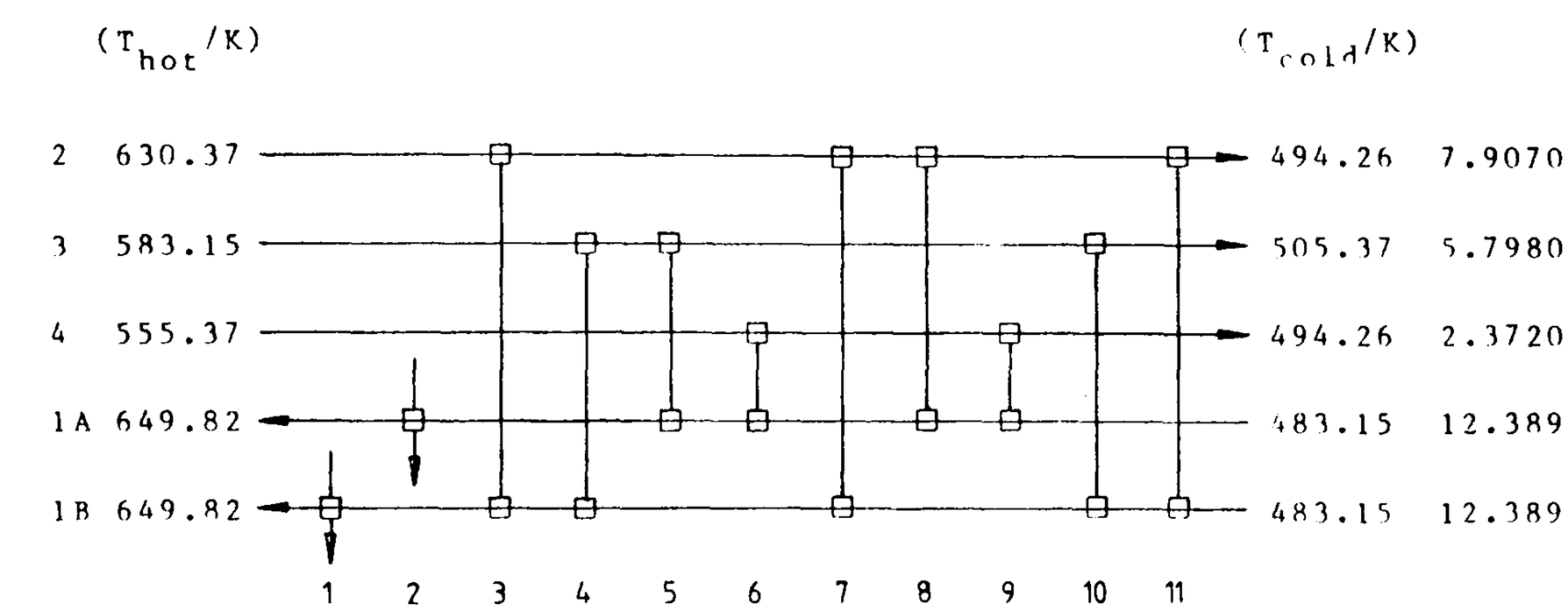


ΔT_{min} = 11.111K Cost Indication = 39.161

Heat exchanger loads/kW

1 : 962.3	2 : 1495.4	3 : 452.8	4 : 296.5	5 : 189.3
6 : 26.4	7 : 286.9	8 : 7.3	9 : 137.6	10 : 137.6
11 : 137.6				

Figure 7.56 The second eleven-shell above-pinch solution to 7SP4

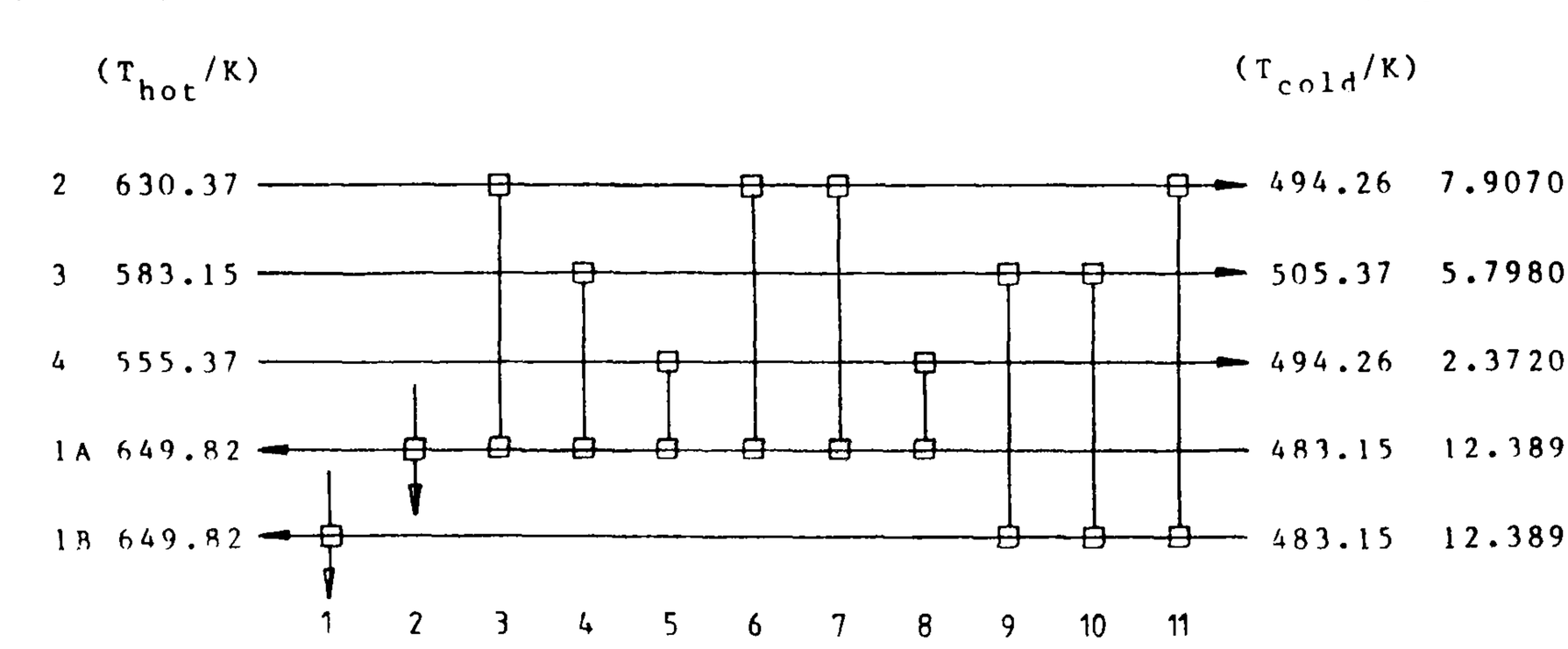


ΔT_{min} = 11.111K Cost Indication = 39.474

Heat exchanger loads/kW

1 : 962.3	2 : 1495.4	3 : 307.0	4 : 104.4	5 : 208.9
6 : 7.3	7 : 415.9	8 : 215.7	9 : 137.6	10 : 137.6
11 : 137.6				

Figure 7.57 The third eleven-shell above-pinch solution to 7SP4

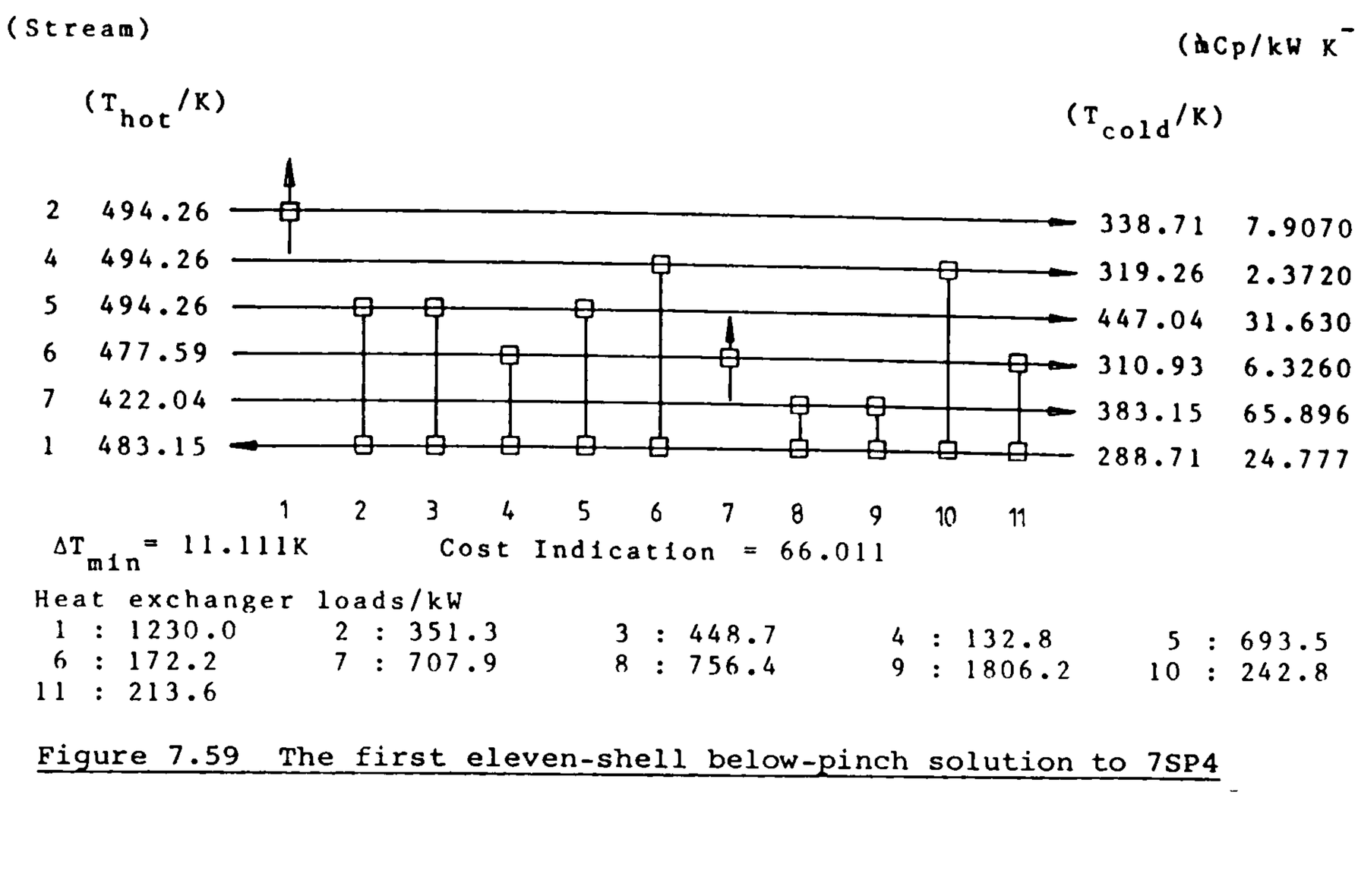


ΔT_{min} = 11.111K Cost Indication = 38.953

Heat exchanger loads/kW

1 : 1495.4	2 : 962.3	3 : 384.9	4 : 19.2	5 : 7.3
6 : 337.8	7 : 215.7	8 : 137.6	9 : 294.2	10 : 137.6
11 : 137.6				

Figure 7.58 The fourth eleven-shell above-pinch solution to 7SP4

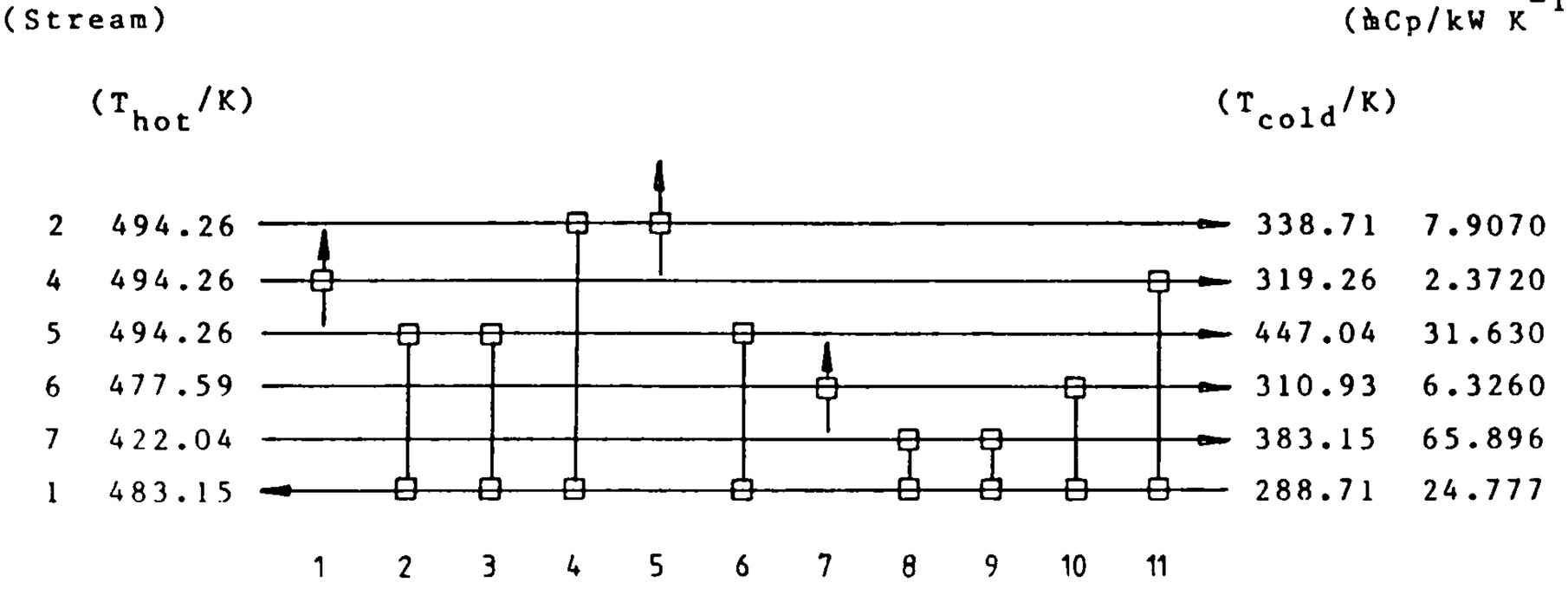


ΔT_{min} = 11.111K Cost Indication = 66.011

Heat exchanger loads/kW

1 : 1230.0	2 : 351.3	3 : 448.7	4 : 132.8	5 : 693.5
6 : 172.2	7 : 707.9	8 : 756.4	9 : 1806.2	10 : 242.8
11 : 213.6				

Figure 7.59 The first eleven-shell below-pinch solution to 7SP4

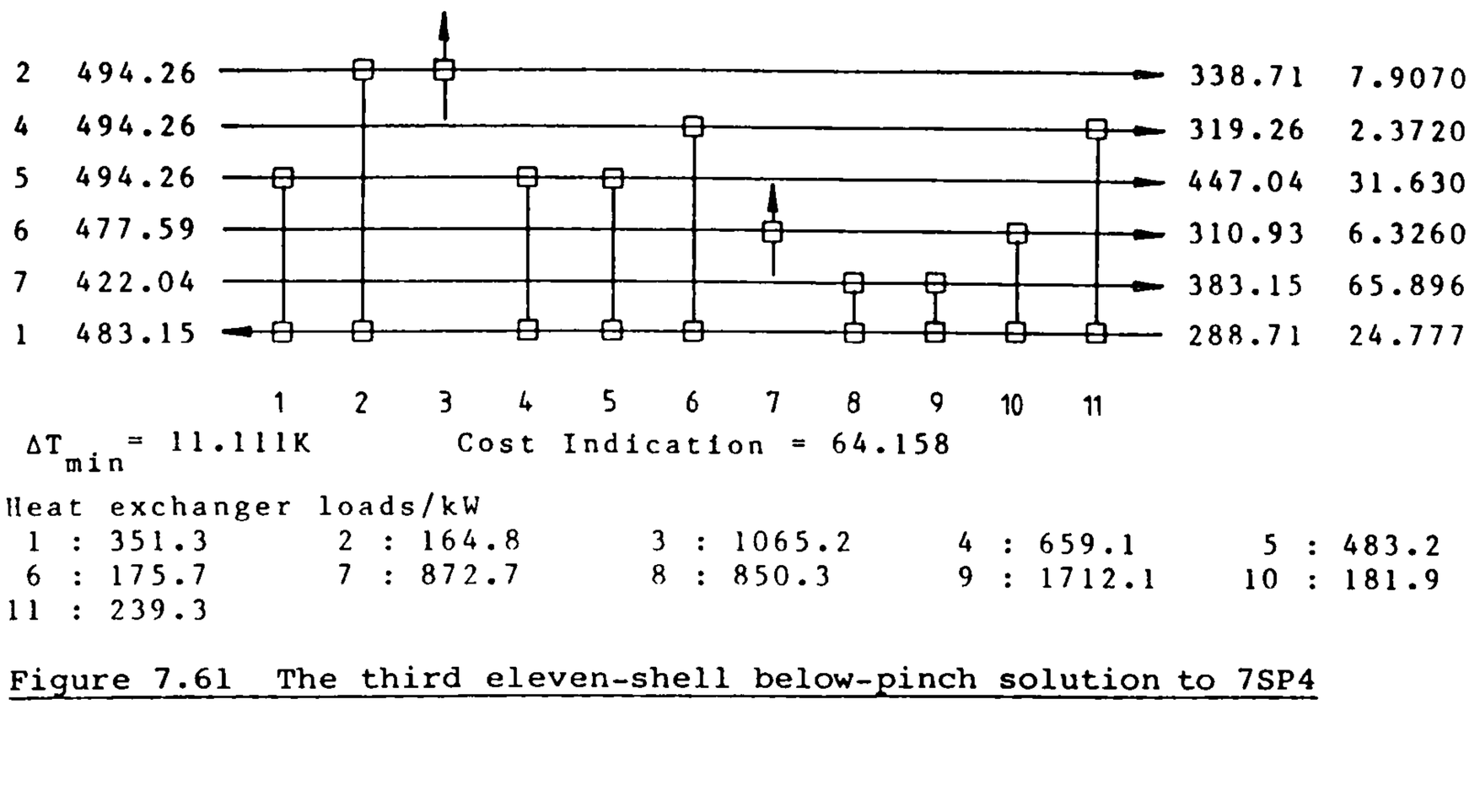


ΔT_{min} = 11.111K Cost Indication = 63.619

Heat exchanger loads/kW

1 : 382.2	2 : 351.3	3 : 448.7	4 : 343.2	5 : 886.8
6 : 693.5	7 : 669.0	8 : 858.3	9 : 1704.4	10 : 385.3
11 : 33.0				

Figure 7.60 The second eleven-shell below-pinch solution to 7SP4

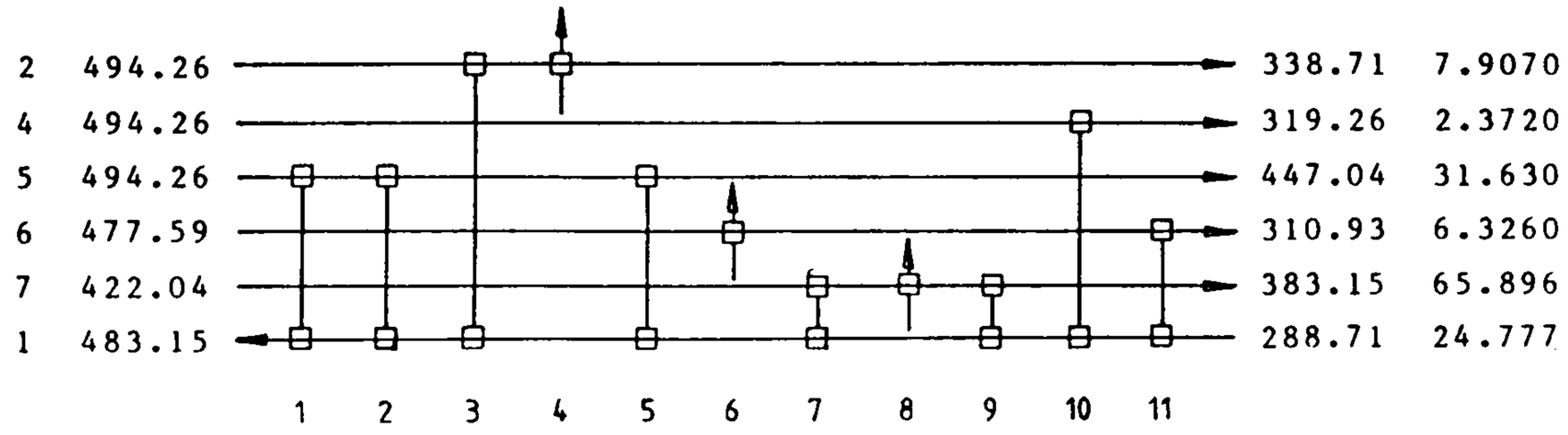


ΔT_{min} = 11.111K Cost Indication = 64.158

Heat exchanger loads/kW

1 : 351.3	2 : 164.8	3 : 1065.2	4 : 659.1	5 : 483.2
6 : 175.7	7 : 872.7	8 : 850.3	9 : 1712.1	10 : 181.9
11 : 239.3				

Figure 7.61 The third eleven-shell below-pinch solution to 7SP4



ΔT_{min} = 11.111K Cost Indication = 64.080

Heat exchanger loads/kW

1 : 351.3	2 : 448.7	3 : 343.2	4 : 886.8	5 : 693.5
6 : 913.7	7 : 858.3	8 : 137.7	9 : 1567.1	10 : 415.0
11 : 140.5				

Figure 7.62 The fourth eleven-shell below-pinch solution to 7SP4

7.9 THE CUT-OFF TEMPERATURE RANGE

All the solutions to problems found so far have been obtained by tree searches which are almost complete. The only constraints imposed on the searches have been on the number of solutions found at any particular level of the tree, which led to tree pruning in some cases. As problems become larger a complete search becomes infeasible and the expansion of the tree must be carefully controlled. This is achieved by using heuristics to limit the size of the search tree to manageable proportions.

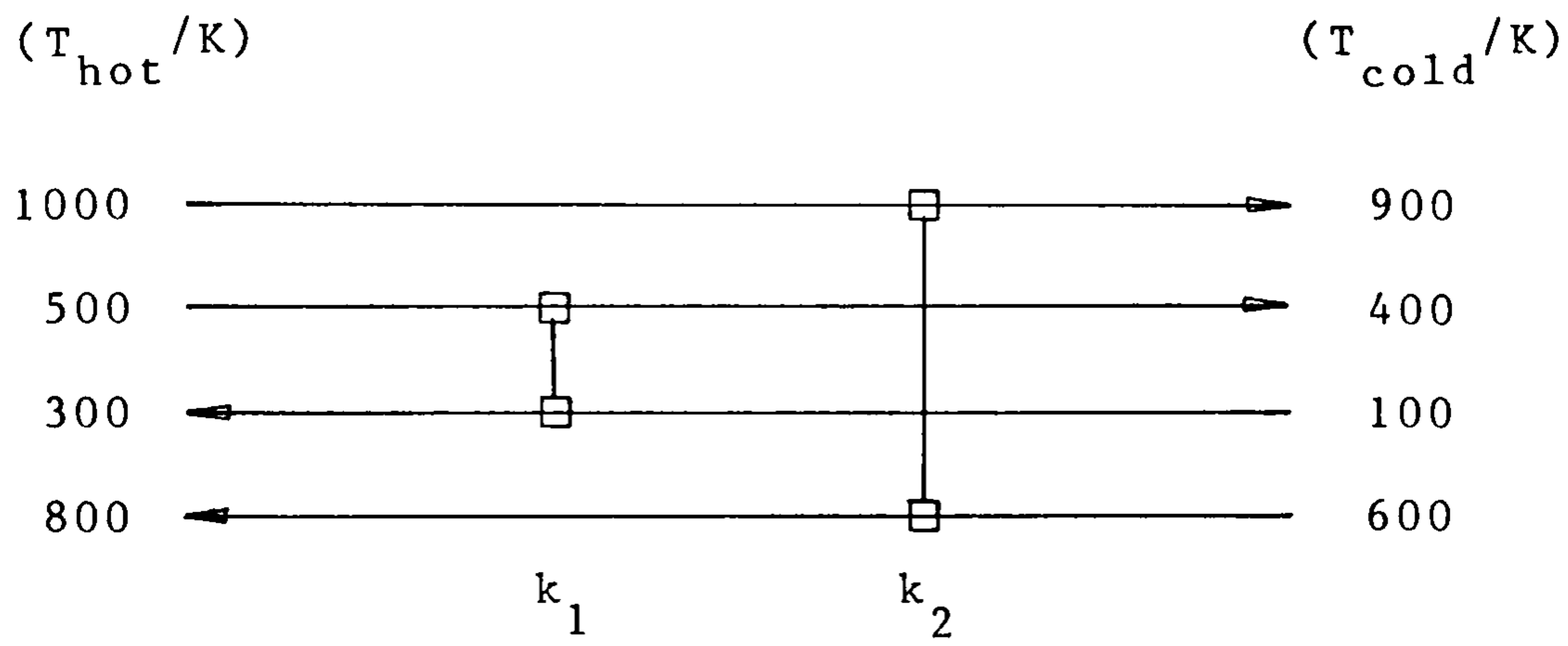
One such heuristic used in this work limits the number of elements stored in the ordered lists which are used by the branching strategy. This is achieved by ensuring all the members of the ordered lists are within a certain temperature range of the top element of the list. This temperature range is called the cut-off temperature range. A small value for this temperature range, say 25K, often reduces the search tree size by many orders of magnitude but also makes the risk of finding only sub-optimal solutions unacceptably high. Large values of the cut-off temperature range, say 150K or above, do little to reduce the size of the problem tree. Obviously there is some intermediate value to be found which balances the work involved in exploring the search tree against the risk of not finding an optimal solution.

