



จุฬาลงกรณ์มหาวิทยาลัย

ทุนวิจัย

กองทุนรัชดาภิเษกสมโภช

รายงานวิจัย

การศึกษาและพัฒนาเครือข่ายแลกเปลี่ยนความร้อนของหอกลั่น

เพื่อรองรับน้ำมันดิบต่างชนิด

โดย

กิติพัฒน์ สีมานนท์

มกราคม 2557

Acknowledgements

The authors would like to thank the fundings of this research work provided by Petroleum and Petrochemical College, and Ratchadapisek Sompote Fund, Chulalongkorn University.

ชื่อโครงการวิจัย	การศึกษาและพัฒนาเครือข่ายแลกเปลี่ยนความร้อนของหอกลั่นเพื่อรองรับ น้ำมันดิบต่างชนิด
ผู้ทำวิจัย	กิติพัฒน์ สีมานนท์ การวิณี ตังนันทนาการ วัรมพร เพชรพิเชษฐกุล และ บงกช ยิ้มแย้ม
เดือนปีที่ทำวิจัยเสร็จ	มกราคม 2557

บทคัดย่อ

โรงกลั่นน้ำมันดิบเป็นหน่วยที่มีการใช้พลังงานมากที่สุด โดยเตาเผาที่ใช้ในการให้ความร้อนแก่น้ำมันดิบเป็นหน่วยที่ต้องการพลังงานมากที่สุด การจัดการพลังงานเป็นส่วนสำคัญในการควบคุมต้นทุนทั้งหมดจึงถูกนำมาใช้เพื่อบูรณาการทางความร้อน การปรับปรุงเครือข่ายแลกเปลี่ยนความร้อนของหน่วยกลั่นน้ำมันดิบนำไปสู่การลดการใช้พลังงานภายนอก งานวิจัยนี้ใช้โปรแกรมการหาค่าศักยภาพของการปรับปรุง (the retrofit potential program) ของโกศล (2012) เพื่อหาจุดพินช์ (pinch point) ที่เหมาะสมนำไปสู่ขั้นตอนการออกแบบและปรับปรุงผังกระบวนการของเครื่องแลกเปลี่ยนความร้อนด้วยซึ่งเป็นการเขียนโปรแกรมทางคณิตศาสตร์ (mathematical programming) โดยใช้แบบจำลองลำดับขั้น (stage model) ของ Yee และ Grossman (1990) เพื่อหาผังกระบวนการที่ใช้เครื่องแลกเปลี่ยนความร้อนน้อยที่สุด และนำความร้อนภายในระบบมาใช้ที่เครื่องแลกเปลี่ยนความร้อนที่จุดพินช์มากที่สุดจากนั้นจำลองผังกระบวนการด้วยซอฟต์แวร์จำลองโปรตุ (Pro/ID) เพื่อออกแบบและแสดงปริมาณการใช้พลังงานทั้งหมด ตัวอย่างปัญหาของเครื่องแลกเปลี่ยนความร้อนของหน่วยกลั่นน้ำมันดิบ ซึ่งป้อนด้วยน้ำมันเบา, ปานกลางและหนัก โดยมีรอบการกลั่น 100, 150 และ 100 วันต่อปีตามลำดับ ถูกนำมาใช้เพื่อหาผังกระบวนการที่ใช้เหมาะสมและให้ผลตอบแทนในรูปแบบมูลค่าปัจจุบันสุทธิสูงสุด

Research Title An Investigation and Development on Retrofit of Crude Preheat

Train under Different Kinds of Crude Oils

Researchers: Kitipat Seimanond, Parawinee Tangnanthanakan

Warumporn Pejpichestakul, Bongkoch Yimyam

Project Completion **January 2014**

Abstract

Crude distillation unit (CDU) is the major energy consuming unit in refineries. Because of the high energy consumption of crude furnace in crude preheat train, heat integration by retrofitting heat exchanger network (HEN) is represented in this research by using a retrofit potential program and stage-model mathematical programming. A retrofit potential program from Kosol (2012) based on pinch technology was used to identify the optimum heat recovery approach temperature (HRAT). A stage model from Yee and Grossman (1990), is a mathematical programming using General Algebraic Modeling System (GAMS), was used to generate the retrofit HEN with the minimum number of exchanger and maximum heat recovery at pinch exchangers as an objective function. The simulation software (Pro II) was used to validate designs and perform total utility consumption. An example of CDU is simulated with the operation of feeding light, medium and heavy crude oils for a period of 100, 150 and 100 days per year respective. The retrofit of crude preheat train was applied to find the optimal and profitable HEN design that yields the highest net present value (NPV).

TABLE OF CONTENTS

	PAGE
Title Page	i
Abstract (in English)	iii
Abstract (in Thai)	iv
Acknowledgements	v
Table of Contents	vi
List of Tables	ix
List of Figures	x
Abbreviations	xii
List of Symbols	xiii
 CHAPTER	
I INTRODUCTION	1
II LITERATURE REVIEW	3
2.1 Heat Exchanger Network Design Methodologies	3
2.1.1 Pinch Analysis Concept	4
2.1.2 Sequential Approaches	6
2.1.3 Simultaneous Approaches	8
2.2 Multiperiod Heat Exchanger Network	12
2.2.1 Degree of Flexibility	12
2.2.2 Review of Multiperiod HEN Designs	13
2.2.3 Simultaneous MINLP Model	15
2.3 Heat Integration in Crude Distillation Unit (CDU)	24
2.3.1 Crude Oil Refinery	24
2.3.2 Crude Oil Assay	24
2.3.3 Heat Integration in Crude Distillation Unit	25

CHAPTER	PAGE
III EXPERIMENTAL	27
3.1 Equipment	27
3.1.1 Computer	27
3.1.2 Software	27
3.2 Methodology	27
3.2.1 Formulation of Stage-wise Superstructure Model for Single Period	27
3.2.2 Sequential Approach for Multiperiod HEN Synthesis	28
3.2.3 Simultaneous Approach for Multiperiod HEN Synthesis	28
3.2.4 Application to the Industrial Case: CDU Process	28
3.2.5 Model Improvement	29
 IV RESULTS AND DISCUSSION	 30
4.1 Formulation of MINLP Single Period Model	30
4.1.1 Model Components	30
4.1.2 Assumptions	31
4.1.3 Formulations	32
4.2 Sequential Approach for Multiperiod HEN Synthesis	36
4.2.1 Algorithm	36
4.2.2 Case Study	40
4.2.3 Results	43
4.3 Developing a Multiperiod Model for Simultaneous Approach	59
4.3.1 Formulation of MINLP Multiperiod Model	59
4.3.2 Results	64
4.4 Application to Crude Preheat Train in Crude Distillation Unit	66
4.4.1 Case Study Simulation	66
4.4.2 Results	74

CHAPTER		PAGE
V	CONCLUSIONS AND RECOMMENDATIONS	87
	REFERENCES	89
	APPENDICES	93
	Appendix A Source Code of MINLP Single Period Model	93
	Appendix B Source Code of Models for HEN Adaptation	102
	Appendix C Source Code of Simultaneous MINLP Multiperiod Model	140
	CURRICULUM VITAE	150

LIST OF TABLES

TABLE	PAGE
4.1 Stream Properties for SOR	41
4.2 Stream Properties for MOR	41
4.3 Stream Properties for EOR	42
4.4 Results of multiperiod HEN from each method	49
4.5 Results of multiperiod HEN with different starting period for initial HEN	58
4.6 Comparison of multiperiod HEN from sequential and simultaneous approach	65
4.7 True boiling point data of light crude	67
4.8 True boiling point data of medium crude	67
4.9 True boiling point data of heavy crude	68
4.10 Crude used for simulated case study	69
4.11 Light end composition of crude	69
4.12 Products specifications	71
4.13 Column feed and side draw tray information	71
4.14 Column utilities information	72
4.15 Stream properties of light crude (Troll)	73
4.16 Stream properties of medium crude (Forozan)	73
4.17 Stream properties of heavy crude (Souedie)	74
4.18 ALMTD of match between process streams in each period	80
4.19 ALMTD of match between process stream and utilities in each period	81
4.20 Summary result of HEN with and without initialization for refinery case study	82
4.21 Summary result of HEN with and without initialization for another refinery case study	84
4.22 Summary result of multiperiod HEN after validation	85

LIST OF FIGURES

FIGURE	PAGE
2.1 Hot and cold composite curves.	4
2.2 Economic trade-off between energy cost and capital cost.	6
2.3 Heat flows in interval k.	7
2.4 Hyperstructure of Floudas and Ciric (1989).	9
2.5 Stage-wise superstructure model.	10
2.6 Excluded HEN configuration of simplified superstructure.	11
2.7 Feasible region of operation.	13
2.8 Illustration of variables involving in a heat exchanger.	16
2.9 LMTD Approximation.	19
2.10 A crude distillation complex.	24
2.11 TBP distillation curve of different crude oils.	25
2.12 CDU process flow diagram.	26
4.1 Algorithm of sequential approach for multiperiod HEN synthesis.	36
4.2 Grid diagram of HEN-2.	43
4.3 Grid diagram of HEN-2-A ₁ .	44
4.4 Grid diagram of HEN-2-A ₃ .	45
4.5 Grid diagram of HEN-2-B ₁ .	45
4.6 Grid diagram of HEN-2-B ₃ .	46
4.7 Grid diagram of HEN-2-C ₁ .	46
4.8 Grid diagram of HEN-2-C ₃ .	47
4.9 Grid diagram of multiperiod HEN-2A.	48
4.10 Grid diagram of multiperiod HEN-2B.	48
4.11 Grid diagram of multiperiod HEN-2C.	49
4.12 Overall procedure of sequential method by using model C.	51
4.13 Grid diagram of HEN-1.	52
4.14 Grid diagram of HEN-2.	53
4.15 Grid diagram of HEN-3.	53
4.16 Grid diagram of HEN-1-C ₂ .	54

FIGURE	PAGE
4.17 Grid diagram of HEN-1-C ₃ .	54
4.18 Grid diagram of HEN-2-C ₁ .	55
4.19 Grid diagram of HEN-2-C ₃ .	55
4.20 Grid diagram of HEN-3-C ₁ .	56
4.21 Grid diagram of HEN-3-C ₂ .	56
4.22 Grid diagram of multiperiod HEN-1C.	57
4.23 Grid diagram of multiperiod HEN-2C.	57
4.24 Grid diagram of multiperiod HEN-3C.	58
4.25 Grid diagram of multiperiod HEN synthesized by simultaneous approach.	65
4.26 Distillation curves of each crude oil.	68
4.27 The crude distillation unit with preflash drum.	70
4.28 Composite curves of period 1 (Light crude).	75
4.29 Composite curves of period 2 (Medium crude).	75
4.30 Composite curves of period 3 (Heavy crude).	76
4.31 Grid diagram of multiperiod HEN for crude preheat train.	77
4.32 Grid diagram of multiperiod HEN for crude preheat train with initialization technique.	83
4.33 Grid diagram of validated HEN for crude preheat train with initialization technique.	85

ABBREVIATIONS

ADU	Atmospheric distillation unit
ALMTD	Average logarithmic mean temperature difference
ASTM	American society for testing and material
CDU	Crude distillation unit
CW	Cooling water
EMAT	Exchanger minimum approach temperature
EOR	End of run
GAMS	General algebraic modeling system
HEN	Heat exchanger network
HENS	Heat exchanger network synthesis
HRAT	Heat recovery approach temperature
LMTD	Logarithmic mean temperature difference
LP	Linear programming
MILP (MIP)	Mixed integer linear programming
MINLP	Mixed integer nonlinear programming
MOR	Mid of run
NLP	Nonlinear programming
SOR	Start of run
TAC	Total annualized cost
TBP	True boiling point
VDU	Vacuum distillation unit
VGO	Vacuum gas oil
ΔT_{\min}	Minimum temperature difference

LIST OF SYMBOLS

Indices

i	Hot process stream or hot utility
j	Cold process stream or hot utility
k	Stage number or temperature interval
p	Period of operation

Sets

CP	Set of cold process streams i
HP	Set of hot process streams j
PR	Set of operational periods, $p = 1, 2, 3, \dots$, NOP
ST	Set of stages, $k = 1, 2, 3, \dots$, NOK

Parameters

AF	Annualisation factor, dimensionless
B	Exponent factor for area cost, dimensionless
C	Area cost coefficient for heat exchanger, \$/unit
C_p	Specific heat capacity, kJ/kg.K
C_{cu}	Per unit cost for cold utility, \$/unit
C_f	Fixed cost for heat exchanger unit, \$/unit
C_{hu}	Per unit cost for hot utility, \$/unit
DOP	Duration of period, dimensionless
DT_{up}	Upper bound on temperature difference, °C
EMAT	Exchanger minimum approach temperature, °C
FCph	Heat capacity flowrate of hot stream i , kW/K
FCpc	Heat capacity flowrate of cold stream i , kW/K
NOK	Number of stages, dimensionless
NOP	Number of periods, dimensionless
Q_{up}	Upper bound of heat load, kW
Th_{in}	Inlet temperature of hot stream, °C
Th_{out}	Outlet temperature of hot stream, °C

T_{cin}	Inlet temperature of cold stream, °C
T_{cout}	Outlet temperature of cold stream, °C
T_{huout}	Outlet temperature of hot utility, °C
T_{cuout}	Outlet temperature of cold utility, °C
U	Overall heat transfer coefficient, kW/m ² .K

Variables

Area	Heat exchanger area for match of hot stream and cold stream, m ²
Area_max	Maximum area for match of hot stream and cold stream, m ²
AreaCU_max	Maximum area for match of hot stream and cold utility, m ²
AreaHU_max	Maximum area for match of hot utility and cold stream, m ²
d _{tcu}	Temperature difference for match of hot stream and cold utility, m ²
d _{thu}	Temperature difference for match of cold stream and hot utility, m ²
dt	Temperature difference for match of hot stream and cold stream, °C
q	Heat load between hot stream and cold stream, kW
q _{cu}	Heat load between hot stream and cold utility, kW
q _{hu}	Heat load between cold stream and cold utility, kW
th	Exchanger inlet temperature of hot stream, °C
tc	Exchanger inlet temperature of cold stream, °C

Binary variables

z _{cu}	Existence of exchanger for match of cold utility and hot stream
z _{hu}	Existence of exchanger for match of hot utility and cold stream
z	Existence of exchanger for match of hot stream and cold stream

CHAPTER I

INTRODUCTION

Energy demand has increased continuously and becomes more important worldwide issue in the last few decades. Petroleum refinery industry encounters the similar crisis especially in crude distillation unit (CDU) which is one of the largest energy-consuming units in the refinery plant. Heat exchanger network synthesis (HENS) is a widely used method to recover excess energy from heat source (hot process streams) and transfer to heat sink (cold process streams).

For CDU, the hot process streams are generally main products from distillation column and pump-around streams between two trays, while the cold streams are mainly crude oil feed in the crude preheat train. In general, HEN is designed to serve a process which has only one set of steady state condition throughout the operational time. That means the design parameters consists of fixed temperatures and flowrate capacities. Therefore, the particular HEN can be efficiently used only for such specific condition, but it may have less efficiency to operate or even no feasibility at other conditions. In crude refinery, crude oil feeds are produced from different areas and reservoirs. It doubtlessly has different compositions due to its geological history of each area. Furthermore, refinery plants sometimes have to blend various kinds of crude before feeding into the process in a period of time as a result of economic aspects. As far as variety of crude characteristics is concerned, multiperiod HEN design is introduced to apply in refinery plants for more flexibility in each period of operation.

Many researchers have attempted to find systematic methodology for designing HENs. Floudas and Grossmann (1986) proposed a multiperiod Mixed Integer Linear Programming (MILP) model based on transshipment model developed by Papoulias and Grossmann (1983). They were able to generate HEN configurations automatically under the utilization of computer software rather than designing HEN manually. The objective is to target minimum number of heat exchangers and minimum total annualized cost (TAC) sequentially. Yee *et al.* (1990) introduced a stage-wise simplified superstructure formulation to solve HEN optimization problems and the model was later extended by Yee and Grossmann (1990). Chen and

Hung (2004) proposed a three-step sequential approach. The problem is decomposed into three main iterative steps: HEN synthesis based on stage-wise superstructure, flexibility analysis, and improvement of unqualified networks. Apart from sequential approach, Aaltola (2002) adopted the stage-wise superstructure model in order to apply with multiperiod problems without sequential decomposition. This caused the optimized solution have tendency not to stuck in local optima. Modification to Aaltola's model was done by Verheyen and Zhang (2006) to improve performance of the model and become more practical in real problems.

The purpose of this research is to study and develop HEN formulation for multiperiod design for using mathematical programming based on stage-wise superstructure of Yee and Grossmann (1990). The model will be developed targeting minimum utility and capital cost_m simultaneously. Two techniques of multiperiod HEN synthesis, which are sequential and simultaneous approaches, will be proposed and compared by using adapted case study of (Verheyen and Zhang, 2006). The most effective method will be applied to an industrial case study of CDU to see the performance of the method when dealing with larger problem. Moreover, the final results will be simulated in PRO/II to ensure its operability in real process.

CHAPTER II

LITERATURE REVIEW

2.1 Heat Exchanger Network Design Methodologies

The world's energy price has been increasing since 1970s and it has been considered one of important issues especially in chemical engineering processes. Before that period of time, most chemical plants mainly focused on capital investment cost and tried to keep the cost as low as possible. When energy crisis took place, many efforts have been made to reduce the energy consumption in the plants to reduce operational cost. There are many methods to improve energy efficiency in processes such as adjusting operating condition, changing or modifying equipment, using other chemicals, etc. One of commonly used techniques is installation of heat exchanger network (HEN) which can recover excess energy from heat source to heat sink.

Chemical plants generally involve with thermal processes because heat is a critical factor. For instance, it helps in catalyzing reactions or being used to heat up the products in separation unit corresponding to the boiling point. In HEN synthesis, streams are divided into two major types which are hot streams that need to be cooled and cold streams that need to be heated. A heat exchanger will match at least one hot stream and one cold stream in order to transfer excess heat from the hot stream (heat source) to the cold stream (heat sink). Thus, HEN is defined as a network that consists of many heat exchangers integrated in a system in order that all target temperatures are satisfied. The configuration can be generated in various possibilities such as heat exchanger in series, parallel, stream bypassing, stream splitting, etc. HEN will satisfy target temperatures of every stream incorporating with hot and/or cold utility.

The early methods for synthesizing HEN were based on thermodynamic principles and heuristic methods and then were developed to be more systematic with the implementation of mathematical computer programming (Verheyen and Zhang, 2006). To explain the evolution of HEN synthesis, the proposed methodologies will

be categorized into three major groups which will be reviewed and compared in the next sections.

2.1.1 Pinch Analysis Concept

Pinch analysis was first introduced by Hohmann (1971) and then was refined and published by Linnhoff and Flower (1978). It is used for preliminary prediction of maximum energy recovery (or minimum utility required) and minimum number of heat exchangers corresponding to streams data (temperatures and heat capacity flowrates) and a chosen minimum temperature difference (ΔT_{\min}). In HEN research area, pinch analysis is widely known as a thermodynamic analysis applying first and second laws of thermodynamics (Verheyen and Zhang, 2006).

A diagram called composite curve is used as a representation of all stream data. An example is shown in Figure 2.1. The upper and lower lines are hot and cold composite curves, respectively. The overlapping region represents the amount of energy recovered within the process. The non-overlapping region on both sides represents minimum hot (right) and cold utility (left) requirement at a chosen ΔT_{\min} .

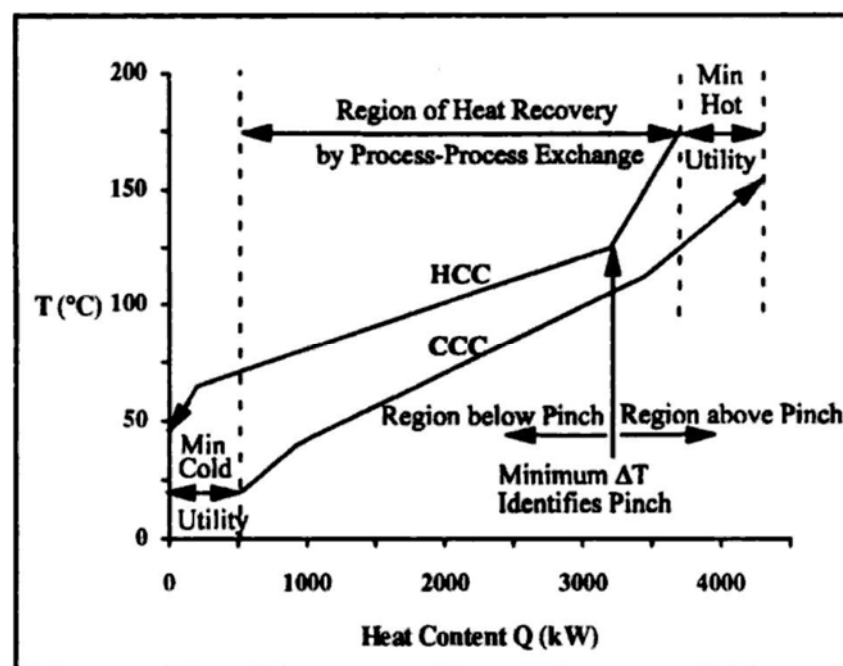


Figure 2.1 Hot and cold composite curves (Shenoy, 1995).

For a given ΔT_{\min} , pinch is located on the diagram where the distance between hot and cold composite curves is narrowest and the temperature difference between two curves is equal to the ΔT_{\min} . The composite curve is separated into two subsystems: above pinch and below pinch. The HEN design of each subsystem must be done separately starting from pinch location because there are more designing rules and constraints. Once we finish creating the network at pinch, the constraints are more relaxed and there is more flexibility to place a match depending on one's judgment.

Three pivotal rules of pinch analysis are described following:

- Heat transfer is not allowed across pinch.
- Each heat exchanger must have temperature difference larger than ΔT_{\min} .
- Hot and cold utility are placed only at the end of the streams below and above pinch respectively in case target temperature is not reached.

As pinch design method creates a structure of HEN, some heat exchanger might have too large area, too small area, or higher number of exchangers than predicted one. Loop and path technique can be applied for energy relaxation in order to change heat exchanger area or remove some heat exchangers in case of unsatisfaction and high degree of complexity. However, it will result in increment of utility usage.

In reality, the objective of HEN synthesis is to minimize total annualized cost (TAC) which is the sum of utility cost per year and annualized capital (area and equipment) cost. If ΔT_{\min} decreases (more energy recovery), the minimum utilities required will also decrease whereas the heat exchanger area needed is increased. Therefore, a trade-off between utility cost and area cost should be taken into account in order to find the optimized ΔT_{\min} as shown in Figure 2.2. This is so-called supertargetting.

Pinch analysis is a concept that is easy to comprehend because it is a graphical method. Many sequential methods take advantage of this method for obtaining minimum utility and minimum number of units. The method, however, has

some critical drawbacks. It does not deliberate heat transfer coefficients and heat exchanger areas properly; therefore, it might lead to ineffective solutions somehow (Verheyen and Zhang, 2006).

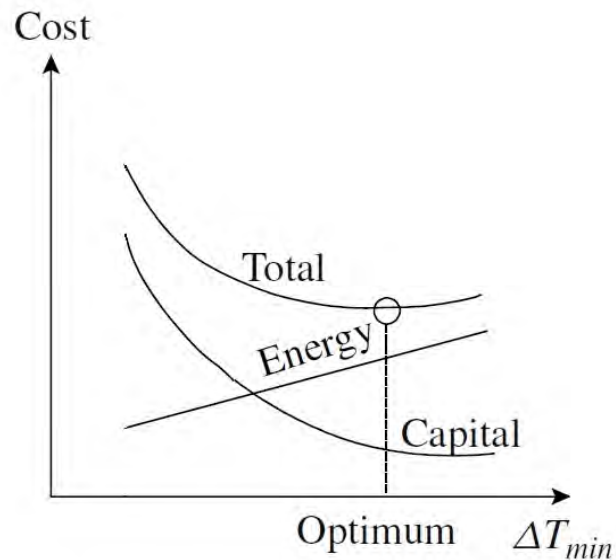


Figure 2.2 Economic trade-off between energy cost and capital cost (Shenoy, 1995).

2.1.2 Sequential Approaches

When mathematical programming initially caught considerable attention from researchers, the existing computers did not have adequate performance; in addition, optimization techniques had not been developed well enough. The procedures, as a result, were decomposed into several subproblems and then solved step by step. This method is known as sequential approach. In general, HENs optimization is mostly decomposed to these three subproblems (Biegler *et al.*, 1997):

- Minimum utility cost
- Minimum number of units
- Minimum total investment cost

Papoulias and Grossmann (1983) proposed a transshipment model. There are two main steps. The first step is linear programming (LP) problem which subjects to minimum utility cost. Second, the number of units is minimized using mixed integer linear programming (MILP). The concept of transshipment model is to

distribute heat source (hot streams and hot utilities) to heat sink (cold streams and cold utilities). All streams are divided into temperature intervals. From Figure 2.3, each temperature interval (warehouse) will receive heat from higher temperature interval or hot utilities and then distribute to every cold stream in the same interval. The heat remained will be cascaded to lower temperature interval. Because the amount of heat source, heat sink, and temperatures are fixed, heat residuals which are passed to the next interval is the only one design variable to be optimized by LP model.

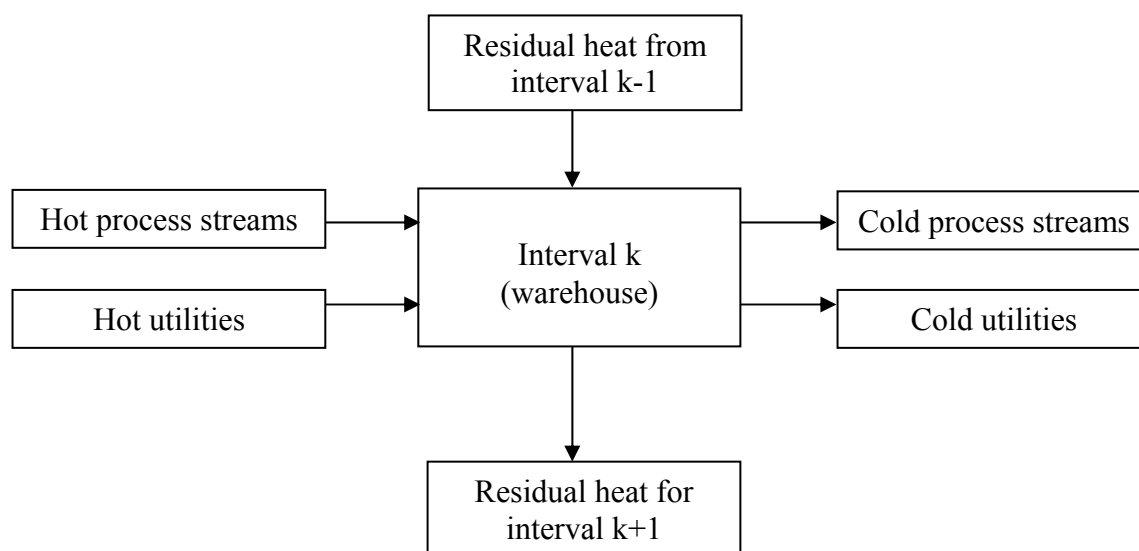


Figure 2.3 Heat flows in interval k (Biegler *et al.*, 1997).

For the MILP of transshipment model which is subject to minimum number of units, the formulation is analogous to LP model except that binary integer is used and denoted as the existence of a heat exchanger.

The extension of the transshipment model was addressed by Floudas *et al.* (1986). The HEN configurations will be generated automatically by using computer programming. Network synthesis is derived starting with LP transshipment model to predict minimum utility cost and pinch point will divide temperature range into subnetworks. Then the fewest number of exchangers is minimized by MILP transshipment model. The solution also provides the amount of heat exchanged in each match. Next step, a superstructure is derived for each subnetwork corresponding

to those matches which are predicted by MILP transshipment model. The superstructure embeds various alternative configurations such as stream splitting, bypassing, matches in series, matches in parallel, matches in series-parallel, matches in parallel-series, etc. The superstructure is modified by NLP formulation featuring minimum investment cost. Note that the heat loads are treated as fixed parameter but flowrates and temperatures are design variables. However, as this methodology is a sequential method, the solutions of HENs might have been led to sub-optimal solutions because some good feasible solutions might not be included in search space of another subproblem.

A new decomposition method was introduced by Zhu (1995) for the purpose of automated synthesis of HENs using block decomposition and heuristic rules. The concept is to simplify a problem by decomposing composite curve into a number of blocks. In each block, two straight lines (for hot and cold composite curves) represent an enthalpy interval. Those two lines are called quasi-composite curves. Then, the design is performed using area targeting and newly heuristic rules for match selection. The final design with cost optimization is obtained via NLP model.

2.1.3 Simultaneous Approaches

As time progressed, a number of sequential approaches had been proposed, many researchers attempted to solve those decomposed problems simultaneously accompanying with modern computer technology and higher performance of optimization techniques. Since the simultaneous techniques consider all trade-offs, i.e. area targeting, number-of-unit targeting, and utility cost, their formulations are mostly MINLP model. They usually give more preferable results more than sequential methods, but they have more complexity which is the major problem of these approaches. In other words, they are mostly nonlinear, non-convex, and non-continuous that can mislead to locally optimal solutions. Therefore, some assumptions should be made in order to reduce the complexity and size of models.

Ciric and Floudas (1991) mentioned that the decomposition approach can lead to uncertainty of the optimality of final HEN. That means globally optimal solution is not guaranteed and most cases showed that the solutions tended to be

local optima. From those reasons, they proposed a HEN synthesis method using MINLP simultaneous technique. The model includes hyperstructure of Floudas and Ciric (1989) and modified transshipment model by Papoulias and Grossmann (1983). Many different configurations of HEN are embedded in the hyperstructure as shown in Figure 2.4. The authors illustrated two cases of HEN synthesis which began with different assumptions so that the designs are differentiated by pinch point. One is strict-pinch design, another one is pseudo-pinch design. The comparison between two designs demonstrated that a pseudo-pinch approach which allows heat to transfer across pinch point leads to more desirable HENs.

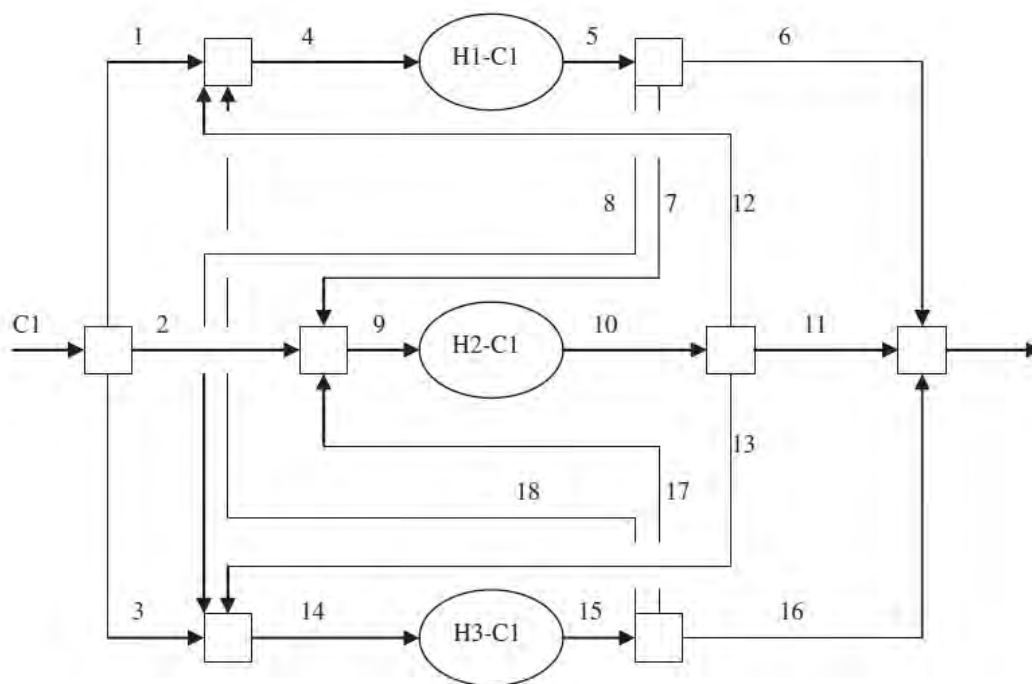


Figure 2.4 Hyperstructure of Floudas and Ciric (1989) (Verheyen and Zhang, 2006).

In the meantime, a stage-wise simplified superstructure was developed by Yee *et al.* (1990). The schematic is shown in Figure 2.5 but its detailed formulations will be written in the next section. Unlike transshipment model, the superstructure neither relies on pinch design method nor division into temperature intervals. For the superstructure, all constraints will be linear which results in rigorous model since the model was simplified by making following assumptions:

- Isothermal mixing; temperatures in location k must be equal in every split stream and mixed together before entering to the next stage.
- no split stream flowing through more than one heat exchanger
- hot and cold utilities are placed at the end of stream
- no stream bypass

The idea of the simplified superstructure model is to partition the structure into a number of stages. The illustration of two hot and two cold streams system is shown in Figure 2.5. In each stage, hot streams are split up into a number of cold streams and cold streams are similarly split into a number of hot streams. Every hot and cold split stream will be paired in all possible matches. One match represents a heat exchanger which is expressed by a binary integer and a set of matches will be chosen during optimization.

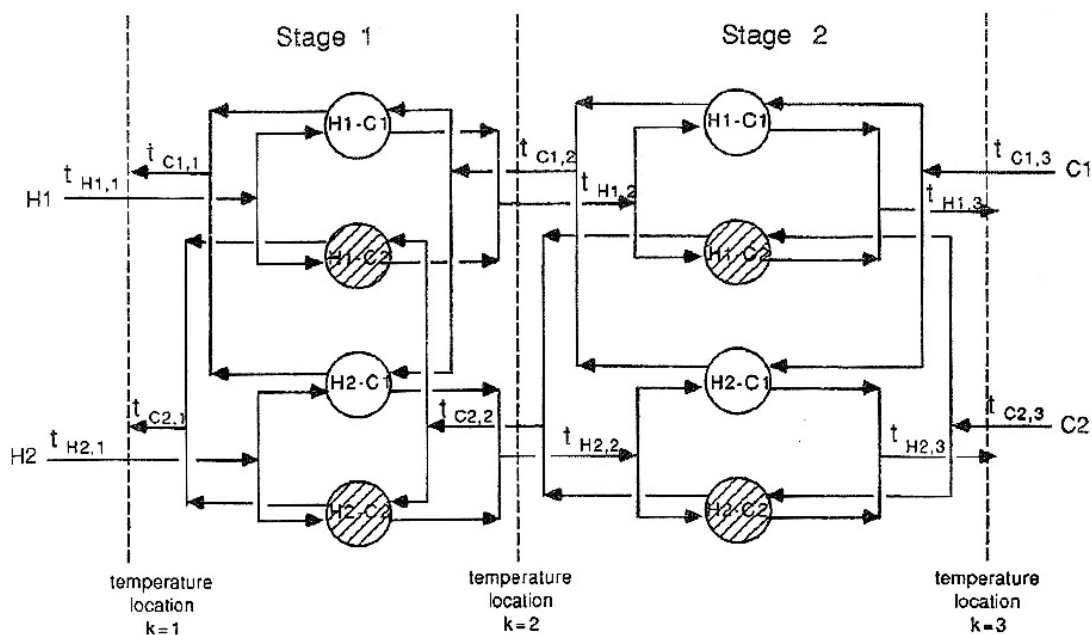


Figure 2.5 Stage-wise superstructure model (Yee *et al.*, 1990).

As the number of stages is concerned, it can be arbitrarily selected by designers; however, there is a rule of thumb stating that it is commonly chosen to be

equal to either maximum number of hot and cold streams. Anyway, increasing of the number of stages may give better value of objective function or it may cause no effect on that.

Floudas (1995) had observed some weaknesses of superstructure. He pointed out that there are several network configurations which are excluded from the model. As can be seen in Figure 2.6, one branch cannot have more than one heat exchanger in series. Moreover, bypassing from one branch to another is not allowed in a stage and the combination of those two features is not available either.

In spite of its limitations, an important strength of the simplified superstructure is that all equations and constraints are linear except the objective function which is nonlinear due to the area calculation terms. As a result, the model was later extended in a number of studies.

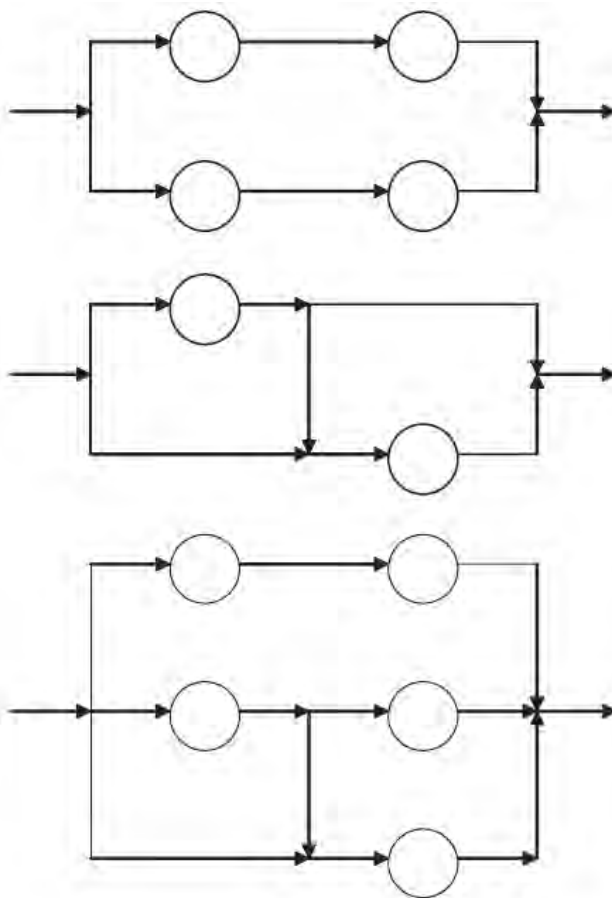


Figure 2.6 Excluded HEN configuration of simplified superstructure (Verheyen and Zhang, 2006).

One of the extensions of Synheat (another name of stage-wise superstructure) model was developed by Björk and Westerlund (2002). The main purposes of this work are elimination of the isothermal mixing assumption, which leads to a significant increase in number of variables and constants, together with using global optimization technique. Rather than using branching procedure, the strategy of this technique is try to convexify the non-convex terms in area equations so that when all equations in the problem are convex, global optimization can be obtained.

2.2 Multiperiod Heat Exchanger Network

In the past, most HENs were synthesized for only a fixed condition without taking into account the changes in parameters. Such HENs might cause deviation from optimal solution, or even no longer be operable or stay out of feasible region. The possible changes in operating conditions arise from two major reasons. First, uncertainties or unintentional changes around one nominal value, this kind of change is known as a resilient problem such as malfunction of process control system. Second, periodic changes can be either seasonal changes or intentional changes such as multiple feeds and a need for higher temperature in operating condition due to deactivation of catalyst.

2.2.1 Degree of Flexibility

As multiperiod HENs had been studied since 1980s, some efforts to create an index indicating flexibility of such HEN designs were introduced. For example, Saboo *et al.* (1985) proposed the resilient index (RI). It is a quantitative index which is used to compare between different HENs and guide the most potential candidate. At the same time, a flexibility index was proposed by Swaney and Grossmann (1985). The flexibility index indicates maximum deviation of uncertain variable and also lying in the operable or feasible region. Moreover, it provides information of critical points that restrict the design. From Figure 2.7, the point at the center of feasible region denotes as the nominal values of uncertain parameters. Each

rectangular represents the maximum deviation of each parameters (θ_1 , θ_2) while remaining in the feasible region (R).

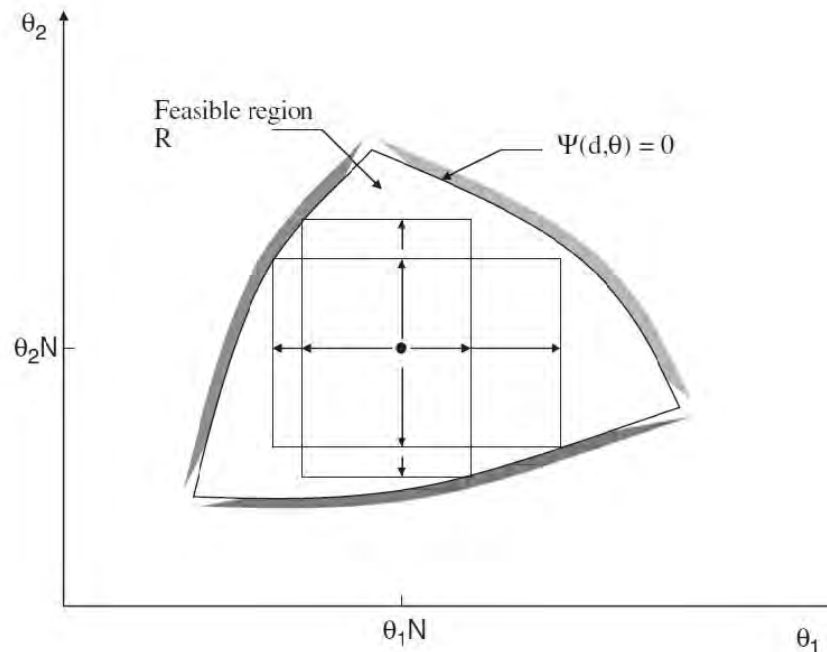


Figure 2.7 Feasible region of operation (Verheyen and Zhang, 2006).

