NTNU
Norwegian University of
Science and Technology

Combinatorial Problems in the Design of Heat Exchanger Networks

Martin Johnsen Megård

Master of Science in Mechanical Engineering	
Submission date:	February 2016
Supervisor:	Truls Gundersen, EPT
Co-supervisor:	Rahul Anantharaman, SINTEF Energy Reseach

Norwegian University of Science and Technology
Department of Energy and Process Engineering

EPT-M-2015

MASTER THESIS

for

Student Martin Johnsen Megård

Fall 2015

Combinatorial Problems in the Design of Heat Exchanger Networks

Kombinatoriske problemer ved design av varmevekslernettverk

Background and objective

Heat Exchanger Network Synthesis (HENS) is one of the most developed areas of process synthesis having received a great deal of attention from the industry and academia. The Sequential Framework methodology developed in the Department of Energy and Process Engineering is an iterative and sequential procedure for design and optimization of heat exchanger networks with the main objective to solve industrial sized problems. This is done by utilizing the strategy of dividing the HENS problem into a series of more managable sub-problems to reduce the computational complexity of obtaining a network design. This Sequential Framework has been implemented in the software tool SeqHENS based on combined use of GAMS and Excel.

Despite the decomposition of the HENS problem mentioned above (LP for utility targeting, MILP for units targeting with corresponding matching of hot and cold stream, and NLP for the actual network generation and optimization with given matches from the MILP stage), there are still severe combinatorial problems related to the task of matching the hot and cold streams (the MILP stage). The current status is that SeqHENS can handle problems with around 30 process streams, however, some problems are easier to solve, some are harder, depending on the actual stream data.

A match reduction approach has been proposed in the literature, where a sequence of subproblems are solved, each posed as a transportation problem, and where the set of matches is successively reduced from one stage to the next. A motivation for this approach is that for large problems, the process streams will naturally fall into groups that can be handled separately. In contrast, addressing the problem in a simultaneous way considering all possible matches between hot and cold streams results in a problem with a very high number of binary variables (equal to the product of the number of hot and the number of cold streams) and correspondingly severe combinatorial issues.

The original work on this new "grouping" approach was carried out at Aabo Academy in Finland, and a similar methodology has later been adopted by a group at Ecole Polytecnic Federal de Lausanne (EPFL) in Switzerland.

The ***main objective*** of this Master Thesis is to evaluate the proposed match reduction procedure in the framework of the SeqHENS software against the criteria of quality of the obtained solutions and the computational load involved in obtaining these solutions.

The following tasks are to be considered:

1. Make a brief literature search on recent approaches to solving the matching problem in HENS, with focus on the contributions from the mentioned groups at EPFL and Aabo Academy.
2. Explain similarities and differences between the two matching approaches mentioned in Task 1.
3. Implement this new matching approach in the software tool SeqHENS with focus on solving the fewest number of units sub-problem.
4. Test the procedure for 3 carefully selected cases with varying number of process streams in the range from 10 to at least 40 streams and compare with established approaches (in SeqHENS and/or the literature). Describe the quality and computational load by using the new procedure.
5. If time allows, consider the potential use of the new matching approach for the 2nd MILP in SeqHENS, where matches with favorable features related to total heat transfer area in the final network are established. If time does not allow its full implementation, then at least discuss the feasibility of this option and explain why it can or cannot be used.

-- " --

Within 14 days of receiving the written text on the master thesis, the candidate shall submit a research plan for his project to the department.

When the thesis is evaluated, emphasis is put on processing of the results, and that they are presented in tabular and/or graphic form in a clear manner, and that they are analyzed carefully.

The thesis should be formulated as a research report with summary both in English and Norwegian, conclusion, literature references, table of contents etc. During the preparation of the text, the candidate should make an effort to produce a well-structured and easily readable report. In order to ease the evaluation of the thesis, it is important that the cross-references are correct. In the making of the report, strong emphasis should be placed on both a thorough discussion of the results and an orderly presentation.

The candidate is requested to initiate and keep close contact with his/her academic supervisor(s) throughout the working period. The candidate must follow the rules and regulations of NTNU as well as passive directions given by the Department of Energy and Process Engineering.

Risk assessment of the candidate's work shall be carried out according to the department's procedures. The risk assessment must be documented and included as part of the final report. Events related to the candidate's work adversely affecting the health, safety or security, must be documented and included as part of the final report. If the documentation on risk assessment represents a large number of pages, the full version is to be submitted electronically to the supervisor and an excerpt is included in the report.

Pursuant to "Regulations concerning the supplementary provisions to the technology study program/Master of Science" at NTNU §20, the Department reserves the permission to utilize all the results and data for teaching and research purposes as well as in future publications.

The final report is to be submitted digitally in DAIM. An executive summary of the thesis including title, student's name, supervisor's name, year, department name, and NTNU's logo and name, shall be submitted to the department as a separate pdf file. Based on an agreement with the supervisor, the final report and other material and documents may be given to the supervisor in digital format.

□ Work to be done in lab (Water power lab, Fluids engineering lab, Thermal engineering lab)
□ Field work

Department of Energy and Process Engineering, 28 September 2015

Olav Bolland Department Head	Truls Gundersen Academic Supervisor

Co-supervisor:

Dr. Rahul Anantharaman, SINTEF Energy Research, E-mail: Rahul.Anantharaman@sintef.no

Abstract

Heat Exchanger Network Synthesis (HENS) is one of the most researched areas in process synthesis. The main objective of this work is to evaluate the possibility of implementing an alternative stream match generator into the Sequential Framework, to reduce the computational problems in the Mixed Integer Linear Programs (MILPs) formulations in larger HENS problems. The Sequential Framework is an iterative framework for the near-optimal synthesis of heat exchanger networks. Previous research has proven that the Sequential Framework has trouble solving larger problems (30+ streams), due to the large amount of binary variables. An alternative formulation, the sequential match reduction approach, is evaluated and implemented into the Sequential Framework, due to its ability to solve larger HENS problems by introducing binary variables for a selected group of matches.

The sequential match reduction approach has been coded into three General Algebraic Modeling System (GAMS) solvers, which have been implemented into the original framework. The two approaches have been used to solve 3 case studies of varying size from literature, and the results from the optimization process is documented in this thesis.

The solution quality for the original Sequential Framework proved a preferable alternative for HENS of smaller problems, where both the qualitative and quantitative aspects of the "optimal" networks generated were closer to the optimal solution.

Even though the implementation of an alternative stream match generator in SeqHENS led to networks with inferior qualitative and quantitative aspects to the original SeqHENS, the adapted framework had an ability of generating heat load distributions (HLDs) for larger problems, where the original framework could not generate a HLD due to combinatorial complexities.

However, after the generation of HLDs for larger problems, limitations in the network design procedure were detected. Due to the constraints of the Non-Linear Program (NLP) formulation used to optimize the network structure for the HLDs, the "optimal" HLDs generated for larger problems could not be designed. This limits the comparability of the networks for larger case studies.

Sammendrag

Syntese av varmevekslernettverk er et område prosessindustrien har viet mye fokus til over lengre tid. Hovedmålet med dette arbeidet er å evaluere muligheten for å implementere en alternativ prosedyre for å koble varme og kalde strømmer inn i det Sekvensielle rammeverket, for å redusere de kombinatoriske problemene som oppstår i større blandet heltalls og lineær programmering formuleringer når man optimalisere større varmevekslernettverk. Det Sekvensielle rammeverket er et iterativt rammeverk for nært-optimal syntese av varmevekslernettverk. Tidligere forsking har vist at det Sekvensielle rammeverk sliter med å løse større problemer (30+ strømmer), som følge av det høye antallet med binære variabler. En alternativ metode, den Sekvensielle reduksjon av enheter metoden, er evaluert og implementert i det Sekvensielle rammeverket. Dette er på grunn av at metodens evne til å løse større problemer ved hjelp av introduksjon av binære variabler for et vist antall strømpar.

Den Sekvensielle reduksjon av enheter metoden har blitt programmert inn i tre GAMS løsere som videre er blitt implementert i det Sekvensielle rammeverk. Det originale og alternative rammeverket er blitt brukt til å optimalisere 3 problemer fra litteraturen med ulik størrelse, og resultatene er presentert i denne oppgaven.

Resultatene viser at det originale rammeverket genererer nettverk som har bedre kvalitative og kvantitative evner for mindre problemer. Det alternative rammeverket generer løsninger på strømkoblinger i problemer på størrelser som det originale rammeverket ikke klarer, på grunn av de kombinatoriske problemene som oppstår.

Etter strømkoblingene for de større problemene ble generert, ble begrensningene i den ikke-lineære programmering formuleringen tydeligere. Design prosedyren i SeqHENS, som er softwaret basert på det Sekvensielle rammeverket, har begrensninger på antall strømmer. Dette fører at det «optimale» nettverket for større strømkoblinger ikke kunne bli designet.

Preface

This report presents the work done as part of the master thesis (TEP4905) in the fifth year of the study program of MSc in Mechanical Engineering at the Norwegian University of Science and Technology. The work was carried out at the Department of Energy and Process Engineering at the Faculty of Engineering Science and Technology.

I would like to thank my supervisors, Professor Truls Gundersen and Dr. Rahul Anantharaman, for their guidance and support throughout both my project work and master thesis.

Table of Content

List of Figures

List of Tables

Nomenclature

List of Roman Symbols

Symbol	*Description*	*Unit*
A	Area	[m^2]
c	Cost factor	[-]
C	Area Cost	[$/yr]
dt	Approximated log mean temp. difference	[K]
E	Energy	[kJ]
f_t	Correction factor	[-]
h	Heat transfer coefficient	[kJ/m^2*K]
I	Number of hot process streams	[-]
J	Number of cold process streams	[-]
M	Number of utility units	[-]
mCp	Heat Capacity rate	[kW/K]
N	Number of binary variables	[-]
ny	Number of subsystems	[-]
T	Temperature	[^oC]
U	Number of units	[-]
U	Overall heat transfer coefficient	[kJ/m^2*K]
w	Weight factor	[-]
y	Binary variable	[-]
Q	Heat	[kW]

List of Greek Symbols

ξ	Number of unpacked subsystems	[-]
φ	Subsystem index	[-]

List of Subscripts

cu	Cold utility
hu	Hot utility
i	Hot stream index
in	Inlet
j	Cold stream index
min	Minimum
out	Outlet

List of Abbreviations

B&B	Branch and bound
CP	Cold Process stream
CU	Cold Utility stream
EMAT	Exchanger Minimum Approach Temperature
GAMS	General Algebraic Modeling System
HEN	Heat Exchanger Network
HENS	Heat Exchanger Network Synthesis
HLD	Heat Load Distribution

HRAT	Heat Recovery Minimum Approach Temperature
HP	Hot process stream
HU	Hot utility stream
LMTD	Log Mean Temperature Difference
LP	Linear Program
MILP	Mixed Integer Linear Program
MINLP	Mixed Integer Non-Linear Program
NLP	Non-Linear Program
ObjFun	Objective Function
TAC	Total Annual Cost

1. Introduction

The heat exchanger network synthesis (HENS) problem is one of the most researched areas in the process industry the past 40 years, due to its ability to improve control and reduce cost by increasing the efficiency of a process. HENS involves solving the three-way trade-off between utility consumption (E), number of units (U) and area (A).

The problem has been proven to be a challenging task, which has led to several different methods for solving the problem has been proposed.

Heuristic approaches to the HENS problem have governed the process industry since its beginning. Until the 1970s, when Linnhoff and Flowers [1] introduced the pinch point, HENS was treated as a single problem. After the discovery of the pinch point researchers started to separate the HENS problem into three sub-problems that were solved sequentially. This sequential method reduced the complexity of the problem, but due to the problems complexity and three-way trade-off, it cannot be solve in a manual way.

The limitations discovered in the late 1980s, led to researchers focusing on simultaneous approaches, which treated the problem as one single problem to evaluate the different trade-offs in the problem.

After the HENS problem was proven to be NP-hard in the strong sense, a renewed interest in sequential synthesis methods was experienced. However, for larger problems, the number of binary variables causes computational explosions in the mixed integer linear program (MILP) formulations of the sequential methods.

This thesis presents three different sequential HENS methodologies, while focusing on the possibility of reducing the computational complexities in the MILPs for the Sequential Framework. The Sequential Framework is an iterative framework for the near-optimal synthesis of heat exchanger networks. The formulations of the framework are implemented in the SeqHENS-software, which consists of several General Algebraic Modeling System (GAMS)-solvers and Excel add-in files. The software has difficulties solving larger (30+ streams) HENS problems, with its bottleneck in the MILPs. The possibility of implementing the sequential match reduction approach by Pettersson [2] into SeqHENS is discussed and tested in this thesis.

1.1. Previous works

This thesis is an extension of a Specialization Project carried out in the spring of 2015. The project involved studies of different HENS methodologies and their solution quality on three different case studies. Two of the case studies selected in this thesis have previously been solved by the SeqHENS-software in the Specialization Project and is used as a comparison for the results in this thesis. This paper is based on the report written in context of the Specialization Project and some of the content is reused.

1.2. Risk Assessment

There has neither been done any laboratory work nor excursions related to this thesis, and it has therefore been regarded unnecessary to perform a risk assessment.

2. Heat Exchanger Network Synthesis

This chapter contains a short introduction of the HENS problem, as well as an introduction to heat exchanger network synthesis.

2.1. HENS problem

The heat exchanger network synthesis problem involves solving a three way trade-off between energy consumption (E), heat transfer area (A) and total number of heat exchangers (U).

The basic HENS problem was stated by Furman and Sahinidis [3] as follows:

Given a set of hot process streams (HP), which should be cooled from its supply temperature to its target temperature; a set of cold process streams (CP), which should be heated from its supply temperature to its target temperature ; the heat capacities and flow rates of the hot and cold process streams; the utilities available (e.g. hot utilities (HU) and cold utilities (CU) and their corresponding temperature and costs, develop a heat exchanger network with the minimum annualized investment and annual operating costs, i.e. minimum Total Annual Cost (TAC).

Including the quantitative aspect (TAC) of the HENS problem mentioned above, there are qualitative aspects such as complexity, operability and controllability of networks that needs to be accounted for when determining the «optimal» network.

2.2. HENS methods

Ever since the discovery of the heat recovery pinch in the 1970s, heat exchanger network synthesis has been an important research area in the process industry. Throughout the years, different methods have been created to try to solve the HENS problem and generate the optimal network for a given problem.

The large focus on HENS has led to several unique methods of solving the HENS problem. However, most methods can be categorized as either sequential or simultaneous.

2.2.1. Pinch Analysis

Pinch Analysis is a set of thermodynamically based heat exchanger network synthesis methods that generates the minimum energy consumption for a network. Pinch Analysis analyzes a chemical process and the surrounding utility systems with the use of the First and Second Laws of Thermodynamics, and solves the HENS problem sequentially.

Pinch Analysis separates the process design in to separate systems by finding the point(s) in which the design is at its most constrained, the «pinch point», and then starting the design from these points. This way of designing a heat exchanger network (HEN) makes the process reach its energy targets (Qh_{min} and Qc_{min}).

2.2.2. Sequential synthesis

Sequential synthesis methods divided the HENS problem into a multiple of subproblems to reduce the computational requirements and complexities of the problem.

The three most common subproblems, which are solved successively, are:

1. The minimum utility usage (Qc_{min} and Qh_{min}).
2. The minimum number of exchanger units (U_{min}).
3. The minimum area of the network (A).

Sequential synthesis methods can further be divided into two sub-categories:

1. Evolutionary design methods such as pinch design method, dual temperature approach and pseudo-pinch methods.

   Pinch-based design methods divides the problem into intervals, by using the heat recovery approach temperature (HRAT) and pinch point(s). An initial network is generated, which can be improved with the use of design guidelines and heuristic rules.

   The dual temperature approach method (DTAM) is also a pinch-based design method, but it allows heat transfer a cross the pinch by setting the exchanger minimum approach temperature (EMAT) to a lower value than HRAT. This method has a tendency to create networks with simpler structure with larger areas, due to the lower driving forces in the matches.

2. Mathematical programming techniques based on the sequential solution of continuous, integer linear programs, and nonlinear optimization problems.

   The multiple trade-offs and complexity of the HENS problem cannot be solved manually, and therefore optimization methods have been applied to solve these trade-offs. Using mathematical programming to solve the HENS problem in sequential synthesis consists of dividing the problem into the three sub problems mentioned previously.

   The minimum utility usage is usually formulated as a linear programming (LP) problem. The minimum utility target is then transferred as a constant to the MILP formulation for determining the minimum number of units and the HLD. The HLD from the stream matching MILP is used to generate a final HEN with a nonlinear programming (NLP) model. The NLP solves for the minimum capital cost, by minimizing the heat exchanger area in the matches from the HLD.

2.2.3. Simultaneous synthesis

Simultaneous synthesis involves finding the optimal heat exchanger network without dividing the HENS problem into sub problems. This is the optimal method of evaluating the multiple trade-offs in the HENS problem, as it assesses all the three factors at the same time.

Simultaneous synthesis methods are usually multi integer non-linear program (MINLP) formulations of the HENS problem, due to capital cost usually being exponential. The complexity of the problem leads to the addition of some simplifications to the formulation to reduce computational complexity.

One of the simplifications is the alternative Chen approximation (2.2) [4] for the log mean temperature difference(LMTD), instead of the actual LMTD. The Chen approximation removes the logarithmic terms to remove the computational problems that occur when $\Delta t_1 = \Delta t_2$.

$$LMTD = \frac{\Delta t_1 - \Delta t_2}{\ln(\frac{\Delta t_1}{\Delta t_2})} \qquad (2.1)$$

$$dt = (\Delta t_1 \times \Delta t_2 \times \frac{(\Delta t_1 + \Delta t_2)}{2})^{0.33} \qquad (2.2)$$

The Chen approximation underestimates the actual value of the LMTD, and therefore the area and cost of the heat exchanger will be slightly larger for the approximation. This makes the actual cost of the final network lower than the expected cost.

3. The Sequential Framework

This chapter will present an overview of the methodology of the Sequential Framework for heat exchanger network synthesis [5-7], as well as explaining the process of generating networks in the SeqHENS-software based on the Sequential Framework.

3.1. Introduction

As a compromise between Pinch Analysis and simultaneous synthesis, the Sequential Framework is an iterative framework for the near-optimal synthesis of heat exchanger networks. The framework takes into account how the selection of Heat Load Distributions (HLD) affects both the quantitative aspects and qualitative aspects of networks. The HENS problems are decomposed in the Sequential Framework based on the users knowledge about the problem. SeqHENS consists of several General Algebraic Modeling System (GAMS)-solvers and Excel add-in files.

3.2. Methodology

The Sequential Framework divides HENS problems into 4 subtasks: establishing the minimum energy consumption (LP), determining the minimum number of units (MILP), finding sets of matches and corresponding Heat Load Distributions for the minimum or given number of units (MILP), and a network generation and optimization (NLP) subtask as presented in Figure 1.