7.10 THE INTERACTION BETWEEN THE BRANCHING STRATEGY AND THE PROCEDURE TO ENSURE ONE-TO-ONE CORRESPONDENCE OF SOLUTIONS AND NETWORKS

The determination of a suitable value for the cut-off temperature range is complicated by an interaction between the procedure which ensures that any particular solution

is generated only once (i.e. the procedure which guarantees one-to-one correspondence of solutions and networks, see sections 5.13.4.1 and 6.5.2) and the heuristic which curtails the expansion of the ordered lists used by the branching strategy. One-to-one correspondence of solutions to networks is achieved by the association of a unique number (k_n) with each possible match. The number of the match about to be made is then checked against the numbers of the already existing matches to ascertain the legality of that match. This results in disjoint matches (i.e. matches with neither hot nor cold stream in common) appearing in one order only. However, this particular order may not be allowed by the shortened lists used by the branching strategy. For example, consider the two matches shown in figure 7.63. The procedure to guarantee one-to-one correspondence requires the matches to be made so that the unique number for the second match is greater than that for the first. However, when the branching strategy works down from the hot end it requires exchanger two be made before exchanger one unless the cut-off temperature range is greater than or equal to 500K. Obviously these requirements can clash making the generation of the above pair of exchangers impossible.

The conflict can be reduced by careful solution of the unique numbers for the matches. However, this is only strictly correct for the first few levels of the search tree and could conceivably fail further down in the tree. Nevertheless, in practice, this problem has not been apparent.



a) The one-to-one correspondence procedure requires k_1 to be greater than k_2 (see section 5.13.4.1)

b) The branching strategy requires exchanger 2 to be placed before exchanger 1 unless the cut-off temperature range is greater than or equal to 500K

Figure 7.63 Two exchangers which show a possible clash in the requirements of two of the tree-search procedures

7.11 SUITABLE VALUES FOR THE CUT-OFF TEMPERATURE RANGE

Appendix N shows in tabular form the way the number of solutions found to a problem varies with cut-off temperature. In most cases a figure in the range 75 to 100K seems to be suitable for initial tree searches.

7.12 EFFICIENT USE OF HERNE

The tables in appendix N show that the printing out of solutions forms a sizeable fraction of the total workload. Network synthesis is therefore best carried out by making an initial search which locates just one feasible solution at each level of the search tree (i.e. pruning the tree after one solution is found at any particular level). The output produced is thus very small. Once the number of heat exchangers in the optimal network has been found a second search can be carried out to locate more optimal solutions. The amount of output is thus kept to a minimum.

7.13 THE EFFECTIVENESS OF THE MATCHING STRATEGY

Table 7.6 shows that the matching strategy is not very effective in finding optimal solutions quickly. The number of nodes expanded to obtain the first optimal solution is not as efficient as was expected. This is probably a result of the interaction between the branching strategy and the procedure to ensure one-to-one correspondence.

Problem	Number of nodes to first optimal solution		Total number of nodes	
1	3797	-	4637	-
2	4253	-	10478	-
3	2798[x]	-	4049	-
4	1	10	1	54
5	3	1907	3	4767
6	9	1380	534	1875
7	16[x]	-[+]	1909	38894
	above	below	above	below
	pinch point		pinch point	

cut-off temperature = 75K

[+] optimal solution not found

[x] pruning took place at levels below the optimal level of the tree

Table 7.1 Number of nodes expanded to obtain the first optimal solution compared with the total number of nodes expanded for a cut-off temperature of 75K

7.14 SUMMARY OF RESULTS FOR THE LITERATURE PROBLEMS

The results are summarised in tables 7.2 and 7.3. It can be seen that the number of shells in most networks can be reduced by the new method. The decrease in network cost is not dramatic and further analysis would be required to see if the new networks obtained, with their more complex piping requirements, were cheaper than the modified counter-current networks. However, the use of the new method has shown that the number of stream splits can often be reduced, which brings a reduction in the cost of control equipment, without incurring any extra heat exchanger costs. In addition, the fouling problems often associated with stream splits can be reduced or removed completely.

Table 7.3 shows that the costs of alternative networks with the same number of heat exchanger shells is remarkably constant with the largest range in costs being of the order of 14%. This figure is comparable with the accuracy of the costing itself which brings into serious doubt the validity of chosing networks in initial synthesis procedures on the basis of cost.

The best value to use for the cut-off temperature range would appear to be of the order of 75K when the minimum temperature difference is approximately 10K.

Problem	New networks		Remodelled counter-current networks		Reduction in	
	Cost	Shells	Cost	Shells	Cost(%)	Shells
1	76.835	10	81.531	14	5.76	4
2	49.664	11	54.138	12	8.26	1
3	39.098	11	39.747	12	1.63	1
4	28.493	9	29.236	9	3.51	0
5	30.112	17	33.563	18	3.32	1
6	15.024	16	15.999	19	6.09	3
7	101.710	22	109.069	26	6.75	4

Table 7.2 Results summary

Problem	Lowest network cost	Hightest network cost	Range	% Spread from cheapest net-work
1	76.835	81.167	4.332	1.54
2	49.664	54.057	4.393	8.85
3	39.098	44.423	5.325	13.62
4	28.493	29.236	0.743	2.61
5	30.112	30.434	0.322	1.07
6	15.024	15.650	0.626	4.17
7	101.710	107.046	5.336	5.25

Table 7.3 Results summary

7.15 A NEW UNPINCHED PROBLEM

A new problem containing twenty-two process streams is the penultimate problem to be considered. The relevant data are given in table 7.4. The construction of the temperature interval table for the root node shows that the problem is unpinched and that cold utilities are required.

A large cooling utility heat load is indicated by the temperature interval table for the root node which is equivalent to the heat content of all the hot streams from 381.37K down to their outlet temperatures. Thus streams 3,4,5,6,10,17 and 19, which have entire ranges below this figure, are immediately removed from the problem by matching with cold utilities. Streams eleven and fifteen are also removed from the problem because the heat load of these streams above 381.37K is very small. A quick inspection of the remaining problem (see figure 7.64) shows that stream seven cannot be satisfied at its upper end because of the match restrictions imposed. Stream seven can only match with stream eight which enters at 389K. Thus stream seven can only be heated to 379K. Therefore the upper end of stream seven is matched with a utility stream and removed from the problem.

The new problem (see figure 7.65) was tackled using the new methods but no solution was forthcoming. A graph of the running totals from the temperature interval table for the root node plotted against temperature corrected for ΔTmin reveals the reason (see figure 7.66). In the temperature range 400.2K down to 384.0K there is an area where the heat cascaded is very small. A second pinch point almost exists at this point. An analysis of the stream population and heat capacity flowrates around this 'pseudo-pinch' shows that stream seven, with its

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp$/kWK^{-1} or $\dot{m}\lambda$ /kW	Stream Type
1	338.000	357.200	19.22	Cold process
2	368.800	368.800	8871.00	Cold process
3	368.800	338.000	13.34	Hot process
4	349.900	347.300	3360.80	Hot process
5	343.100	335.000	5.00	Hot process
6	334.000	329.000	974.60	Hot process
7	335.000	449.000	8.09	Cold process
8	389.000	380.000	1517.56	Hot process
9	345.000	371.000	112.27	Cold process
10	370.400	359.000	264.21	Hot process
11	389.000	380.000	3.56	Hot process
12	389.000	463.000	0.95	Cold process
13	515.000	338.000	16.95	Hot process
14	348.900	362.800	6.47	Cold process
15	395.200	338.000	5.56	Hot process
16	395.200	395.200	1249.00	Cold process
17	351.400	347.300	278.05	Hot process
18	364.400	364.400	107.00	Cold process
19	364.400	337.800	0.34	Hot process
20	364.400	477.000	1.70	Cold process
21	463.000	338.000	1.64	Hot process
22	411.000	338.000	3.12	Hot process
C	301.000	301.000	-	Cold utility

ΔTmin = 10.000K

Disallowed matches

Stream 7 cannot match with streams 13,21 and 22

Table 7.4 Data for the new unpinched problem

restricted match options, cannot be catered for at its upper end without a stream split. This is because the upper end of stream seven starts at the bottom of the highly constrained area (the top end of stream seven, at 379.0K, becomes 384.0K when corrected for ΔTmin). Indeed, the problem can be solved easily if the match restrictions are lifted. A further decrease of a few degrees in the upper-end temperature of stream seven makes the problem solvable. Application of the new methods to the problem with the upper end of stream seven set to 370.0K (see figure 7.67) resulted in the speedy discovery of a solution with fourteen shells. Many more solutions with fourteen shells were obtained and a typical network is shown in figure 7.68.

(Stream)	(T_{hot}/K)		(T_{cold}/K)	($\dot{m}Cp$/kW K^{-1}) (or $\dot{m}\lambda$/kW)
8	389.0	——→	380.0	1517.6
13	515.0	——→	338.0	16.95
21	463.0	——→	338.0	1.64
22	411.0	——→	338.0	3.12
1	357.2	←——	338.0	19.22
2	368.8	←——	368.8	8871.0
7	449.0	←——	335.0	8.09
9	371.0	←——	345.0	112.27
12	463.0	←——	389.0	0.95
14	362.8	←——	348.9	6.47
16	395.2	←——	395.2	1249.0
18	364.4	←——	364.4	107.00
20	477.0	←——	364.4	1.70

ΔT_{min} = 10.0K

Figure 7.64 The first grid diagram for the new unpinched problem

(Stream)	(T_{hot}/K)		(T_{cold}/K)	($\dot{m}Cp$/kW K^{-1}) (or $\dot{m}\lambda$/kW)
8	389.0	——→	380.0	1517.6
13	515.0	——→	338.0	16.95
21	463.0	——→	338.0	1.64
22	411.0	——→	338.0	3.12
1	357.2	←——	338.0	19.22
2	368.8	←——	368.8	8871.0
7	379.0	←——	335.0	8.09
9	371.0	←——	345.0	112.27
12	463.0	←——	389.0	0.95
14	362.8	←——	348.9	6.47
16	395.2	←——	395.2	1249.0
18	364.4	←——	364.4	107.00
20	477.0	←——	364.4	1.70

ΔT_{min} = 10.0K

Figure 7.65 The second grid diagram for the new unpinched problem

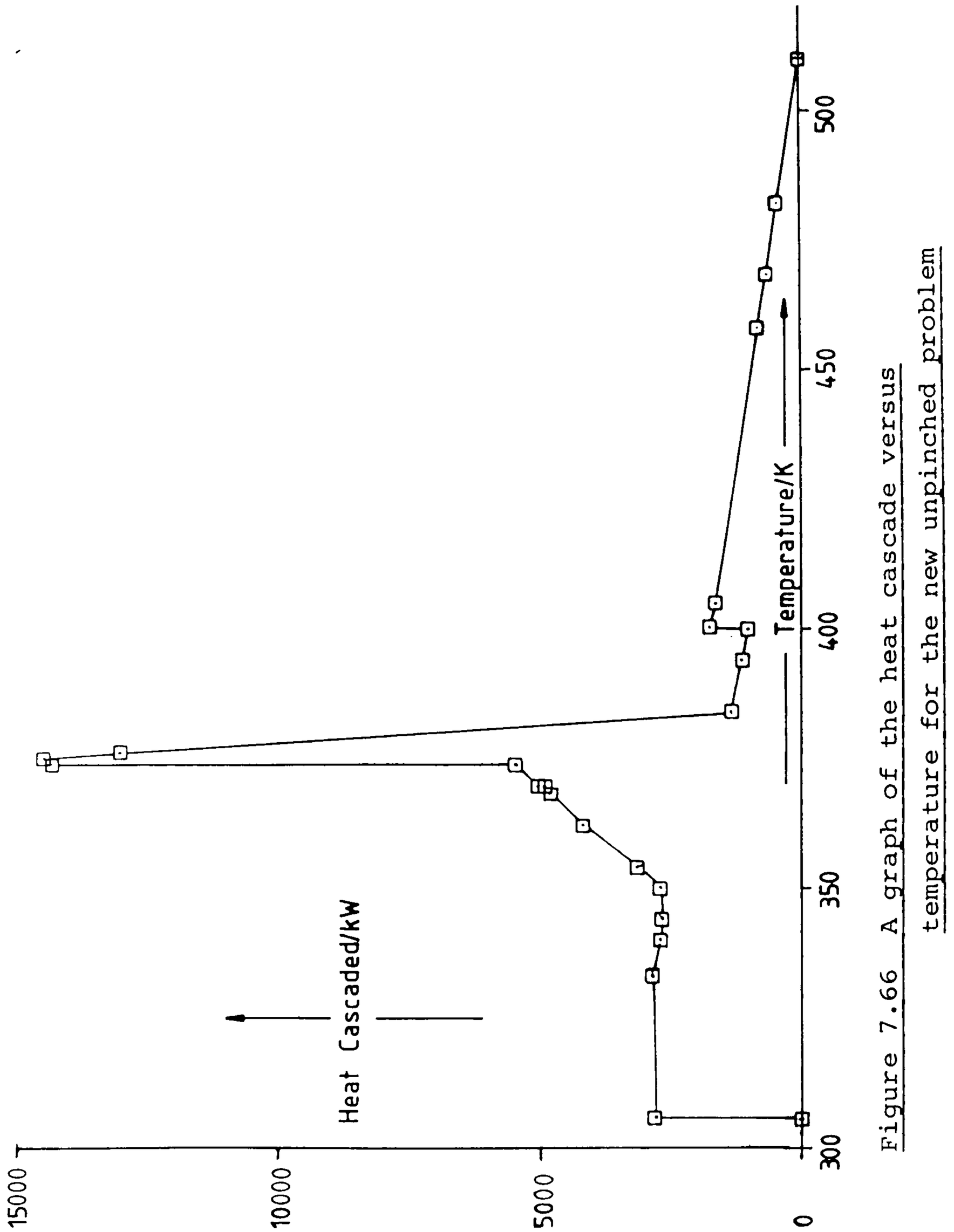
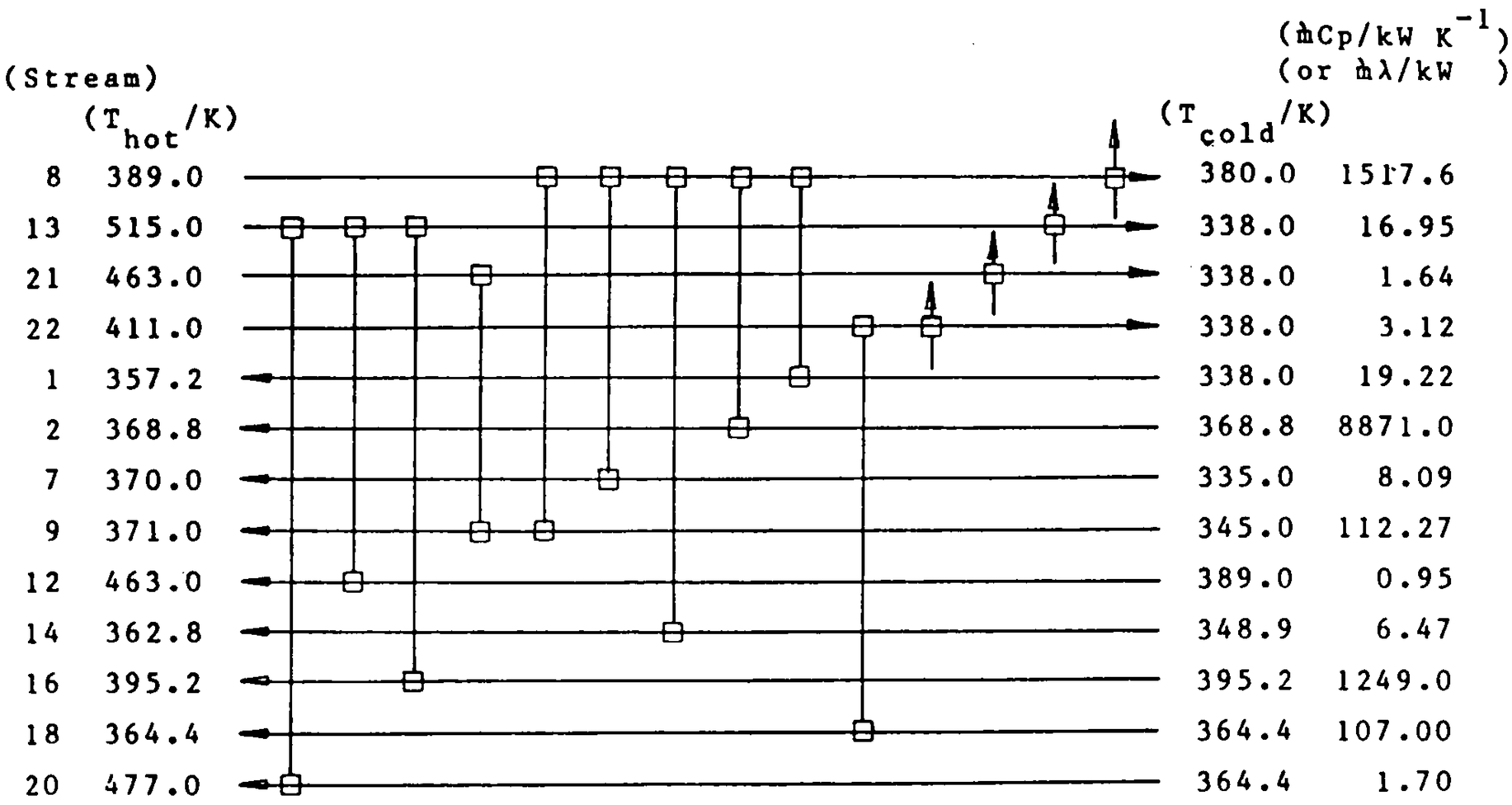


Figure 7.66 A graph of the heat cascade versus temperature for the new unpinched problem

(Stream)	(T_{hot}/K)		(T_{cold}/K)	($\dot{m}Cp$/kW K^{-1}) (or $\dot{m}\lambda$/kW)
8	389.0	——→	380.0	1517.6
13	515.0	——→	338.0	16.95
21	463.0	——→	338.0	1.64
22	411.0	——→	338.0	3.12
1	357.2	←——	338.0	19.22
2	368.8	←——	368.8	8871.0
7	370.0	←——	335.0	8.09
9	371.0	←——	345.0	112.27
12	463.0	←——	389.0	0.95
14	362.8	←——	348.9	6.47
16	395.2	←——	395.2	1249.0
18	364.4	←——	364.4	107.00
20	477.0	←——	364.4	1.70

ΔT_{min} = 10.0K

Figure 7.67 The third grid diagram for the new unpinched problem



Heat exchanger loads/kW

1 : 191.4	2 : 70.3	3 : 1249.0	4 : 136.5	5 : 2782.5
6 : 283.2	7 : 89.9	8 : 8871.0	9 : 369.0	10 : 107.0
11 : 120.8	12 : 68.5	13 : 1489.4	14 : 1262.6	

Figure 7.68 A fourteen-shell solution to the new unpinched problem

7.16 A NEW PINCHED PROBLEM

A new problem containing eleven process streams is the last problem to be considered. The relevant data are given in table 7.5. The construction of the temperature interval table for the root node shows that the problem is pinched and so both hot and cold utilities are required.

An analysis of the problem around the pinch point shows that a two- and a three-way split are required on the two cold streams immediately above the pinch point if maximum energy is to be recovered (see figure 7.69). The complexity of this arrangement was considered to be unacceptable and so streams ten and eleven, the sections of the two streams above the pinch with the smallest heat contents, were moved completely into the below-pinch sub-problem. The resulting two sub-problems are shown in figures 7.70 and 7.71. The constraints to be satisfied immediately adjacent to the pinch restrict the possible solutions in this area of the problem to only a few alternatives. If only one stream is to be split it must be stream one. The heat capacity flowrate of the larger branch of this stream is restricted to the range 10.35 to 11.95 by the requirements of the heat capacity flowrate ratio constraints. Thus a median value of 11.15 was chosen.

Above the pinch point the options are very few once the essential matches have been made. A typical solution, obtained by hand, is given in figure 7.72. Working from the right, the first three matches are essential, the fourth is added to make the mixing point on stream one approximately isothermal, two matches then follow that could conceivably be replaced by others and finally an essential utility heat exchanger is incorporated.

Below the pinch point the options are much more numerous.

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp$/kWK^{-1}	Stream type
1	284.000	719.166	19.16	Cold process
2	351.000	435.000	5.18	Cold process
3	470.000	490.000	18.10	Cold process
4	415.000	435.000	6.58	Cold process
5	631.000	320.000	10.35	Hot process
6	480.500	409.750	11.20	Hot process
7	520.200	453.300	11.95	Hot process
8	579.700	515.800	6.71	Hot process
9	480.500	320.000	5.81	Hot process
10	531.500	320.000	2.05	Hot process
11	607.500	320.000	1.57	Hot process
H	800.000	800.000	-	Hot utility
C	1.000	1.000	-	Cold utility

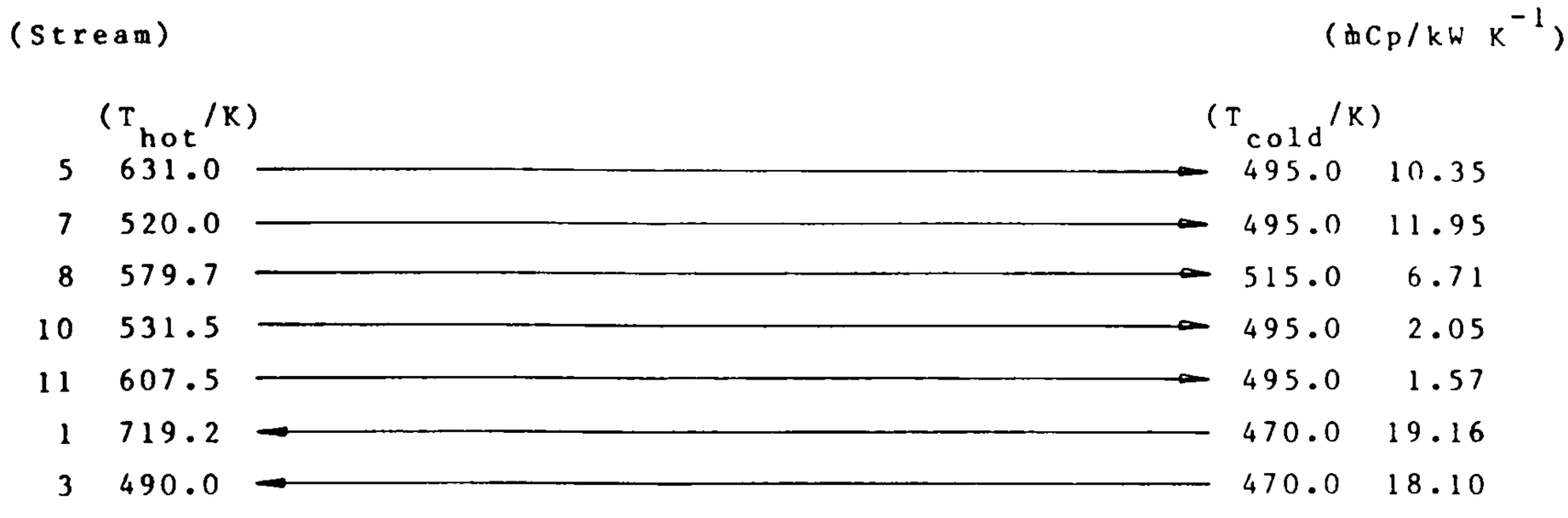
ΔTmin = 25.000K

Table 7.5 Data for the new pinched problem

Two essential matches are required between the two branches of stream one and streams five and seven (see figure 7.73). The stream split can be opened at this point, as in figure 7.73, or the exchangers doubled up, as in figure 7.74, before opening the stream split. Both of these alternatives were investigated.