2.2.2 Review of Multiperiod HEN Designs

There has been an increased interest in multiperiod features of chemical process designs including HENs since 1980s. One of renowned efforts was made by Floudas and Grossmann (1986). They took advantage of their work dealing with fixed conditions by using the same principles to apply to multiperiod problems. To recall the concept, LP transshipment model will be solved to predict minimum utility requirement and then MILP transshipment model is used to synthesize HEN configuration for each period featuring minimum utility cost and fewest number of units. The final network that satisfies all operational periods is obtained by assembling each topologies of each period manually. However, there are two major drawbacks for this method: sizing heat exchangers and bypassing around heat exchanger are not performed.

The model discussed above was improved by Floudas and Grossmann (1987) to overcome those limitations. They adopted the strategy proposed by Floudas *et al.* (1986), i.e. LP/MILP transshipment model is carried out, then the superstructure based on topology from transshipment model is derived and NLP formulations is run to improve the design afterwards. In case of multiperiod design, the only one difference is that all procedures have to be done for each period and the final solution is generated by integration of each subnetwork. In addition, when solving an NLP model, there are actually a lot of variables and constraints because all possible interconnections for the matches are taken into account. Hence, graph representation was introduced in order to reduce the problem size.

Iyer and Grossmann (1996) proposed an NLP model to find multiperiod HENs design with initially fixed configuration. The algorithm for global optimization by Quesada and Grossmann (1993) was used. Briefly, the objective function of the model is discrete function due to Max operation; thus, it is not guaranteed that the solution will be global optimum. To solve the problem, some constraints must be added using concept of convex underestimators to alter from non-convex objective function to the convex one.

A simultaneous MINLP model was developed by Aaltola (2002) based on superstructure of Yee and Grossmann (1990) which does not rely on pinch point. The objective function includes utility cost, area cost, and capital cost of units where assumption of average area of all periods is ruled to maintain linearity of the objective function. LP/NLP search algorithm is applied in the next step to improve the system. There are four main purposes of this step: trade-offs between utility and area cost, eliminating bypass streams which increase complexity of the network, removing the impractically average area assumption, and eliminating an isothermal mixing assumption.

From the model proposed by Aaltola (2002), Verheyen and Zhang (2006) observed its weaknesses and presented a new improved model. The new model comprises of a simultaneous MINLP model with maximum area formulation in the objective function and an improved NLP model in which slack variables and weighed parameter are not included.

Chen and Hung (2004) also studied simultaneous synthesis of multiperiod HENs and adopted the flexibility test using flexibility index (Swaney and Grossmann, 1985) to examine the solution whether it is qualified or not. The MINLP model based on Yee and Grossmann (1990) and extended model by Aaltola (2002) were applied using finite number of extreme operating conditions, i.e. the conditions that have tendency to demand the heat exchanger area as large as possible. The network that can be used for those operating conditions is tested by flexibility index to check ability of full-range operations. If the network is qualified, the synthesis procedure will be terminated. But if the network is unqualified, one more iteration has to be performed while some constraints are added to prevent from attaining the same disqualified network or, in other words, to reduce search space.

Recently Ma *et al.* (2008) pointed out that the models of Aaltola (2002) and Verheyen and Zhang (2006) have difficulty when solving more complex problems such as increase in number of periods. Such models give over-synthesized networks for all operational periods. Two-stage method was introduced for multiperiod operation. In the first stage, temperature-enthalpy (T-H) diagram is used to synthesize an over-synthesized HEN based on the stream pseudo-temperature. Compared with MINLP superstructure model, this method has less complexity and smaller size. Moreover, it can guarantee the feasibility of the initial solution to be used in the second stage. In the second stage, the over-synthesized HEN is improved. An area which is less than maximum area from the initial solution will be optimized. The main idea is that instead of using the maximum areas of each period, the optimal areas which are not satisfied in some periods, but the insufficient required area will be compensated by utilities. Genetic/simulated annealing (GA/SA) algorithm is also applied to guarantee global optimization at high probability.

2.2.3 Simultaneous MINLP Model

In this work, a model based on stage-wise simplified superstructure by Yee and Grossmann (1990) and also the extension to multiperiod version of Verheyen and Zhang (2006) will be applied because it is a rigorous model without decomposition and can provide good results (Verheyen and Zhang, 2006). The overall concept of the simplified superstructure has been explained in previous

section. Isothermal mixing assumption will be applied initially because it significantly helps decrease a number of variables such as temperatures and heat capacity flowrates of outlet each heat exchanger in each branch. A single period model actually resembles a multiperiod model except that one more index, p , referring to period of operation is introduced.

2.2.3.1 Area Calculations

The heat transfer area for both process-process heat exchanger and utility-process heat exchanger can be calculated using these following information: heat load of each match, inlet and outlet temperatures of both hot and cold streams, and overall heat transfer coefficient for each match. The variables are illustrated in Figure 2.8. Area calculation is performed using Eq. 2.1.

$$A(i, j, k, p) = \frac{q(i, j, k, p)}{LMTD(i, j, k, p) \cdot U(i, j)} \quad (2.1)$$

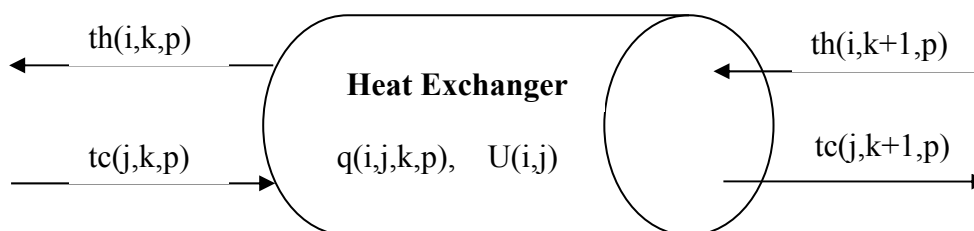


Figure 2.8 Illustration of variables involving in a heat exchanger.

The log mean temperature difference is calculated as Eq. 2.2:

$$LMTD(i, j, k, p) = \frac{[th(i, k, p) - tc(j, k, p)] - [th(i, k + 1, p) - tc(j, k + 1, p)]}{\ln \frac{[th(i, k, p) - tc(j, k, p)]}{[th(i, k + 1, p) - tc(j, k + 1, p)]}} \quad (2.2)$$

This formulation is generally used when hand calculation is performed. However, the difficulties probably come up in case of using mathematical

programming. The reason is that there might be one or more set of variables in search space in which the temperature difference on both sides of heat exchanger are the same values. In consequence, the division by zero value will take place and cause the set of variables cannot be used due to undefined solution, but actually it is operable in realistic. For this reason, several approximations for LMTD were proposed as follows:

- Average LMTD
- Paterson approximation
- Chen approximation

The average LMTD is a simple one. The temperature differences of both sides are just averaged as shown in Eq. 2.3:

$$LMTD(i, j, k, p) = \frac{[th(i, k, p) - tc(j, k, p)] + [th(i, k + 1, p) - tc(j, k + 1, p)]}{2} \quad (2.3)$$

This approximation is very rough; therefore, it results in large error when comparing with the actual value of LMTD. The error will be amplified as the temperature differences of both sides are not in the same order of magnitude. Then the predicted area will be under-estimate.

The second approximation was proposed by Paterson (1984) as shown in Eq. 2.4:

$$\begin{aligned} LMTD(i, j, k, p) &= \frac{2}{3} \cdot [(th(i, k, p) - tc(j, k, p)) \cdot (th(i, k + 1, p) - tc(j, k + 1, p))]^{0.5} \\ &+ \frac{1}{6} [(th(i, k, p) - tc(j, k, p)) + (th(i, k + 1, p) - tc(j, k + 1, p))] \end{aligned} \quad (2.4)$$

This approximation gives slightly over-estimate LMTD or under-estimate heat transfer area.

Lastly, the approximation was introduced by Chen (1987) as formulated in Eq. 2.5:

$$\begin{aligned}
 &LMTD(i, j, k, p) \\
 &= \left[(th(i, k, p) - tc(i, k, p)) \cdot (th(i, k + 1, p) \right. \\
 &\quad \left. - tc(j, k + 1, p)) \cdot \frac{(th(i, k, p) - tc(j, k, p)) + (th(i, k + 1, p) - tc(j, k + 1, p))}{2} \right]^{1/3}
 \end{aligned}
 \tag{2.5}$$

The LMTD value of Chen approximation tends to be underestimated or gives underestimated area. A good point of this approximation is that in case the temperature differences on both sides of heat exchanger are zero, the approximation equation will give a zero value. The accuracy of each proposed approximation is illustrated compared to actual LMTD value in Figure 2.9. It demonstrates that the average LMTD greatly deviates from the actual value of LMTD while Paterson's and Chen's approximations are much more accurate. In this work, Chen's approximation is selected because not only it predicts LMTD precisely, but also the over-estimated heat exchanger area can be considered as reserved area to assure capability of operation according to the designed networks.

2.2.3.2 Multiperiod MINLP Model Formulation

In this section, all equations in multiperiod MINLP model by an optimization program called GAMS (General Algebraic Modeling System) are presented. Those equations are not able to be put in the program directly, but they are needed to be converted into GAMS's language appropriately. The objective function of the model is to minimize total annualized cost (TAC) comprised of utilities cost, heat exchanger areas cost, and capital cost of heat exchangers. Binary variable will be used to determine the existence of each match for heat exchanger. The design variables are temperatures at every location in stage model and heat loads of each heat exchanger.

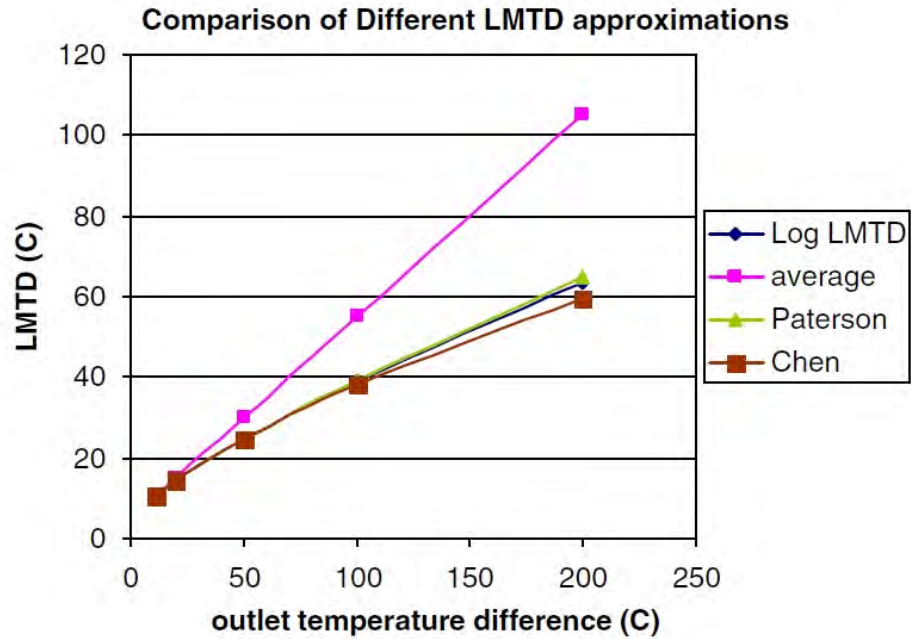


Figure 2.9 LMTD Approximation (Verheyen and Zhang, 2006).

Overall stream heat balances are performed to make sure that the total heat load required for each stream is sufficient in each period. The heat balance equations for both hot and cold streams are shown in Eq. 2.6 and Eq. 2.7.

$$\begin{aligned}
 & [Th_{in}(i, p) - Th_{out}(i, p)]. FCph(i, p) \\
 & = \sum_{k \in ST} \sum_{j \in CP} q(i, j, k, p) + q_{cu}(i, p) \quad i \in HP, p \in PR \quad (2.6)
 \end{aligned}$$

$$\begin{aligned}
 & [Tc_{in}(j, p) - Tc_{out}(j, p)]. FCpc(j, p) \\
 & = \sum_{k \in ST} \sum_{i \in HP} q(i, j, k, p) + q_{hu}(i, p) \quad j \in CP, p \in PR \quad (2.7)
 \end{aligned}$$

According to the isothermal mixing assumption, temperatures at each location for both sides of heat exchangers within a stage have to be equal. They are calculated from total amount of heat added or rejected in each stream as shown in Eq. 2.8 and Eq. 2.9 for hot and cold streams, respectively.

$$\begin{aligned}
& [th(i, k, p) - th(i, k + 1, p)].FCph(i, p) \\
& = \sum_{j \in CP} q(i, j, k, p) \quad k \in ST, i \in HP, p \in PR \quad (2.8)
\end{aligned}$$

$$\begin{aligned}
& [tc(j, k, p) - tc(j, k + 1, p)].FCpc(j, p) \\
& = \sum_{i \in HP} q(i, j, k, p) \quad k \in ST, j \in CP, p \in PR \quad (2.9)
\end{aligned}$$

The target temperatures of all streams in each period will be assigned to the first location (location = 1) for hot streams and the last location (location = NOK+1) for cold streams as formulated as Eq. 2.10 and Eq. 2.11.

$$Th_{in}(i, p) = th(i, 1, p) \quad i \in HP, p \in PR \quad (2.10)$$

$$Tc_{in}(j, p) = tc(j, NOK + 1, p) \quad j \in CP, p \in PR \quad (2.11)$$

In the stage-wise superstructure model, there must be monotonic increase or decrease in temperature. In other words, temperatures at the left-side location will always be greater than the right-side for every stream. That means the temperature of hot stream decreases continuously until it reaches the target temperature at the outlet. For cold streams, in other way round, temperature increases continuously because of heat received from hot stream until it reaches the desire temperature at the outlet. The formulations are shown in Eq. 2.12 and Eq. 2.12.

In case the temperature at the last location (for hot stream) or the first location (for cold stream) does not reach its target temperature, cold utility or hot utility have to be utilized respectively. Therefore, for hot streams, the outlet temperature at last location will be greater than or equal to target temperature. For cold streams, the outlet temperature at the first location will be less than or equal to target temperature. The equations are shown in Eq. 2.13 and Eq. 2.14.

$$th(i, k, p) \geq th(i, k + 1, p) \quad k \in ST, i \in HP, p \in PR \quad (2.11)$$

$$tc(j, k, p) \geq tc(i, k + 1, p) \quad k \in ST, j \in CP, p \in PR \quad (2.12)$$

$$th(i, NOK + 1, p) \geq Th_{out}(i, p) \quad i \in HP, p \in PR \quad (2.13)$$

$$TC_{out}(j, p) \geq tc(j, 1, p) \quad j \in CP, p \in PR \quad (2.14)$$

Energy balances are also performed to find utility loads required to make temperatures of each stream reach its target temperature. The following equations are used. The equations are shown in Eq. 2.15 and Eq. 2.16.

$$[th(i, NOK + 1, p) - Th_{out}(i, p)].FCph(i, p) = qcu(i, p) \quad i \in HP, p \in PR \quad (2.15)$$

$$[TC_{out}(j, p) - tc(j, 1, p)].FCpc(j, p) = qhu(j, p) \quad j \in CP, p \in PR \quad (2.16)$$

As binary variables is used to represent the existence of matches. There has to be logical constraints which are conducted to determine values of binary variables. If a match takes place or there is heat load, binary variable (z) is forced to be unity and heat load (q) is controlled to be less than its upper bound. But if there is no heat load ($q=0$), the value of z can be either 0 or 1. In fact, it should be only 0 because it is impossible to have heat exchanger with no heat load. However, it may be 0 since the model tries to minimize overall number of exchangers. Such constraints for process-process exchangers and utility exchangers are shown as Eq. 2.17, Eq.2.18, and Eq. 2.19.

$$q(i, j, k, p) - Q_{up} \cdot z(i, j, k) \leq 0 \quad i \in HP, j \in CP, k \in ST, p \in PR \quad (2.17)$$

$$qcu(i, p) - Q_{up} \cdot zcu(i) \leq 0 \quad i \in HP, p \in PR \quad (2.18)$$

$$qhu(j, p) - Q_{up} \cdot zhu(j) \leq 0 \quad j \in CP, p \in PR \quad (2.19)$$

Where $z(i, j, k), zcu(i), zhu(i) \in \{0,1\}$

Driving force for heat transfer is temperature difference. To assure their feasibility of both sides, i.e. cold stream temperature should not be greater than hot stream and the temperature difference should be high enough so that the heat exchanger area will not be too large, following constraints must be included as shown in Eq.2.20, Eq.2.21, and Eq.2.22. When a match exists, the temperature difference is forced to be higher than a certain value which usually equal to exchanger minimum approach temperature (EMAT). Therefore, Eq. 2.23 must be included.

$$dt(i, j, k, p) \leq th(i, k, p) - tc(j, k, p) + DT_{up} \cdot (1 - z(i, j, k))$$

$$i \in HP, j \in CP, k \in ST, p \in PR \quad (2.20)$$

$$dt(i, j, k + 1, p) \leq th(i, k + 1, p) - tc(j, k + 1, p) + DT_{up} \cdot (1 - z(i, j, k))$$

$$i \in HP, j \in CP, k \in ST, p \in PR \quad (2.21)$$

$$dtcu(i, p) \leq th(i, NOK + 1, p) - Tcu_{out} + DT_{up} \cdot (1 - zcu(i))$$

$$i \in HP, p \in PR \quad (2.22)$$

$$dt(i, j, p) \geq EMAT \quad (2.23)$$

To identify required areas of each match that is used in all periods, several constraints are included. The aim of these constraints is to find maximum area among all considered periods in which it can be operable for all periods. The formulations are shown in Eq. 2.24, Eq. 2.25, and Eq. 2.26.

$$Amax(i, j, k) \geq \frac{q(i, j, k, p)}{LMTD(i, j, k, p) \cdot U(i, j)} \quad i \in HP, j \in CP, k \in ST, p \in PR$$

$$(2.24)$$

$$Ahumax(j) \geq \frac{qhu(j,p)}{LMTD(j,p).U(hu,j)} \quad j \in CP, p \in PR \quad (2.25)$$

$$Acumax(i) \geq \frac{qcu(i,p)}{LMTD(i,p).U(i,cu)} \quad i \in HP, p \in PR \quad (2.26)$$

Finally, the objective function is formulated to calculate the total annualized cost (TAC) consisting of all fixed costs (capital cost and maximum area cost) and operating costs (hot and cold utility costs) as shown in Eq. 2.27. The objective function will be asked to minimize by varying design variables within search space.

Objective function = minimize TAC

$$\begin{aligned} &= AF \left[\sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C_f \cdot z(i,j,k) + \sum_{i \in HP} \sum_{CU} C_f \cdot zcu(i) \right. \\ &\quad \left. + \sum_{j \in CP} \sum_{HU} C_f \cdot zhu(j) \right] + AF \cdot \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C \cdot Amax(i,j,k)^B \\ &\quad + AF \cdot \sum_{j \in CP} C \cdot Ahumax(j)^B + AF \cdot \sum_{i \in HP} C \cdot Acumax(i)^B \\ &\quad + \sum_{p \in PR} \frac{DOP(p)}{NOP} \cdot \sum_{i \in HP} C_{cu} qcu(i,p) \\ &\quad + \sum_{p \in PR} \frac{DOP(p)}{NOP} \cdot \sum_{j \in CP} C_{hu} qhu(j,p) \end{aligned} \quad (2.27)$$

2.3 Heat Integration in Crude Distillation Unit (CDU)

2.3.1 Crude Oil Refinery

Petroleum refinery process begins with crude oil distillation, and then the products will be treated in subsequent processes, such as hydrotreating, catalytic cracking, etc., to recover more valuable products as much as possible. It produces variety of petroleum products such as light gas, light naphtha, heavy naphtha, kerosene gasoil, and residue. There are mainly two types of distillation column which are atmospheric distillation unit (ADU) and vacuum distillation unit (VDU). Additionally, one of important equipment is desalter. It is considered as a part of atmospheric distillation unit facilities. It is used for removing salts, solids, and water from crude oil before entering to the distillation column to prevent damages. In Figure 2.10, the intermediate petroleum products will be separated based on different boiling points. The heavy products, i.e. atmospheric residue, will be sent further to VDU.

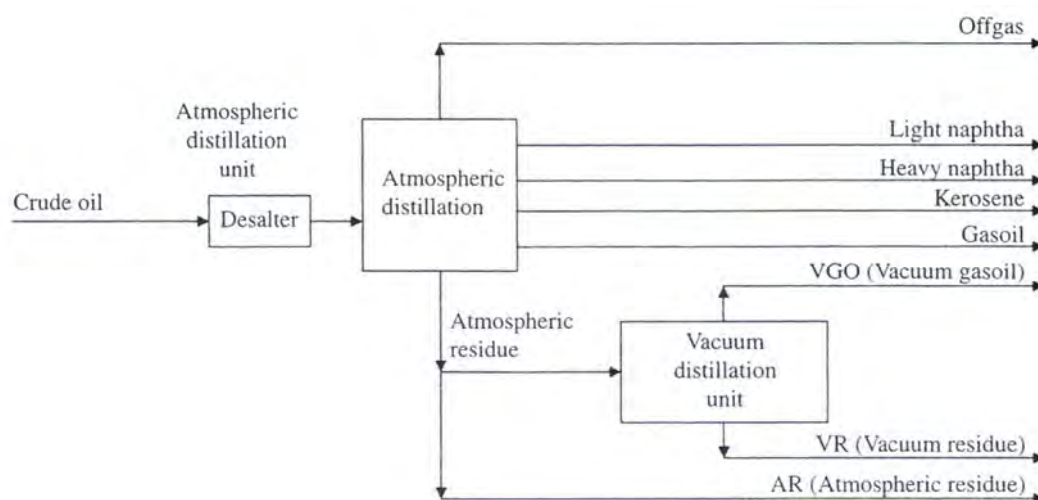


Figure 2.10 A crude distillation complex (Petroleum, 2000).

2.3.2 Crude Oil Assay

Crude oil is complex mixture of hydrocarbon composed of millions of compounds, most of which cannot be identified. Only light compounds such as methane, ethane, propane, and benzene can be identified. It also has some impurities,

for instance, sulfur, oxygen, nitrogen, and metal. These compositions and other properties of particular crude oil are declared in a crude assay. One of important characteristics that affect the design of ADU is true boiling point (TBP) distillation (Petroleum, 2000). This property is specific for each crude oil as shown in Figure 2.11. It illustrates the product yield in percentage of cumulative volume distilled according to its TBP cut range determined by product specifications and market demand. Moreover, there are other properties reported in a crude assay such as API gravity (specific gravity used in petroleum industry), flash point, and sulfur content.

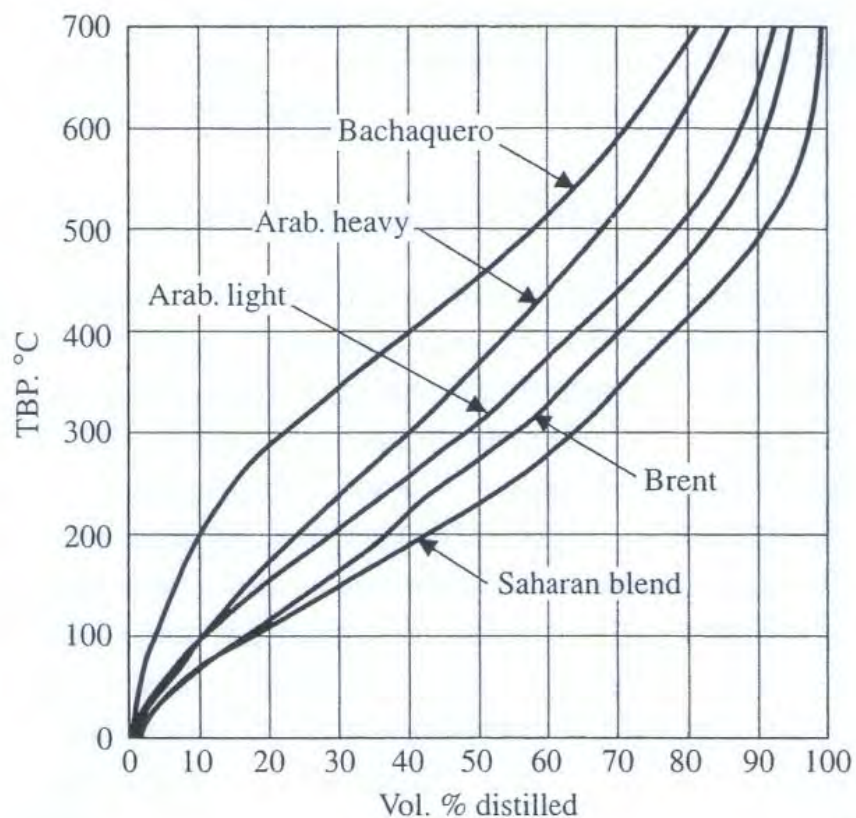


Figure 2.11 TBP distillation curve of different crude oils (Petroleum, 2000).

2.3.3 Heat Integration in Crude Distillation Unit

Atmospheric distillation unit is one of the largest energy consuming units in petroleum refinery plant. Generally, cold streams are crude oil feed and hot streams are the intermediate products leaving from the distillation column. After the crude oil is preheated, it will be sent to furnace to be heated up to the desired

temperature around 310-370°C (Petroleum, 2000). The example for heat integration in atmospheric distillation unit process flow diagram is shown in Figure 2.12. Because crude oil characteristics are spontaneously uncertain in each reservoir, it causes temperatures and heat capacity flowrates varying over the period. Therefore, multiperiod heat exchanger network design is essential so that the HEN will be flexible and can be operable efficiently for every type of crude oil.

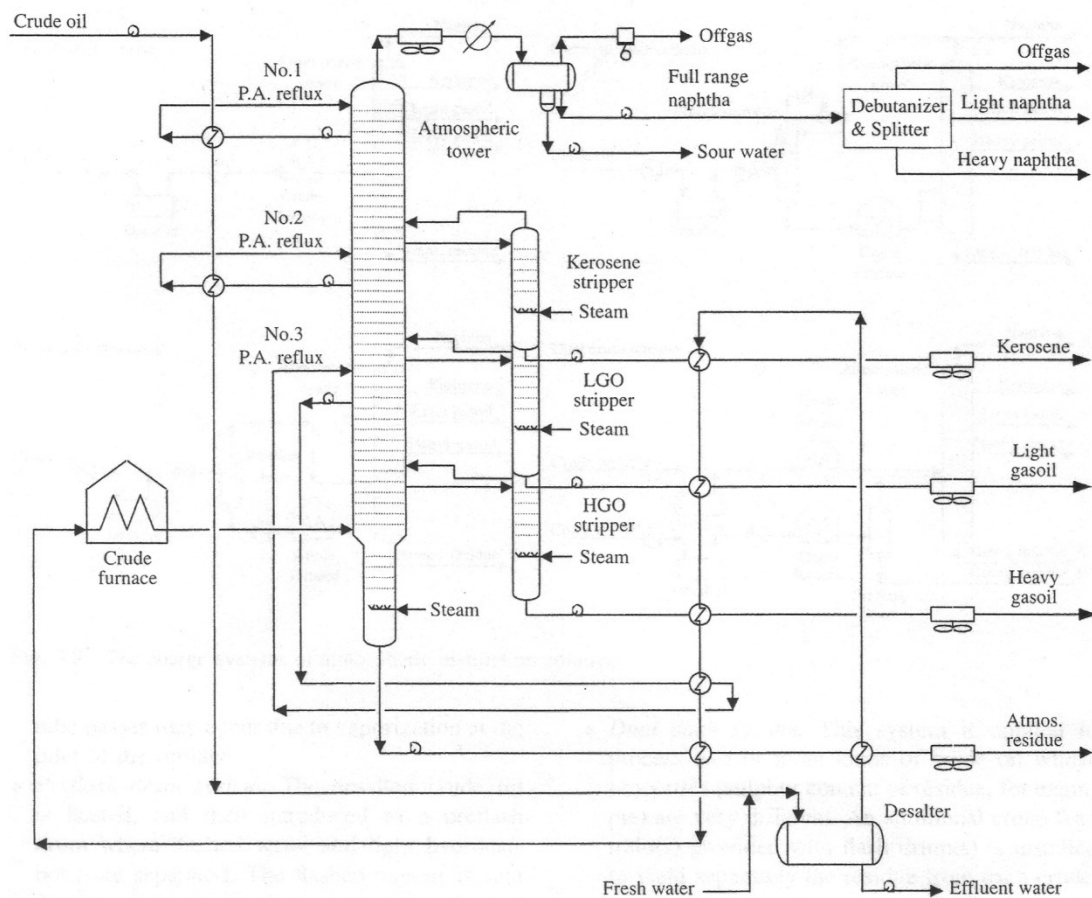


Figure 2.12 CDU process flow diagram (Petroleum, 2000).

- Aaltola, J. (2002). Simultaneous synthesis of flexible heat exchanger network. Applied Thermal Engineering 22(8), 907-918.
- Biegler, L.T., Grossmann, I.E. and Westerberg, A.W. (1997). Systematic Methods of Chemical Process Design. Upper Saddle River (US): Pearson Education.
- Björk, K.-M. and Westerlund, T. (2002). Global optimization of heat exchanger network synthesis problems with and without the isothermal mixing assumption. Computers & Chemical Engineering 26(11), 1581-1593.
- Chen, C.-L. and Hung, P.-S. (2004). Simultaneous synthesis of flexible heat-exchange networks with uncertain source-stream temperatures and flow rates. Industrial & Engineering Chemistry Research 43(18), 5916-5928.
- Chen, J.J.J. (1987). Comments on improvements on a replacement for the logarithmic mean. Chemical Engineering Science 42(10), 2488-2489.
- Ciric, A.R. and Floudas, C.A. (1991). Heat exchanger network synthesis without decomposition. Computers & Chemical Engineering 15(6), 385-396.
- Floudas, C.A. (1995). Nonlinear and Mixed-Integer Optimization. New York: Oxford University Press.
- Floudas, C.A. and Ciric, A.R. (1989). Strategies for overcoming uncertainties in heat exchanger network synthesis. Computers & Chemical Engineering 13(10), 1133-1152.
- Floudas, C.A., Ciric, A.R. and Grossmann, I.E. (1986). Automatic synthesis of optimum heat exchanger network configurations. AIChE Journal 32(2), 276-290.
- Floudas, C.A. and Grossmann, I.E. (1986). Synthesis of flexible heat exchanger networks for multiperiod operation. Computers & Chemical Engineering 10(2), 153-168.
- Floudas, C.A. and Grossmann, I.E. (1987). Automatic generation of multiperiod heat exchanger network configurations. Computers & Chemical Engineering 11(2), 123-142.
- Hohmann, E.C. (1971). Optimum networks for heat exchange, University of Southern California. Ph.D.
- Iyer, R. and Grossmann, I. (1996). Global Optimization of Heat Exchanger Networks with Fixed Configuration for Multiperiod Design. Global Optimization in Engineering Design. Grossmann, I., Springer US. 9: 289-308.

Linnhoff, B. and Flower, J.R. (1978). Synthesis of heat exchanger networks: I. Systematic generation of energy optimal networks. *AIChE Journal* 24(4), 633-642.

Ma, X., Yao, P., Luo, X. and Roetzel, W. (2008). Synthesis of multi-stream heat exchanger network for multi-period operation with genetic/simulated annealing algorithms. *Applied Thermal Engineering* 28(8-9), 809-823.

Papoulias, S.A. and Grossmann, I.E. (1983). A structural optimization approach in process synthesis—II: Heat recovery networks. *Computers & Chemical Engineering* 7(6), 707-721.

Paterson, W.R. (1984). A replacement for the logarithmic mean. *Chemical Engineering Science* 39(11), 1635-1636.

Petroleum, I.o. (2000). *Modern Petroleum Technology*: Wiley.

Quesada, I. and Grossmann, I.E. (1993). Global optimization algorithm for heat exchanger networks. *Industrial & Engineering Chemistry Research* 32(3), 487-499.

Saboo, A.K., Morari, M. and Woodcock, D.C. (1985). Design of resilient processing plants—VIII. A resilience index for heat exchanger networks. *Chemical Engineering Science* 40(8), 1553-1565.

Shenoy, U.V. (1995). *Heat Exchanger Network Synthesis: Optimization Process by Energy and Resource Analysis*. Texas: Gulf Publishing Company.

Swaney, R.E. and Grossmann, I.E. (1985). An index for operational flexibility in chemical process design. Part I: Formulation and theory. *AIChE Journal* 31(4), 621-630.

Verheyen, W. and Zhang, N. (2006). Design of flexible heat exchanger network for multi-period operation. *Chemical Engineering Science* 61(23), 7730-7753.

Yee, T.F. and Grossmann, I.E. (1990). Simultaneous optimization models for heat integration—II. Heat exchanger network synthesis. *Computers & Chemical Engineering* 14(10), 1165-1184.

Yee, T.F., Grossmann, I.E. and Kravanja, Z. (1990). Simultaneous optimization models for heat integration—I. Area and energy targeting and modeling of multi-stream exchangers. *Computers & Chemical Engineering* 14(10), 1151-1164.

Zhu, X.X. (1995). Automated synthesis of HENs using block decomposition and heuristic rules. *Computers & Chemical Engineering* 19, Supplement 1(0), 155-160.

CHAPTER III

EXPERIMENTAL

3.1 Equipment

3.1.1 Computer

- A. Lenovo Y450 Intel® Core 2 Duo P7450 (2.13 GHz)
- B. Acer Intel® Core 2 Quad Processor Q6600 (2.4 GHz)

3.1.2 Software

- A. Mathematical programming software: Generic Algebraic Modeling System (GAMS).
- B. Commercial process engineering simulation software (PRO/II).
- C. Microsoft Visio.
- D. Microsoft Excel Spreadsheet.

3.2 Methodology

3.2.1 Formulation of Stage-wise Superstructure Model for Single Period

3.2.1.1 Model

Firstly, a single period model is formulated based on stage-wise superstructure of Yee and Grossmann (1990) using GAMS as the optimization program. This single period model will be the starting point of following modified models.

3.2.1.2 Case Study

A simple case study of three hot streams and four cold streams is used to verify the model. The data is adapted from a literature of Verheyen and Zhang (2006). As their case study is a multiperiod problem of vacuum gas oil (VGO) hydrotreating unit in oil refinery. Therefore, the selected stream data is chosen from one of those three periods.

3.2.2 Sequential Approach for Multiperiod HEN Synthesis

3.2.2.1 *Algorithm*

Several algorithms of multiperiod heat exchanger network synthesis will be proposed. The methodology is based on the utilization of only MINLP single period model from section 3.2.1. Some modifications on the model may be needed in accordance with each algorithm.

3.2.2.2 *Case Study*

A case study is applied with the proposed algorithms. As mentioned before, the problem is adapted from the literature of Verheyen and Zhang (2006). It composes of three operational periods of VGO hydrotreating unit. Those different conditions of each period result from deactivation of catalyst used in the process. After applying each algorithm, the multiperiod HENs will be compared by considering total annualized cost (TAC).

3.2.3 Simultaneous Approach for Multiperiod HEN Synthesis

3.2.3.1 *Model*

For simultaneous approach, the MINLP single period model will be modified to obtain an MINLP multiperiod model where it can solve the problem as all-at-once step by taking into account all stream data of every period concurrently. Therefore, there is no need to assemble the solution of each period.

3.2.3.2 *Case Study*

The case study is similar to the one that is used in sequential approach. The best final solutions of sequential and simultaneous approaches will be compared.

3.2.4 Application to the Industrial Case: CDU Process

There are some differences between case study and real case. Basically, the real case problem is larger and more complex which means that there are more streams involved in HEN. This makes the model non-convex and difficult to solve. For example, it may require much more computational time to solve or the solution may fall to a local optimum because the search space is very large.

For real industrial case study, firstly, simulation of the real process has to be done by using PRO/II, and then essential data will be extracted from the program to be applied with the model. The procedures are shown as follows:

- A. Simulating the refinery process using PRO/II as a real case study.
- B. Extracting the required data from PRO/II which are needed for GAMS
- C. Applying the MINLP multiperiod model with the real case data.
- D. Validating the results from GAMS on PRO/II in order to test its feasibility.

The validation step is important because some assumptions are used in mathematical model for simplification such as constant heat capacity flowrates. But, in fact, heat capacity is a function of temperature. Therefore, as the temperature changes, heat capacity changes. Because of this issue, validation of final HEN has to be done to see its feasibility in real simulation.

3.2.5 Model Improvement

During developing models, some ideas of model improvement may arise to reduce the computational time. This is because solving by GAMS is very sensitive by nature of solver especially MINLP which cannot guarantee the global optimal solution.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Formulation of MINLP Single Period Model

4.1.1 Model Components

A single period model was developed based on stage-wise superstructure proposed by Yee and Grossmann (1990). GAMS has been used to implement the model. Because the model is an MINLP involving linear binary variables, linear and non-linear continuous variables, so DICOPT was selected as a solver to solve the problem. This model subjected to minimize total annualized cost (TAC) where the trade-off of capital cost, area cost, and utility cost has been done simultaneously.

The components of a model in GAMS are shown as follows:

4.1.1.1 Sets

Sets are indices in algebraic representations that are used to identify the domain of a specific group of data. For example, i indicates hot stream i and j indicates cold stream j .

4.1.1.2 Data

Data is the input design parameters of the problem such as inlet/outlet temperatures, heat capacity flowrates, heat transfer coefficient, and economic parameter. There are several ways for data entry in GAMS such as declaring Parameter, Tables, or Scalars.

4.1.1.3 Variables

Variables are the decision variables which are optimized and determined by GAMS such as inter-stage temperatures, heat load, areas, and topology. It consists of binary variables and continuous variables.

4.1.1.4 Equations

Equations are the statements that express relationship between data and variables. The model has both equality constraint and non-equality constraint.

4.1.1.5 *Model and Solve Statement*

Model is the collection of equations declared by a chosen name and the solve statement will call for solver in order to solve the problem corresponding to objective of optimization (maximizing or minimizing).

4.1.1.6 *Other Components*

These GAMS's components are optional which are Display statements and Assignment of bound and/or initial values.

4.1.2 Assumptions

The MINLP type of model has more complexity and non-convex than the other types because of non-continuous and nonlinear functions in model. Hereby, some assumptions have to be made for simplicity.

4.1.2.1 *Isothermal-mixing*

In any stage, if the streams are splitted and pass through more than one exchanger, when they return to mix again before going to the next stage, they must have the same temperatures.

4.1.2.2 *Constant Heat Capacity Flowrates*

Heat capacity flowrate is a fluid property which is a function of temperature. Thus, in reality, the heat capacity flowrates in HEN would have been unconstant, but the constant heat capacity flowrates are assumed to reduce complexity of the model because the temperatures of the streams are changed and optimized throughout the model solving.

4.1.2.3 *No Stream Bypass*

Every splitted stream is needed to pass a heat exchanger. The important reason is that it cannot be splitted and mixed without passing any heat exchanger because of the assumption of isothermal-mixing.

4.1.2.4 *No Split Stream Passing Through more than One Heat Exchanger*

The splitted streams are allowed to pass through only one exchanger. A series of heat exchangers for a splitted stream is not included.

4.1.2.5 Utilities are Places at the End of Each Stream

The utility exchangers are placed at the end of each stream only. The inter-stage utility is not included.

4.1.3 Formulations

The MINLP single period model consists of a series of equations as expressed below from Eq. 4.1-4.26. For more detailed description of each equation, see section 2.2.3.2.