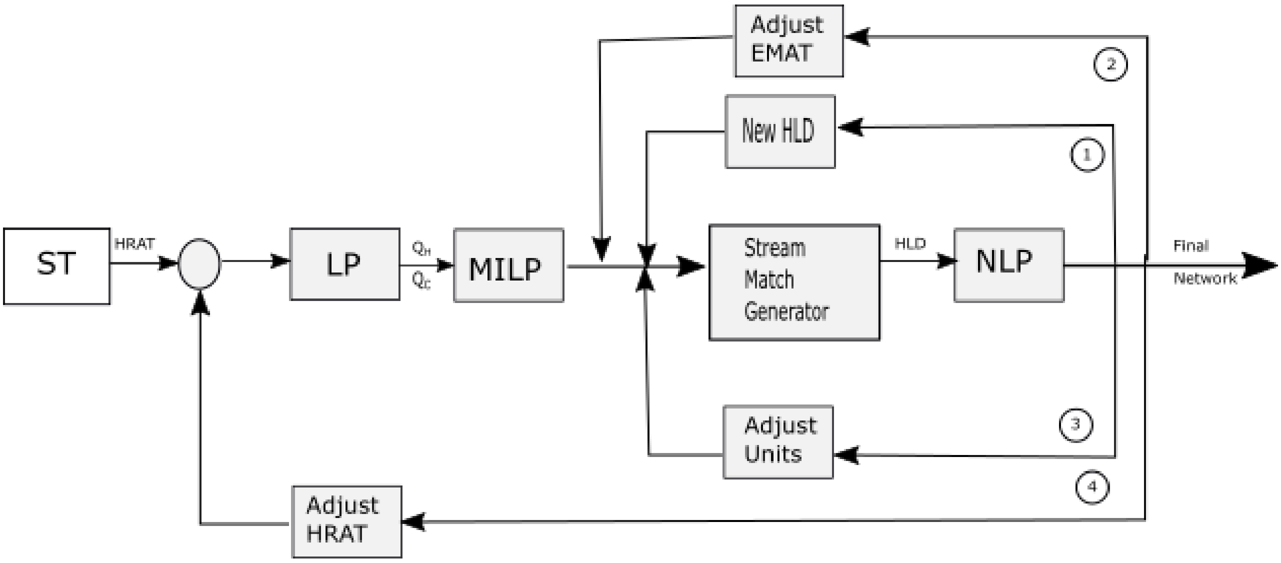
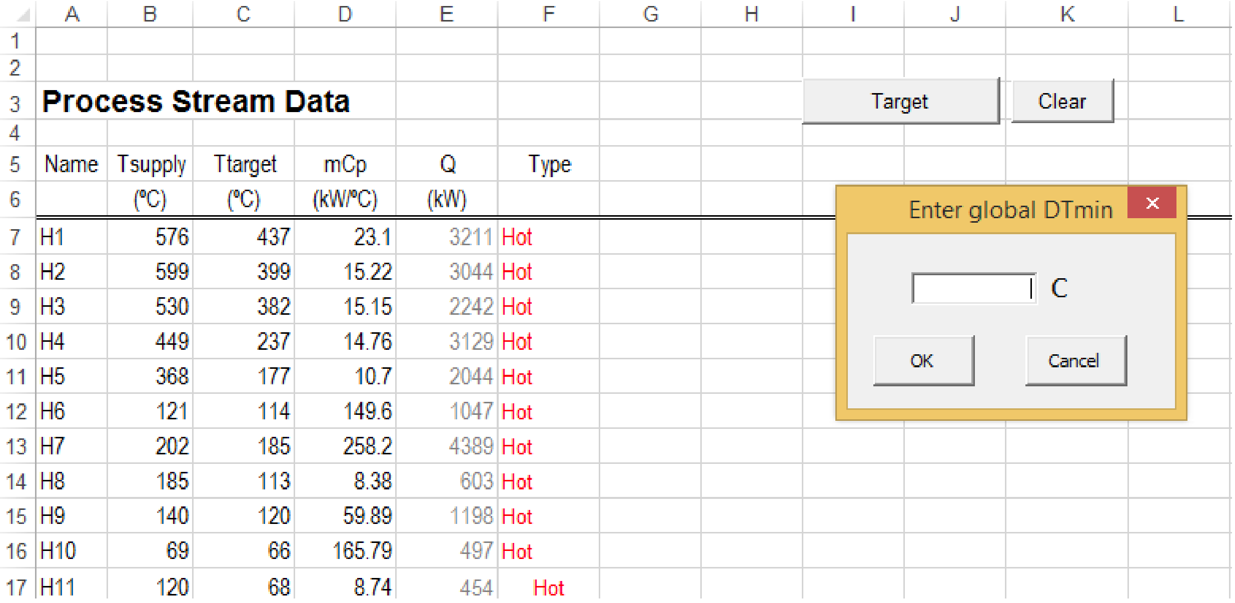


Figure 1 Methodology of the Sequential Framework

3.2.1. Minimum Utilities Targeting

For a given HRAT, the minimum hot and cold utility requirements are calculated using a LP transshipment model by Papoulias and Grossman [8] which is extended to include multiple utilities. The minimum utility target will be constant in the three inner loops of the framework, until the HRAT is adjusted in loop four.

SeqHENS uses an Excel file with the inserted stream data to calculate the minimum utility target for a given HRAT, as well as the pinch temperature(s) and the grand composite curves of the streams. The Excel interface is illustrated in Figure 2.



Process Stream Data

Name	Tsupply (°C)	Ttarget (°C)	mCp (kW/°C)	Q (kW)	Type
H1	576	437	23.1	3211	Hot
H2	599	399	15.22	3044	Hot
H3	530	382	15.15	2242	Hot
H4	449	237	14.76	3129	Hot
H5	368	177	10.7	2044	Hot
H6	121	114	149.6	1047	Hot
H7	202	185	258.2	4389	Hot
H8	185	113	8.38	603	Hot
H9	140	120	59.89	1198	Hot
H10	69	66	165.79	497	Hot
H11	120	68	8.74	454	Hot

Figure 2 Excel interface in SeqHENS.

3.2.2. Minimum Number of Units.

The minimum number of units subproblem is formulated as an MILP transshipment problem based on the model presented by Papoulias and Grossmann [8]. The SeqHENS model differs from the model by Papoulias and Grossmann, since no sub-networks are considered, i.e. there is no pinch decomposition and heat transfer across pinch point(s) can occur.

To allow heat transfer across pinch point(s), the exchanger minimum approach temperature (EMAT) is set to zero and the energy target calculated in the LP model are used as an input to discover the absolute minimum number of units for network.

SeqHENS uses an Excel add-in file where the user inputs problem data, similar to the one in the LP solver, and this data is transfered to a General Algebraic Modeling System (GAMS)-code to find the minimum number of units (U_{min}).

Due to the introduction of binary variables in the minimum units subproblem, this is the main bottleneck of SeqHENS, and will be discussed to further extent in Section 3.4.

3.2.3. Stream Match Generator

The Stream Match Generator problem generates a HLD for a given utility target, number of units and EMAT. The sub problem is formulated as an MILP transportation model based on the model published by Cerdá and Westerberg [9]. The objective of the function is to generate stream matches that reduce the «pseudo-area» of the heat exchanger network.

SeqHENS uses an Excel interface to collect problem data, and creates a GAMS-file that uses the CPLEX solver to find the «optimal» heat load distribution.

Because of the combinatorial complexity of this sub-problem, this step is one of the bottlenecks of the simulation. The simulation time can vary between a couple of seconds to several hours due to combinatorial explosions that may occur. For larger industrial problems (30+ streams) SeqHENS struggles to locate a HLD.

3.2.4. Network Generation and Optimization

The final step of the Sequential Framework involves locating the network with the lowest investment cost for the given HLD from the stream match generator and the set number of units. The Sequential Framework uses a nonlinear programming (NLP) formulation of the superstructure by Floudas et al. [10] for the network design. The total number of units on a specific process steam decides which superstructure that is used.

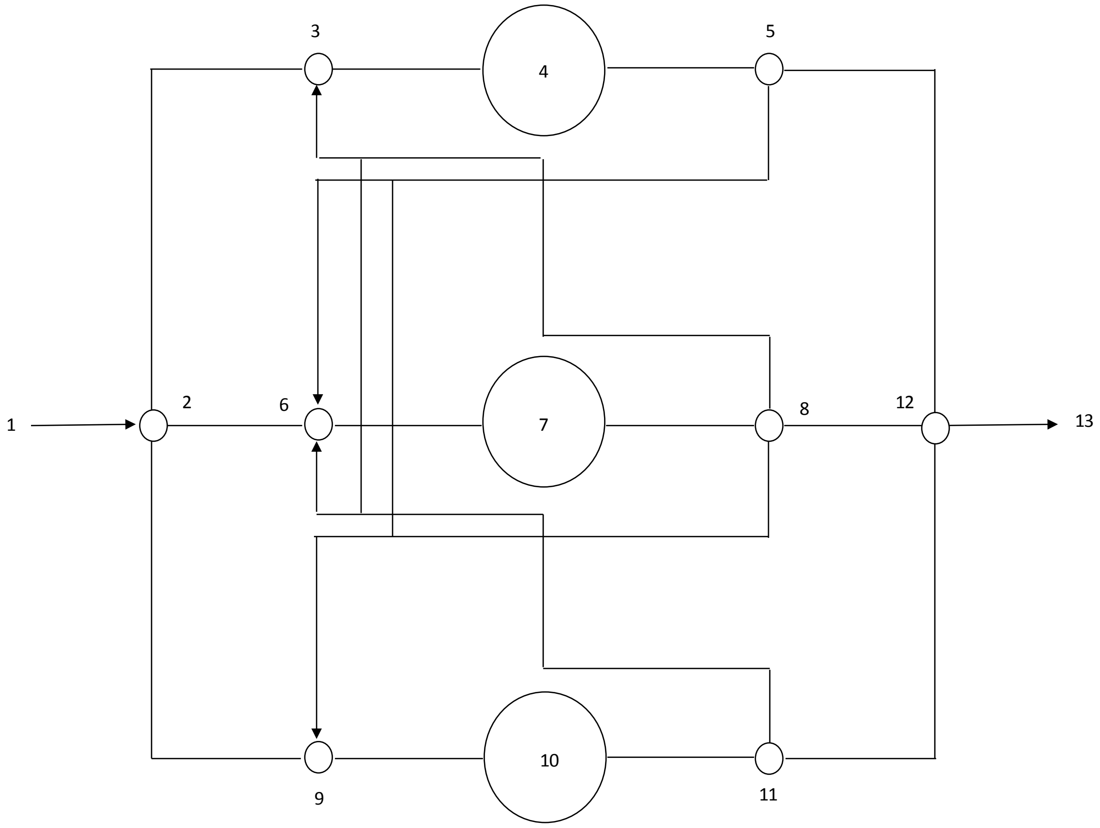
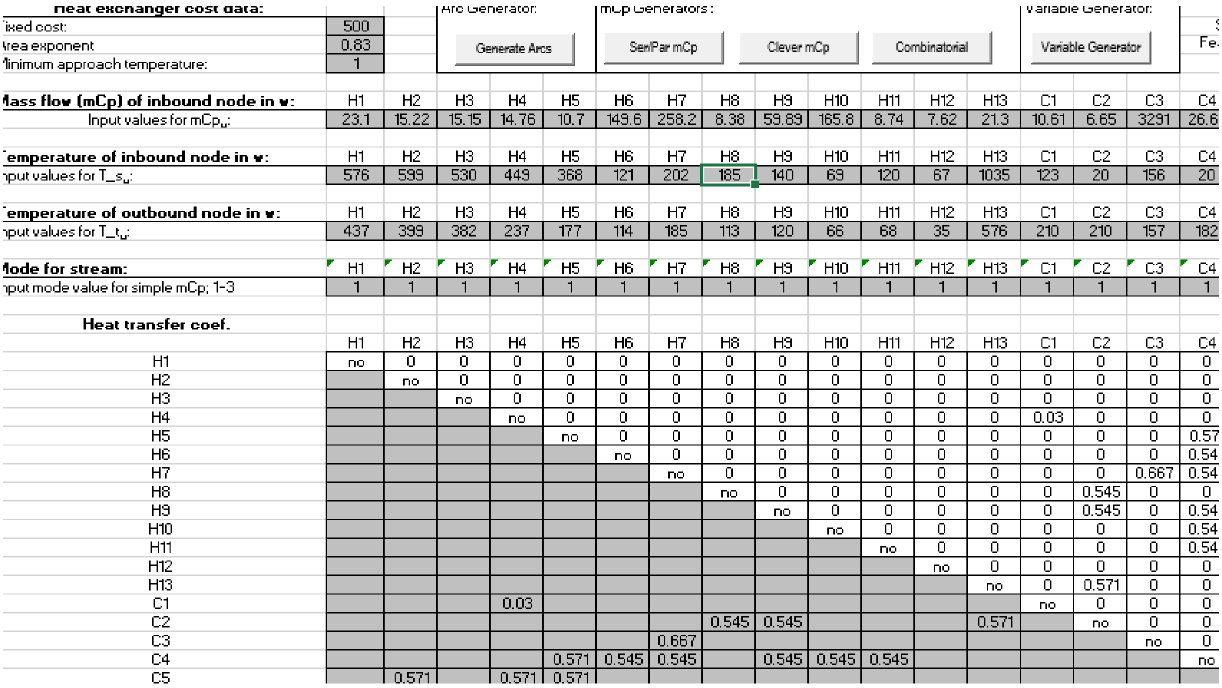


Figure 3 Superstructure for a three-unit process stream.

Each of the matches in the HLD from the stream match generator step represents a heat exchanger in the superstructure. Superstructures are generated for all of the process streams and the possible configurations for matches are included.

After the user has mapped out the superstructures for each process stream, the final network design is established from the combination of all the superstructures. This last part of the network design procedure is done manually and can be fairly time consuming for problems with several process streams.



heat exchanger cost data:	
ixed cost:	500
rea exponent	0.83
linimum approach temperature:	1

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	C1	C2	C3	C4
lass flow (mCp) of inbound node in v:	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	C1	C2	C3	C4
Input values for mCp_u:	23.1	15.22	15.15	14.76	10.7	149.6	258.2	8.38	59.89	165.8	8.74	7.62	21.3	10.61	6.65	3291	26.6
emperature of inbound node in v:	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	C1	C2	C3	C4
nput values for T_s_u:	576	599	530	449	368	121	202	185	140	69	120	67	1035	123	20	156	20
emperature of outbound node in v:	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	C1	C2	C3	C4
nput values for T_t_u:	437	399	382	237	177	114	185	113	120	66	68	35	576	210	210	157	182
lode for stream:	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	C1	C2	C3	C4
nput mode value for simple mCp; 1-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Heat transfer coef.

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	C1	C2	C3	C4
H1	no	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H2		no	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H3			no	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H4				no	0	0	0	0	0	0	0	0	0	0.03	0	0	0
H5					no	0	0	0	0	0	0	0	0	0	0	0	0.57
H6						no	0	0	0	0	0	0	0	0	0	0	0.54
H7							no	0	0	0	0	0	0	0	0	0.667	0.54
H8								no	0	0	0	0	0	0	0.545	0	0
H9									no	0	0	0	0	0	0.545	0	0.54
H10										no	0	0	0	0	0	0	0.54
H11											no	0	0	0	0	0	0.54
H12												no	0	0	0	0	0
H13													no	0	0.571	0	0
C1				0.03										no	0	0	0
C2								0.545	0.545				0.571		no	0	0
C3							0.667									no	0
C4					0.571	0.545	0.545		0.545	0.545	0.545						no
C5		0.571		0.571	0.571												

Figure 4 SeqHENS Excel NLP interface.

3.2.5. Loops in the Sequential Framework.

The four loops in the Sequential Framework (Figure 1) represent the three way trade-off in the HENS problem. Loops 1 and 2 can be thought of as area loops, loop 3 as the unit loop and loop 4 as the energy loop.

The loops lets the user evaluate the multiple networks with different parameters, which helps the user locate the HLD closest to the optimal solution.

SeqHENS does not go through the loops automatically and it is therefore necessary for the user to run the simulation and generate a new network for each parameter change to establish the network design with the lowest possible Total Annual Cost (TAC).

3.3. Advantages

The Sequential Framework has several advantages. The subtasks of the framework, the MILP and NLP problems, are easier numerically than the MINLP used in simultaneous HENS. The design procedure of the framework maintains user interaction while still being automated.

The 4 loops in the framework lets the user assess and compare multiple networks with different utility consumption, number of units and EMAT, and decide the «optimal» network based on both qualitative and quantitative aspects. This is a feature missing in other sequential methods.

The Sequential Framework creates networks with industrial realism, due to no model simplifications or assumptions, which sets it apart from simultaneous synthesis methods.

3.4. Challenges and Limitations

The main challenges of the Sequential Framework are in the minimum number of units subproblem and in the stream match generator. Due to «combinatorial explosion» that occur when faced with a large number of streams (20+), the sequential framework struggles to solve industrial sized problems without improvements.

The sequential framework is also limited by its inability to generate networks with more than one interconnection between the same two process streams.

3.5. Possible improvements

The Sequential Framework´s bottlenecks are at the minimum number of units subproblem and the stream matching subproblem. Both these are solved by using an MILP, and for larger problems (20+ streams), the MILPs will struggle with the large number of binary variables. This is explained in more detail in chapter 4.1.

The main bottleneck is therefore the minimum number of units subproblem, and will need to be improved before SeqHENS is a viable option for larger HENS problems.

4. Methods for reducing the computational issues in larger MILP problems

This chapter explains the problems that occur when solving larger MILP formulations, introduces advances in MILP solvers and takes a more extensive look at two specific optimization methods that try to reduce the computational complexities in HENS problems.

4.1. MILP Problems in HENS

The simultaneous methodology is the superior method for evaluating the multiple trade-offs of the HENS problem, however, the computational explosion that occurs during optimization has led to an increased focus on sequential approaches.

Even though improvements have been made to MILP solvers, the minimum units and stream match MILPs are difficult to solve for larger problems. This is due to the large number of binary variable in the HENS problem, and is explained in more detail in Section 4.1.1.

The number of binary variables in MILPs and MINLPs is very important when assessing computational time for specific problems. Equation 4.1. calculates the number of binary variables, *N*, for a problem with *I* hot streams and *J* cold streams.

$$N = I \cdot J \tag{4.1}$$

The computational time usually increases exponentially with the total number of streams in a problem. This is due to the Branch and Bound (B&B) algorithm frequently used when minimizing or maximizing MILP problems.

4.1.1. Branch and Bound

The branch and bound algorithm method [11] is the most commonly used method for solving optimization problems that are NP-hard in the strong sense.

The B&B algorithm consists of a systematic enumeration of candidate solutions by means of state space search. The set of solutions is thought of as forming a rooted tree with all the different solutions at the leaf node of the tree. When minimizing or maximizing an objective function, the different branches are cut off when a solution with a larger or smaller value than the current optimal solution is discovered. This reduces the computational time for most MILPs, but does not work as well for the MILPs in HENS.

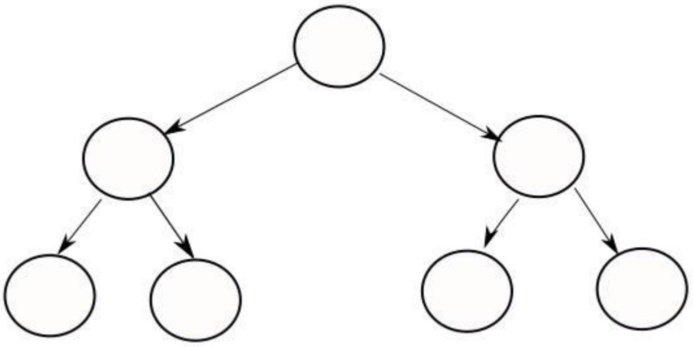


Figure 5 Illustration of the Branch & Bound optimization tree.

The objective function for the minimum units problem has coefficients that are 1.0 for all the binary variables, which tend to introduce degeneracy, meaning that multiple solutions have the same objective value. The large number of solutions with $U=U_{min}$, $U = U_{min} + 1$ and $U = U_{min} + 2$ in the branches lead to several solution branches not being cut off until the final splits, and thus increasing the computational time exponentially with the number of streams.

The fact that the B&B method may have several branches that lead to the same minimal solution of the objective function, which is the minimum units function, explains why unbalanced problems are easier to solve than balanced problems with close to the same number of hot and cold streams. The unbalanced problems may have less symmetry (i.e. branches with the same values), due to more restrictive matches for some of the streams with larger duties and heat capacity flows.

4.2. Advances in MILP solvers

Slow computational processes for the MILPs have led to an increased focus on reducing the bottleneck in the MILP for the minimum units problem in HENS. This chapter presents some computational strategies presents by Chen et al. [12] and by Anantharaman [6] for large-scale MILP transhipment models for HENS.