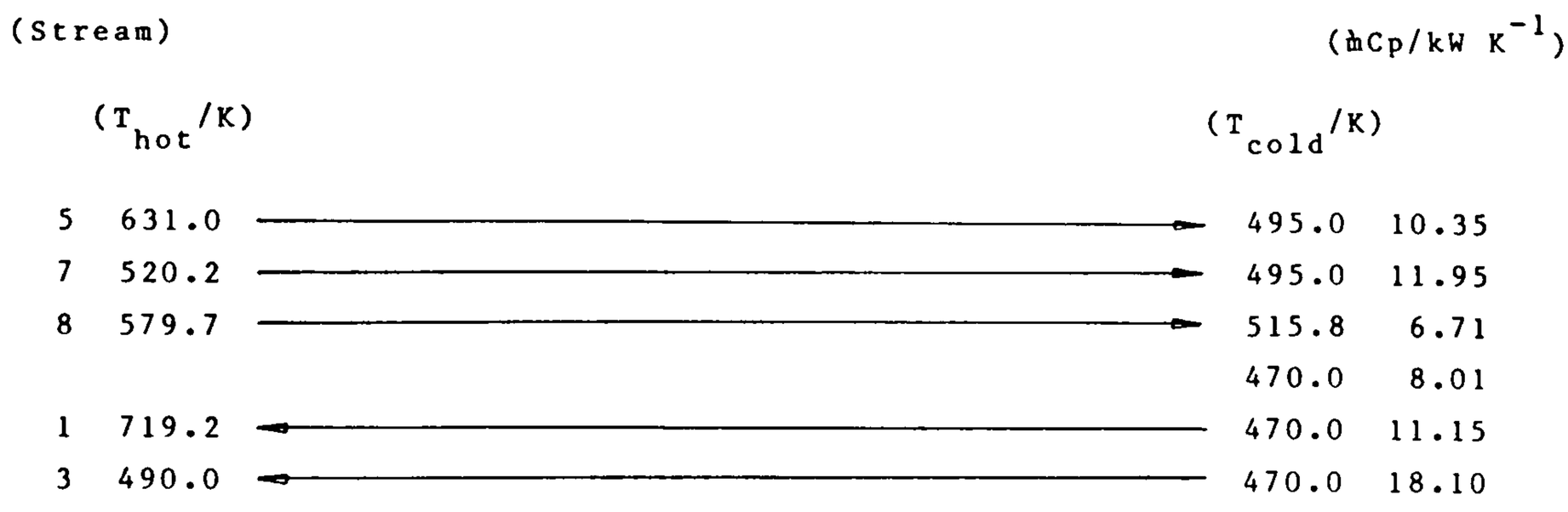
This new problem highlights the need for a deeper investigation of the methods for placement of stream splits. Stream splits are required where problems are pinched or the heat cascaded from higher temperatures is small but the range over which they should extend is not known. The use of non-counter-current exchangers exacerbates this problem because the number of likely positions that must be considered for the ends of stream splits is considerably greater.

The new method was applied to the remaining part of the problem and several networks obtained. An example of a solution to the first sub-problem is given in figure 7.75 and a solution to the second sub-problem in figure 7.76. The number of possible solutions is large and so doubling-up of exchangers was attempted (see section 6.3.1). An example solution using paired exchangers, containing one more shell than the previous network, is shown in figure 7.77.



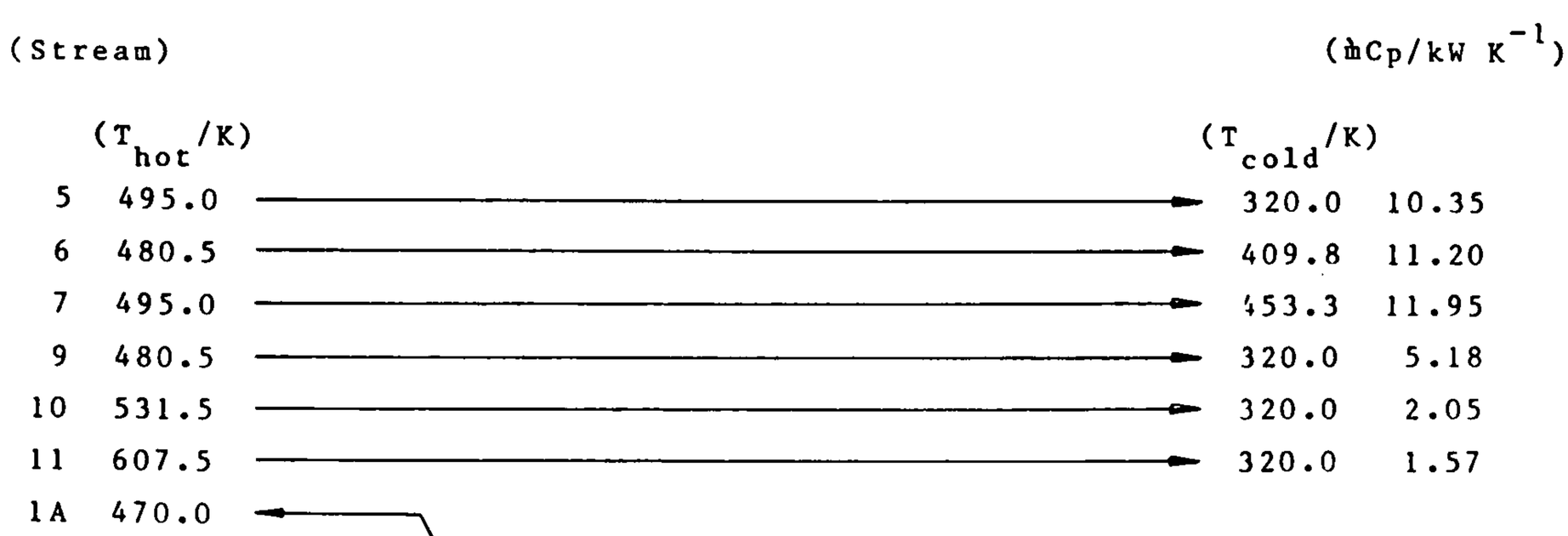
ΔT_{min} = 25.0K

Figure 7.69 The first grid diagram for the above-pinch section of the new pinched problem



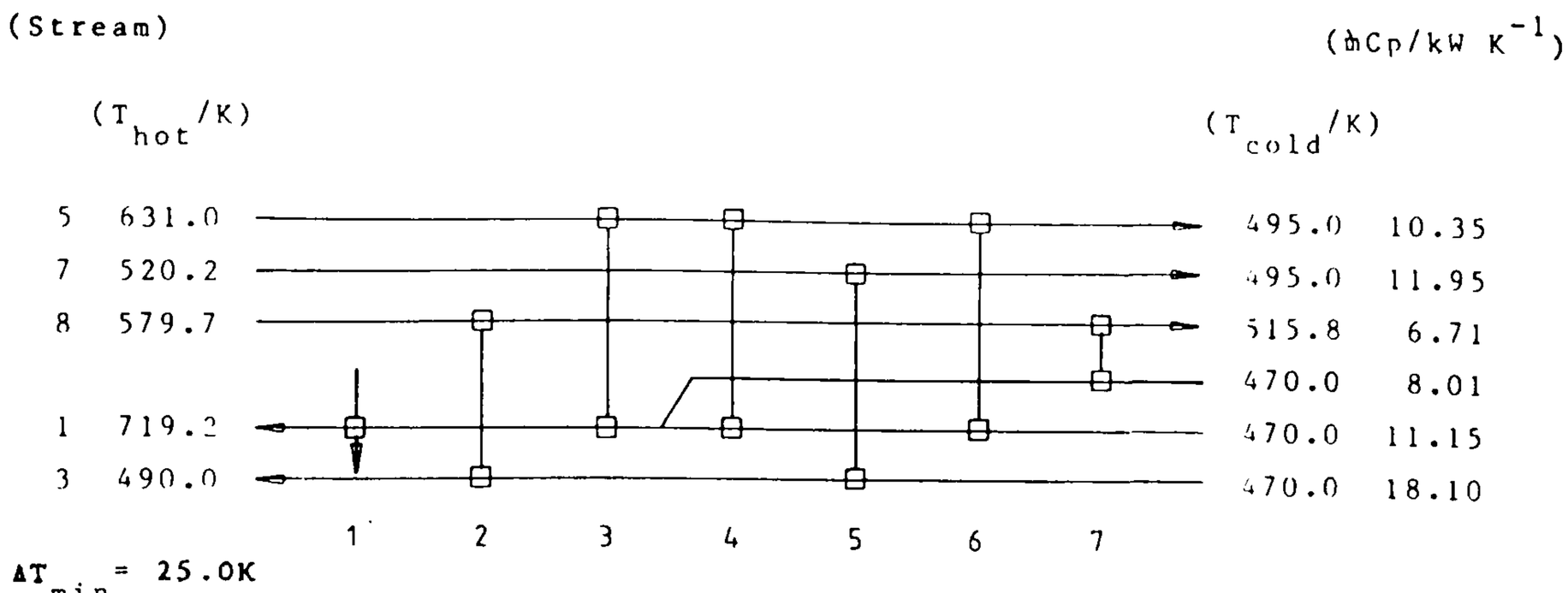
ΔT_{min} = 25.0K

Figure 7.70 The second grid diagram for the above-pinch section of the new pinched problem



ΔT_{min} = 25.0K

Figure 7.71 The grid diagram for the below-pinch section of the new pinched problem



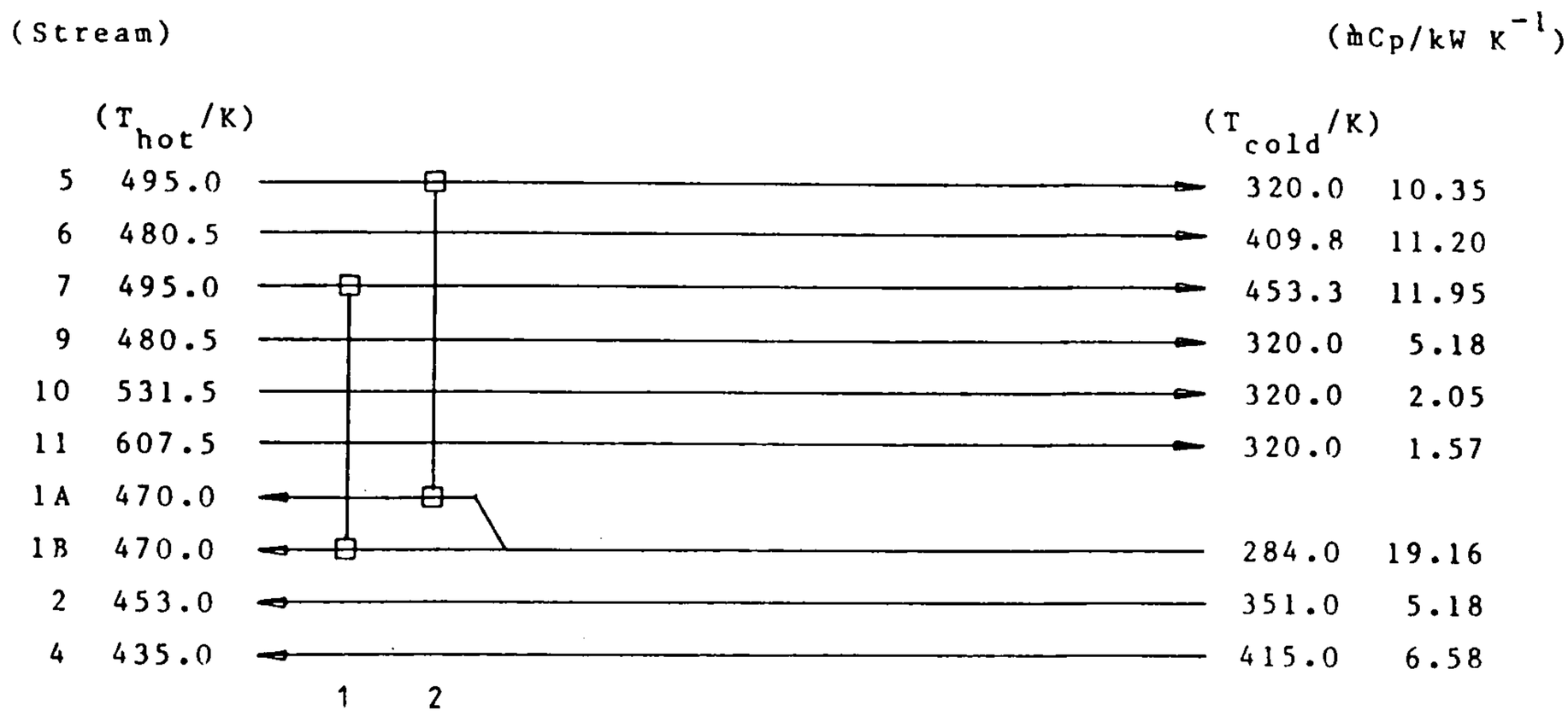
ΔT_{min} = 25.0K

Heat exchanger loads/kW

1 : 2998.5 2 : 60.9 3 : 828.6 4 : 300.3 5 : 301.1

6 : 278.8 7 : 367.9

Figure 7.72 A solution to the above-pinch section of the new pinched problem

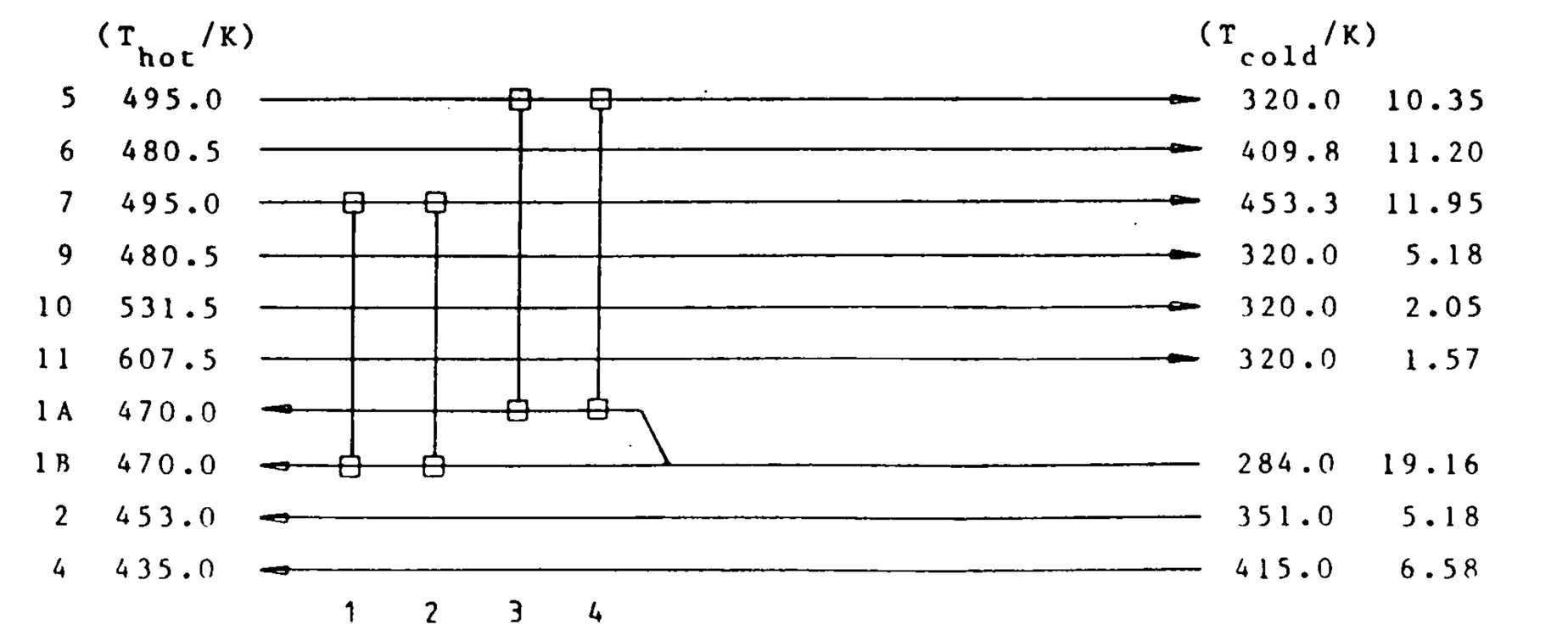


ΔT_{min} = 25.0K

Heat exchanger loads/kW

1 : 298.8 2 : 214.6

Figure 7.73 The first partial solution to the below-pinch section of the new pinched problem

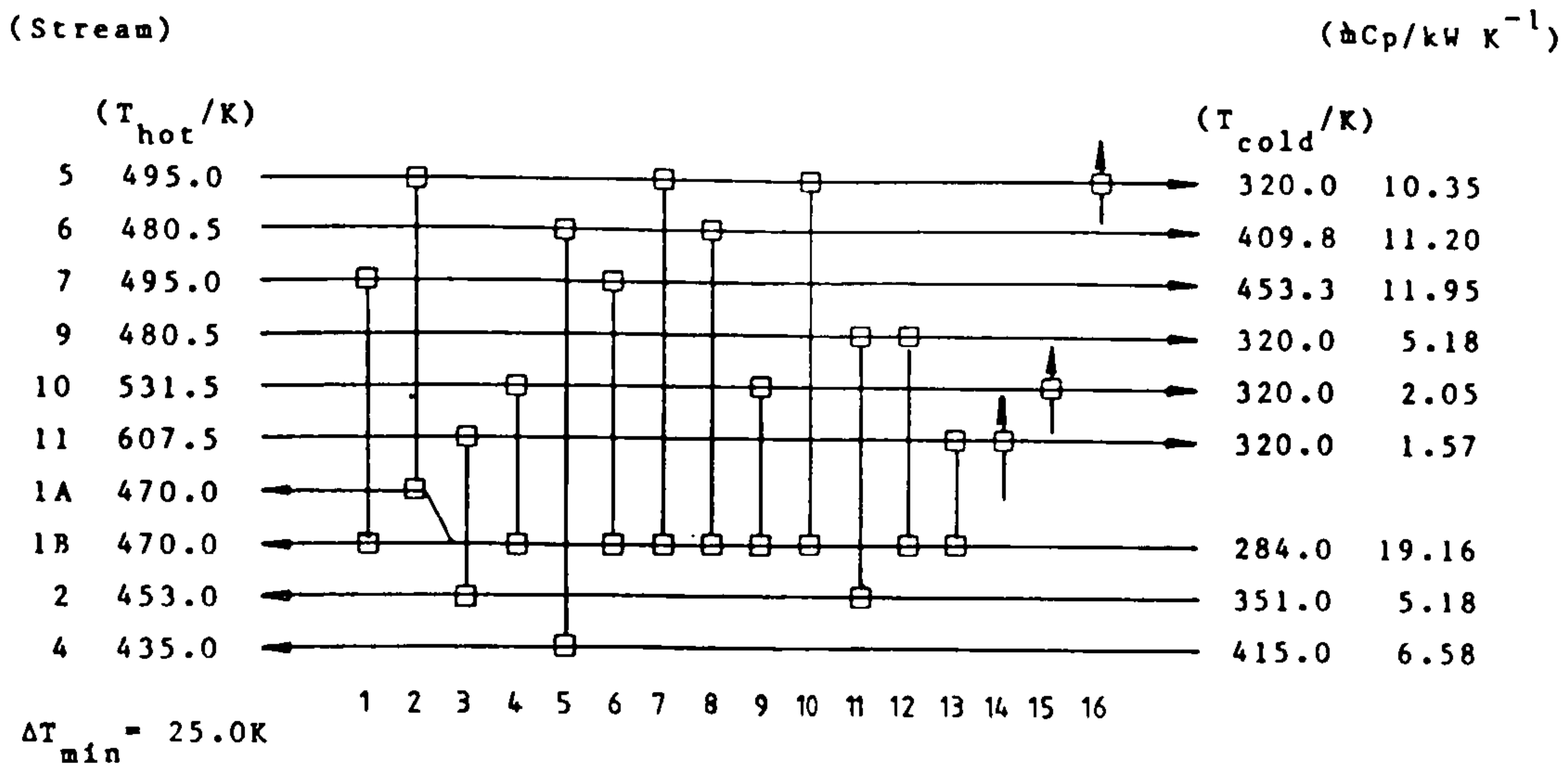


ΔT_{min} = 25.0K

Heat exchanger loads/kW

1 : 298.8 2 : 199.6 3 : 214.6 4 : 143.4

Figure 7.74 The second partial solution to the below-pinch section of the new pinched problem

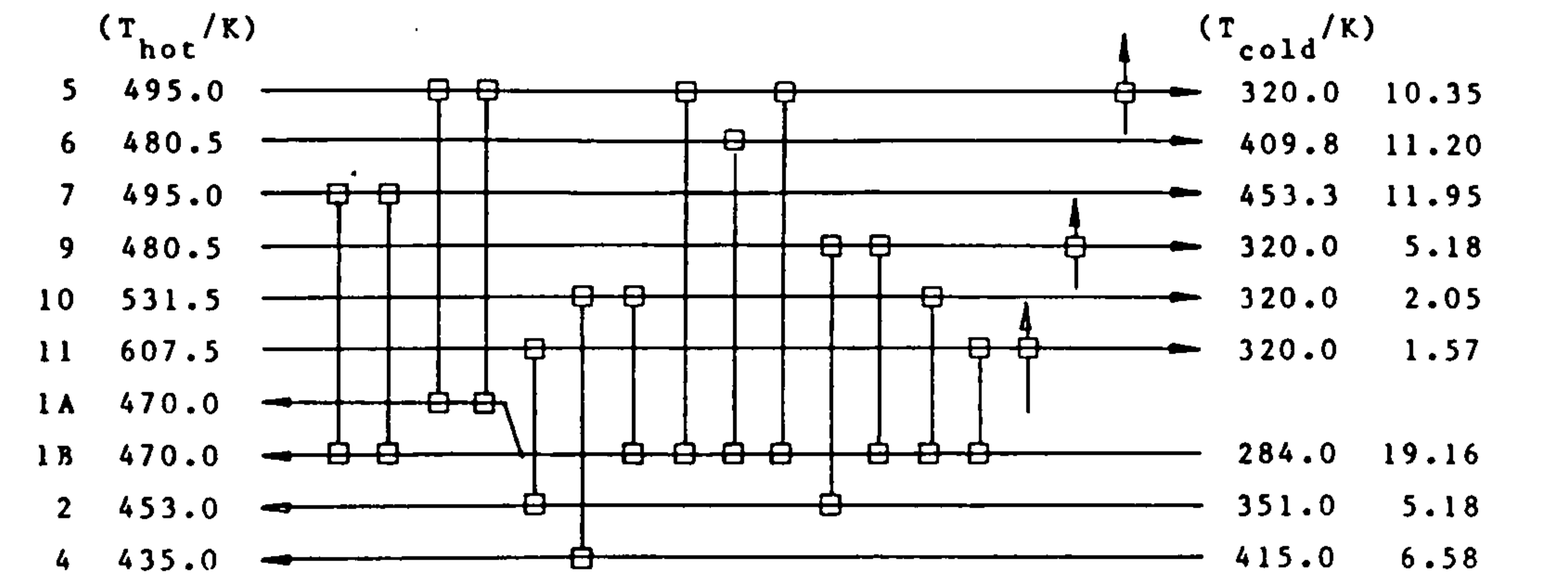


ΔT_{min} = 25.0K

Heat exchanger loads/kW

1 : 298.8	2 : 214.6	3 : 242.6	4 : 145.3	5 : 131.6
6 : 199.5	7 : 507.7	8 : 660.8	9 : 164.0	10 : 719.6
11 : 285.8	12 : 545.7	13 : 107.7	14 : 101.1	15 : 124.4
16 : 369.1				