4.1.3.1 Overall Energy Balance of Each Stream

$$[Th_{in}(i) - Th_{out}(i)].FCph(i) = \sum_{k \in ST} \sum_{j \in CP} q(i, j, k) + q_{cu}(i) \quad i \in HP \quad (4.1)$$

$$[Tc_{in}(j) - Tc_{out}(j)].FCpc(j) = \sum_{k \in ST} \sum_{i \in HP} q(i, j, k) + q_{hu}(i) \quad j \in CP \quad (4.2)$$

4.1.3.2 Energy Balance at Each Stage

$$[th(i, k) - th(i, k + 1)].FCph(i) = \sum_{j \in CP} q(i, j, k) \quad k \in ST, i \in HP \quad (4.3)$$

$$[tc(j, k) - tc(j, k + 1)].FCpc(j) = \sum_{i \in HP} q(i, j, k) \quad k \in ST, j \in CP \quad (4.4)$$

4.1.3.3 Assignment of Inlet Temperatures

$$Th_{in}(i) = th(i, 1) \quad i \in HP \quad (4.5)$$

$$Tc_{in}(j) = tc(i, NOK + 1) \quad j \in CP \quad (4.6)$$

4.1.3.4 Monotonic Decrease in Temperatures

$$th(i, k) \geq th(i, k + 1) \quad k \in ST, i \in HP \quad (4.7)$$

$$tc(j, k) \geq tc(j, k + 1) \quad k \in ST, j \in CP \quad (4.8)$$

$$th(i, NOK + 1) \geq Th_{out}(i) \quad i \in HP \quad (4.9)$$

$$TC_{out}(j) \geq tc(j, 1) \quad j \in CP \quad (4.10)$$

4.1.3.5 Hot and Cold Utility Loads

$$[th(i, NOK + 1) - Th_{out}(i)].FCph(i) = qcu(i) \quad i \in HP \quad (4.11)$$

$$[TC_{out}(j) - tc(j, 1)].FCpc(j) = qhu(j) \quad j \in CP \quad (4.12)$$

4.1.3.6 Logical Constraints

$$q(i, j, k) - Q_{up} \cdot z(i, j, k) \leq 0 \quad i \in HP, j \in CP, k \in ST \quad (4.13)$$

$$qcu(i) - Q_{up} \cdot zcu(i) \leq 0 \quad i \in HP \quad (4.14)$$

$$qhu(j) - Q_{up} \cdot zhu(j) \leq 0 \quad j \in CP \quad (4.15)$$

where $z(i, j, k), zcu(i), zhu(i) \in \{0, 1\}$

4.1.3.7 Minimum Approach Temperature Constraints

$$dt(i, j, k) \leq th(i, k) - tc(j, k) + DT_{up} \cdot (1 - z(i, j, k))$$

$$i \in HP, j \in CP, k \in ST \quad (4.16)$$

$$dt(i, j, k + 1) \leq th(i, k + 1) - tc(j, k + 1) + DT_{up} \cdot (1 - z(i, j, k))$$

$$i \in HP, j \in CP, k \in ST \quad (4.17)$$

$$dtku(i) \leq th(i, NOK + 1) - Tcu_{out} + DT_{up} \cdot (1 - zcu(i)) \quad i \in HP \quad (4.18)$$

$$dthu(j, p) \leq Thu_{out} - tc(j, 1, p) + DT_{up} \cdot (1 - zhu(i)) \quad j \in CP \quad (4.19)$$

4.1.3.8 Log Mean Temperature Difference

$$LMTD(i, j, k) = \left[dt(i, j, k) \cdot dt(i, j, k + 1) \cdot \frac{dt(i, j, k) + dt(i, j, k + 1)}{2} \right]^{1/3}$$

$$i \in HP, j \in CP, k \in ST \quad (4.20)$$

$$LMTDHU(j) = \left[dthur(j) \cdot dthul(j) \cdot \frac{dthur(j) + dthul(j)}{2} \right]^{1/3} \quad j \in CP \quad (4.21)$$

$$LMTDCU(i) = \left[dtcur(i) \cdot dtcul(i) \cdot \frac{dtcur(i) + dtcul(i)}{2} \right]^{1/3} \quad i \in HP \quad (4.22)$$

4.1.3.9 Heat Exchanger Area Calculations

$$Area(i, j, k) \geq \frac{q(i, j, k)}{LMTD(i, j, k) \cdot U(i, j)} \quad i \in HP, j \in CP, k \in ST \quad (4.23)$$

$$AreaHU(j) \geq \frac{qhu(j)}{LMTDHU(j) \cdot UHU(j)} \quad j \in CP \quad (4.24)$$

$$AreaCU(i) \geq \frac{qcu(i)}{LMTDCU(i) \cdot UCU(i)} \quad i \in HP \quad (4.25)$$

4.1.3.10 Objective Function

$$\begin{aligned} \min TAC = AF & \left[\sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C_f \cdot z(i, j, k) + \sum_{i \in HP} \sum_{CU} C_f \cdot zcu(i) \right. \\ & \left. + \sum_{j \in CP} \sum_{HU} C_f \cdot zhu(j) \right] + AF \cdot \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C \cdot Area(i, j, k)^B \\ & + AF \cdot \sum_{j \in CP} C \cdot AreaHU(j)^B + AF \cdot \sum_{i \in HP} C \cdot AreaCU(i)^B \\ & + \sum_{i \in HP} C_{cu} qcu(i) + \sum_{j \in CP} C_{hu} qhu(j) \end{aligned} \quad (4.26)$$

The set of all formulations shown above would be used as original model for extension to other models in both sequential and simultaneous approaches. It was approved by a simple simulated case study which illustrated that the model could run properly.

4.2 Sequential Approach for Multiperiod HEN Synthesis

4.2.1 Algorithm

Multiperiod HEN synthesis usually results from the assembly of HEN from each period. The concept of this proposed sequential method is that HEN will be generated for each period separately by using either original MINLP single period model or modified model from section 4.1. Figure 4.1 shows the proposed sequential approach algorithm.

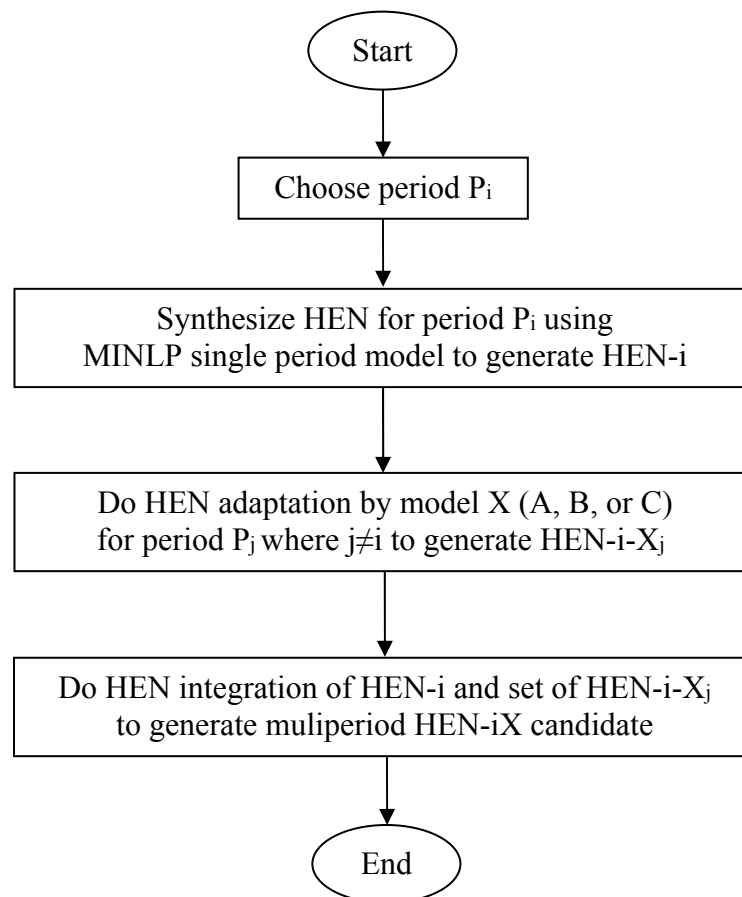


Figure 4.1 Algorithm of sequential approach for multiperiod HEN synthesis.

4.2.1.1 Single Period HEN Synthesis

In this step, the MINLP single period model from section 4.1 is used to generate an initial HEN for period P_i . Note that, for multiperiod problem, there are stream properties for more than one period, but only one of those periods will be chosen to be the input data for this step. For example, as illustrate in Figure 4.1, period P_i is used as the starting point. For simplification, this generated HEN is so-called HEN-i which means HEN for period P_i .

4.2.1.2 HEN Adaptation

This procedure is to adapt and apply the existing initial HEN from the previous step by maintaining the structure as much as possible. As a result, the adapted HEN will be able to operate for the particular periods. Hence, the input stream data for this step is the data of remaining periods (P_j) besides period P_i . There are three different modified models can be used in HEN adaptation which are model A, B, and C. As mentioned before, each model was modified from original MINLP single period model from section 4.1. They have different algorithms, but mainly retain the structure of the initial HEN. The principles of those three modified models are described as follows:

A. Model A

Model A is the MINLP single period model with fixed process exchanger area. The principle of this model is to keep the same topology and area of process-process exchangers only. The location of existing utility exchanger is also similar to initial HEN. But while solving, model A will allow new utility exchanger to be added and some of utility exchangers may need to switch between hot and cold utility when changing the operational periods. The objective function of this model is to minimize total additional area of utility exchangers as shown in Eq. 4.27.

$$\begin{aligned} \text{Total additional area} = & \sum_{j \in CP} \max[0, \text{AreaHU}_{\text{new}}(j) - \text{AreaHU}_{\text{old}}(j)] \\ & + \sum_{i \in HP} \max[0, \text{AreaCU}_{\text{new}}(i) - \text{AreaCU}_{\text{old}}(i)] \end{aligned}$$

(4.27)

B. Model B

Model B is the NLP single period model with fixed exchanger topology. The concept of this model is to maintain the old structure of initial HEN and not to allow the addition or removal of any heat exchangers. However, some heat exchangers may have to change their areas to satisfy heat balance. The objective function of this model is least square of area change as shown in Eq. 4.28.

$$\begin{aligned}
 \text{Total area change} = & \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} [Area_new(i, j, k) - Area_old(i, j, k)]^2 \\
 & + \sum_{j \in CP} [AreaHU_new(j) - AreaHU_old(j)]^2 \\
 & + \sum_{i \in HP} [AreaCU_new(i) - AreaCU_old(i)]^2
 \end{aligned}
 \tag{4.28}$$

C. Model C

Model C is the NLP single period model with fixed exchanger topology. This model is indeed similar to model B except the objective function. The objective function of this model is to minimize additional area and does not take into account the change of reduced area. The equation is illustrated in Eq. 4.29.

Total additional area

$$= \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} \max[0, Area_new(j, j, k) - Area_old(i, j, k)]$$

$$\begin{aligned}
& + \sum_{j \in CP} \max[0, AreaHU_new(j) - AreaHU_old(j)] \\
& + \sum_{i \in HP} \max[0, AreaCU_new(i) - AreaCU_old(i)]
\end{aligned}
\tag{4.29}$$

Note that in these three modified models, the other variables such as temperatures and heat loads are not fixed. Thus, they may have to change due to the variation of each period condition. And because these variables do not directly affect the cost of HEN, they are not included in the objective function.

From Figure 4.1, after HEN-i is adapted in the second step, it is named as HEN-i-X_j, where X refers to the modified model which can be either A, B, or C and j means that HEN is designed for period P_j.

4.2.1.3 HEN Integration

From HEN synthesis and HEN adaptation steps, HENs for each particular period are generated. Definitely the plant would not like to build all of those HENs for each period. Therefore, they have to be integrated to obtain a HEN which can operate for every condition. This is so-called multiperiod HEN-iX where i indicates the chosen period P_i in HEN synthesis step and X represents the modified model being used in HEN adaptation step. The integration can be done by using these two concepts:

A. Every match that exists at least one time in any period has to exist in multiperiod HEN. In other words, every heat exchanger that is needed in each period will be included in multiperiod HEN even though it is used for only one or less than N period.

B. Considering the area of each exchanger in multiperiod HEN, the maximum required area among all period will be selected. Because if the maximum area is not used, the area may not enough for heat to transfer in the period which requires more heat load.

When multiperiod HEN is obtained from HEN integration step, the actual TAC has to recalculate by using Eq. 4.30:

$$\begin{aligned}
TAC = AF & \left[\sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C_f \cdot z(i, j, k) + \sum_{i \in HP} \sum_{CU} C_f \cdot zcu(i) + \sum_{j \in CP} \sum_{HU} C_f \cdot zhu(j) \right] \\
& + AF \cdot \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C \cdot Area_max(i, j, k)^B + AF \cdot \sum_{j \in CP} C \cdot AreaHU_max(j)^B \\
& + AF \cdot \sum_{i \in HP} C \cdot AreaCU_max(i)^B + \sum_{p \in PR} \frac{DOP(p)}{TOP} \sum_{j \in CP} C_{hu} qhu(j, p) \\
& + \sum_{p \in PR} \frac{DOP(p)}{TOP} \sum_{i \in HP} C_{cu} qcu(i, p)
\end{aligned} \tag{4.30}$$

4.2.2 Case Study

The case study was adapted from Verheyen and Zhang (2006). It is the vacuum gas oil (VGO) hydrotreating unit in oil refinery. This unit can simultaneously treat some impurities such as sulfur and convert some fraction of VGO to more valuable products. Hydrotreating is a catalytic reaction. The catalyst used in the process will gradually deactivate until a certain time before regeneration. During the deactivation, rate of reaction will decrease. Therefore, to compensate the deactivation of catalyst, the increase of temperature will help keep the reaction at approximately the same rate. In this case study there are three periods which are start-of-run (SOR), mid-of-run (MOR), and end-of-run (EOR). The change of temperature in each period will cause the variation of outlet compositions and flowrates.

Table 4.1, 4.2, and 4.3 shows stream properties for period SOR, MOR, and EOR respectively. Since the data of heat transfer coefficient was not provided, it will be assumed. Furthermore, the duration of each period is assumed to be equal.

Table 4.1 Stream properties for SOR

Stream	Inlet temperature	Outlet temperature	Heat capacity	Heat transfer
--------	-------------------	--------------------	---------------	---------------

	(°C)	(°C)	flowrate (kW/K)	coefficient (kW/m ² .°C)
H1	393	60	201.6	2.0
H2	160	40	185.1	2.0
H3	354	60	137.4	2.0
C1	72	356	209.4	1.5
C2	62	210	141.6	1.5
C3	220	370	176.4	2.0
C4	253	284	294.4	2.0
HU	400	399	-	2.0
CU	15	20	-	1.0

Table 4.2 Stream properties for MOR

Stream	Inlet temperature (°C)	Outlet temperature (°C)	Heat capacity flowrate (kW/K)	Heat transfer coefficient (kW/m ² .°C)
H1	406	60	205.0	2.0
H2	160	40	198.8	2.0
H3	362	60	136.4	2.0
C1	72	365	210.3	1.5
C2	62	210	141.0	1.5
C3	220	370	175.4	2.0
C4	250	290	318.7	2.0
HU	400	399	-	2.0
CU	15	20	-	1.0

Table 4.3 Stream properties for EOR

Stream	Inlet temperature	Outlet temperature	Heat capacity	Heat transfer
--------	-------------------	--------------------	---------------	---------------

	(°C)	(°C)	flowrate (kW/K)	coefficient (kW/m ² .°C)
H1	420	60	208.5	2.0
H2	160	40	175.2	2.0
H3	360	60	134.1	2.0
C1	72	373	211.1	1.5
C2	62	210	140.5	1.5
C3	220	370	174.5	2.0
C4	249	286	271.2	2.0
HU	400	399	-	2.0
CU	15	20	-	1.0

For the economic parameters in cost evaluation, hot and cold utility cost are 115.2 \$/kW.yr and 1.3 \$/kW.yr, respectively. Annualization factor is 0.2 which corresponds to 10 years of project life time and 15% prevailing rate of interest. The exchanger minimum approach temperature (EMAT) is 5°C. The capital cost of heat exchanger is calculated using Eq. 4.31.

$$\text{Heat exchanger cost} = C_f + C_A(\text{Area})^B \quad (4.31)$$

Where C_f is fixed cost of heat exchanger = 8333.3 \$/exchanger and C_A is area cost = 641.7 \$/m^(2B) and B is area exponent = 1.

4.2.3 Results

4.2.3.1 *Determining the Best Model for HEN Adaptation*

Since there are three modified models can be used to adapt the initial HEN in HEN adaptation step. Firstly, the best model should be investigated. Thus, in HEN synthesis step, one of three periods will be chosen as the controlled variable. The mathematical model was implemented on GAMS 21.4 with DICOPT2x-C (CONOPT3 and CPLEX 9.0) as MINLP solver. The computer platform is Lenovo Y450 with Intel® Core 2 Duo T6400 CPU at 2.0 GHz.

A. Generated Initial HEN

From the case study, the second period (MOR) was designated as a controlled variable to synthesize the initial HEN. Because it represents the average value among every period, it should be suitable to be the controlled data for determining the best model for HEN adaptation step. In Figure 4.2, HEN-2 for MOR was generated by using MINLP single period model.

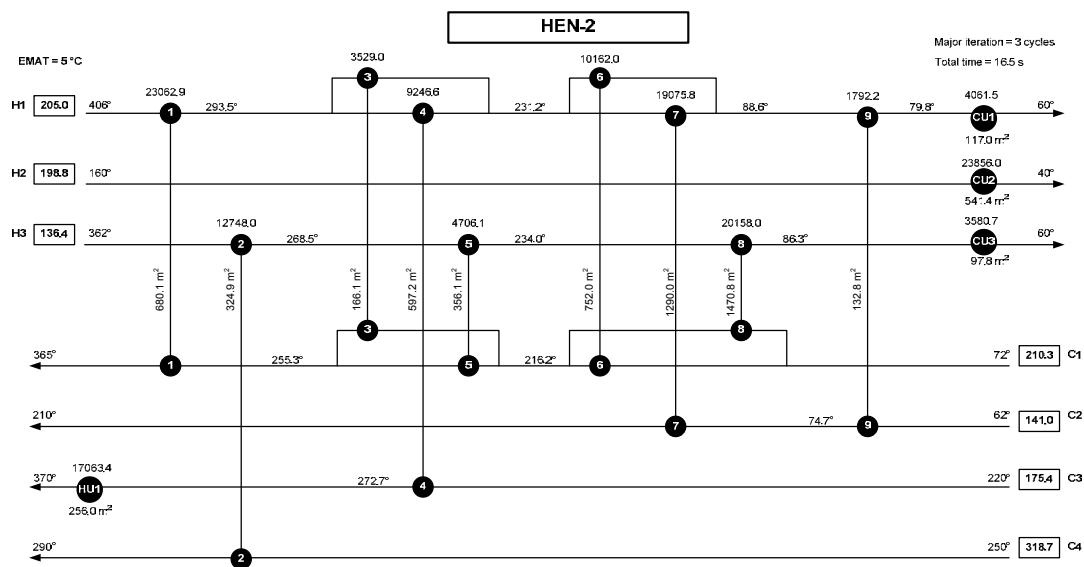


Figure 4.2 Grid diagram of HEN-2.

B. Adapted HENs

Based on initial HEN or HEN-2, it was adapted by entering the some required information of HEN-2 and stream data of period P₁ (SOR) and P₃ (EOR) into the modified models which are model A, B, and C. Some variables such as areas and topology were fixed depending on strategy of each model. Therefore, in this step, there were two more HENs for each modified model (HEN-2-A₁ and HEN-2-A₃ for model A, HEN-2-B₁ and HEN-2-B₃ for model B, HEN-2-C₁ and HEN-2-C₃ for model C). Grid diagrams of adapted HENs are shown in Figure 4.3 to 4.8.

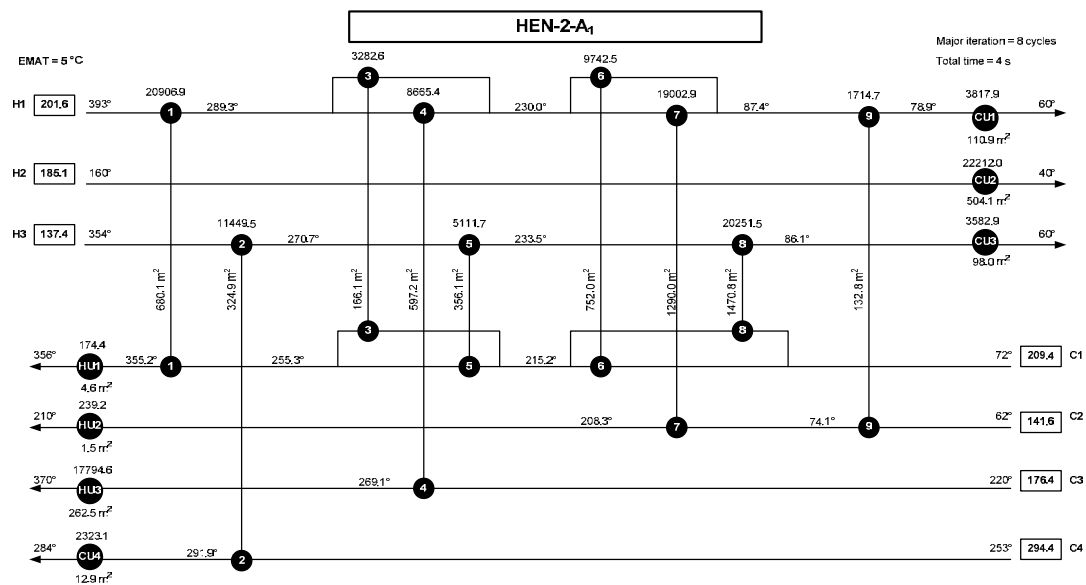


Figure 4.3 Grid diagram of HEN-2-A₁.

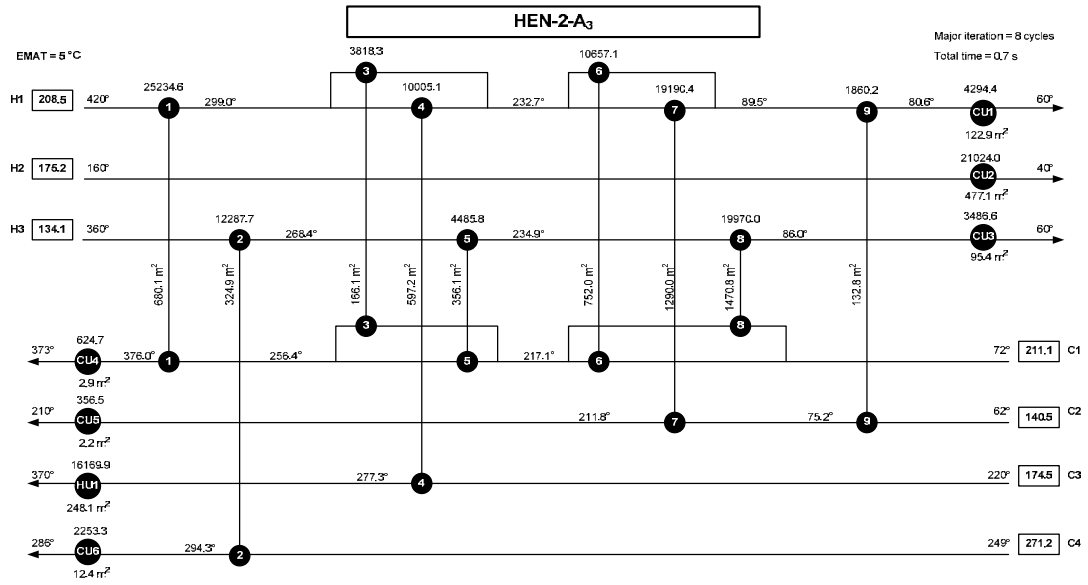


Figure 4.4 Grid diagram of HEN-2-A₃.

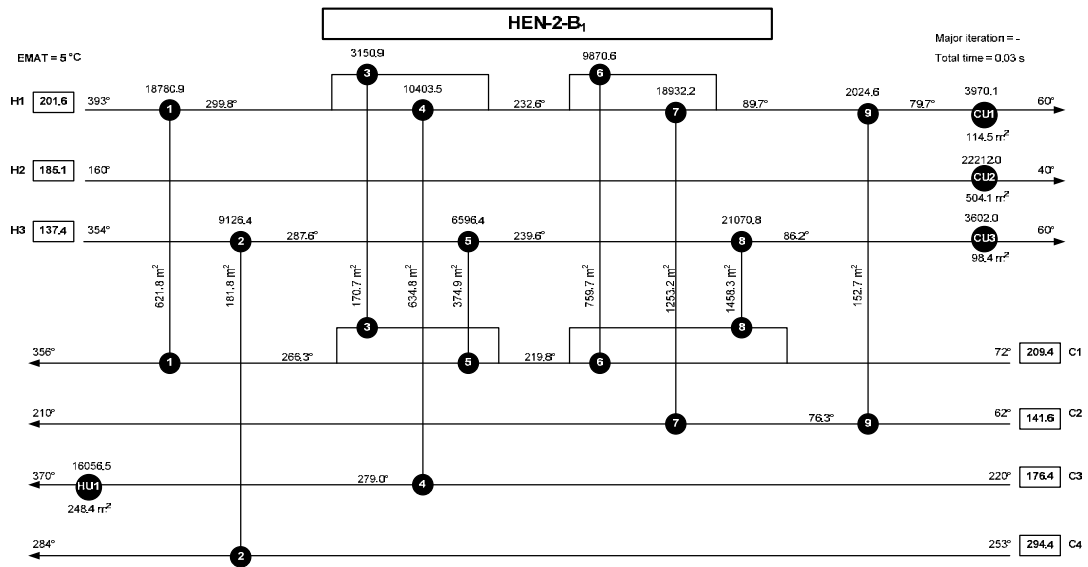


Figure 4.5 Grid diagram of HEN-2-B₁.

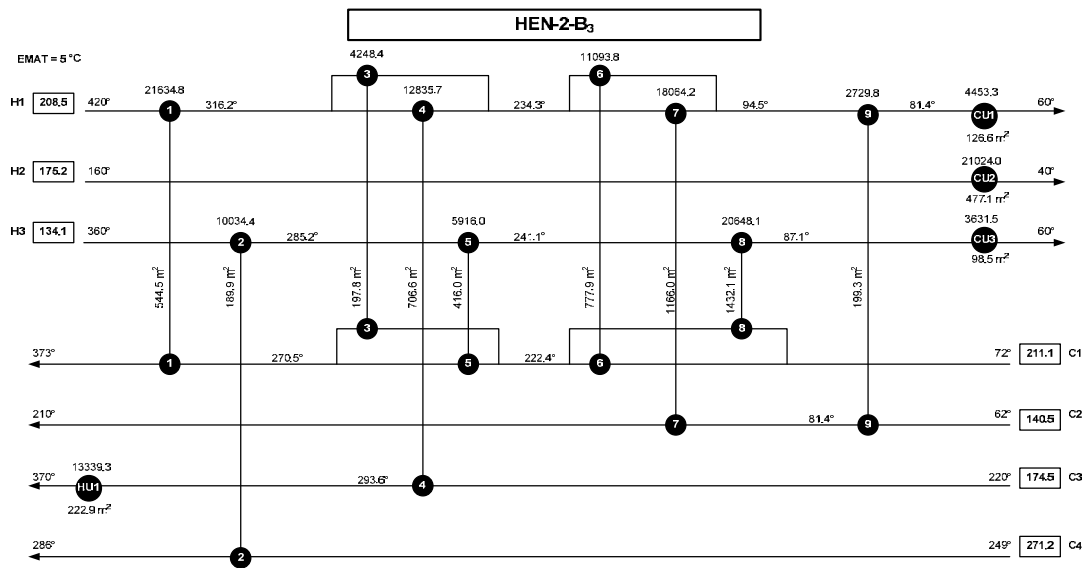


Figure 4.6 Grid diagram of HEN-2-B₃.

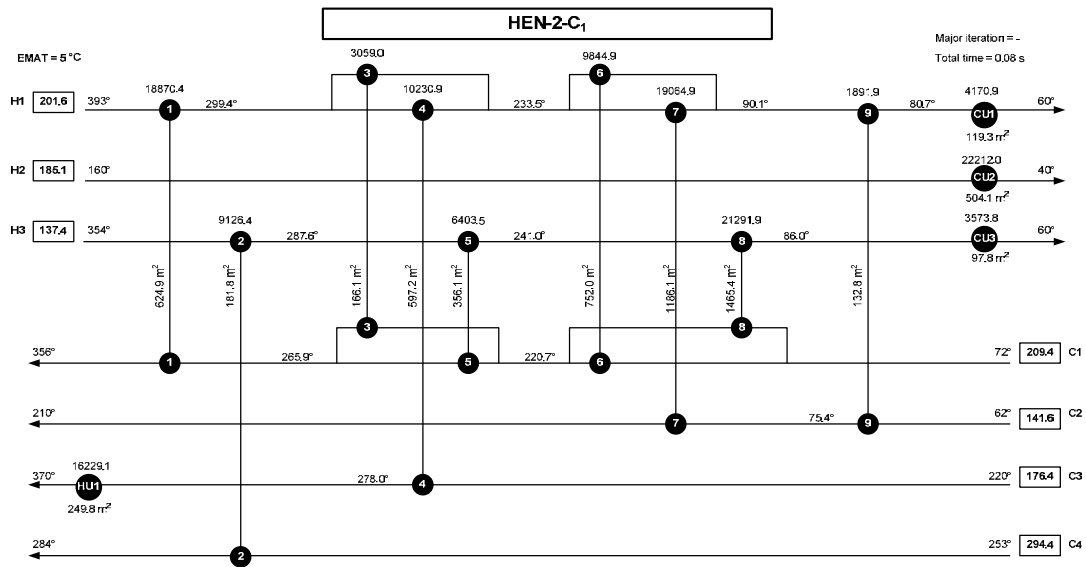


Figure 4.7 Grid diagram of HEN-2-C₁.

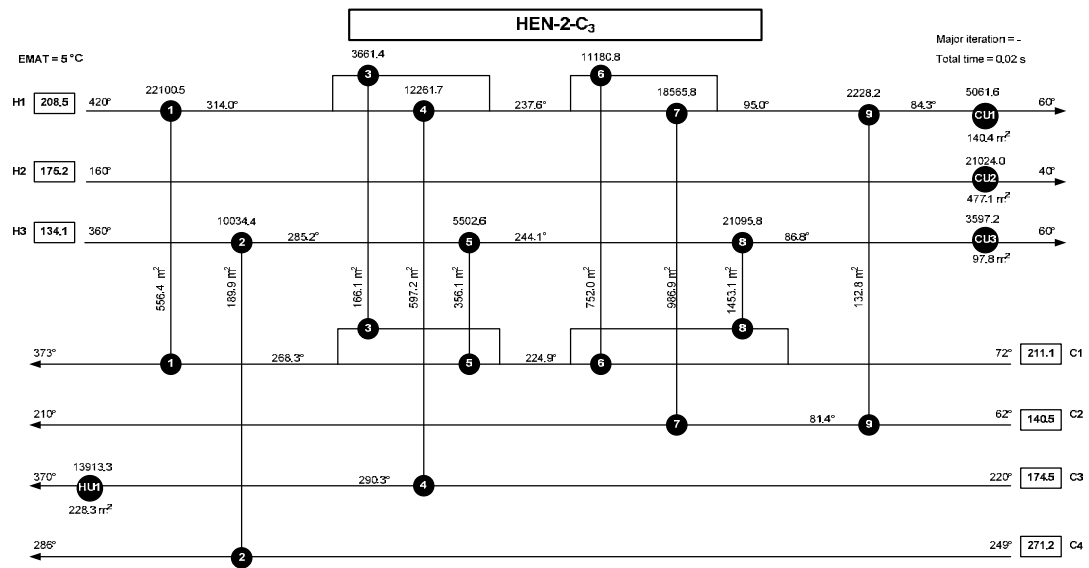


Figure 4.8 Grid diagram of HEN-2-C₃.

C. Integrated HEN

In this step, three HENs for each period of each method will be integrated to get a multiperiod HEN. For example, HEN-2, HEN-2-A₁, and HEN-2-A₃ will be merged by gathering matches that exist in at least one period and maximum area among all periods. The multiperiod HEN-2A, 2B, and 2C are shown in Figure 4.9, 4.10, and 4.11 respectively.

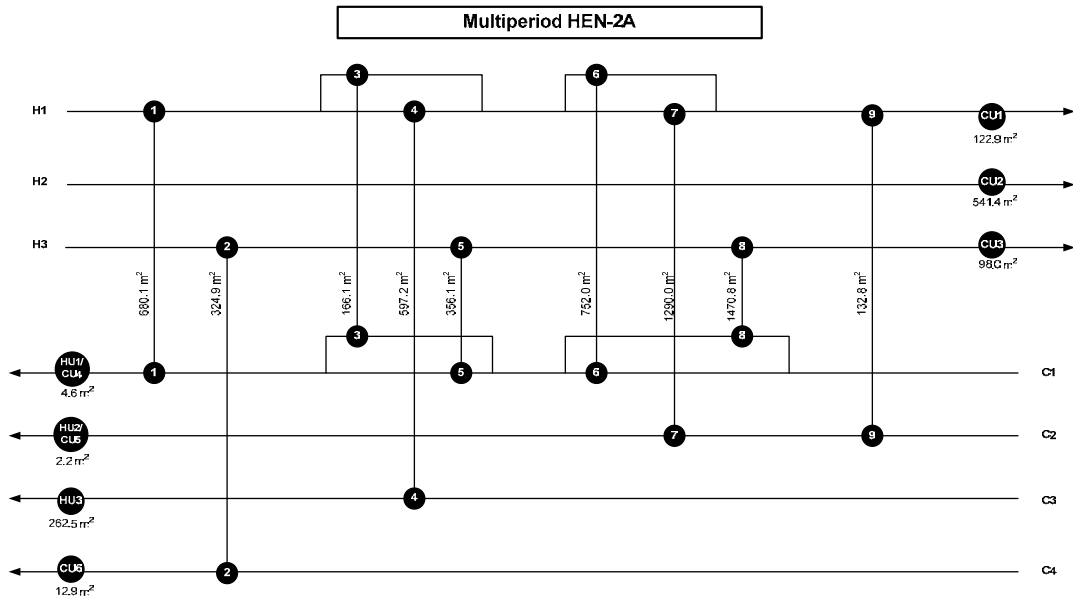


Figure 4.9 Grid diagram of multiperiod HEN-2A.

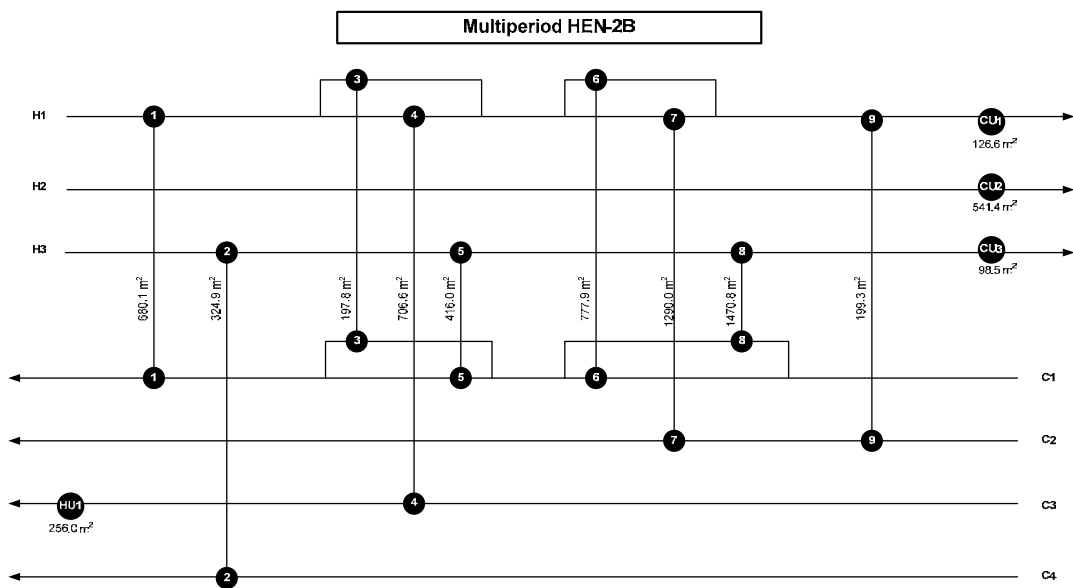


Figure 4.10 Grid diagram of multiperiod HEN-2B.

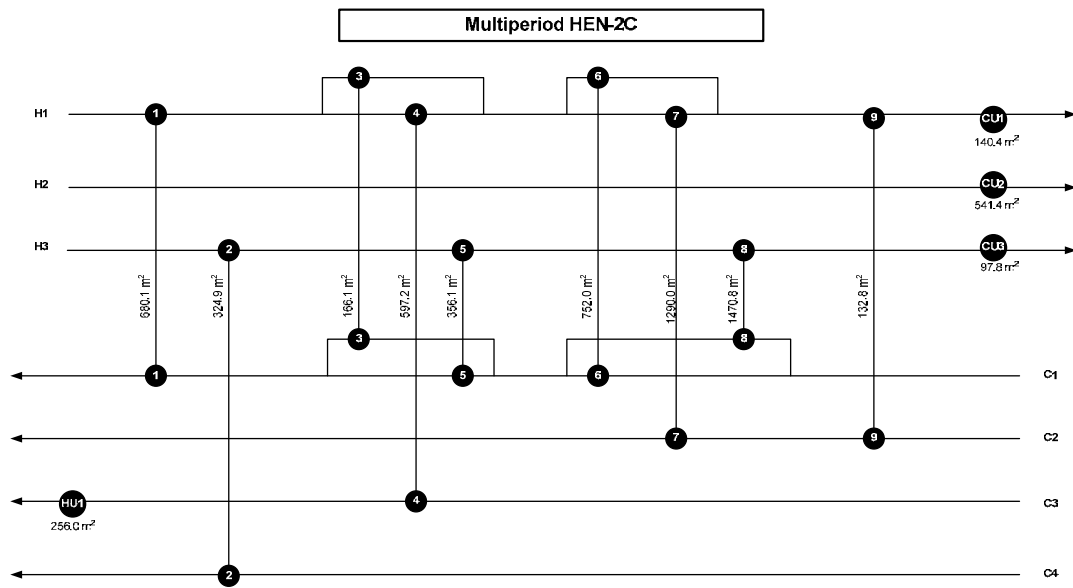


Figure 4.11 Grid diagram of multiperiod HEN-2C.

D. Discussion

Table 4.4 shows the information of multiperiod HEN-2A, 2B, and 2C. It was found that multiperiod HEN-2C, where modified model C was used in HEN adaptation step, gave the lowest TAC at \$2,747,294 per year.

Table 4.4 Results of multiperiod HEN from each method

	Multiperiod HEN-2A	Multiperiod HEN-2B	Multiperiod HEN-2C
No. of heat exchangers	16	13	13
Total area (m ²)	6,814.5	7,085.8	6,805.6
Fixed cost (\$/yr)	26,666	21667	21667
Area cost (\$/yr)	874,573	909,397	873,432
Utility cost (\$/yr)	2,011,953	1,823,201	1,852,195
TAC (\$/yr)	2,913,193	2,754,264	2,747,294

When using model A for HEN adaptation, it was quite a complex model because of some issues. First, sometimes it might need hot utility in some hot streams and similarly cold utility in some cold streams. This issue should not have occurred because the overall requirement of utility will increase. But this was inevitable because many variables, such as topology and areas of process-process exchangers, were fixed. So the remaining relaxed variables were area and topology of utility exchangers including the types of utility. However, if those nonconventional utility exchangers were not allowed to switch between hot and cold utility, the model would not be able to obtain any feasible solutions. Second, some equations of monotonic temperatures had to be neglected due to the allowance of nonconventional utility exchanger. Moreover, the researcher had to find these nonconventional streams manually because the model cannot find them by itself and it was quite difficult and complicated to identify them.

Model B is more relaxed than model A, hence there is no such nonconventional streams that require the switching of hot and cold utility on the same stream. Since the objective function of model B is least square of area change, the model will try to distribute the change of area to every heat exchanger. Even though the area changes are forced to be as small as possible, the total required area of HEN is more than other models and this causes lowest utility cost. Moreover, it was observed that the areas of heat exchangers are all changed from initial HEN.

Although model C is quite similar to model B, it yielded different solution. It was observed that this objective function would force the model to keep the areas of heat exchangers remain the same value as much as possible. The areas of some heat exchangers would be increased only if it was necessary. So the total area required is lowest, while the utility cost is moderate.

To conclude, model C was the best model for HEN adaptation. Not only because it yielded the lowest TAC when compared to other models, but also there were only some of heat exchangers that needed to be changed while operating. This can be done by bypassing streams when the need of area is lower than the maximum area. And there is also no concern about switching between

hot and cold utilities. From these reasons, the sequential approach accompanying model C was the best method to synthesize a robust multiperiod HEN.

4.2.3.2 Finding the Best Solution by Applying Sequential Method with the Best Modified Model

From 4.3.2.1, model C is the most robust modified model for HEN adaptation. In this section, it was applied again, but the chosen period for initial HEN synthesis was varied by every period so that multiperiod HEN-1C, 2C, and 3C were obtained. Finally, one of the multiperiod HENs would be designated as the best multiperiod HEN from sequential approach. The overall procedure is shown in Figure 4.12.

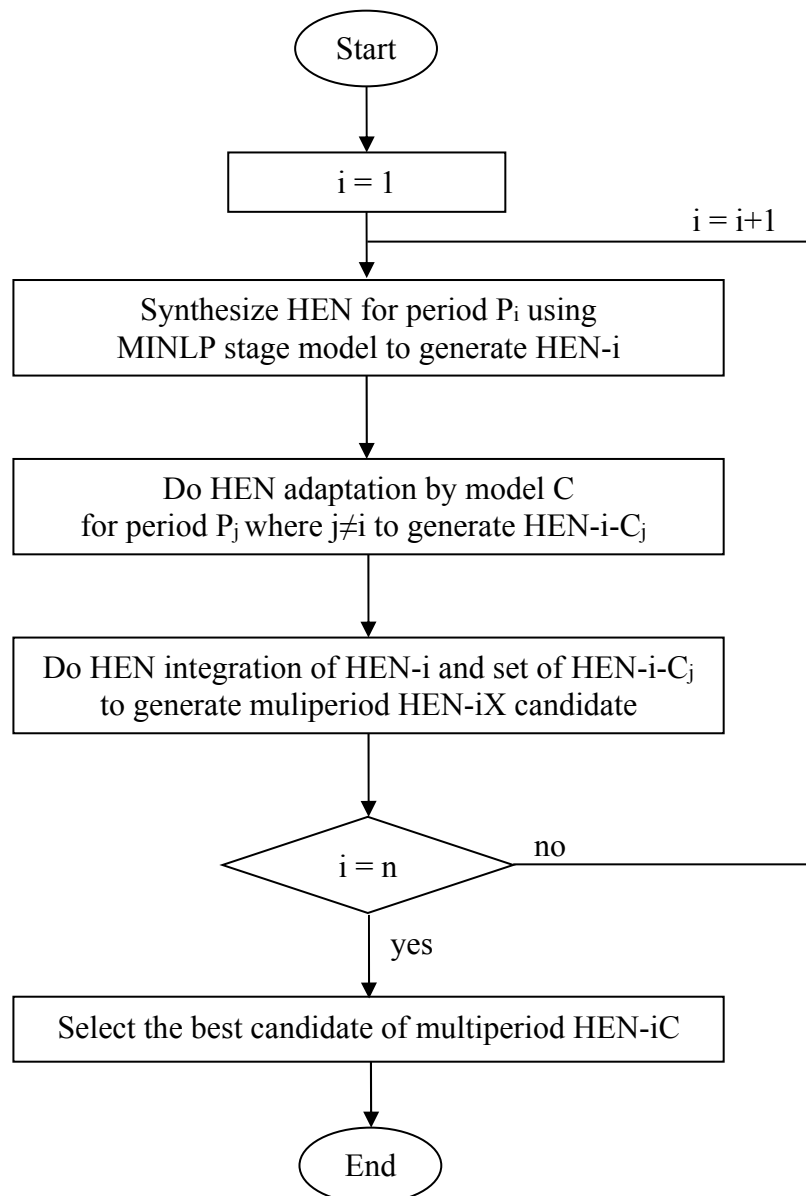


Figure 4.12 Overall procedure of sequential method by using model C.