4.2.1. Weighted model

Introducing weighted matches to the objective function (Equation 4.2) is a way of making the branches in the B&B unsymmetrical, thus reducing the computational time for the process. The weight factors, w, can be determined by either stream flow rates or proximity of streams. Papoulias and Grossman [8] and Baliban et al. [13] have both introduced the use of weight factors to improve the stream matching problem.

$$\text{ObjFun} = min \sum_{i\in I} \sum_{j\epsilon J} w_{ij} y_{ij} \qquad (4.2)$$

The problem with the weighted model is that the number of binary variables, y, remains the same. The model still struggles to solve larger problems due to the explosion in number of combinations of hot and cold streams. The number of nodes in the B&B search tree is given by Equation 4.3, where N is the number of binary variables.

$$\text{Nodes} = 2^{N+1}\text{-}1 \qquad (4.3)$$

The results from Chen et al. [12] indicates that the problem with computational time occurs at about 30 streams for balanced problems and 40+ streams for unbalanced problems.

4.2.2. Pre-processing to reduce model size

Pre-processing the problem involves using insight and heuristics to reduce the model size by fixing some of the binary variables based on the stream properties.

One way of using the insight of the engineer is branching priority. Branching priority involves identifying the possible matches with a larger upper heat transfer bound and placing them higher in the branch and bound tree, so that they are determined earlier in the B&B algorithm.

4.2.3. Model modifications and reformulations

One way of reducing computational complexity is to reduce the difference between the LP relaxation and the actual binary solution, i.e. the gap. The size of the gap is dependent on the big M formulation, given in Equation 4.4. Here, Q_{ijk} is the heat transferred between a hot stream i and a cold stream j in a temperature interval k, and Y_{ij} is the binary variable representing a match between a hot and a cold stream.

$$Q_{ijk} - M_{ij}Y_{ij} \leq 0 \tag{4.4}$$

A larger gap between the LP relaxation and the actual binary solution leads to longer computational times. The big M was set to be the upper bound of the heat that can be exchanged between a hot process stream or utility and a cold process stream or utility (Equation 4.5) by Papoulias and Grossmann [8].

$$M_{ij} = \min[\textstyle\sum_k Q_h(i,k), \sum_l Q_c(j,l)] \tag{4.5}$$

The drawback of Equation 4.5 is that the temperatures of the streams are not taken into account when setting the M_{ij}. A more recent definition is the M_{ij} by Gundersen et al. [7] in Equation 4.6.

$$M_{ij} = \min[\textstyle\sum_k Q_h(i,k), \sum_l Q_c(j,l)]\cdot\max[\min(mCp_i, mCp_j)\cdot (Ts_i - Ts_j - EMAT),0] \tag{4.6}$$

This big M reduces the gap between the LP relaxation and the actual binary solutions, and reduces the state space search.

4.3. The sequential match reduction approach

This section presents the sequential match reduction approach by Pettersson [2].

4.3.1. Introduction

The sequential match reduction approach by Pettersson [2], is a method for HENS that is designed to ease the computational complexity of larger problems, by reducing the number of binary variables in the problem. This method will be implemented in the Sequential Framework and compared to the original framework in chapter 7.

4.3.2. Methodology

The sequential match reduction approach consists of a four-step process, illustrated in Figure 6, which generates a HEN as close as possible to the optimal design. The first three subproblems generate the HLD and groups the stream matches into manageable sized design tasks before the final design step, where the subsystems are solved separately.

All the formulations of the method are presented in Appendix A.

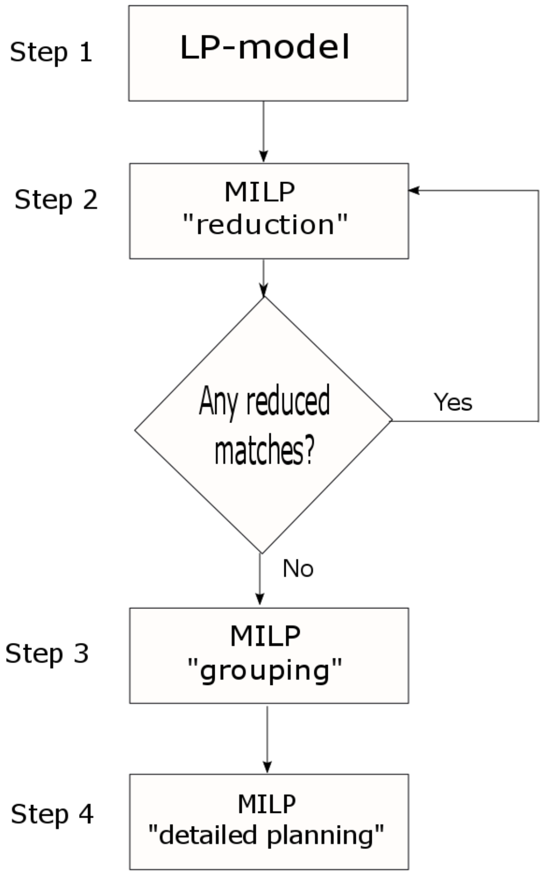


Figure 6 The sequential match reduction approach methodology.

Linear Transportation formulation

Step 1 is a linear transportation (LP) formulation (A1) designed to find the solution with the lowest total annual cost. The solution generated from the LP model is a feasible solution; however, the objective function, which is a linearized annual cost function of the total network (4.7),with a linear cost, does not contain the unit cost.

$$\text{ObjFun} = \min\,[c_1Q_{HU} + c_2Q_{CU} + c_3A] \qquad (4.7)$$

The annual cost factors for the objective function are given by the constants, c_1, c_2 and c_3. The unit area cost, $c_{3,}$ is the same for all types of units; however, this assumption can be relaxed to accommodate the linear cost function to problems with different unit costs for certain types of units.

Since the generated solution from the LP formulation does not include the unit cost, it normally has a considerable number of units and stream splits, making it economically unattractive. However, due to the high number of matches present in this solution, it is reasonable to assume that the most promising matches are included. The next steps in the sequential match reduction approach identifies the matches that should be included in the final design.

The objective function assesses the trade-offs between hot and cold utility and area, and that leads to the HLD generated being highly dependent on the cost factors of the individual problem. The HRAT is not a fixed value in the approach, and therefore the utility usage changes if different cost factors are introduced.

The minimum values for the objective function from step 1 is acting as a benchmark for the minimum cost of the problem throughout the sequential match reduction approach.

Match set reduction

The number of matches from step 1 is reduced by implementing the unit cost into a new objective function (4.8) and thereby challenging the matches in the HLD from step 1. The introduction of a binary variable, y, to represent a unit makes it necessary to use a MILP formulation (A2).

$$\text{ObjFun} = \min\ [c_1Q_{HU} + c_2Q_{CU} + c_3A + c_4\sum_{h\in H}\sum_{c\in C} y_{h,c} \qquad (4.8)$$

$$+ c_5\sum_{h\in H}\sum_{cu\in CU} y_{h,cu} + c_6\sum_{hu\in HU}\sum_{c\in C} y_{hu,c}]$$

Solving larger LP problems does generally not present computational problems, but for MILP problems size is very important. To keep the calculation time reasonably low, binary variables are only introduced for a selected number of matches, such as the matches with smaller heat duties.

The matches with smaller duties are penalized in the objection function which now assesses the trade-off between unit, area and utility cost. The cost function is minimized and some of the matches with binary variables are removed from the HLD. This process is repeated in the loop with new binary variables until the set of matches is no longer reduced. The user decides which matches to introduce binary variables to, and it is normally the matches with the lowest duty. However, some streams may have a very low duty compared to other streams, so one should consider to introduce binary variables to the units with the lowest fraction of the total duty of the streams, instead of only focusing on the duty of the single unit.

Grouping

The third step consists of identifying the streams that are not interconnected in the final HLD and grouping streams that are interconnected into subsets. The design procedure is then performed separately for these subsets. The formulation is presented in Appendix A.2.

If the size of subsets are too large or they cannot be found, some matches have to be eliminated from the set. The matches, which normally are eliminated, are the ones with smaller heat duties.

A process stream can only belong to one subgroup. If a process stream would belong to two or more subgroups the subgroups would not be independent and the design stage, step 4, would not solve the final network design faster than a non-grouped HLD.

Final Design

The fourth and final step in the process is the design stage. The subgroups from the grouping subproblem are implemented in a MILP formulation and the groups are then designed separately.

This step could, however, be solved using multiple methods for designing HENS problems. The model presented in Petterson [2] is only a suggested method for the network design problem, and not a necessity.

4.3.3. Benefits and obstacles

The gradual introduction of binary variables in the sequential match reduction approach enables the method to solve larger HENS problems. The method also evaluates the trade-off between utility and area throughout the entire process, which leads to a more economical solution.

All the problems in the sequential match reduction approach are linearized. The annual cost for a heat exchanger normally has an exponential coefficient for each unit as is given in Equation 4.9.

$$\text{Area Cost} = 145.63A^{0.6}\ \$/\text{year} \tag{4.9}$$

A linearized version of Equation 4.9. is presented in Equation 4.10, and the differences in the functions are illustrated in Figure 7.

$$\text{Area Cost} = 50 + 25A\ \$/\text{year} \tag{4.10}$$

The fact that this function has to be linearized is normally not crucial from a practical point of view. Pettersson [2] explains that the pricing policies vary between manufactures and that the driving force for cost reductions are the overall investment cost and not the individual unit area.

However, linearized cost functions leads to no economy of scale for the problem. The combination of linearized cost functions and user interaction in the selection of matches that should be subject to evaluation (i.e. binary variables are introduced to model the existence of these matches) prevents the match reduction step to guarantee an optimal design.

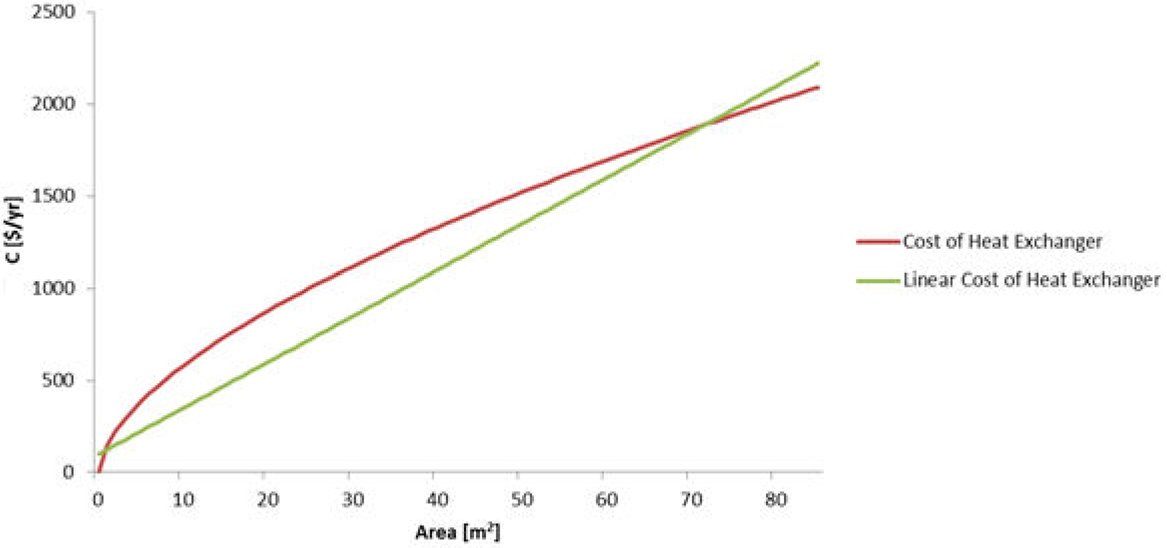


Figure 7 Area cost, *C*, as a function of area. Linearized and non-linear cost functions of Equations 4.9 and 4.10.

Figure 7 shows how the linearized area cost function differs from the exponential cost function. This difference in cost in the objective function may lead to larger utility usage for the linear method.

It is therefore important to select the appropriate cost factor so that the trade-off between area, utility consumption and unit cost is applicable to the non-linear area cost used in the total annual cost function for the final design.

4.4. The subsystem approach for HENS

This chapter presents the subsystem approach by Pouransari and Maréchal [14]. The subsystem approach is a method of reducing the computational work for MILPs in larger heat exchanger network synthesis problems by grouping of streams.

4.4.1. Methodology

The subsystem approach is a sequential optimization method with multiple steps (Figure 8), which groups process streams into subsystems and minimizes the number of connections between them, while finding a close-to-optimal network with respect to the total number of connections.

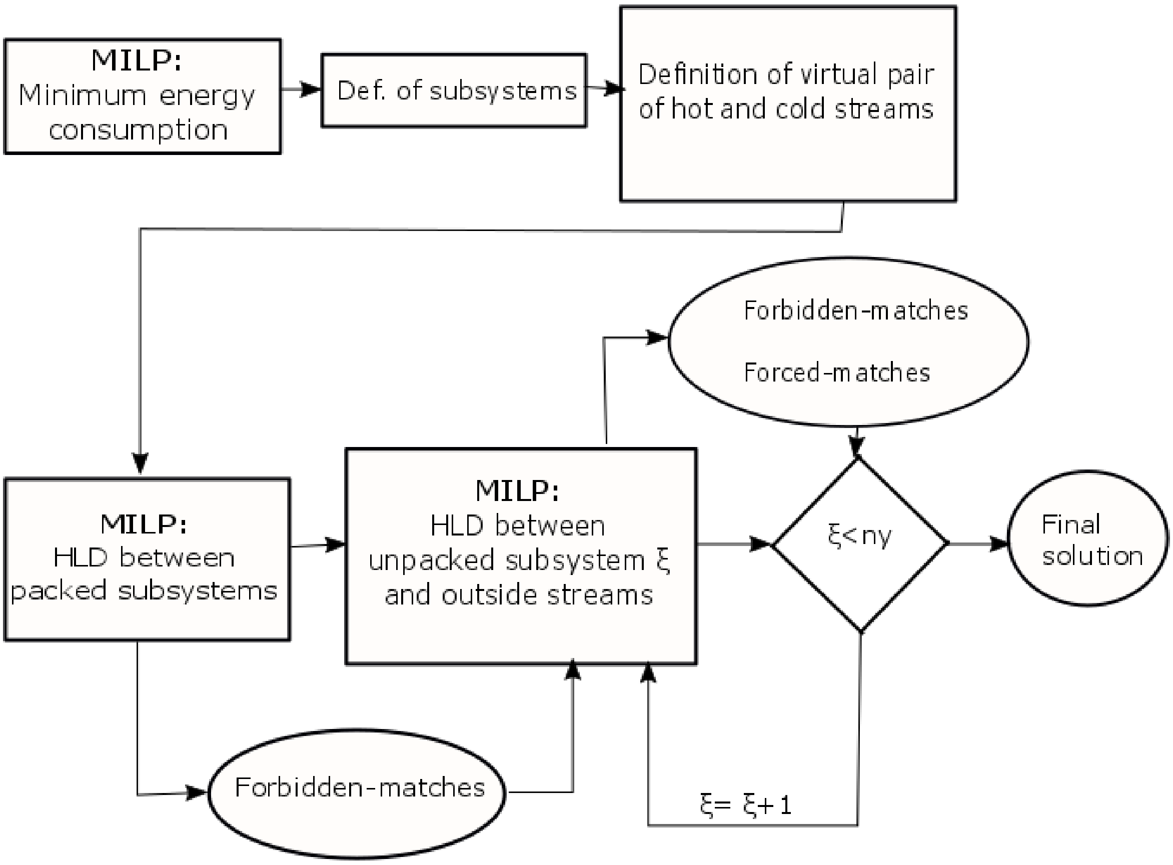

MILP:
Minimum energy consumption
Def. of subsystems
Definition of virtual pair of hot and cold streams
Forbidden-matches
Forced-matches
MILP:
HLD between packed subsystems
MILP:
HLD between unpacked subsystem ξ and outside streams
ξ<ny
Final solution
Forbidden-matches
ξ= ξ+1

Figure 8 Methodology for the subsystem approach by Pouransari and *Maréchal* [14].

Minimum Operating Cost

The subsystem approach uses an MILP to identify the total site minimum heating and cooling requirements, as well as the optimal flow rate of the utilities. The utilities and operating costs are then fixed for the remaining steps of the methodology. The MILP model was initially proposed by Papoulias and Grossmann [8] and later by Marèchal and Kalitventzeff [15].

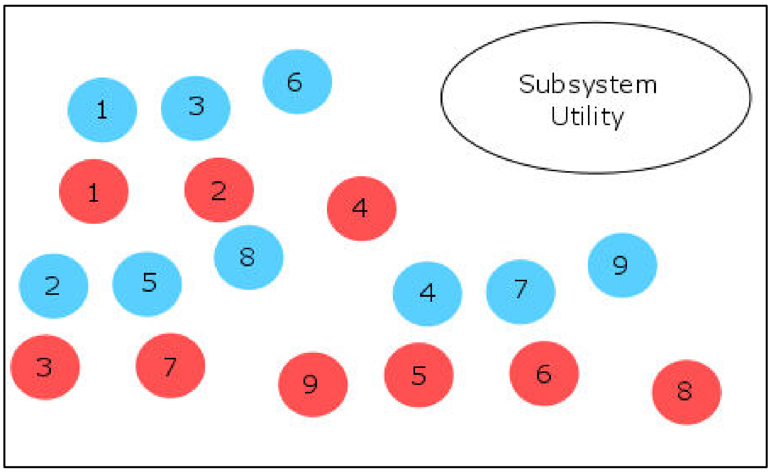


Figure 9 Calculating minimum operating cost.

Defining subsystem

The second step of the method is to divide the process streams into a number of subsystems. The user groups the streams based on the parameters that are important for the individual cases. This could be done based on location of streams in the plant layout as well as certain factors that would make a certain stream matching undesirable, such as process type or operating period.

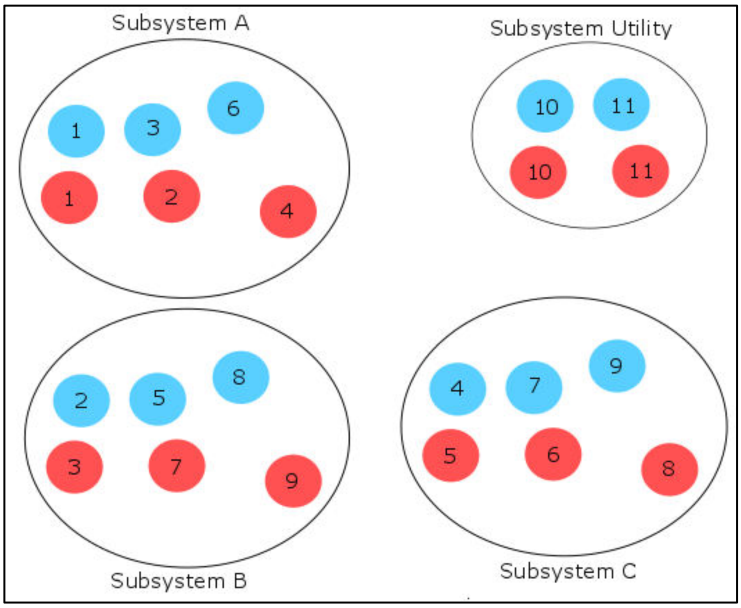


Figure 10 Defining subsystems.

Define virtual streams of subsystems

Once the subsystems are defined, the streams are packed and replaced by a virtual hot and cold stream with a non-linear T-H profile, which is constructed by solving the heat cascade between the streams of each subsystem.

Packaging of streams reduces the number of binary variables for the minimum units and HLD problem. A 10+10 even stream problem has a hundred binary variables, but if five hot and five cold streams are packed into a virtual stream, the number of streams are down to six hot and six cold with a total of thirty-six binary variables.