Figure 7.75 A sixteen-shell sample solution to the below-pinch section of the new pinched problem

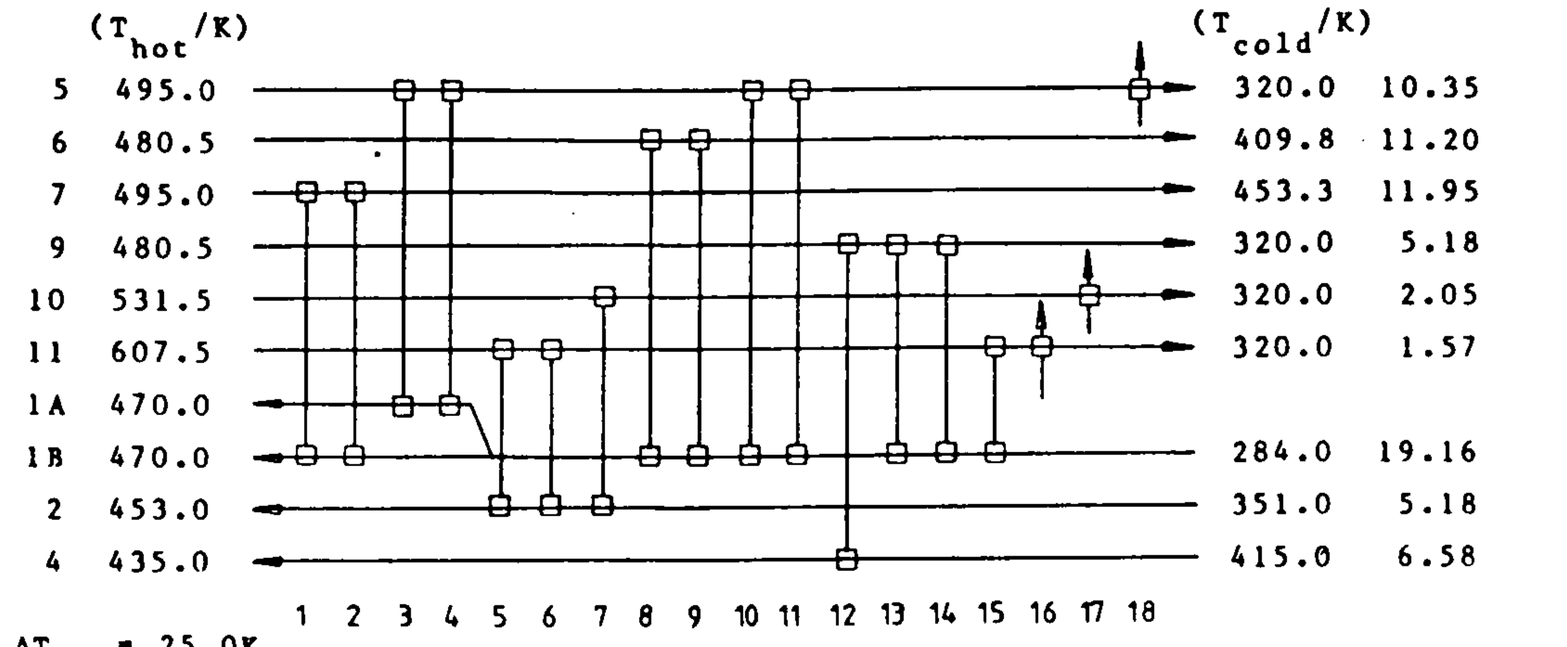


ΔT_{min} = 25.0K

Heat exchanger loads/kW

1 : 298.8	2 : 199.6	3 : 214.6	4 : 143.4	5 : 242.6
6 : 131.6	7 : 39.0	8 : 273.2	9 : 792.5	10 : 686.7
11 : 285.8	12 : 484.6	13 : 262.9	14 : 168.6	15 : 40.2
16 : 61.2	17 : 493.2			

Figure 7.76 A seventeen-shell sample solution to the below-pinch section of the new pinched problem



ΔT_{min} = 25.0K

Heat exchanger loads/kW

1 : 298.8	2 : 199.6	3 : 214.6	4 : 143.4	5 : 242.6
6 : 49.2	7 : 236.6	8 : 618.1	9 : 174.4	10 : 791.3
11 : 367.1	12 : 131.6	13 : 681.5	14 : 18.2	15 : 56.9
16 : 102.7	17 : 196.9	18 : 294.9		

Figure 7.77 An eighteen-shell sample solution to the below-pinch section of the new pinched problem

7.17 SUMMARY OF RESULTS FOR THE NEW PROBLEMS

The two new problems were successfully solved by the new methods although deficiencies were found. The program HERNE was used, as intended, as a method of screening alternative networks when the options become too numerous to consider manually. Thus the areas of problems which are hightly constrained, pinch points and 'pseudo-pinch' points, must still be dealt with by other means. These areas can normally be considered by hand or by using an interactive computer program.

The bounding algorithm used to determine the utility requirement does not account for disallowed matches and hence difficulties arise when forbidden matches preclude maximum energy recovery, as in the new unpinched problem. Although methods exist to determine utility usage under restricted match conditions these are major algorithms in their own right and not suitable for inclusion as part of a branch-and-bound tree search (see section 2.2.1).

The need for a procedure to determine the optimum range for a stream split was clearly demonstrated by the new pinched problem. Although solvable by multiple case studies, a more rigorous approach would be an improvement.

The new unpinched problem exhibited a feature which is not usually present in smaller problems. As the number of process streams increases the possibility of more than one pinch existing or 'pseudo-pinches' being present increases dramatically. In addition, there may be no clearly defined pinch point but a pinch area where the composite hot and cold curves are approximately parallel. The problems that these features create are offset somewhat by the self-correcting nature of synthesis with 1/2 heat exchangers. The prohibition of temperature crosses tends to

reduce the cascade of heat across pinches and 'pseudo-pinches' automatically.

The doubling-up of heat exchangers as a means of increasing the 'modularity' of networks was only partially successful. The present algorithm adds in an additional shell as long as the utility usage bound is not violated. This results in some matches removing too much cascaded heat and precluding a solution because, although a solution is theoretically possible, very little heat is left to cascade in the remaining problem. A better indication of the suitability of a second or subsequent shell in a heat exchanger train is required.

CHAPTER EIGHT

CONCLUSIONS AND FUTURE WORK

8.1 CONCLUSIONS

A set of tools to aid the synthesis of heat exchanger networks has been developed which has been applied successfully to a set of problems of academic and industrial importance.

Counter-current flow in all heat exchangers has been assumed by nearly every worker in the field of network synthesis. However, in many existing chemical plants the exchangers used are far from counter-current. Therefore a study has been made into the properties of non-counter-current heat exchangers and two procedures developed to aid their introduction into a synthesis algorithm.

An algorithm to determine the minimum number of stream splits required at problem pinch points, if maximum energy recovery is to be obtained, has been developed. A complete set of all the possible matching options is given for the streams immediately adjacent to the pinch point. Each of these partial networks can then developed, by the allocation of heat capacity flowrate to each stream branch, to allow suitable candidates to be chosen.

Once one or more suitable partial networks has been selected, the synthesis of the rest of the network can proceed. In the general case this development is carried out by hand or by interactive computer program until the number of options becomes too large to consider manually. Another computer program, based on a branch-and-bound tree

search, has been developed to continue the synthesis from this point to its completion.

This program incorporates non-counter-current heat exchangers on a sequential shell-by-shell basis with an aim of minimising the total number of shells. Complete enumeration of the search tree is avoided by the use of bounding algorithms which calculate the minimum number of shells and amount of utilities required to complete partial networks.

The use of an algorithm to produce networks shell-by-shell, rather than match-by-match, was shown to bring questionable improvements in cases where no stream splitting is required. The decrease in the exchanger cost is probably more than offset by the increase in the piping costs. However, in the cases where stream splitting is required, the number and range of stream splits can often be reduced by the use of the new methods.

The synthesis procedure requires only moderate development to produce an algorithm which would maintain the shell-by-shell approach and which would introduce in an effective manner exchanger trains similar to the match-by-match approach. In this way the more realistic costs calculated in this work would be retained.

Networks containing the same number of exchangers and recovering equal amounts of heat have been shown to have remarkably constant costs when 1/2 shell-and-tube heat exchangers are used. The use of cost as a method of screening networks is therefore discredited totally.

The solution of branch-and-bound tree searches was shown to be improved greatly by the use of the information for one level of the search tree in the next level down. This

applies to both branching strategies and to bounding algorithms. Computationally expensive sorting is then centred on the root node and hence occurs only once.

8.2 FUTURE WORK

The need for further work on the incorporation of stream splits into networks has been clearly shown. In particular, a procedure to isolate the optimum range is required.

A better indication of the suitability of second or subsequent shells in a heat exchanger train is required. The development of an algorithm to do this would result in a dramatic decrease in the problem tree size.

Additional work on the utility bounding algorithm to cope, to some extent, with disallowed matches would be beneficial. Linear programming solutions appear to be precluded because of their computational complexity.

The occurrence of 'pseudo-pinches' and pinch areas and the need for stream splits around these features requires further investigation.

The stream-splitting algorithm, described in chapter five, needs amendment in order to handle problems in which maximum energy recovery is not achievable because of disallowed matches or stream-splitting restrictions.

APPENDICES

APPENDIX A THE HEAT EXCHANGER NETWORK SYNTHESIS PROBLEM FORMULATED AS AN MINLP

The synthesis of heat exchanger networks can be formulated as an MINLP in the following way:

A set of hot process streams of known inlet and outlet temperatures and heat capacity flowrates are to be matched with a set of cold process streams of similar known properties. A set of hot and cold utility streams are also available at specified conditions and cost.

The objective is to synthesize the lowest cost heat exchanger network which satisfies the above process streams using heat exchangers of a specified type. The cost, C, of any network created is given by:

$$C = \sum_{i=1}^{Np} dc(A^i)^a + \sum_{j=1}^{Nhu} \left[tu_{hu}\, \dot{m}_{hu}^{j} + dc(A^j)^a\right] + \sum_{k=1}^{Ncu} \left[tu_{cu}\, \dot{m}_{cu}^{k} + dc(A^k)^a\right]$$

where Np is the number of process heat exchangers,
Nhu is the number of hot utility heat exchangers,
Ncu is the number of cold utility heat exchangers,
d is the rate of return,
c and a are the heat exchanger cost parameters,
A is the heat transfer surface area of an exchanger,
t is the operating time per year,
u is the unit cost of a utility and

$\dot{m}$ is the mass flowrate of a utility.

This synthesis involves the determination of the configuration (i.e. the integer variables) and the design parameters and state variables (i.e. the continuous variables). Clearly this is an MINLP of considerable magnitude.

APPENDIX B AN EXPRESSION FOR THE MAXIMUM VALUE OF TEMPERATURE EFFICIENCY GIVEN THE HEAT CAPACITY FLOWRATE RATIO

Equation (3.10a) can be manipulated by using equations (3.2), (3.4) and (3.6a) to (3.8a) or (3.6b) to (3.8b) to give an expression for NTU:

$$NTU = \frac{1}{\sqrt{R^2+1}} \ln \left[\frac{2-P(R+1-\sqrt{R^2+1})}{2-P(R+1+\sqrt{R^2+1})} \right] \qquad (B.1)$$

(valid for R equal to unity)

The maximum value of P, Pmax, must occur when the surface area and hence NTU becomes infinite, thus:

$$2-Pmax\ (R+1+\sqrt{R^2+1}) = 0$$

$$\therefore \quad Pmax = 2/(R+1+\sqrt{R^2+1}) \qquad (B.2)$$

At this limit F becomes zero.

APPENDIX C THE MINIMUM VALUE FOR 'P/Pmax' FOR FIXED VALUES OF F

The log-mean temperature difference correction factor, F, is given by equation (3.10a):

$$F = \frac{\sqrt{R^2+1}}{(1-R)} \cdot \frac{\ln\ (1-PR)/(1-P)}{\ln\left[\dfrac{2-P(R+1-\sqrt{R^2+1})}{2-P(R+1+\sqrt{R^2+1})}\right]}$$

Let S = P/Pmax and substitute for P giving:

$$F = \frac{\sqrt{R^2+1}}{(1-R)} \cdot \frac{\ln\ (1-RSPmax)/(1-SPmax)}{\ln\left[\dfrac{2-SPmax(R+1-\sqrt{R^2+1})}{2-SPmax(R+1+\sqrt{R^2+1})}\right]}$$

Now, from equation (3.11):

$$Pmax = 2/(R+1+\sqrt{R^2+1})$$

$$\therefore \quad F = \frac{\sqrt{R^2+1}}{(1-R)} \cdot \frac{\ln\ (R+1+\sqrt{R^2+1}-2SR)/(R+1+\sqrt{R^2+1}-2S)}{\ln\left[\dfrac{R+1+\sqrt{R^2+1}-S(R+1-\sqrt{R^2+1})}{R+1+\sqrt{R^2+1}-S(R+1+\sqrt{R^2+1})}\right]}$$

Differentiating w.r.t. R with constant F leads to:

$$\frac{dS}{dR} = \frac{D-E-S(1-R)F/A}{G-(R^2+1)F/(1-S)} \qquad (C.1)$$

where

$$A = R+1+\sqrt{R^2+1}$$

$$B = R+1-\sqrt{R^2+1}$$

$$C = R/\sqrt{R^2+1}$$

$$D = \frac{2S(R^2+1)\left[(1+C)(1-R)+A-2S\right]}{(A-2S)(A-2SR)}$$

$$E = \frac{(1+R)}{(1-R)} \ln\left[\frac{A-2SR}{A-2S}\right]$$

$$F = \frac{2\sqrt{R^2+1}}{(A-SB)}\left[\frac{\ln\left[(A-2SR)/(A-2S)\right]}{\ln\left[(A-SB)/(A-SA)\right]}\right]$$

$$G = \frac{2A(R^2+1)(1-R)}{(A-2S)(A-2SR)}$$

This expression can be shown to have a form of symmetry about R equal to unity. Every point given by R=X, where X is some feasible value, has the same value of R(dS/dR) as R=1/X except for the sign, i.e.:

$$R\left[\frac{dS}{dR}\right]_{R=X} = -R\left[\frac{dS}{dR}\right]_{R=1/X} \qquad (C.2)$$

It follows directly that $\left[\frac{dS}{dR}\right]_{R=1}$ is zero and hence S has an extremum at R equal to unity. Some form of symmetry is to be expected because the original equation for F applies equally well if R is defined as the hot heat capacity flowrate divided by the cold or vice versa.

Substitution of numerical values into equation (C.1) shows that when R is equal to unity dS/dR is indeed a minimum

and not a maximum. Again, this is to be expected because as R tends to zero or positive infinity counter-current flow is approximated and thus P/Pmax must tend to a maximum.

APPENDIX D THE GRAPHS PLOTTING 'P/Pmax' AGAINST R SUGGESTING MONOTONICITY

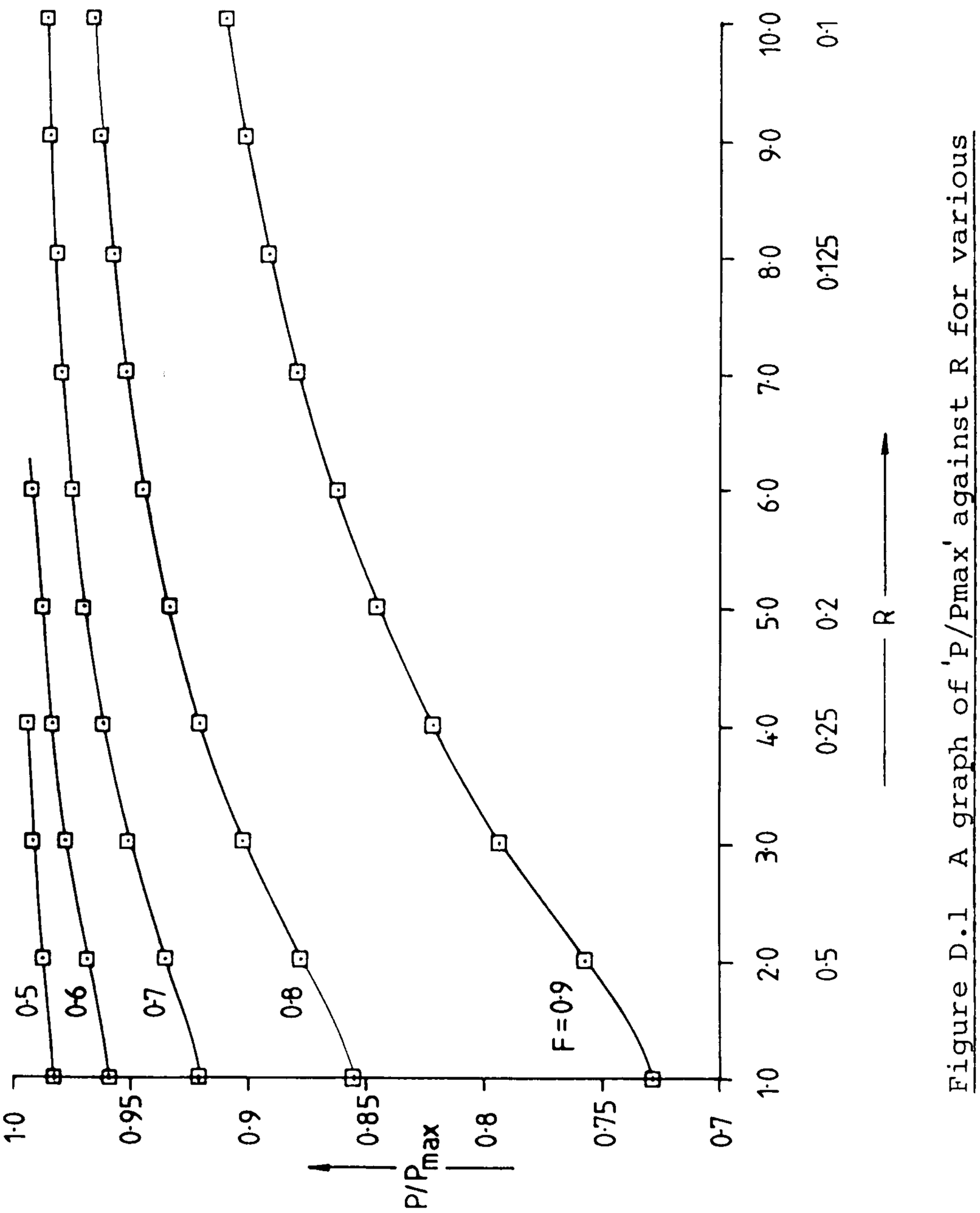


Figure D.1 A graph of 'P/Pmax' against R for various values of F

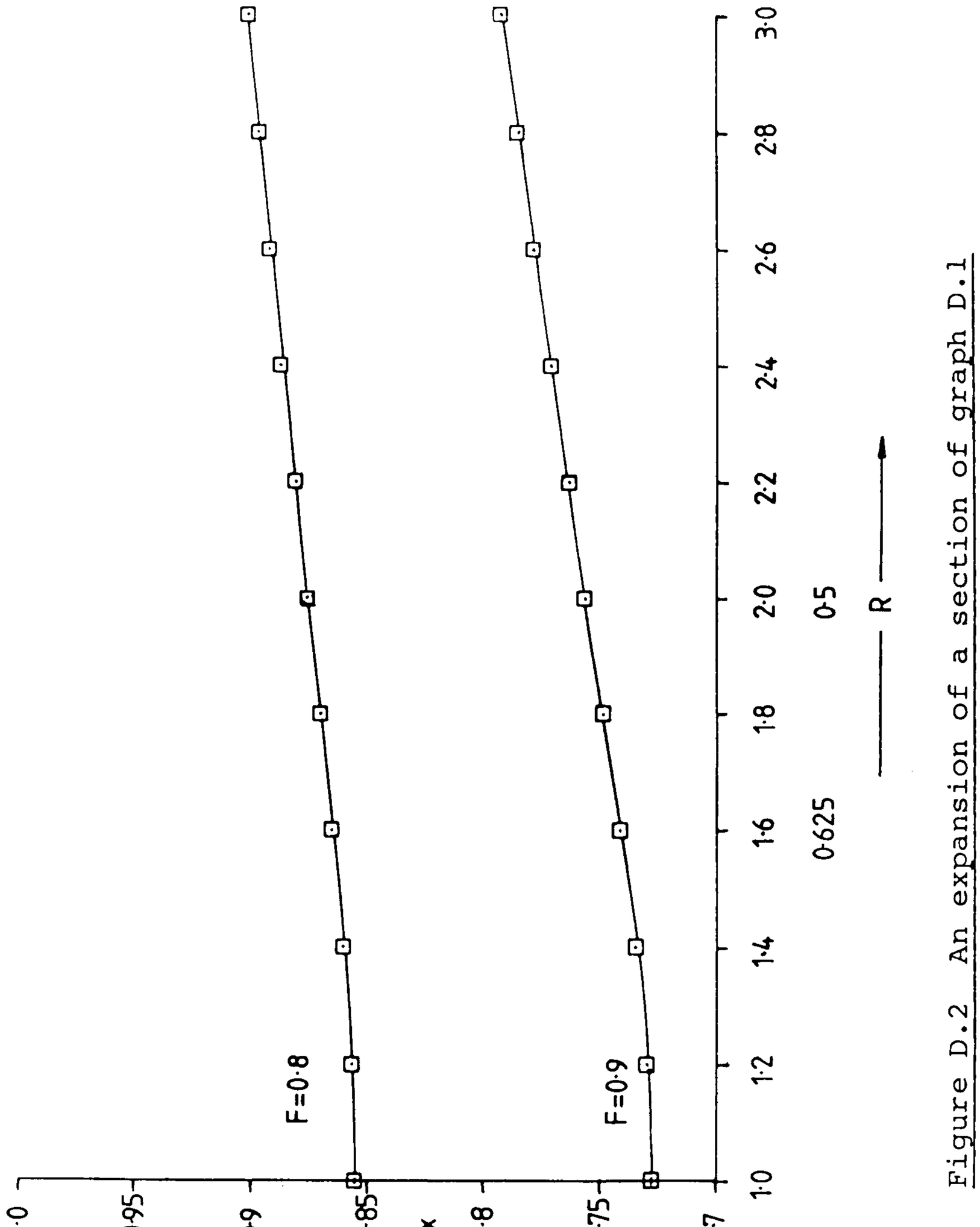


Figure D.2 An expansion of a section of graph D.1

APPENDIX E A PROGRAM TO CALCULATE TEMPERATURE EFFICIENCY (P) GIVEN THE LMTD CORRECTION FACTOR (F) AND HEAT CAPACITY FLOWRATE RATIO (R)

This FORTRAN 77 program consists of just one main routine which calls the sub-program C05AZF from the NAG FORTRAN library. This in turn calls F01AAF, X02AAF and X02ABF from the same library. Full details on all these sub-programs can be obtained from the NAG manuals (NAG 1983).

The program initially requests the value of F for which the values of P will be required and the accuracy to which P should be calculated. The value of P/Pmax is then calculated to a high accuracy for R equal to unity, starting with the bounds P equal to Pmax and P equal to zero (i.e. the entered accuracy is not used for this calculation). This ratio, $P/Pmax|_{R=1}$, is then used to give initial bounds on the value of P for subsequent values of R entered by the user. These calculations are carried out to the accuracy entered by the user. The program is terminated by entering a zero-length line in response to the prompt for a new value of R. See microfiche.

APPENDIX F A PROGRAM TO CALCULATE TEMPERATURE GAP RATIO (G) GIVEN THE LMTD CORRECTION FACTOR (F) AND HEAT CAPACITY FLOWRATE RATIO (R)

This FORTRAN 77 program is very similar to the program to calculate P given above. In this case, the equivalent ratio to $P/Pmax|_{R=1}$ is $(1-G)/(1-Gmin)|_{R=1}$. See microfiche.

