HEAT EXCHANGER NETWORK DESIGN, MONITORING AND OPTIMIZATION

HEAT EXCHANGER NETWORK DESIGN, MONITORING AND OPTIMIZATION

by

UMA MAHESHWAR KIRAN ATI,

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Applied Science

McMaster University

MASTER OF APPLIED SCIENCE (2009) McMaster University
(Chemical Engineering) Hamilton, Ontario, Canada.

TITLE:	Heat Exchanger Network Design, Monitoring and Optimization
AUTHOR:	Uma Maheshwar Kiran Ati.
SUPERVISOR:	Dr. Vladimir Mahalec.
NUMBER OF PAGES:	xiv, 137

Abstract

In process industries, heat exchanger networks represent an important part of the plant structure. The purpose of the networks is to maximize heat recovery, thereby lowering the overall plant costs. Previously published research on heat exchanger networks deals with two categories:

- Synthesis of heat exchanger networks with the goal of designing a structure that provides the lowest total (capital plus operating) costs.
- Data reconciliation with the goal of establishing true performance of the network and identifying correct heat transfer coefficients for individual exchangers in the network.

Since heat exchanger models are highly nonlinear due to presence of log mean temperature difference term, solution of the network models is not always guaranteed. Most of the published results have used some form of approximation of the log mean temperature difference term. The approximations have been designed to provide reasonable accuracy while providing better convergence properties. Nevertheless, these are approximations and lead to the results that are not quite accurate. The goal of this research is to develop heat exchanger network models and algorithms for design, monitoring and optimization that are easy to implement in engineering practice and have excellent convergence properties.

Presented here is a new heat exchanger network simulation algorithm which solves rigorously heat exchanger network equations in three phases:

- Phase 1: Solve mass balance equations for the network, i.e. determine flows in all branches of the network. These equations are linear.
- Phase 2: Compute heat exchanger heat transfer factor for each exchanger in the network. Computation of the factor for each heat exchanger employs current flows through the exchanger and values of the exchanger variables at some base operating conditions.

- Phase 3: Compute all heat exchanger outlet temperatures, given temperatures of the inlet streams and the results from Phase 1 and Phase 2. The computation in this phase is also employing a set of linear equations, while retaining full rigorous of the heat transfer equations.

Hence, we have successfully transformed solution of a heat exchanger network into multi-phase solutions of sets of linear equations. This approach is then used for HEN synthesis and data reconciliation of HENs.

HEN synthesis has been extensively studied over years and significant progress has been achieved in the development of robust methods for design of cost-optimal networks but one of the relatively less addressed issues is to deign HENs based on standard or modular sizes of heat exchangers. The major complexities in HEN synthesis are handling the combinatorial nature of the problem and finding a feasible and optimum solution using simultaneous synthesis methods. In this research, HEN simulation algorithm combined with differential evolutionary optimization is used for design of HENs with modular sizes of heat exchangers. This approach is successfully applied to examples available in the literature. Previously published results have used heat exchangers that have been sized for a placement at a specific location in a heat exchanger network, thereby aiming to provide the lowest cost solution. The research presented here shows that equally good or better solutions can be obtained by using standard, modular sizes of the heat exchangers. The approach used in this work is more realistic, since in practice heat exchangers are available in standards sizes, not custom made (in other words, a heat exchanger typically would have a size of 50 or 100 sq ft, but not 49.8 or 101.9 sq ft as may be calculated by the methods published in the literature).

Data reconciliation and parameter estimation is an important step in HEN performance monitoring. In the current research, the HEN simulation algorithm is extended to develop a framework for data reconciliation of HEN and to estimate the change in the overall heat transfer coefficient of heat exchangers. This methodology is successfully implemented on two case studies from the literature.

Acknowledgements

I would like to thank my supervisor Dr. Vladimir Mahalec for his continued guidance and support throughout the past two years; I would like to sincerely thank him for providing me an excellent opportunity to learn both in chemical engineering and software development perspective through this research project.

I would like to thank McMaster University and MACC for financially supporting my studies here. I would like to thank all the professors at MACC, who have provided me an excellent background for my research through the graduate courses. I would like to thank all colleagues at MACC for their support and help. Special thanks to Rahul, Siam and Zheng for their good welcoming days at McMaster.

I would like to specially thank all my friends at Woodland apartments and Camelot towers, who have made my days at McMaster very joyful and memorable. I am very thankful to them for all their support and encouragement over the course of this study.

I would like to thank all my teachers in India, who have imparted good education to me, which lead me to reach this point. I would like to thank all my friends in India who supported and encouraged me in every moment of my life. Special thanks to my friend Prabhat, who always have time to hear my words.

My eternal gratitude goes to my Grandmother, Ati Shankaramba, Parents, Ati V.V.S. Murthy and Ati Lalitha, Sister, Ati Uma Shankari and Brother, Ati Balasubramanyam for all their sacrifices and hard work they made to make me reach this point. I would like to say a special thanks to my sister and brother who have emotionally supported me during my downtrend.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In process industries, during operation of any heat exchanger network (HEN), the major aim is to focus on the best performance of the network. Frequently one encounters problems that degrade the HEN performance, like heat exchanger fouling, leakage in tubes, changes in process stream conditions (flow rate, temperature), frequent changes in arrangement of utility streams to optimize heat recovery in the network, shutdown of heat exchangers for maintenance, etc [25]. Since the changes can take place in any of the heat exchangers in the network, a complete analysis of the network in an integrated approach is required. In order to handle these issues a good understanding of modeling and simulation of HENs in a simultaneous approach is necessary.

The design and optimization of HENs has been extensively studied over years and significant progress has been achieved in the development of robust methods for design of cost-optimal networks. A thorough review of these methodologies is presented in [22]. The major complexities in HEN synthesis are handling the combinatorial nature of the problem and finding a feasible and optimum solution using simultaneous synthesis methods.

Until recently, HEN design and optimization is largely handled using traditional optimization methods, which are robust and rigorous but are often restricted in handling large-scale problems due to the size of the problem and exponentially growing demand for computational work. Recently, non-traditional optimization methods (evolutionary methods) like genetic algorithm (GA), differential evolution have been catching up the

popularity for handling large-scale combinatorial problems [21]. These algorithms have been extensively applied to HEN synthesis problems which promised simple, fast and robust optimization solutions.

Considering these aspects, the research plan is to develop a novel HEN simulation algorithm that converges quickly, design of HEN synthesis and optimization via differential evolution optimization and extend this framework for data reconciliation and parameter estimation.

1.1 Motivation and Goals

The goal of this research project is to develop heat exchanger network models and algorithms for design, monitoring and optimization that are easy to implement in engineering practice and preferably linear models without approximations for heat transfer equations. The major issue in heat exchanger model equations is handling the nonlinear logarithmic mean temperature difference (LMTD) calculations, to eliminate the nonlinearity in the model several reformulated heat exchangers models are developed either based on linearization of LMTD or approximations of geometric mean temperature difference (GMTD), arithmetic mean temperature difference AMTD or Chen's approximations [7]. The drawbacks of these approaches are they are approximate models. So, development of heat exchanger network models that are linear and rigorous is considered in this research.

There are simulation algorithms that have been developed for solving HEN simultaneously; these simulation algorithms are extended to sensitivity analysis and flexibility analysis [25, 34]. The drawback in these approaches is they require an iterative approach to converge the network when there are simultaneous changes in several inputs. So, development of HEN simulation algorithm that is non-iterative and rigorous is considered in this research.

In the literature several HEN synthesis methodologies have been developed to design cost optimum network, in all these designs of the one of the relatively less addressed issues is to deign HENs based on standard sizes of heat exchangers available in the market. So, considering this fact a methodology to design a cost optimum HEN consisting heat exchangers of modular sizes in the final network is accounted in this research.

Considering the above issues, the objective of this research is further detailed into three sub-objectives as

1. Develop a new algorithm for heat exchanger network simulation that is non-iterative and rigorous.
2. Design a framework for heat exchanger network synthesis with heat exchangers that are produced in standards sizes instead of custom
3. Extend the above methodology for online parameter estimation and real time optimization.

1.2 Main Contributions

The major contributions of this research is detailed into three sub-objectives as

1) **Heat Exchanger Network Modeling and Simulation Algorithm:** A reformulated heat exchanger model is used in this work. This model is extended for development of a new, non-iterative, robust, algorithm for HEN simulation.

2) **HEN Synthesis Using Modular Sizes of Heat Exchangers:** The HEN simulation algorithm is extended to synthesize HEN networks consisting of modular sizes of heat exchangers. A methodology of simultaneous HEN Synthesis via differential evolutionary optimization is developed.

3) **Data Reconciliation of HEN:** The HEN simulation framework is extended to data reconciliation of HEN to calculate the change in the overall heat transfer coefficient of heat exchangers.

4) **Software for Modeling and Synthesis of Large Heat Exchanger Networks:** The equations for the HEN are developed using auto equation generator (AEG), a software application developed for simulation and optimization of chemical engineering applications. AEG has been developed by a team of six contibrutors[Aman .T, Charumitra .P, Mahalec .V, Masoori, M., Shefali .K, Uma .M. K. A].

1.3 Thesis Overview

Chapter 2- Literature Review

In this chapter an introduction to HEN modeling and a literature review on the existing models is discussed, in addition a literature review on the existing methodologies of HEN simulation algorithms, synthesis and data reconciliation are discussed.

Chapter 3- HEN Modeling and Simulation Algorithm

This chapter describes a review of the available heat exchanger model formulations in the literature, the details of the reformulated heat exchanger model used in this study, the details of a novel, non-iterative, robust HEN simulation algorithm and a short overview on AEG.

Chapter 4- HEN Synthesis via Differential Evolution

This chapter provides an overview of evolutionary optimization approaches and the different types of approaches available, followed by introduction to differential

evolutionary optimization approach. The extension of HEN simulation algorithm combined with DE to design HEN networks consisting of modular sizes of heat exchangers is discussed along with the application of the this approach on four case studies reported in the literature to test its performance.

Chapter 5- Data Reconciliation and Parameter Estimation

In this chapter the simulation algorithm discussed in chapter 3 is extended to data reconciliation of steady state HEN. A short overview on the variables classification, parameter observability analysis and QR factorization is presented. Problem formulation of DRPE for HEN is discussed along with the application of this approach on two case studies.

Chapter 6- Conclusions and Recommendations

A summary of the thesis with major results achieved in this research and future extension of this research are also discussed.

Chapter 2

Literature Review

This chapter provides an overview of the process heat exchanger network synthesis problem, followed by details of the literature survey done on existing methods to eliminate LMTD term in the equations of a heat exchanger model, the current state of simulation algorithms for solving heat exchanger networks simultaneously in a linear fashion and the existing methodologies for heat exchanger network synthesis. An overview of the evolutionary optimization methods and a detailed explanation on differential evolution optimization method is presented.