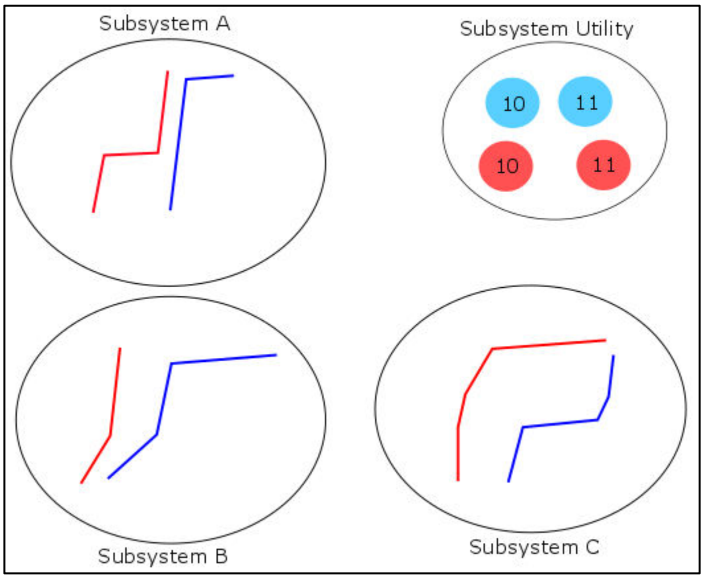


Figure 11 Packaging of streams in subsystems.

HLD model between virtual streams and utility

An MILP model for the packed subsystem and utilities is used to generate a HLD for the system. The model minimizes the number of connections between virtual and utility streams with the objective function in Equation 4.10, where ny is the total number of subsystems and nscφ and nshφ is the total number of cold and hot streams in one subsystem, φ.

$$\text{ObjFun} = \sum_{\varphi=1}^{ny} \sum_{j=1}^{nsc\varphi} \sum_{i=1}^{nsh\varphi} y_{ij\varphi} \tag{4.10}$$

After the HLD is generated for the virtual streams, internal connections between virtual streams in the same subsystem are added as a forced-match in the form of a constraint. This insures that the minimum number of connections is obtained. The binary values of the subsystems without connections are set to zero to represent a forbidden match in the next step of the process.

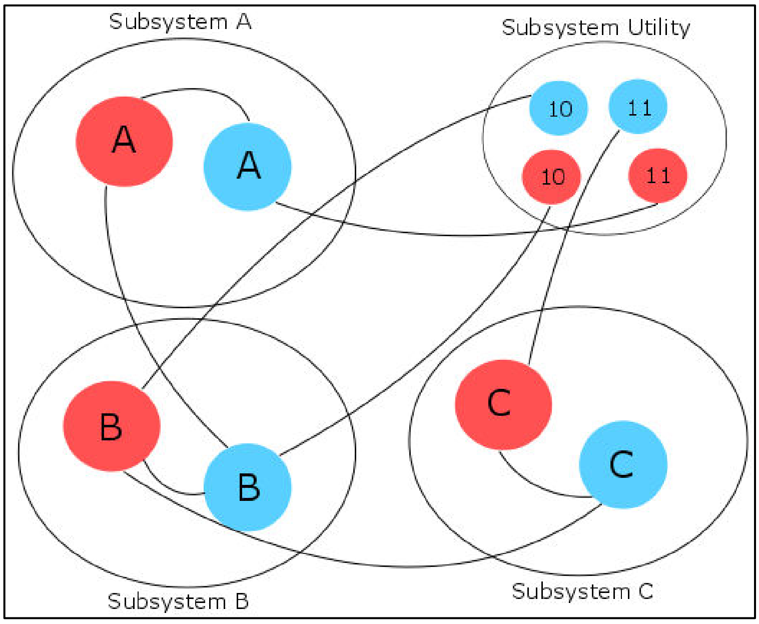


Figure 12 Stream match generator for the virtual streams.

Sequential unpacking of substystems

The forced and forbidden matches from the packed HLD model are transferred to this next step where the virtual streams/subsystems are unpacked (switched back to the original streams). The subsystems are unpacked sequentially, while new constraints are added to the problem after every optimization process.

The sequential unpacking of subsystems and addition of constraints reduces the number of binary variables by fixing several forced matches and forbidden matches for certain streams throughout the model. Once the final subsystems is unpacked the number of binary variables has been reduced from the initial number.

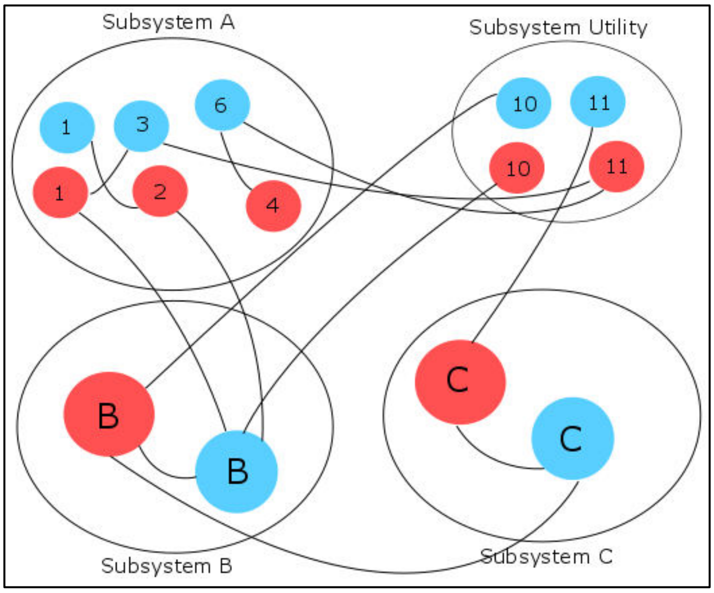


Figure 13 Sequential unpacking of subsystems.

Final design

The number of unpacked subsystems are checked, and if all the subsystems are unpacked, the HLD of the optimization process is used to generate a HEN manually or with the help of a NLP optimization model.

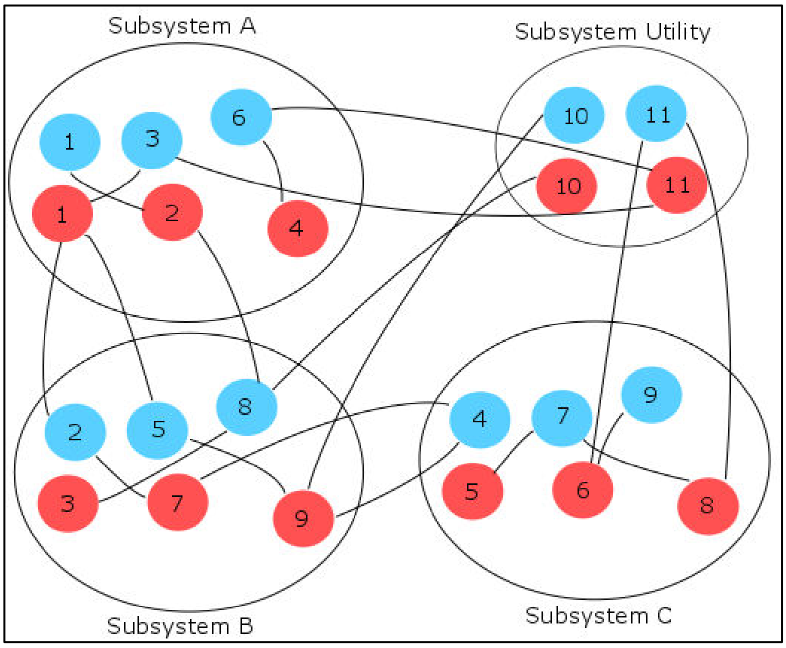


Figure 14 Final HLD for the subsystem approach.

4.4.2. Advantages and disadvantages

The subsystem approach is practical for industrial applications due to its ability to solve larger problems and provide numerous alternative designs for decision making by restricting certain matches before generating heat load distributions.

The method does not, however, guarantee the optimal solution due to the method being sequential and not accounting for the cost of area at every step, as well as the user interaction in the subsystem grouping process, which can result in exclusion of beneficial matches.

5. Similarities and differences

This chapter compares the subsystem approach by Pouransari and Maréchal [14] and the sequential match reduction approach by Pettersson [2] by considering similarities and differences.

5.1. Introduction

The sequential match reduction approach and the subsystem approach are designed to handle larger HENS problems by reducing the number of binary variables in the optimization process. The basic principles may be similar, but the methods are very different in their approach to reducing the number of binary variables.

5.2. Similarities

Both methods described in this report are sequential methods designed to handle larger MILP problems in process synthesis, which have been proven the bottleneck in heat exchanger network synthesis. None of the methods can guarantee "optimal" networks since they are both sequential in nature and involve user interaction in the decision making on beneficial matches.

The reduction of computational complexity in both methods is achieved by reducing the number of binary variables in the problem, and once all the binary variables are introduces, several of them have fixed values.

Both the sequential match reduction approach and the subsystem approach use grouping as a step in their processes to reduce the computational effort required to solve HENS problems.

5.3. Differences

Even though the subsystem approach and the sequential match reduction approach have the same purpose of reducing computational complexity of larger MILPs they are fairly different in their methodology.

Both methods introduce binary variables gradually throughout their processes. However, the sequential match reduction approach introduces binary variables for a few selected number of matches in the HLD from the LP formulation, while the subsystem approach introduces binary variables by creating subsystems with virtual streams and using binary variables to all matches between their respective streams.

The sequential match reduction approach assesses the trade-off between utility and area cost throughout the entire process, while the subsystem approach focuses on reducing the total number of units for the entire system until the capital cost of the network is introduced in the final NLP optimization model. This difference in objective functions between the two methods is likely to generate different "optimal" solutions.

The two methods use different area cost functions in their processes. The subsystem approach has a non-linear cost function that accounts for the economy of scale, while the sequential match reduction approach evaluates the trade-off between area and utility cost throughout the entire method with a linearized cost function, which may lead to a larger area than the optimal cost minimized solution.

The utility use for the subsystem approach is fixed in the model to achieve the minimum number of units, while the utilities can change in the sequential reduction approach to assess the trade-off between utility cost and capital cost.

Even though both methods use grouping as a way of reducing the computational effort of heat exchanger network synthesis, they use it in different steps. The subsystem approach focuses on the minimum utility problem and groups stream before solving the problem. The sequential match reduction approach groups matches in the reduced HLD and designs the different subsystems separately.

6. Case Studies

This chapter presents the three case studies chosen from literature to represent problems with different sizes, ranging from small (10 streams) to large problems (30+ streams).

6.1. Introduction

The problem data for all case studies are listed in the Tables 1-3. Problems with an even/ close to even number of hot and cold streams are chosen in order to study the performance of the two computational strategies.

6.1. Problem Data

Table 1 Problem data for Case Study 1 (Papoulias and Grosssmann [8])

Stream	T_{in} [°C]	T_{out} [°C]	mCp [kW/°C]	h [kW/m^2°C]
H1	433	366	8.79	1.7
H2	522	411	10.55	1.7
H3	544	422	12.56	1.7
H4	500	339	14.77	1.7
H5	472	339	17.73	1.7
C1	355	450	17.28	1.7
C2	366	478	13.90	1.7
C3	311	494	8.44	1.7
C4	333	433	7.62	1.7
C5	389	495	6.08	1.7
HU	509	509		1.7
CU	311	355		3.41

Cost of Heat Exchangers [\$/yr] = $145.63A^{0.6}$
Linear Cost of Heat Exchangers [\$/yr] = 50+20A
Cost of Cooling Utility = 18.12 [\$/(kW*yr)]
Cost of Heating Utility = 37.64 [\$/(kW*yr)]

Table 2 Problem data for Case Study 2 (adapted from Sorsak and Kravanja [16])

Stream	T_{in} [°C]	T_{out} [°C]	mCp [kW/°C]	h [kW/m^2°C]
H1	576	437	23.1	0.06
H2	599	399	15.22	0.06
H3	530	382	15.15	0.06
H4	449	237	14.76	0.06
H5	368	177	10.7	0.06
H6	121	114	149.6	1.00
H7	202	185	258.2	1.00
H8	185	113	8.38	1.00
H9	140	120	59.89	1.00
H10	69	66	165.79	1.00
H11	120	68	8.74	1.00
H12	67	35	7.62	1.00
H13	1034.5	576	21.3	0.06
C1	123	210	10.61	0.06
C2	20	210	6.65	1.20
C3	156	157	3291	2.00
C4	20	182	26.63	1.20
C5	182	318	31.19	1.20
C6	318	320	4011.83	2.00
C7	322	923.78	17.6	0.06
CU	9	17		1.00
HU	927	927		5.00

Cost of Heat Exchangers [\$/yr] = $4000+500\ A^{0.83}$
Linear Cost of Heat Exchangers [\$/yr] = 8000+180A
Cost of Cooling Utility = 25 [\$/(kW*yr)]
Cost of Heating Utility = 250 [\$/(kW*yr)]

Table 3 Problem data for Case Study 3 (Pettersson [2]).

Stream	Tin[°C]	Tout[°C]	mCp[kW/K]	h[kW/m²K]
H1	180	75	30	2.00
H2	280	120	15	2.50
H3	180	75	30	2.00
H4	140	45	30	2.00
H5	220	120	25	1.50
H6	180	55	10	2.00
H7	170	45	30	2.00
H8	180	50	30	2.00
H9	280	90	15	2.00
H10	180	60	30	2.00
H11	120	45	30	2.00
H12	220	120	25	2.00
H13	180	55	10	2.00
H14	140	45	20	2.00
H15	140	60	70	2.00
H16	220	50	15	2.50
H17	220	60	10	2.50
H18	150	70	20	2.00
H19	140	80	70	2.00
H20	220	50	35	2.00
H21	180	60	10	2.00
H22	150	45	20	2.50
C1	40	230	20	1.50
C2	120	260	35	1.00
C3	40	190	35	1.50
C4	50	190	30	2.00
C5	50	250	60	2.00
C6	40	150	20	2.00
C7	40	150	20	2.00
C8	120	210	35	2.50
C9	40	130	35	2.50
C10	60	120	30	2.50
C11	50	150	10	3.00
C12	40	130	20	1.00
C13	120	160	35	1.00
C14	40	90	35	1.75
C15	50	90	30	1.50
C16	50	150	30	2.00
C17	30	150	50	2.00
Hot utility	325	325	-	1.00
Cold utility	25	40	-	2.00

Cost of Heat Exchangers [$/yr] $= 8000+800A^{0.83}$

Linear Cost of Heat Exchangers [$/yr] $= 15000+250A$

Cost of Cooling Utility = 10 [$/(kW*yr)]

Cost of Heating Utility = 70 [$/(kW*yr)]

7. Results & Discussion

The problems are solved using the original SeqHENS and the adapted SeqHENS software, with the implemented sequential match reduction approach by Pettersson [2]. The quantitative and qualitative aspects of the final network designs will be compared and discussed for all case studies.

7.1. Adaptation of the sequential match reduction approach

After an extensive look at the sequential match reduction approach and its methodology, as well as programming the first three steps in GAMS (Appendix B), it is clear that the formulations cannot be adapted to solve the combinatorial problems in the minimum unit problem that occurs for larger problems in SeqHENS.

Unlike the Sequential Framework, the sequential match reduction approach focuses on assessing the trade-off between area and operating cost throughout the entire approach. Therefore, the approach needs to be altered for the purposes of comparison..

After revising the objective with my supervisors, the original Sequential Framework and an adapted Sequential Framework, will be used to optimize 3 case studies. The solutions of the two frameworks will be evaluated on the basis of their quantitative and qualitative aspects.

The LP-formulation and stream match generator MILP from the sequential match reduction approach has, for the purpose of comparison, been implemented into the Sequential Framework and SeqHENS. For this adapted framework the minim utility LP and the design NLP from SeqHENS remain.

The utility consumption in the stream match reduction step is fixed in the inner loop, and the second LP-formulation therefore minimize the total area instead of the trade-off between operating and capital cost.

7.1.1. Loops in the framework

The four loops in the Sequential Framework exist to evaluate multiple networks with different parameters (HLD, U, EMAT, HRAT), which enables the user to generate a network as close to possible to the optimal network design. The original problem/objective statement was to ease the computational problems in the minimum unit MILP in SeqHENS.

However, the Sequential Framework starts the optimization at the minimum possible number of units and adds units gradually, while the sequential match reduction approach does not take in to account unit cost in the LP formulation. Therefore the sequential match reduction approach will generate an initial HLD with a large number of units and then challenge the units by introducing binary variables to some of the matches. This led to the implementation of the entire stream match generator step in the sequential match reduction approach into the loops of the Sequential Framework (Figure 15). The EMAT loop is completely removed, due to the sequential match reduction not using EMAT. The unit loop is altered from the original addition of units in the Sequential Framework, to the reduction of matches in the sequential match reduction approach.

The remaining loops are the HRAT and the HLD loop. The HLD loop is altered from the original framework. Instead of demanding an alternative HLD, the loop introduces binary variables to a different set of matches to prevent the user from challenging and removing beneficial units in the final HLD. Figure 15 presents the adapted Sequential Framework.

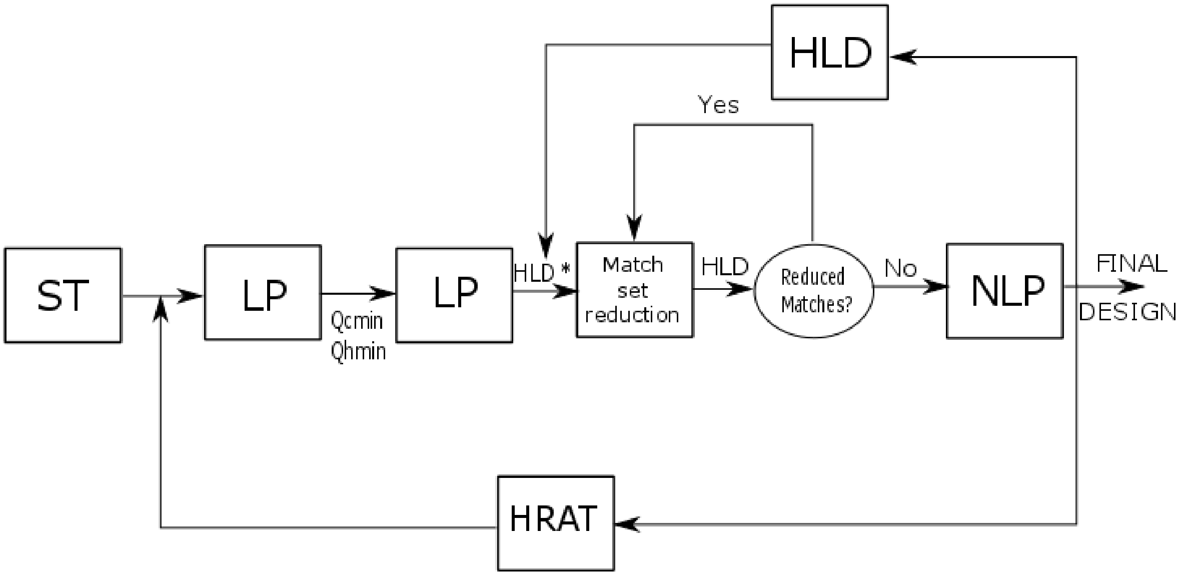


Figure 15 Adapted Sequential Framework with the implemented sequential match reduction LP and MILP for HLD generation

7.2. Case Study 1

Case Study 1 is a relatively small problem with 10 streams (5 hot and 5 cold). This case study was previously used in my Specialization Project to compare the results from the original Sequential Framework to several other HENS methods used in Escobar and Trierweiler [17].

This is such a small problem that the MILPs should experience no computational explosions due to the problem size, and is a good problem for evaluating the solution quality of the two formulations.

7.2.1. Minimum utility calculation

The problem data for Case Study 1 is added to the excel add-in file to calculate the minimum utility consumption for a set HRAT in the LP formulation. Table 4 presents the results from the LP formulation.