APPENDIX G THE MINIMUM NUMBER OF BRANCHES VERSUS THE MINIMUM NUMBER OF EXCHANGERS

These two objective functions are similar but not identical. The differences occur in solutions where complete streams bypass and column streams are split simultaneously. In some cases a match could be placed on the bypass stream to remove a branch on another. This solution has the same number of exchangers but one less branch. This substitution method works satisfactorily when all matches are allowed but can cause problems when matches are disallowed or branches limited. In this case it may not be possible to discard solutions with bypasses with the optimum objective function value because these may be the only solutions. The occurance of this is rare, so the simplified formulation is given below:

$$\min_{X^{ij}} F = \sum_{i=1}^{N_h} \sum_{j=1}^{N_c} X^{ij}$$

Above pinch:

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c} \dot{m}Cp_h^{ij} \qquad i = 1 \ldots N_h+1$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h+1} \dot{m}Cp_h^{ij} \qquad j = 1 \ldots N_c$$

where

$$\dot{m}Cp_h^{N_h+1} = \sum_{j=1}^{N_c} \dot{m}Cp_c^j - \sum_{i=1}^{N_h} \dot{m}Cp_h^i$$

and

$$X^{ij} = 0 \text{ if } \dot{m}Cp_h^{ij} = 0$$

$$X^{ij} = 1 \text{ if } \dot{m}Cp_h^{ij} > 0$$

Below pinch:

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c+1} \dot{m}Cp_c^{ij} \qquad i = 1 \ldots N_h$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h} \dot{m}Cp_c^{ij} \qquad j = i \ldots N_c+1$$

where

$$\dot{m}Cp_c^{N_c+1} = \sum_{i=1}^{N_h} \dot{m}Cp_h^i - \sum_{j=1}^{N_c} \dot{m}Cp_c^j$$

and

$$X^{ij} = 0 \text{ if } \dot{m}Cp_c^{ij} = 0$$

$$X^{ij} = 1 \text{ if } \dot{m}Cp_c^{ij} > 0$$

In this formulation the variable Y does not occur.

APPENDIX H THE PROBLEM FORMULATION BBMIP IS DESIGNED TO SOLVE

BBMIP (Branch and Bound Mixed Integer Programme) is de-

signed to solve the mixed integer programme given by:

Minimise: $z = c^T x + \bar{c}^T y$

Subject to:

$$Ax + \bar{A}y \begin{bmatrix} \leqslant \\ = \\ \geqslant \end{bmatrix} b$$

$0 \leqslant x \leqslant U$ Integral valued

$0 \leqslant y \leqslant \bar{U}$

where

n is the total number of variables

n_1 is the number of integer variables

m is the number of constraints

c, x, U are vectors of n_1 components

$\bar{c}$, y, $\bar{U}$ are vectors of $n - n_1$ components

b is a vector of m components

A is an m. n_1 matrix

$\bar{A}$ is an m.$(n - n_1)$ matrix

The method is based on that of Land and Doig (1960).

In this case $\overline{c}$ is a null matrix, U is a unit vector and $\overline{U}$ is a vector filled with the largest heat capacity flowrates of the row streams.

APPENDIX I THE TEST PROBLEM DATA AND SOLUTIONS

Stream Number	Hot stream heat capacity flowrates (MW/K)	Cold stream heat capacity flowrates (MW/K)	
1	0.70	1.20	Problem 1
2	0.30	0.20	
1	0.50	0.40	Problem 2
2	0.10	0.30	
1	0.30	0.60	Problem 3
2	0.25	0.50	
3	0.20	-	
4	0.05	-	
1	0.2062	0.2000	Problem 4
2	0.1098	0.1094	
3	0.0739	0.0698	
4	-	0.0618	

Table I.1 The test problem data

Problems 1, 2, 3a and 4a have no restriction on the number of ways a stream can be split.

Problems 3b and 4b have a restriction of two imposed on the number of ways a stream can be split.

Problem number	BBMIP performance		New method performance	
	Number of solutions found	CPU time (s)	Number of solutions found	CPU time[+*] (s)
1	1	0.524	1	0.170
2	1	0.430	2	0.160
3a	3	1.535	9	0.350
3b	2	1.141	5	0.250
4a	2	8.909	8	0.540
4b	2	7.152	6	0.380

Table I.2 The test problem solutions

[+]CDC Cyber 170/720

[*]The reading in and printing out of data carries with it an overhead of approximately 0.15 s which is significant in these smaller problems. A comparative figure for BBMIP is not known.

APPENDIX J THE FIXED-COST TRANSPORTATION PROBLEM AND LOOPS IN THE LP TABLEAU

Consider an LP tableau which contains a problem solution with loops. The contribution of this loop to the objec-

tive function is simply the number of elements in the loop. Unless secondary upper-bound constraints are imposed on elements in the loop, it is always possible to distribute the contents of one of the elements of the loop thoughout the other members. No other elements in the tableau are involved in any way. For example:

Destination →		1	2	3
Source ↓		12	5	2
1	11	9	1	1
2	7	3	4	
3	1			1

Figure J.1a

becomes:

	12	5	2
11	10		1
7	2	5	
1			1

Figure J.1b

Figure J.1 Removal of a loop from an LP tableau

by transferring 1 unit around the loop. The value of the objective function drops from 6 to 5 during this process. Obviously, if secondary upper-bound constraints are imposed, the situation may be different. For example, if, in the above example, an upper limit of nine is placed on the transportation of 'goods' from source one to destination one the solution given without a loop becomes infeasible. (If that was the only upper-bound constraint imposed, the loop could still be removed by transferring 4 units around the loop in the opposite direction to that shown above, i.e. to give:

	12	5	2
11	5	5	1
7	7		
1			1

Figure J.2 Another way to remove a loop from an LP tableau

However, no general conclusions can be drawn concerning the existance of loops in problems with upper-bound constraints.)

APPENDIX K COMPASS - A COMPUTER PROGRAM TO GENERATE HEAT EXCHANGER CONFIGURATIONS WITH THE MINIMUM NUMBER OF PARALLEL STREAM SPLITS

This program is composed of seven FORTRAN 77 sub-programs (ASSIGN, BLOCKS, BOUNDS, COMPAS, FLAGER, OPENER and ORDERS) and a machine dependent sub-program (SECOND).

The main sub-program, COMPAS, controls the enumeration of the search tree and reports on any solution found. It issues calls to ASSIGN, BOUNDS, FLAGER, OPENER and SECOND in the process.

Sub-program SECOND is used to time the program and is not essential. The two calls to this sub-program can be removed if desired.

Sub-program OPENER opens the necessary logical units for input and output, reads and checks the input data and sets up the information for the search-tree root node. Sub-program ORDERS is called by OPENER.

Sub-program FLAGER determines the amount of heat capacity

flowrate used at each node and hence which rows and columns are to be deleted.

Sub-program ASSIGN takes the information for one node and modifies it to form the data for the nodes which form the next level down in the search tree. The information involving the row and column matched at the current node is updated which leads to a correction of the ordered lists.

Sub-program BOUNDS calculates the lower bound on the number of further branches required to complete a partial design.

Sub-program BLOCKS initialises the program common blocks.

Sub-program ORDERS creates a list of pointers which allows another array to be sorted into descending order.

Full details on all the variables stored in the common blocks and all the key local variables are given in the program listing. Compilation with an optimising compiler is recommended due to the extensive use of two dimensional arrays. See microfiche.

APPENDIX L AN EXAMPLE OF THE APPLICATION OF THE HEAT EXCHANGER NETWORK SYNTHESIS BOUNDING ALGORITHM

Introduction

A worked example is set out below to demonstrate the utility usage bound amendment procedure. The problem, known conventionally as 5SP1, was introduced by Masso and Rudd (1969) and a summary of the relevant data is given in table L.1. The development of a complete solution by sequentially adding heat exchangers is monitored by calculating the relevant utility bound at each stage. Feasibility is thus shown to be maintained throughout the process.

It is convenient to include the utility streams as well as the process streams in the ordered lists and then check that the extra utility usage at each level remains at zero to ensure feasibility. This automatically guarantees that the temperature level as well as the amount of utility is sufficient to complete the network.

A minor problem occurs when calculations are carried out with numbers of finite precision. The extra utility usage indicated by the bounding procedure cannot be guaranteed to be equal exactly to zero and a suitable tolerance must be applied when deciding if a match is feasible.

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp$/kWK^{-1}	Stream type
1	310.928	477.594	11.387	Cold process
2	522.039	394.261	16.606	Hot process
3	338.706	455.372	12.916	Cold process
4	477.594	338.706	13.285	Hot process
5	366.483	477.594	13.021	Cold process
H	508.860	508.860	-	Hot utility

ΔT_{min} = 11.111K

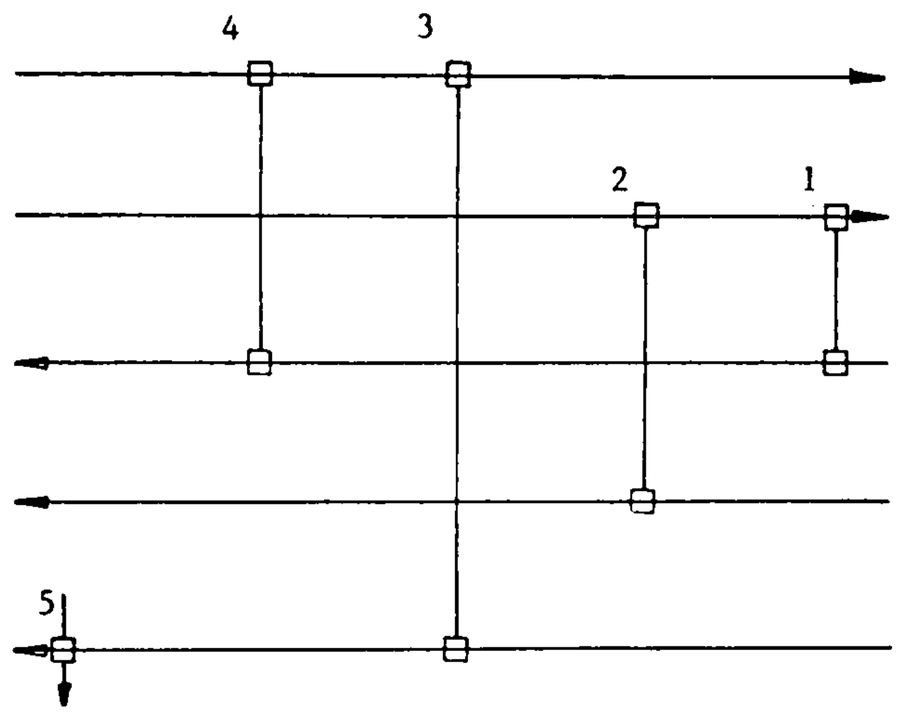
Table L.1 The stream data for problem 5SP1

The development of the network

A bounding procedure does not determine how a search tree is to be expanded but indicates whether a partial solution is suitable for further expansion. Thus the network used here to demonstrate the procedure was obtained by other means. The relevant network is shown in figure L.1 and described in the next section. In a branch-and-bound tree-searching method the streams to be matched are selected by the branching strategy and the amount of heat to be transferred is fixed by the match generator.

The 5SP1 problem is unpinched and so only one utility stream, a hot utility, is required. The bounding procedure can be applied equally well to pinched problems.

(Stream)	(T_{hot}/K)		(T_{cold}/K)	($\dot{m}Cp$/kW K^{-1})
2	522.039		394.261	16.606
4	477.594		338.706	13.285
1	477.594		310.928	11.387
3	455.372		338.706	12.916
5	477.594		366.483	13.021



(Heat Load/kW) 884.48 1559.6 562.32 1506.8 338.27

ΔT_{min} = 11.111K

Figure L.1 A pictorial representation of the typical solution to problem 5SP1 used in this example

The typical solution to problem 5SP1 used in this example

Match 1

Stream 4 is matched with stream 1 to transfer 338.27kW in the following ranges:

Stream 1	340.635K	to 310.928K	
Stream 4	364.169K	to 338.706K	

Match 2

Stream 4 is matched with stream 3 to transfer 1506.8kW in the following ranges:

Stream 3	455.372K	to 338.706K	(stream complete)
Stream 4	477.594K	to 364.169K	(stream complete)

Match 3

Stream 2 is matched with stream 5 to transfer 562.32kW in the following ranges:

Stream 5	409.668K	to 366.483K
Stream 2	428.124K	to 394.261K

Match 4

Stream 2 is matched with stream 1 to transfer 1559.6kW in the following ranges:

Stream 1	477.594K	to 340.635K	(stream complete)
Stream 2	522.039K	to 428.124K	(stream complete)

Match 5

The hot utility stream is matched with stream 5 to transfer 884.48kW in the following range:

Stream 5 477.594K to 409.668K (stream complete)

Problem initialisation

Before the development of the network can take place the temperature interval table must be generated for the root node (see table L.2). This is done by sorting all the temperature interval delimiters into order after a correction for the minimum temperature difference (ΔTmin/2 is subtracted from all the hot stream temperatures and Δ Tmin/2 is added to all the cold stream temperatures). The heat content for each temperature interval is then determined by calculating the net heat capacity flowrate in the interval and multiplying by the temperature difference across the interval. The heat content of any interval that exhibits a temperature difference of zero is set equal to the relevant latent heat flowrate, if this exists, rather than zero. A running total is then created of the interval heat contents from the hot to the cold end of the table to enable the utility requirements and pinch point position, if any, to be determined.

Delimiter	Temperature K	Stream	Temperature Difference/K	$\dot{m}\lambda$/kW or $\dot{m}Cp$/kWK^{-1}	Heat Content/kW	$\sum$Heat Contents/kW
1	316.483	1	16.667	-11.387	189.79	-0.01
2	333.150	4	11.111	1.898	21.09	189.78
3	344.261	3	27.778	-11.018	-306.06	168.69
4	372.039	5	16.667	-24.039	-400.66	474.75
5	388.706	2	72.222	-7.433	-536.83	875.41
6	460.928	3	11.111	5.483	60.92	1412.24
7	472.039	4	11.111	-7.802	-86.69	1351.32
8	483.150	5	0.000	5.219	0.00	1438.01
9	483.150	1	20.154	16.606	334.68	1438.01
10	503.304	H	0.000	884.48	884.48	1103.33
11	503.304	H	13.179	16.606	218.85	218.85
12	516.483	2				

Table L.2 The ordered list before any matches are made

Amendments to account for match one

The first new delimiters are 346.191K for stream 1 and 358.613K for stream 4 (i.e. the new stream ends altered by Δ Tmin/2). Considering each old delimiter in turn, denoting heat capacity flowrate of stream n by $\dot{m}Cp_n$:

1. Stream 1 moves from phase 1 to phase 2 so the old delimiter is not transferred to the new list. The new ends of the matched streams do not occur before the next old delimiter so the search continues for the first new delimiter.

2. Stream 4 moves from phase 1 to phase 2 so the old delimiter is not transferred to the new list. The new ends of the matched streams do not occur before the next old delimiter so the search continues for the first new delimiter.

3. The first new delimiter has been found and is transferred to the new list.

4. The new ends of stream 1 and stream 2 occur before this delimiter. The new end of stream 1 occurs first and is therefore inserted into the new list. The heat content of the old interval, -306.06, is split, on a pro rata basis, into two sections, one (Qa) to be used to complete the current new interval, the other (Qb) to be carried over for use in the next new interval.

$$Qa = -306.06 \times \frac{(346.191-344.261)}{(372.039-344.261)} = -21.26 \text{ kW}$$

$$Qb = -306.06 + 21.26 = -284.80 \text{ kW}$$

Qa is then amended for the removal of streams 1 and 4 to give Qc which is then put on the new list:

$$Qc = Qa - (\dot{m}Cp_4 - \dot{m}Cp_1) \times (346.191 - 344.261)$$

$$= -21.26 - 3.66$$

$$= -24.92 \text{ kW}$$

Stream 1 now enters phase 3.

The next new interval is now considered starting with heat content Qb. The new end of stream 4 still occurs before the next old delimiter so a similar procedure is followed to that given above to give Qd and Qe corresponding to Qa and Qb respectively:

$$Qd = Qb \times \frac{(358.613 - 346.191)}{(372.039 - 346.191)} = -136.87 \text{ kW}$$

$$Qe = -284.80 + 136.87 = -147.93 \text{ kW}$$

Qd is then amended for the removal of stream 4 to give Qf, similar to Qc, which is then placed on the new list:

$$Qf = Qd - \dot{m}Cp_4 \times (358.613 - 346.191)$$

$$= -136.87 - 165.03$$

$$= -301.90 \text{ kW}$$

Stream 4 now enters phase 3.

The new interval is now continued with heat content Qe. The next old delimiter, number 4, now becomes active and can be transferred.

5 to 12 Both matched streams are now in phase 3 and all heat carry-over is complete so no further amendment is required. All the remaining delimiters and interval heat contents can be transferred directly. Stream 1 passes into phase 4 at delimiter 9 and stream 4 at delimiter 7.

Delimiter	Temperature K	Stream	Heat content/kW	∑Heat contents/kW
1	344.261	3		
			-24.92	0.00
2	346.191	1		
			-301.90	24.92
3	358.613	4		
			-147.93	326.82
4	372.039	5		
			-400.66	474.75
5	388.706	2		
			-536.83	875.41
6	460.928	3		
			60.92	1412.24
7	472.039	4		
			-86.69	1351.32
8	483.150	5		
			0.00	1477.01
9	483.150	1		
			334.68	1437.01
10	503.304	H		
			884.48	1103.33
11	503.304	H		
			218.85	218.85
12	516.483	2		

Table L.3 The ordered list after match one

Amendments to account for match two

Stream 3 and 4 are completely satisfied and so all the associated old delimiters are to be deleted and no new delimiters inserted. Considering each old delimiter in turn:

1. Steam 3 moves from phase 1 to phase 2 so the old delimiter is not transferred to the new list.

2. The first new delimiter has been found and is transferred to the new list.

3. Stream 4 moves from phase 1 to phase 2 so the old delimiter is not transferred. The heat content on the old list is corrected for the removal of stream 3 to give Qa which is carried over:

$$Qa = -301.90 + 12.916 \times (358.613-346.191)$$

$$= -141.46 \text{ kW}$$

4. A new delimiter has been found and is transferred to the new list. The heat content of the old interval is corrected for the removal of streams 3 and 4 to give Qb:

$$Qb = -147.93 - (13.285-12.916) \times (372.039-358.613)$$

$$= -152.88 \text{ kW}$$

This is added to Qa carried over from the last old interval to give the new interval heat content Qc:

$$Qc = -141.46 - 152.88 = -294.34 \text{ kW}$$

5. A new delimiter has been found and is transferred to the new list. The heat content of the interval is corrected for the removal of stream 3 and 4 to give Qa which is added to the new list:

$$Qa = -400.66 - (13.285-12.916) \times (388.706-372.039)$$

= -406.81 kW

6. Stream 3 moves from phase 2 to phase 4 so the old delimiter is not transferred. The heat content on the old list is corrected for the removal of streams 3 and 4 to give Qa which is carried over:

Qa = -536.83 - (13.285-12.916) x (460.928-388.706)

= -563.48 kW

7. Stream 4 moves from phase 2 to phase 4 so the old delimiter is not transferred. The heat content on the old list is corrected for the removal of stream 4 to give Qb which is carried over:

Qb = 60.92 - 13.285 x (472.039-460.928)

= -86.69 kW

8. A new delimiter has been found and is transferred to the new list. The heat content of this old interval is added to Qa and Qb to give the new interval heat content Qc:

Qc = -563.48 - 86.69 - 86.69

= -736.86 kW

9 to 12. Both matched streams are now in phase 4 and all heat carry-over is complete so no further amendment is required. All the remaining delimiters and interval heat contents can be transferred directly.

Delimiter	Temperature K	Stream	Heat content/kW	∑Heat contents/kW
1	346.191	1		
			-294.34	0.00
2	372.039	5		
			-406.81	294.34
3	388.706	2		
			-736.86	701.15
4	483.150	5		
			0.00	1438.01
5	483.150	1		
			334.68	1438.01
6	503.304	H		
			884.48	1103.33
7	503.304	H		
			218.85	218.85
8	516.483	2		

Table L.4 The ordered list after match two

Amendments to account for match three

The new delimiters are 422.568K for stream 2 and 415.223K for stream 5. Considering each old delimiter in turn:

1. The first new delimiter has been found and is transferred to the new list.

2. Stream 5 moves from phase 1 to phase 2 so the old delimiter is not transferred to the new list. The heat content of the old interval is assigned to Qa and carried over:

   Qa = -294.34 kW

3. Stream 2 moves from phase 1 to phase 2 so the old delimiter is not transferred to the new list. The heat content of the old interval is corrected for the removal of stream 5 to give Qb which is carried over:

$$Qb = -406.81 + 13.021 \times (388.706-372.039)$$

$$= -189.79 \text{ kW}$$

4. The new ends of streams 2 and 5 occur before this delimiter. The new end of stream 5 occurs first and is therefore inserted into the new list. The heat content of the old interval, -736.86, is split, on a pro rata basis, into two sections; one (Qc) to be used to complete the current interval, the other (Qd) to be carried over for use in the next interval:

$$Qc = -736.86 \times (415.223-388.706) = -206.89 \text{ kW}$$

$$\frac{\qquad\qquad\qquad\qquad}{(483.150-388.706)}$$

$$Qd = -736.86 + 206.89 \qquad = -529.97 \text{ kW}$$

Qc is then amended for the matched stream in phase 2 to give Qe:

$$Qe = Qc - (\dot{m}Cp_2 - \dot{m}Cp_5) \times (415.223-388.706)$$

$$= -206.89 - 95.06$$

$$= -301.95 \text{ kW}$$

Qe is added to Qa and Qb to give the complete interval heat content Qf which is added to the new list:

$$Qf = Qe + Qa + Qb$$

$$= -301.95 - 294.34 - 189.79$$

$$= -786.08 \text{ kW}$$

Stream 5 now enters phase 3.

The next new interval is now considered, starting with heat content Qd. The new end of stream 2 is next to be inserted and a similar procedure to that described above is followed giving Qg and Qh corresponding to Qc and Qd respectively:

$$Qg = Qd \times \frac{(422.568-415.223)}{(483.150-415.223)} = -57.31 \text{ kW}$$

$$Qh = Qd - Qg = -472.66 \text{ kW}$$

Qg is then amended for the removal of stream 2 to give Qi, similar to Qe, which is then placed on the new list:

$$Q_i = Q_g - \dot{m}Cp_2 \times (422.568-415.223)$$

$$= -179.28 \text{ kW}$$

Stream 2 now enter phase 3

The new interval is now continued with heat content Qh. The next old delimiter, number 4, now becomes active and can be transferred.

5 to 8 Both matched streams are now in phase 3 and all heat carry-over is complete so no further amendment is required. All the remaining delimiters and interval heat contents can be transferred directly.

Delimiter	Temperature / K	Stream	Heat content/kW	∑Heat contents/kW
1	346.191	1		
			-786.08	-0.01
2	415.223	5		
			-179.28	786.07
3	422.568	2		
			-472.66	965.35
4	483.150	5		
			0.00	1438.01
5	483.150	1		
			334.68	1438.01
6	503.304	H		
			884.48	1103.33
7	503.304	H		
			218.85	218.85
8	516.483	2		

Table L.5 The ordered list after match three

Amendments to account for match four

Streams 1 and 2 are completely satisfied and so all the associated old delimiters are to be deleted and no new delimiters inserted.

1. Stream 1 moves from phase 1 to phase 2 so the old delimiter is not transferred to the new list.

2. The first new delimiter has been found and is transferred to the new list.

3. Stream 2 moves from phase 1 to phase 2 so the old delimiter is not transferred to the new list. The heat content of the old interval is corrected for the removal of stream 1 to give Qa which is carried over:

Qa = -179.28 + 11.387 x (422.568-415.223)

= -95.64 kW

4. A new delimiter has been found and is transferred to the new list. The heat content of the old interval is corrected for the removal of streams 1 and 2 to give Qb:

Qb = -472.66 - (16.606-11.387) x (483.150-422.568)

= -788.84 kW

This is added to Qa to give the new interval heat content Qc which is added to the new list:

Qc = -95.64 - 788.84

= -884.48 kW

5. Stream 1 moves from phase 2 to phase 4 so the old delimiter is not transferred. The heat content on the old list would normally be corrected for the removal of streams 1 and 2 at this point but the interval has a temperature range of zero making amendment unnecessary.