2.1 Process Description – HEN

Heat exchangers are extensively used in process industries for adjustment of temperatures of the process streams. A set of heat exchangers are connected to develop a heat exchanger network (HEN). Proper placement or adjustment of the heat exchangers on the hot and cold streams to recover maximum energy from the process streams, decrease the demand for more hot and cold utilities usage in the industries. Due to the constant increase in energy prices lead to an extensive research in design and development of simulation and optimization methods that improves the performance of HENs [22].

A HEN problem is define as following,

- A set of hot streams that are to be cooled from a specified inlet temperature to a specified outlet temperature
- A set of cold streams that are to be heated from a specified inlet temperature to a specified outlet temperature
- Given input conditions are specific heat capacities of the streams, mass flow rates of the hot and cold streams.
- The mass flow rate, temperature conditions of the hot and cold utilities is provided with their cost information.

For the given information, the aim would be to develop a HEN that would result in minimum annual cost for maintenance of the HEN. To design such networks in general mathematical programming approaches are implemented. In this research evolutionary optimization approach is used to design HEN, while optimization of the operating conditions in a given HEN is carried out by simultaneous optimization of the capital cost and the operating cost via differential evolution optimization approach using modular sizes of heat exchangers. A super-structure based representation of HEN proposed by Yee and Grossmann [40] is used in the current study to develop HEN. A representation of the super-structure for a two hot and two cold streams for a two stage problem is shown in Figure 2.1. In this structure, there are a maximum of eight heat exchangers, with maximum of four heat exchangers in each stage. The assumptions in this structure formulation are splitting of streams at each stage are allowed, isothermal mixing of streams at the end of each stage, not more than one heat-exchanger is permissible on each split stream and utilities are present at the end of the super structure.

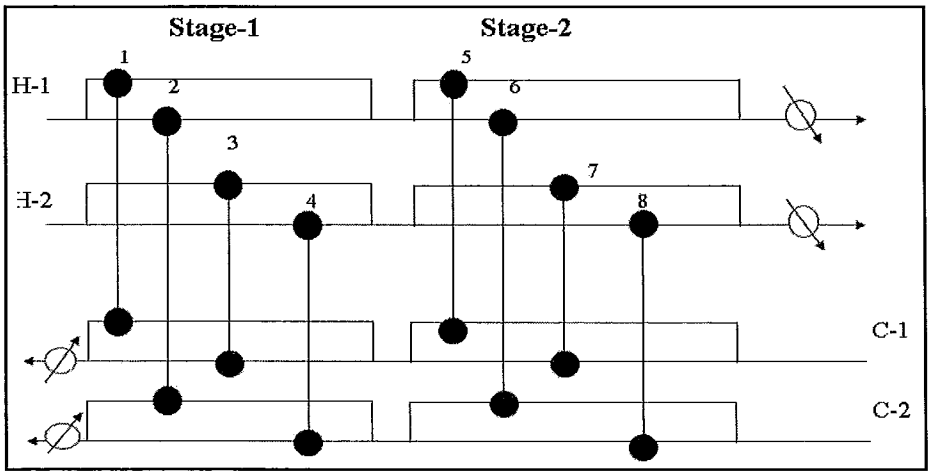


Figure 2.1: An example for two stage superstructure HEN developed based on Yee and Grossman [40].

2.2 HEN Modeling and Synthesis

Heat Exchanger Modeling and Network Solution Algorithms

In published literature, there are many approximate models addressed to eliminate LMTD calculations in heat exchanger model that cause a nonlinear behavior and may lead to indeterminable solutions at times due to division by zero. Approximation by Paterson [42] represents LMTD in terms of geometric mean temperature difference (GMTD) and arithmetic mean temperature difference (AMTD).

$$AMTD = \frac{1}{2}(\theta_1 + \theta_2) \tag{2.1}$$

$$GMTD = (\theta_1 \theta_2)^{1/2} \tag{2.2}$$

$$\theta_{LM} = \frac{2}{3} GMTD + \frac{1}{3} AMTD \tag{2.3}$$

Where,

θ_1 - Temperature difference between the two streams at the hot end

θ_2 - Temperature difference between the two streams at the cold end

Another approximation to LMTD is Chen's approximation [7] in this LMTD is expressed as power function in terms of temperature differences of the hot and cold streams.

$$\theta_{CM} = \left(\frac{1}{2}\theta_1^{0.3275} + \frac{1}{3}\theta_2^{0.3275} \right)^{1/0.3275} \tag{2.4}$$

These approximations methods have been extensively used in the design, monitoring and optimization. Since, these models are approximate they may lead to results that are not quite accurate.

Completely overlooked is a reformulation of a heat exchanger model proposed by Goyal [32]. This method is based on the fact that the new heat duty of a heat exchanger can be expressed in terms of the base heat duty conditions and the maximum temperature range of the hot and cold streams (inlet temperature of the hot stream and inlet temperature of the cold stream) thus eliminating the LMTD factor by introducing a factor which is a ratio of the heat exchanger duty at new conditions vs. heat exchanger duty at base conditions (so called Φ factor). This model is rigorous and linear in temperatures if the flows are known.

Heat Exchanger Network Simulation Algorithms

In earlier studies, there are several simulation algorithms proposed for solving HEN simultaneously, where the mass and energy balance equations of the HEN are represented

through a set of linear equations with respect to mass flow rate and the temperature of the streams.

One of the earlier works in linear formulation of HEN was presented by R.Ratnam.et al. [34]. In this approach, nonlinear nature of the LMTD in the design equation of a heat exchanger is eliminated by algebraic manipulations to derive the relationship between the outlet and the inlet temperatures in linear form. This was extended to sensitivity analysis for HEN, The approach proved to be a non-iterative approach for case studies which resulted in variation in single variable / parameter, but in the cases of simultaneous change in several variables/ parameters it resulted in an iterative approach to converge.

Recently, another simulation method designed based on NTU method (number of transfer units) was proposed by Luiz O. et al [25]. In this approach the network structure and the equations related to the process system are represented in matrix form, consisting of mass balance and energy balance equations. This method was extended to steady-state simulation of HEN and for evaluation of heat transfer coefficients.

Heat Exchanger Network Synthesis

Synthesis of heat exchanger networks has been an extensively explored areas of research in process systems engineering over the past three decades. Recently, a detailed review on the HEN synthesis methodologies was presented by [22]. In this section, brief overview of the major contributions in HEN synthesis and optimization, which made a significant impact and provided an impetus to search for new optimization algorithms to improve the optimization of HEN are presented.

The major areas of research in this field can be classified into two groups, sequential synthesis and simultaneous synthesis methods. The sequential synthesis methodology is based on pinch analysis [1]. In this method the HEN problem is divided into sub

problems by dividing the temperature range of the problem into temperature intervals while obeying the thermodynamic principles. The simultaneous synthesis methods are based on mathematical programming of models as non-linear, mixed integer linear and non-linear models [30]. In these methods the optimization problem is solved simultaneously without decomposition into sub problems to find the optimal HEN network.

Sequential Synthesis of HEN

In sequential synthesis method three sub problems are solved intermittently in the order of, minimum utility cost, minimum number of heat exchanger units in the network and minimum investment / capital cost of the network. Mathematical programming approach to solve these sub problems is implemented in three stages, In the first step; the minimum utility cost problem is solved for a given pinch point ΔT_{min} as a linear programming problem (LP) based on the LP transshipment model proposed by Papoulias and Grossmann [38]. In the second step, minimum number of heat exchanger units in the network problem is solved based on an MILP formulation proposed by Papoulias and Grossmann [38] to determine the minimum number of units. In the third step, a nonlinear programming model is implemented to determine the minimum capital cost of the network subject to heat load distribution and heat exchanger matches obtained in the previous step.

The main drawback in sequential approaches of HEN synthesis are the trade-off between the utilities, the number of units and the area of the heat exchangers are not considered in an integrated approach. In addition, decomposition of the problem into sub-problems based on the pinch analysis and optimization of these sub-problems lead to sub-optimal networks. Therefore, the necessity to solve the HEN synthesis problem simultaneously in single-take is required.

Synthesis via Simultaneous Solution of a Superstructure

To solve HEN synthesis problems simultaneously super-structure based MINLP formulation was first proposed by Yuan et al 1989 [43], the major drawback in this approach was splitting of the streams was not allowed.

In 1989, 1990 Floudas and Ciric [4, 6] proposed an MINLP formulation of HEN based on "hyperstructure"; in this approach the second and the third steps discussed earlier are solved simultaneously to determine the capital cost of the network. This formulation is developed by combining the transshipment model for match selection proposed by Papoulias and Grossmann [38] and "hyperstructure" model for minimum capital cost network proposed by Floudas and Grossmann [5] to calculate the temperatures, area of the heat exchangers and the flow rates .The advantage of this approach is it does not require any decomposition of the problem and simultaneously optimize the network for stream matches and network configuration.

Some of the uncertainties to the optimality of the final network with the sequential three-task procedure are also presented in the study by [4]. Uncertainties in the network optimization stage, where possibility of local optimum solutions may arise due to the nonlinear, nonconvex nature of the optimization problem, to address this issue a global optimum search approach is proposed [4]. Uncertainties in the selection of matches stage, there are several feasible solutions; to address this "hyperstructure" based optimization approach is presented [4].

A stage-wise super structure MINLP formulation of the HEN network was proposed by Yee and Grossmann [40]. The structure consists of all the possible matches between the hot and cold streams and this is repeated in each stage of the network. Assuming isothermal mixing of the streams, maximum of one exchanger for each match, utilities are placed at the end of super-structure and stream bypasses are not allowed. This formulation of HEN super-structure became a bench-mark for the simultaneous synthesis and optimization problems.

Pettersson F. [15] developed a simultaneous MILP formulation of HEN synthesis based on piece-wise linearization of geometric mean temperature difference (GMTD) to eliminate logarithmic mean temperature difference (LMTD) and thus the non-linearity in the evaluation of the heat transfer area. The disadvantage of this approach is, for small size synthesis problems the annual cost of the HEN is fairly close to the solution obtained with LMTD approach but in cases of large size problems the solution obtained with GMTD underestimates the solution obtained with that of LMTD.

Massimiliano et al. [29] proposed an incidence matrix representation for HEN developed based on the superstructure and implemented a mixed integer programming algorithm for the synthesis of maximum energy recovery from HEN. Similar form of HEN representation was used by Krishna .M.Y [23] to synthesize HEN using differential evolutionary optimization.

Evolutionary Optimization Methods for HEN Synthesis

One of the early implementations of evolutionary optimization methods in HEN synthesis was implemented by G.Athier et al. [16]. In this approach simulated annealing (evolutionary optimization strategy) was used for combinatorial optimization to determine feasible network structures i.e. the binary variables and the continuous variables (flows, temperatures) of the HENs are optimized by NLP for each network obtained from previous step. This work lead several researchers to look into evolutionary optimization methods for HEN synthesis to overcome the difficulties resulted from traditional optimization methods.