A. Generated Initial HEN

From case study, all of three periods were separately used as a chosen period for generating initial HEN in the first step. Figure 4.13, 4.14, and 4.15 shows grid diagrams solved by MINLP single period model for period SOR, MOR, and EOR, respectively.

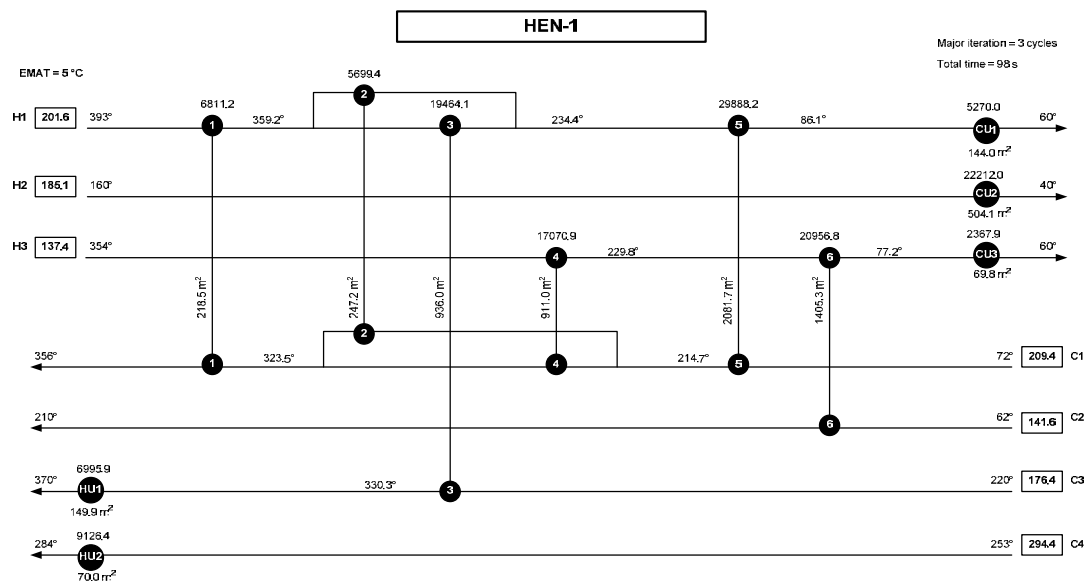


Figure 4.13 Grid diagram of HEN-1.

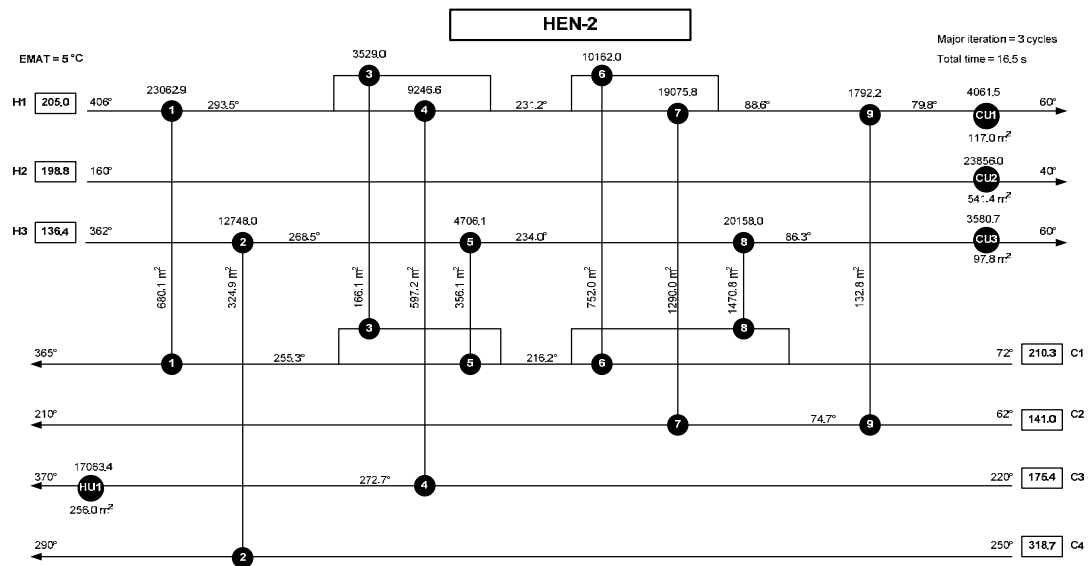


Figure 4.14 Grid diagram of HEN-2.

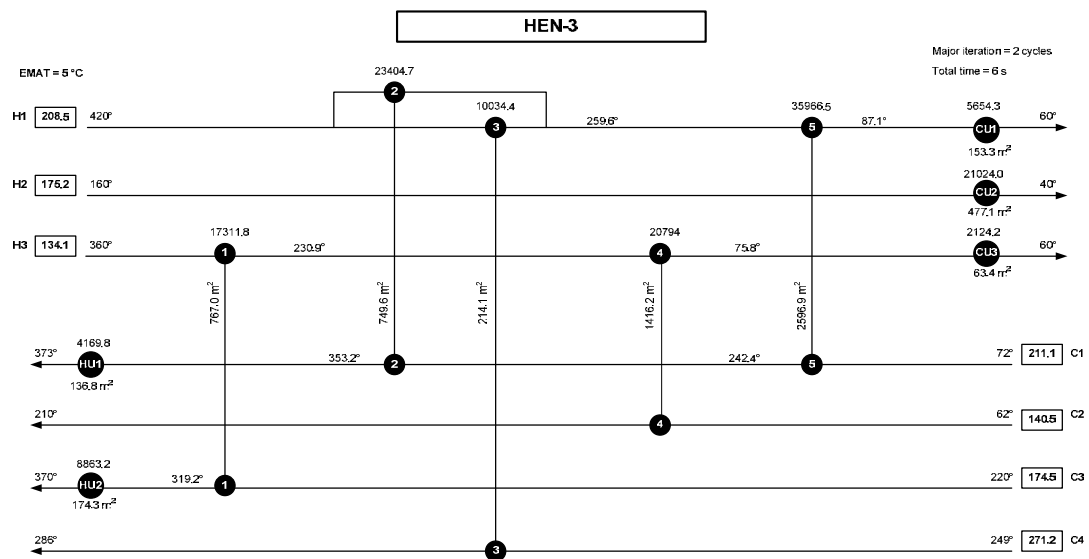


Figure 4.15 Grid diagram of HEN-3.

B. Adapted HENs

From the initial HENs, they were adapted by using modified model C where topology of process exchanger was fixed. The entry of stream properties was from the remaining periods besides the chosen period in HEN synthesis step. For instance, if the chosen period is period 1, the data entry for HEN

adaptation will be period 2 and 3. Therefore, for this case study, there were two more HENs for each chosen period (HEN-1-C₂ and HEN-1-C₃ for period 1, HEN-2-C₁ and HEN-2-C₃ for period 2, HEN-3-C₁ and HEN-3-C₂ for period 3). The grid diagrams of adapted HENs are shown in Figure 4.16 to 4.21.

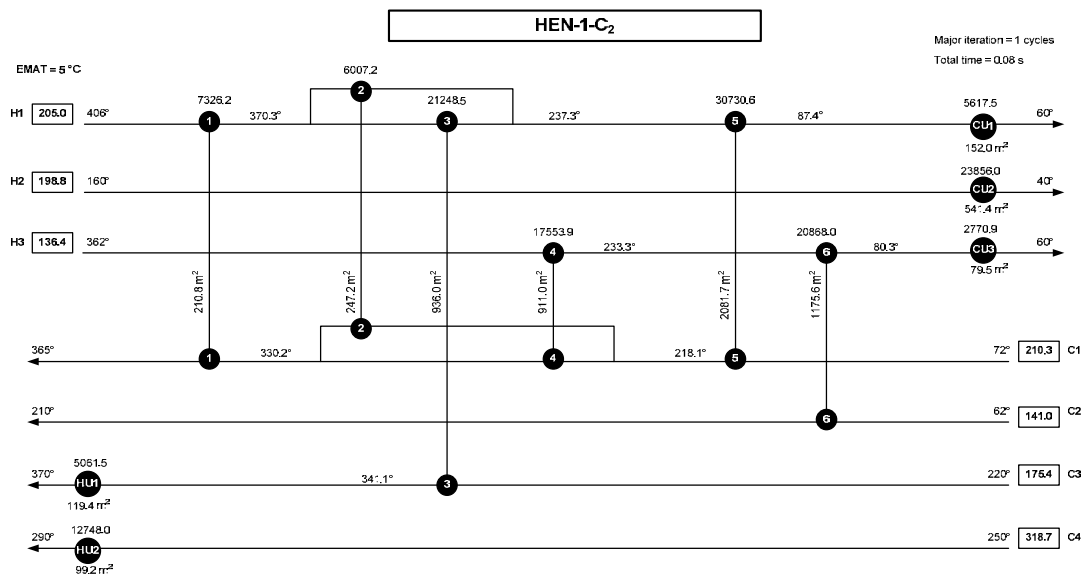


Figure 4.16 Grid diagram of HEN-1-C₂.

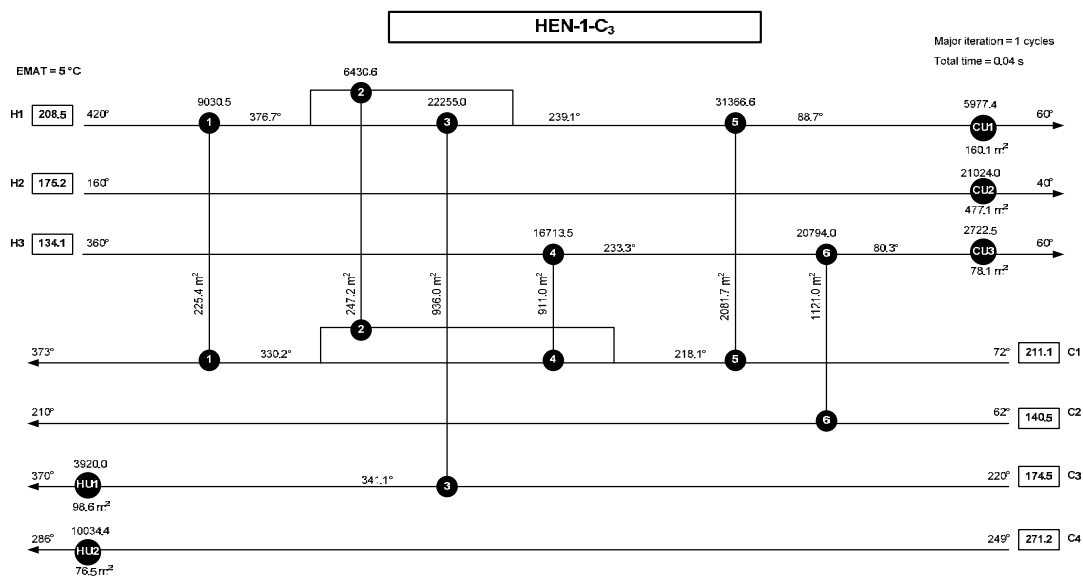


Figure 4.17 Grid diagram of HEN-1-C₃.

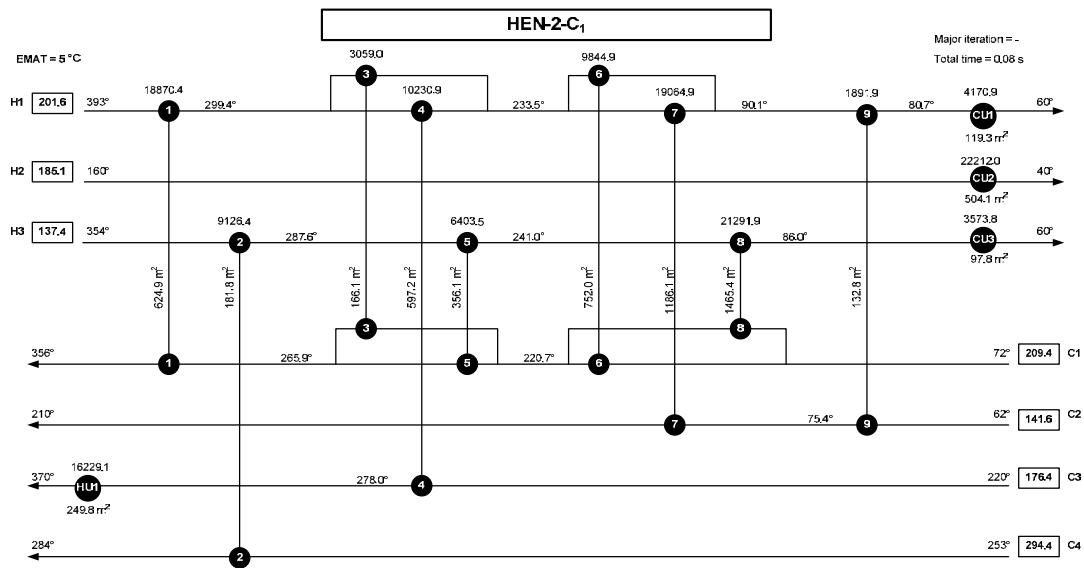


Figure 4.18 Grid diagram of HEN-2-C₁.

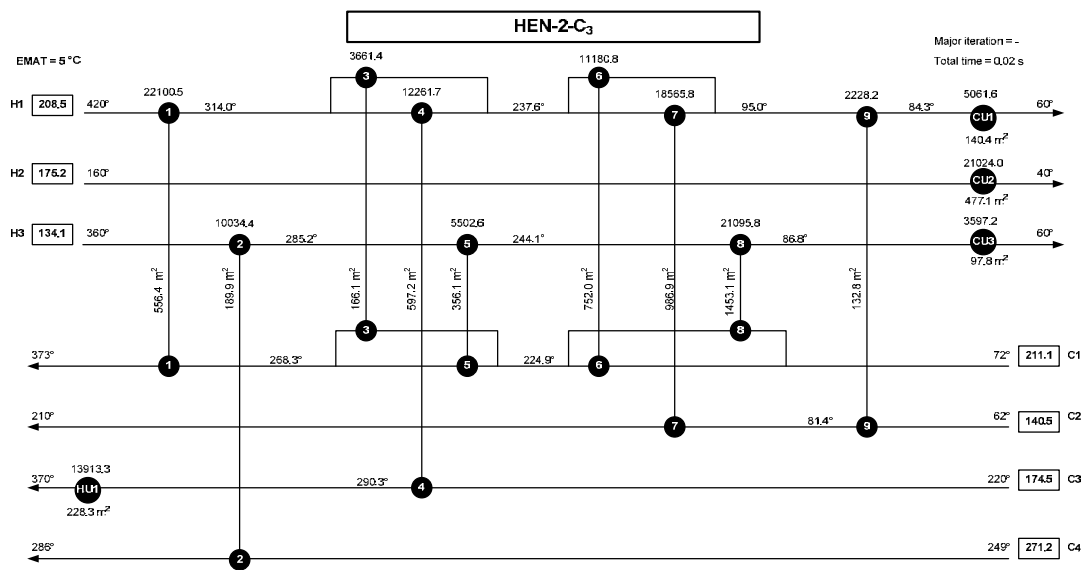


Figure 4.19 Grid diagram of HEN-2-C₃.

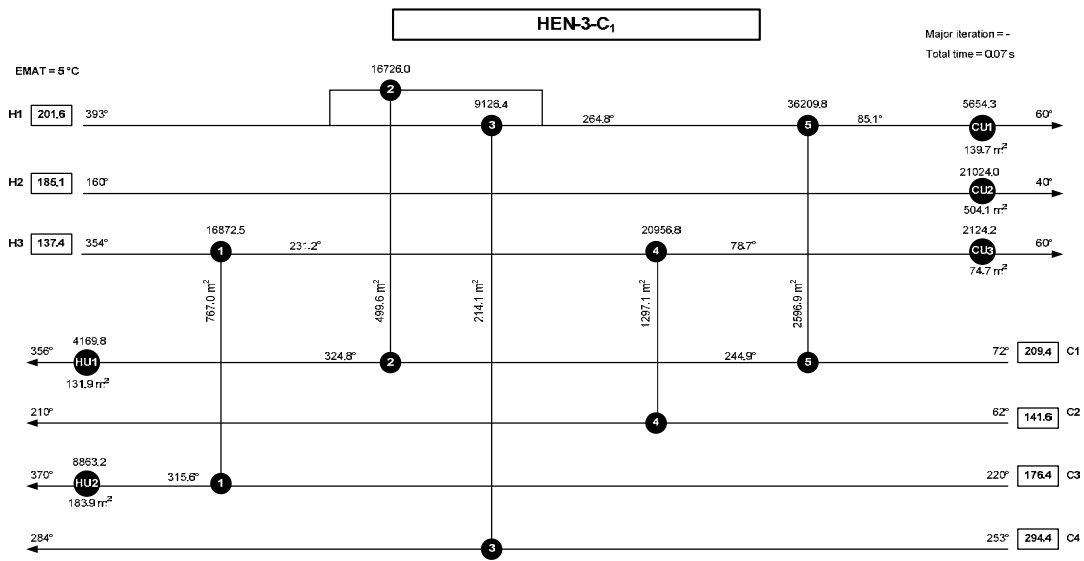


Figure 4.20 Grid diagram of HEN-3-C1.

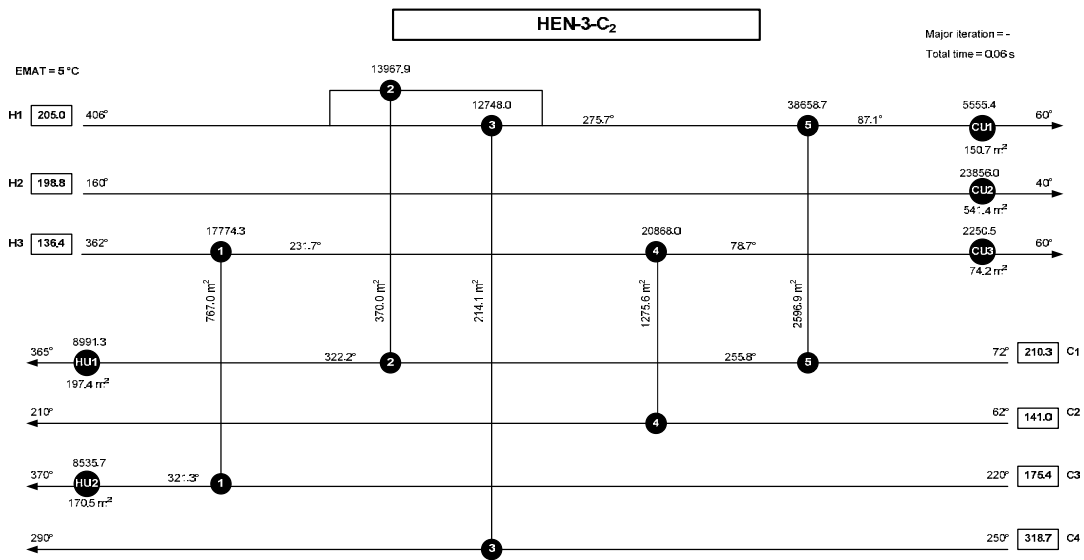


Figure 4.21 Grid diagram of HEN-3-C2.

C. Integrated HEN

Again, three HENs which based on each chosen period (for instance; HEN-1, HEN-1-C₂, and HEN-1-C₃) were assembled to be multiperiod HEN-iC. The multiperiod HEN-1C, 2C, and 3C are shown in Figure 4.22, 4.23, and 4.24 respectively.

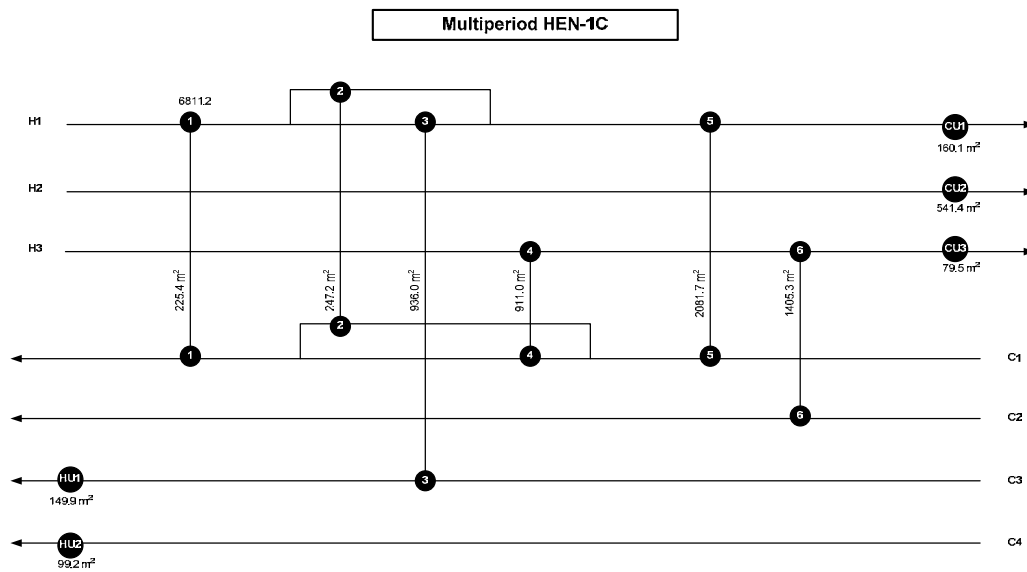


Figure 4.22 Grid diagram of multiperiod HEN-1C.

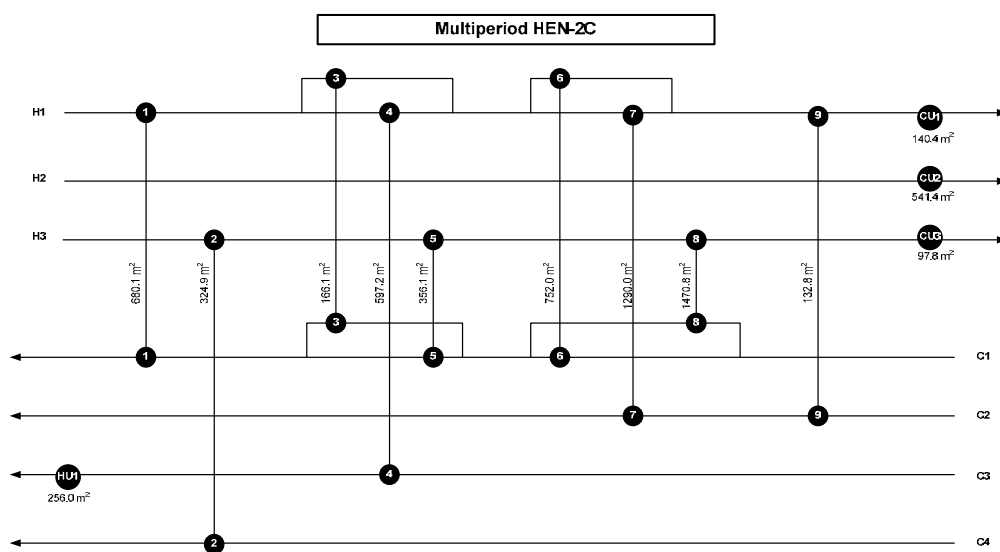


Figure 4.23 Grid diagram of multiperiod HEN-2C.

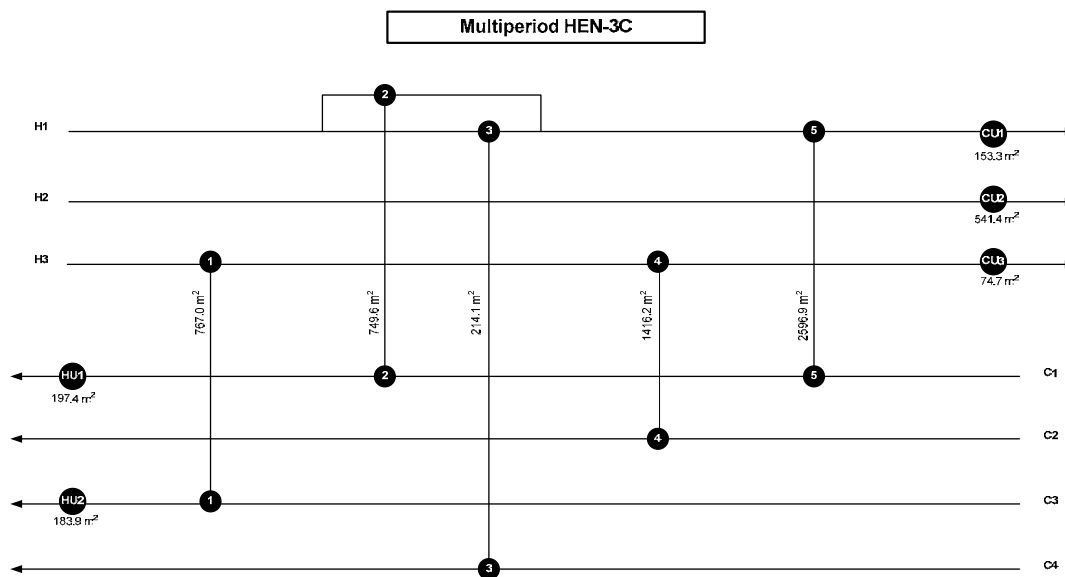


Figure 4.24 Grid diagram of multiperiod HEN-3C.

D. Discussion

Table 4.5 shows the summary information of multiperiod HEN-1C, 2C, and 3C. It was found that multiperiod HEN-3C, where period 3 (EOR) was the chosen period for initial HEN synthesis, yielded the lowest TAC at \$2,733,340 per year.

Table 4.5 Results of multiperiod HEN with different starting period for initial HEN

	Multiperiod HEN-1C	Multiperiod HEN-2C	Multiperiod HEN-3C
No. of heat exchangers	11	13	10
Total area (m ²)	6,836.8	6,805.6	6,894.5
Fixed cost (\$/yr)	18,333	21667	16,666
Area cost (\$/yr)	877,436	873,432	884,840
Utility cost (\$/yr)	1,878,620	1,852,195	1,831,833
TAC (\$/yr)	2,774,389	2,747,294	2,733,340

It is quite complicated to discuss why using period 3 as the chosen period could give the best solution. There are many relevant factors to be considered.

Firstly, because the utility cost is relatively high when compared to the capital cost; therefore, the network which has lowest utility consumption especially hot utility will tend to have lowest TAC as shown in table 4.5.

Secondly, the efficiency of area usage might have affected the best solution. The efficiency of area usage means how much available area can be used while operating. If the area of heat exchanger from initial HEN design is totally used in other periods, in that case it will have maximum efficiency because there is no need to increase area anymore. In other words, the efficiency means how fit the initial HEN is when it is operated in other periods. However, after calculating the efficiency of area usage, the result is the average efficiency for multiperiod HEN-1C, 2C, and 3C are 96.1%, 94.6%, and 94.3% respectively. Although the efficiency of area usage for multiperiod-3C is lowest, it can still give the best solution. It may be because the effect of utility cost dominates the area cost.

Lastly, the duration of each period should have effect on the best solution. It was anticipated that the period which has longest duration will tend to be more suitable chosen period for generating initial HEN because the network that is based on the longest period will be operate at its maximum efficiency for most of the time. But for this case study, it was assumed that the duration of each period is equal, so it was unable to discuss for this issue.

4.3 Developing a Multiperiod Model for Simultaneous Approach

4.3.1 Formulation of MINLP Multiperiod Model

An MINLP multiperiod model was developed based on stage-wise superstructure model by following the modified model of Verheyen and Zhang (2006) including set of bounding equations. Since the MINLP multiperiod model was developed from MINLP single period model in section 4.1, some modifications were made as follows.

- A. Add set P which is the index for period p.
- B. Input stream properties of every period.
- C. Add P as another dimension for most parameters and variables that depend on period.
- D. Add variables Area_max for representing the maximum area among all periods.
- E. Adjust bounding.
- F. Include equation for determining Area_max.
- G. Change the calculation of utility cost so that it depends on duration of each period.

The MINLP multiperiod model for simultaneous approach consists of a series of equations as shown in Eq. 4.32-4.60.

4.3.1.1 Overall Energy Balance of Each Stream

$$[Th_{in}(i, p) - Th_{out}(i, p)]. FCph(i, p) = \sum_{k \in ST} \sum_{j \in CP} q(i, j, k, p) + q_{cu}(i, p)$$

$$i \in HP, p \in PR \quad (4.32)$$

$$[Tc_{in}(j, p) - Tc_{out}(j, p)]. FCpc(j, p) = \sum_{k \in ST} \sum_{i \in HP} q(i, j, k, p) + q_{hu}(i, p)$$

$$j \in CP, p \in PR \quad (4.33)$$

4.3.1.2 Energy Balance at Each Stage

$$[th(i, k, p) - th(i, k + 1, p)]. FCph(i, p) = \sum_{j \in CP} q(i, j, k, p)$$

$$k \in ST, i \in HP, p \in PR \quad (4.34)$$

$$[tc(j, k, p) - tc(j, k + 1, p)]. FCpc(j, p) = \sum_{i \in HP} q(i, j, k, p)$$

$$k \in ST, j \in CP, p \in PR \quad (4.35)$$

4.3.1.3 Assignment of Inlet Temperatures

$$Th_{in}(i, p) = th(i, 1, p) \quad i \in HP, p \in PR \quad (4.36)$$

$$Tc_{in}(j, p) = tc(i, NOK + 1, p) \quad j \in CP, p \in PR \quad (4.37)$$

4.3.1.4 Monotonic Decrease in Temperatures

$$th(i, k, p) \geq th(i, k + 1, p) \quad k \in ST, i \in HP, p \in PR \quad (4.38)$$

$$tc(j, k, p) \geq tc(i, k + 1, p) \quad k \in ST, j \in CP, p \in PR \quad (4.39)$$

$$th(i, NOK + 1, p) \geq Th_{out}(i, p) \quad i \in HP, p \in PR \quad (4.40)$$

$$TC_{out}(j, p) \geq tc(j, 1, p) \quad j \in CP, p \in PR \quad (4.41)$$

4.3.1.5 Hot and Cold Utility Loads

$$[th(i, NOK + 1, p) - Th_{out}(i, p)]. FCph(i, p) = qcu(i, p) \quad i \in HP, p \in PR \quad (4.42)$$

$$[TC_{out}(j, p) - tc(j, 1, p)]. FCpc(j, p) = qhu(j, p) \quad j \in CP, p \in PR \quad (4.43)$$

4.3.1.6 Logical Constraints

$$q(i, j, k, p) - Q_{up} \cdot z(i, j, k) \leq 0 \quad i \in HP, j \in CP, k \in ST, p \in PR \quad (4.44)$$

$$qcu(i, p) - Q_{up} \cdot zcu(i) \leq 0 \quad i \in HP, p \in PR \quad (4.45)$$

$$qhu(j, p) - Q_{up} \cdot zhu(j) \leq 0 \quad j \in CP, p \in PR \quad (4.46)$$

$$z(i, j, k), zcu(i), zhu(i) \in \{0, 1\}$$

4.3.1.7 Minimum Approach Temperature Constraints

$$dt(i, j, k, p) \leq th(i, k, p) - tc(j, k, p) + DT_{up} \cdot (1 - z(i, j, k)) \\ i \in HP, j \in CP, k \in ST, p \in PR \quad (4.47)$$

$$dt(i, j, k + 1, p) \leq th(i, k + 1, p) - tc(j, k + 1, p) + DT_{up} \cdot (1 - z(i, j, k)) \\ i \in HP, j \in CP, k \in ST, p \in PR \quad (4.48)$$

$$dtcu(i, p) \leq th(i, NOK + 1, p) - Tcu_{out} + DT_{up} \cdot (1 - zcu(i)) \\ i \in HP, p \in PR \quad (4.49)$$

$$dthu(j, p) \leq Thu_{out} - tc(j, 1, p) + DT_{up} \cdot (1 - zhu(i)) \\ j \in CP, p \in PR \quad (4.50)$$

4.3.1.8 Log Mean Temperature Difference

$$LMTD(i, j, k, p) = [dt(i, j, k, p) \cdot dt(i, j, k + 1, p) \cdot \frac{dt(i, j, k, p) + dt(i, j, k + 1, p)}{2}]^{1/3}$$

$$i \in HP, j \in CP, k \in ST, p \in PR \quad (4.51)$$

$$LMTDHU(j, p) = \left[dthur(j, p) \cdot dthul(j, p) \cdot \frac{dthur(j, p) + dthul(j, p)}{2} \right]^{1/3}$$

$$j \in CP, p \in PR \quad (4.52)$$

$$LMTDCU(i, p) = \left[dtcur(i, p) \cdot dtcul(i, p) \cdot \frac{dtcur(i, p) + dtcul(i, p)}{2} \right]^{1/3}$$

$$i \in HP, p \in PR \quad (4.53)$$

4.3.1.9 Heat Exchanger Area Calculations

$$Area(i, j, k) = \frac{q(i, j, k, p)}{LMTD(i, j, k, p) \cdot U(i, j)}$$

$$i \in HP, j \in CP, k \in ST \quad (4.54)$$

$$AreaHU(j) = \frac{qhu(j, p)}{LMTDHU(j, p) \cdot UHU(j)}$$

$$j \in CP \quad (4.55)$$

$$AreaCU(i) = \frac{qcu(i, p)}{LMTDCU(i, p) \cdot UCU(i)}$$

$$i \in HP \quad (4.56)$$

4.3.1.10 Determining Maximum Area among all Periods

$$Area_max(i, j, k) \geq Area(i, j, k) \quad i \in HP, j \in CP, k \in ST, p \in PR \quad (4.57)$$

$$AreaHU_max(j) \geq AreaHU(j) \quad j \in CP, p \in PR \quad (4.58)$$

$$AreaCU_max(i) \geq AreaCU(i) \quad i \in HP, p \in PR \quad (4.59)$$

4.3.1.11 Objective Function

$$\begin{aligned} minTAC = AF & \left[\sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C_f \cdot z(i, j, k) + \sum_{i \in HP} \sum_{CU} C_f \cdot zcu(i) \right. \\ & \left. + \sum_{j \in CP} \sum_{HU} C_f \cdot zhu(j) \right] + AF \cdot \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C \cdot Area_max(i, j, k)^B \\ & + AF \cdot \sum_{j \in CP} C \cdot AreaHU_max(j)^B + AF \cdot \sum_{i \in HP} C \cdot AreaCU_max(i)^B \\ & + \sum_{p \in PR} \frac{DOP(p)}{TOP} \sum_{j \in CP} C_{hu} qhu(j, p) + \sum_{p \in PR} \frac{DOP(p)}{TOP} \sum_{i \in HP} C_{cu} qcu(i, p) \end{aligned} \quad (4.60)$$

4.3.2 Results

The case study is similar to the one that was used in sequential approach. The difference between sequential and simultaneous approaches is that the MINLP multiperiod model of simultaneous method will concurrently solve the problem which has stream properties of all periods. Thus, The HEN adaptation and HEN integration procedures were excluded.

Table 4.6 shows the results of multiperiod HEN synthesized by MINLP multiperiod model and the best HEN from sequential approach in the previous section. And Figure 4.25 shows grid diagram for multiperiod HEN of simultaneous method.

Table 4.6 Comparison of multiperiod HEN from sequential and simultaneous approach

	Sequential approach (Multiperiod HEN-3C)	Simultaneous approach
No. of heat exchangers	10	10
Total area (m ²)	6,894.5	6,899.7
Fixed cost (\$/yr)	16,666	16,666
Area cost (\$/yr)	884,840	885,515
Utility cost (\$/yr)	1,831,833	1,811,172
TAC (\$/yr)	2,733,340	2,713,354

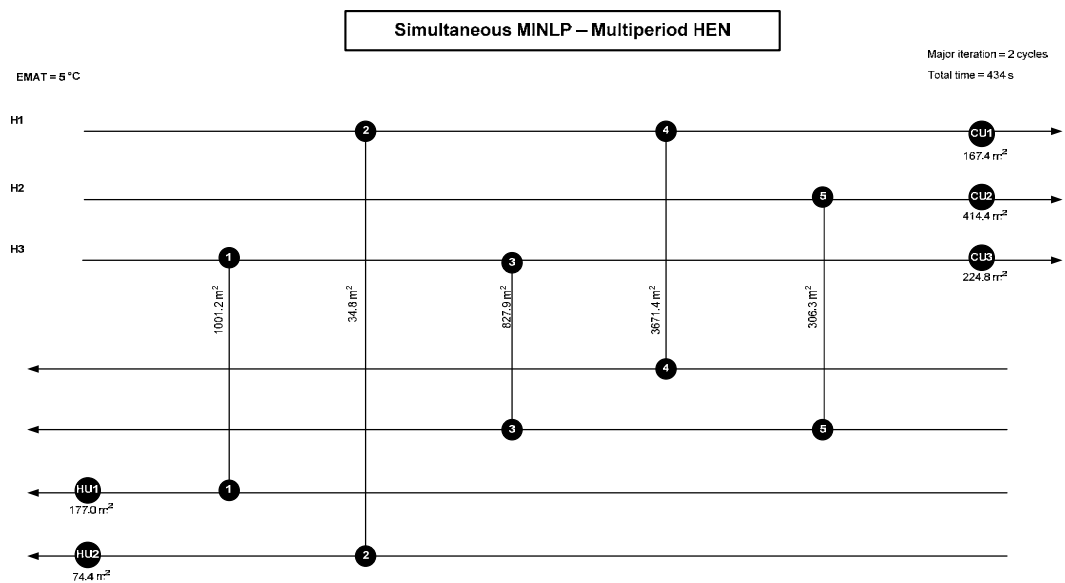


Figure 4.25 Grid diagram of multiperiod HEN synthesized by simultaneous approach.

It was found that the solution from simultaneous approach was better than that of sequential method in term of economy. Because the simultaneous

method took into account all of three periods while solving for the optimal multiperiod HEN, thus the obtained network was not trapped in sub-optimal solution like sequential method. However, this is a small case study, so the objective function of simultaneous approach was improved by just 0.74% when compared to sequential approach. Furthermore, if the structure of HEN in Figure 4.25 was considered, the topology of HEN from simultaneous method was likely to have less complication since there was not any splitted stream in the network, while there was one splitted stream existing in the best HEN from sequential approach.

4.4 Application to Crude Preheat Train in Crude Distillation Unit

Another case study which was carried out to ensure the performance of the proposed method is crude distillation unit in crude oil refinery plant. It is one of the multiperiod problems in chemical processes. Because refinery plants usually purchase various kinds of crude oil from many resources and sometimes blend it together, it will result in different compositions of crude oil. In this study, the simultaneous approach was applied to the case study to synthesize grassroots HEN for crude preheat train. The case study was simulated by using PRO/II with the data taken from Pejpichestakul (2013). After that, the necessary data was collected from the simulation

4.4.1 Case Study Simulation

The simulated case study is crude distillation unit with preflash drum in oil refinery plant. There are total 11 streams comprising of 8 hot streams and 3 cold streams. The CDU operates under three types of service; Troll (light crude), Forozan (medium crude), and Souedie (heavy crude). The crude oil assay data of each crude type is shown in Table 4.7-4.9. The distillation curves or TBP (true boiling point) curves based on the crude assays were plotted as displayed in Figure 4.26.

Table 4.7 True boiling point data of light crude

Percent Distilled	Temperature (°C)
2.2	15
5.6	65
11.9	100
22.8	150
32.3	200
41.8	250
53.2	300
63.5	350
67.2	370
80.2	450
86.5	500
91.0	550

Table 4.8 True boiling point data of medium crude

Percent Distilled	Temperature (°C)
1.3	15
6.3	75
18.0	145
33.0	230
56.8	375
80.6	560
81.2	565

Table 4.9 True boiling point data of heavy crude

Percent Distilled	Temperature (°C)	Percent Distilled	Temperature (°C)
13.9	140	43.9	350
15.3	150	45.4	360
16.7	160	46.9	370
19.4	180	48.3	380
22.0	200	49.8	390
24.7	220	51.2	400
27.5	240	58.3	450
29.0	250	61.7	475
30.4	260	65.1	500
36.3	300	68.4	525
37.8	310	71.7	550
39.4	320	73.7	565
40.9	330		

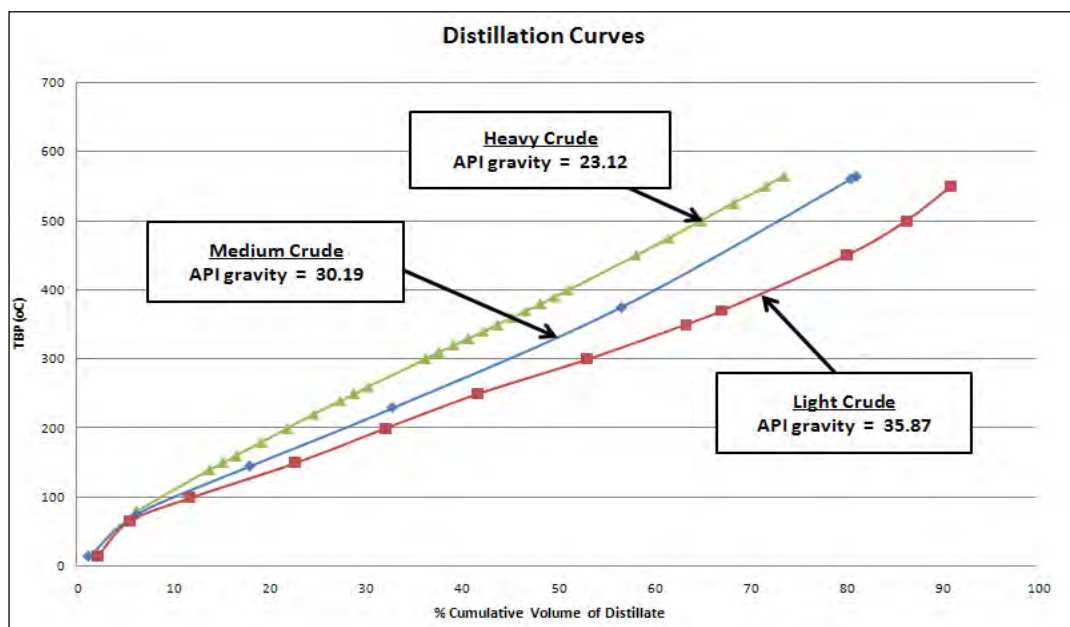


Figure 4.26 Distillation curves of each crude oil.