Table 4 Minimum utility consumption for fixed HRAT in Case Study 1

HRAT	10 °C
Qh_{min}	17280 kW
Qc_{min}	25000 kW

The HRAT and its minimum utility will be fixed for both formulation assessed in this thesis.

7.2.2. Sequential Framework optimization

The results from the original Sequential Framework are presented in the following subchapters.

Parameter setting

The absolute minimum number of units (Umin = 10) is located in the first MILP of SeqHENS. Previous experiences of using SeqHENS have showed that the “optimal” network is frequently located in the interval $U_{min} \leq U \leq U_{min}+2$. The unit loop will therefore stop when two additional units have been added to the minimum number of units.

To prevent the user from not locating the “optimal” network, the EMAT loop will use 1/8, ¼ and 3/8 of the fixed HRAT. The smaller EMATs have lower driving forces, but may generate less complex design structures, which may be the “optimal” design for the problem.

The number of B&B leaf nodes with silmar values close to the optimal minimized solution makes it necessary to evaluate two different HLDs for each fixed U, EMAT and HRAT.

Optimization results

After all combinations of parameters are evaluated, the HLDs generated in the stream match generator are designed and the total annual cost is calculated. Table 5 presents the TAC for all possible combinations. The operating cost is fixed due to the fixed HRAT-loop.

Table 5 Total annual cost for Case Study 1.

Sim #	U	EMAT	HLD	Total Area	Capital Cost	Operating Cost	TAC
		[°C]		[m²]	[$/yr]	[$/yr]	[$/yr]
1	**10**	**1.25**	**A**	**236.2**	**9313**	**34046**	**43359**
2	**10**	**1.25**	**B**	**224.5**	**9064**	**34046**	**43110**
3	11	1.25	A	5653	59540	34046	93586
4	11	1.25	B	5503.2	58571	34046	92617
5	12	1.25	A	5765.2	62229	34046	96275
6	12	1.25	B		Does not generate a HLD		
7	**10**	**2.5**	**A**	**229.3**	**8947**	**34046**	**42993**
8	10	2.5	B	2755.3	39795	34046	73841
9	11	2.5	A	234	9722	34046	43768
10	11	2.5	B	250.7	9769	34046	43815
11	12	2.5	A	5217.9	60673	34046	94719
12	12	2.5	B		Does not generate a HLD		
13	10	3.75	A	243.1	9590	34046	43636
14	**10**	**3.75**	**B**	**229.3**	**8947**	**34046**	**42993**
15	11	3.75	A	244.5	10011	34046	44057
16	11	3.75	B	236.2	9533	34046	43579
17	12	3.75	A		Does not generate a HLD		
18	12	3.75	B		Does not generate a HLD		

The four networks outlined in Table 5 were all assessed on their qualitative and quantitative aspects, due to their low TAC. Simulation number 7 and 14 have the same HLD and therefore have identical network designs. Simulation 5 and 7 generate the network with the lowest TAC and it is presented in Figure 16.

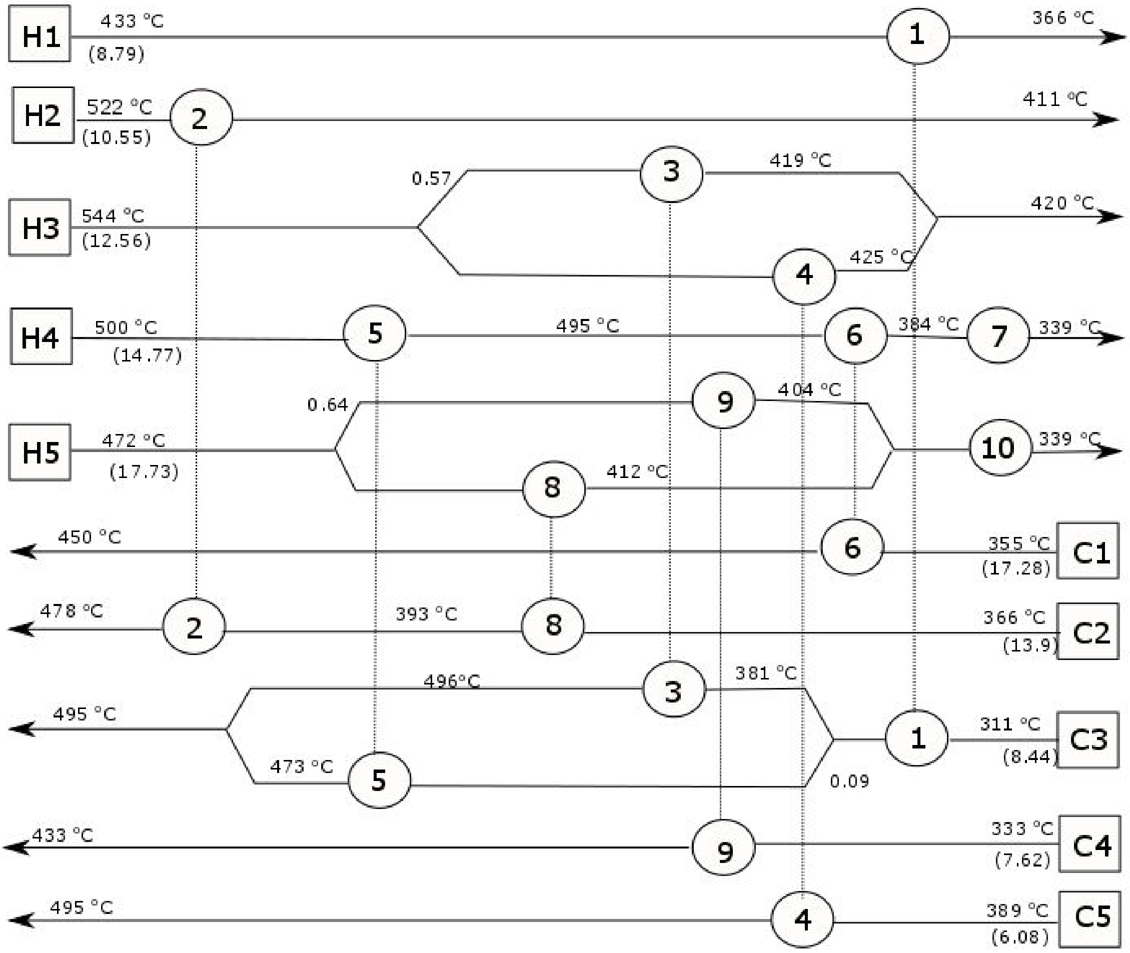


Figure 16 Network for the simulation with the lowest TAC in Case Study 1.

The network presented in Figure 16 has the lowest TAC and has a simple structure with only three stream splits. However, when reducing the EMAT to 1.25 the network generated has one less split, which reduces the complexity of the network. The network for simulation 2 is presented in Figure 17.

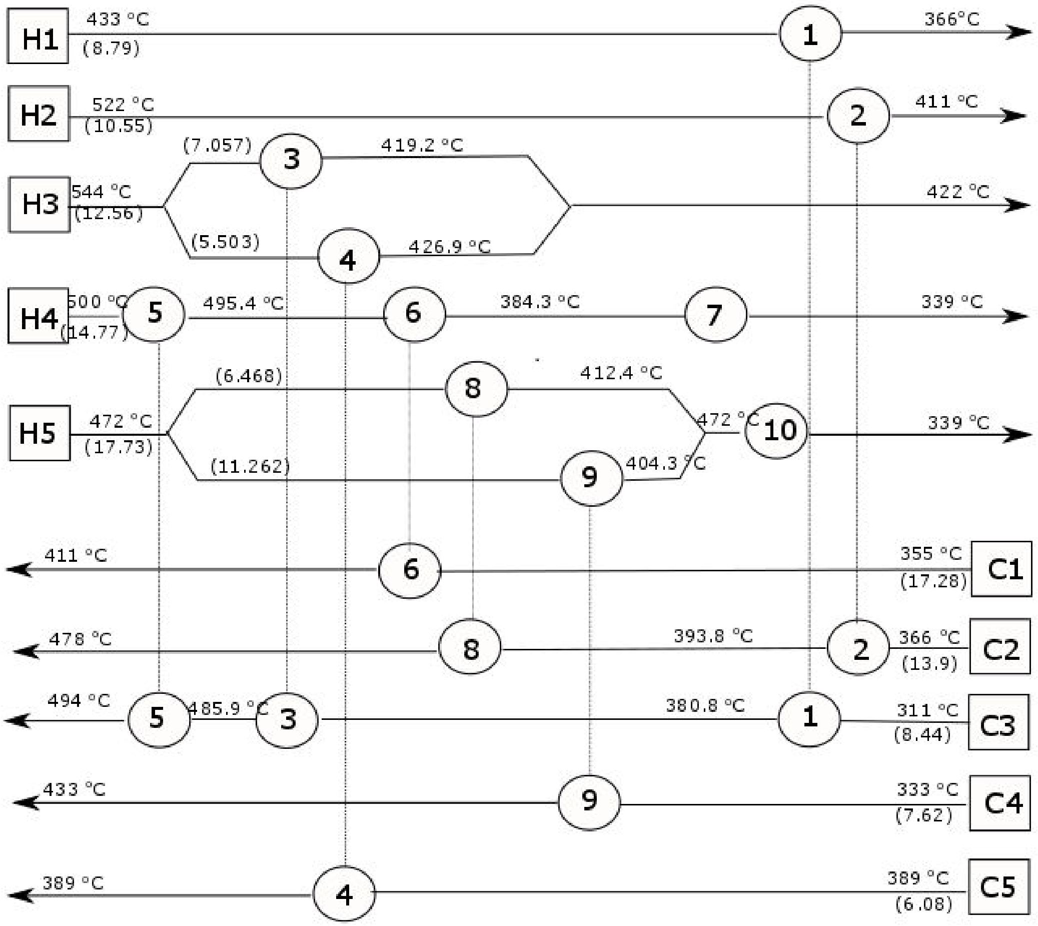


Figure 17 Network for simulation number 2 from the SeqHENS.

The total annual cost of this network is similar to the one in simulation 7, with the TAC only being 0.27 percent higher. The networks presented in Figure 16 and 17 have different benefits. The benefit of having one less stream split has to be evaluated when selecting the "optimal" network. When only considering the simplified cost function for the problem data, simulation 7 is the "optimal" network. However, the qualitative aspects of simulation 2 is beneficial from a complexity point of view and could be the network with the lowest TAC for a more detailed cost function where stream split cost is included. Due to the TAC of these two networks being similar, the network with only two splits would be the network closest to the optimal solution.

The final network outlined in Table 5 is the first HLD generated for the initial conditions where U = 10 and EMAT = 1.25. The network has a higher TAC and a more complex structure than the alternative HLDs for these conditions. This shows the importance of evaluating two HLD for every condition. The network is presented in Appendix C.

7.2.3. Adapted Sequential Framework optimization

Case Study 1 is such a small problem that binary variables can be introduced to all possible matches without a significal increase in computational time. The introduction of binary variables for all matches prevents the user from challenging matches that could be beneficial for the final HLD. Therefore, the LP minimizing total area is not a necessary step in the optimization process.

The problem data and linear area cost function are implemented in the MILP formulation by Pettersson [2] (Appendix A2), which minimizes total area and unit cost. The approximated linear cost function is presented in Equation 7.1. The addition of a small unit cost is to prevent networks with a complicated structure with an excessive amount of units and stream splits. The generated HLD is presented in Table 6.

$$\text{Linear Cost of Heat Exchangers } [\$/\text{yr}] = 50 + 20A \qquad (7.1)$$

Table 6 Adapted Sequential Framework HLD for Case Study 1

	C1	C2	C3	C4	C5	CW
H1				401.2		187.7
H2		810.3		360.8		
H3			887.8		644.5	
H4	733.1	746.5				898.3
H5	908.5		656.7			792.9

The HLD is then transferred to the NLP from SeqHENS, and the “optimal” network is presented in Figure 18. The cost data for the network is presented in Table 7.

Table 7 Cost data for network generated by the adapted Sequential Framework Case Study 1.

U	Total Area	Capital cost	Operating cost	TAC
	[m²]	[$/yr]	[$/yr]	[$/yr]
11	230.63	100071	34046	134117

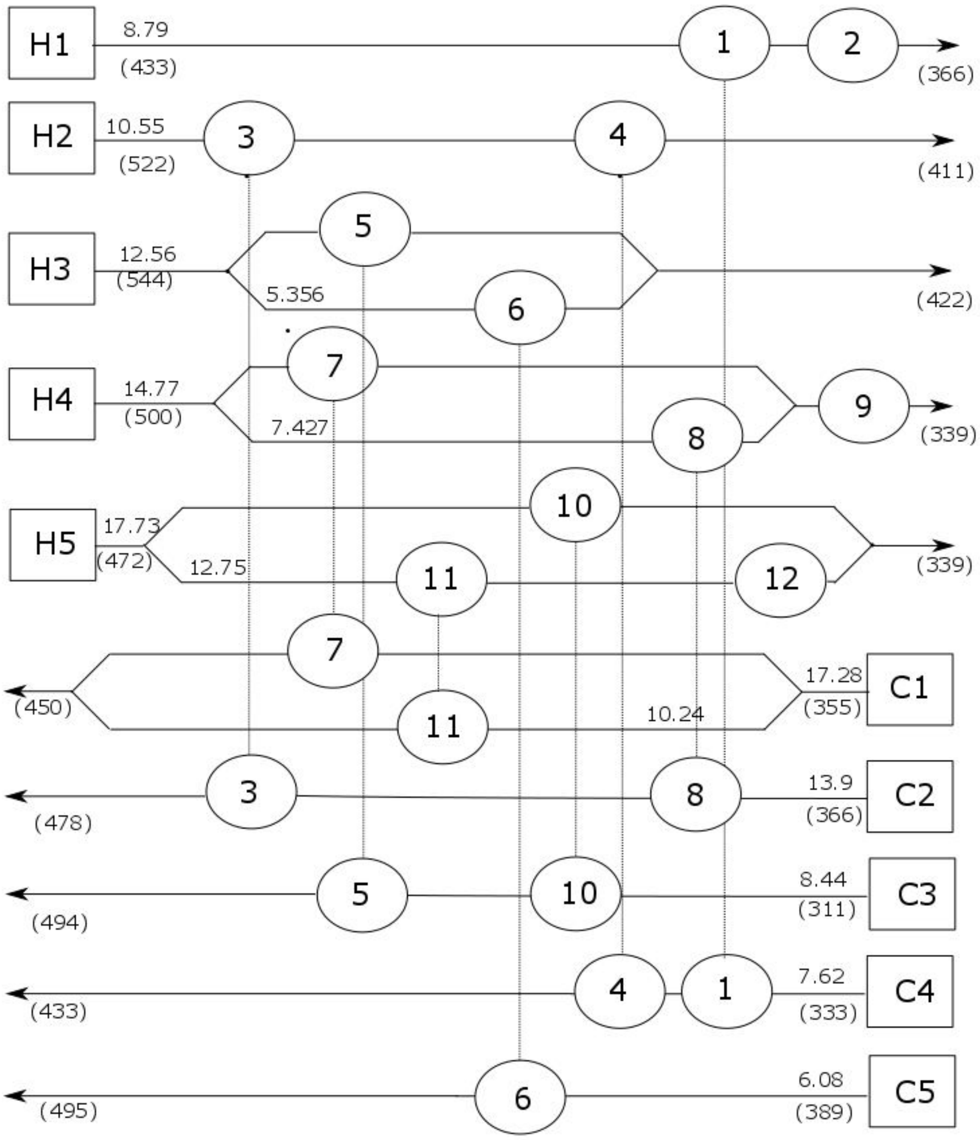


Figure 18 Network for the adapted Sequential Framework Case Study 1.

7.2.4. Comparison

The three "optimal" networks presented in Figure 16-18 will serve as the basis for this comparison of the two formulations. On a quantitative basis, the networks generated by the original Sequential Framework have a lower TAC than the adapted framework, while also having a simpler structure. The "optimal" network from the original formulation has two less stream splits than the adapted formulation; this would increase the cost difference in a real life problem. The cost functions in the case studies solved in this master thesis are simplifications of the actual cost of a network. Stream split controllers are expensive and increases the complexity and cost of a network.

The network from SeqHENS also has two less units than the adapted method, which is favourable from a operability point of view. However, the network from the adapted method contains one more process-utility unit, which will increase the controllability of the network

The networks generated by SeqHENS are the better options for this case study, and will be the ones closest to the optimal network design.

7.3. Case Study 2

Case Study 2 is a medium sized problem from Soršak and Kravanja [16],with 20 process streams (13 hot and 7 cold). This case study was previously solved during my Project Work, and caused some problems for SeqHENS due to some restrictive matches, caused by the heat capacity flow rates of the problem..

7.3.1. Minimum utility calculation

The problem data is implemented in the minimum utility LP with a fixed HRAT. For this fixed heat recovery, the minimum utility consumption is presented in Table 8.

Table 8 Utility consumption for fixed HRAT in Case Study 2.

HRAT	20 °C
Qh_{min}	1117.988 kW
Qc_{min}	338.95 kW

7.3.2. Sequential Framework

The results from the original Sequential Framework are presented in the following subsections.

Parameter setting

The heuristic rules used to establish the parameters intervals in case study 1are used in for this case. The minimum number of units is located in the first MILP in SeqHENS to be 21. The uit loop will therefore be adjusted from 21 to 23. To prevent the user from not locating the "optimal" network, the EMAT loop will use 1/8, ¼ and 3/8 of the fixed HRAT, and two HLDs for all combinations of parameters will be assessed.

Optimization results

After all the simulations are evaluated, the HLDs generated in the stream match generator are designed and the total annual cost is calculated. Table 9 presents the TAC for all possible combinations. The operating cost is fixed due to the fixed HRAT-loop.

Table 9 Total annual cost for Case Study 2.

Sim #	U	EMAT	HLD	Total Area	Capital Cost	Operating Cost	TAC
		[°C]		[m^2]	[$/yr]	[$/yr]	[$/yr]
1	21	2.5	A	4054.1	819558	287973	1107531
2	21	2.5	B	4251.7	849895	287973	1137868
3	**22**	**2.5**	**A**	**3977.2**	**818258**	**287973**	**1106231**
4	22	2.5	B		Does not generate a HLD		
5	23	2.5	A	4077.9	837118	287973	1125091
6	23	2.5	B		Does not generate a HLD		
7	21	5	A	4128.3	830215	287973	1118188
8	21	5	B	4145.7	834798	287973	1122771
9	22	5	A	4069.4	828830	287973	1116803
10	22	5	B	4493.2	905481	287973	1193454
11	23	5	A		Does not generate a HLD		
12	23	5	B		Does not generate a HLD		
13	21	7.5	A		Does not generate a HLD		
14	21	7.5	B		Does not generate a HLD		
15	22	7.5	A		Does not generate a HLD		
16	22	7.5	B		Does not generate a HLD		
17	23	7.5	A		Does not generate a HLD		
18	23	7.5	B		Does not generate a HLD		

The large driving forces demanded when EMAT = 7.5, leads to SeqHENS not being able to generate a HLD for these conditions.

Figure 19 presents the network outlined in Table 9. The network has the lowest TAC of all the simulations, as has the lowest number of stream splits. The network structure is relatively simple with four splits in total.