Hongmei Yu et al [17] combined genetic algorithms (GA) and simulated annealing for synthesis of large scale HENs. The traditional GA was improved by including orthogonal crossover (OC) and effective crowding (EC) factor and was integrated with simulated annealing to optimized large heat exchanger networks. This approach proved very

promising when compared to other traditional optimization methods and evolutionary optimization methods.

Ravagnani M.A.S.S [30] developed a methodology for HEN synthesis using evolutionary optimization methods, such as genetic algorithms. In this approach, initially the optimum pinch point (ΔT_{min}) is to be determined. As ΔT_{min} increases, there is an increase in energy demand and decrease in heat exchanger area, so an optimum value of ΔT_{min} is determined by minimizing energy cost and capital cost using genetic algorithms. In the second stage based on the pinch point the problem is divided into two networks (above the pinch and below the pinch) and it is optimized to determine the minimum global cost of the network.

Krishna .M.Y [23] proposed a simultaneous synthesis of HEN networks using differential evolution optimization approach to optimize the heat exchanger networks. The optimizing variables are the network structure, heat loads of the heat exchangers, split fractions of the hot and cold streams and ΔT_{min} minimum approach temperature a user specified input parameter. For the HEN structure representation in this work, string representation of HEN is used in this work.

The above work proved to be functioning well for standard HEN synthesis problems. This framework of HEN synthesis is extended in the current study to synthesis of HEN networks with modular sizes of heat exchangers. Differential Evolution (DE) algorithm has proved to function better than genetic algorithms because the DE operates on floating point numbers and GA operates on binary code which is later decoded in terms of the defined bounds of the optimizing variable, this leads to huge amount of memory usage for storing a variable information, in addition to handle negative numbers in binary form becomes a complicated task [3].

2.3 Differential Evolution Optimization Algorithm

In traditional optimization or gradient based optimization methods for multi-modal functions there is always a challenge to find the global optimum [21]. In these methodologies at certain times based on the starting point and the degree of non-linearity of the objective function and the constraints there is a likeness that these methods get trapped in local optimum. Recently to overcome these difficulties non-traditional optimization methods that are gradient free and population based (multiple starting points) methods such as genetic algorithms (GA) [18, 14], simulated annealing [39] and differential evolution DE [21] have been caching up the interest of many researchers in nonlinear optimization in various domains.

Evolutionary optimization methods are stochastic in nature, with probabilistic transition rules [21]. These methods mimic the Darwinian evolution principle and try to generate better solutions by recombination, mutation and survival of the fittest. The advantages of evolutionary optimization methods are they do not require any special conditions or constraints on the objective function and the constraints of the problem; they are applicable for both continuous and combinatorial optimization problems and multi-objective optimization. The disadvantages of these methods are, they require huge computational time. In the current research DE is used for optimization, the algorithm for DE is mentioned below.

DE Algorithm adapted from [41]

The optimization problem is formulated as,

$$\begin{aligned} &\min_{X} f(X) \\ &S.T \quad L \le X \le H \\ &\qquad X \in IR^{D} \end{aligned} \tag{2.1}$$

Let the optimum solution for this optimization problem be represented by X^*, this is a vector of optimizing parameters $x_i^* = 1,....,D$ such that the values of these optimizing parameters obey the boundary constraints $L \leq X \leq H$. The objective function $f(X)$ is a scalar function that is to be minimized or maximized.

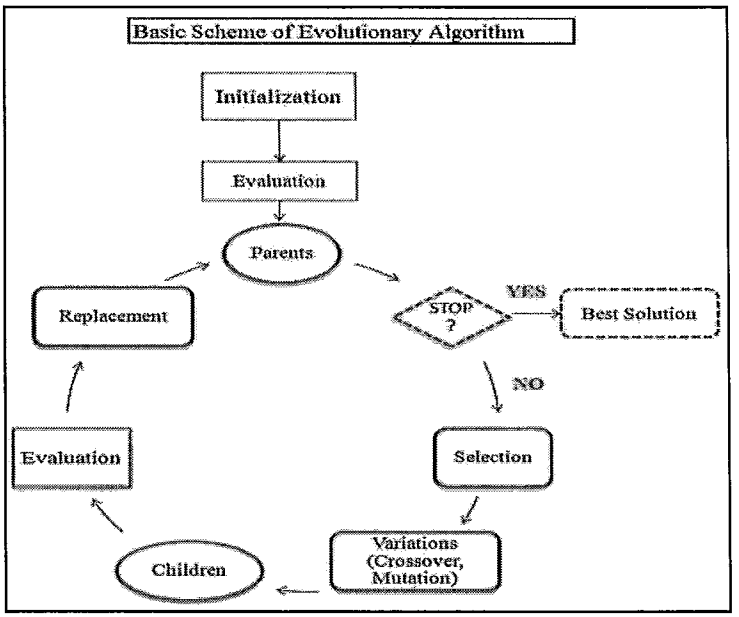


Figure 2.2: A simplified flowchart of the DE algorithm

In the first step of this optimization, initial population members (NP) of the optimizing variables are generated by a random floating-point number such that $L \leq X \leq H$ covering the entire range of the boundary constraints. Initial population members are termed at target population members. These generated population members are evaluated or the fitness of the objective function is evaluated. Then the optimization routine starts and performs the following operations until a maximum number of iterations termed as generations are attained or terminating criteria is reached.

1. In this iterative process, the first step is to select 3 individuals randomly from the target population that are mutually different from each other and the current individual (j).

$$r_{1,2,3} \in [1,\ldots..,NP], r_1 \neq r_2 \neq r_3 \tag{2.2}$$

2. Based on the above randomly selected individuals a trial vector is generated according to the probabilistic rule

$$x_i = \begin{cases} x_{i,r_3} + F.(x_{i,r_1} - x_{i,r_2}).....if(rand_{ij}[0,1] < CR) \forall (rand = i) \\ otherwise \\ x_{i,j} \end{cases}$$
$$i = 1,....,D \tag{2.3}$$

Where, F- Mutation factor, CR – Crossover factor

3. The above generated trial vectors are checked for any constraint violations of the optimizing variables. Such entries are corrected to bring them back into feasible boundary constraints.

$$if(x_i \notin [L,H])$$
$$x_i = L + (H - L).rand_i[0,1] \tag{2.4}$$

4. The best population members for the next generation or iteration are selected in the final step. In this step, first the fitness of the trial population members generated in step-3 is evaluated. Then the fitness function of the trial and the target population are compared, if objective function value of the trial vector is less than that of the target then the target population member is replaced with the trial population member.

The above four steps are repeated until maximum number of generation or termination criteria is reached. The different control parameters in this algorithm are mutation factor (F), crossover factor (CR) and size of population NP and the terminating criteria is maximum number of generations (GEN).

To understand how the control parameters affect the optimization of the problem and the convergence rate, an explanation for the same is mentioned in [41]. In the step-2 of the DE the scaled difference of the two randomly chosen individuals is added to a third vector or the current reference point. In principle, the scaled difference of two randomly chosen vectors gives the direction and step length of the search direction (in comparison to gradient based methods). Generally, the mutation factor (F) acts like a tuning parameter that decides the step length, and "manages a trade-off between exploitation and exploration of the space" [41].

The principle use of the mutation step is to adjust the step length by a factor in the search space. This is evident from the functioning of DE, in the initial generations the step length will be very large because the population members are spread over the entire region but as the DE progresses and the population members converge, the step length becomes very small. Thus the randomness of the search direction and the base point or reference point ensures global optimum for a given optimization problem [41].

The crossover probability (CR) ensures that at least one parameter will be changed in the dimension 1 to D (dimension of the optimizing variables) by the random number generator in every generation.

Handling Binary Variables and Continuous Variables in DE

DE algorithm originally functions on continuous floating-point values. This is extended to handle binary variables by a simple modification to DE algorithm proposed by Corne [13, 33].

$$y_i = \begin{cases} x_i \\ \text{int}(x_i) \end{cases} \tag{2.5}$$

$\text{int}(x_i)$ denotes the integer value of the variable, this operation is done only for evaluation of target and trial population members but these values are not assigned anywhere, to ensure that the DE always runs on continuous variables. In the present study this representation of binary variables is used for developing different HEN structures during HEN synthesis via DE. An algorithm for DE implementation is presented in Figure 2.3.

Algorithm(*V. Feoktistov, Differential Evolution)*
Generation $g \leftarrow 0$
Population $IP^g \leftarrow$ *Initialize*
Fitness $f(IP^g) \leftarrow$ *Evaluate*

while (not stopping condition) do
//Proceed to the next evolutionary cycle//
$g \leftarrow g+1$
Parents ← Select from
Children ← Vary Parents
(Crossover, Mutation,....)
Fitness ← Evaluate Children
Replacement IP^g ← Survive Parents & Children
end while

Figure 2.3: Algorithm of DE adapted from [41]

Chapter 3

HEN Modeling and Simulation Algorithm

In this chapter, LMTD methods and it approximation and methods are discussed followed by Φ factor formulation for modeling heat exchanger models. Next, the HEN simulation algorithm developed based on the Φ factor formulation is presented followed by the illustration of HEN simulation algorithm through an example problem and representation of equations phase wise in matrix form.

3.1 LMTD and Its Approximations

In the literature, several approximations methods are proposed to eliminate LMTD calculations to solve and evaluate the area of the heat-exchangers for counter-current flow configuration. The assumptions considered for deriving LMTD method are, no heat losses from the heat exchanger, constant specific heat capacity of the streams and constant overall heat transfer coefficient and the heat exchanger is at steady state conditions. A detailed review on the derivation of the LTMD is presented in [35].

The major concern with LMTD based simulation approaches is handling the division by zero. Therefore, solving optimization problems with LMTD may lead to indeterminable solutions.

LMTD (θ_{LM}):

$$\theta_{LM} = \frac{\theta_1 - \theta_2}{\ln\left(\theta_1 / \theta_2\right)} \tag{3.1}$$

Where,

θ_1 - Temperature difference between the two streams at the hot end

θ_2 - Temperature difference between the two streams at the cold end

Paterson Approximation [42] (θ_{PM})

It is based on the fact that LMTD is bounded by arithmetic mean temperature difference (AMTD) and geometric mean temperature difference (GMTD).The expression for Paterson approximation [42] is as mentioned below

$$AMTD = \frac{1}{2}(\theta_1 + \theta_2) \tag{3.2}$$

$$GMTD = (\theta_1 \theta_2)^{1/2} \tag{3.3}$$

$$\theta_{LM} = \frac{2}{3} GMTD + \frac{1}{3} AMTD \tag{3.4}$$

Chen Approximation: [7] (θ_{CM})

$$\theta_{CM} = \left(\frac{1}{2}\theta_1^{0.3275} + \frac{1}{3}\theta_2^{0.3275}\right)^{1/0.3275} \tag{3.5}$$

In earlier studies both Paterson and Chen approximations have been extensively used for HEN simulation, design and optimization. A detailed review of these approaches is mentioned in [15]. Paterson and Chen approximations are used for evaluating the heat

exchanger network performance [40]. AMTD has been used for HEN synthesis [19]. Linearized form of GMTD is used for HEN synthesis using a MILP approach [15]. In these approaches, the approximate models are non-linear in nature which may lead to inconsistent results from HEN synthesis and optimization.