The API gravity data and influent flowrates are shown in Table 4.10. The assay light end composition of each crude oil is shown in Table 4.11.

Table 4.10 Crude used for simulated case study

Crude	API	Throughput (m ³ /h)
Light crude	35.87	795
Medium crude	30.19	795
Heavy crude	23.12	795

Table 4.11 Light end composition of crude

Compound	Vol%		
	Light crude	Medium crude	Heavy crude
Ethane	0.22	0.01	0
Propane	0.58	0.28	0.07
i-Butane	0.39	0.26	0.13
n-Butane	1.04	0.97	0.63

In the CDU, crude oil is first heated up to 125 °C and then fed to the desalter in order to remove salt in crude oil by dissolving in water. After salt removal, it is heated up again to 170 °C which is the operating temperature of preflash drum. The overhead vapor-phase product from flash drum is fed directly to the distillation column, and the bottom liquid-phase product is further preheated by crude preheat train and furnace to raise the temperature up to 370 °C before entering the fractionation column. Figure 4.27 illustrates the simple flow diagram of CDU with preflash drum.

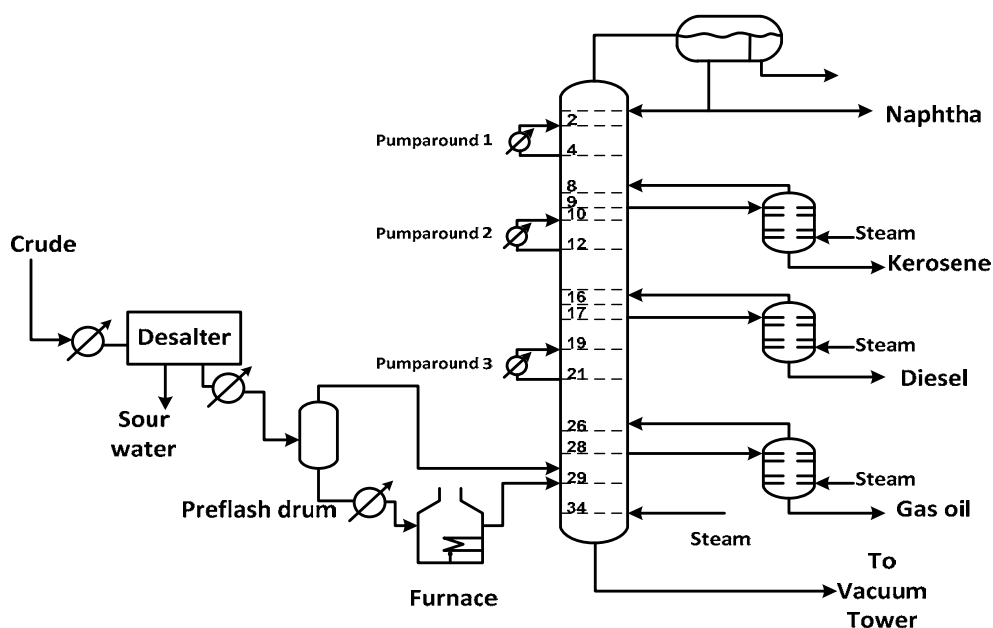


Figure 4.27 The crude distillation unit with preflash drum.

For the column specification, products are specified by using ASTM D86 (5-95 gap and 95% point). The product specification data, overflash rate, feed tray, and withdrawal tray are shown in Table 4.12. The locations of draw and return tray for pump-around and side stripper are shown in Table 4.13. The initial value, condenser outlet temperature, heat rate and return temperature of pump-arounds are illustrated in Table 4.14.

Table 4.12 Products specifications

Product	Specification	Withdrawal tray
Naphtha	D86 (95%point) = 182 °C	1
Kerosene	D86 (95%point) = 271 °C	9
Diesel	D86 (95%point) = 327 °C	17
AGO	D86 (95%point) = 410 °C	28
Overflash rate	0.01	
Kerosene-Naphtha	(5-95) Gap = 17.2 °C	
Diesel-Kerosene	(5-95) Gap = 0.6 °C	
AGO-Diesel	(5-95) Gap = -3.4 °C	
Feed tray		29
Total trays		34

Table 4.13 Column feed and side draw tray information

Number of Plates	34
Number of trays (Side Strippers)	4
Pump-around 1 (PA1) Draw	Tray 4
Pump-around 1 (PA1) Return	Tray 2
Pump-around 2 (PA2) Draw	Tray 12
Pump-around 2 (PA2) Return	Tray 10
Pump-around 3 (PA3) Draw	Tray 21
Pump-around 3 (PA3) Return	Tray 19
Kerosene Side-Stripper Return	Tray 8
Diesel Side-Stripper Return	Tray 16
AGO Side-Stripper Return	Tray 26
Crude Feed	Tray 29

Table 4.14 Column utilities information

Variable	Value		
	Light crude	Medium crude	Heavy crude
Kerosene Stripper Steam at 260 °C, 4.4 atm (kg/hr)	522.5	832.5	818.6
Diesel Stripper Steam at 260 °C, 4.4 atm (kg/hr)	2616.5	2080.6	2217.5
AGO Stripper Steam at 260 °C, 4.4 atm (kg/hr)	2843.0	1860.0	1696.0
Main Steam at 260 °C, 4.4 atm (kg/hr)	3493		
Overflash	1%		
Condenser Temperature	32.22 °C		
Pump-around 1 (PA1) Return Temperature	104.44 °C		
Pump-around 2 (PA2) Return Temperature	148.89 °C		
Pump-around 3 (PA3) Return Temperature	232.22 °C		
Pump-around 1 (PA1) Heat Rate	11.7 MW		
Pump-around 2 (PA2) Heat Rate	8.8 MW		
Pump-around 3 (PA3) Heat Rate	8.8 MW		

From the simulation, the stream properties were extracted after running completely. Stream properties, which consist of inlet and outlet temperature, heat capacity flowrates, and heat transfer coefficients, are shown in Table 4.15-4.17. The capital cost of heat exchanger is demonstrated in Eq. 4.61.

$$\text{capital cost of heat exchanger (\$)} = 26,460 + 389 * \text{Area}(m^2) \quad (4.61)$$

The process operates 350 working days with the period of 100, 150, and 100 days for light, medium, and heavy crude, respectively. The project life time of 5 years and prevailing rate of interest of 10% were assumed. Hot and cold utility costs are \$134/kW and \$6.7/kW, respectively. The exchanger minimum approach temperature (EMAT) of 5 °C was used.

Table 4.15 Stream properties of light crude (Troll)

Stream	FC _p (kW/°C)	T _{in} (°C)	T _{out} (°C)	h (kW/m ² .°C)
H1	121.02	201.17	104.44	1.293
H2	69.91	274.71	148.89	1.318
H3	98.60	321.17	232.22	1.298
H4	105.22	32.22	30.00	1.058
H5	67.76	234.40	30.00	1.395
H6	49.64	273.17	30.00	1.423
H7	59.98	326.40	30.00	1.343
H8	135.33	341.73	30.00	0.892
C1	380.57	25.00	125.00	0.654
C2	434.32	125.00	170.00	0.632
C3	585.63	166.64	370.00	0.788

Table 4.16 Stream properties of medium crude (Forozan)

Stream	FC _p (kW/°C)	T _{in} (°C)	T _{out} (°C)	h (kW/m ² .°C)
H1	125.28	198.28	104.44	1.092
H2	71.80	271.63	148.89	1.235
H3	101.36	319.12	232.22	1.270
H4	91.92	32.22	30.00	1.253
H5	56.28	225.57	30.00	1.394
H6	34.77	269.78	30.00	1.431
H7	41.91	326.26	30.00	1.413
H8	210.12	357.39	30.00	0.888
C1	387.57	25.00	125.00	0.652
C2	443.70	125.00	170.00	0.630
C3	587.80	168.84	370.00	0.782

Table 4.17 Stream properties of heavy crude (Souedie)

Stream	FC _p (kW/°C)	T _{in} (°C)	T _{out} (°C)	h (kW/m ² .°C)
H1	132.07	193.31	104.44	1.075
H2	74.03	267.77	148.89	1.221
H3	104.43	316.69	232.22	1.270
H4	70.57	32.22	30.00	1.309
H5	46.81	221.36	30.00	1.393
H6	29.33	263.57	30.00	1.438
H7	32.46	322.00	30.00	1.419
H8	268.65	353.52	30.00	0.826
C1	392.24	25.00	125.00	0.651
C2	449.76	125.00	170.00	0.630
C3	555.77	167.81	370.00	0.780

4.4.2 Results

4.4.2.1 *HEN Synthesis by Simultaneous MINLP Multiperiod Model*

The refinery case study of CDU is relatively larger scale than the adapted case study from Verheyen and Zhang (2006) since it has more process streams and different time in each period. The solution of multiperiod HEN was obtained in 326,667 seconds by using Acer Intel® Core 2 Quad Processor Q6600 (2.4 GHz). The MINLP multiperiod model, which was used to solve this case study, included some additional constraints which are minimum and maximum of both hot and cold utilities required. These equations are quite similar to boundings for total heat load. Thus, they led to the reduction of computational time. The minimum utilities required were obtained by constructing the composite curves of each period by using HRAT (heat recovery approach temperature) equal to 0 °C. For the maximum utilities required were calculated by summing up total enthalpy of hot streams (for cold utility) or cold streams (for hot utility). The composite curves for

each period are shown in Figure 4.28-4.30. The result of multiperiod HEN for crude preheat train was demonstrated by grid diagram as illustrates in Figure 4.31.

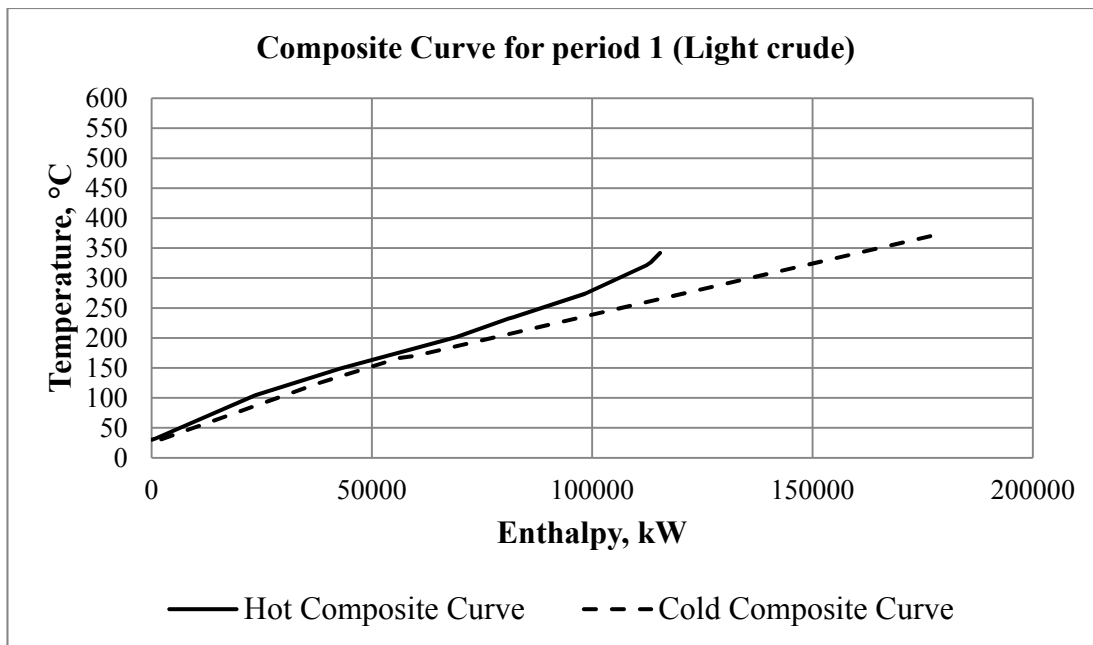


Figure 4.28 Composite curves of period 1 (Light crude).

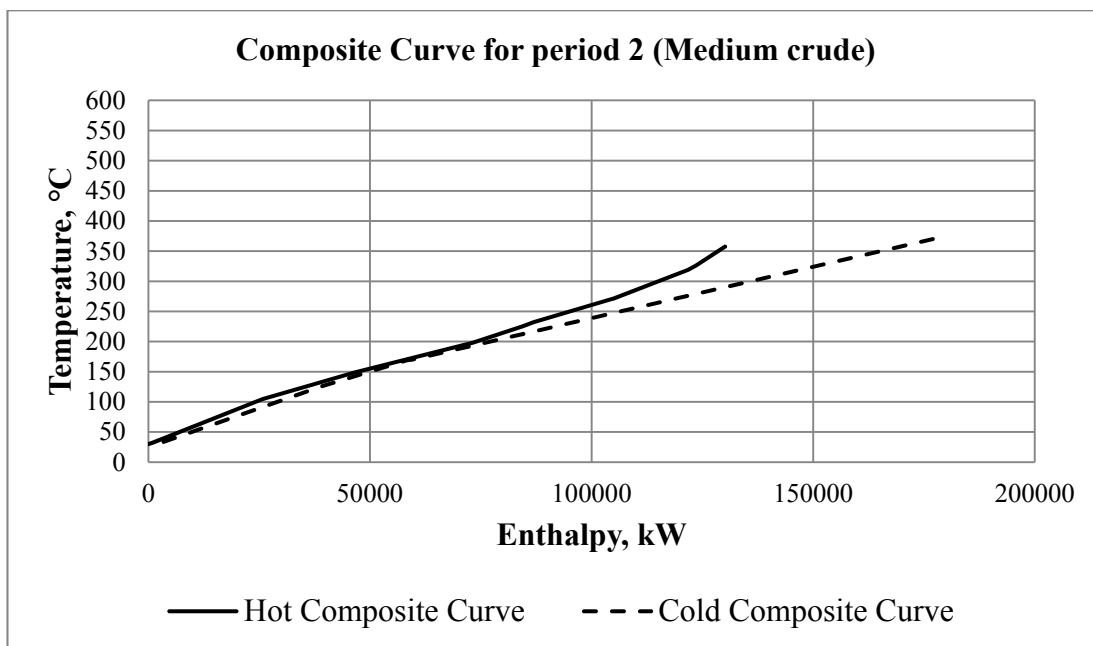


Figure 4.29 Composite curves of period 2 (Medium crude).

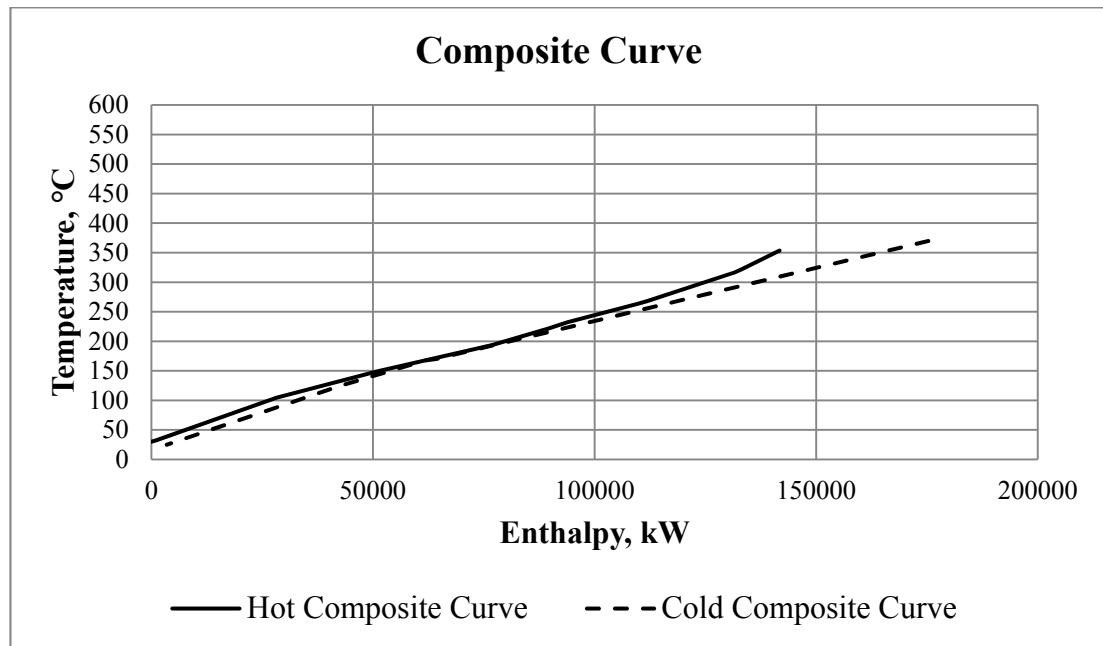


Figure 4.30 Composite curves of period 3 (Heavy crude).

The TAC of multiperiod HEN for the refinery case study was \$9,141,912 per year. There were 21 process exchangers and 8 utility exchangers. The total area of heat exchanger was 15,029 m². It demonstrated that the simultaneous MINLP multiperiod model could perform well and gave a satisfactory solution for large problem. Even though this problem was more complex, it did not require any initial feasible solution before solving. When the computational time was concerned, it spent 644,211 seconds (~ 7.5 days) to solve the problem which has 11 streams. While the small case study of VGO taken from literature with 7 streams spent only 318 seconds to solve. It could be seen that the computational time was considerably sensitive to the number of streams because it caused the number of single equations, single variables, and discrete variables to increase exponentially.

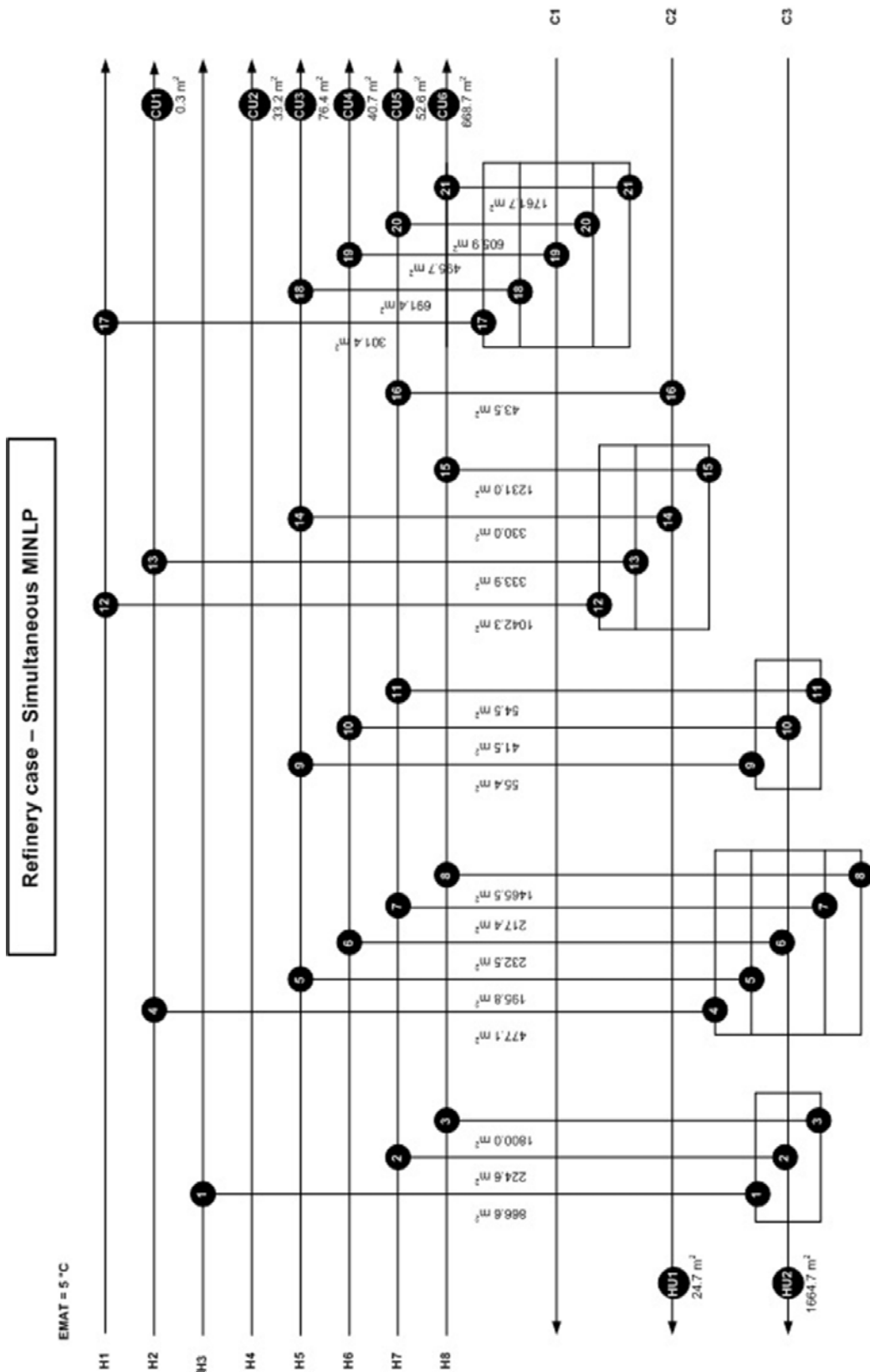


Figure 4.31 Grid diagram of multiperiod HEN for crude preheat train.

4.4.2.2 Model Improvement by Initialization Technique

Due to the overmuch time spent on solving the refinery case study, the initialization technique was adopted. Basically, initialization technique is advantageous to computational time usage and/or solution improvement.

In this study, the principle is to eliminate all nonlinear equations in MINLP multiperiod model so that it becomes an MILP model. Then the MILP model is used to preliminary solve to obtain rough solution which may close to the optimal solution. This preliminary result from MILP will be use as the initial value for every variable in MINLP multiperiod model. Therefore, the MINLP model will start solving from that initial point instead of beginning with upper and lower bound of each variable.

A. Initialization Strategy

The strategy of initialization technique is to modify all nonlinear equations to linear equations. In the simultaneous MINLP multiperiod model, the area and LMTD (log mean temperature difference) calculation are nonlinear equations. First, in LMTD calculation as illustrated in Eq. 4.62, its nonlinear terms are from the multiplication of temperature differences and the power number of 1/3. These LMTD variables were replaced by constant values which were called average LMTD (ALMTD). Second, for the area calculation as shown in Eq. 4.63, it was considered as nonlinear function because there is a division of variable $q(i,j,k,p)$ by variable $LMTD(i,j,k,p)$. Now that the variables $LMTD(i,j,k,p)$ were replaced by $ALMTD(i,j,p)$; therefore, the area equation would become a linear function.

$$LMTD(i,j,k,p) = [dt(i,j,k,p) \cdot dt(i,j,k+1,p) \cdot \frac{dt(i,j,k,p) + dt(i,j,k+1,p)}{2}]^{1/3} \quad (4.62)$$

$$Area(i,j,k) = \frac{q(i,j,k,p)}{LMTD(i,j,k,p) \cdot U(i,j)} \quad (4.63)$$

B. ALMTD calculation Methodology

ALMTD is a constant parameter calculated for each specific match of hot stream i and cold stream j . Firstly, with the assistance of potential program developed by Siemanond and Kosol (2012), optimum HRATs of each period were identified. Secondly, composite curves for each period were plotted with the use of obtained optimum HRAT. Then the composite curves were divided into many temperature intervals (n). In each interval, the LMTD was calculated by using Eq. 4.64 which is real LMTD.

$$LMTD(i, j, k) = \frac{[th(i, k) - tc(j, k)] - [th(i, k + 1) - tc(j, k + 1)]}{\ln \frac{[th(i, k) - tc(j, k)]}{[th(i, k + 1) - tc(j, k + 1)]}} \quad (4.64)$$

For every match of hot stream i and cold stream j , the ALMTDs were obtained by calculating weighted average value of LMTD and heat load (q) in overlapped region as shown in Eq. 4.65.

$$ALMTD(i, j, p) = \frac{\sum_N q(i, j, p, n) \cdot LMTD(p, n)}{\sum_N q(i, j, p, n)} \quad (4.65)$$

C. Results

The ALMTD values of every possible match in each period are shown in Table 4.18 for process-process streams and Table 4.19 for utility-process stream. For the matches which have ALMTD equal to EMAT, it means that those matches have no overlapped zone in the composite curves. However, in real situation, it is impossible for heat to exchange between those matches. Thus, the values of binary variables which indicate the existence for these matches could be preset as 0.

Table 4.18 ALMTD of match between process streams in each period

Streams		ALMTD (°C)		
		Period 1 (light crude)	Period 2 (medium crude)	Period 3 (heavy crude)
H1	C1	22.0	23.4	27.6
H1	C2	16.8	15.7	16.3
H1	C3	17.7	13.7	13.4
H2	C1	EMAT		
H2	C2	16.5	15.3	16.0
H2	C3	28.3	22.5	18.9
H3	C1	EMAT		
H3	C2	EMAT		
H3	C3	45.2	36.5	27.6
H4	C1	EMAT		
H4	C2	EMAT		
H4	C3	EMAT		
H5	C1	19.8	23.9	30.7
H5	C2	16.8	15.7	16.3
H5	C3	22.2	16.5	15.0
H6	C1	19.8	23.9	30.7
H6	C2	16.8	15.7	16.3
H6	C3	28.1	22.3	18.6
H7	C1	19.8	23.9	30.7
H7	C2	16.8	15.7	16.3
H7	C3	37.7	30.7	23.8
H8	C1	19.8	23.9	30.7
H8	C2	16.8	15.7	16.4
H8	C3	41.3	37.0	28.2

Table 4.19 ALMTD of match between process stream and utilities in each period

Stream	Utility	ALMTD (°C)		
		Period 1 (light crude)	Period 2 (medium crude)	Period 3 (heavy crude)
H1	CU	79.4	79.4	79.4
H2	CU	123.9	123.9	123.9
H3	CU	207.2	207.2	207.2
H4	CU	10.2	12.8	16.1
H5	CU	10.2	12.8	16.1
H6	CU	10.2	12.8	16.1
H7	CU	10.2	12.8	16.1
H8	CU	10.2	12.8	16.1
HU	C1	375.0	375.0	375.0
HU	C2	330.0	330.0	330.0
HU	C3	403.7	389.3	377.2

After executing the modified MILP multiperiod model with these ALMTDs, the solution was taken further to be used as the initial value for MINLP multiperiod model. The grid diagram of generated HEN by MINLP accompanying with initialization is illustrated in Figure 4.32. Its investment cost in TAC was \$9,130,627 per year. There were 21 process exchangers which was equal to the number of process exchanger from HEN without initialization technique. But the number of utility exchanger was only 4 which was less than that from HEN without initialization. The summary results of HEN with and without initialization technique is illustrated in Table 4.20. It demonstrated that the time resource required was decrease substantially by over 70 % when using the initialization strategy. Moreover, it showed that the model with initialization could generate better multiperiod HEN design than that without initialization. The TAC was improved by 0.12 %. It was obviously because the initial value obtained from MILP initialization

model was able to lead the direction of searching algorithm of solver to the better optimal solution due to the use of suitable starting point.

Table 4.20 Summary result of HEN with and without initialization for refinery case study

Parameter	Without initialization	With initialization
No. of heat exchangers	29	25
Total area (m ²)	15,029	16,079
Fixed cost (\$/yr)	202,578	174,636
Area cost (\$/yr)	1,543,422	1,651,209
Utility cost (\$/yr)	7,395,913	7,304,782
TAC (\$/yr)	9,141,913	9,130,627
Time resource (s)	644,221 (~7.5 days)	163,433 (~1.9 days)

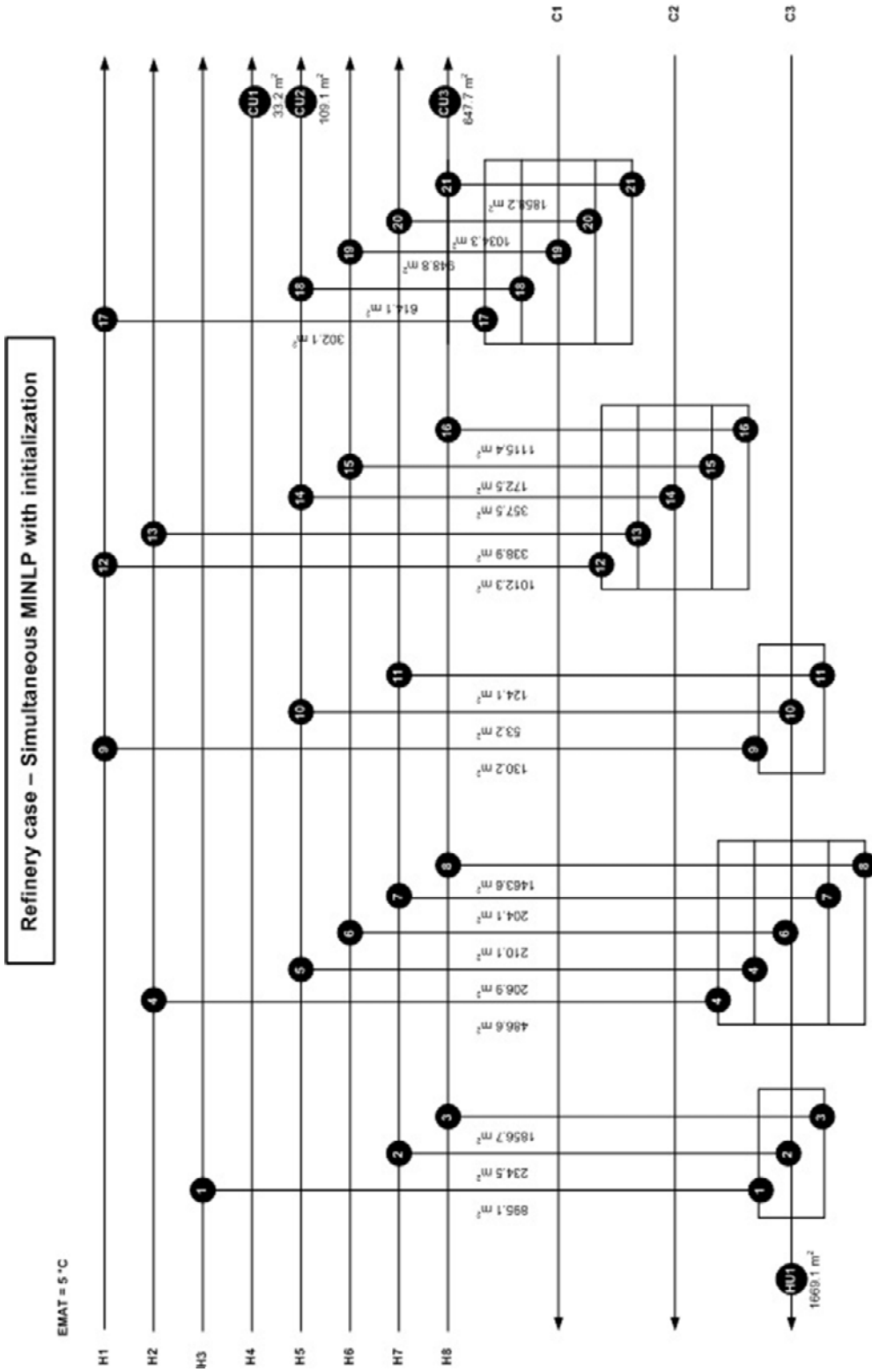


Figure 4.32 Grid diagram of multiperiod HEN for crude preheat train with initialization technique.

In order to confirm the performance of the proposed initialization strategy in term of its benefit to reduce computational time, another implementation of MINLP with initialization technique was carried out. The case study was the same refinery case study except that the time durations for light, medium, and heavy crude were changed to 150, 50, and 150 days, respectively. The summary result is shown in Table 4.21. The time resource required was decreased from 932,021 (without initialization) to 2,337 seconds (with initialization). The reduction of the time resource was even more than the last case study; however, the change of time duration for each period does not relate to the required computational time because it just changes the solution space. From the result, it can be concluded that the proposed initialization strategy could help reduce the computational time considerably with acceptable result of multiperiod HEN. It is possible to obtain better solution by allowing the model with initialization to run for longer time.

Table 4.21 Summary result of HEN with and without initialization for another refinery case study

Parameter	Without initialization	With initialization
No. of heat exchangers	30	22
Total area (m ²)	15,433	15,814
Fixed cost (\$/yr)	209,563	153,680
Area cost (\$/yr)	1,584,933	1,624,078
Utility cost (\$/yr)	7,396,134	7,418,977
TAC (\$/yr)	9,190,631	9,196,735
Time resource (s)	932,021	2,337

4.4.2.3 Validation of HEN

The final procedure of HEN synthesis for refinery case study was to validate the best obtained solution and apply back in the process simulation on PRO/II. This step would ensure the feasibility and reliability of the result from the proposed model in the real process. The data of heat exchanger areas and topology were used in the simulation.

Due to the assumption of constant heat capacity flowrates, it was found that some modifications had to be made because the outlet temperature of the process streams, which do not have utility exchangers installed at the end, did not reach the desire temperatures. Hence, some exchanger areas had to be changed by using controller tool in PRO/II to adjust the values of area. Moreover, one utility exchanger was added at the end of stream H3. The final applicable HEN was illustrated in Figure 4.33.

After validation, the TAC was recalculated because the area of some heat exchangers were changed and one more heat exchanger was added into the multiperiod HEN. Table 4.22 shows the new summary result and TAC after validation.

Table 4.22 Summary result of multiperiod HEN after validation

Parameter	Multiperiod HEN after validation
No. of heat exchangers	26
Fixed cost (\$/yr)	181,621
Area cost (\$/yr)	1,498,283
Utility cost (\$/yr)	729,955
TAC (\$/yr)	8,979,499

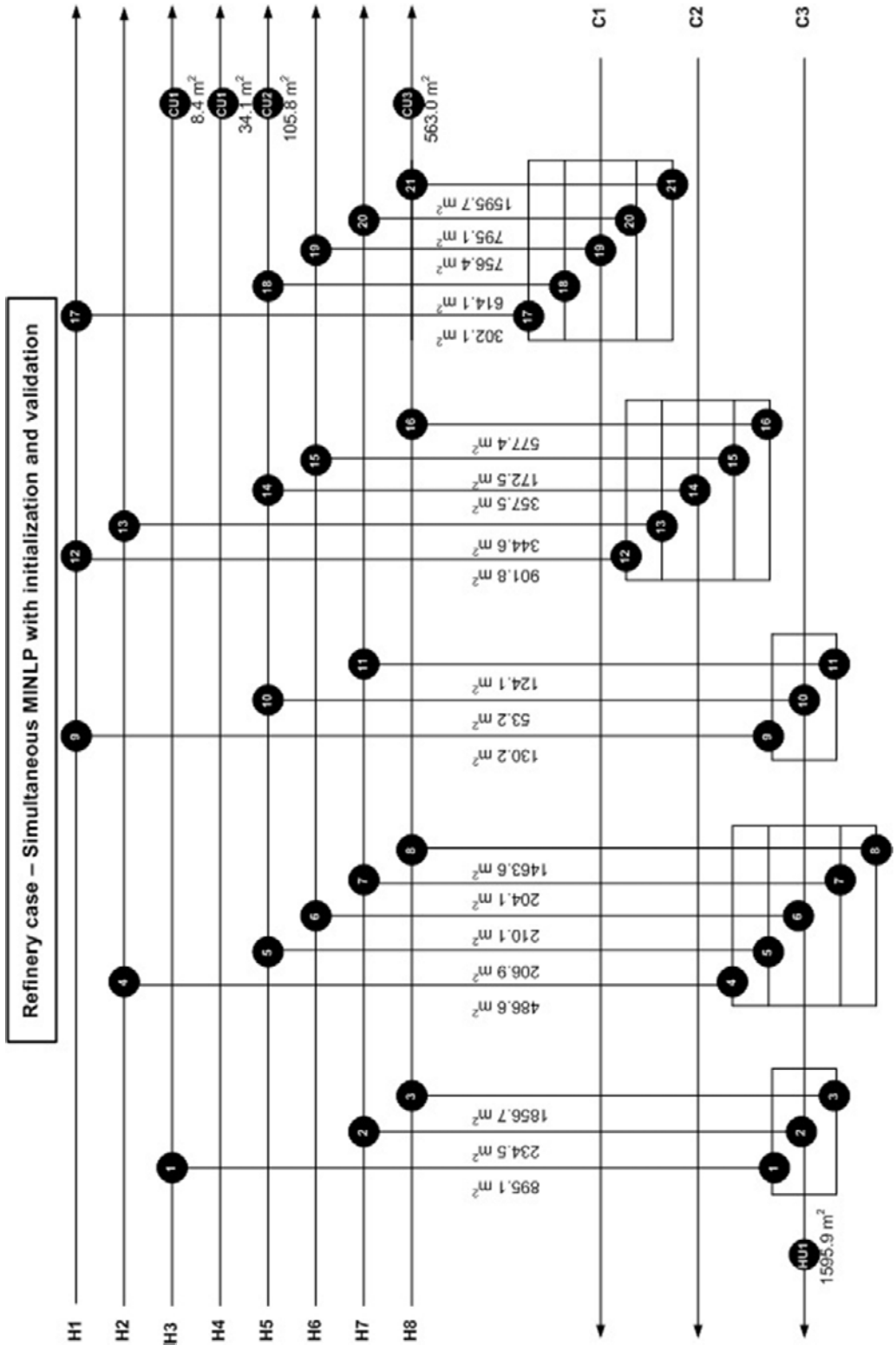


Figure 4.33 Grid diagram of validated HEN for crude preheat train with initialization technique.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

In this work, the sequential and simultaneous approaches for multiperiod HEN synthesis have been developed. The three-step sequential method comprises of HEN synthesis, HEN adaptation, and HEN integration. There are three alternative strategies can be used in HEN adaptation step. Each strategy was formulated in each model: model A is an MINLP with fixed process-exchanger area, model B is an NLP with fixed-exchanger topology using least square error of exchanger area as the objective function, and model C is an NLP with fixed-exchanger topology using minimum additional area as the objective function. As they were applied to a case study, it was found that model C could perform more effectively in HEN adaptation procedure when compared to other models because the algorithm of model is likely to retain the old existing area as much as possible. Moreover, there were less exchangers that needed to change area while changing periods. Therefore, it would be less complexity for the process control. For simultaneous approach, it was carried out using simultaneous MINLP multiperiod model which can be solved in one step.

The best sequential approach procedure and the simultaneous approach were compared by adapted case study of VGO hydrotreating process from a literature. It has been shown that the simultaneous approach can perform better than the three-step sequential approach in term of economic concern because it gave the solution with lower TAC. Furthermore, the obtained HEN from simultaneous method was less complex than that from sequential method.

Due to the higher performance of simultaneous approach, it was applied further to the refinery case study of CDU. It illustrated that the simultaneous MINLP multiperiod model could perform well and gave a satisfactory solution for large problem. Moreover, the simultaneous MINLP multiperiod model is quite rigorous since no initial feasible solution is needed for both of two case studies. But, as the problem size increases, the computational time required is also increased substantially. From this problem, an initialization technique was hence developed to find an initial feasible solution for the MINLP model. The average LMTDs (ALMTDs) of each stream match and each period were calculated with the

utilization of composite curves. These values of ALMTD were used in modified MILP model in order to generate rough solution which was entered as the initial value in the MINLP model. In conclusion, the initialization technique was successfully applied to the model. It could help reduce the time resource dramatically; moreover, it could improve the solution of HEN design.

It is suggested that when dealing with large problem that consists of many streams, the sequential approach would be more preferable because of a tendency to use less computational time. Or another way is to use the simultaneous approach with initialization technique.

The simultaneous MINLP multiperiod model on the other hand is limited due to the fact that the generated superstructure does not include the non-isothermal mixing feature. And a series of heat exchangers in splitted streams could have probably caused some improvement on the solution. Moreover, it should be worthwhile to use GAMS for automated calculation of ALMTDs without any hand calculation.

REFERENCES

- Aaltola, J. (2002) Simultaneous synthesis of flexible heat exchanger network. Applied Thermal Engineering 22(8), 907-918.
- Biegler, L.T., Grossmann, I.E., and Westerberg, A.W. (1997) Systematic Methods of Chemical Process Design. Upper Saddle River (US): Pearson Education.
- Björk, K.M. and Westerlund, T. (2002) Global optimization of heat exchanger network synthesis problems with and without the isothermal mixing assumption. Computers & Chemical Engineering 26(11), 1581-1593.
- Briones, V. and Kokossis, A.C. (1999) Hypertargets: a Conceptual Programming approach for the optimisation of industrial heat exchanger networks — II. Retrofit design. Chemical Engineering Science 54(4), 541-561.
- Chen, C.L. and Hung, P.S. (2004) Simultaneous synthesis of flexible heat-exchange networks with uncertain source-stream temperatures and flow rates. Industrial & Engineering Chemistry Research 43(18), 5916-5928.
- Chen, J.J.J. (1987) Comments on improvements on a replacement for the logarithmic mean. Chemical Engineering Science 42(10), 2488-2489.
- Ciric, A.R. and Floudas, C.A. (1991) Heat exchanger network synthesis without decomposition. Computers & Chemical Engineering 15(6), 385-396.
- Floudas, C.A. (1995) Nonlinear and Mixed-Integer Optimization. New York: Oxford University Press.
- Floudas, C.A. and Ciric, A.R. (1989) Strategies for overcoming uncertainties in heat exchanger network synthesis. Computers & Chemical Engineering 13(10), 1133-1152.
- Floudas, C.A., Ciric, A.R., and Grossmann, I.E. (1986) Automatic synthesis of optimum heat exchanger network configurations. AIChE Journal 32(2), 276-290.
- Floudas, C.A. and Grossmann, I.E. (1986) Synthesis of flexible heat exchanger networks for multiperiod operation. Computers & Chemical Engineering 10(2), 153-168.