6. A new delimiter has been found and is transferred to the new list. The heat content of the old interval is corrected for the removal of stream 2 to give Qa which is added to the carry-over from the last interval (zero) and added to the new list:

$$Qa = 334.68 - 16.606 \times (503.304-483.150)$$

$$= 0.00 \text{ kW}$$

7. A new delimiter has been found and is transferred to the new list. The interval contains the heat content of a single-component phase change which is not involved in the current match and, hence, does not require amendment. The heat content is transferred to the new list.

8. Stream 1 moves from phase 2 to phase 4. No more delimiters exist on the old list and the procedure terminates.

Delimiter	Temperature K	Stream	Heat content/kW	∑Heat contents/kW
1	415.223	5	-884.48	0.0
2	483.150	5	0.0	884.48
3	503.304	H	884.48	884.48
4	503.304	H		

Table L.6 The ordered list after match four

Amendments to account for match five

Stream 5 and the hot utility stream are matched to complete the problem. All the delimiters are deleted.

APPENDIX M THE BASIC PROBLEM DATA

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp/kWK^{-1}$	Stream Type
1	269.261	488.706	36.902	Cold process
2	533.150	316.483	10.544	Hot process
3	494.261	383.150	26.359	Hot process
4	477.594	316.483	15.815	Hot process
H	508.860	508.860	-	Hot utility

ΔT_{min} = 11.111K

Table M.1 The data for problem 4SP2

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp/kWK^{-1}$	Stream type
1	310.928	477.594	11.387	Cold process
2	522.039	394.261	16.606	Hot process
3	338.706	455.372	12.916	Cold process
4	477.594	338.706	13.285	Hot process
5	366.483	477.594	13.021	Cold process
H	508.860	508.860	-	Hot utility

$\Delta Tmin = 11.111K$

Table M.2 The data for problem 5SP1

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp/kWK^{-1}$	Stream type
1	366.483	477.594	8.4348	Cold process
2	310.928	494.261	8.4348	Cold process
3	583.150	477.594	12.527	Hot process
4	422.039	477.594	21.762	Cold process
5	517.039	366.483	8.3136	Hot process
6	338.706	410.928	13.833	Cold process
7	551.483	338.706	6.9587	Hot process
H	508.860	508.860	-	Hot utility

$\Delta Tmin = 11.111K$

Table M.3 The data for problem 7SP2

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp$/kWK^{-1}	Stream type
1	333.150	433.150	7.6177	Cold process
2	433.150	366.483	8.7863	Hot process
3	388.706	533.150	6.0783	Cold process
4	522.039	410.928	10.5435	Hot process
H	555.330	555.330	-	Hot utility
C	310.930	355.370	-	Cold utility

ΔTmin = 11.111K

Table M.4 The data for problem 4SP1

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp$/kWK^{-1}	Stream type
1	333.150	453.150	3.0	Cold process
2	453.150	313.150	2.0	Hot process
3	303.150	378.150	2.6	Cold process
4	423.150	313.150	4.0	Hot process
H	531.150	531.150	-	Hot utility
C	353.150	303.150	-	Cold utility

ΔTmin = 10.000K

Table M.5 The data for problem TC1

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp/kWK^{-1}$	Stream type
1	291.15	396.15	0.933	Cold process
2	391.15	466.15	1.961	Cold process
3	462.15	559.15	1.796	Cold process
4	432.15	350.15	2.285	Hot process
5	540.15	353.15	0.204	Hot process
6	616.15	363.15	0.538	Hot process
H	576.15	576.15	-	Hot utility
C	333.15	300.15	-	Cold utility

ΔT_{min} = 10.000K

Table M.6 The data for the six-stream problem

Stream	T^{in}/K	T^{out}/K	$\dot{m}Cp/kWK^{-1}$	Stream type
1	288.706	649.817	24.777	Cold process
2	630.372	338.706	7.908	Hot process
3	583.150	505.372	5.799	Hot process
4	555.372	319.261	2.372	Hot process
5	494.261	447.039	31.631	Hot process
6	477.594	310.928	6.326	Hot process
7	422.039	383.150	65.897	Hot process
H	699.817	699.817	-	Hot utility
C	299.817	333.150	-	Cold utility

ΔT_{min} = 11.111K

Table M.7 The data for problem 7SP4

APPENDIX N THE EFFECT OF CUT-OFF TEMPERATURE RANGE ON THE PROBLEM-TREE SEARCHES

The CPU times quoted in the following tables refer to a CDC Cyber 855. Two sets of CPU times are given, the first is the total CPU time used and the second is the total CPU time less the time spent reading in and checking the problem data, setting up the root node or nodes and printing out the solutions. Of these operations the printing out of solutions is by far the most expensive. Each solution printed out expends approximately 0.03 s CPU time which adds up to an appreciable running time if the solutions are numerous.

Cut-off temperature range/K	Solutions found	Total CPU time/s	Calculation CPU time/s
25	1	0.355	0.229
50	1	1.029	0.900
75	2	1.282	1.124
500	2	1.316	1.163

Table N.1 The effect of cut-off temperature range on the search tree of problem 4SP2

Cut-off temperature range/K	Solutions found	Total CPU time/s	Calculation CPU time/s
25	4	0.251	0.031
50	28	1.183	0.306
75	89	3.803	1.239
100	126	6.492	2.809
150	131	8.285	4.494
500	131	8.379	4.580

Table N.2 The effect of cut-off temperature range on the search tree of problem 5SP1

Cut-off temperature range/K	Solutions found	Total CPU time/s	Calculation CPU time/s
25	0	0.124	0.006
50	10	0.466	0.076
75	100*	3.371	0.449
500	100*	6.617	3.720

* upper limit

Table N.3 The effect of cut-off temperature range on the search tree of problem 7SP2

Cut-off temperature range/K	Solutions found		Total CPU time/s	Calculation CPU time/s
	above	below		
	pinch point			
25	1	1	0.150	0.006
50	1	3	0.201	0.018
75	1	3	0.201	0.018
500	1	3	0.204	0.023

Table N.4 The effect of cut-off temperature range on the search tree of problem 4SP1

Cut-off temperature range/K	Solutions found		Total CPU time/s	Calculation CPU time/s
	above	below		
	pinch point			
25	1	0	0.162	0.035
50	1	2	0.745	0.551
75	1	4	1.164	0.905
100	1	4	1.301	1.047
150	1	5	1.363	1.085
500	1	5	1.364	1.085

Table N.5 The effect of cut-off temperature range on the search tree of problem TC1

Cut-off temperature range/K	Solutions found		Total CPU time/s	Calculation CPU time/s
	above	below		
	pinch point			
25	8	1	0.388	0.070
50	52	1	1.359	0.214
75	52	7	1.585	0.317
100	52	8	1.705	0.436
500	58	8	1.711	0.483

Table N.6 The effect of cut-off temperature range on the search tree of the six-stream problem

Cut-off temperature range/K	Solutions found		Total CPU time/s	Calculation CPU time/s
	above	below		
	pinch point			
25	44	0	1.681	0.329
50	100*	0	3.545	0.665
75	100*	0	4.073	1.150
100	100*	0	4.898	1.963
150	100*	6	5.835	2.747
200	100*	10	6.770	3.606
500	100*	10	6.805	3.621

* upper limit

Table N.7 The effect of cut-off temperature range on the search tree of problem 7SP4

APPENDIX O REPRODUCTION OF A PAPER GIVEN AT THE FIRST UK NATIONAL HEAT TRANSFER CONFERENCE, LEEDS, JULY, 1984

AN ALGORITHM TO DEVELOP HEAT EXCHANGER NETWORKS WITH THE MINIMUM NUMBER OF STREAM SPLITS AND MAXIMUM ENERGY RECOVERY

K.W. Hanson* and A.R.H. Cornish*

Linnhoff and Hindmarsh (1983) have stated the conditions which must be satisfied in the vicinity of pinch points in heat exchanger networks in order to achieve maximum energy recovery. To meet those conditions it is often necessary for one or more streams to be split into a number of parallel branches. However, the conditions themselves do not provide guidance as to which streams should be split in order to minimise the number of splits. An algorithm to solve the mixed integer linear programme has been developed which provides the complete set of suitable networks in the vicinity of the pinch.

INTRODUCTION

Many heat exchanger network synthesis problems can be solved with configurations which do not contain process streams split into parallel branches. In other cases stream splits are necessary and there may be several possible choices of which streams should be split.

In some cases there exist one or two streams of large heat capacity which require heat exchange with a larger number of streams of much smaller heat capacity so that the choice of which streams to split may be obvious. However, there is another type of problem in which parallel splits are necessary but their placement is far from obvious. It is these problems that this paper addresses. In particular, the paper considers the regions where stream splitting is most likely to be needed: the pinch points.

PROBLEM DESCRIPTION

At a problem pinch point a number (N_h) of process streams to be cooled are to be matched with a number (N_c) of process streams to be heated. Conventionally, these are referred to as the hot streams and the cold streams, respectively.

Each stream has a known heat capacity flowrate. These are denoted thus:

$\dot{m}Cp_h^i$ heat capacity flowrate of hot stream i

$\dot{m}Cp_c^j$ heat capacity flowrate of cold stream j

*Department of Chemical Engineering and Chemical Technology, Imperial College, London SW7 2AZ

Each stream may split up into any number of parallel branches.. The number of branches on hot stream i is given by B_h^i and cold stream j by B^j. In any practical network only a small number of all the possible matches between branches of hot and cold streams exist. The existence of a match between a branch of hot stream i and a branch of cold stream j is indicated by the binary integer X^{ij} which is assigned a value of unity when a match occurs and a value of zero otherwise.

The parallel nature of any stream splits necessitates a designation for the heat capacity flowrate in each branch. These are denoted thus:

$\dot{m}Cp_h^{ij}$ heat capacity flowrate of the branch of hot stream i that matches with a branch of cold stream j

$\dot{m}Cp_c^{ij}$ heat capacity flowrate of the branch of cold stream j that matches with a branch of hot stream i

Both $\dot{m}Cp_h^{ij}$ and $\dot{m}Cp_c^{ij}$ are defined only when $X^{ij} = 1$.

Finally, the heat capacity flowrate ratio, R^{ij}, of any match is defined, when $X^{ij} = 1$, as:

$$R^{ij} = \frac{\dot{m}Cp_h^{ij}}{\dot{m}Cp_c^{ij}} \qquad (1)$$

The problem is to determine which streams should be split, if any, and the matches required to minimise the total number of branches. The use of more branches than is absolutely necessary contradicts present day engineering practice.

BYPASSES

It is possible for streams, or branches of streams, not to be matched at a pinch point. Thus the occurrence of bypasses must be allowed. Bypass heat capacity flowrates are denoted by matches with pseudostreams $N_c + 1$ for hot streams and $N_h + 1$ for cold. Thus variables:

X^{i,N_c+1} and $\dot{m}Cp_h^{i,N_c+1}$ are defined for hot streams and

$X^{N_h+1,j}$ and $\dot{m}Cp_c^{N_h+1,j}$ are defined for cold.

Maximum energy recovery is, of course, achievable only if there is no bypassing of hot streams immediately above a pinch point and of cold streams immediately below (Linnhoff and Hindmarsh, 1983).

DEFINITIONS

By definition:

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c+1} \dot{m}Cp_h^{ij} \qquad (2)$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h+1} \dot{m}Cp_c^{ij} \quad (3)$$

Further, the number of branches on a hot stream can easily be found from:

$$B_h^i = \sum_{j=1}^{N_c+1} X^{ij} \quad (4)$$

and on a cold stream:

$$B_c^j = \sum_{i=1}^{N_h+1} X^{ij} \quad (5)$$

CONDITIONS FOR MAXIMUM ENERGY RECOVERY

The definition of a pinch point and the conditions, given below, that must be satisfied in its immediate vicinity were derived from the observations of Linnhoff and Hindmarsh (1983).

Above a pinch point

From the definition of a pinch point:

$$\sum_{i=1}^{N_h} \dot{m}Cp_h^i < \sum_{j=1}^{N_c} \dot{m}Cp_c^j \quad (6a)$$

To ensure all hot stream branches are serviced:

$$\sum_{i=1}^{N_h+1} B_h^i \leqslant \sum_{j=1}^{N_c+1} B_c^j \quad (7a)$$

with $B_c^{N_c+1} = 0$

To ensure the minimum temperature difference constraint is not violated:

$$R^{ij} \leqslant 1 \text{ if } X^{ij} = 1 \quad (8a)$$

Below the pinch point

Similarly below the pinch point:

$$\sum_{i=1}^{N_h} \dot{m}Cp_h^i > \sum_{j=1}^{N_c} \dot{m}Cp_c^j \quad (6b)$$

$$\sum_{i=1}^{N_h+1} B_h{}^i \geqslant \sum_{j=1}^{N_c+1} B_c{}^j \qquad (7b)$$

with $B_h{}^{N_h+1} = 0$

and $R^{ij} \geqslant 1$ if $X^{ij} = 1$ (8b)

AN EXAMPLE

Consider a pinch region of a problem which involves eleven process streams, five hot and six cold, as shown in figure 1. Each has a specified heat capacity flowrate which is assumed to be constant in the vicinity of this pinch point.

Since the heat capacity flowrates are constant the pinch point must be positioned at the inlet temperature of one of the streams, in this case: cold stream six. Application of equation (6a) and (6b) confirms that this point is a potential pinch point.

Above the pinch point

$$\sum_{i=1}^{5} \dot{m}Cp_h{}^i = 0.6354, \quad \sum_{j=1}^{6} \dot{m}Cp_c{}^j = 0.6755$$

$$\sum_{i=1}^{5} \dot{m}Cp_h{}^i < \sum_{j=1}^{6} \dot{m}Cp_c{}^j \text{ as required by (6a)}$$

Below the pinch point

$$\sum_{i=1}^{5} \dot{m}Cp_h{}^i = 0.6354, \quad \sum_{j=1}^{5} \dot{m}Cp_c{}^j = 0.6137$$

$$\sum_{i=1}^{5} \dot{m}Cp_h{}^i > \sum_{j=1}^{5} \dot{m}Cp_c{}^j \text{ as required by (6b)}$$

Consider the above problem further, by applying equation (7a) to the unsplit streams, it can be seen that splitting is not required just because of the members of streams since:

$$\sum_{i=1}^{6} B_h{}^i = 5, \quad \sum_{j=1}^{7} B_c{}^j = 6$$

$$\sum_{i=1}^{6} B_h{}^i \leqslant \sum_{j=1}^{7} B_c{}^j \text{ as required by (7a)}$$

However, the heat capacity flowrate ratios show that hot stream one must be split into two branches at least in order to satisfy equation (8a). This increases the sum of B_h^i to six, the upper limit. If the problem could be solved with this single split the example would be simple. However, this is not the case, and a further hot stream must be split, violating equation (8a). Thus a cold stream must be split also. The problem is fast becoming unmanageable and the need for an algorithm to isolate minimum branch solutions is clearly shown.

Below the pinch point the problem is much simpler. As before, splitting is not required just because of the numbers of streams, since:

$$\sum_{i=1}^{6} B_h^i = 5, \quad \sum_{j=1}^{6} B_c^j = 5$$

$$\sum_{i=1}^{6} B_h^i \geq \sum_{j=1}^{6} B_c^j \text{ as required by (7b)}$$

However, in this case, it is possible to match the hot and cold streams consecutively (1 to 1, 2 to 2, 3 to 3, etc.) without violating equation (8b). Thus no branching is required below the pinch point.

PICTORIAL REPRESENTATION

The problem can be considered pictorially (see figure 2). The streams with the smaller total heat capacity flowrate are listed in a column on the left hand side of a matrix (hot streams in an above pinch design) and the streams with the larger total heat capacity flowrate are listed in a row along the top.

Associated with each element of the matrix are three numbers. The first is X^{ij}, indicating the existence or otherwise of a match, the second and third $\dot{m}Cp_h^{ij}$ and $\dot{m}Cp_c^{ij}$, the heat capacity flowrates associated with the match. Multiple non-zero X^{ij} entries in any row or column denote the splitting of that stream.

An extra row is now added to the bottom of the matrix headed by the difference between the two total heat capacity flowrates (denoted by $\dot{m}Cp_h^{N_h+1}$ in the example). Associated with each element in this row are two numbers. The first is X^{ij} ($X^{N_h+1,J}$ in the example), indicating the existence or otherwise of a stream bypass, the second $\dot{m}Cp^{ij}$ ($\dot{m}Cp_c^{N_h+1,J}$ in the example), the heat capacity flowrate associated with this bypass. Note the third heat capacity flowrate, which is associated with the elements of other rows, does not exist, since it would be meaningless. In order to ensure maximum energy recovery, a similar column for bypasses associated with the row streams does not exist.

The objective of minimising the number of branches can now be seen as the minimisation of the sum of all the X^{ij} within the constraints of (2), (3) and (8a) or (8b).

The heat capacity flowrate data supplied determines whether an above or below pinch design is being considered. By construction, the limitation on the number of branches, equations (7a) or (7b), is automatically satisfied since the total number of stream branches associated with the rows must be less than the total number associated with the columns by the number of stream branches in the row headed $\dot{m}Cp_h^{N_h+1}$.

PROBLEM REDUCTION

A branch, rather than a complete stream, which bypasses matches immediately adjacent to a pinch point brings little practical benefit to standard network designs. It forces the heat capacity flowrate ratios of the branches that are matched closer to unity and hence reduces the mean temperature differences of the corresponding exchangers.

Reasons for branch bypassing could be for flexibility or controllability purposes or to increase the area of a very small exchanger up to some minimum manufacturable size. However, bypassing of branches will be discounted and only total stream bypassing allowed.

The effect of this restriction is that any heat capacity flowrate in the bottom row of the matrix can be regarded as being 'spare' and available for distribution among the matches in the column. Only if no other entries are in a column does an entry in the bottom row of that column take on any physical significance.

Since partial bypassing is no longer allowed, the objective function must now be changed to take account of this restriction by removing the X^{ij} in the bottom row of any column which has another non-zero entry. The X^{ij} of the bottom row could be removed altogether but in certain, fairly unusual, cases this can cause problems since the modified objective function would then minimise the number of exchangers at the pinch rather than the number of branches (see appendix).

A large proportion of the variables can now be removed from the problem. Preliminary matches can be made so that $R^{ij} = 1$ for each match (satisfying (7a) and (7b)) and the heat capacity flowrates associated with the column streams removed from all elements except those in the bottom row. The bottom row now contains the difference of the row heat capacity flowrate entries in that column and the total column heat capacity flowrate (see figure 3).

Feasible solutions cannot be ruled out by this approach since each match utilises the minimum amount of heat capacity flowrate of the column streams and leaves the greatest flexibility for succeeding matches.

THE BOTTOM ROW

The amount of 'slack' in each column, given by the entry in the bottom row, has important physical consequences. In general, a constant ratio between those entries and the total column heat capacity flowrates is beneficial since each match can diverge away from the pinch at an equal rate. Thus the problem solutions where some of the bottom row entries are very small are likely to be uneconomic.

'Slack' heat capacity flowrate can be moved from one column to another only if a row stream is split. This allows a simplex to be drawn from the bottom row of one column, through the branches of the split row stream, down to the bottom row of another column. 'Slack' heat capacity flowrate can then be transferred from one column to the other by successively subtracting and adding a fixed amount from each element on the simplex in the usual way. Complicated paths of many elements can be built up for this purpose. An example is shown in figure 4 where the matrix in figure 4a is converted to that in figure 4b. The third row is split between columns three and four enabling 'slack' to be transferred between these two columns. Transfer of a heat capacity flowrate of 0.5 MW/K is shown in figure 4.

MATHEMATICAL FORMULATION

The problem can be written in mathematical notation as:

Above pinch

$$\min_{X^{ij}} F = \sum_{i=1}^{N_h} \sum_{j=1}^{N_c} X^{ij} + \sum_{j=1}^{N_c} Y^{j}$$

where

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c} \dot{m}Cp_h{}^{ij} \qquad i = 1 \ldots N_h+1$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h+1} \dot{m}Cp_h{}^{ij} \qquad j = 1 \ldots N_c$$

$$\dot{m}Cp_h{}^{N_h+1} = \sum_{j=1}^{N_c} \dot{m}Cp_c^j - \sum_{i=1}^{N_h} \dot{m}Cp_h^i$$

and

$X^{ij} = 0$ if $\dot{m}Cp_h{}^{ij} = 0$

$X^{ij} = 1$ if $\dot{m}Cp_h{}^{ij} > 0$

Y^j binary integer

$$\sum_{i=1}^{N_h} X^{ij} + Y^j \geq 1 \qquad j = 1 \ldots N_c$$

Below pinch

$$\min_{X^{ij}} F = \sum_{i=1}^{N_h} \sum_{j=1}^{N_c} X^{ij} + \sum_{i=1}^{N_h} Y^{i}$$

where

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c+1} \dot{m}Cp_c{}^{ij} \qquad i = 1 \ldots N_h$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h} \dot{m}Cp_c{}^{ij} \qquad j = 1 \ldots N_c+1$$

$$\dot{m}Cp_c{}^{N_c+1} = \sum_{i=1}^{N_h} \dot{m}Cp_h^i - \sum_{j=1}^{N_c} \dot{m}Cp_c^j$$

and

$X^{ij} = 0$ if $\dot{m}Cp_c{}^{ij} = 0$

$X^{ij} = 1$ if $\dot{m}Cp_c{}^{ij} > 0$

Y^i binary integer

$$\sum_{j=1}^{N_c} X^{ij} + Y^i \geq 1 \qquad i = 1 \ldots N_h$$

The variable Y caters for the completely bypassed stream, taking the value of unity only when all the other elements in the column are zero.

Added constraints

Further constraints can be added to limit the numbers of stream branches or to disallow matches. In this case a solution cannot be guaranteed. To limit the number of branches on any one stream, say hot stream i, a constraint of the form:

$$\sum_{j=1}^{N_c} X^{ij} \leq B_h{}^{i}\,\max$$

must be imposed. To disallow a match, say between hot stream i and cold stream j, a constraint of the form:

$$X^{ij} = 0$$

must be imposed.

MIXED INTEGER LINEAR PROGRAMME

In the form stated, the problem is a mixed integer linear programme (MILP). Computer codes are readily available to solve such problems. Initially software developed at North-Western University was used (MPOS). In particular, a branch and bound technique (BBMIP) within the package was found to be applicable (see appendix).

BBMIP initially solves the problem with all the integer variables replaced by continuous variables bounded within the same limits. If a feasible solution can be found the package successively establishes values for the integer variables and solves the reduced linear programme in order to obtain bounds and determine the search strategy.

The results achieved for a number of test problems (see appendix) supported the validity of the approach. However, only problems up to about nine streams could be solved because of computer running time limitations. The running time was strongly dependent on the upper bound placed on the total number of branches. As problems became larger it became more difficult to estimate this bound and the running times became unrealistic. Additionally, not all the solutions with the optimum objective function value were obtained. The algorithm only finds a verified optimum solution and does not supply all the alternatives.

TRANSPORTATION PROBLEM

The problem can also be seen as a variant of the transportation problem in linear programming. Unlike the classical problem ('Hitchcock problem') in which the transportation costs are based on a unit cost, in this formulation the costs are fixed, i.e. independent of the amount transported. Many codes exist to solve the classical problem and some to solve the fixed and unit cost problem (where the cost of transportation is a mixture of a fixed cost and a unit cost). However, codes that take advantage of the simplifications that arise when there are only fixed costs seem difficult to obtain. In addition, the need to find all the solutions with the optimum function value indicated the need to develop a new approach.

SOLUTION TECHNIQUE

An algorithm to solve this problem has been developed by the authors. The solution technique is based on a branch-and-bound technique. The need to solve a linear program at each node of the search tree, as in MPOS, has been removed by taking advantage of the fixed cost transportation problem characteristics. In particular, the algorithm makes use of the fact that a simplex cannot occur in the main body of the matrix shown in figure 2, (i.e. excluding the bottom row).

A full description of the solution method will be the basis of a further paper.

THE EXAMPLE SOLUTION

A limitation of a maximum of two branches per stream was imposed. The solutions obtained are shown in figures 5, 6 and 7. These three configurations were obtained in a computer running time of 34.89 s (CDC Cyber 170/720). All that is left for the process engineer to carry out is the placement of the 'slack' described earlier to give the final design. Due regard must be made of the relative heat transfer coefficients and the size of each exchanger when allocating the 'slack' since this determines the rate at which the streams diverge from the minimum temperature difference as heat is transferred.