In the literature, another method called Φ factor, was proposed by Om.P.Goyal [32] to eliminate the LMTD calculations in a heat exchanger. The details of the formulation of Φ factor are discussed in later sections. This method was developed based on the fact that the new heat duty of a heat exchanger can be expressed in terms of the base heat duty conditions and the maximum temperature range of the hot and cold streams and eliminating the LMTD factor by introducing the Φ factor. This method has been least addressed for HEN design and optimization to the best knowledge of the authors. In the current research study, the heat exchanger model is formulated based on the Φ factor.

3.2 Reformulated Heat Exchanger Model

In the current study, Φ factor formulation proposed by Om. P. Goyal [32] is used for the development of heat exchanger model. The basis for this methodology is that if the base operating conditions of a heat exchanger process are known then the LMTD calculations can be eliminated by expressing LMTD in terms of the maximum temperature span, hot and cold temperature conditions of the process streams of heat exchanger by introducing an external parameter, Φ factor. Derivation of the Φ factor is present in O. P. Goyal [32].

This methodology is used for heat exchangers that have counter-current flow configuration and with the assumptions that there is no phase change in the fluid flow, no changes in the specific heat capacity of both hot and cold streams,

Goyal proposed that a countercurrent heat exchanger be described by the following equations:

$$Q_h = m_h Cp_h (Th_i - Th_o) \quad (3.6)$$

$$Q_c = m_c Cp_c (Tc_o - Tc_i) \quad (3.7)$$

$$Q' = \Phi * Q_0 \quad (3.8)$$

Where,

Q_h, Q_c - Heat transfer rate of the hot and cold stream respectively.

m_h, m_c - Mass flow rate of the hot and cold stream respectively

Cp_h, Cp_c - Specific heat capacity of the hot and cold stream respectively

Th_i, Th_o - Inlet and outlet temperature of the hot stream respectively

Tc_i, Tc_o - Inlet and outlet temperature of the cold stream respectively

Q_o, Q - Heat transfer rate at the base condition and new condition respectively.

Φ - Φ factor (Heat duty ratio)

In the above equations, the first two equations represent the heat transfer equations of a heat exchanger on the hot side and the cold side respectively. In the third equation, the new heat duty (**Q`**) of a heat exchanger is expressed in terms of the base heat duty ($\mathbf{Q_0}$) conditions and the Φ factor. The Φ factor depends on the new flow rate conditions, area and temperatures of the heat exchanger. The major advantage of the reformulated heat exchanger model is that if the flows and the area of the heat exchanger are known then the above equations are linear in terms of heat duty and the temperatures. The reformulated heat exchanger model is extended to develop a simulation algorithm for HEN simulation, where the mass and energy balance equations of the network are represented through a set of linear equations with respect to mass flow rates and temperature of the streams. The details of which are discussed in the following section.

3.3 Φ Factor Formulation

The key aspect of this method is, the Φ-factor takes into account the change in all the variables mass flow rate, temperature and the parameters of a heat exchanger the heat transfer coefficient and area of the heat exchanger. The derivation of Φ-factor is given in Appendix-A.

$$\Phi_{flow} = \frac{\left(\frac{\Delta t_1}{\Delta t_2}\right)^X - 1}{\left(\frac{\Delta t_1}{\Delta t_2}\right)^X \left(\frac{m_h}{m'_h}\right)\Delta t_h - \left(\frac{m_c}{m'_c}\right)\Delta t_c} \tag{3.9}$$

$$X = \frac{\left(\frac{U'}{U}\right)\left(\frac{A'}{A}\right)\left[\left(\frac{m_h}{m'_h}\right)\Delta t_h - \left(\frac{m_c}{m'_c}\right)\Delta t_c\right]}{(\Delta t_h - \Delta t_c)} \tag{3.10}$$

$$\Phi = \Phi_{flow} * \Delta t'_{\alpha} \tag{3.11}$$

Where,

(`)	- Indicates value at new equilibrium
A, A`	- Heat Transfer Area
Cp_h, Cp_c	- Specific heat of hot and cold streams
Φ	- Φ factor
m_h, m`$_h$	- Mass flow rate of the hot stream
m_c, m`$_c$	- Mass flow rate of the cold stream
U,U`	- Overall heat transfer coefficient
t_{c1}, t`$_{c1}$	- Temperature of cold stream at the inlet
t_{c2}, t`$_{c2}$	- Temperature of cold stream at the outlet

t_{h1}, $t`_{h1}$ - Temperature of hot stream at the inlet

t_{h2}, $t`_{h2}$ - Temperature of hot stream at the outlet

Δt_c, $\Delta t`_c$ - Temperature range of the cold stream

Δt_h, $\Delta t`_h$ - Temperature range of the hot stream

Δt_1, $\Delta t`_1$ - Temperature approach at the hot end

Δt_2, $\Delta t`_2$ - Temperature approach at the cold end

Δt_α, $\Delta t`_\alpha$ - Maximum temperature span

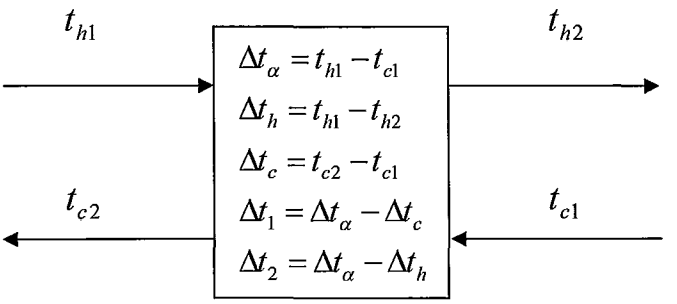


Figure 3.1: Heat exchanger model with the details of temperatures differences.

If we assume that the heat transfer coefficient does not change with the flow rate, then factor Φ dependence on the flows of hot and cold streams is as shown in Fig. 3.2 and Fig. 3.3.

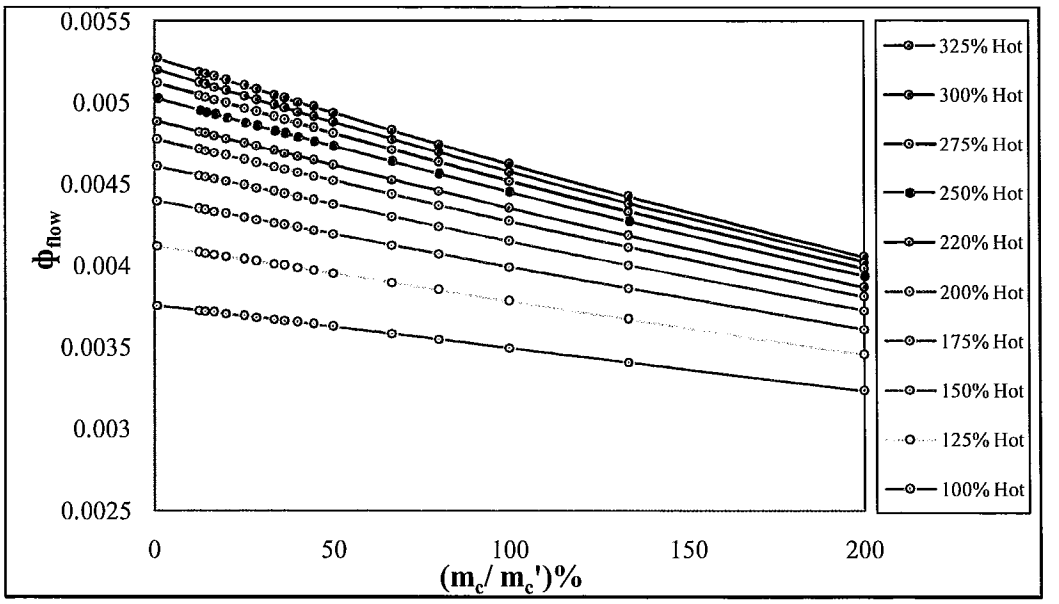


Figure 3.2 Φ_{flow} vs $\left(m_c / m_c'\right)\%$ with change in mass flow rate of cold and hot streams, with constant U and constant A.

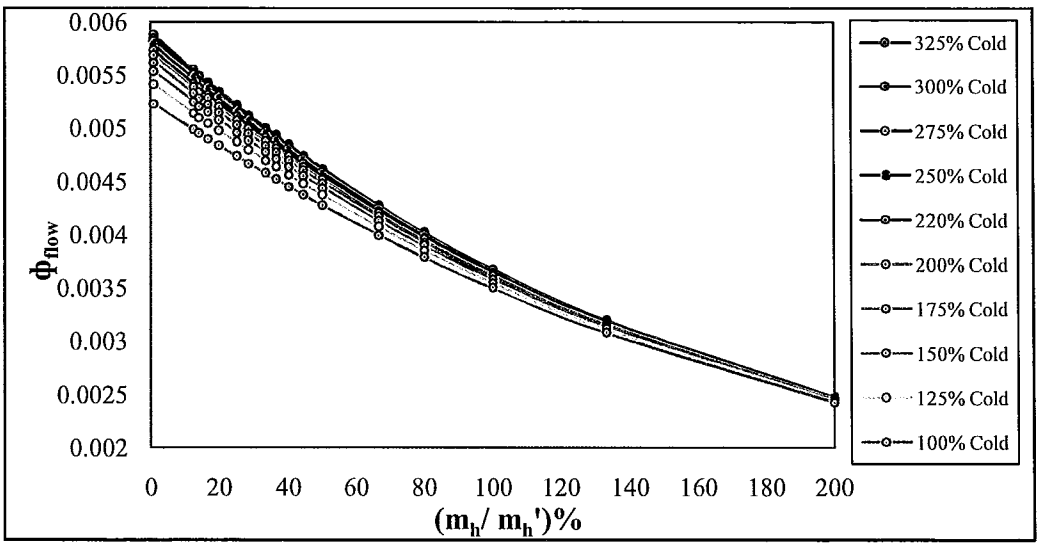


Figure 3.3: Profile for Φ_{flow} vs $\left(m_h / m_h'\right)\%$ with change in mass flow rate of cold and hot streams with constant U and constant A.