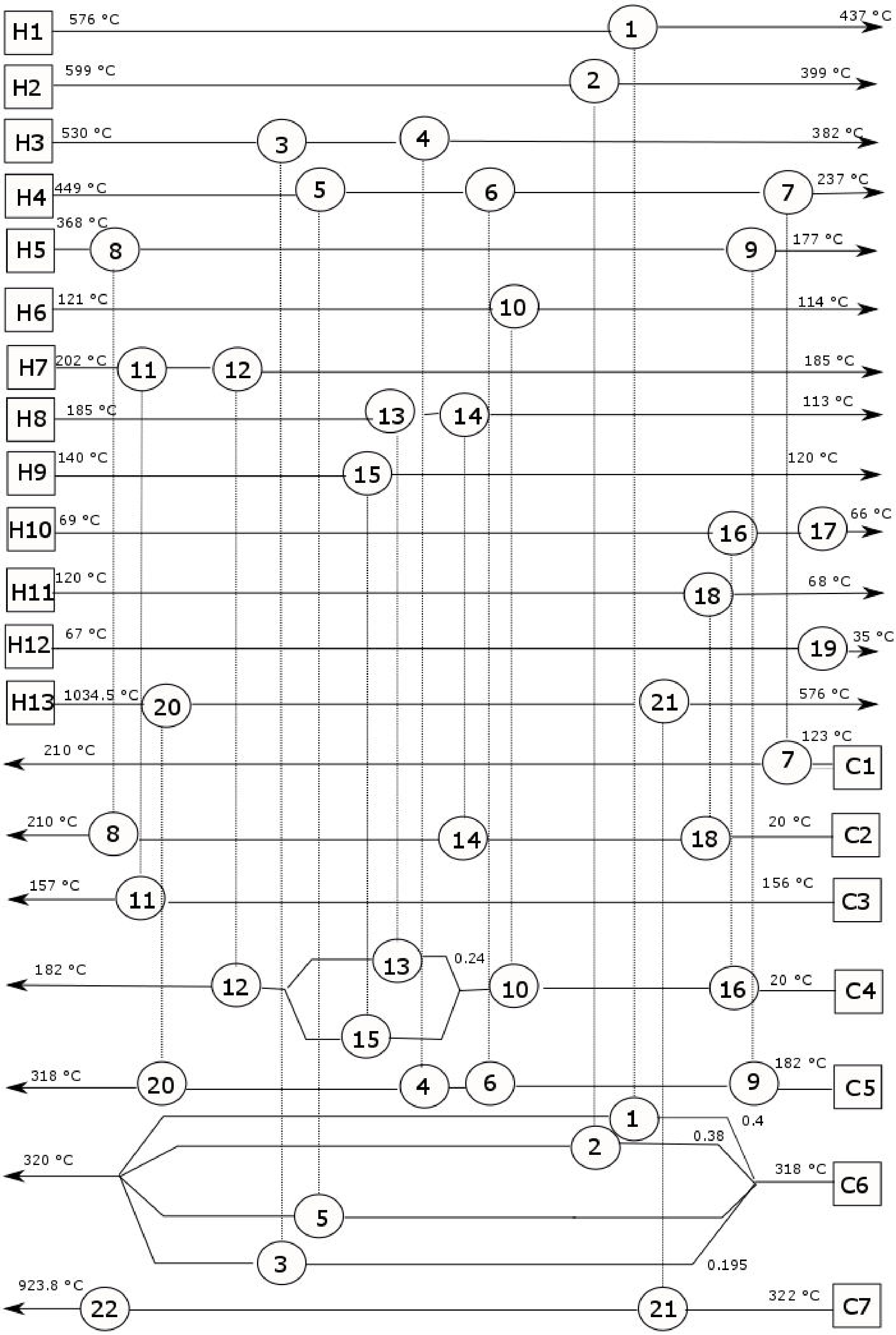


Figure 19 Network structure for Case Study 2 optimized by SeqHENS.

Complications in the design procedure

The design NLP in SeqHENS has trouble optimizing networks with 5 or more units per stream. This was a problem for the HLD generated by the stream match generator for this case study. With help from my co-supervisor the final network was designed by the NLP, and is the optimized network for this HLD.

7.3.3. Adapted Sequential Framework

To prevent the minimizing area the user from challenging matches that could be beneficial for the final HLD, binary variables are introduced for all matches in the MILP. The LP minimizing total area is therefore not performed for this case study.

The problem data and linear area cost function are implemented in the stream match generator (Appendix B.2), which minimizes total area and unit cost. The approximated linear cost function is presented in Equation 7.2. The generated HLD is presented in Table 10.

$$\text{Linear Cost of Heat Exchangers [\$/yr]} = 8000 + 180A \tag{7.2}$$

Table 10 «Optimal» HLD from the adapted Sequential Framework.

	C1	C2	C3	C4	C5	C6	C7	CW
H1						3210.9		
H2						3044		
H3					766.15	1476.05		
H4		300.77			2828.35			
H5	923.07			473.29	647.34			
H6				1047.2				
H7			3291	1098.4				
H8		603.36						
H9				1197.8				
H10				497.37				
H11		359.37						95.11
H12								243.84
H13						292.71	9473.34	
ST							1117.99	

The HLD duty and heat transfer coefficients are implemented in the NLP Excel add-in file. However, the 5 unit cold stream C4, causes the NLP from not being able to locate the "optimal" network. This same restriction occurred for the HLD generated by the original framework. The network is therefore designed by hand, which does not guarantee the HLD with the lowest TAC. The network for the adapted Sequential Framework is presented in Figure 20.

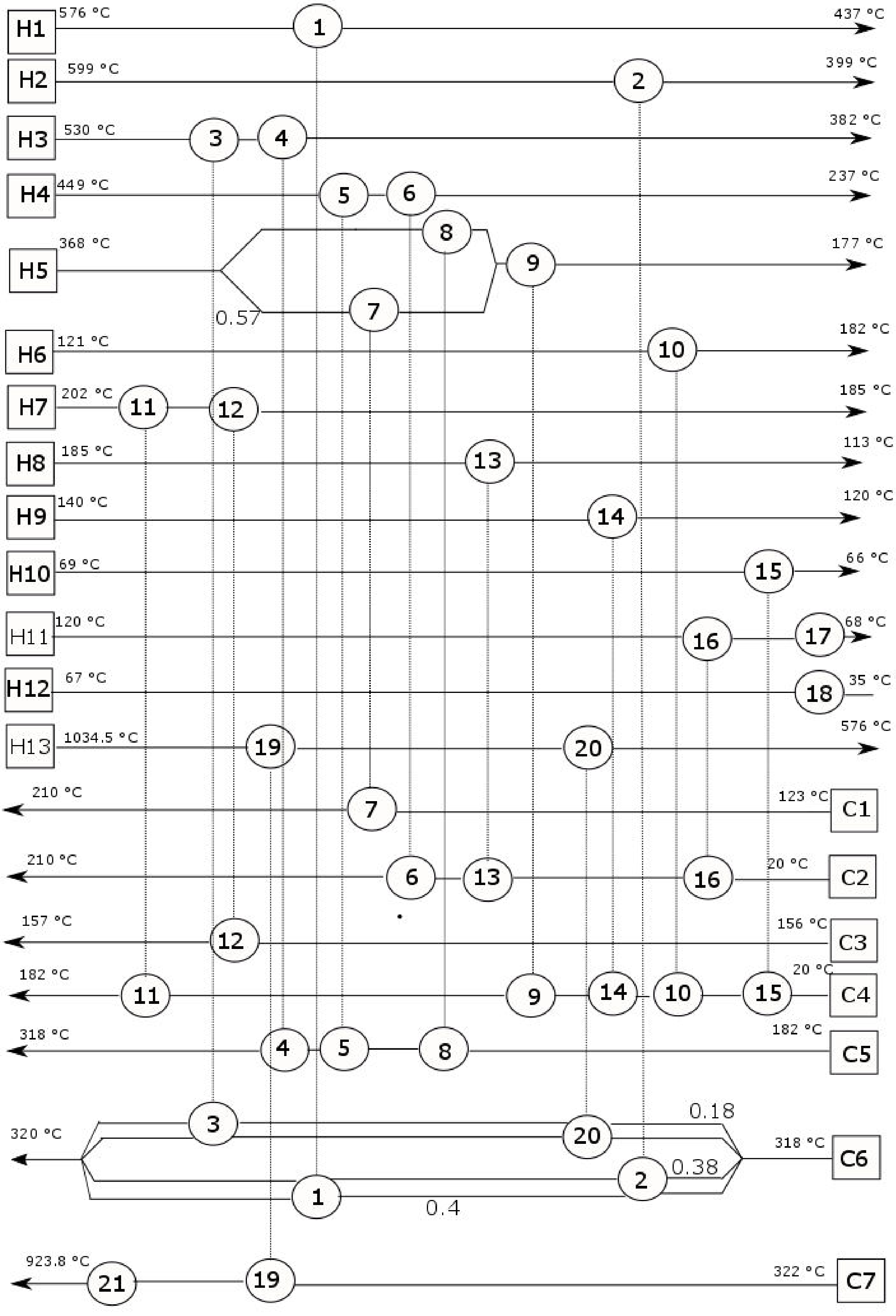


Figure 20 Network structure for Case Study 2 adapted Sequential Framework.

The network, presented in Figure 20, has a fairly simple structure with 4 stream splits and 21 units. The cost data for this network is presented in Table 11.

Table 11 Cost data for the adapted Sequential Framework Case Study 2

U	Total Area [m²]	Capital cost [$/yr]	Operating cost [$/yr]	TAC [$/yr]
21	5238.644	1016231.22	287973	1304204.22

7.3.4 Comparison

The networks presented in Figure 19 and 20 both have 4 stream splits. The adapted network generates a HLD with one less unit compared to the original framework, which is beneficial from an operability point of view. The TAC, however, for the adapted framework is 17 % higher than for the original framework. The driving forces are lower in the network presented in Figure 20, which causes this large cost difference.

The networks structure similarity leads to a conclusion based purely on TAC, where the original framework outperforms the adapted network.

Since the network generations are done by hand, there could be better solutions present for the HLDs designed in this thesis.

7.4. Case Study 3

Case Study 3 is considered a large problem with 39 process streams (22 hot streams and 17 cold streams) from Pettersson [2].

7.4.1. Minimum utility consumption

The problem data is implemented in the minimum utility LP with a HRAT equal to 10 °C. For this fixed heat recovery, the minimum utility consumption is presented in Table 12.

Table 12 Utility consumption for fixed HRAT in Case Study 3.

HRAT	10 °C
Qhmin	4450 kW
Qcmin	7750 kW

7.4.2. Sequential Framework

SeqHENS has previously had trouble solving problems of this size, due to the combinatorial explosions that occur in the software's MILPs. However, the minimum units MILP locates U_{min} to be 37. The combinatorial complexity of the stream match generator is, however, too difficult to solve with the original SeqHENS software.

7.4.3 Adapted Sequential Framework

Originally, a problem of this size was thought of as being too large to solve with a MILP, and the idea of introducing binary variables to a selected group of matches was planned. However, the adapted framework generates an "optimal" HLD in the MILP in 17 minutes, which is considered unusual for problems with this number of streams. Other problems this size may be harder to solve, and will require the introduction of binary variables sequentially to the HLD generated by the LP in the adapted framework.

The minimum utility from the fixed HRAT is implemented in the MILP in the adapted Sequential Framework, and the generated "optimal" HLD is presented in Appendix D.

The network design problem that occurs for Case Study 2, will be present in the HLD for this problem. Multiple streams have more than 5 units, which is problematic to solve in the NLP in SeqHENS. The main problem with the design of a network this size is the limitation of number of streams. The Excel add-in file that transfers the HLD and stream data to the NLP has a constraint restricting the number of process streams and utility units given in Equation 7.1., where I is the number of hot process streams, J is the number of cold process streams and M is the number of utility units.

$$M + (I+J) \leq 40 \qquad (7.1)$$

The HLD generated form the adapted Sequential Framework exceeds the constraint in Equation 7.1. An attempt of implementing the grouping formulation (Appendix A.3) from the stream match reduction approach was made, but due to the binary matches *y(i,j)* being a variable in the formulation, the combinations of the two formulations was unsuccessful.

The HLD generated could therefore not be optimized in the frameworks NLP, but could be designed by hand, as for Case Study 2. Due to the original frameworks inability to create a network for comparison and the complexity of designing a network this size by hand, the network design was not performed.

8. Conclusion

The implementation of an alternative stream match generator in the Sequential Framework was successful. After altering the original objective of this master thesis to focus on the complete design of networks, instead of the minimum unit subproblem, the stream match generator from the sequential match reduction approach was implemented into the Sequential Framework.

After evaluating the results from the optimization procedure for the 3 case studies selected, the general impression is that the implementation of the sequential match reduction approach to the Sequential Framework led to network with less favourable qualitative and quantitative aspects. For two of the case studies evaluated in this thesis the "optimal" networks generated by the adapted framework had a higher total annual cost, with either similar or more complex structure.

The removal of loops and replacing of the original stream match generators led to an approach that focuses on generating a network with the lowest TAC, instead of evaluating different networks that could have a higher TAC, but a less complex structure. The main benefit from the Sequential Framework is the focus on evaluating multiple networks with different fixed parameters. This enables the user to evaluate networks on both their quantitative and qualitative performances. The adapted framework, on the other side, does not take into account how the complexity influences the selection of the "optimal" network.

However, the adaptation of the Sequential Framework led to SeqHENS being able to solve larger problems, which was the purpose of this thesis. The adapted framework was outperformed for the 2 smaller case studies in this thesis; however, the fact that it can solve problems the original framework cannot, proves that the adapted approach could be useful for lager problems. The solution quality of the two approaches should be evaluated on a larger set of problems for further comparison.

8.1 Further works

The implementation of the sequential match reduction approach has led to SeqHENS being able to solve larger problems. However, the solution quality for larger problem have yet to be established, due to the inability to design problems with a certain number of streams or HLDs with 5 or more units per stream.

To further evaluate and compare the results generated by the original and the adapted framework it would be necessary to increase the capacity of the design NLP and the Excel add-in file used to transfer data into GAMS.

References

1. Linnhoff, B. and J.R. Flower, *Synthesis of heat exchanger networks: I. Systematic generation of energy optimal networks.* AIChE Journal, 1978. **24**(4): p. 633-642.
2. Petterson, F., *Synthesis of large-scale heat exchanger networks using a sequential match reduction approach.* Computers and Chemical Engineering, 2005. **29**: p. 993-1007.
3. Furman, K. and N. Sahinidis, *A critical review and annotated bibliography for heat exchanger network synthesis in the 20th century*, in *Ind. Eng. Chem. Res.* 2002. p. 2335-2370.
4. Chen, J.J.J., *Comments on improvements on a replacement for the logarithmic mean*. 1987. p. 2488-2489.
5. Gundersen, T., S. Duvold, and A. Hashemiahmady, *''An extended vertical MILP model for heat exchanger network synthesis''.* Comput. Chem. Eng., 1996. **20**: p. S97-S102.
6. Anantharaman, R., et al., *The sequential framework for heat exchanger network synthesis-The minimum number of units sub-problem.* Computers and Chemical Engineering, 2010. **34**(11): p. 1822-1830.
7. Gundersen, T., P. Traedal, and A. Hashemiahmady, *Improved sequential strategy for the synthesis of near-optimal heat exchanger networks.* Comput. Chem. Eng., 1997. **21**: p. S59-S64.
8. Papoulias, S.A. and I.E. Grossmann, *A structural optimization approach in process synthesis—II: Heat recovery networks.* Computers and Chemical Engineering, 1983. **7**(6): p. 707-721.
9. Cerda, J. and A.W. Westerburg, *Synthesizing heat exchanger networks having restricted stream/stream matches using transportation problem formulations.* Chemical Engineering Science, 1983. **38**(10): p. 1723-1740.
10. Floudas, C.A., A.R. Ciric, and I.E. Grossmann, *Automatic synthesis of optimum heat exchanger network configurations.* AIChE Journal, 1986. **32**(2): p. 276-290.
11. Land, A.H. and A.G. Doig, *An automatic method of solving discrete programming problems.* Econometrica : journal of the Econometric Society, an internat. society for the advancement of economic theory in its relation to statistics and mathematics, 1960. **28**(3): p. 497-520.
12. Chen, Y., I.E. Grossmann, and D.C. Miller, *Computational strategies for large-scale MILP transshipment models for heat exchanger network synthesis.* Computers and Chemical Engineering, 2015. **82**: p. 68-83.
13. Baliban, R.C., J.A. Elia, and C.A. Floudas, *Toward novel hybrid biomass, coal, and natural gas processes for satisfying current transportation fuel demands, 1: Process alternatives, gasification modeling, process simulation, and economic analysis.* Industrial and Engineering Chemistry Research, 2010. **49**(16): p. 7343-7370.
14. Pouransari, N. and F. Maréchal, *Heat recovery networks synthesis of large-scale industrial sites: Heat load distribution problem with virtual process subsystems.* Energy Conversion and Management, 2015. **89**: p. 985-1000.
15. Marechal, F. and B. Kalitventzeff, *Process integration: Selection of the optimal utility system.* Comput. Chem. Eng., 1998. **22**: p. S149-S156.
16. Soršak, A. and Z. Kravanja, *Simultaneous MINLP synthesis of heat exchanger networks comprising different exchanger types.* Computers and Chemical Engineering, 2002. **26**(4): p. 599-615.
17. Escobar, M. and J.O. Trierweiler, *Optimal heat exchanger network synthesis: A case study comparison.* Applied Thermal Engineering, 2012. **51**(1-2): p. 801-826.

Appendix A Sequential match reduction formulations

A.1 Linear transportation formulation

Sets:

- I={i|i is a hot stream}
- H={h|h is a hot process stream}
- HU={hu|hu is a hot utility stream}
- J={j|j is a cold stream}
- C={c|c is a cold process stream}
- CU={cu|cu is a cold utility stream}
- K={k|k is an element in iϵI}
- L={l|l is an element in jϵJ}
- S={(h,c)|hϵH,cϵC, match between h and c allowed}

Variables:

A_{tot}	Total heat transfer area
A_{hu}(hu,c,j)	Heat transfer area for the heat transfer from hot utility hu and the element l in cold stream c.
A_{cu}(h,k,cu)	Heat transfer area for the heat transfer from element k in hot stream h and cold utility cu
A(h,k,c,l)	Heat transfer area for the heat transfer from element k in hot stream h and element l in cold stream c
Q_{hutot}	Total hot utility duty
q_{hu}(hu,c,l)	Heat transfer between hot utility hu and element l in cold stream c
Q_{cutot}	Total cold utility duty
q_{cu}(h,k,cu)	Heat transfer between element k in hot stream h and cold utility cu
q(h,k,j,l)	Heat transfer between element k in hot stream h and element l in cold stream c
Q_h(h,k)	Available heat in element k in stream h
Q_c(c,l)	Required heat in element l in stream c

Parameters:

c_1,c_2,c_3	Annual cost factors

Equations:

$$ObjFun = c_1 Q_{HU} + c_2 Q_{CU} + c_3 A_{tot} \tag{A.1}$$

$$Q_{HUTOT} = \sum_{hu \in HU} \sum_{c \in C} \sum_{l \in L} q_{hu}(hu, \mathrm{c}, \mathrm{l}) \tag{A.2}$$

$$Q_{CUTOT} = \sum_{cu \in CU} \sum_{h \in H} \sum_{k \in K} q_{cu}(h, \mathrm{k}, \mathrm{cu}) \tag{A.3}$$

$$\sum_{c\in C}\sum_{l\in L} q(h,\mathrm{k},\mathrm{c},\mathrm{l}) + \sum_{cu\in CU} q_{cu}(h,\mathrm{k},\mathrm{cu}) = \mathrm{Q}_h(h,k) \quad \text{(A.4)}$$

$$\sum_{h\in H}\sum_{k\in K} q(h,\mathrm{k},\mathrm{c},\mathrm{l}) + \sum_{hu\in HU} q_{hu}(\mathrm{hu},\mathrm{c},\mathrm{l}) = \mathrm{Q}_C(\mathrm{c},\mathrm{l}) \quad \text{(A.5)}$$

$$Ahu(\mathrm{hu},\mathrm{c},\mathrm{l}) = \frac{1}{U(hu,c) * LMTD(\mathrm{c},\mathrm{l})} q_{HU}(\mathrm{hu},\mathrm{c},\mathrm{l}) \quad \text{(A.6)}$$

$$A_{CU}(\mathrm{h},\mathrm{k},\mathrm{cu}) = \frac{1}{U(h,cu) * LMTD(\mathrm{h},\mathrm{k})} q_{CU}(\mathrm{h},\mathrm{k},\mathrm{cu}) \quad \text{(A.7)}$$

$$A(\mathrm{h},\mathrm{k},\mathrm{c},\mathrm{l}) = \frac{1}{U(h,c) * LMTD(\mathrm{h},\mathrm{k},\mathrm{c},\mathrm{l})} q(\mathrm{h},\mathrm{k},\mathrm{c},\mathrm{l}) \quad \text{(A.8)}$$

$$A_{tot} = \sum_{h\in H}\sum_{k\in K}\sum_{c\in C}\sum_{l\in L} A(h,\mathrm{k},\mathrm{c},\mathrm{l}) + \sum_{hu\in HU}\sum_{c\in C}\sum_{l\in L} A_{HU}(hu,c,l) + \sum_{h\in H}\sum_{k\in K}\sum_{cu\in CU} A_{CU}(h,\mathrm{k},\mathrm{cu}) \quad \text{(A.9)}$$

A.2. Match set reduction

Additional binary variables:

y(h,c)	Binary variable representing the match between hot stream h and cold stream c
y_{hu}(hu,c)	Binary variable representing the match between hot utility hu and cold stream c
y_{cu}(h,cu)	Binary variable representing the match between hot stream h and cold utility cu

Additional parameters:

c_4,c_5,c_6	Fixed cost factors of heat exchanger units
M(i,j)	Smallest total heat content of hot stream i and cold stream j

Equations:

Equations A.2-A9 are included in the model.

$$ObjFun = c_1 Q_{HU} + c_2 Q_{CU} + c_3 A_{tot} + c_4 y(\mathrm{h},\mathrm{c}) + c_5 y_{HU}(\mathrm{hu},\mathrm{c}) + c_6 y_{CU}(\mathrm{h},\mathrm{cu}) \quad \text{(A.10)}$$

$$q(h,c) \le M(\mathrm{h},\mathrm{c})\, y(h,c) \quad \text{(A.11)}$$

$$q(h,cu) \le M(\mathrm{h},\mathrm{cu})\, y_{CU}(h,cu) \quad \text{(A.12)}$$

$$q(hu,c) \le M(\mathrm{hu},\mathrm{c})\, y_{HU}(hu,c) \quad \text{(A.13)}$$

A.3. Grouping

None of the equations, sets or parameters are included in this model. The only thing transferred from the match set reduction step is the matches.