- Floudas, C.A. and Grossmann, I.E. (1987) Automatic generation of multiperiod heat exchanger network configurations. Computers & Chemical Engineering 11(2), 123-142.
- Hohmann, E.C. (1971) Optimum networks for heat exchange. Ph.D. Thesis. University of Southern California, Los Angeles, California, USA.
- Iyer, R. and Grossmann, I. (1996) Global Optimization of Heat Exchanger Networks with Fixed Configuration for Multiperiod Design. In I.E. Grossmann, Global optimization in engineering design (pp. 289-308). Boston: Kluwer Academic.
- Linnhoff, B. and Flower, J.R. (1978) Synthesis of heat exchanger networks: I. Systematic generation of energy optimal networks. AIChE Journal 24(4), 633-642.
- Linnhoff, B. and Flower, J.R. (1978) Synthesis of heat exchanger networks: II. Evolutionary generation of networks with various criteria of optimality. AIChE Journal 24(4), 642-654.
- Ma, X., Yao, P., Luo, X., and Roetzel, W. (2008) Synthesis of multi-stream heat exchanger network for multi-period operation with genetic/simulated annealing algorithms. Applied Thermal Engineering 28(8–9), 809-823.
- Papalexandri, K.P. and Pistikopoulos, E.N. (1994) Synthesis and retrofit design of operable heat exchanger networks. 1. Flexibility and structural controllability aspects. Industrial & Engineering Chemistry Research 33(7), 1718-1737.
- Papalexandri, K.P. and Pistikopoulos, E.N. (1994) Synthesis and retrofit design of operable heat exchanger networks. 2. Dynamics and control structure considerations. Industrial & Engineering Chemistry Research 33(7), 1738-1755.
- Papoulias, S.A. and Grossmann, I.E. (1983) A structural optimization approach in process synthesis—II: Heat recovery networks. Computers & Chemical Engineering 7(6), 707-721.
- Pejpichestakul, W. (2013) Retrofit of distillation column in petrochemical processes. M.S. Thesis. The Petroleum and Petrochemical College, Chulalongkorn University, Bangkok, Thailand.

- Paterson, W.R. (1984) A replacement for the logarithmic mean. Chemical Engineering Science 39(11), 1635-1636.
- Dawe, R.A. (2000) Modern Petroleum Technology. New York: Wiley.
- Quesada, I. and Grossmann, I.E. (1993) Global optimization algorithm for heat exchanger networks. Industrial & Engineering Chemistry Research 32(3), 487-499.
- Saboo, A.K., Morari, M., and Woodcock, D.C. (1985) Design of resilient processing plants—VIII. A resilience index for heat exchanger networks. Chemical Engineering Science 40(8), 1553-1565.
- Shenoy, U.V. (1995) Heat Exchanger Network Synthesis: Optimization Process by Energy and Resource Analysis. Texas: Gulf Publishing Company.
- Siemanond, K. and Kosol, S. (2012) Heat exchanger network retrofit by pinch design method using stage-model mathematical programming. Chemical Engineering Transaction 29, 367-372.
- Sieniutycz, S. and Jezowski, J. (2009) Energy Optimization in Process Systems: Elsevier Science.
- Smith, R. (2005). Chemical Process: Design and Integration: Wiley.
- Swaney, R.E. and Grossmann, I.E. (1985) An index for operational flexibility in chemical process design. Part I: Formulation and theory. AIChE Journal 31(4), 621-630.
- Tantimuratha, L., Asteris, G., Antonopoulos, D.K., and Kokossis, A.C. (2001) A conceptual programming approach for the design of flexible HENs. Computers & Chemical Engineering 25(4-6), 887-892.
- Verheyen, W. and Zhang, N. (2006) Design of flexible heat exchanger network for multi-period operation. Chemical Engineering Science 61(23), 7730-7753.
- Yee, T.F. and Grossmann, I.E. (1990) Simultaneous optimization models for heat integration—II. Heat exchanger network synthesis. Computers & Chemical Engineering 14(10), 1165-1184.
- Yee, T.F., Grossmann, I.E., and Kravanja, Z. (1990) Simultaneous optimization models for heat integration—I. Area and energy targeting and modeling of multi-stream exchangers. Computers & Chemical Engineering 14(10), 1151-1164.

- Yee, T.F., Grossmann, I.E., and Kravanja, Z. (1990) Simultaneous optimization models for heat integration—III. Process and heat exchanger network optimization. Computers & Chemical Engineering 14(11), 1185-1200.
- Zhu, X.X. (1995) Automated synthesis of HENs using block decomposition and heuristic rules. Computers & Chemical Engineering 19, Supplement 1(0), 155-160.
- Zhu, X.X. (1997) Automated design method for heat exchanger network using block decomposition and heuristic rules. Computers & Chemical Engineering 21(10), 1095-1104.
- Zhu, X.X., O'Neill, B.K., Roach, J.R., and Wood, R.M. (1995) Area-targeting methods for the direct synthesis of heat exchanger networks with unequal film coefficients. Computers & Chemical Engineering 19(2), 223-239.
- Zhu, X.X., O'Neill, B.K., Roach, J.R., and Wood, R.M. (1995) A new method for heat exchanger network synthesis using area targeting procedures. Computers & Chemical Engineering 19(2), 197-222.

APPENDICES

Appendix A Source Code of MINLP Single Period Model

The following text is source code which is entered in GAMS for synthesizing HEN for single period.

SETS

```
I hot streams          /I1*I3/  
J cold streams        /J1*J4/  
K index of stage or location /K1*K4,KL/ //last number is no. of stages
```

```
//dynamic sets//
```

```
stage(k) all stages  
;  
stage(k) = yes; stage('KL') = no;
```

SCALARS

```
//Cost//
```

```
CA per unit cost of heat exchanger area($ per unit) /641.7/  
B exponent for area cost(dimensionless) /1/  
CF fixed cost of heat exchanger($ per unit) /8333.3/  
  
CHU per unit cost of hot utility($ per kW) /115.2/  
CCU fixed cost of heat exchanger($ per kW) /1.3/  
  
AF annualisation factor /0.2/
```

```
//other//
```

```
EMAT exchanger minimum approach temperature(C) /5/  
  
HHU heat transfer coefficient of hot utility(kW|(m2.C)) /2/  
HCU heat transfer coefficient of cold utility(kW|(m2.C)) /1/  
;
```

PARAMETERS

```
//Hot Streams//
```

```
THIN(i) inlet temperature of hot streams(C)  
/ I1 406  
I2 160  
I3 362  
/  
THOUT(i) outlet temperature of hot streams(C)  
/ I1 60
```

```

        I2  40
        I3  60
    /
//Cold Streams//
    TCIN(j) inlet temperature of cold streams(C)
    /      J1  72
           J2  62
           J3  220
           J4  250
    /
    TCOUT(j) outlet temperature of cold streams(C)
    /      J1  365
           J2  210
           J3  370
           J4  290
    /
//Heat Capacity Flowrates//
    FCPH(i) heat capaity flowrates fo hot streams(C)
    /      I1  205.0
           I2  198.8
           I3  136.4
    /
    FCPC(j) heat capaity flowrates fo cold streams(C)
    /      J1  210.3
           J2  141.0
           J3  175.4
           J4  318.7
    /
//Utility//
    THUIN(j) inlet temperature of hot utility(C)
    /      J1  400
           J2  400
           J3  400
           J4  400
    /
    THUOUT(j) outlet temperature of hot utility(C)
    /      J1  399
           J2  399
           J3  399
           J4  399
    /
    TCUIN(i) inlet temperature of cold utility(C)
    /      I1  15
           I2  15
           I3  15
    /
    TCUOUT(i) outlet temperature of cold utility(C)

```

```

/      I1  20
      I2  20
      I3  20
/
//Heat transfer coefficient//
HH(i)  heat capacity flowrates of hot streams(kW|(m2.C))
/      I1  2
      I2  2
      I3  2
/
HC(j)  heat capacity flowrates of cold streams(kW|(m2.C))
/      J1  1.5
      J2  1.5
      J3  2
      J4  2
/
//overall heat transfer coefficient//
U(i,j)      Overall heat transfer coefficient for match(i,j)
UHU(j)      Overall heat transfer coefficient for cold stream(j) and hot utility
UCU(i)      Overall heat transfer coefficient for hot stream(i) and cold utility

//upper bound for logical constraints//
GAMMA_HX(i,j,k)  upper bound heat load for match (i,j)
GAMMA_HU(j)      upper bound heat load for hot utility
GAMMA_CU(i)      upper bound heat load for cold utility

TAU_HX(i,j,k)   upper bound of temperature difference for heat exchanger
TAU_HUL(j)      upper bound of temperature difference for left side of hot
utility exchanger
TAU_HUR(j)      upper bound of temperature difference for right side of hot
utility exchanger
TAU_CUL(i)      upper bound of temperature difference for left side pf cold
utility exchanger
TAU_CUR(i)      upper bound of temperature difference for right side of cold
utility exchanger

//parameter for answer verification//
z1(i,j,k)       correct existence of match(i,j) at stage k
zhu1(j)         correct existence of hot utility for cold stream j
zcu1(i)         correct existence of cold utility for hot stream i

dt1(i,j,k)      correct temperature difference(C) for match(i,j) at location k
dthul1(j)       correct temperature difference(C) for left side of hot utility ex
changer
dthur1(j)       correct temperature difference(C) for right side of hot utility ex
changer

```

dtcul1(i) correct temperature difference(C) for left side of cold utility exchanger
 dtcur1(i) correct temperature difference(C) for right side of cold utility exchanger

 LMTD1(i,j,k) correct log mean temperature difference of match(i,j) at stage k
 LMTDHU1(j) correct log mean temperature difference of hot utility exchanger for cold stream j
 LMTDCU1(i) correct log mean temperature difference of cold utility exchanger for cold stream i

 AREAHX1(i,j,k) correct area of heat exchanger of match(i,j) at stage k
 AREAHU1(j) correct area of hot utility exchanger of cold stream j
 AREACU1(i) correct area of cold utility exchanger of hot stream i

 Cost_fix1 correct fixed cost
 Cost_area1 correct area cost
 Cost_utility1 correct utility cost

 TAC1 correct total annualized cost

 TOT_AREA total area of heat exchangers including utility exchangers
 TOT_HU total hot utility
 TOT_CU total cold utility
 ;

//Overall heat transfer coefficient calculations//

$U(i,j) = 1/(1/HH(i)+1/HC(j));$
 $UHU(j) = 1/(1/HC(j)+1/HHU);$
 $UCU(i) = 1/(1/HH(i)+1/HCU);$

VARIABLES

th(i,k) temperature(C) of hot streams i at location k
 tc(j,k) temperature(C) of cold streams j at location k

 q(i,j,k) heat load(kW) of match(i,j) at stage k
 qhu(j) heat load(kW) of hot utility for cold stream j
 qcu(i) heat load(kW) of cold utility for hot stream i

 z(i,j,k) existence of match(i,j) at stage k
 zhu(j) existence of hot utility for cold stream j
 zcu(i) existence of cold utility for hot stream i

 dt(i,j,k) temperature difference(C) for match(i,j) at location k
 dthul(j) temperature difference(C) for left side of hot utility exchanger
 dthur(j) temperature difference(C) for right side of hot utility exchanger
 dtcul(i) temperature difference(C) for left side of cold utility exchanger

dtcur(i) temperature difference(C) for right side of cold utility exchanger

LMTD(i,j,k) log mean temperature difference of match(i,j) at stage k

LMTDHU(j) log mean temperature difference of hot utility exchanger for cold stream j

LMTDCU(i) log mean temperature difference of cold utility exchanger for cold stream i

AREA(i,j,k) area of heat exchanger of match(i,j) at stage k

AREAHU(j) area of hot utility exchanger of cold stream j

AREACU(i) area of cold utility exchanger of hot stream i

Cost_fix fixed cost

Cost_area area cost

Cost_utility utility cost

TAC total annualized cost

;

POSITIVE VARIABLES q(i,j,k), qhu(j), qcu(i),
dt(i,j,k), dthul(j), dthur(j), dtcul(i), dtcur(i),
LMTD(i,j,k),LMTDHU(j),LMTDCU(i),
AREA(i,j,k),AREAHU(j),AREACU(i)
;

BINARY VARIABLES z(i,j,k), zhu(j), zcu(i) ;

//variable bounding//

th.up(i,k) = THIN(i);

th.lo(i,k) = THOUT(i);

tc.up(j,k) = TCOUT(j);

tc.lo(j,k) = TCIN(j);

q.up(i,j,k)\$stage(k) = max(min(FCPH(i)*(th.up(i,k)-
max(th.lo(i,k+1),tc.lo(j,k+1)+EMAT)) , FCPC(j)*(min(th.up(i,k)-
EMAT,tc.up(j,k))-tc.lo(j,k+1))) , 0)\$((th.up(i,k)-tc.lo(j,k) >= EMAT) and
(th.up(i,k+1)-tc.lo(j,k+1) >= EMAT)) +0\$((th.up(i,k)-tc.lo(j,k) < EMAT) or
(th.up(i,k+1)-tc.lo(j,k+1) < EMAT)) ;

qhu.up(j) = max(FCPC(j)*(min(THUIN(j)-EMAT,TCOUT(j))-
tc.lo(j,'K1')) , 0)\$((THUIN(j)-TCOUT(j) >= EMAT) and (THUOUT(j)-
tc.lo(j,'K1') >= EMAT)) +0\$((THUIN(j)-TCOUT(j) < EMAT) or (THUOUT(j)-
tc.lo(j,'K1') < EMAT)) ;

$qcu.up(i) = \max(FCPH(i)*(th.up(i,'KL') - \max(THOUT(i), TCUIN(i) + EMAT)), 0) * ((th.up(i,'KL') - TCUOUT(i) \geq EMAT) \text{ and } (THOUT(i) - TCUIN(i) \geq EMAT)) + 0 * ((th.up(i,'KL') - TCUOUT(i) < EMAT) \text{ or } (THOUT(i) - TCUIN(i) < EMAT));$

$z.up(i,j,k) \$stage(k) = 0 * (q.up(i,j,k) = 0) + 1 * (q.up(i,j,k) \neq 0);$
 $zhu.up(j) = 0 * (qhu.up(j) = 0) + 1 * (qhu.up(j) \neq 0);$
 $zcu.up(i) = 0 * (qcu.up(i) = 0) + 1 * (qcu.up(i) \neq 0);$

$dt.up(i,j,k) = \max(EMAT, (th.up(i,k) - tc.lo(j,k)));$
 $dt.lo(i,j,k) = EMAT;$
 $dthul.up(j) = \max(EMAT, THUIN(j) - TCOUT(j));$
 $dthul.lo(j) = EMAT;$
 $dthur.up(j) = \max(EMAT, THUOUT(j) - tc.lo(j,'K1'));$
 $dthur.lo(j) = EMAT;$
 $dteul.up(i) = \max(EMAT, th.up(i,'KL') - TCUOUT(i));$
 $dteul.lo(i) = EMAT;$
 $dteur.up(i) = \max(EMAT, THOUT(i) - TCUIN(i));$
 $dteur.lo(i) = EMAT;$

$LMTD.up(i,j,k) \$stage(k) = (dt.up(i,j,k) * dt.up(i,j,k+1) * (dt.up(i,j,k) + dt.up(i,j,k+1)) / 2) ** (1/3);$
 $LMTD.lo(i,j,k) \$stage(k) = (dt.lo(i,j,k) * dt.lo(i,j,k+1) * (dt.lo(i,j,k) + dt.lo(i,j,k+1)) / 2) ** (1/3);$
 $LMTDHU.up(j) = (dthur.up(j) * dthul.up(j) * (dthur.up(j) + dthul.up(j)) / 2) ** (1/3);$
 $LMTDHU.lo(j) = (dthur.lo(j) * dthul.lo(j) * (dthur.lo(j) + dthul.lo(j)) / 2) ** (1/3);$
 $LMTDCU.up(i) = (dteul.up(i) * dteur.up(i) * (dteul.up(i) + dteur.up(i)) / 2) ** (1/3);$
 $LMTDCU.lo(i) = (dteul.lo(i) * dteur.lo(i) * (dteul.lo(i) + dteur.lo(i)) / 2) ** (1/3);$

$AREA.up(i,j,k) \$stage(k) = q.up(i,j,k) / LMTD.lo(i,j,k) / U(i,j);$
 $AREAHU.up(j) = qhu.up(j) / LMTDHU.lo(j) / UHU(j);$
 $AREACU.up(i) = qcu.up(i) / LMTDCU.lo(i) / UCU(i);$

//Assign bounding to pamaters in logical constraint//

$GAMMA_HX(i,j,k) = q.up(i,j,k);$
 $GAMMA_HU(j) = qhu.up(j);$
 $GAMMA_CU(i) = qcu.up(i);$

$TAU_HX(i,j,k) = -th.lo(i,k) + tc.up(j,k) + dt.lo(i,j,k);$
 $TAU_HUL(j) = -THUIN(j) + TCOUT(j) + dthul.lo(j);$
 $TAU_HUR(j) = -THUOUT(j) + tc.up(j,'K1') + dthur.lo(j);$
 $TAU_CUL(i) = -th.lo(i,'KL') + TCUOUT(i) + dteul.lo(i);$
 $TAU_CUR(i) = -THOUT(i) + TCUIN(i) + dteur.lo(i);$

//Last stage forcing bound//

q.fx(i,j,'KL') =0;
 z.fx(i,j,'KL') =0;
 LMTD.fx(i,j,'KL') =0;
 AREA.fx(i,j,'KL') =0;

EQUATIONS

OVERALL_H(i) overall energy balance of hot stream i
 OVERALL_C(j) overall energy balance of cold stream j

EBAL_H(i,k) energy balance of hot stream i in stage k
 EBAL_C(j,k) energy balance of cold stream j in stage k

TIN_H(i) assignment inlet temperature of hot stream i
 TIN_C(j) assignment inlet temperature of cold stream j

MONOT_H(i,k) monotonic in temperature of hot stream i at location k
 MONOT_C(j,k) monotonic in temperature of cold stream j at location k
 MONOT_HU(j) monotonic in outlet temperature of cold stream j
 MONOT_CU(i) monotonic in outlet temperature of hot stream i

EBAL_HU(j) energy balance of hot utility of cold stream j
 EBAL_CU(i) energy balance of cold utility of hot stream i

LOGIC_HX(i,j,k) logical constraint to define z of match(i,j) at stage k
 LOGIC_HU(j) logical constraint to define zhu of cold stream j at stage k
 LOGIC_CU(i) logical constraint to define zcu of hot stream i at stage k

DTFEAS_HXL(i,j,k) temperature feasibility of heat exchanger of match(i,j)
 at location k
 DTFEAS_HXR(i,j,k) temperature feasibility of heat exchanger of match(i,j)
 at location k+1
 DTFEAS_HUL(j) temperature feasibility of hot utility of outlet cold
 stream j
 DTFEAS_HUR(j) temperature feasibility of hot utility of inlet cold stream j
 DTFEAS_CUL(i) temperature feasibility of cold utility of inlet hot stream i
 DTFEAS_CUR(i) temperature feasibility of cold utility of outlet hot stream i

LOGMTD(i,j,k) log mean temperature of match(i,j) at stage k
 LOGMTDHU(j) log mean temperature of hot utility exchanger
 LOGMTDCU(i) log mean temperature of cold utility exchanger

A_HX(i,j,k) area of heat exchanger of match(i,j) at stage k
 A_HU(j) area of hot utility exchanger of cold stream j
 A_CU(i) area of cold utility exchanger of hot stream i

FIXCOST fixed cost
 AREACOST area cost
 UTILITYCOST utility cost

OBJ objective function (total annualized cost)
 ;

OVERALL_H(i) .. (THIN(i) -THOUT(i))*FCPH(i) =e=
 sum((j,k),q(i,j,k))+qcu(i);
 OVERALL_C(j) .. (TCOUT(j)-TCIN(j)) *FCPC(j) =e=
 sum((i,k),q(i,j,k))+qhu(j);

EBAL_H(i,k)\$stage(k) .. (th(i,k)-th(i,k+1))*FCPH(i) =e= sum(j,q(i,j,k));
 EBAL_C(j,k)\$stage(k) .. (tc(j,k)-tc(j,k+1))*FCPC(j) =e= sum(i,q(i,j,k));

TIN_H(i) .. THIN(i) =e= th(i,'K1');
 TIN_C(j) .. TCIN(j) =e= tc(j,'KL');

MONOT_H(i,k)\$stage(k) .. th(i,k) =g= th(i,k+1);
 MONOT_C(j,k)\$stage(k) .. tc(j,k) =g= tc(j,k+1);
 MONOT_HU(j) .. TCOUT(j) =g= tc(j,'K1');
 MONOT_CU(i) .. THOUT(i) =l= th(i,'KL');

EBAL_HU(j) .. (TCOUT(j) - tc(j,'K1'))*FCPC(j) =e= qhu(j);
 EBAL_CU(i) .. (th(i,'KL') - THOUT(i))*FCPH(i) =e= qcu(i);

LOGIC_HX(i,j,k)\$stage(k) .. q(i,j,k) - GAMMA_HX(i,j,k)*z(i,j,k) =l= 0;
 LOGIC_HU(j) .. qhu(j) - GAMMA_HU(j) *zhu(j) =l= 0;
 LOGIC_CU(i) .. qcu(i) - GAMMA_CU(i) *zcu(i) =l= 0;

DTFEAS_HXL(i,j,k)\$stage(k) .. dt(i,j,k) =l= th(i,k) -tc(j,k) +TAU_HX(i,j,k)
 *(1-z(i,j,k));
 DTFEAS_HXR(i,j,k)\$stage(k) .. dt(i,j,k+1) =l= th(i,k+1) -tc(j,k+1)
 +TAU_HX(i,j,k+1)*(1-z(i,j,k));
 DTFEAS_HUL(j) .. dthul(j) =l= THUIN(j) -TCOUT(j) +TAU_HUL(j)
 *(1-zhu(j));
 DTFEAS_HUR(j) .. dthur(j) =l= THUOUT(j) -tc(j,'K1')+TAU_HUR(j)
 *(1-zhu(j));
 DTFEAS_CUL(i) .. dtcul(i) =l= th(i,'KL')-TCUOUT(i) +TAU_CUL(i)
 *(1-zcu(i));
 DTFEAS_CUR(i) .. dtcur(i) =l= THOUT(i) -TCUIN(i) +TAU_CUR(i)
 *(1-zcu(i));

LOGMTD(i,j,k)\$stage(k) .. LMTD(i,j,k) =e=
 (dt(i,j,k)*dt(i,j,k+1)*(dt(i,j,k)+dt(i,j,k+1))/2)**(1/3);

```

LOGMTDHU(j)      .. LMTDHU(j) =e= (dthur(j) *dthul(j) *(dthur(j)
                  +dthul(j)) /2)**(1/3);
LOGMTDCU(i)      .. LMTDCU(i) =e= (dteul(i) *dteur(i) *(dteul(i)
                  +dteur(i)) /2)**(1/3);

A_HX(i,j,k)$stage(k) .. AREA(i,j,k) *LMTD(i,j,k)*U(i,j) =e= q(i,j,k);
A_HU(j)          .. AREAHU(j) *LMTDHU(j) *UHU(j) =e= qhu(j);
A_CU(i)          .. AREACU(i) *LMTDCU(i) *UCU(i) =e= qcu(i);

FIXCOST          .. cost_fix =e= AF*(sum((i,j,k)$stage(k),CF*z(i,j,k)) +
                  sum(j,CF*zhu(j)) + sum(i,CF*zcu(i)));
AREACOST         .. cost_area =e=
                  AF*(sum((i,j,k)$stage(k),CA*AREA(i,j,k)))
                  + AF*(sum(j,CA*AREAHU(j)))
                  + AF*(sum(i,CA*AREACU(i)));
UTILITYCOST      .. cost_utility =e= sum(j,CHU*qhu(j)) +
                  sum(i,CCU*qcu(i));

OBJ              .. TAC =e= cost_fix + cost_area + cost_utility;

model STAGEMODEL_SINGLEPERIOD /all/;

option iterlim=1e9;
option domlim=0;
option reslim=1e8;
STAGEMODEL_SINGLEPERIOD.optfile=0;

solve STAGEMODEL_SINGLEPERIOD using MINLP minimizing TAC;

display th.l,tc.l,
        z.l,zhu.l,zcu.l,
        q.l,qhu.l,qcu.l,
        AREA.l,AREAHU.l,AREACU.l,
        dt.l,dthur.l,dthul.l,dteul.l,dteur.l,
        LMTD.l,
        TAC.l,
        TAU_HX,
        q.up,q.lo,qhu.up,qhu.lo,qcu.up,qcu.lo,
        dt.up,dt.lo,dthul.lo,dthul.up, dthur.lo,dthur.up, dteul.lo,dteul.up,
        dteur.lo,dteur.up,
        cost_fix.l, cost_area.l, cost_utility.l;

```

Appendix B Source Code of Models for HEN Adaptation

The following text is source code which is entered in GAMS for each model used in HEN adaptation step. Please note that this is just an example for only one case.

B1 Model A

SETS

```
I hot streams          /I1*I3/
J cold streams         /J1*J4/
K index of stage or location /K1*K4,KL/ //last number is no. of stages
```

```
stage(k)              all stages
HEN(i,j,k)            define where HX exists
HEN_HOT(i,k)          define where HX exists 2 dimensions for hot streams
HEN_COLD(j,k)         define where HX exists 2 dimensions for cold streams
HU(j)                 define where HU exists
HUNEW_HU(j)           define where HU could have in cold stream
HUNEW_CU(j)           define where CU could have in cold stream
CU(i)                 define where CU exists
;
```

//dynamic set for stage pointer

```
stage(k) = yes;
stage('KL') = no;
```

//dynamic set for existing HEN// <---- To fix topology

```
HEN(i,j,k) = no;
```

TABLE Z_OLD(i,j,k) Existing HX from original HEN of each period

	K1	K2	K3	K4
I1.J1	1.000	1.000	1.000	
I1.J2			1.000	1.000
I1.J3		1.000		
I3.J1		1.000	1.000	
I3.J4	1.000			

```
;
```

```
loop((i,j,k)$(Z_OLD(i,j,k)=1), HEN(i,j,k) = yes);
```

//dynamic set for existing Utility exchanger// <---- To fix topology

```
//hot utility
```

```

HU(j) = no;
PARAMETER ZHU_OLD(j) Existing HU from original HEN of each period
/I3 1.000/;
loop(j$(ZHU_OLD(j)=1), HU(j) = yes);
HUNEW_HU('J1') = yes;
HUNEW_HU('J2') = yes;
HUNEW_CU('J4') = yes;
//cold utility
CU(i) = no;
PARAMETER ZCU_OLD(i) Existing CU from original HEN of each period
/I1 1.000, I2 1.000, I3 1.000/;
loop(i$(ZCU_OLD(i)=1), CU(i) = yes);

//define dynamic set for existing HEN but only 2 dimensions (for hot and cold)//
HEN_HOT(i,k) = no;
HEN_COLD(j,k) = no;
loop((i,j,k)$(HEN(i,j,k)), HEN_HOT(i,k) = yes);
loop((i,j,k)$(HEN(i,j,k)), HEN_COLD(j,k) = yes);

```

SCALARS

```

//Cost//
CA      per unit cost of heat exchanger area($ per unit)    /641.7/
B       exponent for area cost(dimensionless)              /1/
CF      fixed cost of heat exchanger($ per unit)           /8333.3/

CHU     per unit cost of hot utility($ per kW)             /115.2/
CCU     fixed cost of heat exchanger($ per kW)             /1.3/

AF      annualisation factor                                /0.2/

//other//
EMAT    exchanger minimum approach temperature(C)         /5/

HHU     heat transfer coefficient of hot utility(kW|(m2.C)) /2/
HCU     heat transfer coefficient of cold utility(kW|(m2.C)) /1/

TAU     tau /500/
GAMMA   gamma /1e6/
;

```

PARAMETERS

```

//Hot Streams//
THIN(i) inlet temperature of hot streams(C)
/ I1 393

```

```

        I2  160
        I3  354
    /
    THOUT(i) outlet temperature of hot streams(C)
    /
        I1  60
        I2  40
        I3  60
    /
//Cold Streams//
    TCIN(j) inlet temperature of cold streams(C)
    /
        J1  72
        J2  62
        J3  220
        J4  253
    /
    TCOUT(j) outlet temperature of cold streams(C)
    /
        J1  356
        J2  210
        J3  370
        J4  284
    /
//Heat Capacity Flowrates//
    FCPH(i) heat capacity flowrates fo hot streams(C)
    /
        I1  201.6
        I2  185.1
        I3  137.4
    /
    FCPC(j) heat capacity flowrates fo cold streams(C)
    /
        J1  209.4
        J2  141.6
        J3  176.4
        J4  294.4
    /
//Utility//
    THUIN(j) inlet temperature of hot utility(C)
    /
        J1  400
        J2  400
        J3  400
        J4  15
    /
    THUOUT(j) outlet temperature of hot utility(C)
    /
        J1  399
        J2  399
        J3  399
        J4  20
    /
    TCUIN(i) inlet temperature of cold utility(C)

```

```

/      I1  15
      I2  15
      I3  15
/
TCUOUT(i) outlet temperature of cold utility(C)
/      I1  20
      I2  20
      I3  20
/
//Heat transfer coefficient//
HH(i)  heat capaity flowrates of hot streams(kW|(m2.C))
/      I1  2
      I2  2
      I3  2
/
HC(j)  heat capaity flowrates of cold streams(kW|(m2.C))
/      J1  1.5
      J2  1.5
      J3  2
      J4  2
/
//overall heat transfer coefficient//
U(i,j)  Overall heat transfer coefficient for match(i,j)
UHU(j)  Overall heat transfer coefficient for cold stream(j) and hot utility
UCU(i)  Overall heat transfer coefficient for hot stream(i) and cold utility

//upper bound for logical constraints//
GAMMA_HX(i,j,k)  upper bound heat load for match (i,j)
GAMMA_HU(j)      upper bound heat load for hot utility
GAMMA_CU(i)      upper bound heat load for cold utility

TAU_HX(i,j,k)   upper bound of temperature difference for heat exchanger
TAU_HUL(j)      upper bound of temperature difference for left side of hot
utility exchanger
TAU_HUR(j)      upper bound of temperature difference for right side of hot
utility exchanger
TAU_CUL(i)      upper bound of temperature difference for left side pf cold
utility exchanger
TAU_CUR(i)      upper bound of temperature difference for right side of cold
utility exchanger
;

//Old value of variables from original HEN//

```


//Area//

TABLE AREA_OLD(i,j,k) old area from original HEN of each period

	K1	K2	K3	K4
--	----	----	----	----

I1.J1	680.055	166.126	751.970	
I1.J2		1290.040	132.794	
I1.J3		597.190		
I3.J1		356.123	1470.846	
I3.J4	324.859			

;

PARAMETERS

AREAHU_OLD(j) old area of HU from original HEN

/J3 256.025/

AREACU_OLD(i) old area of CU from original HEN

/I1 117.036, I2 541.396, I3 97.762/

;

//Temperature//

TABLE TH_OLD(i,k) old hot temperature from original HEN

	K1	K2	K3	K4	KL
--	----	----	----	----	----

I1	406.000	293.498	231.178	88.555	79.812
I2	160.000	160.000	160.000	160.000	160.000
I3	362.000	268.540	234.038	86.252	86.252

;

TABLE TC_OLD(j,k) old cold temperature from original HEN

	K1	K2	K3	K4	KL
--	----	----	----	----	----

J1	365.000	255.334	216.175	72.000	72.000
J2	210.000	210.000	210.000	74.710	62.000
J3	272.717	272.717	220.000	220.000	220.000
J4	290.000	250.000	250.000	250.000	250.000

;

//Heat load//

TABLE Q_OLD(i,j,k) old heat load from original HEN

	K1	K2	K3	K4
--	----	----	----	----

I1.J1	23062.852	3528.959	10162.028	
I1.J2		19075.844	1792.156	
I1.J3		9246.623		
I3.J1		4706.050	20158.011	
I3.J4	12748.000			

;

PARAMETERS

QHU_OLD(j) old heat load of HU from original HEN

/J3 17063.377/

QCU_OLD(i) old heat load of CU from original HEN

/I1 4061.538, I2 23856.000, I3 3580.739/

;

//Temperature Difference//

TABLE DT_OLD(i,j,k) old temperature differene from original HEN

K1 K2 K3 K4 KL

I1.J1	41.000	38.165	15.003	16.555	
I1.J2			21.178	13.844	17.812
I1.J3		20.781	11.178		
I3.J1		13.206	17.863	14.252	
I3.J4	72.000	18.540			

;

PARAMETERS

DTHUL_OLD(j) old left-side temperature differene of HU from original HEN

/J3 30.000/

DTHUR_OLD(j) old right-side temperature differene of HU from original HEN

/J3 126.283/

DTCUL_OLD(i) old left-side temperature differene of CU from original HEN

/I1 59.812, I2 140.000, I3 66.252/

DTCUR_OLD(i) old right-side temperature differene of CU from original HEN

/I1 45.000, I2 25.000, I3 45.000/

;

//Log Mean Temperature Difference//

TABLE LMTD_OLD(i,j,k) old log mean temperature from original HEN

K1 K2 K3 K4

I1.J1	39.565	24.783	15.766	
I1.J2			17.252	15.745
I1.J3		15.484		
I3.J1		15.417	15.989	
I3.J4	39.242			

;

PARAMETERS

LMTDHU_OLD(j) old log mean temperature of HU from original HEN

/J3 66.647/

LMTDCU_OLD(i) old log mean temperature of CU from original HEN

/I1 52.055, I2 66.096, I3 54.941/

;

//Overall heat transfer coefficient calculations//

$$U(i,j) = 1/(1/HH(i)+1/HC(j));$$

$$UHU(j)\$(HU(j) \text{ or } HUNEW_HU(j)) = 1/(1/HC(j)+1/HHU);$$

$$UHU(j)\$HUNEW_CU(j) = 1/(1/HC(j)+1/HCU); \quad //\text{suppose it is cold utility}$$

$$UCU(i) = 1/(1/HH(i)+1/HCU);$$

VARIABLES

th(i,k) temperature(C) of hot streams i at location k
tc(j,k) temperature(C) of cold streams j at location k

q(i,j,k) heat load(kW) of match(i,j) at stage k
qhu(j) heat load(kW) of hot utility for cold stream j
qcu(i) heat load(kW) of cold utility for hot stream i

zhu(j) addition of hot utility
zcu(i) addition of cold utility

dt(i,j,k) temperature difference(C) for match(i,j) at location k
dthul(j) temperature difference(C) for left side of hot utility exchanger
dthur(j) temperature difference(C) for right side of hot utility exchanger
dtcul(i) temperature difference(C) for left side of cold utility exchanger
dteur(i) temperature difference(C) for right side of cold utility exchanger

lmt(i,j,k) log mean temperature difference of match(i,j) at stage k
lmtdhu(j) log mean temperature difference of hot utility exchanger for cold stream j
lmtdcu(i) log mean temperature difference of cold utility exchanger for cold stream i

area(i,j,k) area of heat exchanger of match(i,j) at stage k
areahu(j) area of hot utility exchanger of cold stream j
areacu(i) area of cold utility exchanger of hot stream i

areahu_add(j) additional area of hot utility exchanger of cold stream j
areacu_add(i) additional area of cold utility exchanger of hot stream i

Cost_fix_add additional fixed cost from new utility
Cost_area_add area cost
Cost_utility utility cost

tot_area_add total additional cost
;

POSITIVE VARIABLES q(i,j,k), qcu(i), // qhu(j)
dt(i,j,k), dthul(j), dthur(j), dtcul(i), dteur(i),
lmt(i,j,k), lmtdhu(j), lmtdcu(i),
areahu(j), areacu(i),
areahu_add(j), areacu_add(i);

BINARY VARIABLES zhu(j), zcu(i);

//variable bounding//

th.up(i,k) = THIN(i);
th.lo(i,k) = THOUT(i);
tc.up(j,k)\$(HU(j) or HUNEW_HU(j)) = TCOUT(j); //bound only exist HU
tc.lo(j,k) = TCIN(j);

q.up(i,j,k)\$stage(k) = max(min(FCPH(i)*(th.up(i,k)-
max(th.lo(i,k+1),tc.lo(j,k+1)+EMAT)) , FCPC(j)*(min(th.up(i,k)-
EMAT,tc.up(j,k))-tc.lo(j,k+1))) , 0)\$((th.up(i,k)-tc.lo(j,k) >= EMAT) and
(th.up(i,k+1)-tc.lo(j,k+1) >= EMAT)) +0\$((th.up(i,k)-tc.lo(j,k) < EMAT) or
(th.up(i,k+1)-tc.lo(j,k+1) < EMAT)) ;

//only for HU(j) that exist hot utility

qhu.up(j)\$(HU(j) or HUNEW_HU(j)) =
max(FCPC(j)*(min(THUIN(j)-EMAT,TCOUT(j))-tc.lo(j,'K1')) , 0)\$((THUIN(j)-
TCOUT(j) >= EMAT) and (THUOUT(j)-tc.lo(j,'K1') >= EMAT)) +0\$((THUIN(j)-
TCOUT(j) < EMAT) or (THUOUT(j)-tc.lo(j,'K1') < EMAT)) ;

qhu.lo(j)\$(HU(j) or HUNEW_HU(j)) = 0;
qcu.up(i) = max(FCPH(i)*(th.up(i,'KL')-
max(THOUT(i),TCUIN(i)+EMAT)) , 0)\$((th.up(i,'KL')-TCUOUT(i) >= EMAT)
and (THOUT(i)-TCUIN(i) >= EMAT)) +0\$((th.up(i,'KL')-TCUOUT(i) < EMAT) or
(THOUT(i)-TCUIN(i) < EMAT)) ;

qcu.lo(i) = 0;

dt.up(i,j,k) = max(EMAT,(th.up(i,k)-tc.lo(j,k)));

dt.lo(i,j,k) = EMAT;

dthul.up(j)\$(HU(j) or HUNEW_HU(j)) = max(EMAT,THUIN(j)-TCOUT(j));

//bound only exist HU

dthul.lo(j) = EMAT;

dthur.up(j)\$(HU(j) or HUNEW_HU(j)) = max(EMAT,THUOUT(j)-
tc.lo(j,'K1')); //bound only exist HU

dthur.lo(j) = EMAT;

dteul.up(i) = max(EMAT,th.up(i,'KL')-TCUOUT(i));

dteul.lo(i) = EMAT;

dteur.up(i) = max(EMAT,THOUT(i)-TCUIN(i));

dteur.lo(i) = EMAT;

lmtd.up(i,j,k)\$stage(k) =

(dt.up(i,j,k)*dt.up(i,j,k+1)*(dt.up(i,j,k)+dt.up(i,j,k+1))/2)**(1/3);

lmtd.lo(i,j,k)\$stage(k) =

(dt.lo(i,j,k)*dt.lo(i,j,k+1)*(dt.lo(i,j,k)+dt.lo(i,j,k+1))/2)**(1/3);

```

lmtdhu.up(j)$ (HU(j) or HUNEW_HU(j))      =
(dthur.up(j)*dthul.up(j)*(dthur.up(j)+dthul.up(j))/2)**(1/3); //bound only exist HU
lmtdhu.lo(j)      = (dthur.lo(j)*dthul.lo(j)*(dthur.lo(j)+dthul.lo(j))/2)**(1/3);
lmtdhu.up(i)      = (dthul.up(i)*dthur.up(i)*(dthul.up(i)+dthur.up(i))/2)**(1/3);
lmtdhu.lo(i)      = (dthul.lo(i)*dthur.lo(i)*(dthul.lo(i)+dthur.lo(i))/2)**(1/3);

```

```

area.up(i,j,k)$stage(k) = q.up(i,j,k)/lmtd.lo(i,j,k)/U(i,j);
areahu.up(j)$ (HU(j) or HUNEW_HU(j)) = qhu.up(j)/lmtdhu.lo(j)/UHU(j);
//bound only exist HU
areacu.up(i)      = qcu.up(i)/lmtdhu.lo(i)/UCU(i);

```

```
//Assign bounding to pamaters in logical constraint//
```

```

GAMMA_HX(i,j,k) = q.up(i,j,k);
GAMMA_HU(j)$ (HU(j) or HUNEW_HU(j)) = qhu.up(j);
GAMMA_HU(j)$HUNEW_CU(j) = GAMMA;
GAMMA_CU(i) = qcu.up(i);

```

```

TAU_HX(i,j,k) = -th.lo(i,k)+tc.up(j,k)+dt.lo(i,j,k);
TAU_HUL(j)$ (HU(j) or HUNEW_HU(j)) = -THUIN(j)+TCOUT(j)
+dthul.lo(j); //bound only exist HU
TAU_HUL(j)$HUNEW_CU(j) = TAU; //Arbitrary create a value
TAU_HUR(j)$ (HU(j) or HUNEW_HU(j)) = -THUOUT(j)+tc.up(j,'K1')
+dthur.lo(j); //bound only exist HU
TAU_HUR(j)$HUNEW_CU(j) = TAU; //Arbitrary create a value
TAU_CUL(i) = -th.lo(i,'KL')+TCUOUT(i)+dthul.lo(i);
TAU_CUR(i) = -THOUT(i)+TCUIN(i)+dthur.lo(i);