COMPARISON BETWEEN MPOS AND THE NEW METHOD

A reduced version of the problem given above was solved using both MPOS and the new algorithm (see figure 8). Because of machine memory limitations the MPOS solution was obtained using the simplified formulation given in the appendix.

When an upper limit of two branches per stream was imposed BBMIP found 3 solutions in a run time of 293 s. The new method found 8 solutions in a run time of 2.55 s.

Additionally, if an upper limit of three branches per stream is imposed the BBMIP run time increased to 428 s and the new method's run time increased to 3.48 s.

CONCLUSIONS

An algorithm has been developed to determine all the configurations of heat exchangers adjacent to problem pinch points that utilise the minimum number of branches. The algorithm completes its search in a time which is two orders of magnitude quicker than a general purpose optimisation code. Additionally, the process engineer can be certain that all such solutions have been isolated and is free to choose between them on grounds such as flexibility and operability.

FUTURE WORK

In the cases where the number of branches are constrained or matches are disallowed a solution is not guaranteed. Maximum energy recovery may not be possible and some alternative approach is required.

The problem can be viewed as the servicing of all the rows of the matrix without recourse to using utilities if at all possible. In the case where no solution can be found an extra column must be added to represent utility usage. A suitably high cost must be put on matches with this column to minimise its use. Inclusion of this facility is intended.

NOMENCLATURE

B_{c_j}	Number of branches cold stream j is split into
$B_{c_j}{}_{max}$	Maximum number of branches cold stream j can be split into
B_{h_i}	Number of branches hot stream i is split into
$B_{h_i}{}_{max}$	Maximum number of branches hot stream i can be split into
$\dot{m}Cp_c^j$	Heat capacity flowrate of the branch of cold stream j

$\dot{m}Cp_c^{ij}$ Heat capacity flowrate of the branch of cold stream j that matches with a branch of hot stream i (defined only if $X^{ij} = 1$)

$\dot{m}Cp_h^{i}$ Heat capacity flowrate of hot stream i

$\dot{m}Cp_h^{ij}$ Heat capacity flowrate of the branch of hot stream i that matches with a branch of cold stream j (defined only if $X^{ij} = 1$)

N_c Number of cold streams

N_h Number of hot streams

R^{ij} Heat capacity flowrate ratio $\dot{m}Cp_h^{ij}/\dot{m}Cp_c^{ij}$ (defined only if $X^{ij} = 1$)

X^{ij} Binary integer indicating match existence. Equals unity if match exists, zero otherwise

ACKNOWLEDGEMENTS

This work was part of a research programme supported by BP International plc in the form of an extra-mural research award.

Mr K.W. Hanson was financially supported during the course of this work by an SERC award and a bursary from BP International plc.

REFERENCES

1. Land, A.H., Doig, A.G., "An automatic method for solving discrete programming problems", Econometrica 28, 497-520 (1960)
2. Linnhoff, B., Hindmarsh, E., "The pinch design method for heat exchanger networks", Chem.Eng.Sci, 38, 745-763 (1983)

APPENDICES

BBMIP

BBMIP (Branch and Bound Mixed Integer Programme) solves the mixed integer programme given by:

Minimise: $z = c^T x + \bar{c}^T y$

Subject to:

$$Ax + \bar{A}y \begin{bmatrix} \leqslant \\ = \\ \geqslant \end{bmatrix} b$$

$$0 \leqslant x \leqslant U \qquad \text{integral valued}$$

$$0 \leqslant y \leqslant \bar{U}$$

where $\underline{c}$, x, $\underline{U}$ are vectors of n_1 components
$\bar{c}$, y, $\bar{U}$ are vectors of $n - n_1$ components
b is a vector of m components
$\underline{A}$ is an m. n_1 matrix
$\bar{A}$ is an m.$(n - n_1)$ matrix

The method is based on that of Land and Doig (1960).

In this case $\bar{c}$ is a null matrix, U is a unit vector and $\bar{U}$ is a vector filled with the largest heat capacity flowrates of the row streams.

Minimum number of branches versus minimum number of exchangers

These two objective functions are similar but not identical. The differences occur in solutions where complete streams bypass and column streams are split simultaneously. In some cases a match could be placed on the bypass stream to remove a branch on another. This solution has the same number of exchangers but one less branch. This substitution method works satisfactorily when all matches are allowed but can cause problems when matches are disallowed or branches limited. In this case it may not be possible to discard solutions with bypasses with the optimum objective function value because these may be the only solutions. The occurance of this is rare, so the simplified formulation is given below:

$$\min_{X^{ij}} F = \sum_{i=1}^{N_h} \sum_{j=1}^{N_c} X^{ij}$$

Above pinch:

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c} \dot{m}Cp_h{}^{ij} \qquad i = 1 \ldots N_h+1$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h+1} \dot{m}Cp_h{}^{ij} \qquad j = 1 \ldots N_c$$

where

$$\dot{m}Cp_h{}^{N_h+1} = \sum_{j=1}^{N_c} \dot{m}Cp_c^j - \sum_{i=1}^{N_h} \dot{m}Cp_h^i$$

and

$$X^{ij} = 0 \text{ if } \dot{m}Cp_h{}^{ij} = 0$$

$$X^{ij} = 1 \text{ if } \dot{m}Cp_h{}^{ij} > 0$$

Below pinch:

$$\dot{m}Cp_h^i = \sum_{j=1}^{N_c+1} \dot{m}Cp_c{}^{ij} \qquad i = 1 \ldots N_h$$

$$\dot{m}Cp_c^j = \sum_{i=1}^{N_h} \dot{m}Cp_c{}^{ij} \qquad j = 1 \ldots N_c+1$$

where

$$\dot{m}Cp_c{}^{N_c+1} = \sum_{i=1}^{N_h} \dot{m}Cp_h^i - \sum_{j=1}^{N_c} \dot{m}Cp_c^j$$

and

$$X^{ij} = 0 \text{ if } \dot{m}Cp_c{}^{ij} = 0$$

$$X^{ij} = 1 \text{ if } \dot{m}Cp_c{}^{ij} > 0$$

In this formulation the variable Y does not occur.

Test problem data

Stream Number	Hot stream heat capacity flowrates (MW/K)	Cold stream heat capacity flowrates (MW/K)	
1	0.70	1.20	Problem 1
2	0.30	0.20	
1	0.50	0.40	Problem 2
2	0.10	0.30	
1	0.30	0.60	Problem 3
2	0.25	0.50	
3	0.20	-	
4	0.05	-	
1	0.2062	0.2000	Problem 4
2	0.1098	0.1094	
3	0.0739	0.0698	
4	-	0.0618	

Problems 1, 2, 3a and 4a had no restriction on the number of ways a stream could be split.

Problems 3b and 4b had a restriction of two imposed on the number of ways a stream could be split.

Test problem solutions

Problem number	BBMIP performance		New method performance	
	Number of solutions found	CPU time+ (s)	Number of solutions found	CPU time+* (s)
1	1	0.524	1	0.170
2	1	0.430	2	0.160
3a	3	1.535	9	0.350
3b	2	1.141	5	0.250
4a	2	8.909	8	0.540
4b	2	7.152	6	0.380

+CDC Cyber 170/720

*The reading in and printing out of data carries with it an overhead of approximately 0.15 s which is significant in these smaller problems. A comparative figure for BBMIP is not known.

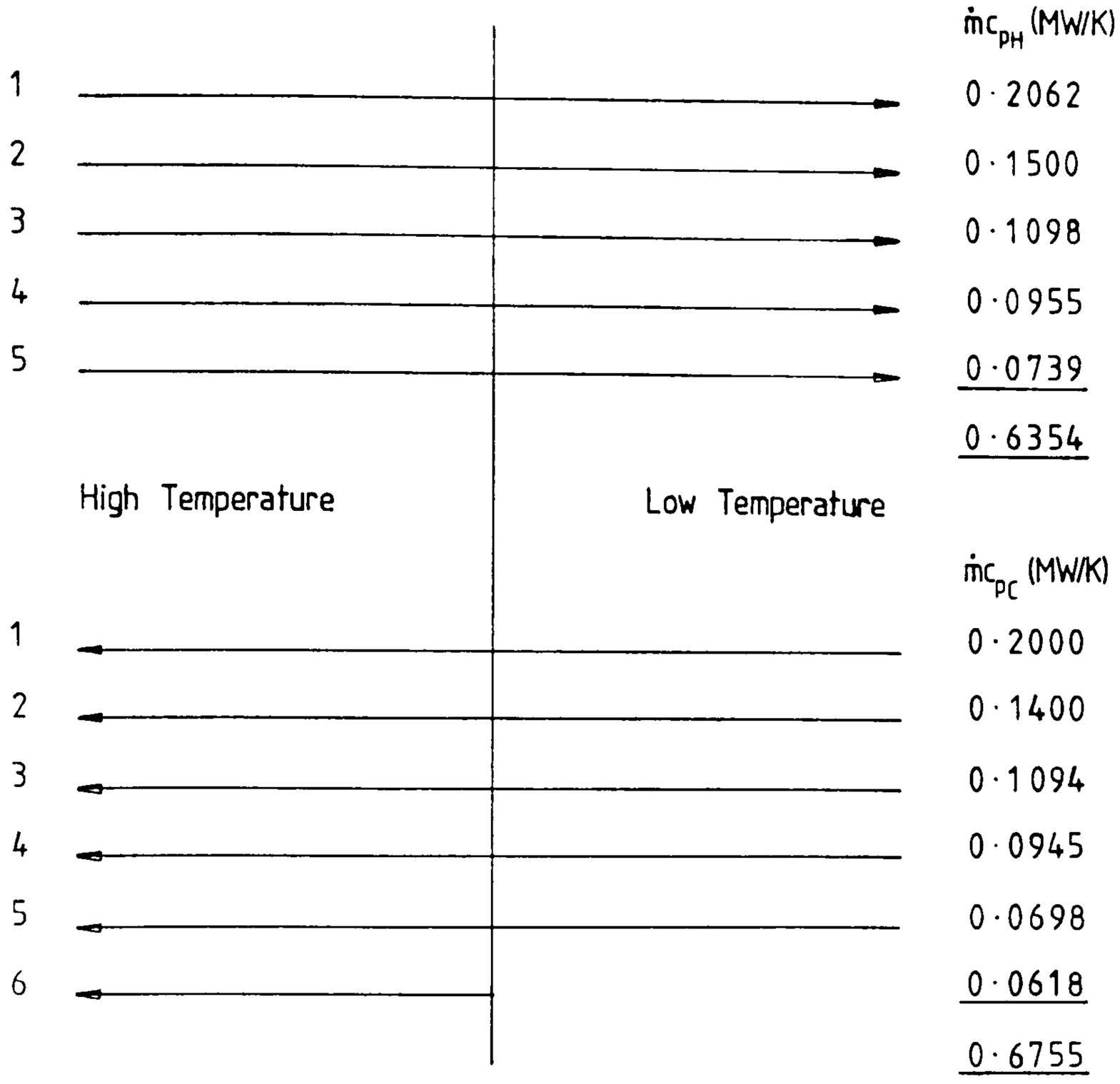


Figure 1. A Pinch Point of the Example Problem.

COLD / HOT	$\dot{m}c_{PC}^{1}$		$\dot{m}c_{PC}^{j}$		$\dot{m}c_{PC}^{Nc}$
$\dot{m}c_{PH}^{1}$	X^{11} $\dot{m}c_{PH}^{11}$ $\dot{m}c_{PC}^{11}$				
$\dot{m}c_{PH}^{i}$			X^{ij} $\dot{m}c_{PH}^{ij}$ $\dot{m}c_{PC}^{ij}$		
$\dot{m}c_{PH}^{Nh}$					
$\dot{m}c_{PH}^{Nh+1}$	$X^{Nh+1,1}$ $\dot{m}c_{PC}^{Nh+1,1}$				

Figure 2. Pictorial Representation for an Above Pinch Point Design.

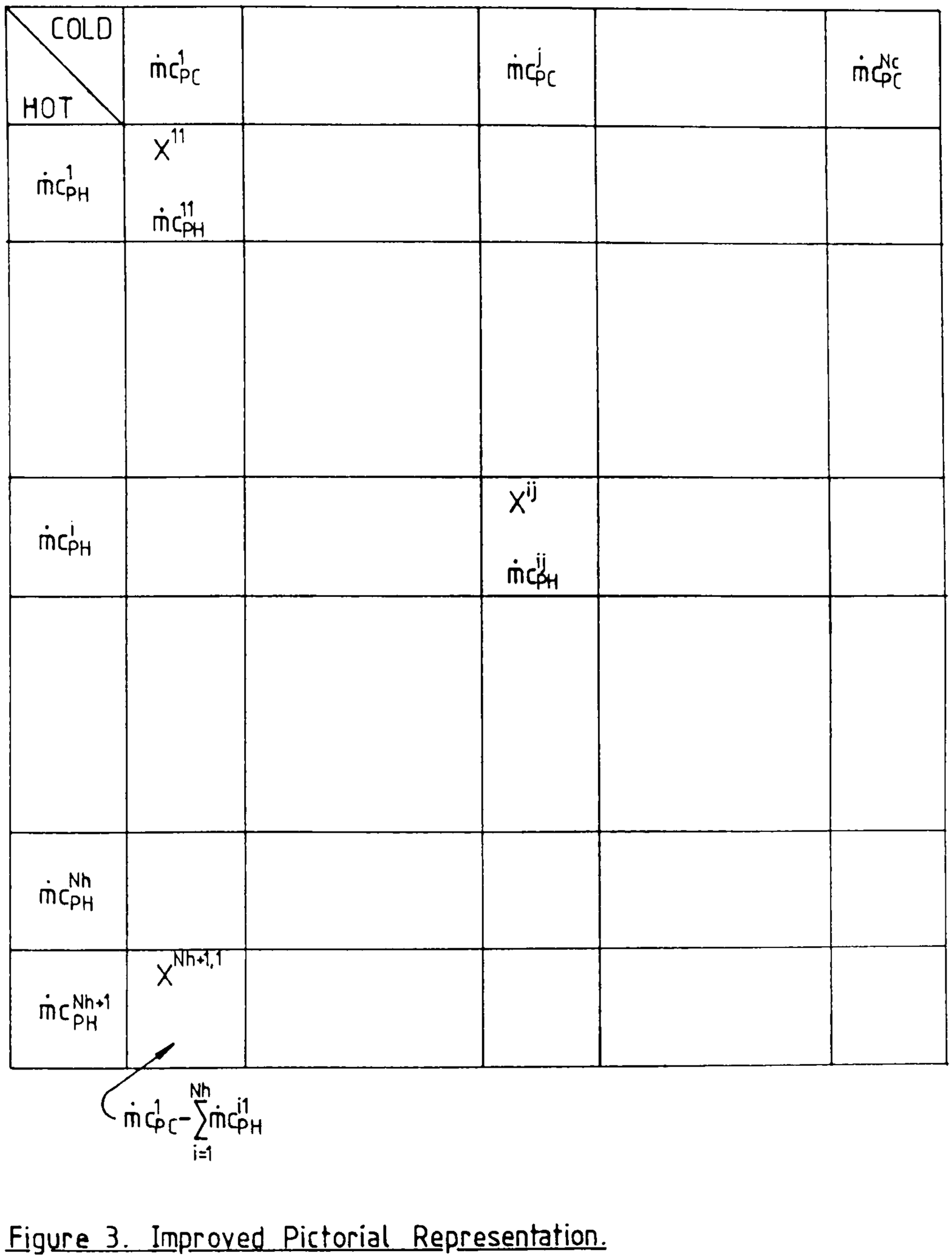


Figure 3. Improved Pictorial Representation.

	6	6	2	2
5	5	0	0	0
4	0	4	0	0
3	0	0	2	1
◆ 4	1	2	0	1

(4a) Before

	6	6	2	2
5	5	0	0	0
4	0	4	0	0
3	0	0	1·5	1·5
◆ 4	1	2	0·5	0·5

(4b) After

Figure 4. Movement of "Slack" using Row Splits

◆ The "Slack" Row

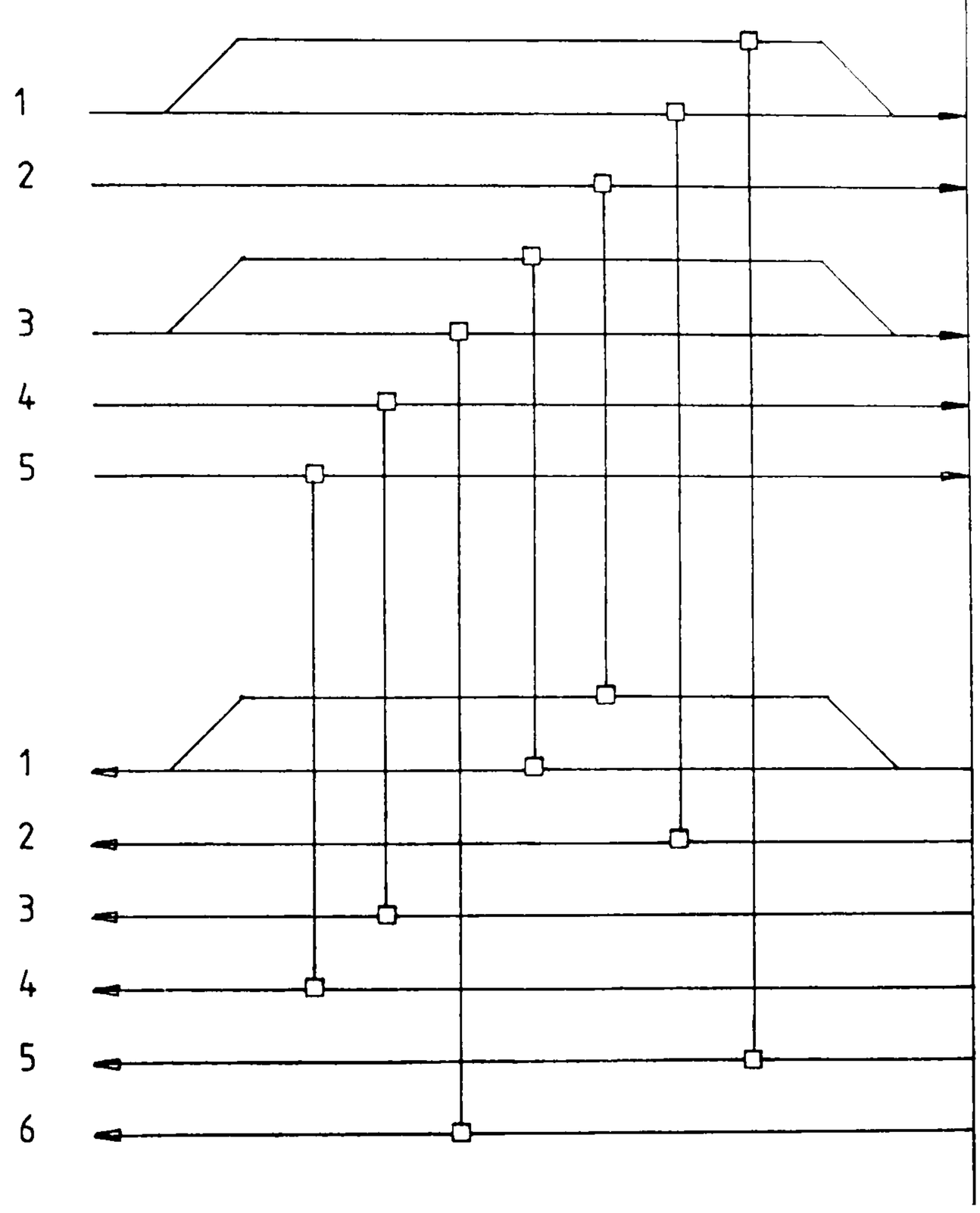


Figure 5. Solution One.

(For $\dot{m}c_p$ data see figure one)

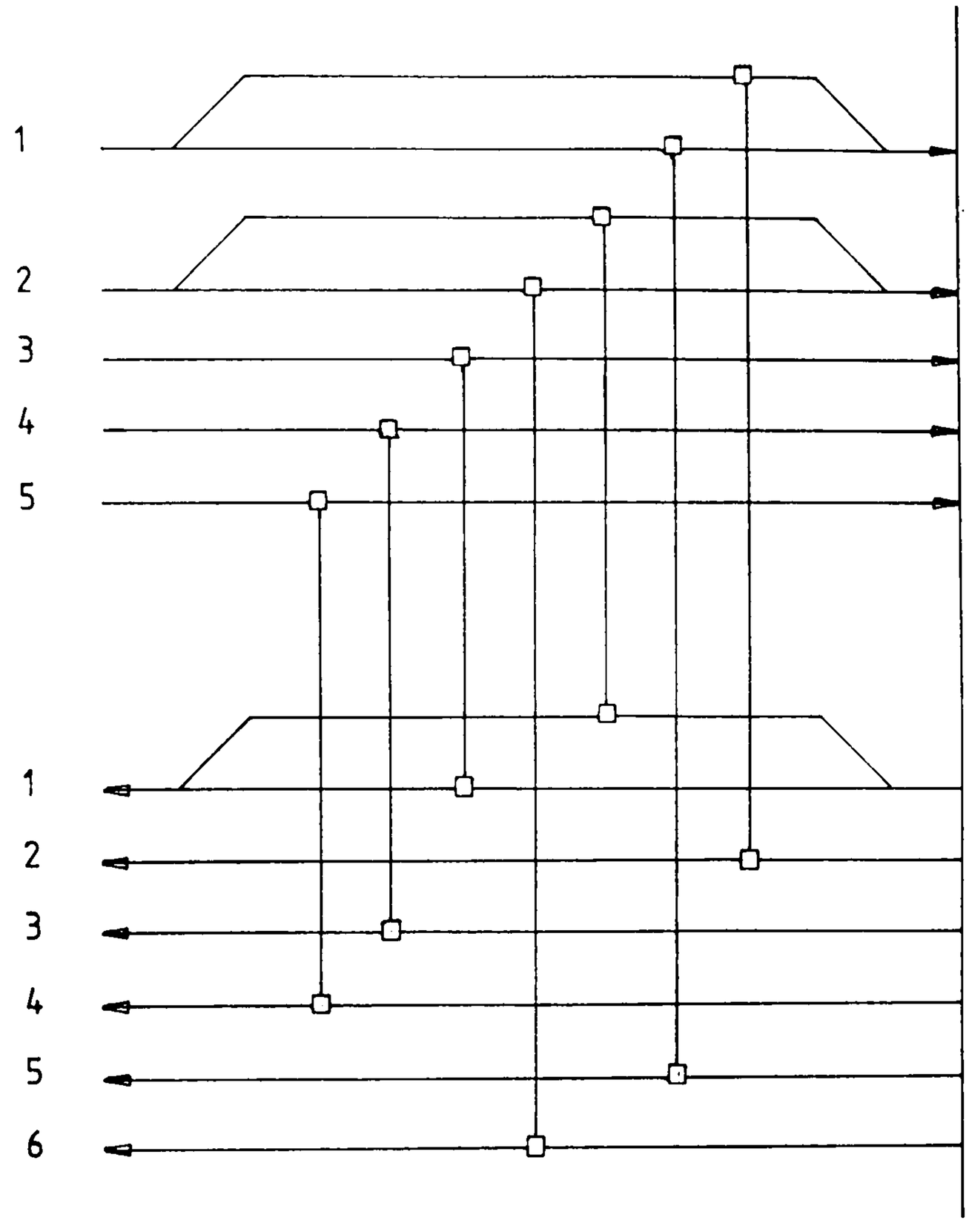


Figure 6. Solution Two.