If heat exchanger coefficient U varies with the flows (i.e for cases when there are large changes in flows), we can assume that the changes in U can be expressed via equation 3.9 [32]. Then U' dependence on the flows of hot and cold streams is as shown in Fig. 3.3 and Fig. 3.4.

$$U' = \frac{U}{1 + \frac{U}{h_{io}}\left[\left(\frac{m_t N_p'}{m_t' N_p}\right)^{0.8} - 1\right] + \frac{U}{h_o}\left[\left(\frac{m_s}{m_s'}\right)^{0.6} - 1\right]} \tag{3.12}$$

Where,

(`)	- Indicates value at new equilibrium
$U, U`$	- Overall heat transfer coefficient
h_{io}	- Inside film heat transfer coefficients
h_o	- Outside film heat transfer coefficients
m_t, m_t'	- Tube side mass flow rate (hot stream flow rate)
m_s, m_s'	- Shell side mass flow rate (cold stream flow rate)
N_p, N_p'	- Tube side number of passes

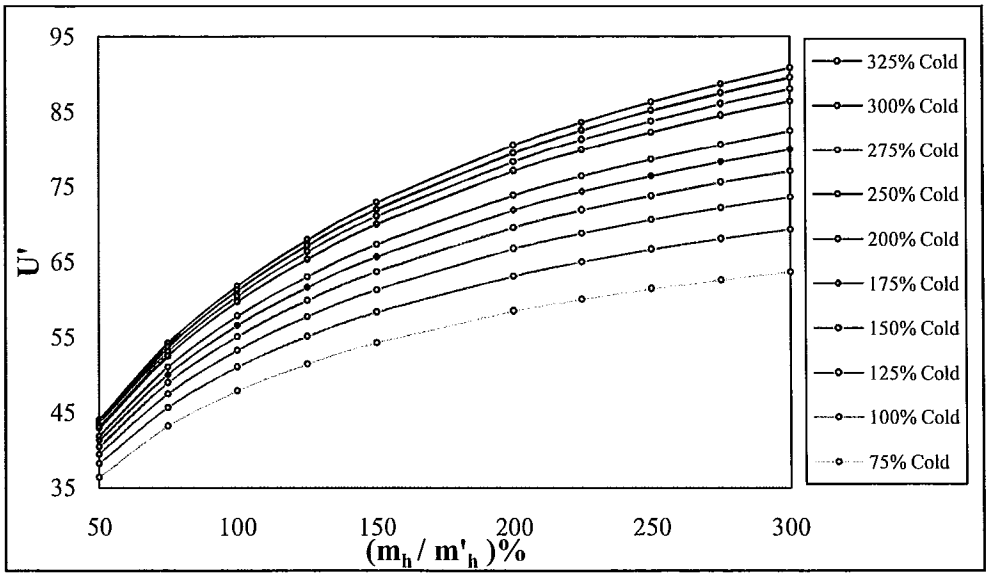


Figure 3.4: Profile for U' vs $\left(m_h / m_h'\right)\%$ with change in mass flow rate of cold and hot streams

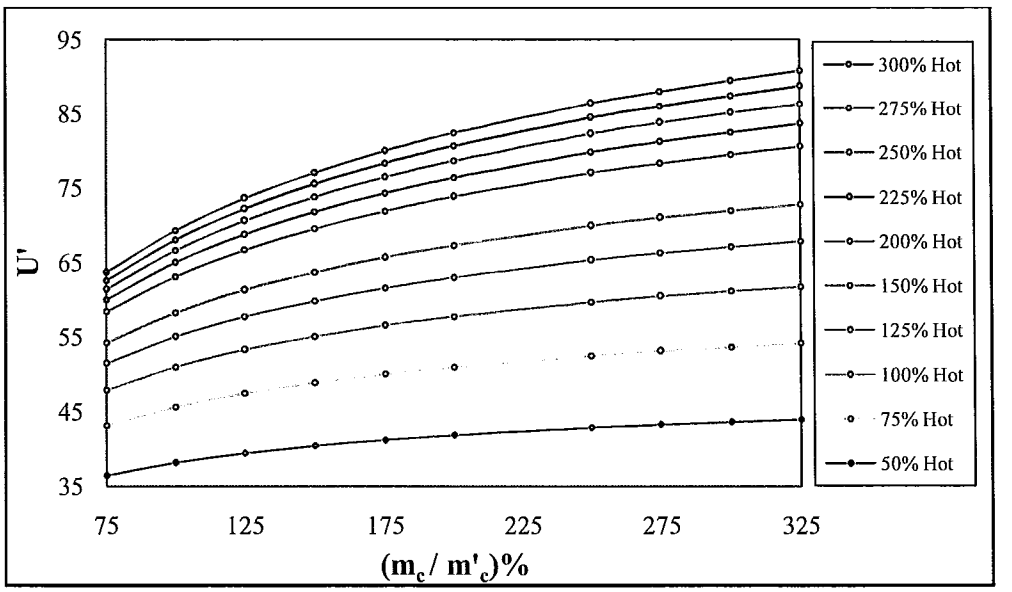


Figure 3.5: Profile for U' vs $\left(m_c / m_c'\right)\%$ with change in mass flow rate of cold and hot streams

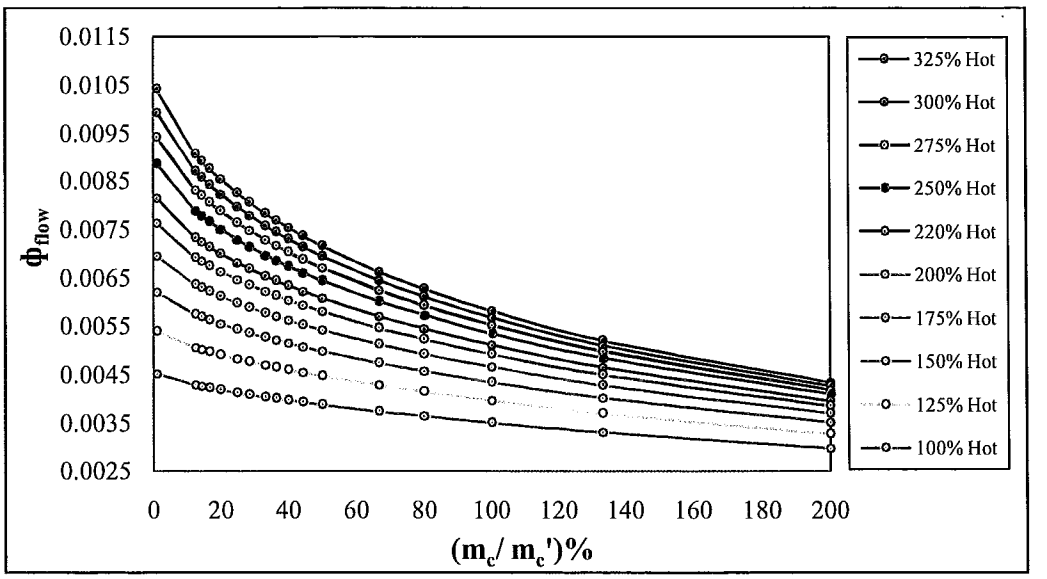


Figure 3.6 Φ_{flow} vs $\left(m_c / m_c'\right)\%$ with change in mass flow rate of cold and hot streams, change in U` and constant A.

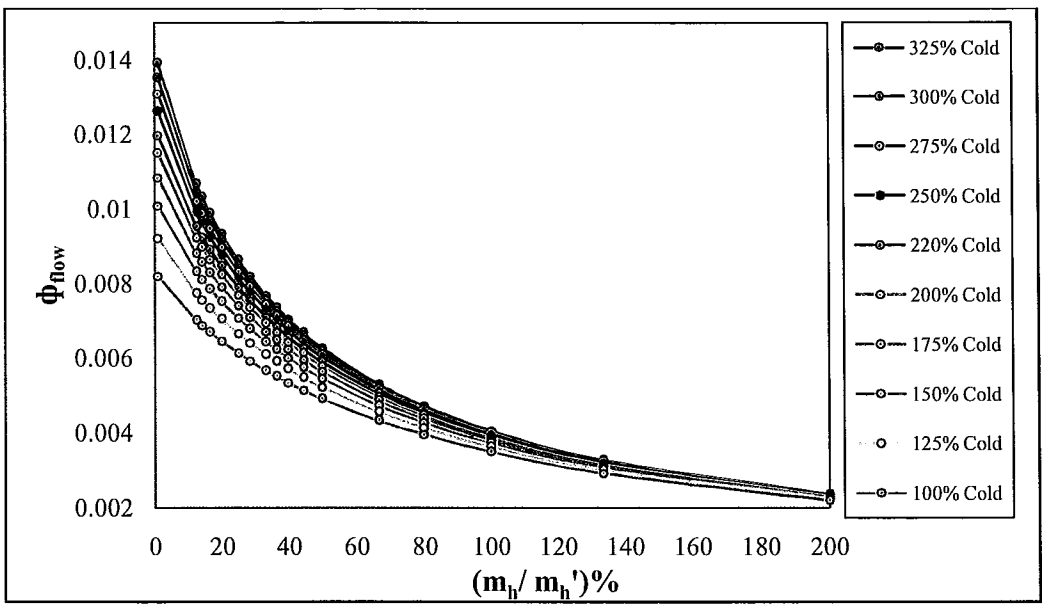


Figure 3.7 Φ_{flow} vs $\left(m_h / m_h'\right)\%$ with change in mass flow rate of cold and hot streams with change in U` and constant A.

3.4 Two Phase Algorithm for HEN Simulation

In the current research a non-iterative, robust algorithm for HEN simulation is developed based on the Φ-factor formulation and reformulated heat exchanger model as discussed earlier. The mass and energy balance equations of the model are represented in matrix form. In the network structure, the connectivity equations between the equipments / nodes are established by a connector node. For a specified inlet mass flow rate, temperature conditions and heat exchanger parameters, the resulting network can be solved to determine the flows and the temperatures across the network.

The details of the methodology are organized as follows. Section 3.4 details of the individual models/ equipments of the network topology. Section 3.5 to build the HEN equations in a matrix form through network connections using HEN simulation algorithm.

Details of the Individual Models of the Network Topology

In this section, the individual models / equipments used for developing a HEN are described. The HEN model is solved in 4 phases. Equation types solved in each phase are:

- **Mass Balance**: The first phase is to solve linear mass balance and volumetric balance equations to determine the mass flow rate across the HEN.
- **Propagate Phase**: The purpose of this phase is to initialize stream physical properties (specific heat capacity, density) across all streams in the network.
- **After Mass Balance** – This phase involves the calculation of intermediate and nonlinear function variables (mc_p, Φ_{flow}).
- **Energy Balance** - The final phase involves solving the linear energy balance equations to determine the temperatures and heat duties of the exchanger across the entire network, in this step the set of equations representing the energy model are linear with respect to the temperatures as the flow rates are determined in the first step.