```

```
//Last stage forcing bound//
```

```

q.fx(i,j,'KL') = 0;
lmtd.fx(i,j,'KL') = 0;
area.fx(i,j,'KL') = 0;

```

```
//To force no q, area change where there is no existing HX//
```

```

q.fx(i,j,k)$ (not HEN(i,j,k)) = 0;
area.fx(i,j,k)$ (not HEN(i,j,k)) = 0;

```

```
//Fix area in the process-process HX//
```

```

area.fx(i,j,k)$ (HEN(i,j,k)) = AREA_OLD(i,j,k); //and(not HEN('I3','J3','K1'))

```

```
//initial value for variables//
```

```

areahu.l(j) = AREA_HU_OLD(j);
areacu.l(i) = AREA_CU_OLD(i);

```

```

areahu_add.l(j) = 0;
areacu_add.l(i) = 0;

```

$$\begin{aligned}
 th.l(i,k) &= TH_OLD(i,k); \\
 tc.l(j,k) &= TC_OLD(j,k); \\
 \\
 q.l(i,j,k) &= Q_OLD(i,j,k); \\
 qhu.l(j) &= QHU_OLD(j); \\
 qcu.l(i) &= QCU_OLD(i); \\
 \\
 dt.l(i,j,k) &= DT_OLD(i,j,k); \\
 dthul.l(j) &= DTHUL_OLD(j); \\
 dthur.l(j) &= DTHUR_OLD(j); \\
 dtcul.l(i) &= DTCUL_OLD(i); \\
 dtcur.l(i) &= DTCUR_OLD(i); \\
 \\
 lmtl.l(i,j,k) &= LMTD_OLD(i,j,k); \\
 lmtdhu.l(j) &= LMTDHU_OLD(j); \\
 lmtdcl.l(i) &= LMTDCU_OLD(i); \\
 \\
 zhu.l(j)\$(not HU(j)) &= 0; \\
 zcu.l(i)\$(not CU(i)) &= 0;
 \end{aligned}$$

EQUATIONS

OVERALL_H(i)	overall energy balance of hot stream i
OVERALL_C(j)	overall energy balance of cold stream j
EBAL_H(i,k)	energy balance of hot stream i in stage k
EBAL_C(j,k)	energy balance of cold stream j in stage k
TIN_H(i)	assignment inlet temperature of hot stream i
TIN_C(j)	assignment inlet temperature of cold stream j
MONOT_H(i,k)	monotonic in temperature of hot stream i at location k
MONOT_C(j,k)	monotonic in temperature of cold stream j at location k
MONOT_HNOT(i,k)	monotonic in temperature of hot stream i at location k if no HX
MONOT_CNOT(j,k)	monotonic in temperature of cold stream j at location k if no HX
MONOT_HU(j)	monotonic in outlet temperature of cold stream j
MONOT_CU(i)	monotonic in outlet temperature of hot stream i
EBAL_HU(j)	energy balance of hot utility of cold stream j
EBAL_CU(i)	energy balance of cold utility of hot stream i
EBAL_HUNEW_CU(j)	energy balance of hot utility of cold stream j
EBAL_CUNOT(i)	energy balance of cold utility of hot stream i
LOGIC_HUNEW_HU(j)	logical constraint to define zhu of cold stream j at stage k (hot utility)

LOGIC_HUNEW_CU(j)	logical constraint to define zhu of cold stream j at stage k (cold utility)
LOGIC_CU(i)	logical constraint to define zcu of hot stream i at stage k
DTFEAS_HXL(i,j,k)	temperature feasibility of heat exchanger of match(i,j) at location k
DTFEAS_HXR(i,j,k)	temperature feasibility of heat exchanger of match(i,j) at location k+1
DTFEAS_HUL(j)	temperature feasibility of hot utility of outlet cold stream j
DTFEAS_HUR(j)	temperature feasibility of hot utility of inlet cold stream j
DTFEAS_CUL(i)	temperature feasibility of cold utility of inlet hot stream i
DTFEAS_CUR(i)	temperature feasibility of cold utility of outlet hot stream i
DTFEAS_HUL_HUNEW_HU(j)	temperature feasibility of new hot utility of outlet cold stream j
DTFEAS_HUR_HUNEW_HU(j)	temperature feasibility of new hot utility of inlet cold stream j
DTFEAS_HUL_HUNEW_CU(j)	temperature feasibility of new cold utility of outlet cold stream j
DTFEAS_HUR_HUNEW_CU(j)	temperature feasibility of new cold utility of inlet cold stream j
DTFEAS_CULNOT(i)	temperature feasibility of cold utility of inlet hot stream i
DTFEAS_CURNOT(i)	temperature feasibility of cold utility of outlet hot stream i
LOGMTD(i,j,k)	log mean temperature of match(i,j) at stage k
LOGMTDHU(j)	log mean temperature of hot utility exchanger
LOGMTDCU(i)	log mean temperature of cold utility exchanger
ADDIAREA_HU1(j)	additional area needed for hot utility
ADDIAREA_HU2(j)	force ADDAREA_HU > 0
ADDIAREA_CU1(i)	additional area needed for cold utility
ADDIAREA_CU2(i)	force ADDAREA_CU > 0
A_HX(i,j,k)	area of heat exchanger of match(i,j) at stage k
A_HU(j)	area of hot utility exchanger of cold stream j
A_HUNOT(j)	area for cold utility for J1 J2
A_CU(i)	area of cold utility exchanger of hot stream i
OBJ	objective function (total additional area)
;	
OVERALL_H(i)	.. (THIN(i) - THOUT(i))*FCPH(i) =e= sum((j,k),q(i,j,k))+qcu(i);

OVERALL_C(j) .. (TCOUT(j)-TCIN(j)) *FCPC(j) =e= sum((i,k),q(i,j,k))+qhu(j);

EBAL_H(i,k)\$HEN_HOT(i,k) .. (th(i,k)-th(i,k+1))*FCPH(i) =e= sum(j,q(i,j,k)); //do only exist HEN

EBAL_C(j,k)\$HEN_COLD(j,k) .. (tc(j,k)-tc(j,k+1))*FCPC(j) =e= um(i,q(i,j,k));

TIN_H(i) .. THIN(i) =e= th(i,'K1');

TIN_C(j) .. TCIN(j) =e= tc(j,'KL');

MONOT_H(i,k)\$HEN_HOT(i,k) .. th(i,k) =g= th(i,k+1); //if HX --> monotonic

MONOT_C(j,k)\$HEN_COLD(j,k) .. tc(j,k) =g= tc(j,k+1);

MONOT_HNOT(i,k)\$((not HEN_HOT(i,k)) and stage(k)) .. th(i,k) =e= th(i,k+1); //if no HX --> equal

MONOT_CNOT(j,k)\$((not HEN_COLD(j,k)) and stage(k)) .. tc(j,k) =e= tc(j,k+1);

MONOT_HU(j)\$ (HU(j) or HUNEW_HU(j)) .. TCOUT(j) =g= tc(j,'K1'); //if HX --> monotonic

MONOT_CU(i)\$CU(i) .. th(i,'KL') =g= THOUT(i);

EBAL_HU(j)\$ (HU(j) or HUNEW_HU(j)) .. (TCOUT(j) - tc(j,'K1'))*FCPC(j) =e= qhu(j); //do only exist HEN

EBAL_CU(i)\$CU(i) .. (th(i,'KL') - THOUT(i)) *FCPH(i) =e= qcu(i);

EBAL_HUNEW_CU(j)\$HUNEW_CU(j) .. (TCOUT(j) - tc(j,'K1'))*FCPC(j) =e= qhu(j);

EBAL_CUNOT(i)\$ (not CU(i)) .. (th(i,'KL') - THOUT(i)) *FCPH(i) =e= qcu(i);

LOGIC_HUNEW_HU(j)\$HUNEW_HU(j) .. qhu(j) - GAMMA_HU(j) *zhu(j) =l= 0;

LOGIC_HUNEW_CU(j)\$HUNEW_CU(j) .. qhu(j) + GAMMA_HU(j) *zhu(j) =g= 0; //change - to + because qhu = -

LOGIC_CU(i)\$ (not CU(i)) .. qcu(i) - GAMMA_CU(i) *zcu(i) =l= 0;

DTFEAS_HXL(i,j,k)\$HEN(i,j,k) .. dt(i,j,k) =e= th(i,k) -tc(j,k) ; //if exist HEN --> equal

DTFEAS_HXR(i,j,k)\$HEN(i,j,k) .. dt(i,j,k+1) =e= th(i,k+1) -tc(j,k+1);

DTFEAS_HUL(j)\$HU(j) .. dthul(j) =e= THUIN(j) -TCOUT(j) ;

DTFEAS_HUR(j)\$HU(j) .. dthur(j) =e= THUOUT(j) -tc(j,'K1'); //if exist utility --> equal

DTFEAS_CUL(i)\$CU(i) .. dtcul(i) =e= th(i,'KL')-TCUOUT(i) ;

DTFEAS_CUR(i)\$CU(i) .. dtcur(i) =e= THOUT(i) -TCUIN(i) ;

DTFEAS_HUL_HUNEW_HU(j)\$HUNEW_HU(j) .. dthul(j) =l= THUIN(j) - TCOUT(j) +TAU_HUL(j) *(1-zhu(j)); //if no utility --> use TAU and binary

DTFEAS_HUR_HUNEW_HU(j)\$HUNEW_HU(j) .. dthur(j) =l= THUOUT(j) -tc(j,'K1')+TAU_HUR(j) *(1-zhu(j));


```

DTFEAS_HUL_HUNEW_CU(j)$HUNEW_CU(j) .. dthul(j) =|= TCOUT(j) -
THUIN(j) +TAU_HUL(j) *(1-zhu(j)); //if no utility --> use TAU and binary
DTFEAS_HUR_HUNEW_CU(j)$HUNEW_CU(j) .. dthur(j) =|= tc(j,'K1')
-THUOUT(j) +TAU_HUR(j) *(1-zhu(j));
DTFEAS_CULNOT(i)$ (not CU(i)) .. dtcul(i) =|= th(i,'KL')-TCUOUT(i)
+TAU_CUL(i) *(1-zcu(i));
DTFEAS_CURNOT(i)$ (not CU(i)) .. dtcur(i) =|= THOUT(i) -TCUIN(i)
+TAU_CUR(i) *(1-zcu(i));

LOGMTD(i,j,k)$HEN(i,j,k) .. lmtd(i,j,k) =e=
(dt(i,j,k)*dt(i,j,k+1)*(dt(i,j,k)+dt(i,j,k+1))/2)**(1/3); //only HEN
LOGMTDHU(j) .. lmtdhu(j) =e= (dthur(j) *dthul(j) *(dthur(j)
+dthul(j)) /2)**(1/3);
LOGMTDCU(i) .. lmtdcu(i) =e= (dcul(i) *dcur(i) *(dcul(i)
+dcur(i)) /2)**(1/3);

A_HX(i,j,k)$HEN(i,j,k) .. area(i,j,k) *lmtd(i,j,k)*U(i,j) =e= q(i,j,k); //only HEN
A_HU(j)$ (HU(j) or HUNEW_HU(j)) .. areahu(j) *lmtdhu(j) *UHU(j) =e= qhu(j);
A_HUNOT(j)$HUNEW_CU(j) .. areahu(j) *lmtdhu(j) *UHU(j) =e= -qhu(j);
//qhu = -
A_CU(i) .. areacu(i) *lmtdcu(i) *UCU(i) =e= qcu(i);

ADDIAREA_HU1(j) .. areahu_add(j) =g= areahu(j) - AREA_HU_OLD(j);
ADDIAREA_HU2(j) .. areahu_add(j) =g= 0;
ADDIAREA_CU1(i) .. areacu_add(i) =g= areacu(i) - AREA_CU_OLD(i);
ADDIAREA_CU2(i) .. areacu_add(i) =g= 0;

//Only area penalty
OBJ .. TOT_AREA_ADD =e= sum(j,areahu_add(j)) +
sum(i,areacu_add(i));

model STAGEMODEL_SINGLEPERIOD /all/;

option iterlim=1e9;
option domlim=0;
option reslim=1e8;
STAGEMODEL_SINGLEPERIOD.optfile=0;

solve STAGEMODEL_SINGLEPERIOD using MINLP minimizing
TOT_AREA_ADD;

```

```
display th.l,tc.l,  
q.l,qhu.l,qcu.l,  
zhu.l,zcu.l,  
area.l,areahu.l,areacu.l,  
dt.l,dthul.l,dthur.l,dtcul.l,dtcur.l,  
lmtl.l,lmtdhu.l,lmtdcu.l,  
tot_area_add.l,  
area.lo,area.up,areahu.lo,areahu.up,areacu.lo,areacu.up,  
q.up,q.lo,qhu.up,qhu.lo,qcu.up,qcu.lo  
dt.up,dt.lo,dthul.lo,dthul.up,dthur.lo,dthur.up,dtcul.lo,dtcul.up,  
dtcur.lo,dtcur.up  
;
```

B2 Model B

SETS

I hot streams /I1*I3/
 J cold streams /J1*J4/
 K index of stage or location /K1*K4,KL/ //last number is no. of stages

stage(k) all stages
 HEN(i,j,k) define where HX exists
 HEN_HOT(i,k) define where HX exists 2 dimensions for hot streams
 HEN_COLD(j,k) d efine where HX exists 2 dimensions for cold streams
 HU(j) define where HU exists
 CU(i) define where CU exists
 ;

//dynamic set for stage pointer

stage(k) = yes;
 stage('KL') = no;

//dynamic set for existing HEN// <---- To fix topology

HEN(i,j,k) = no;

TABLE Z_OLD(i,j,k) Existing HX from original HEN of each period

	K1	K2	K3	K4
I1.J1	1.000	1.000	1.000	
I1.J2			1.000	1.000
I1.J3		1.000		
I3.J1		1.000	1.000	
I3.J4	1.000			

;

loop((i,j,k)\$(Z_OLD(i,j,k)=1), HEN(i,j,k) = yes);

//dynamic set for existing Utility exchanger// <---- To fix topology

//hot utility

HU(j) = no;
 PARAMETER ZHU_OLD(j) Existing HU from original HEN of each period
 /J3 1.000/;
 loop(j\$(ZHU_OLD(j)=1), HU(j) = yes);

//cold utility

CU(i) = no;

```

PARAMETER ZCU_OLD(i) Existing CU from original HEN of each period
/I1 1.000, I2 1.000, I3 1.000/;
loop(i$(ZCU_OLD(i)=1), CU(i) = yes);

```

```
//define dynamic set for existing HEN but only 2 dimensions (for hot and cold)//
```

```

HEN_HOT(i,k) = no;
HEN_COLD(j,k) = no;
loop((i,j,k)$(HEN(i,j,k)), HEN_HOT(i,k) = yes);
loop((i,j,k)$(HEN(i,j,k)), HEN_COLD(j,k) = yes);

```

SCALARS

```
//Cost//
```

CA	per unit cost of heat exchanger area(\$ per unit)	/641.7/
B	exponent for area cost(dimensionless)	/1/
CF	fixed cost of heat exchanger(\$ per unit)	/8333.3/
CHU	per unit cost of hot utility(\$ per kW)	/115.2/
CCU	per unit cost of cold utility(\$ per kW)	/1.3/
AF	annualisation factor	/0.2/

```
//other//
```

EMAT	exchanger minimum approach temperature(C)	/5/
HHU	heat transfer coefficient of hot utility(kW (m ² .C))	/2/
HCU	heat transfer coefficient of cold utility(kW (m ² .C))	/1/

;

PARAMETERS

```
//Hot Streams//
```

THIN(i)	inlet temperature of hot streams(C)
/ I1	393
I2	160
I3	354
/	
THOUT(i)	outlet temperature of hot streams(C)
/ I1	60
I2	40
I3	60
/	

```
//Cold Streams//
```

TCIN(j)	inlet temperature of cold streams(C)
/ J1	72

```

        J2  62
        J3  220
        J4  253
    /
    TCOUT(j) outlet temperature of cold streams(C)
    /   J1  356
        J2  210
        J3  370
        J4  284
    /
//Heat Capacity Flowrates//
    FCPH(i) heat capaity flowrates fo hot streams(C)
    /   I1  201.6
        I2  185.1
        I3  137.4
    /
    FCPC(j) heat capaity flowrates fo cold streams(C)
    /   J1  209.4
        J2  141.6
        J3  176.4
        J4  294.4
    /
//Utility//
    THUIN(j) inlet temperature of hot utility(C)
    /   J1  400
        J2  400
        J3  400
        J4  400
    /
    THUOUT(j) outlet temperature of hot utility(C)
    /   J1  399
        J2  399
        J3  399
        J4  399
    /
    TCUIN(i) inlet temperature of cold utility(C)
    /   I1  15
        I2  15
        I3  15
    /
    TCUOUT(i) outlet temperature of cold utility(C)
    /   I1  20
        I2  20
        I3  20
    /
//Heat transfer coefficient//
    HH(i) heat capaity flowrates of hot streams(kW|(m2.C))

```

```

/      I1  2
      I2  2
      I3  2
/
HC(j) heat capacity flowrates of cold streams(kW|(m2.C))
/      J1  1.5
      J2  1.5
      J3  2
      J4  2
/
//overall heat transfer coefficient//
U(i,j)      Overall heat transfer coefficient for match(i,j)
UHU(j)      Overall heat transfer coefficient for cold stream(j) and hot utility
UCU(i)      Overall heat transfer coefficient for hot stream(i) and cold utility

//upper bound for logical constraints//
GAMMA_HX(i,j,k)  upper bound heat load for match (i,j)
GAMMA_HU(j)      upper bound heat load for hot utility
GAMMA_CU(i)      upper bound heat load for cold utility

TAU_HX(i,j,k)   upper bound of temperature difference for heat exchanger
TAU_HUL(j)      upper bound of temperature difference for left side of hot
utility exchanger
TAU_HUR(j)      upper bound of temperature difference for right side of hot
utility exchanger
TAU_CUL(i)      upper bound of temperature difference for left side pf cold
utility exchanger
TAU_CUR(i)      upper bound of temperature difference for right side of cold
utility exchanger
;

//Old value of variables from original HEN//

//Area//
TABLE AREA_OLD(i,j,k) old area from original HEN of each period
      K1      K2      K3      K4

I1.J1  680.055  166.126  751.970
I1.J2           1290.040  132.794
I1.J3           597.190
I3.J1           356.123  1470.846
I3.J4  324.859
;

PARAMETERS
AREAHU_OLD(j) old area of HU from original HEN
/J3 256.025/

```

AREACU_OLD(i) old area of CU from original HEN

/I1 117.036, I2 541.396, I3 97.762/

;

//Temperature//

TABLE TH_OLD(i,k) old hot temperature from original HEN

K1 K2 K3 K4 KL

I1	406.000	293.498	231.178	88.555	79.812
I2	160.000	160.000	160.000	160.000	160.000
I3	362.000	268.540	234.038	86.252	86.252

;

TABLE TC_OLD(j,k) old cold temperature from original HEN

K1 K2 K3 K4 KL

J1	365.000	255.334	216.175	72.000	72.000
J2	210.000	210.000	210.000	74.710	62.000
J3	272.717	272.717	220.000	220.000	220.000
J4	290.000	250.000	250.000	250.000	250.000

;

//Heat load//

TABLE Q_OLD(i,j,k) old heat load from original HEN

K1 K2 K3 K4

I1.J1	23062.852	3528.959	10162.028	
I1.J2		19075.844	1792.156	
I1.J3		9246.623		
I3.J1		4706.050	20158.011	
I3.J4	12748.000			

;

PARAMETERS

QHU_OLD(j) old heat load of HU from original HEN

/J3 17063.377/

QCU_OLD(i) old heat load of CU from original HEN

/I1 4061.538, I2 23856.000, I3 3580.739/

;

//Temperature Difference//

TABLE DT_OLD(i,j,k) old temperature difference from original HEN

K1 K2 K3 K4 KL

I1.J1	41.000	38.165	15.003	16.555
I1.J2		21.178	13.844	17.812
I1.J3		20.781	11.178	
I3.J1		13.206	17.863	14.252

I3.J4 72.000 18.540

;

PARAMETERS

DTHUL_OLD(j) old left-side temperature difference of HU from original HEN
/J3 30.000/

DTHUR_OLD(j) old right-side temperature difference of HU from original HEN
/J3 126.283/

DTCUL_OLD(i) old left-side temperature difference of CU from original HEN
/I1 59.812, I2 140.000, I3 66.252/

DTCUR_OLD(i) old right-side temperature difference of CU from original HEN
/I1 45.000, I2 25.000, I3 45.000/

;

//Log Mean Temperature Difference//

TABLE LMTD_OLD(i,j,k) old log mean temperature from original HEN

	K1	K2	K3	K4
--	----	----	----	----

I1.J1	39.565	24.783	15.766	
-------	--------	--------	--------	--

I1.J2		17.252	15.745	
-------	--	--------	--------	--

I1.J3		15.484		
-------	--	--------	--	--

I3.J1		15.417	15.989	
-------	--	--------	--------	--

I3.J4	39.242			
-------	--------	--	--	--

;

PARAMETERS

LMTDHU_OLD(j) old log mean temperature of HU from original HEN
/J3 66.647/

LMTDCU_OLD(i) old log mean temperature of CU from original HEN
/I1 52.055, I2 66.096, I3 54.941/

;

//Overall heat transfer coefficient calculations//

$U(i,j) = 1/(1/HH(i)+1/HC(j));$

$UHU(j) = 1/(1/HC(j)+1/HHU);$

$UCU(i) = 1/(1/HH(i)+1/HCU);$

VARIABLES

th(i,k) temperature(C) of hot streams i at location k

tc(j,k) temperature(C) of cold streams j at location k

q(i,j,k) heat load(kW) of match(i,j) at stage k

qhu(j) heat load(kW) of hot utility for cold stream j

qcu(i) heat load(kW) of cold utility for hot stream i

dt(i,j,k) temperature difference(C) for match(i,j) at location k

dthul(j) temperature difference(C) for left side of hot utility exchanger
 dthur(j) temperature difference(C) for right side of hot utility exchanger
 dtcul(i) temperature difference(C) for left side of cold utility exchanger
 dtcur(i) temperature difference(C) for right side of cold utility exchanger

 lmtd(i,j,k) log mean temperature difference of match(i,j) at stage k
 lmtdhu(j) log mean temperature difference of hot utility exchanger for cold stream j
 lmtdcu(i) log mean temperature difference of cold utility exchanger for cold stream i

 area(i,j,k) area of heat exchanger of match(i,j) at stage k
 areahu(j) area of hot utility exchanger of cold stream j
 areacu(i) area of cold utility exchanger of hot stream i

tot_lsqr_error total least square error
 ;

POSITIVE VARIABLES q(i,j,k), qhu(j), qcu(i),
 dt(i,j,k), dthul(j), dthur(j), dtcul(i), dtcur(i),
 lmtd(i,j,k), lmtdhu(j), lmtdcu(i),
 area(i,j,k), areahu(j), areacu(i)
 area_add(i,j,k), areahu_add(j), areacu_add(i);

//variable bounding//

th.up(i,k) = THIN(i);
 th.lo(i,k) = THOUT(i);
 tc.up(j,k) = TCOU(j);
 tc.lo(j,k) = TCIN(j);

$$q.up(i,j,k) \$stage(k) = \max(\min(FCPH(i)*(th.up(i,k) - \max(th.lo(i,k+1), tc.lo(j,k+1) + EMAT)), FCPC(j)*(min(th.up(i,k) - EMAT, tc.up(j,k)) - tc.lo(j,k+1))), 0) \$((th.up(i,k) - tc.lo(j,k) \geq EMAT) \text{ and } (th.up(i,k+1) - tc.lo(j,k+1) \geq EMAT)) + 0 \$((th.up(i,k) - tc.lo(j,k) < EMAT) \text{ or } (th.up(i,k+1) - tc.lo(j,k+1) < EMAT));$$

$$qhu.up(j) = \max(FCPC(j)*(min(THUIN(j) - EMAT, TCOU(j)) - tc.lo(j, 'K1')), 0) \$((THUIN(j) - TCOU(j) \geq EMAT) \text{ and } (THUOUT(j) - tc.lo(j, 'K1') \geq EMAT)) + 0 \$((THUIN(j) - TCOU(j) < EMAT) \text{ or } (THUOUT(j) - tc.lo(j, 'K1') < EMAT));$$

$$qcu.up(i) = \max(FCPH(i)*(th.up(i, 'KL') - \max(THOUT(i), TCUIN(i) + EMAT)), 0) \$((th.up(i, 'KL') - TCUOUT(i) \geq EMAT) \text{ and } (THOUT(i) - TCUIN(i) \geq EMAT)) + 0 \$((th.up(i, 'KL') - TCUOUT(i) < EMAT) \text{ or } (THOUT(i) - TCUIN(i) < EMAT));$$

```

dt.up(i,j,k) = max(EMAT,(th.up(i,k)-tc.lo(j,k)));
dt.lo(i,j,k) = EMAT;
dthul.up(j) = max(EMAT,THUIN(j)-TCOUT(j));
dthul.lo(j) = EMAT;
dthur.up(j) = max(EMAT,THUOUT(j)-tc.lo(j,'K1'));
dthur.lo(j) = EMAT;
dteul.up(i) = max(EMAT,th.up(i,'KL')-TCUOUT(i));
dteul.lo(i) = EMAT;
dteur.up(i) = max(EMAT,THOUT(i)-TCUIN(i));
dteur.lo(i) = EMAT;

lmtd.up(i,j,k)$stage(k) =
    (dt.up(i,j,k)*dt.up(i,j,k+1)*(dt.up(i,j,k)+dt.up(i,j,k+1))/2)**(1/3);
lmtd.lo(i,j,k)$stage(k) =
    (dt.lo(i,j,k)*dt.lo(i,j,k+1)*(dt.lo(i,j,k)+dt.lo(i,j,k+1))/2)**(1/3);
lmtdhu.up(j) = (dthur.up(j)*dthul.up(j)*(dthur.up(j)+dthul.up(j))/2)**(1/3);
lmtdhu.lo(j) = (dthur.lo(j)*dthul.lo(j)*(dthur.lo(j)+dthul.lo(j))/2)**(1/3);
lmtdteu.up(i) = (dteul.up(i)*dteur.up(i)*(dteul.up(i)+dteur.up(i))/2)**(1/3);
lmtdteu.lo(i) = (dteul.lo(i)*dteur.lo(i)*(dteul.lo(i)+dteur.lo(i))/2)**(1/3);

area.up(i,j,k)$stage(k) = q.up(i,j,k)/lmtd.lo(i,j,k)/U(i,j);
areahu.up(j) = qhu.up(j)/lmtdhu.lo(j)/UHU(j);
areacu.up(i) = qcu.up(i)/lmtdteu.lo(i)/UCU(i);

//Assign bounding to pamaters in logical constraint//
GAMMA_HX(i,j,k) = q.up(i,j,k);
GAMMA_HU(j) = qhu.up(j);
GAMMA_CU(i) = qcu.up(i);

TAU_HX(i,j,k) = -th.lo(i,k)+tc.up(j,k)+dt.lo(i,j,k);
TAU_HUL(j) = -THUIN(j)+TCOUT(j)+dthul.lo(j);
TAU_HUR(j) = -THUOUT(j)+tc.up(j,'K1')+dthur.lo(j);
TAU_CUL(i) = -th.lo(i,'KL')+TCUOUT(i)+dteul.lo(i);
TAU_CUR(i) = -THOUT(i)+TCUIN(i)+dteur.lo(i);

//Last stage forcing bound//
q.fx(i,j,'KL') = 0;
lmtd.fx(i,j,'KL') = 0;
area.fx(i,j,'KL') = 0;
area_add.fx(i,j,'KL') = 0;

//To force no q, area change where there is no existing HX//
q.fx(i,j,k)$not HEN(i,j,k) = 0;
area_add.fx(i,j,k)$not HEN(i,j,k) = 0;

```

//initial value for variables//

area.l(i,j,k) = AREA_OLD(i,j,k);
 areahu.l(j) = AREAHU_OLD(j);
 areacu.l(i) = AREACU_OLD(i);

th.l(i,k) = TH_OLD(i,k);
 tc.l(j,k) = TC_OLD(j,k);

q.l(i,j,k) = Q_OLD(i,j,k);
 qhu.l(j) = QHU_OLD(j);
 qcu.l(i) = QCU_OLD(i);

dt.l(i,j,k) = DT_OLD(i,j,k);
 dthul.l(j) = DTHUL_OLD(j);
 dthur.l(j) = DTHUR_OLD(j);
 dtcul.l(i) = DTCUL_OLD(i);
 dtcur.l(i) = DTCUR_OLD(i);

lmtl.l(i,j,k) = LMTD_OLD(i,j,k);
 lmtdhu.l(j) = LMTDHU_OLD(j);
 lmtdcu.l(i) = LMTDCU_OLD(i);

EQUATIONS

OVERALL_H(i) overall energy balance of hot stream i
 OVERALL_C(j) overall energy balance of cold stream j

EBAL_H(i,k) energy balance of hot stream i in stage k
 EBAL_C(j,k) energy balance of cold stream j in stage k

TIN_H(i) assignment inlet temperature of hot stream i
 TIN_C(j) assignment inlet temperature of cold stream j

MONOT_H(i,k) monotonic in temperature of hot stream i at location k
 MONOT_C(j,k) monotonic in temperature of cold stream j at location k
 MONOT_HNOT(i,k) monotonic in temperature of hot stream i at location k
 if no HX
 MONOT_CNOT(j,k) monotonic in temperature of cold stream j at location
 k if no HX

MONOT_HU(j) monotonic in outlet temperature of cold stream j
 MONOT_CU(i) monotonic in outlet temperature of hot stream i
 MONOT_HUNOT(j) monotonic in outlet temperature of cold stream j if no HX
 MONOT_CUNOT(i) monotonic in outlet temperature of hot stream i if no HX

EBAL_HU(j) energy balance of hot utility of cold stream j
 EBAL_CU(i) energy balance of cold utility of hot stream i

DTFEAS_HXL(i,j,k) temperature feasibility of heat exchanger of match(i,j)
 at location k
 DTFEAS_HXR(i,j,k) temperature feasibility of heat exchanger of match(i,j)
 at location k+1
 DTFEAS_HUL(j) temperature feasibility of hot utility of outlet cold
 stream j
 DTFEAS_HUR(j) temperature feasibility of hot utility of inlet cold stream j
 DTFEAS_CUL(i) temperature feasibility of cold utility of inlet hot stream i
 DTFEAS_CUR(i) temperature feasibility of cold utility of outlet hot stream i

LOGMTD(i,j,k) log mean temperature of match(i,j) at stage k
 LOGMTDHU(j) log mean temperature of hot utility exchanger
 LOGMTDCU(i) log mean temperature of cold utility exchanger

A_HX(i,j,k) area of heat exchanger of match(i,j) at stage k
 A_HU(j) area of hot utility exchanger of cold stream j
 A_CU(i) area of cold utility exchanger of hot stream i

OBJ objective function (total additional area)
 ;

OVERALL_H(i) .. (THIN(i) -THOUT(i))*FCPH(i) =e=
 sum((j,k),q(i,j,k))+qcu(i);
 OVERALL_C(j) .. (TCOUT(j)-TCIN(j)) *FCPC(j) =e=
 sum((i,k),q(i,j,k))+qhu(j);

EBAL_H(i,k)\$HEN_HOT(i,k) .. (th(i,k)-th(i,k+1))*FCPH(i) =e=
 sum(j,q(i,j,k)); //do only exist HEN
 EBAL_C(j,k)\$HEN_COLD(j,k) .. (tc(j,k)-tc(j,k+1))*FCPC(j) =e=
 sum(i,q(i,j,k));

TIN_H(i) .. THIN(i) =e= th(i,'K1');
 TIN_C(j) .. TCIN(j) =e= tc(j,'KL');

MONOT_H(i,k)\$HEN_HOT(i,k) .. th(i,k) =g= th(i,k+1);
 //if HX --> monotonic
 MONOT_C(j,k)\$HEN_COLD(j,k) .. tc(j,k) =g= tc(j,k+1);
 MONOT_HNOT(i,k)\$((not HEN_HOT(i,k)) and stage(k)) .. th(i,k) =e= th(i,k+1);
 //if no HX --> equal
 MONOT_CNOT(j,k)\$((not HEN_COLD(j,k)) and stage(k)) .. tc(j,k) =e= tc(j,k+1);

MONOT_HU(j)\$HU(j) .. TCOUT(j) =g= tc(j,'K1'); //if HX --> monotonic
 MONOT_CU(i)\$CU(i) .. th(i,'KL') =g= THOUT(i);
 MONOT_HUNOT(j)\$((not HU(j)) .. TCOUT(j) =e= tc(j,'K1'); //if no HX --> equal
 MONOT_CUNOT(i)\$((not CU(i)) .. th(i,'KL') =e= THOUT(i);

```

EBAL_HU(j)$HU(j)      .. (TCOUT(j) - tc(j,'K1'))*FCPC(j) =e= qhu(j);
                                     //do only exist HEN
EBAL_CU(i)$CU(i)      .. (th(i,'KL') - THOUT(i)) *FCPH(i) =e= qcu(i);

DTFEAS_HXL(i,j,k)$HEN(i,j,k) .. dt(i,j,k) =e= th(i,k) -tc(j,k) ;
DTFEAS_HXR(i,j,k)$HEN(i,j,k) .. dt(i,j,k+1) =e= th(i,k+1) -tc(j,k+1);
DTFEAS_HUL(j)$HU(j)      .. dthul(j) =e= THUIN(j) -TCOUT(j) ;
DTFEAS_HUR(j)$HU(j)      .. dthur(j) =e= THUOUT(j) -tc(j,'K1');
                                     //if exist HEN and no TAU
DTFEAS_CUL(i)$CU(i)      .. dtcul(i) =e= th(i,'KL')-TCUOUT(i) ;
DTFEAS_CUR(i)$CU(i)      .. dtcur(i) =e= THOUT(i) -TCUIN(i) ;

LOGMTD(i,j,k)$HEN(i,j,k) .. lmtd(i,j,k) =e=
(dt(i,j,k)*dt(i,j,k+1)*(dt(i,j,k)+dt(i,j,k+1))/2)**(1/3); //only HEN
LOGMTDHU(j)$HU(j)      .. lmtdhu(j) =e= (dthur(j) *dthul(j) *(dthur(j)
+dthul(j)) /2)**(1/3);
LOGMTDCU(i)$CU(i)      .. lmtdcu(i) =e= (dtcul(i) *dtcur(i)
*(dtcul(i) +dtcur(i)) /2)**(1/3);

A_HX(i,j,k)$HEN(i,j,k)      .. area(i,j,k) *lmtd(i,j,k)*U(i,j) =e= q(i,j,k);
                                     //only HEN
A_HU(j)$HU(j)      .. areahu(j) *lmtdhu(j) *UHU(j) =e= qhu(j);
A_CU(i)$CU(i)      .. areacu(i) *lmtdcu(i) *UCU(i) =e= qcu(i);

//least square error
OBJ      .. TOT_LSQR_ERROR =e=
sum((i,j,k)$HEN(i,j,k),power(area(i,j,k)-AREA_OLD(i,j,k),2))
+ sum(j$HU(j),power(areahu(j)-AREAHU_OLD(j),2))
+ sum(i$CU(i),power(areacu(i)-AREACU_OLD(i),2)));

model STAGEMODEL_SINGLEPERIOD /all/;

option iterlim=1e9;
option domlim=0;
option reslim=1e8;
STAGEMODEL_SINGLEPERIOD.optfile=0;

solve STAGEMODEL_SINGLEPERIOD using NLP minimizing
TOT_LSQR_ERROR;

```

```
display th.l,tc.l,  
q.l,qhu.l,qcu.l,  
area.l,areahu.l,areacu.l,  
dt.l,dthul.l,dthur.l,dtcul.l,dteur.l,  
lmtl.l,lmtdu.l,lmtcu.l,  
tot_lsqr_error.l  
;
```

B3 Model C

SETS

I hot streams /I1*I3/
 J cold streams /J1*J4/
 K index of stage or location /K1*K4,KL/ //last number is no. of stages

stage(k) all stages
 HEN(i,j,k) define where HX exists
 HEN_HOT(i,k) define where HX exists 2 dimensions for hot streams
 HEN_COLD(j,k) define where HX exists 2 dimensions for cold streams
 HU(j) define where HU exists
 CU(i) define where CU exists
 ;

//dynamic set for stage pointer

stage(k) = yes;
 stage('KL') = no;

//dynamic set for existing HEN// <---- To fix topology

HEN(i,j,k) = no;

TABLE Z_OLD(i,j,k) Existing HX from original HEN of each period

	K1	K2	K3	K4
I1.J1	1.000	1.000	1.000	
I1.J2			1.000	1.000
I1.J3		1.000		
I3.J1		1.000	1.000	
I3.J4	1.000			

;

loop((i,j,k)\$ (Z_OLD(i,j,k)=1), HEN(i,j,k) = yes);

//dynamic set for existing Utility exchanger// <---- To fix topology

//hot utility

HU(j) = no;
 PARAMETER ZHU_OLD(j) Existing HU from original HEN of each period
 /J3 1.000/;
 loop(j\$ (ZHU_OLD(j)=1), HU(j) = yes);

//cold utility

CU(i) = no;
 PARAMETER ZCU_OLD(i) Existing CU from original HEN of each period
 /I1 1.000, I2 1.000, I3 1.000/;
 loop(i\$ (ZCU_OLD(i)=1), CU(i) = yes);

```
//define dynamic set for existing HEN but only 2 dimensions (for hot and cold)//
```

```
  HEN_HOT(i,k) = no;
  HEN_COLD(j,k) = no;
  loop((i,j,k)$ (HEN(i,j,k)), HEN_HOT(i,k) = yes);
  loop((i,j,k)$ (HEN(i,j,k)), HEN_COLD(j,k) = yes);
```

SCALARS

```
//Cost//
```

CA	per unit cost of heat exchanger area(\$ per unit)	/641.7/
B	exponent for area cost(dimensionless)	/1/
CF	fixed cost of heat exchanger(\$ per unit)	/8333.3/
CHU	per unit cost of hot utility(\$ per kW)	/115.2/
CCU	fixed cost of heat exchanger(\$ per kW)	/1.3/
AF	annualisation factor	/0.2/

```
//other//
```

EMAT	exchanger minimum approach temperature(C)	/5/
HHU	heat transfer coefficient of hot utility(kW (m2.C))	/2/
HCU	heat transfer coefficient of cold utility(kW (m2.C))	/1/

```
;
```

PARAMETERS

```
//Hot Streams//
```

THIN(i)	inlet temperature of hot streams(C)
/	I1 393
	I2 160
	I3 354
/	
THOUT(i)	outlet temperature of hot streams(C)
/	I1 60
	I2 40
	I3 60
/	

```
//Cold Streams//
```

TCIN(j)	inlet temperature of cold streams(C)
/	J1 72
	J2 62
	J3 220
	J4 253
/	
TCOUT(j)	outlet temperature of cold streams(C)


```

/      J1  356
      J2  210
      J3  370
      J4  284
/
//Heat Capacity Flowrates//
FCPH(i) heat capaity flowrates fo hot streams(C)
/      I1  201.6
      I2  185.1
      I3  137.4
/
FCPC(j) heat capaity flowrates fo cold streams(C)
/      J1  209.4
      J2  141.6
      J3  176.4
      J4  294.4
/
//Utility//
THUIN(j) inlet temperature of hot utility(C)
/      J1  400
      J2  400
      J3  400
      J4  400
/
THUOUT(j) outlet temperature of hot utility(C)
/      J1  399
      J2  399
      J3  399
      J4  399
/
TCUIN(i) inlet temperature of cold utility(C)
/      I1  15
      I2  15
      I3  15
/
TCUOUT(i) outlet temperature of cold utility(C)
/      I1  20
      I2  20
      I3  20
/
//Heat transfer coefficient//
HH(i) heat capaity flowrates of hot streams(kW|(m2.C))
/      I1  2
      I2  2
      I3  2
/
HC(j) heat capaity flowrates of cold streams(kW|(m2.C))

```

```

/      J1  1.5
      J2  1.5
      J3  2
      J4  2
/
//overall heat transfer coefficient//
U(i,j)      Overall heat transfer coefficient for match(i,j)
UHU(j)      Overall heat transfer coefficient for cold stream(j) and hot utility
UCU(i)      Overall heat transfer coefficient for hot stream(i) and cold utility

//upper bound for logical constraints//
GAMMA_HX(i,j,k)  upper bound heat load for match (i,j)
GAMMA_HU(j)      upper bound heat load for hot utility
GAMMA_CU(i)      upper bound heat load for cold utility

TAU_HX(i,j,k)   upper bound of temperature difference for heat exchanger
TAU_HUL(j)      upper bound of temperature difference for left side of hot
utility exchanger
TAU_HUR(j)      upper bound of temperature difference for right side of hot
utility exchanger
TAU_CUL(i)      upper bound of temperature difference for left side pf cold
utility exchanger
TAU_CUR(i)      upper bound of temperature difference for right side of cold
utility exchanger
;

//Old value of variables from original HEN//

//Area//
TABLE AREA_OLD(i,j,k) old area from original HEN of each period
      K1      K2      K3      K4

I1.J1  680.055  166.126  751.970
I1.J2           1290.040  132.794
I1.J3           597.190
I3.J1           356.123  1470.846
I3.J4  324.859
;

PARAMETERS
AREAHU_OLD(j) old area of HU from original HEN
/J3 256.025/
AREACU_OLD(i) old area of CU from original HEN
/I1 117.036, I2 541.396, I3 97.762/
;