(For $\dot{m}c_p$ data see figure one)

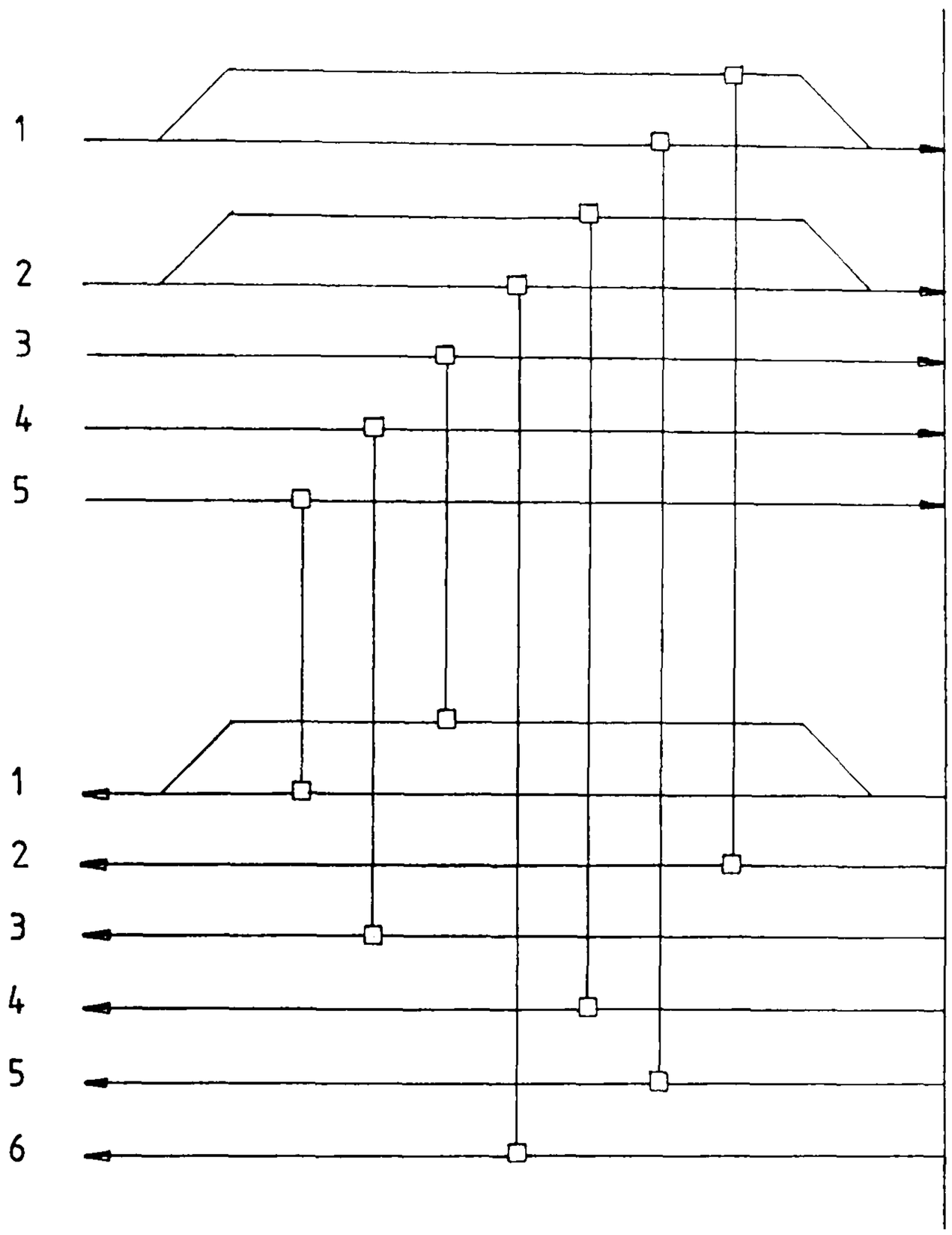


Figure 7. Solution Three.

(For $\dot{m}c_p$ data see figure one)

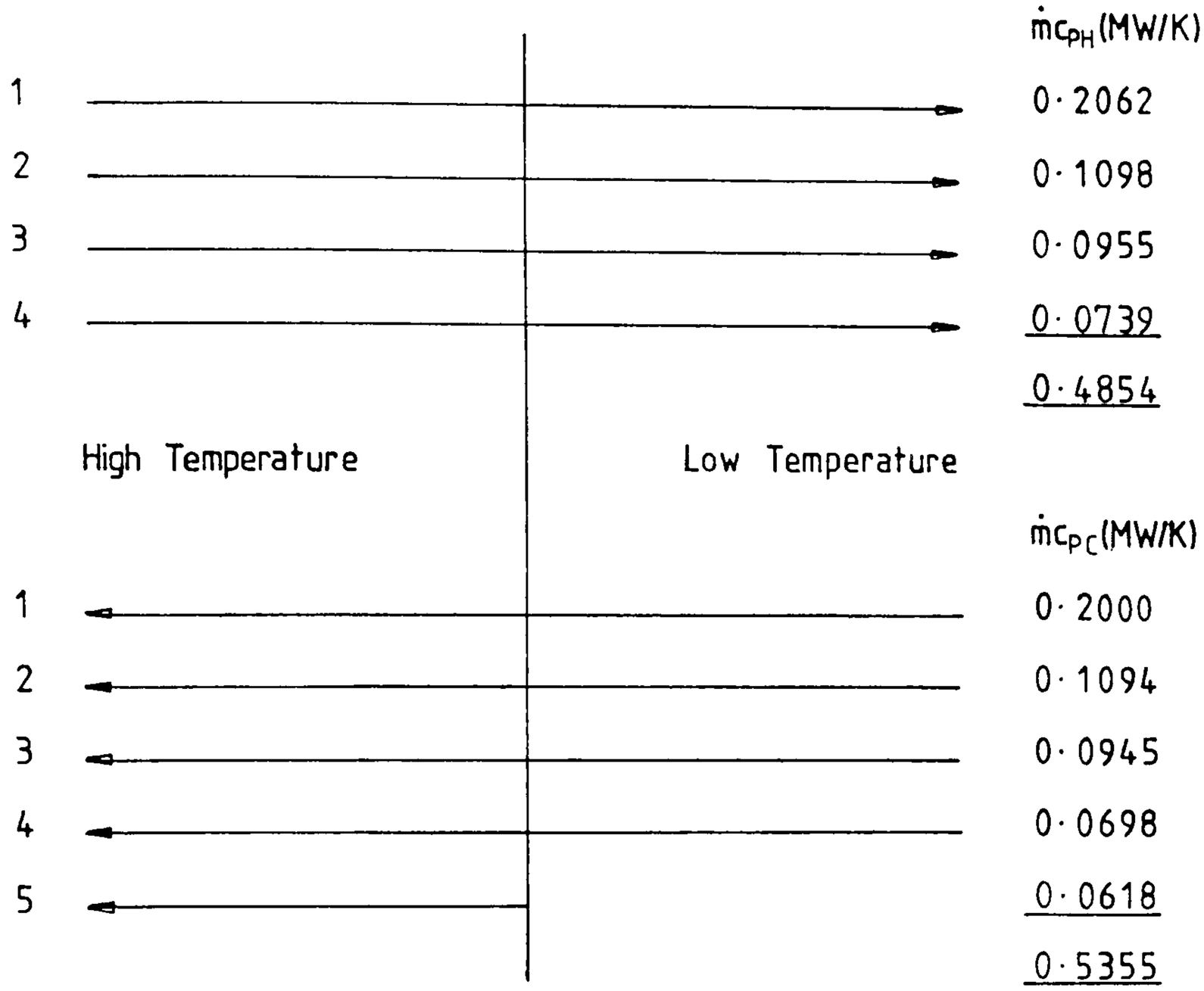


Figure 8. The Reduced Problem.

REFERENCES

Boland.D.,Linnhoff.B. (1979)
'The preliminary design of networks for heat exchange by systematic methods'
Chem. Engnr. No. 343 (April),222-228
also paper given at H.T.F.S. research symp.,Oxford, 1978

Cena.V.,Mustacchi.C.,Natali.F. (1977)
'Synthesis of heat exchange networks: a non-iterative approach'
Chem. Eng. Sci. 32,1227-1231

Cerda.J. (1980)
'Transportation models for the optimal synthesis of heat exchanger networks'
Ph.D. thesis,Carnegie-Mellon Univ.

Cerda.J.,Westerberg.A.W. (1983)
'Synthesizing heat exchanger networks having restricted stream/stream matches using transportation problem formulations'
Chem. Eng. Sci. 38,1723-1740
developed from D.R.C. report,DRC-06-26-81, Carnegie-Mellon Univ., 1981 and D.R.C. report,DRC-06-27-81,Carnegie-Mellon Univ., 1981

Cerda.J.,Westerberg.A.W.,Mason.D.,Linnhoff.B. (1983)
'Minimum utility usage in heat exchanger network synthesis. A transportation problem'
Chem. Eng. Sci. 38,373-387
developed from D.R.C. report,DRC-06-16-80, Carnegie-Mellon Univ., 1979 and D.R.C. report, DRC-06-25-81, Carnegie-Mellon Univ., 1981

Challand.T.B.,Colbert.R.W.,Venkatesh.C.K. (1981)
'Computerized heat exchanger networks'
Chem. Eng. Prog. 77(No. 7,July),65-71
also paper given at A.I.Ch.E. national meeting, Houston, 1981

Colbert,R.W.(1982)
'Industrial heat exchange networks'
Chem. Eng. Prog. 78(No. 7, July),47-54

de Mateo.F. (1979)
'Computer aided synthesis of heat exchanger networks'
M.S. thesis, Univ. Houston

Dodd.R. (1980)
'Mean temperature difference and temperature efficiency for shell and tube heat exchangers connected in series with two tube passes per shell pass'
Trans. I.Chem.E. 58,9-15

Dodd.R. (1982)
'Temperature efficiency of heat exchangers with one shell pass and even number of tube passes'
Trans. I.Chem.E. 60,364-368

Dyson.A.E.S.,Kenny.P.J. (1982)
'The optimisation of oil refinery heat exchanger networks'
Paper given at Chemeca, Sydney.

Fan.L.T.,Hwang.C.L.,Wang.C.S. (1965)
'Optimization of multistage heat exchanger system by the discrete maximum principle'
Chem. Eng. Prog. Symp. Series No. 59, 61,243-252

Flower.J.R.,Linnhoff.B. (1980)
'A thermodynamic-combinatorial approach to the design of optimum heat exchanger networks'
A.I.Ch.E. J. 26,1-9
also paper given at 85th. A.I.Ch.E. national meeting, Philadelphia, 1978

Greenkorn.R.A.,Koppel.L.B.,Raghavan.S. (1978)
'Heat exchange network synthesis - a thermodynamic approach'
Paper given at 71st. A.I.Ch.E annual meeting, Miami

Grimes.L.E. (1980)
'The synthesis and evolution of networks of heat exchange that feature the minimum number of units'
M.S. thesis, Carnegie-Mellon Univ.

Grimes.L.E.,Rychener.M.D.,Westerberg.A.W. (1982)
'The synthesis and evolution of networks of heat exchange that feature the minimum number of units'
Chem. Eng. Comm. 14,339-360
also D.R.C. report, DRC-06-22-81, Carnegie-Mellon Univ., 1981

Grossmann.I.E. (1975)
'Optimum design of heat exchange networks'
M.Sc. thesis, Univ. London

Grossmann.I.E. (1977)
'Problems in the optimum design of chemical plants'
Ph.D. thesis, Univ. London

Grossmann.I.E.,Sargent.R.W.H. (1978)
'Optimum design of heat exchanger networks'
Comp. & Chem. Eng. 2,1-7

Hama.A.,Matsumura.M. (1982)
'Stage-by-stage synthesis of cost optimal heat exchanger networks'
Paper given at Intnl. Symp. Process Systems Eng., Kyoto

Hendry.J.E.,Rudd.D.F.,Seader.J.D. (1973)
'Synthesis in the design of chemical processes'
A.I.Ch.E. J. 19,1-15

Heydweiller.J.C.,Fan.L.T. (1982)
'Process synthesis using structural parameters:further discussion of inequality constraints'
A.I.Ch.E. J. 28,166-168

Hlavacek.V. (1978)
'Synthesis in the design of chemical processes'
Comp. & Chem. Eng. 2,67-75

Hohmann.E.C. (1971)
'Optimum networks for heat exchange'
Ph.D thesis, Univ. S. California

Hohmann.E.C.,Lockhart.F.J. (1976)
'Optimum heat exchanger network synthesis'
Paper given at 82nd. A.I.Ch.E. national meeting, Atlantic City

Hwa.C.S. (1965)
'Mathematical formulation and optimisation of exchanger networks using separable programming'
Paper given at A.I.Ch.E./I.Chem.E. joint symp. No. 4, Pp 83-88

I.Chem.E. (1982)
'User guide on process integration for the efficient use of energy'
I.Chem.E., Rugby

Ichikawa.A.,Fan.L.T. (1973)
'Optimal synthesis of process systems. Necessary condition for optimal system and its use in synthesis of systems'
Chem. Eng. Sci. 28,357-373

Jezowski.J. (1981)
'Heat exchanger network synthesis - algorithms of ordered search'
Inzynieria Chemiczna I Procesowa 2,45-57

Jezowski.J. (1982)
'Heat exchanger network synthesis by combined integer programming - thermodynamic approach'
Paper given at Chemcomp, Antwerp

Jezowski.J.,Pruchnicki.B. (1981)
'Heat exchange network synthesis - a combinatorial analytical approach'
Paper given at Chisa, Prague

Jezowski.J.M.,Stanislawski.J.,Halat.A. (1983)
'A comparison of some methods of heat exchange network synthesis'
Paper given at 3rd. Intnl. Cong. Comp. & Chem. Eng., Paris

Kafarov.V.V.,Meshalkin.V.P.,Nguyen.N.S.,Van Boa.N. (1980)
'Algorithm for heat-exchange system design using a Lagrange function and an implicit breakdown strategy'
D.A.N. S.S.S.R. 250,150-154

Kafarov.V.V.,Shmidt.L.A. (1981)
'Quantized-composition method of sythesizing optimal technological networks in heat-exchange systems'
D.A.N. S.S.S.R. 257,948-952

Kays.W.M.,London.A.L. (1964)
'Compact heat exchangers'
McGraw Hill, New York

Kelahan.R.C.,Gaddy.J.L.(1977)
'Synthesis of heat exchange networks by mixed integer optimization'
A.I.Ch.E. J. 23,816-822
also paper given at 82nd. A.I.Ch.E. national meeting, Atlantic City, 1976

Kestler.M.G.,Parker.R.O. (1969)
'Optimal networks of heat exchange'
Chem. Eng. Prog. Symp. Series No. 92, 65,111-120

Kobayashi.S.,Umeda.T.,Ichikawa.A. (1971)
'Synthesis of optimal heat exchange systems - an approach by the optimal assignment problem in linear programming'
Chem. Eng. Sci. 26,1367-1380

Land.A.h.,Doig.A.G. (1960)
'An automatic method of solving discrete programming problems'
Econometrica 28,497-520,(1960)

Lee.K.F.,Masso.A.H.,Rudd.D.F. (1970)
'Branch and bound synthesis of integrated process designs'
Ind. & Eng. Chemistry, Fundamentals 9,48-58

Liapis.A.I.,Walter,O.H.D. (1977)
'The use of concepts of thermodynamic efficiency in the synthesis of heat exchanger networks'
Paper given at 4th. Research Meeting, I.Chem.E., Swansea

Linnhoff.B. (1979)
'Thermodynamic analysis in the design of process networks'
Ph.D. thesis, Leeds

Linnhoff.B. (1983)
'New concepts in thermodynamics for better chemical process design'
Proc.Roy.Soc. A 386,1-33
also Esso Energy Award Lecture given at the Royal Society, London, 1982 and Chem. Eng. Research & Des. 61,207-233,(1983)

Linnhoff.B.,Flower.J.R. (1978a)
'Synthesis of heat exchanger networks: I. Systematic generation of energy optimal networks'
A.I.Ch.E. J. 24,633-642

Linnhoff.B.,Flower.J.R. (1978b)
'Synthesis of heat exchanger networks: II. Evolutionary generation of networks with various criteria of optimality'
A.I.Ch.E. J. 24,642-654

Linnhoff.B.,Hindmarsh.E. (1983)
'The pinch design method for heat exchanger networks'
Chem. Eng. Sci. 38,745-763
developed from paper given at I.Chem.E. Symp. No. 74, Lancaster, 1982

Linnhoff.B,Mason.D.R.,Wardle.I.(1979)
'Understanding heat exchanger networks'
Comp. & Chem. Eng. 3,295-302
also paper given at 12th. Symp. Comp. & Chem. Eng., Montreux, 1979

Linnhoff.B.,Senior.P.R. (1983)
'Energy targets clarify scope for better heat integration'
Process Eng. 64(No. 3, March),29-33

Linnhoff.B.,Turner.J.A. (1981)
'Heat-recovery networks: new insights yield big savings'
Chem. Eng. 88 (No. 22,November 2nd.),56-70

Masso.A.H.,Rudd.D.F. (1969)
'The synthesis of system designs. II. Heuristic structuring'
A.I.Ch.E. J. 15,10-17

McGalliard.R.L.,Westerberg.A.W. (1972)
'Structural sensitivity analysis in design synthesis'
Chem. Eng. J. 4,127-138

Menzies.M.A.,Johnson.A.I. (1972)
'Synthesis of optimal energy recovery networks using discrete methods'
Canadian J. Chem. Eng. 50,290-296

Muraki.M.,Hayakawa.T. (1982)
'Practical synthesis method for heat exchanger network'
J. Chem. Eng. Japan 15,136-141

N.A.G. (Numerical Algorithms Group) (1983)
'FORTRAN library manual. Mark 10'
N.A.G., Cambridge

Naka.Y.,Takamatsu.T. (1982)
'Design of heat exchanger networks in the heat availability diagram'
Paper given at Chemcomp, Antwerp
also paper given at 15th. Soc. Chem. Eng. Japan Autumn meeting, 1981

Nash.D.B.,Hohmann.E.C.,Beckman.J.,Dobranski.L (1978)
'A simplified approach to heat exchanger network analysis'
Paper given at 85th. A.I.Ch.E. national meeting, Philadelphia

Nilsson.N.J. (1971)
'Problem-solving methods in artificial intelligence'
McGraw Hill, New York

Nishida.N.,Kobayashi.S.,Ichikawa.A. (1971)
'Optimal synthesis of heat exchange systems. Necessary conditions for minimum heat transfer area and their application to systems synthesis'
Chem. Eng. Sci. 26,1841-1856

Nishida.N.,Liu.Y.A.,Lapidus.L. (1977)
'Studies in chemical process design and synthesis: III. A simple and practical approach to optimal synthesis of heat exchanger networks'
A.I.Ch.E. J. 23,77-93
also paper given at 82nd. A.I.Ch.E. national meeting, Atlantic City, 1976

Nishida.N.,Stephanopoulos.G.,Westerberg.A.W. (1981)
'A review of process synthesis'
A.I.Ch.E. J. 27,321-351
also D.R.C. report, DRC-06-17-80, Carnegie-Mellon Univ., 1980

Nishimura.H. (1980)
'A theory for the optimal synthesis of heat-exchanger systems'
J. Opt. Theory & Applications 30,423-450

Nishitani.H.,Kunugita.E.,Fan.L.T. (1982)
'On the vector optimization of heat exchange'
J. Chem. Eng. Japan 15,475-480
also paper given at 45th. Soc. Chem. Eng. Japan annual meeting, 1980

Osakada.K.,Fan.L.T. (1973)
'Synthesis of an optimal large-scale interconnected system by structural parameter method coupled with multilevel technique'
Canadian J. Chem. Eng. 51,94-101

Papoulias.S.A.,Grossmann.I.E. (1983a)
'A structural optimization approach in process synthesis - 1. Utility systems'
Comp. & Chem. Eng. 7,695-706
also paper given at 75th. A.I.Ch.E. annual meeting, Los Angeles, 1982 and D.R.C. report, DRC-06-35-82, Carnegie-Mellon Univ., 1982

Papoulias.S.A.,Grossmann.I.E. (1983b)
'A structural optimization approach in process synthesis - 2. Heat recovery networks'
Comp. & Chem. Eng. 7,707-722
also paper given at 75th. A.I.Ch.E. annual meeting,Los Angeles, 1982 and D.R.C. report, DRC-06-35-82, Carnegie-Mellon Univ., 1982

Pho.T.K.,Lapidus.L. (1973)
'Topics in computer aided design. Part II. Synthesis of optimal heat exchange networks by tree searching algorithms'
A.I.Ch.E. J. 19,1182-1189

Polley.G.T. (1980a)
'Synthesis of heat exchanger networks. Survey of literature to 1978'
N.E.L. report, C/267, DoI

Polley.G.T. (1980b)
'Synthesis of heat exchanger networks. An industrial problem'
N.E.L. report, C/270, DoI

Ponton.J.W.,Donaldson.R.A.B.(1974)
'A fast method for the synthesis of optimal heat exchanger networks'
Chem. Eng. Sci. 29,2375-2377

Raghavan.S. (1977)
'Heat exchanger network synthesis. A thermodynamic approach'
Ph.D. thesis,Purdue Univ.

Rathore.R.N.S. (1982)
'Process resequencing for energy conservation'
Chem. Eng. Prog. 78(No. 12, December),75-82

Rathore.R.N.S.,Powers.G.J. (1975)
'A forward branching scheme for the synthesis of energy recovery systems'
Ind. & Eng. Chemistry, Process Des. & Dev. 14,175-181

Rev.E.,Fonyo.Z. (1982)
'Synthesis of heat exchanger networks'
Chem. Eng. Comm. 18,97-106

Shah.J.V.,Westerberg.A.W. (1975)
'Evolutionary synthesis of heat exchanger networks'
Paper given at 68th. A.I.Ch.E. annual meeting, Los Angeles

Shah.J.V.,Westerberg.A.W. (1977)
'Process synthesis using structural parameters: a problem with inequality constraints'
A.I.Ch.E. J. 23,378-380

Shah.J.V.,Westerberg.A.W. (1980)
'EROS: A program for quick evaluation of energy recovery systems'
Comp. & Chem. Eng. 4,21-32

Shiroko.K,Umeda.T. (1981)
'A practical approach to optimal design of heat exchange system'
Paper given at Chisa, Prague

Su.J.L. (1979)
'A loop-breaking evolutionary method for the synthesis of heat exchanger networks'
M.S. thesis, Washington Univ.

Su.J.L.,Motard.R.L. (1984)
'Evolutionary synthesis of heat-exchanger networks'
Comp. & chem. Eng. 8,67-80

Umeda.T. (1983)
'Computer aided process synthesis'
Comp. & Chem. Eng. 7,279-309
also paper given at Intnl. Symp. Process Systems Eng., Kyoto, 1982

Umeda.T.,Harada.T.,Shiroko.K. (1979)
'A thermodynamic approach to the synthesis of heat integration systems in chemical processes'
Comp. & Chem. Eng. 3,273-282
also paper given at 12th. Symp. Comp. & Chem.'Eng., Montreux, 1979

Umeda.T.,Hirai.A.,Ichikawa.A. (1972)
'Synthesis of optimal processing system by an integrated approach'
Chem. Eng. Sci. 27,795-804

Umeda.T.,Itoh.J.,Shiroko.K. (1978)
'Heat exchange system synthesis'
Chem. Eng. Prog. 74(No. 7,July),70-76

Underwood.A.J.V. (1934)
'The calculation of the mean temperature difference in multipass heat exchangers'
J. Inst. Pet. Tech. 20,145-158

Wales.R.E. (1981)
'Mean temperature difference in heat exchangers'
Chem. Eng. 88(No. 4, February 23rd.),77-81

Wells.G.,Hodgkinson.M. (1977)
'The heat content diagram way to heat exchange networks'
Process Eng. 58(No. 8, August),59-63

Westbrook.G.T (1961)
'Use this method to size each stage for best operation'
Hydro. Processing & Pet. Refiner 40 (No. 9,September),201-206

Westerberg.A.W. (1980)
'A review of process synthesis'
Article in 'Computer applications to chemical engineering'
Edited by Squires.R.G.,Reklaitis.G.V.
A.C.S. Symp. Series No. 124, Pp 53-87, Washington
also D.R.C. report, DRC-06-14-79, Carnegie-Mellon Univ., 1979

Westerberg.A.W.,Shah.J.V. (1978)
'Assuring a global optimum by the use of an upper bound on the lower (dual) bound'
Comp. & Chem. Eng. 2,83-92
also paper given at 85th. A.I.Ch.E. national meeting, Philadelphia, 1978

Whistler.A.M. (1948)
'Heat exchangers as money makers'
Pet. Refiner 27(No.3,September),83-86

Wood.R.M. (1982)
'NERDDP sponsored heat exchanger network workshop - case study'
Chem. Eng. Australia 7(No. 3,September),44-48

Zachoval.J.,Konecny.Z. (1982)
'An improved method for the generation of heat exchangers networks, especially suitable for the networks formed of crossflow exchangers'
Paper given at Chemcomp, Antwerp