3.4.2 Heat Exchanger

Mass Balance Equations for a Heat Exchanger

The equations describing the mass balances around a heat exchanger are listed below

$$m_{c,i} - m_{c,o} = 0 \tag{3.13}$$

$$m_{h,i} - m_{h,o} = 0 \tag{3.14}$$

Where,

$m_{c,i}$	Mass flow-rate of cold inlet stream
$m_{c,o}$	Mass flow-rate of cold outlet stream
$m_{h,i}$	Mass flow-rate of hot inlet stream
$m_{h,o}$	Mass flow-rate of hot outlet stream

Propagate Phase Equations for a Heat Exchanger

The equations describing the propagate phase equations across a heat exchanger on hot and cold side respectively are represented as below

$$Cp_{h,i} - Cp_{h,o} = 0 \tag{3.15}$$

$$Cp_{c,i} - Cp_{c,o} = 0 \tag{3.16}$$

Where,

$Cp_{c,i}$	Specific heat capacity of cold inlet stream
$Cp_{c,o}$	Specific heat capacity of cold outlet stream
$Cp_{h,i}$	Specific heat capacity of hot inlet stream
$Cp_{h,o}$	Specific heat capacity of hot outlet stream

After-Mass Balance Phase Equations a for Heat Exchanger

Intermediate calculations between the mass balance phase and energy balance phase are implemented in this phase. Φ_{flow} and product of mass flow rate and specific heat capacity of inlet streams ($m_h Cp_h = C_h$, $m_c Cp_c = C_c$) to heat exchangers are calculated in this phase.

Energy Balance Phase Equations for a Heat Exchanger

The equations describing the energy balance phase equations across a heat exchanger on hot and cold side respectively are represented as below

$$Q_h = C_h(T_{h1} - T_{h2}) \tag{3.17}$$

$$Q_c = C_c(T_{c2} - T_{c1}) \tag{3.18}$$

$$\Phi = \Phi_{flow} * (T_{h1} - T_{c1}) \tag{3.19}$$

$$Q_{new} = \Phi * Q_{base} \tag{3.20}$$

Where,

C_h, C_c - Heat capacities of hot and cold streams

Φ - Phi factor

Φ_{flow} - Phi factor with respect to change in flows only

Q_h, Q_c - Heat transfer rate of hot and cold stream respectively.

T_{c1}, T_{c2} - Temperature of cold stream at the inlet and at the outlet respectively

T_{h1}, T_{h2} - Temperature of hot stream at the inlet and at the outlet respectively

3.4.3 Mixers

Mass Balance Equations for a Mixer

The mass balance equations for a mixer are valid for both hot and cold process streams.

$$m_{i,1} + m_{i,2} + m_{i,3} \ldots\ldots - m_{o,1} = 0 \tag{3.21}$$

Where,

$m_{i,1}, m_{i,2}, m_{i,3}$	Mass flow-rate of inlet streams
$m_{i,o}$	Mass flow-rate of outlet stream

Propagate Phase Equations for a Mixer

The equations representing the propagation phase across a mixer are represented as mentioned below. Mass averaging specific heat capacity of each stream to calculate specific heat capacity of the outlet stream

$$m_{i,1}Cp_{i,1} + m_{i,2}Cp_{i,2} + m_{i,3}Cp_{i,3} \ldots\ldots - m_{o,1}Cp_{o,1} = 0 \tag{3.22}$$

Where,

$Cp_{i,1}, Cp_{i,2}, Cp_{i,3}$	Specific heat capacity of the inlet streams
$Cp_{o,1}$	Specific heat capacity of the outlet stream

Energy Balance Equations for a Mixer

The energy balance equations for a mixer are represented as mentioned below.

$$m_{i,1}Cp_{i,1}T_{i,1} + m_{i,2}Cp_{i,2}T_{i,2} + m_{i,3}Cp_{i,3}T_{i,3} \ldots\ldots m_{o,1}Cp_{o,1}T_{o,1} = 0 \tag{3.23}$$

Where,

$T_{i,1}, T_{i,2}, T_{i,3}$	Temperature of the inlet streams
$T_{o,1}$	Temperature of the outlet stream

3.4.4 Splitter

Mass Balance Equations for a Splitter

The mass balance equations for the splitter model are as mentioned below

$$m_{i,1} - m_{o,1} - m_{o,2} - m_{o,3} = 0 \tag{3.24}$$

$$\alpha_1 * m_{i,1} - m_{o,1} = 0 \tag{3.25}$$

$$\alpha_2 * m_{i,1} - m_{o,1} = 0 \tag{3.26}$$

Where,

$m_{i,1}$	Mass flow-rate of inlet stream
$m_{o,1}$, $m_{o,2}$, $m_{o,3}$	Mass flow-rate of outlet streams
α_1 , α_2	Split fractions of the stream splits

Propagate Phase Equations for a Splitter

The equations representing the propagation of specific heat capacity across a splitter are represented as mentioned below

$$Cp_{i,1} - Cp_{o,1} = 0 \tag{3.27}$$

$$Cp_{i,1} - Cp_{o,2} = 0 \tag{3.28}$$

$$Cp_{i,1} - Cp_{o,n} = 0 \tag{3.29}$$

Where,

$Cp_{i,1}$	Specific heat capacity of the inlet stream
$Cp_{o,1}$ $Cp_{o,2}$, $Cp_{o,n}$	Specific heat capacity of the outlet stream

Energy balance equations for a splitter

The energy balance equations for the splitter model are as mentioned below.

$$T_{i,1} - T_{o,1} = 0 \tag{3.30}$$

$$T_{i,1} - T_{o,2} = 0 \tag{3.31}$$

$$T_{i,1} - T_{o,n} = 0 \tag{3.32}$$

Where,

$T_{i,1}$	Temperature of the inlet stream
$T_{o,1}, T_{o,2}, T_{o,3}$	Temperature of the outlet streams

3.4.5 Connector Node

In order to achieve high modularity of the software, each node and its associated streams are represented as a stand-alone entity. In other words, all stream variables "belong" to a node that the stream is connected to. Therefore, variables for any given stream appear in its origin node and in its destination node. A simplified name of a variable may be:

Origin_Node.Stream_S.Variable_V1 (3.33)

Destination_Node.Stream_S.Variable_V1 (3.34)

Where,

Origin_Node / Destination_Node: denotes the name of the node

Stream_S: denotes name of the stream connected to a node.

Variable_V1: denotes the name of the stream variable.

Purpose of the connector node is to equate above two variables, since they represent the same entity in the same stream. Such methodology of connecting nodes in an equation oriented modeling is a common practice in the software industry [27]

Mass Balance Equations for a Connector

The mass balance equations for the connector are as mentioned below

$$m_{i,1} - m_{o,1} = 0 \tag{3.35}$$

Where,

$m_{i,1}$ Mass flow-rate of the inlet stream

$m_{o,1}$ Mass flow-rate of the outlet stream

Propagate Phase Equations for a Connector

The mass balance equations for the connector are as mentioned below

$$Cp_{i,1} - Cp_{o,1} = 0 \tag{3.36}$$

Where,

$Cp_{i,1}$ Specific heat capacity of the inlet stream

$Cp_{o,1}$ Specific heat capacity of the outlet stream

Energy Balance Equations for a Connector

The energy balance equations for the connector model are as mentioned below.

$$T_{i,1} - T_{o,1} = 0 \tag{3.37}$$

Where,

$T_{i,l}$	Temperature of the inlet stream
$T_{o,l}$	Temperature of the outlet stream

3.4.6 Source Node

In a source node, the inputs to the system are specified by the user, i.e., the input flow rates, specific heat capacity of the streams and the inlet temperature conditions.

3.5 Heat Exchanger Network Model Simulation Algorithm

In engineering practice and research studies, generating equations for large HEN for simulation and optimization by hand results in a tedious task and often results in re-working to correct the equations. The key interest in this research is to overcome such difficulties in designing large network problems, which can further enhance the user to lay emphasis more on the type of optimization problem rather on the modeling issues. Considering these factors in an integrate perspective a software application is designed which supports the following features,

- An auto equation generator (AEG) for HEN design specified by user
- To simulate / solve the HEN generated in the above step with a sparse solver developed in C++
- To optimize the HEN synthesis problems via differential evolution optimization solver (DES) developed in C++.

The auto equation generator (AEG) application has been developed in C# programming environment and relational database management system (RDBMS). The contribution to this project are made by the team [Aman .T, Charumitra .P, Mahalec .V, Masoori, M., Shefali .K, Uma .M. K. A].

My contribution to the development of AEG has been implementation of HEN node models in form of templates, testing of the auto equation generator and implementation of Differential Evolution algorithm for synthesis of heat exchanger networks.

This application is used in the current research to auto generate the equations for HENs. A linear sparse solver (LSS) is used to solve the HEN, LSS was originally developed by Kenneth Kundert [20] in C++ programming environment and an interface for AEG and LSS is developed by the team for the current study. The differential evolution solver (DES) was originally developed by Lester Godwin [24] in C++ programming environment and an interface for AEG and DES is developed by the team.

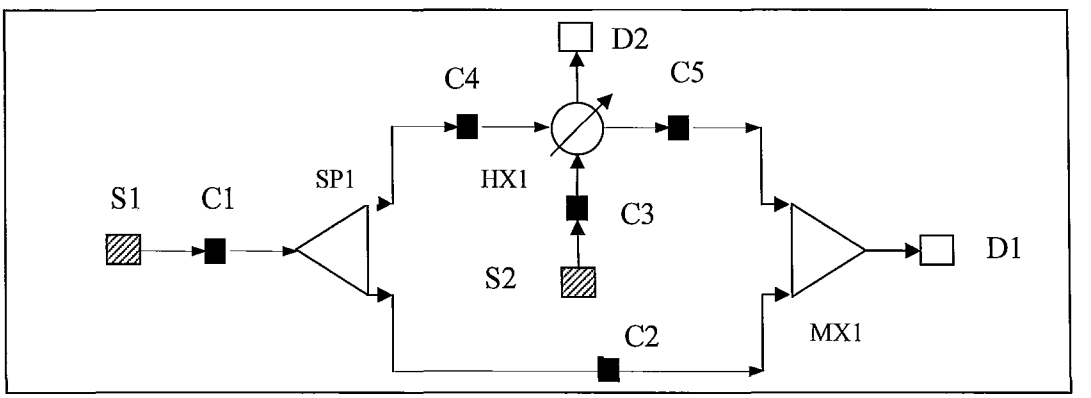
The nodes/equipments described in the previous sections are connected to build a HEN model. The equipments are denoted as nodes **N** which includes heat exchanger $\mathbf{N^{HE}}$, mixer $\mathbf{N^{MX}}$, splitter $\mathbf{N^{SP}}$, source $\mathbf{N^{SO}}$, destination $\mathbf{N^{DS}}$ and connector $\mathbf{N^{CN}}$ nodes. These nodes are interconnected to each other using streams **S** and the connector nodes, which includes hot streams ($\mathbf{S^{H}}$) and cold streams ($\mathbf{S^{C}}$).