Sets:
G={g|g is a group with matches}

Variables:

e(h,c) binary variable indicating that the match (h, c) is eliminated,
u(h,g) binary variable indicating that the hot stream h belongs to group g,
v(c,g) binary variable indicating that the cold stream c belongs to group g,

Parameters:
M(h) number of matches in the set S that involve hot stream h,
N maximum number of matches allowed in one subgroup

Equations:

$$\min \sum_{h \epsilon H} \sum_{c \epsilon C} e(h,c) \qquad \text{(A.14)}$$

$$\sum_{g \epsilon G} u(h,g) = 1 \qquad \forall\, h \in S \qquad \text{(A.15)}$$

$$\sum_{g \epsilon G} v(c,g) = 1 \qquad \forall\, c \in S \qquad \text{(A.16)}$$

$$v(c,g) - u(h,g) + e(h,c) \geq 0 \qquad \forall\, (h, c) \in S,\ g \in G \qquad \text{(A.17)}$$

$$u(h,g) - v(c,g\ ) + e(h,c) \geq 0 \qquad \forall\, (h, c) \in S,\ g \in G \qquad \text{(A.18)}$$

$$M(h)u(h,g) - \sum_{c \epsilon C} e(h,c) - N \leq 0 \qquad \forall\, h \in S \quad , \quad g \in G \qquad \text{(A.19)}$$

Appendix B GAMS files

B.1. LP formulation

```
*LP-formulation in the sequential match reduction approach

*Coded by Martin J. Megård fall 2015.
*HARDCODED:
*heat transfer coefficients, cost functions and utility consumption

SETS

   i      index for hot process streams and utilities

   h(i)   set of all hot process streams

   HU(i)  Set of all hot utilities

   j      index for cold process streams and utilities

   c(j)   set of all cold process streams

   CU(j)  Set of all cold utilities

   k      index for temperature intervals k=0 is the hottest       /0*12/

   l      index for cold temperature intervals l=0 is the hottest /0*12/

   ;

$call "gdxxrw Pettersson.xls set=i rng='AllHotStreams' cdim=1"

$gdxin Pettersson.gdx

$load i

$call "gdxxrw Pettersson.xls set=H rng='HotProcessStreams' cdim=1"

$gdxin Pettersson.gdx

$load H

$call "gdxxrw Pettersson.xls set=HU rng='HotUtilityStreams' cdim=1"

$gdxin Pettersson.gdx

$load HU

$call "gdxxrw Pettersson.xls set=j rng='AllColdStreams' cdim=1"

$gdxin Pettersson.gdx

$load j

$call "gdxxrw Pettersson.xls set=C rng='ColdProcessStreams' cdim=1"

$gdxin Pettersson.gdx

$load C

$call "gdxxrw Pettersson.xls set=CU rng='ColdUtilityStreams' cdim=1"
```

```
$gdxin Pettersson.gdx

$load CU

PARAMETERS

   tHS(i)  Supply temperature for hot streams

   tHE(i)  Delivery Ttemperature for hot streams

   FH(i)   Heat capacity for hot process streams

   tCS(j)  Supply temperature for cold streams

   tCE(j)  Delivery temperature for cold streams

   FC(j)   Heat capacity for cold process streams

execute "gdxxrw Pettersson.xls par=tHS rng='HotSupplyTemps' cdim=1"

execute_load 'Pettersson.gdx' tHS

execute "gdxxrw Pettersson.xls par=tHE rng='HotTargetTemps' cdim=1"

execute_load 'Pettersson.gdx' tHE

execute "gdxxrw Pettersson.xls par=FH rng='HotFCp' cdim=1"

execute_load 'Pettersson.gdx' FH

execute "gdxxrw Pettersson.xls par=tCS rng='ColdSupplyTemps' cdim=1"

execute_load 'Pettersson.gdx' tCS

execute "gdxxrw Pettersson.xls par=tCE rng='ColdTargetTemps' cdim=1"

execute_load 'Pettersson.gdx' tCE

execute "gdxxrw Pettersson.xls par=FC rng='ColdFCp' cdim=1"

execute_load 'Pettersson.gdx' FC

 Parameters lol1(k)

/0      0,
1       1,
2       2,
3       3,
4       4,
5       5,
6       6,
7       7,
8       8,
9       9,
10      10,
11      11,
12      12/;
```

```
Parameters lol2(l)

/0      12,
1       11,
2       10,
3       9,
4       8,
5       7,
6       6,
7       5,
8       4,
9       3,
10      2,
11      1,
12      0/;

Parameter TINTH(i,k)

Loop(i $ H(i),loop(k,

        TINTH(i,k)=tHs(i)-(tHS(i)-tHe(i))*lol1(k)/12 )) ;

Parameter TINTC(j,l)

Loop(j$C(j),loop(l,

        TINTC(j,l)=tCs(j)-(tCS(j)-tCe(j))*lol2(l)/12 )) ;

PARAMETER hcH(i) Heat transfer coefficient hot strema i

        /H1         2,

        H2          2.5,

        H3          2,

        H4          2,

PARAMETER hcC(j) Heat transfer coefficient cold stream j

        /C1         1.5,

        C2          1,

        C3          1.5,

        C4          2,

        C5          2,

        CW            2/;

PARAMETER EMAT Heat exchanger minimum approach temperature/2.5/;

Parameter

    QH(i,k)  Heat delivered from process stream H to inverval k

    QC(j,l)  Heat recieved by process stream C from interval k
```

```
;

* Assign values to QH and QC

LOOP(i$H(i) , LOOP(k$(ord(k) ge 1),

    IF( ((tHS(i)>TINTH(i,k)) AND (tHE(i)<TINTH(i,k-1))) ,

    QH(i,k)=( MIN(tHS(i),TINTH(i,k-1)) -MAX(tHE(i),TINTH(i,k) ) )*FH(i))));

LOOP(j$C(j) , LOOP(l$(ord(l) ge 1),

    IF( ((tCS(j)<TINTC(j,l-1)) AND (tCE(j)>TINTC(j,l))) ,

    QC(j,l)=( MIN(tCE(j),TINTC(j,l-1)) -MAX(tCS(j),TINTC(j,l) ) )*FC(j))));

SCALAR

  c1     Heating utility cost factor   /70/     ;

SCALAR

    c2     Cooling utility cost factor   /10/   ;

SCALAR

    c3     Area cost factor                /250/  ;

PARAMETER LMTD(i,k,j,l) Log mean temperature difference for heat transfer between temperature
intervals k and l

        DT1      Dummy variable for easy reading

        DT2      Dummy Variable for easy reading

        DTGM     Dummy Variable for easy reading

        DTAM     Dummy variable for easy reading;


Loop(i, LOOP(j$C(j),LOOP(k $ (ord(k) ge 1), LOOP(l $((ord(l) ge 1) and (TINTC(j,l) <=
TINTH(i,k)) and (TINTC(j,l-1) <= TINTH(i,k-1))) ,

       DT1 = TINTH(i,k-1) - TINTC(j,l-1);

       DT2 = TINTH(i,k) - TINTC(j,l);

       DTAM = (DT1 + DT2)/2;

       DTGM = sqrt((DT1*DT2));

       LMTD(i,k,j,l) = 2/3*DTGM+1/3*DTAM;

))));


LOOP(i$H(i),LOOP(j$C(j),LOOP(k $ (ord(k) ge 1), LOOP(l $(ord(l) ge 1),

      LMTD(i,k,j,l) = MAX(0.0000001, LMTD(i,k,j,l));
))));

**HU-C match LMTD

PARAMETER LMTDHU(j,l) Log mean temperature difference for heat transfer between temperature intervals k and l

        DT1HU       Dummy variable for easy reading

        DT2HU       Dummy Variable for easy reading

        DTGMHU      Dummy Variable for easy reading

        DTAMHU      Dummy variable for easy reading;


Loop(j$C(j),LOOP(l $ ((ord(l) ge 0) and (TINTC(j,l+1) <= tHe('st')) and (TINTC(j,l) <= tHs('st')) and (TCE(j)<=THE('st'))) ,

       DT1HU = tHs('st')-TINTC(j,l);

       DT2HU = thE('st')-TINTC(j,l+1);

       DTAMHU = (DT1HU + DT2HU)/2;

       DTGMHU = sqrt((DT1HU*DT2HU));

       LMTDHU(j,l) = 2/3*DTGMHU+1/3*DTAMHU;

));

LOOP(j$C(j),LOOP(l $(ord(l) ge 0),

       LMTDHU(j,l) = MAX(0.000001, LMTDHU(j,l));

));

PARAMETER LMTDCU(i,k) Log mean temperature difference for heat transfer between temperature intervals k and l

        DT1CU       Dummy variable for easy reading

        DT2CU       Dummy Variable for easy reading

        DTGMCU      Dummy Variable for easy reading

        DTAMCU      Dummy variable for easy reading;


Loop(i$H(i),LOOP(k $ ((ord(k) ge 1) and (tcs('cw') <= TINTH(i,k)) and (tce('cw')  <= TINTH(i,k-1))) ,

       DT1CU = TINTH(i,k-1) - tCE('cw');

       DT2CU = TINTH(i,k) - tcs('cw');

       DTAMCU = (DT1CU + DT2CU)/2;

       DTGMCU = sqrt((DT1CU*DT2CU));
       LMTDCU(i,k) = 2/3*DTGMCU+1/3*DTAMCU;
));

Loop(i$H(i),LOOP(k $ ((ord(k) ge 1) and (tcs('cw') <= TINTH(i,k)) and (TINTH(i,k-1)<=(tce('cw'))))
,
       DT1CU = EMAT;
       DT2CU = TINTH(i,k) - tcs('cw');
       DTAMCU = (DT1CU + DT2CU)/2;
       DTGMCU = sqrt((DT1CU*DT2CU));
       LMTDCU(i,k) = 2/3*DTGMCU+1/3*DTAMCU;
));
LOOP(i$H(i),LOOP(k $(ord(k) ge 1),
       LMTDCU(i,k) = MAX(0.000001, LMTDCU(i,k));
));
*Overall heat transfer coefficient for different matches
PARAMETER U(i,j)    Overall heat transfer coefficient
LOOP(i $ H(i), LOOP(j $ C(j),
       U(i,j) = 1.0/(1.0/hcH(i) + 1.0/hcC(j)) ) );
PARAMETER Uhu(i,j)  Overall heat transfer coefficient for hot utility
LOOP(i $ HU(i),LOOP(j,
       Uhu(i,j) = 1.0/(1.0/hcH(i) + 1.0/hcC(j)) ) );
PARAMETER Ucu(i,j)  Overall heat transfer coefficient for cold utility
LOOP(i, LOOP(j $ CU(j),
       Ucu(i,j) = 1.0/(1.0/hcH(i) + 1.0/hcC(j)) ) );
SCALARS
       Qhmintot   TOTAL Minimum hot utility /input hot utility/
       Qcmintot   TOTAL Minimum cold utility /input cold utility/
;
Positive VARIABLES
    Q(i,k,j,l)   heat exchanged between hot process stream h in
*                interval k and cold process stream  c in interval l
```

```
        Qhu(i,j,l) heat exchanged between hot utility and cold stream j in interval l

        Qcu(i,k,j) Heat exchanged between hot stream i and cold utility j in interval k

        ATOTAL      Total Area

        Ahu(i,j,l)    Area of match between hot utility and cold stream c in temperature intervall l

        Acu(i,k,j)

         A(i,k,j,l)

VARIABLES

  ANNCOST Annual cost

   ;

EQUATIONS

        OBJFUN  Objective function to be minimized

        TOTHU   Total hot utility

        TOTCU   Total cooling utility

        HEATAV(i,k)  Availabe heat in element k in the hot stream h

        HEATRQ(j,l)  Required heat in element j in the cold stream c

        ACOMRQ(i,k,j,l)  Required heat transfer area for each combination of elements between h anc c
streams

        ACOMRQHU(i,j,l)     Required heat transfer area for each combination of elements between hu
anc c streams

        ACOMRQCU(i,k,j)     Required heat transfer area for each combination of elements between cu
anc h stream

        ATOT        Total Required Heat transfer area

;

OBJFUN..

         ANNCOST =E= c1*Qhmintot+c2*Qcmintot+c3*ATOTAL;

TOTHU..

       Qhmintot=E=SUM(i $ HU(i),SUM (j $ C(j), SUM (l, qhu(i,j,l))));

TOTCU..

        Qcmintot=E=SUM(j $ CU(j),SUM (i $ H(i), SUM (k, qcu(i,k,j))));

HEATAV(i,k) $ H(i)..

        QH(i,k)=E=SUM(j $ C(j), SUM(l,q(i,k,j,l))) + SUM(j $ CU(j),qcu(i,k,j));

HEATRQ(j,l) $ C(j)..
```

```
        QC(j,l)=E=SUM(i $ H(i), SUM(k,q(i,k,j,l))) + SUM(i $ HU(i),qhu(i,j,l));

ACOMRQ(i,k,j,l) $( H(i) and C(j) and ord(k) ge 0 and ord(l) ge 0)..

        A(i,k,j,l)=E= q(i,k,j,l)/(U(i,j)*LMTD(i,k,j,l));

ACOMRQHU(i,j,l) $( HU(i) and C(j) and ord(l) ge 0)..

        Ahu(i,j,l)=E= (1/(Uhu(i,j)*LMTDHU(j,l)))*Qhu(i,j,l);

ACOMRQCU(i,k,j) $( CU(j)and H(i) and ord(k) ge 1)..

         Acu(i,k,j)=E= (1/(Ucu(i,j)*LMTDCU(i,k)))*Qcu(i,k,j);

ATOT..

        ATOTAL=E=SUM(i $ H(i),SUM(k,SUM(j $ C(j),SUM(l,A(i,k,j,l)))))+SUM(i$HU(i),SUM(j $
C(j),SUM(l,Ahu(i,j,l))))+SUM(i$H(i),SUM(k,SUM(j $ CU(j),Acu(i,k,j))));

* Cplex options

OPTION OPTCR = 0.001;

MODEL Pettersson /ALL/;

SOLVE Pettersson USING LP MINIMIZING ANNCOST;

PARAMETER HLDs(i,j) Heat Load Distribution;

HLDs(i,j) = SUM(k,sum(l, Q.L(i,k,j,l)))+sum(k,Qcu.L(i,k,j))+sum(l,Qhu.L(i,j,l));
```