```

//Temperature//

TABLE TH_OLD(i,k) old hot temperature from original HEN

	K1	K2	K3	K4	KL
I1	406.000	293.498	231.178	88.555	79.812
I2	160.000	160.000	160.000	160.000	160.000
I3	362.000	268.540	234.038	86.252	86.252

;

TABLE TC_OLD(j,k) old cold temperature from original HEN

	K1	K2	K3	K4	KL
J1	365.000	255.334	216.175	72.000	72.000
J2	210.000	210.000	210.000	74.710	62.000
J3	272.717	272.717	220.000	220.000	220.000
J4	290.000	250.000	250.000	250.000	250.000

;

//Heat load//

TABLE Q_OLD(i,j,k) old heat load from original HEN

	K1	K2	K3	K4
I1.J1	23062.852	3528.959	10162.028	
I1.J2		19075.844	1792.156	
I1.J3		9246.623		
I3.J1		4706.050	20158.011	
I3.J4	12748.000			

;

PARAMETERS

QHU_OLD(j) old heat load of HU from original HEN

/J3 17063.377/

QCU_OLD(i) old heat load of CU from original HEN

/I1 4061.538, I2 23856.000, I3 3580.739/

;

//Temperature Difference//

TABLE DT_OLD(i,j,k) old temperature differene from original HEN

	K1	K2	K3	K4	KL
I1.J1	41.000	38.165	15.003	16.555	
I1.J2		21.178	13.844	17.812	
I1.J3		20.781	11.178		
I3.J1		13.206	17.863	14.252	
I3.J4	72.000	18.540			

;

PARAMETERS

DTHUL_OLD(j) old left-side temperature difference of HU from original HEN
/J3 30.000/

DTHUR_OLD(j) old right-side temperature difference of HU from original HEN
/J3 126.283/

DTCUL_OLD(i) old left-side temperature difference of CU from original HEN
/I1 59.812, I2 140.000, I3 66.252/

DTCUR_OLD(i) old right-side temperature difference of CU from original HEN
/I1 45.000, I2 25.000, I3 45.000/

;

//Log Mean Temperature Difference//

TABLE LMTD_OLD(i,j,k) old log mean temperature from original HEN
K1 K2 K3 K4

I1.J1	39.565	24.783	15.766	
I1.J2		17.252	15.745	
I1.J3		15.484		
I3.J1		15.417	15.989	
I3.J4	39.242			

;

PARAMETERS

LMTDHU_OLD(j) old log mean temperature of HU from original HEN
/J3 66.647/

LMTDCU_OLD(i) old log mean temperature of CU from original HEN
/I1 52.055, I2 66.096, I3 54.941/

;

//Overall heat transfer coefficient calculations//

$U(i,j) = 1/(1/HH(i)+1/HC(j));$

$UHU(j) = 1/(1/HC(j)+1/HHU);$

$UCU(i) = 1/(1/HH(i)+1/HCU);$

VARIABLES

th(i,k) temperature(C) of hot streams i at location k

tc(j,k) temperature(C) of cold streams j at location k

q(i,j,k) heat load(kW) of match(i,j) at stage k

qhu(j) heat load(kW) of hot utility for cold stream j

qcu(i) heat load(kW) of cold utility for hot stream i

dt(i,j,k) temperature difference(C) for match(i,j) at location k

dthul(j) temperature difference(C) for left side of hot utility exchanger

dthur(j) temperature difference(C) for right side of hot utility exchanger

dtcul(i) temperature difference(C) for left side of cold utility exchanger

dtcur(i) temperature difference(C) for right side of cold utility exchanger

 lmtd(i,j,k) log mean temperature difference of match(i,j) at stage k
 lmtdhu(j) log mean temperature difference of hot utility exchanger for cold
 stream j
 lmtdcu(i) log mean temperature difference of cold utility exchanger for cold
 stream i

 area(i,j,k) area of heat exchanger of match(i,j) at stage k
 areahu(j) area of hot utility exchanger of cold stream j
 areacu(i) area of cold utility exchanger of hot stream i

 area_add(i,j,k) additional area of heat exchanger of match(i,j) at stage k
 areahu_add(j) additional area of hot utility exchanger of cold stream j
 areacu_add(i) additional area of cold utility exchanger of hot stream i

 tot_area_add total additional area needed
 ;

POSITIVE VARIABLES q(i,j,k), qhu(j), qcu(i),
 dt(i,j,k), dthul(j), dthur(j), dtcul(i), dtcur(i),
 lmtd(i,j,k),lmtdhu(j),lmtdcu(i),
 area(i,j,k),areahu(j),areacu(i)
 area_add(i,j,k), areahu_add(j), areacu_add(i);

//variable bounding//

th.up(i,k) = THIN(i);
 th.lo(i,k) = THOUT(i);
 tc.up(j,k) = TCOU(j);
 tc.lo(j,k) = TCIN(j);

q.up(i,j,k)\$stage(k) = max(min(FCPH(i)*(th.up(i,k)-
 max(th.lo(i,k+1),tc.lo(j,k+1)+EMAT)) , FCPC(j)*(min(th.up(i,k)-
 EMAT,tc.up(j,k))-tc.lo(j,k+1))) , 0)\$ ((th.up(i,k)-tc.lo(j,k) >= EMAT) and
 (th.up(i,k+1)-tc.lo(j,k+1) >= EMAT)) +0\$((th.up(i,k)-tc.lo(j,k) < EMAT) or
 (th.up(i,k+1)-tc.lo(j,k+1) < EMAT)) ;

qhu.up(j) = max(FCPC(j)*(min(THUIN(j)-EMAT,TCOU(j))-
 tc.lo(j,'K1')) , 0)\$((THUIN(j)-TCOU(j) >= EMAT) and (THUOUT(j)-
 tc.lo(j,'K1') >= EMAT)) +0\$((THUIN(j)-TCOU(j) < EMAT) or (THUOUT(j)-
 tc.lo(j,'K1') < EMAT)) ;

qcu.up(i) = max(FCPH(i)*(th.up(i,'KL')-
 max(THOUT(i),TCUIN(i)+EMAT)) , 0)\$ ((th.up(i,'KL')-TCUOUT(i) >= EMAT)
 and (THOUT(i)-TCUIN(i) >= EMAT)) +0\$((th.up(i,'KL')-TCUOUT(i) < EMAT) or
 (THOUT(i)-TCUIN(i) < EMAT)) ;

```

dt.up(i,j,k) = max(EMAT,(th.up(i,k)-tc.lo(j,k)));
dt.lo(i,j,k) = EMAT;
dthul.up(j) = max(EMAT,THUIN(j)-TCOUT(j));
dthul.lo(j) = EMAT;
dthur.up(j) = max(EMAT,THUOUT(j)-tc.lo(j,'K1'));
dthur.lo(j) = EMAT;
dteul.up(i) = max(EMAT,th.up(i,'KL')-TCUOUT(i));
dteul.lo(i) = EMAT;
dteur.up(i) = max(EMAT,THOUT(i)-TCUIN(i));
dteur.lo(i) = EMAT;

lmtd.up(i,j,k)$stage(k) =
(dt.up(i,j,k)*dt.up(i,j,k+1)*(dt.up(i,j,k)+dt.up(i,j,k+1))/2)**(1/3);
lmtd.lo(i,j,k)$stage(k) =
(dt.lo(i,j,k)*dt.lo(i,j,k+1)*(dt.lo(i,j,k)+dt.lo(i,j,k+1))/2)**(1/3);
lmtdhu.up(j) =
(dthur.up(j)*dthul.up(j)*(dthur.up(j)+dthul.up(j))/2)**(1/3);
lmtdhu.lo(j) = (dthur.lo(j)*dthul.lo(j)*(dthur.lo(j)+dthul.lo(j))/2)**(1/3);
lmtdteur.up(i) =
(dteul.up(i)*dteur.up(i)*(dteul.up(i)+dteur.up(i))/2)**(1/3);
lmtdteur.lo(i) = (dteul.lo(i)*dteur.lo(i)*(dteul.lo(i)+dteur.lo(i))/2)**(1/3);

area.up(i,j,k)$stage(k) = q.up(i,j,k)/lmtd.lo(i,j,k)/U(i,j);
areahu.up(j) = qhu.up(j)/lmtdhu.lo(j)/UHU(j);
areacu.up(i) = qcu.up(i)/lmtdteur.lo(i)/UCU(i);

//Assign bounding to parameters in logical constraint//
GAMMA_HX(i,j,k) = q.up(i,j,k);
GAMMA_HU(j) = qhu.up(j);
GAMMA_CU(i) = qcu.up(i);

TAU_HX(i,j,k) = -th.lo(i,k)+tc.up(j,k)+dt.lo(i,j,k);
TAU_HUL(j) = -THUIN(j)+TCOUT(j)+dthul.lo(j);
TAU_HUR(j) = -THUOUT(j)+tc.up(j,'K1')+dthur.lo(j);
TAU_CUL(i) = -th.lo(i,'KL')+TCUOUT(i)+dteul.lo(i);
TAU_CUR(i) = -THOUT(i)+TCUIN(i)+dteur.lo(i);

//Last stage forcing bound//
q.fx(i,j,'KL') = 0;
lmtd.fx(i,j,'KL') = 0;
area.fx(i,j,'KL') = 0;
area_add.fx(i,j,'KL') = 0;

//To force no q, area change where there is no existing HX//
q.fx(i,j,k)$ (not HEN(i,j,k)) = 0;
area_add.fx(i,j,k)$ (not HEN(i,j,k)) = 0;

```

```
//initial value for variables//
area.l(i,j,k)   = AREA_OLD(i,j,k);
areahu.l(j)     = AREAHU_OLD(j);
areacu.l(i)     = AREACU_OLD(i);

area_add.l(i,j,k) = 0;
areahu_add.l(j)   = 0;
areacu_add.l(i)   = 0;

th.l(i,k)       = TH_OLD(i,k);
tc.l(j,k)       = TC_OLD(j,k);

q.l(i,j,k)      = Q_OLD(i,j,k);
qhu.l(j)        = QHU_OLD(j);
qcu.l(i)        = QCU_OLD(i);

dt.l(i,j,k)     = DT_OLD(i,j,k);
dthul.l(j)      = DTHUL_OLD(j);
dthur.l(j)      = DTHUR_OLD(j);
dteul.l(i)      = DTEUL_OLD(i);
dteur.l(i)      = DTEUR_OLD(i);

lmtl.l(i,j,k)   = LMTD_OLD(i,j,k);
lmtdhu.l(j)     = LMTDHU_OLD(j);
lmtdcu.l(i)     = LMTDCU_OLD(i);
```

EQUATIONS

```
OVERALL_H(i)    overall energy balance of hot stream i
OVERALL_C(j)    overall energy balance of cold stream j

EBAL_H(i,k)     energy balance of hot stream i in stage k
EBAL_C(j,k)     energy balance of cold stream j in stage k

TIN_H(i)        assignment inlet temperature of hot stream i
TIN_C(j)        assignment inlet temperature of cold stream j

MONOT_H(i,k)    monotonic in temperature of hot stream i at location k
MONOT_C(j,k)    monotonic in temperature of cold stream j at location k
MONOT_HNOT(i,k) monotonic in temperature of hot stream i at location k
                 if no HX
MONOT_CNOT(j,k) monotonic in temperature of cold stream j at location
                 k if no HX

MONOT_HU(j)     monotonic in outlet temperature of cold stream j
```

MONOT_CU(i)	monotonic in outlet temperature of hot stream i
MONOT_HUNOT(j)	monotonic in outlet temperature of cold stream j if no HX
MONOT_CUNOT(i)	monotonic in outlet temperature of hot stream i if no HX
EBAL_HU(j)	energy balance of hot utility of cold stream j
EBAL_CU(i)	energy balance of cold utility of hot stream i
DTFEAS_HXL(i,j,k)	temperature feasibility of heat exchanger of match(i,j) at location k
DTFEAS_HXR(i,j,k)	temperature feasibility of heat exchanger of match(i,j) at location k+1
DTFEAS_HUL(j)	temperature feasibility of hot utility of outlet cold stream j
DTFEAS_HUR(j)	temperature feasibility of hot utility of inlet cold stream j
DTFEAS_CUL(i)	temperature feasibility of cold utility of inlet hot stream i
DTFEAS_CUR(i)	temperature feasibility of cold utility of outlet hot stream i
LOGMTD(i,j,k)	log mean temperature of match(i,j) at stage k
LOGMTDHU(j)	log mean temperature of hot utility exchanger
LOGMTDCU(i)	log mean temperature of cold utility exchanger
ADDIAREA_HX1(i,j,k)	additional area needed for heat exchanger
ADDIAREA_HX2(i,j,k)	force ADDAREA_HX > 0
ADDIAREA_HU1(j)	additional area needed for hot utility
ADDIAREA_HU2(j)	force ADDAREA_HU > 0
ADDIAREA_CU1(i)	additional area needed for cold utility
ADDIAREA_CU2(i)	force ADDAREA_CU > 0
A_HX(i,j,k)	area of heat exchanger of match(i,j) at stage k
A_HU(j)	area of hot utility exchanger of cold stream j
A_CU(i)	area of cold utility exchanger of hot stream i
OBJ	objective function (total additional area)
;	
OVERALL_H(i)	.. (THIN(i) - THOUT(i))*FCPH(i) =e= sum((j,k),q(i,j,k))+qcu(i);
OVERALL_C(j)	.. (TCOUT(j)-TCIN(j)) *FCPC(j) =e= sum((i,k),q(i,j,k))+qhu(j);
EBAL_H(i,k)\$HEN_HOT(i,k)	.. (th(i,k)-th(i,k+1))*FCPH(i) =e= sum(j,q(i,j,k)); //do only exist HEN
EBAL_C(j,k)\$HEN_COLD(j,k)	.. (tc(j,k)-tc(j,k+1))*FCPC(j) =e= sum(i,q(i,j,k));
TIN_H(i)	.. THIN(i) =e= th(i,'K1');
TIN_C(j)	.. TCIN(j) =e= tc(j,'KL');


```

MONOT_H(i,k)$HEN_HOT(i,k) .. th(i,k) =g= th(i,k+1);
                                //if HX --> monotonic
MONOT_C(j,k)$HEN_COLD(j,k) .. tc(j,k) =g= tc(j,k+1);
MONOT_HNOT(i,k)$((not HEN_HOT(i,k)) and stage(k)) .. th(i,k) =e= th(i,k+1);
                                //if no HX --> equal
MONOT_CNOT(j,k)$((not HEN_COLD(j,k)) and stage(k)) .. tc(j,k) =e= tc(j,k+1);

MONOT_HU(j)$HU(j) .. TCOUT(j) =g= tc(j,'K1'); //if HX --> monotonic
MONOT_CU(i)$CU(i) .. th(i,'KL') =g= THOUT(i);
MONOT_HUNOT(j)$((not HU(j))) .. OUT(j) =e= tc(j,'K1'); //if no HX --> equal
MONOT_CUNOT(i)$((not CU(i))) .. th(i,'KL') =e= THOUT(i);

EBAL_HU(j)$HU(j) .. (TCOUT(j) - tc(j,'K1'))*FCPC(j) =e= qhu(j);
                                //do only exist HEN
EBAL_CU(i)$CU(i) .. (th(i,'KL') - THOUT(i)) *FCPH(i) =e= qcu(i);

DTFEAS_HXL(i,j,k)$HEN(i,j,k) .. dt(i,j,k) =e= th(i,k) -tc(j,k) ;
DTFEAS_HXR(i,j,k)$HEN(i,j,k) .. dt(i,j,k+1) =e= th(i,k+1) -tc(j,k+1);
DTFEAS_HUL(j)$HU(j) .. dthul(j) =e= THUIN(j) -TCOUT(j) ;
DTFEAS_HUR(j)$HU(j) .. dthur(j) =e= THUOUT(j) -tc(j,'K1');
                                //if exist HEN and no TAU
DTFEAS_CUL(i)$CU(i) .. dtcul(i) =e= th(i,'KL') -TCUOUT(i) ;
DTFEAS_CUR(i)$CU(i) .. dtcur(i) =e= THOUT(i) -TCUIN(i) ;

LOGMTD(i,j,k)$HEN(i,j,k) .. lmtd(i,j,k) =e=
                                (dt(i,j,k)*dt(i,j,k+1)*(dt(i,j,k)+dt(i,j,k+1))/2)**(1/3);
                                //only HEN
LOGMTDHU(j)$HU(j) .. lmtdhu(j) =e= (dthur(j) *dthul(j) *(dthur(j)
                                +dthul(j)) /2)**(1/3);
LOGMTDCU(i)$CU(i) .. lmtdhu(i) =e= (dtcul(i) *dtcur(i) *(dtcul(i)
                                +dtcur(i)) /2)**(1/3);

A_HX(i,j,k)$HEN(i,j,k) .. area(i,j,k) *lmtd(i,j,k)*U(i,j) =e= q(i,j,k); //only HEN
A_HU(j)$HU(j) .. areahu(j) *lmtdhu(j) *UHU(j) =e= qhu(j);
A_CU(i)$CU(i) .. areacu(i) *lmtdhu(i) *UCU(i) =e= qcu(i);

ADDIAREA_HX1(i,j,k)$HEN(i,j,k) .. area_add(i,j,k) =g= area(i,j,k) -
                                AREA_OLD(i,j,k);
ADDIAREA_HX2(i,j,k)$HEN(i,j,k) .. area_add(i,j,k) =g= 0;
ADDIAREA_HU1(j)$HU(j) .. areahu_add(j) =g= areahu(j) - AREA_HU_OLD(j);
ADDIAREA_HU2(j)$HU(j) .. areahu_add(j) =g= 0;
ADDIAREA_CU1(i)$CU(i) .. areacu_add(i) =g= areacu(i) - AREACU_OLD(i);
ADDIAREA_CU2(i)$CU(i) .. areacu_add(i) =g= 0;

```

```

//least square
//OBJ      .. TAC =e=
           sum((i,j,k)$HEN(i,j,k) , power(AREAHX(i,j,k)-AREAOLD(i,j,k),2) )
           + sum(j$(HU(j)) , power(AREAHU(j)-AREAHUOLD(j),2) )
           + sum(i$(CU(i)) , power(AREACU(i)-AREACUOLD(i),2) );

//penalty only increased area
OBJ      .. TOT_AREA_ADD =e=
           sum((i,j,k)$HEN(i,j,k),area_add(i,j,k))
           + sum(j$(HU(j)),areahu_add(j))
           + sum(i$(CU(i)),areacu_add(i));

model STAGEMODEL_SINGLEPERIOD /all/;

option iterlim=1e9;
option domlim=0;
option reslim=1e8;
STAGEMODEL_SINGLEPERIOD.optfile=0;

solve STAGEMODEL_SINGLEPERIOD using NLP minimizing
TOT_AREA_ADD;

display th.l,tc.l,
       q.l,qhu.l,qcu.l,
       area.l,areahu.l,areacu.l,
       dt.l,dthul.l,dthur.l,dtcul.l,dteur.l,
       lmt.d.l,lmt.dhu.l,lmt.dcu.l,
       tot_area_add.l,
       ;

```

Appendix C Source Code of Simultaneous MINLP Multiperiod Model

The following text is source code which is entered in GAMS for synthesizing multi-period HEN by simultaneous approach.

SETS

```

I hot streams           /I1*I3/
J cold streams         /J1*J4/
K index of stage or location /K1*K4,KL/ //last number is no. of stages
P period of operation  /P1*P3/

```

//dynamic sets//

```

stage(k) all stages
;
stage(k) = yes; stage('KL') = no;

```

SCALARS

//Cost//

```

CA per unit cost of heat exchanger area($ per unit) /641.7/
CF fixed cost of heat exchanger($ per unit) /8333.3/

CHU per unit cost of hot utility($ per kW) /115.2/
CCU fixed cost of heat exchanger($ per kW) /1.3/

AF annualisation factor /0.2/

```

//other//

```

EMAT exchanger minimum approach temperature(C) /5/

HHU heat transfer coefficient of hot utility(kW|(m2.C)) /2/
HCU heat transfer coefficient of cold utility(kW|(m2.C)) /1/
;

```

//Hot Streams//

```

TABLE THIN(i,p) inlet temperatures of hot stream i in period p(C)
      P1      P2      P3
I1    393     406     420
I2    160     160     160
I3    354     362     360
;

```

```

TABLE THOUT(i,p) outlet temperatures of hot stream i in period p(C)
      P1      P2      P3
I1    60      60      60
I2    40      40      40
I3    60      60      60

```

;

//Cold Streams//

TABLE TCIN(j,p) inlet temperatures of cold stream j in period p(C)

	P1	P2	P3
J1	72	72	72
J2	62	62	62
J3	220	220	220
J4	253	250	249

;

TABLE TCOUT(j,p) outlet temperatures of cold stream j in period p(kW|K)

	P1	P2	P3
J1	356	365	373
J2	210	210	210
J3	370	370	370
J4	284	290	286

;

//Heat Capacity Flowrates//

TABLE FCPH(i,p) heat capacity flowrates of hot stream i in period p(kW|K)

	P1	P2	P3
I1	201.6	205.0	208.5
I2	185.1	198.8	175.2
I3	137.4	136.4	134.1

;

TABLE FCPC(j,p) heat capacity flowrates of cold stream j in period p(kW|K)

	P1	P2	P3
J1	209.4	210.3	211.1
J2	141.6	141.0	140.5
J3	176.4	175.4	174.5
J4	294.4	318.7	271.2

;

//Utility//

TABLE THUIN(j,p) inlet temperature of hot utility for cold stream j in period p(C)

	P1	P2	P3
J1	400	400	400
J2	400	400	400
J3	400	400	400
J4	400	400	400

;

TABLE THUOUT(j,p) outlet temperature of hot utility for cold stream j in period p(C)

	P1	P2	P3
J1	399	399	399
J2	399	399	399
J3	399	399	399

```

      J4  399      399      399
      ;
TABLE  TCUIN(i,p) inlet temperature of cold utility for hot stream i in
              period p(C)
              P1      P2      P3
      I1   15      15      15
      I2   15      15      15
      I3   15      15      15
      ;
TABLE  TCUOUT(i,p) outlet temperature of cold utility for hot stream i in
              period p(C)
              P1      P2      P3
      I1   20      20      20
      I2   20      20      20
      I3   20      20      20
      ;

```

PARAMETERS

//Heat transfer coefficient//

HH(i) heat capacity flowrates of hot streams(kW|(m2.C))

/ I1 2

I2 2

I3 2

/

HC(j) heat capacity flowrates of cold streams(kW|(m2.C))

/ J1 1.5

J2 1.5

J3 2

J4 2

/

//Duration of each period//

DOP(p) duration of each period

/ P1 4

P2 4

P3 4

/

//Total duration//

TOP total duration of all period

//overall heat transfer coefficient//

U(i,j) Overall heat transfer coefficient for match(i,j)

UHU(j) Overall heat transfer coefficient for cold stream(j) and hot utility

UCU(i) Overall heat transfer coefficient for hot stream(i) and cold utility

//upper bound for logical constraints//

GAMMA_HX(i,j,k,p) upper bound heat load for match (i,j) for period p
 GAMMA_HU(j,p) upper bound heat load for hot utility for period p
 GAMMA_CU(i,p) upper bound heat load for cold utility for period p

 TAU_HX(i,j,k,p) upper bound of temperature difference for heat exchanger
 for period p
 TAU_HUL(j,p) upper bound of temperature difference for left side of hot
 utility exchanger for period p
 TAU_HUR(j,p) upper bound of temperature difference for right side of hot
 utility exchanger for period p
 TAU_CUL(i,p) upper bound of temperature difference for left side of cold
 utility exchanger for period p
 TAU_CUR(i,p) upper bound of temperature difference for right side of cold
 utility exchanger for period p

 ;

//Overall heat transfer coefficient calculations//

$U(i,j) = 1/(1/HH(i)+1/HC(j));$
 $UHU(j) = 1/(1/HC(j)+1/HHU);$
 $UCU(i) = 1/(1/HH(i)+1/HCU);$

//TOP calculation//

TOP = sum(p,DOP(p));

VARIABLES

th(i,k,p) temperature(C) of hot streams i at location k for period p
 tc(j,k,p) temperature(C) of cold streams j at location k for period p

 q(i,j,k,p) heat load(kW) of match(i,j) at stage k for period p
 qhu(j,p) heat load(kW) of hot utility for cold stream j for period p
 qcu(i,p) heat load(kW) of cold utility for hot stream i for period p

 z(i,j,k) existence of match(i,j) at stage k
 zhu(j) existence of hot utility for cold stream j
 zcu(i) existence of cold utility for hot stream i

 dt(i,j,k,p) temperature difference(C) for match(i,j) at location k for period p
 dthul(j,p) temperature difference(C) for left side of hot utility exchanger
 for period p
 dthur(j,p) temperature difference(C) for right side of hot utility exchanger
 for period p
 dtcul(i,p) temperature difference(C) for left side of cold utility exchanger
 for period p
 dtcur(i,p) temperature difference(C) for right side of cold utility exchanger
 for period p

$lmtd(i,j,k,p)$ log mean temperature difference of match(i,j) at stage k for period p
 $lmtdhu(j,p)$ log mean temperature difference of hot utility exchanger for cold stream j for period p
 $lmtdcu(i,p)$ log mean temperature difference of cold utility exchanger for cold stream i for period p

 $area(i,j,k,p)$ area of heat exchanger of match(i,j) at stage k for period p
 $areahu(j,p)$ area of hot utility exchanger of cold stream j for period p
 $areacu(i,p)$ area of cold utility exchanger of hot stream i for period p

 $area_max(i,j,k)$ maximum area of heat exchanger of match(i,j) at stage k for all period
 $areahu_max(j)$ maximum area of hot utility exchanger of cold stream j for all period
 $areacu_max(i)$ maximum area of cold utility exchanger of hot stream i for all period

 $cost_fix$ fixed cost
 $cost_area$ area cost
 $cost_utility$ utility cost

 TAC total annualized cost
;

POSITIVE VARIABLES $q(i,j,k,p)$, $qhu(j,p)$, $qcu(i,p)$,
 $dt(i,j,k,p)$, $dthul(j,p)$, $dthur(j,p)$, $dteul(i,p)$, $dteur(i,p)$,
 $LMTD(i,j,k,p)$, $LMTDHU(j,p)$, $LMTDCU(i,p)$,
 $AREAHX(i,j,k,p)$, $AREAHU(j,p)$, $AREACU(i,p)$
 $AREAHX_max(i,j,k)$, $AREAHU_max(j)$, $AREACU_max(i)$;

BINARY VARIABLES $z(i,j,k)$, $zhu(j)$, $zcu(i)$;

//variable bounding//

$th.up(i,k,p) = THIN(i,p)$;
 $th.lo(i,k,p) = THOUT(i,p)$;
 $tc.up(j,k,p) = TCOUT(j,p)$;
 $tc.lo(j,k,p) = TCIN(j,p)$;

$q.up(i,j,k,p)\$stage(k) = \max(\min(FCPH(i,p)*(th.up(i,k,p) - \max(th.lo(i,k+1,p), tc.lo(j,k+1,p) + EMAT)), FCPC(j,p)*(\min(th.up(i,k,p) - EMAT, tc.up(j,k,p)) - tc.lo(j,k+1,p))), 0) \$ ((th.up(i,k,p) - tc.lo(j,k,p) \geq EMAT) \text{ and } (th.up(i,k+1,p) - tc.lo(j,k+1,p) \geq EMAT)) + 0 \$ ((th.up(i,k,p) - tc.lo(j,k,p) < EMAT) \text{ or } (th.up(i,k+1,p) - tc.lo(j,k+1,p) < EMAT))) ;$

$qhu.up(j,p) = \max(FCPC(j,p) * (\min(THUIN(j,p) - EMAT, TCOUT(j,p)) - tc.lo(j, 'K1', p)), 0) \$ ((THUIN(j,p) - TCOUT(j,p) \geq EMAT) \text{ and } (THUOUT(j,p) - tc.lo(j, 'K1', p) \geq EMAT)) + 0 \$ ((THUIN(j,p) - TCOUT(j,p) < EMAT) \text{ or } (THUOUT(j,p) - tc.lo(j, 'K1', p) < EMAT)) ;$

$qcu.up(i,p) = \max(FCPH(i,p) * (\max(THOUT(i,p), TCUIN(i,p) + EMAT)), 0) \$ ((th.up(i, 'KL', p) - TCUOUT(i,p) \geq EMAT) \text{ and } (THOUT(i,p) - TCUIN(i,p) \geq EMAT)) + 0 \$ ((th.up(i, 'KL', p) - TCUOUT(i,p) < EMAT) \text{ or } (THOUT(i,p) - TCUIN(i,p) < EMAT)) ;$

$z.up(i,j,k) \$ stage(k) = 0 \$ (\sum(p, q.up(i,j,k,p)) = 0) + 1 \$ (\sum(p, q.up(i,j,k,p)) \neq 0);$
 $zhu.up(j) = 0 \$ (\sum(p, qhu.up(j,p)) = 0) + 1 \$ (\sum(p, qhu.up(j,p)) \neq 0);$
 $zcu.up(i) = 0 \$ (\sum(p, qcu.up(i,p)) = 0) + 1 \$ (\sum(p, qcu.up(i,p)) \neq 0);$

$dt.up(i,j,k,p) = \max(EMAT, (th.up(i,k,p) - tc.lo(j,k,p)));$
 $dt.lo(i,j,k,p) = EMAT;$
 $dthul.up(j,p) = \max(EMAT, THUIN(j,p) - TCOUT(j,p));$
 $dthul.lo(j,p) = EMAT;$
 $dthur.up(j,p) = \max(EMAT, THUOUT(j,p) - tc.lo(j, 'K1', p));$
 $dthur.lo(j,p) = EMAT;$
 $dteul.up(i,p) = \max(EMAT, th.up(i, 'KL', p) - TCUOUT(i,p));$
 $dteul.lo(i,p) = EMAT;$
 $dteur.up(i,p) = \max(EMAT, THOUT(i,p) - TCUIN(i,p));$
 $dteur.lo(i,p) = EMAT;$

$lmtd.up(i,j,k,p) \$ stage(k) =$
 $(dt.up(i,j,k,p) * dt.up(i,j,k+1,p) * (dt.up(i,j,k,p) + dt.up(i,j,k+1,p)) / 2) ** (1/3);$
 $lmtd.lo(i,j,k,p) \$ stage(k) =$
 $(dt.lo(i,j,k,p) * dt.lo(i,j,k+1,p) * (dt.lo(i,j,k,p) + dt.lo(i,j,k+1,p)) / 2) ** (1/3);$
 $lmtdhu.up(j,p) =$
 $(dthur.up(j,p) * dthul.up(j,p) * (dthur.up(j,p) + dthul.up(j,p)) / 2) ** (1/3);$
 $lmtdhu.lo(j,p) =$
 $(dthur.lo(j,p) * dthul.lo(j,p) * (dthur.lo(j,p) + dthul.lo(j,p)) / 2) ** (1/3);$
 $lmtdteul.up(i,p) =$
 $(dteul.up(i,p) * dteur.up(i,p) * (dteul.up(i,p) + dteur.up(i,p)) / 2) ** (1/3);$
 $lmtdteul.lo(i,p) =$
 $(dteul.lo(i,p) * dteur.lo(i,p) * (dteul.lo(i,p) + dteur.lo(i,p)) / 2) ** (1/3);$

$area.up(i,j,k,p) \$ stage(k) = q.up(i,j,k,p) / LMTD.lo(i,j,k,p) / U(i,j);$
 $areahu.up(j,p) = qhu.up(j,p) / LMTDHU.lo(j,p) / UHU(j);$
 $areacu.up(i,p) = qcu.up(i,p) / LMTDCU.lo(i,p) / UCU(i);$

//Assign bounding to pamaters in logical constraint//

$GAMMA_HX(i,j,k,p) = q.up(i,j,k,p);$
 $GAMMA_HU(j,p) = qhu.up(j,p);$
 $GAMMA_CU(i,p) = qcu.up(i,p);$

$$\begin{aligned}
\text{TAU_HX}(i,j,k,p) &= -\text{th.lo}(i,k,p)+\text{tc.up}(j,k,p)+\text{dt.lo}(i,j,k,p); \\
\text{TAU_HUL}(j,p) &= -\text{THUIN}(j,p)+\text{TCOUT}(j,p)+\text{dthul.lo}(j,p); \\
\text{TAU_HUR}(j,p) &= -\text{THUOUT}(j,p)+\text{tc.up}(j,'K1',p)+\text{dthur.lo}(j,p); \\
\text{TAU_CUL}(i,p) &= -\text{th.lo}(i,'KL',p)+\text{TCUOUT}(i,p)+\text{dcul.lo}(i,p); \\
\text{TAU_CUR}(i,p) &= -\text{THOUT}(i,p)+\text{TCUIN}(i,p)+\text{dcur.lo}(i,p);
\end{aligned}$$

//Last stage forcing bound//

$$\begin{aligned}
\text{q.fx}(i,j,'KL',p) &= 0; \\
\text{z.fx}(i,j,'KL') &= 0; \\
\text{lmtd.fx}(i,j,'KL',p) &= 0; \\
\text{area.fx}(i,j,'KL',p) &= 0;
\end{aligned}$$

EQUATIONS

OVERALL_H(i,p)	overall energy balance of hot stream i for period p
OVERALL_C(j,p)	overall energy balance of cold stream j for period p
EBAL_H(i,k,p)	energy balance of hot stream i in stage k for period p
EBAL_C(j,k,p)	energy balance of cold stream j in stage k for period p
TIN_H(i,p)	assignment inlet temperature of hot stream i for period p
TIN_C(j,p)	assignment inlet temperature of cold stream j for period p
MONOT_H(i,k,p)	monotonic in temperature of hot stream i at location k for period p
MONOT_C(j,k,p)	monotonic in temperature of cold stream j at location k for period p
MONOT_HU(j,p)	monotonic in outlet temperature of cold stream j for period p
MONOT_CU(i,p)	monotonic in outlet temperature of hot stream i for period p
EBAL_HU(j,p)	energy balance of hot utility of cold stream j for period p
EBAL_CU(i,p)	energy balance of cold utility of hot stream i for period p
LOGIC_HX(i,j,k,p)	logical constraint to define z of match(i,j) at stage k for period p
LOGIC_HU(j,p)	logical constraint to define zhu of cold stream j at stage k for period p
LOGIC_CU(i,p)	logical constraint to define zcu of hot stream i at stage k for period p
DTFEAS_HXL(i,j,k,p)	temperature feasibility of heat exchanger of match(i,j) at location k for period p
DTFEAS_HXR(i,j,k,p)	temperature feasibility of heat exchanger of match(i,j) at location k+1 for period p

DTFEAS_HUL(j,p)	temperature feasibility of hot utility of outlet cold stream j for period p
DTFEAS_HUR(j,p)	temperature feasibility of hot utility of inlet cold stream j for period p
DTFEAS_CUL(i,p)	temperature feasibility of cold utility of inlet hot stream i for period p
DTFEAS_CUR(i,p)	temperature feasibility of cold utility of outlet hot stream i for period p
LOGMTD(i,j,k,p)	log mean temperature of match(i,j) at stage k for period p
LOGMTDHU(j,p)	log mean temperature of hot utility exchanger for period p
LOGMTDCU(i,p)	log mean temperature of cold utility exchanger for period p
A_HX(i,j,k,p)	area of heat exchanger of match(i,j) at stage k for period p
A_HU(j,p)	area of hot utility exchanger of cold stream j for period p
A_CU(i,p)	area of cold utility exchanger of hot stream i for period p
AMAX_HX(i,j,k,p)	maximum area of heat exchanger of match(i,j) at stage k for all period
AMAX_HU(j,p)	maximum area of hot utility exchanger of cold stream j for all period
AMAX_CU(i,p)	maximum area of cold utility exchanger of hot stream i for all period
FIXCOST	fixed cost
AREACOST	area cost
UTILITYCOST	utility cost
OBJ	objective function (total annualized cost)
;	
OVERALL_H(i,p)	.. (THIN(i,p) -THOUT(i,p))*FCPH(i,p) =e= sum((j,k),q(i,j,k,p))+qcu(i,p);
OVERALL_C(j,p)	.. (TCOUT(j,p)-TCIN(j,p)) *FCPC(j,p) =e= sum((i,k),q(i,j,k,p))+qhu(j,p);
EBAL_H(i,k,p)\$stage(k)	.. (th(i,k,p)-th(i,k+1,p))*FCPH(i,p) =e= sum(j,q(i,j,k,p));
EBAL_C(j,k,p)\$stage(k)	.. (tc(j,k,p)-tc(j,k+1,p))*FCPC(j,p) =e= sum(i,q(i,j,k,p));
TIN_H(i,p)	.. THIN(i,p) =e= th(i,'K1',p);
TIN_C(j,p)	.. TCIN(j,p) =e= tc(j,'KL',p);
MONOT_H(i,k,p)\$stage(k)	.. th(i,k,p) =g= th(i,k+1,p);

MONOT_C(j,k,p)\$stage(k) .. tc(j,k,p) =g= tc(j,k+1,p);
MONOT_HU(j,p) .. TCOUT(j,p) =g= tc(j,'K1',p);
MONOT_CU(i,p) .. THOUT(i,p) =l= th(i,'KL',p);

EBAL_HU(j,p) .. (TCOUT(j,p) - tc(j,'K1',p))*FCPC(j,p) =e= qhu(j,p);
EBAL_CU(i,p) .. (th(i,'KL',p) - THOUT(i,p))*FCPH(i,p) =e= qcu(i,p);

LOGIC_HX(i,j,k,p)\$stage(k) .. q(i,j,k,p) - GAMMA_HX(i,j,k,p)*z(i,j,k) =l= 0;
LOGIC_HU(j,p) .. qhu(j,p) - GAMMA_HU(j,p) *zhu(j) =l= 0;
LOGIC_CU(i,p) .. qcu(i,p) - GAMMA_CU(i,p) *zcu(i) =l= 0;

DTFEAS_HXL(i,j,k,p)\$stage(k) .. dt(i,j,k,p) =l= th(i,k,p) -tc(j,k,p)
+TAU_HX(i,j,k,p) *(1-z(i,j,k));
DTFEAS_HXR(i,j,k,p)\$stage(k) .. dt(i,j,k+1,p) =l= th(i,k+1,p) -tc(j,k+1,p)
+TAU_HX(i,j,k+1,p)*(1-z(i,j,k));
DTFEAS_HUL(j,p) .. dthul(j,p) =l= THUIN(j,p) -TCOUT(j,p)
+TAU_HUL(j,p) *(1-zhu(j));
DTFEAS_HUR(j,p) .. dthur(j,p) =l= THUOUT(j,p)
- tc(j,'K1',p)+TAU_HUR(j,p) *(1-zhu(j));
DTFEAS_CUL(i,p) .. dtcul(i,p) =l= th(i,'KL',p)-TCUOUT(i,p)
+TAU_CUL(i,p) *(1-zcu(i));
DTFEAS_CUR(i,p) .. dtcur(i,p) =l= THOUT(i,p) -TCUIN(i,p)
+TAU_CUR(i,p) *(1-zcu(i));

LOGMTD(i,j,k,p)\$stage(k) .. lmtd(i,j,k,p) =e=
(dt(i,j,k,p)*dt(i,j,k+1,p)*(dt(i,j,k,p)+dt(i,j,k+1,p))/2)**(1/3);
LOGMTDHU(j,p) .. lmtdhu(j,p) =e= (dthur(j,p) *dthul(j,p) *(dthur(j,p)
+dthul(j,p)) /2)**(1/3);
LOGMTDCU(i,p) .. lmtdhu(i,p) =e= (dtcul(i,p) *dtcur(i,p) *(dtcul(i,p)
+dtcur(i,p)) /2)**(1/3);

A_HX(i,j,k,p)\$stage(k) .. area(i,j,k,p) *lmtd(i,j,k,p)*U(i,j) =e= q(i,j,k,p);
A_HU(j,p) .. areahu(j,p) *lmtdhu(j,p) *UHU(j) =e= qhu(j,p);
A_CU(i,p) .. areacu(i,p) *lmtdhu(i,p) *UCU(i) =e= qcu(i,p);

AMAX_HX(i,j,k,p) .. area_max(i,j,k) =g= area(i,j,k,p);
AMAX_HU(j,p) .. areahu_max(j) =g= areahu(j,p);
AMAX_CU(i,p) .. areacu_max(i) =g= areacu(i,p);

FIXCOST .. cost_fix =e= AF*(sum((i,j,k)\$stage(k),CF*z(i,j,k))
+ sum(j,CF*zhu(j)) + sum(i,CF*zcu(i)));

AREACOST .. cost_area =e=
AF*(sum((i,j,k)\$stage(k),CA*area_max(i,j,k)))
+ AF*(sum(j,CA*areahu_max(j)))
+ AF*(sum(i,CA*areacu_max(i)));

```

UTILITYCOST      .. cost_utility =e=
                   sum(p,DOP(p)/TOP*(sum(j,CHU*qhu(j,p))))
                   + sum(p,DOP(p)/TOP*(sum(i,CCU*qcu(i,p))));

OBJ              .. TAC =e= cost_fix + cost_area + cost_utility;

model STAGEMODEL_SINGLEPERIOD /all/;

option iterlim=1e9;
option domlim=0;
option reslim=1e8;
STAGEMODEL_SINGLEPERIOD.optfile=0;

solve STAGEMODEL_SINGLEPERIOD using MINLP minimizing TAC;

display th.l,tc.l,
       z.l,zhu.l,zcu.l,
       q.l,qhu.l,qcu.l,
       area.l,areahu.l,areacu.l,
       area_max.l,areahu_max.l,areacu_max.l,
       dt.l,dthur.l,dthul.l,dtcul.l,dtcur.l,
       lmt.d.l, lmt.dhu.l, lmt.dcu.l
       cost_fix.l, cost_area.l, cost_utility.l, TAC.l
       ;

```