The concept of stream is used in this study to represent the flow of stream properties (mass flow, temperature, specific heat capacity, density) between two different nodes (origin node and end node). A stream is always either connected by an origin node to a connector node or connector node to an end node in the former case the stream and the stream properties belong to the origin node and in the later case the stream and the stream properties belong to the end node. Thus, the connector node connects the streams belonging to different nodes and establishes the connectivity equations between the nodes across the entire network. The network equations can be represented in a matrix form of dimension **A (N x S),** where for the incoming stream into a node $\mathbf{A_{ij}} = \mathbf{1}$ and for the outgoing stream from a node $\mathbf{A_{ij}} = \mathbf{-1}$ is the convention followed to establish the connectivity.

A brief summary of the procedure for simulating a HEN problem with AEG is presented here.

The user configures the HEN network (topology and heat exchanger parameters) and specifies the input conditions (mass flow rate, temperature and the heat exchanger parameters) and the AEG application generates the data files that contains the list of variables / parameters, list of the equations, structure of the sparse matrix for each calculational phase and a parameter file for any nonlinear function calculations. These generated files are input to the sparse solver to solve the system and the final results of the HEN simulation are later updated in the database.

The above process from a casual user perspective is illustrated through a sample HEN network problem. In the following discussion, the Figure 3.7 describes a sample network problem. Later, a flow chart of the simulation algorithm and a matrix representation of equations and the corresponding variables phase wise is presented.



▨	- Source	S1, S2
■	- Connector	C1, C2, C3, C4, C5
□	- Destination	D1, D2
◁	- Splitter	SP1
▷	- Mixer	MX1
⊘	- Heat exchanger	HX1

Figure 3.8: Example of a HEN flow-sheet

Convention followed in the matrix representation

- The columns represent the variables & the rows represent the node to which the equation belongs.
- The convention followed for describing a variable is mentioned below

  Example.$X_{y,z}$

  - X- Represent the name of the variable

    m - mass flow rate, T- temperature, Cp –specific heat capacity, Φ- Phi factor,

    C- heat capacity flow rates.
  - Y- Represent the name of the node (user specified name for a node on the flow sheet)
  - Z- Represent the type of port to which it belongs

    I- input, O- output, HI – hot input , HO- hot output, CI – cold input , CO- cold output
- The convention followed for representing a node N^{XY}, N- denotes node, XY- user specified name for the node

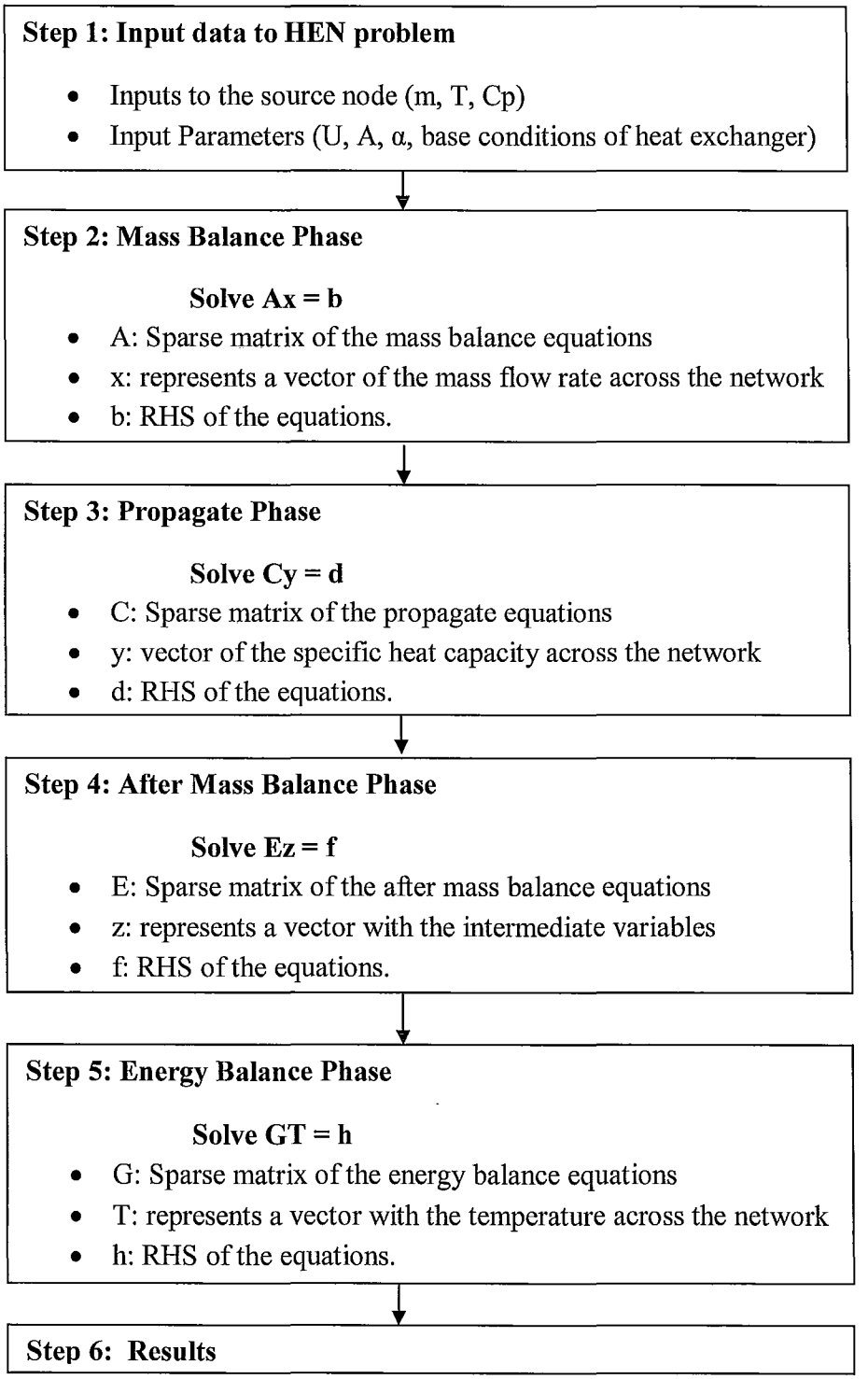


Figure 3.9: HEN simulation algorithm

Mass Balance Phase

Nodes	Mass balance phase (Variables)													RHS
	$m_{S1,O}$	$m_{SP1,I}$	$m_{SP1,O1}$	$m_{HX1,HI}$	$m_{HX1,HO}$	$m_{MX1,I1}$	$m_{MX1,O}$	$m_{SP1,O2}$	$m_{MX1,I2}$	$m_{S2,O}$	$m_{HX1,CI}$	$m_{HX1,CO}$		
S1	1												=	m_{S1}
S2										1			=	m_{S2}
HX1				1	-1								=	0
HX1											1	-1	=	0
C1	1	-1											=	0
C2			1	-1									=	0
C3					1	-1							=	0
C4								1	-1				=	0
C5										1	-1		=	0
SP1		1	-1					-1					=	0
SP1		α_1	-1										=	0
MX1						1	-1		1				=	0

Figure 3.10: Representation of mass balance equations of example HEN problem

Propagate Phase

Nodes	Propagate phase (variables)													RHS
	$Cp_{S1,O}$	$Cp_{SP1,I}$	$Cp_{SP1,O1}$	$Cp_{HX1,HI}$	$Cp_{HX1,HO}$	$Cp_{MX1,I1}$	$Cp_{MX1,O}$	$Cp_{SP1,O2}$	$Cp_{MX1,I2}$	$Cp_{S2,O}$	$Cp_{HX1,CI}$	$Cp_{HX1,CO}$		
N^{S1}	1												=	Cp_{S1}
N^{S2}										1			=	Cp_{S2}
N^{HX1}				1	-1								=	0
N^{HX1}											1	-1	=	0
N^{C1}	1	-1											=	0
N^{C2}			1	-1									=	0
N^{C3}					1	-1							=	0
N^{C4}								1	-1				=	0
N^{C5}										1	-1		=	0
N^{SP1}		1	-1										=	0
N^{SP1}		1						-1					=	0
N^{MX1}						$m_{MX1,I1}$	$-m_{MX1,O}$		$m_{MX1,I2}$				=	0

Figure 3.11: Representation of propagate phase equations of example HEN problem

After Mass Balance Phase

Nodes	After mass balance phase (variables)												RHS
	$C_{HX1,HI}$	$C_{HX1,CI}$	$m_{HX1,HI}$	$m_{HX1,CI}$	Φ_{flow}	$C_{MX1,I1}$	$C_{MX1,I2}$	$C_{MX1,O}$	$m_{MX1,I1}$	$m_{MX1,I2}$	$m_{MX1,O}$		
N^{HX1}	1		$-Cp_{HX1,HI}$									=	0
N^{HX1}		1		$-Cp_{HX1,CI}$								=	0
N^{HX1}	1											=	FUNC
N^{MX1}						1			$-Cp_{MX1,I1}$			=	0
N^{MX1}							1			$-Cp_{MX1,I2}$		=	0
N^{MX1}								1			$-Cp_{MX1,O}$	=	0

Figure 3.12: Representation of after-mass balance equations of example HEN problem

In propagate phase the nonlinear function calculations of Φ_{flow} is evaluated by an external function.

Energy Balance Phase

Nodes	Energy balance phase (Variables)																	RHS
	$T_{S1,O}$	$T_{SP1,I}$	$T_{SP1,O1}$	$T_{HX1,HI}$	$T_{HX1,HO}$	$T_{MX1,I1}$	$T_{MX1,O}$	$T_{SP1,O2}$	$T_{MX1,I2}$	$T_{S2,O}$	$T_{HX1,CI}$	$T_{HX1,CO}$	$Q_{HX1,H}$	$Q_{HX1,C}$	Q_{HX1}	Φ		
N^{S1}	1																=	T_{S1}
N^{S2}										1							=	T_{S2}
N^{HX1}				$C_{HX1,HI}$	$-C_{HX1,HI}$								-1				=	0
N^{HX1}											$-C_{HX1,CI}$	$C_{HX1,CI}$		-1			=	0
N^{HX1}													1		-1		=	0
N^{HX1}														1	-1		=	0
N^{HX1}															1	$-Q_{HX1,B}$	=	0
N^{HX1}				Φ_{flow}							$-\Phi_{flow}$					-1	=	0
N^{C1}	1	-1															=	0
N^{C2}			1	-1													=	0
N^{C3}					1	-1											=	0
N^{C4}								1	-1								=	0
N^{C5}										1	-1						=	0
N^{SP1}		1						-1									=	0
N^{SP1}		1	-1														=	0
N^{MX1}						$C_{MX1,I1}$	$-C_{MX1,O}$		$C_{MX1,I2}$								=	0

Figure 3.13: Representation of energy balance equations of example HEN problem

Chapter 4

HEN Synthesis via DE

In this chapter, an overview of differential evolution optimization is discussed followed by the problem statement and description of the objective function for HEN synthesis problem. Next, the methodology for HEN synthesis and optimization via differential evolution optimization is presented followed by application on four case studies. This chapter concludes with a summary on the methodology and the results. All the case studies in this chapter are implemented using AEG.