B.2. MILP formulation

```
*MILP-formulation of the Match reduction step by F.Pettersson

*Coded by Martin J. Megård fall 2015.

*HARDCODED:

* heat transfer coefficients

*Table d(i,j) introduces binary variables.

*Setting the sets

SETS

   i      index for hot process streams and utilities

   h(i)   set of all hot process streams

   HU(i)  Set of all hot utilities

   j      index for cold process streams and utilities

   c(j)   set of all cold process streams
   CU(j)  Set of all cold utilities
   k      index for temperature intervals k=0 is the hottest /0*12/
   l      index for cold temperature intervals l=0 is the hottest /0*12/
   S(i,k,j,l) Match between hot stream i and cold stream j
   ;
$call "gdxxrw Pettersson.xls set=i rng='AllHotStreams' cdim=1"
$gdxin Pettersson.gdx
$load i
$call "gdxxrw Pettersson.xls set=H rng='HotProcessStreams' cdim=1"
$gdxin Pettersson.gdx
$load H
$call "gdxxrw Pettersson.xls set=HU rng='HotUtilityStreams' cdim=1"
$gdxin Pettersson.gdx
$load HU
$call "gdxxrw Pettersson.xls set=j rng='AllColdStreams' cdim=1"
$gdxin Pettersson.gdx
$load j
$call "gdxxrw Pettersson.xls set=C rng='ColdProcessStreams' cdim=1"
$gdxin Pettersson.gdx
$load C
$call "gdxxrw Pettersson.xls set=CU rng='ColdUtilityStreams' cdim=1"
$gdxin Pettersson.gdx
$load CU
PARAMETERS
   tHS(i)  Supply temperature for hot streams
   tHE(i)  Delivery Ttemperature for hot streams
   FH(i)   Heat capacity for hot process streams
   tCS(j)  Supply temperature for cold streams
   tCE(j)  Delivery temperature for cold streams
```

```
    FC(j)   Heat capacity for cold process streams

execute "gdxxrw Pettersson.xls par=tHS rng='HotSupplyTemps' cdim=1"

execute_load 'Pettersson.gdx' tHS

execute "gdxxrw Pettersson.xls par=tHE rng='HotTargetTemps' cdim=1"

execute_load 'Pettersson.gdx' tHE

execute "gdxxrw Pettersson.xls par=FH rng='HotFCp' cdim=1"

execute_load 'Pettersson.gdx' FH

execute "gdxxrw Pettersson.xls par=tCS rng='ColdSupplyTemps' cdim=1"

execute_load 'Pettersson.gdx' tCS

execute "gdxxrw Pettersson.xls par=tCE rng='ColdTargetTemps' cdim=1"

execute_load 'Pettersson.gdx' tCE

execute "gdxxrw Pettersson.xls par=FC rng='ColdFCp' cdim=1"

execute_load 'Pettersson.gdx' FC

 Parameters lol1(k)

/0      0,
1       1,
2       2,
3       3,
4       4,
5       5,
6       6,
7       7,
8       8,
9       9,
10      10,
11      11,
12      12/;

Parameters lol2(l)

/0      12,
1       11,
2       10,
3       9,
4       8,
5       7,
6       6,
7       5,
8       4,
9       3,
10      2,
11      1,
12      0/;

Parameter TINTH(i,k)

Loop(i,loop(k,

        TINTH(i,k)=tHs(i)-(tHS(i)-tHe(i))*lol1(k)/12 )) ;

DISPLAY TINTH;

Parameter TINTC(j,l)

Loop(j$C(j),loop(l,

        TINTC(j,l)=tCs(j)-(tCS(j)-tCe(j))*lol2(l)/12 )) ;

DISPLAY TINTC;

PARAMETER hcH(i) Heat transfer coefficient hot strema i

        /H1          2,

       H2           2.5,

        H3           2,

        H4           2,

        H5           1.5,

        ST           1  /;

PARAMETER hcC(j) Heat transfer coefficient cold stream j

        /C1          1.5,

        C2           1,

        C3           1.5,

        CW            2/;

Parameters

    QH(i,k)  Heat delivered from process stream H to inverval k

    QC(j,l)  Heat recieved by process stream C from interval k

* Assign values to QH and QC

LOOP(i$H(i) , LOOP(k$(ord(k) ge 1),

     IF( ((tHS(i)>TINTH(i,k)) AND (tHE(i)<TINTH(i,k-1))) ,

     QH(i,k)=( MIN(tHS(i),TINTH(i,k-1)) -MAX(tHE(i),TINTH(i,k) ) )*FH(i))));

LOOP(j$C(j) , LOOP(l$(ord(l) ge 1),
    IF( ((tCS(j)<TINTC(j,l-1)) AND (tCE(j)>TINTC(j,l))) ,
    QC(j,l)=( MIN(tCE(j),TINTC(j,l-1)) -MAX(tCS(j),TINTC(j,l) ) )*FC(j))));
Parameters
M(i,j)   Constraint for heat exchange between hot stream i and cold stream j
Mcu(i)   Constraint for heat exchange between cold utility j and hot stream i
Mhu(j)   Constraint for heat exchange between cold stream j and hot utility i
;
loop(i$H(i),loop(j$C(j),
M(i,j)=min(sum(k,QH(i,k)),sum(l,QC(j,l)))));
loop(i$H(i),
Mcu(i)=sum(k,QH(i,k))) ;
loop(j$C(j),
Mhu(j)=sum(l,QC(j,l))) ;
SCALAR
 c1      Heating utility cost factor   /70/     ;
SCALAR
   c2      Cooling utility cost factor   /10/   ;
SCALAR
   c3      Area cost factor              /250/  ;
SCALAR
        c4      Process heat exchanger unit cost /15000/  ;
SCALAR
        c5      Hot stream cold utility unit cost /15000/;
SCALAR
        c6      COld stream hot uitlity unit cost /15000/;
PARAMETER LMTD(i,k,j,l) Log mean temperature difference for heat transfer between
temperature intervals k and l
         DT1       Dummy variable for easy reading
         DT2       Dummy Variable for easy reading
         DTGM      Dummy Variable for easy reading
        DTAM      Dummy variable for easy reading;

Loop(i$H(i), LOOP(j$C(j),LOOP(k $ (ord(k) ge 1), LOOP(l $((ord(l) ge 1) and
(TINTC(j,l)<= TINTH(i,k)) and (TINTC(j,l-1)<= TINTH(i,k-1))) ,

       DT1 = TINTH(i,k-1) - TINTC(j,l-1);

       DT2 = TINTH(i,k) - TINTC(j,l);

       DTAM = (DT1 + DT2)/2;

       DTGM = sqrt((DT1*DT2));

       LMTD(i,k,j,l) = 2/3*DTGM+1/3*DTAM;

))));

LOOP(i$H(i),LOOP(j$C(j),LOOP(k $ (ord(k) ge 1), LOOP(l $(ord(l) ge 1),

      LMTD(i,k,j,l) = MAX(0.0000001, LMTD(i,k,j,l));

))));

**HU-C match LMTD

PARAMETER LMTDHU(j,l) Log mean temperature difference for heat transfer between
temperature intervals k and l

        DT1HU       Dummy variable for easy reading

        DT2HU       Dummy Variable for easy reading

        DTGMHU      Dummy Variable for easy reading

        DTAMHU      Dummy variable for easy reading;


Loop(j$C(j),LOOP(l $ ((ord(l) ge 0) and (TINTC(j,l+1) <= tHe('st')) and (TINTC(j,l) <=
tHs('st')) and (TCE(j)<=THE('st'))) ,

       DT1HU = tHs('st')-TINTC(j,l);

       DT2HU = thE('st')-TINTC(j,l+1);

       DTAMHU = (DT1HU + DT2HU)/2;

       DTGMHU = sqrt((DT1HU*DT2HU));

       LMTDHU(j,l) = 2/3*DTGMHU+1/3*DTAMHU;

));

*FJERN DENNE LOOP OG SISte I loopen over

*Loop(j$C(j),LOOP(l $ ((ord(l) ge 0) and (tHe('st')<=TCE(j) ) and TcS(J)<=THE('st') ) ,

*       DT1HU = tHs('st')-TINTC(j,l);
*        DT2HU = EMAT;
*        DTAMHU = (DT1HU + DT2HU)/2;
*       DTGMHU = sqrt((DT1HU*DT2HU));
*        LMTDHU(j,l) = 2/3*DTGMHU+1/3*DTAMHU;
*));
LOOP(j$C(j),LOOP(l $(ord(l) ge 0),
        LMTDHU(j,l) = MAX(0.000001, LMTDHU(j,l));
));
PARAMETER LMTDCU(i,k) Log mean temperature difference for heat transfer between
temperature intervals k and l
         DT1CU       Dummy variable for easy reading
         DT2CU       Dummy Variable for easy reading
         DTGMCU      Dummy Variable for easy reading
         DTAMCU      Dummy variable for easy reading;


Loop(i$H(i),LOOP(k $ ((ord(k) ge 1) and (tcs('cw') <= TINTH(i,k)) and (tce('cw')<=
TINTH(i,k-1))) ,
        DT1CU = TINTH(i,k-1) - tCE('cw');
        DT2CU = TINTH(i,k) - tcs('cw');
        DTAMCU = (DT1CU + DT2CU)/2;
        DTGMCU = sqrt((DT1CU*DT2CU));
        LMTDCU(i,k) = 2/3*DTGMCU+1/3*DTAMCU;
));
Parameter EMAT /2.5/;
Loop(i$H(i),LOOP(k $ ((ord(k) ge 1) and (tcs('cw') <= TINTH(i,k)) and (TINTH(i,k-
1)<=(tce('cw')))) ,
        DT1CU = EMAT;
        DT2CU = TINTH(i,k) - tcs('cw');
        DTAMCU = (DT1CU + DT2CU)/2;
        DTGMCU = sqrt((DT1CU*DT2CU));
        LMTDCU(i,k) = 2/3*DTGMCU+1/3*DTAMCU;
```

```
));
LOOP(i$H(i),LOOP(k $(ord(k) ge 1),
        LMTDCU(i,k) = MAX(0.000001, LMTDCU(i,k));
));
*Overall heat transfer coefficient for different matches
PARAMETER U(i,j)    Overall heat transfer coefficient
LOOP(i $ H(i), LOOP(j $ C(j),
        U(i,j) = 1.0/(1.0/hcH(i) + 1.0/hcC(j)) ) );
PARAMETER Uhu(i,j)  Overall heat transfer coefficient for hot utility
LOOP(i $ HU(i),LOOP(j,
        Uhu(i,j) = 1.0/(1.0/hcH(i) + 1.0/hcC(j)) ) );
PARAMETER Ucu(i,j)  Overall heat transfer coefficient for cold utility
LOOP(i, LOOP(j $ CU(j),
        Ucu(i,j) = 1.0/(1.0/hcH(i) + 1.0/hcC(j)) ) );
BINARY Variable
Y(i,j)   Binary variable representing matches between streams that may be removed;

TABLE D(i,j) Setting matches that will have binary variables
“Input table of matches with d(i,j)=1 to introduce binary variables
Positive VARIABLES
    Q(i,k,j,l)   heat exchanged between hot process stream h in
*                interval k and cold process stream  c in interval l
        Qhu(i,j,l) heat exchanged between hot utility and cold stream j in interval l
        Qcu(i,k,j) Heat exchanged between hot stream i and cold utility j in interval k
        ATOTAL      Total Area
        Ahu(i,j,l)    Area of match between hot utility and cold stream c in temperature intervall
l
        Acu(i,k,j)
         A(i,k,j,l)
;
```

```
Scalars

      Qhmintot   TOTAL Minimum hot utility  /input hot utility consumption/

      Qcmintot   TOTAL Minimum cold utility  /input cold utility consumption/;

VARIABLES

  ANNCOST Annual cost

  UNITCOST

  Areacost

  Utilitycost

   ;

EQUATIONS

      OBJFUN         Objective function to be minimized

      AREA           Total cost Area

      UTILITY        Total cost utility

      UNITS          Total cost of units

      TOTHU          Total hot utility

      TOTCU          Total cooling utility

      HEATAV(i,k)     Availabe heat in element k in the hot stream h

      HEATRQ(j,l)     Required heat in element j in the cold stream c

      ACOMRQ(i,k,j,l) Required heat transfer area for each combination of elements between
h anc c streams

      ACOMRQHU(i,j,l) Required heat transfer area for each combination of elements
between hu anc c streams

      ACOMRQCU(i,k,j) Required heat transfer area for each combination of elements
between cu anc h stream

      ATOT           Total Required Heat transfer area

      CONI(i,j)      Heat transfer constraint 1

      CONII(i,j)      Heat transfer constraint 2

      CONIII(i,j)     Heat transfer constraint 3

     ;

OBJFUN..

       ANNCOST =E= AREACOST+Utilitycost+UNITCOST;
```

```
AREA..
AREACOST=E= c3*ATOTAL;
UTILITY..
Utilitycost=E= c1*Qhmintot+c2*Qcmintot;
UNITS..
        UNITCOST =E=
c4*sum(i$H(i),sum(j$C(j),Y(i,j)))+c5*sum(i$H(i),sum(j$CU(j),Y(i,j)))+c6*sum(i$HU(i),sum
(j$C(j),Y(i,j)));
TOTHU..
        Qhmintot=E=SUM(i $ HU(i),SUM (j $ C(j), SUM (l, qhu(i,j,l))));
TOTCU..
        Qcmintot=E=SUM(j $ CU(j),SUM (i $ H(i), SUM (k, qcu(i,k,j))));
HEATAV(i,k) $ H(i)..
        QH(i,k)=E=SUM(j $ C(j), SUM(l,q(i,k,j,l))) + SUM(j $ CU(j),qcu(i,k,j));
HEATRQ(j,l) $ C(j)..
        QC(j,l)=E=SUM(i $ H(i), SUM(k,q(i,k,j,l))) + SUM(i $ HU(i),qhu(i,j,l));
ACOMRQ(i,k,j,l) $( H(i) and C(j) and ord(k) ge 0 and ord(l) ge 0)..
        A(i,k,j,l)=E= (1/(U(i,j)*LMTD(i,k,j,l)))*q(i,k,j,l);
ACOMRQHU(i,j,l) $( HU(i) and C(j) and ord(l) ge 0 )..
        Ahu(i,j,l)=E= (1/(Uhu(i,j)*LMTDHU(j,l)))*Qhu(i,j,l);
ACOMRQCU(i,k,j) $( CU(j)and H(i) and ord(k) ge 0)..
         Acu(i,k,j)=E= (1/(Ucu(i,j)*LMTDCU(i,k)))*Qcu(i,k,j);
ATOT..
        ATOTAL=E=SUM(i $ H(i),SUM(k,SUM(j $
C(j),SUM(l,A(i,k,j,l)))))+SUM(i$HU(i),SUM(j $
C(j),SUM(l,Ahu(i,j,l))))+SUM(i$H(i),SUM(k,SUM(j $ CU(j),Acu(i,k,j))));


CONI(i,j)$(H(i) and C(j) and d(i,j) = 1)..
       M(i,j)*y(i,j)=G=Sum(k,sum(l,q(i,k,j,l)))            ;
CONII(i,j)$(HU(i) and C(j) and d(i,j) = 1)..
        Mhu(j)*y(i,j)=G=sum(l,qhu(i,j,l));
```

```
CONIII(i,j)$(H(i) and CU(j) and d(i,j) = 1)..
        Mcu(i)*y(i,j)=G=sum(k,qcu(i,k,j));
* Cplex options
OPTION OPTCR = 0.01;
MODEL PetterssonMILP /ALL/;
SOLVE PetterssonMILP USING MIP MINIMIZING ANNCOST;
*PARAMETER HLDs(i,j) Heat Load Distribution;
PARAMETER HLDs(i,j) Heat Load Distribution;
HLDs(i,j) = SUM(k,sum(l, Q.L(i,k,j,l)))+sum(k,Qcu.L(i,k,j))+sum(l,Qhu.L(i,j,l));
```

B.3. Grouping formulation

```
*Grouping MILP by Pettersson
*coded by Martin Megård
*Hard coded:
*Number of groups(Just increase number for larger problems)
*HLD represented in table Y(h,c)
*N Number of matches allowed per subgroup
SETS
   h
   c   set of all cold process streams
   G      Groups of matches /1*2/
;
$call "gdxxrw Pettersson.xls set=H rng='HotProcessStreams' cdim=1"
$gdxin Pettersson.gdx
$load H
$call "gdxxrw Pettersson.xls set=C rng='ColdProcessStreams' cdim=1"
$gdxin Pettersson.gdx
$load C
```

```
BINARY VARIABLES
e(h,c)   Binary variable indicating that the match (hc) is eliminated
u(h,g)   Binary variable indicating that the hot stream h belongs to group g
v(c,g)   Binary variable indicating that the cold stream c belongs to group g
;
TABLE y(h,c) HLD
       C1    C2    C3    C4    C5
H1      1     1     0     0     1
H2      0     1     0     0     1
H3      1     0     0     1     0
H4      0     1     1     1     0
Scalar N Maximum number of matches allowed in one subgroup/4/;
Parameter M(h)
loop(h,
M(h)=(sum(c,Y(h,c))));
Variable
ELIM
;
Equations
OBJFUN    Objective function that minimizes the number of eliminated matches
HOTST(h,c)      Constraint that says one hot stream can only belong to one group
COLST(h,c)       Constraint that says one cold stream can only belong to one group
COMBOH      Make sure that a hot  stream in one match belongs in the same group
COMBOC     Make sure that a cold  stream in one match belongs in the same group
SIZECONTROL(g)      Make sure that size of groups does not exceed M
  ;


OBJFUN..
ELIM =E= sum(h,sum(c,e(h,c)));
```

```
HOTST(h,c)$(y(h,c)=1)..
sum(g,u(h,g))=E=1;
COLST(h,c)$(y(h,c)=1)..
sum(g,v(c,g))=E=1;
COMBOH(h,c,g)$(y(h,c)=1)..
v(c,g)-u(h,g)+e(h,c)=G=0;
COMBOC(h,c,g)$(y(h,c)=1)..
u(h,g)-v(c,g)+e(h,c)=G=0;
SIZECONTROL(g)..
sum(h,M(h)*u(h,g))-N-sum(h,sum(c,e(h,c)))=L=0
* Cplex options
OPTION OPTCR = 0.001;
MODEL Grouping /ALL/;
SOLVE Grouping USING MIP MINIMIZING ELIM;
```

Appendix C HEX networks Case Study 1

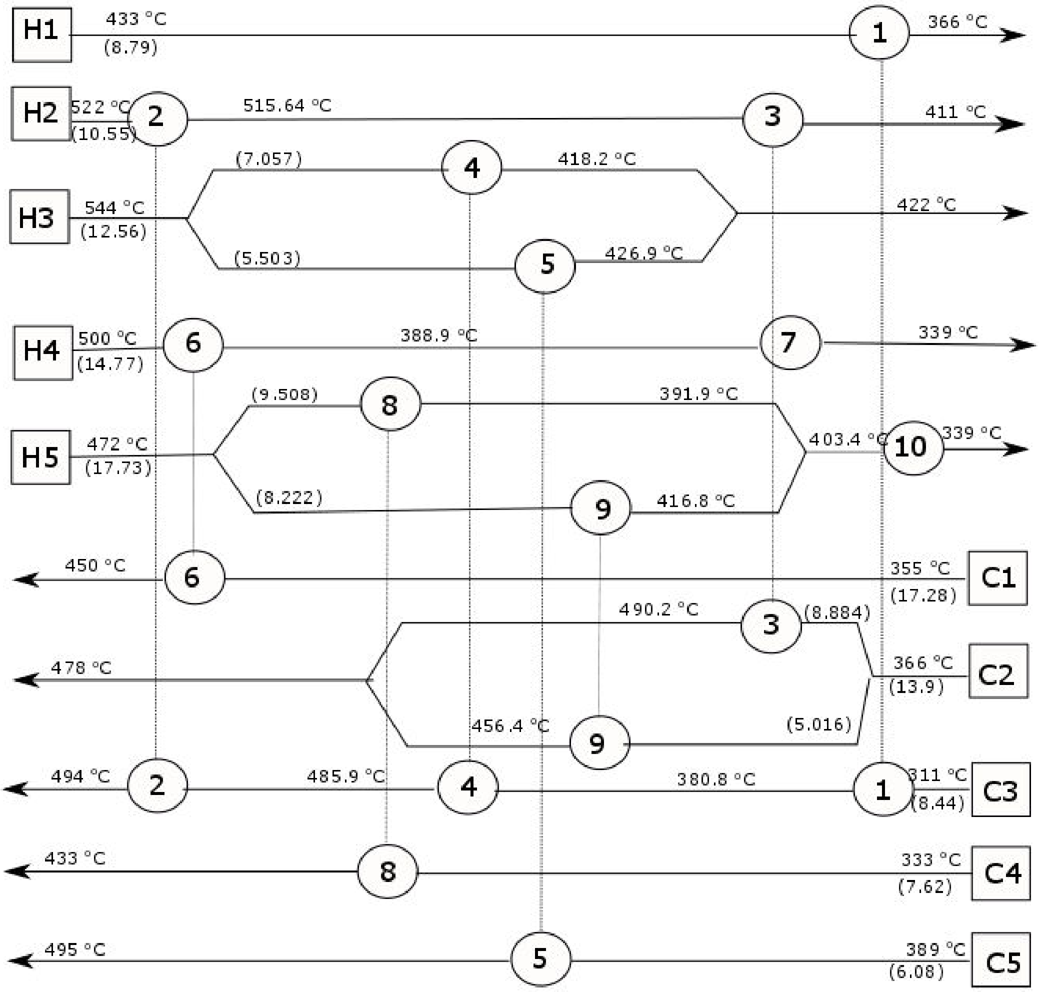


Figure C. 1 Network for simulation 1 Case study 1

Appendix D Heat Load Distribution Case Study 3

Table D1 HLD from the adapted Sequential Framework Case Study 3 pt.1.

	C1	C2	C3	C4	C5	C6	C7	C8	C9
H1					1462.5				
H2	1975		425						
H3		945.8	2204.2						
H4									
H5						754.2		1745.8	
H6									
H7			829.167		1670.8				
H8	1825								
H9		1850							
H10				1041.7			1658.3		
H11									
H12					1095.8			1404.2	
H13									
H14						1445.8	454.2		
H15				1475					
H16			929.2				87.5		
H17			862.5	737.5					
H18				945.833					
H19									3150
H20					4925				
H21		500							
H22									
ST		1604.2			2845.8				

Table D2 HLD from the adapted Sequential Framework Case Study 3 pt.2.

	C10	C11	C12	C13	C14	C15	C16	C17	CW
H1							1687.5		
H2									
H3									
H4							220.8	1441.7	1187.5
H5									
H6								1250	
H7									1250
H8			1145.8	929.2					
H9		1000							
H10									900
H11					1187.5				1062.5
H12									
H13							41.7	1208.3	
H14									
H15	1800								2325
H16				470.8	562.5	500			
H17									
H18			654.2						
H19							1050		
H20									1025
H21						700			
H22								2100	
ST									