4.1 Differential Evolution

Differential evolution belongs to a class of evolutionary optimization methods. It is a non-traditional, population based optimization approach. DE was first proposed by Kenneth Price and Rainer Storn [21]. The major steps involved in differential evolution (DE) optimization are initialization, mutation, recombination and selection.

In initialization step, random population members of the optimization variables or the target vector are created in the search space (within the minimum and maximum bounds of the optimization variables). These, initial population members further undergo mutation, recombination and selection steps to generate new population members. One iteration of this process is termed as generation. The optimization problem is run to a certain maximum number of generations until the desired optimum is reached or until termination criteria is reached.

As the first step of DE, the objective function $f(x)$ to be minimized or maximized is defined, where X is a vector of optimization variables of size m, where $X = \{X_1, X_2, ...X_n\}$. The search space for the optimization variables is defined by specifying the minimum and the maximum bounds ($X_{\min} < X < X_{\max}$).

In the initialization step, NP random vectors of the optimization variables each of dimension m are generated that cover the entire range of the function.

$$X_i^n = X_{\min}^n + rand(0,1) * (X_{\min}^n - X_{\max}^n) \tag{4.1}$$

Where,

$$i = 1.....NP$$
$$n = 1.....\dim$$

The mutation step combined with recombination / crossover and selection step generates new population members from the target population members to drive the search space to the optimum. In the mutation step, a noisy random population is generated from the initial population. Three unique population members X_a, X_b and X_c are randomly selected from the initial population members, the weighted difference of two vectors (X_a , X_b) is added to X_c to generate the mutated population (m_i). In the below equation F is called mutation factor, it is a user supplied constant in the range of 0-1.2. Repeated operation of mutation process would generate new population members that would improve the search space.

$$m_i = X_c + F(X_a - X_b) \tag{4.2}$$

In the crossover step, the target vector undergoes recombination with the mutated vector to generate a new trial vector. Each value of the variable in the trial vector is determined

by the crossover factor a user specified constant parameter that in the range of 0-1 determines whether the value to be copied from the target population or mutated vector.

$$t_i^n = \begin{cases} m_i^n \ldots\ldots\ldots\ldots if(rand(0,1) < CR) \\ \\ X_i^n \ldots\ldots\ldots\ldots elseif(rand(0,1) > CR) \end{cases} \tag{4.3}$$

Where,

$$i = 1 \ldots\ldots\ldots NP$$
$$n = 1 \ldots\ldots\ldots \dim$$

There are several strategies of crossovers available in the literature [21]. Basically, in these crossover strategies they are several ways to select new populations for trial vectors. For example in one-point crossover strategy a single crossover point is selected randomly, all the left entries are copied from target vector and all the right entries from mutated vector. In the current research two such strategies are selected namely uniform (binomial) crossover and exponential crossover.

In exponential crossover randomly a crossover point is selected, all the entries in the trial vector are copied from mutant vector until the random number is less than CR once the random number is greater than CR all the remaining entries are copied form target vector. In uniform crossover a fair chance for survival for both mutant vector and target is given, if the random number is less than CR the value is copied from mutant and if random number is greater than CR then the value is copied from target vector.

In certain cases the population members obtained from the trail vector would have values that result in boundary constraint violations. Such, population members are corrected by setting the values to be within the bounds.

In the selection step, there are two forms of DE one is constrained and the other is unconstrained optimization. In the unconstrained DE operation the fitness of the trial

population is compared with the target vector, the fitter of the two vectors survives in the next generation. But in constrained optimization approach along with the fitness of the objective function the number of constraint violations is also checked. The vector with the fewer constraints violations survives in the next generation.

$$X_i^n = t_i^n \begin{cases} if\,(violations(X_i) > violations(t_i)) \\ if\,(violations(X_i) = violations(t_i)) \\ and \\ f(X_i) > f(t_i) \end{cases} \tag{4.4}$$

The above process is repeated for a certain number of generations or until convergence criteria is met. A flow-chart of the DE is represented in Figure 4.1. This approach of DE combined with the HEN simulation algorithm is extended to HEN synthesis using modular sizes of heat exchangers and is discussed in the following sections.

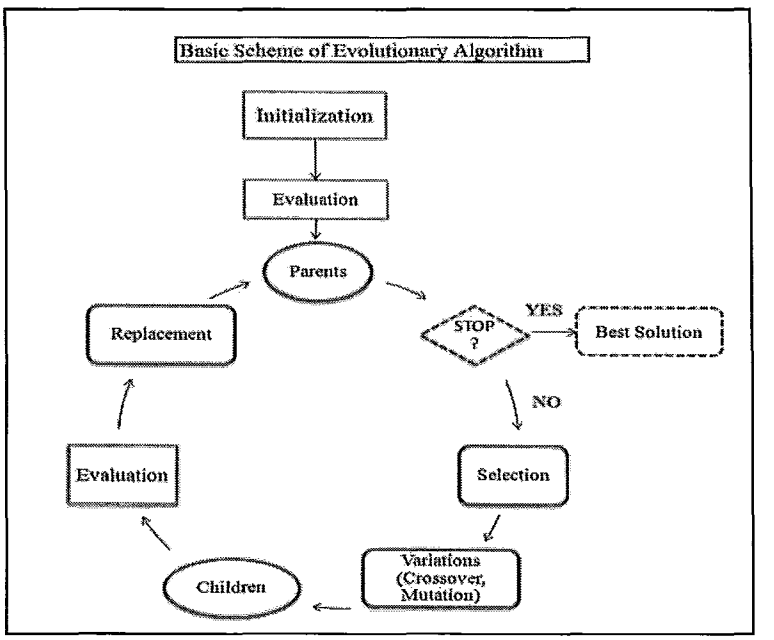


Figure 4.1: Differential evolution algorithm flow-chart

4. 2 Problem Statement

The HEN synthesis problem is to determine a cost optimal cost network of heat exchangers, which provides the desired heat exchange between the hot and cold streams while keeping the global cost GC (investment cost IC (heat exchanger cost) and operating cost OC (utilities cost)) to a minimum. In particular to these types of problems, the following information is specified for optimization of the network.

- A set of hot streams with their corresponding inlet and target or outlet temperatures
- A set of cold streams with their corresponding inlet and target or outlet temperatures
- The specific heat capacities, mass flow rates of the process streams
- Hot and cold utility specifications, their corresponding inlet and outlet temperatures, flow rates, cost of the utilities
- Heat exchangers cost information.

The assumptions made in solving these problems are constant specific heat capacities of the process streams, constant heat transfer coefficients, isothermal mixing of the streams and countercurrent flow heat exchanger configuration.

One of the major issues in HEN design is to handle the combinatorial nature of the problem and to find a feasible solution using simultaneous synthesis methods. Until recently, most of the work was implemented using traditional optimization methods in which the area of the heat exchangers in the final optimal design are continuous (customized). In the current study, this issue is addressed and the proposed HEN simulation methodology combined with differential evolutionary optimization is extended to synthesis of HEN with modular sizes of heat exchangers. The details are discussed in the subsequent sections.

4.2.1 Operating Cost of HEN

Operating cost is the total cost of the utilities used in the network. The operating cost is expressed as a function of the heat loads of the utilities and their cost factors. The equation for the calculation of operating cost is,

$$OC = \sum_{i}^{NHU} C_{HU} . Q_{i,HU} + \sum_{i}^{NCU} C_{CU} . Q_{i,CU} \tag{4.5}$$

Where,

Q_{HU} , Q_{CU} - Heat load of hot utility and cold utility respectively.

C_{HU}, C_{CU} - Individual unit cost of a hot utility and cold utility respectively

NHU , NCU - Number of hot utilities and cold utilities respectively

4.2.2 Investment Cost of HEN

Investment cost is the total expenditure for the installation of the heat exchangers and the utilities in the network. The general equation for the calculation of investment cost of the heat exchangers and the utilities is,

$$IC = \sum_{i}^{NHX} A_0 + A_1 . (area_{HX})^{A_2} + \sum_{i}^{NHU} A_3 + A_4 . (area_{HU})^{A_5} + \sum_{i}^{NCU} A_6 + A_7 . (area_{CU})^{A_8} \tag{4.6}$$

Where

A_0, A_1, A_2 - Cost parameters of heat exchangers.

A_3, A_4, A_5 - Cost parameters of hot utilities.

A_6, A_7, A_8 - Cost parameters of cold utilities.

$area_{HX}$ - Denotes area of heat exchanger.

$area_{HU}$ - Denotes area of hot utility.

$area_{CU}$ - Denotes area of cold utility.

In the current study, the areas of the heat exchangers are optimization variables that are selected from the available modular sizes of heat exchangers. The area of the utilities is determined based on the LMTD calculations.

4.2.3 Global Cost of HEN

Global cost or the total cost of a network is given by the sum of total investment cost and total operating cost.

$$GC = IC + OC \tag{4.7}$$

$$GC = \sum_{i}^{NHU} C_{HU}.Q_{i,HU} + \sum_{i}^{NCU} C_{CU}.Q_{i,CU} + \sum_{i}^{NHU} A_3 + A_4.(area_{HU})^{A_5} + \sum_{i}^{NCU} A_6 + A_7.(area_{CU})^{A_8} + \sum_{i}^{NHX} A_0 + A_1.(area_{HX})^{A_2} \tag{4.8}$$

4.3 HEN Synthesis Methodology via DE

The HEN synthesis algorithm is discussed in 6 sub sections. In the first section, selection of modular sizes of heat exchangers, development of super-structure for a given problem and calculation of base conditions of the heat-exchangers in the super structure is described. In the second section details of implementation of DE to generate initial population members of the heat exchanger networks is presented. In third section, evaluation of the initial population members generated in the above step is discussed. In the fourth section, mutation step and in the fifth section recombination or crossover step of DE and in the final step selection procedure for generation of new population members is discussed.

4.3.1 HEN Superstructure Representation

In HEN synthesis problems the major important criteria to be considered is an appropriate representation of the structure which is generic enough and suitable to all kinds of HEN problem, when only the number of hot and cold process streams and their corresponding initial and target conditions are specified. In general practice, in pinch based analysis [1], the problem is subdivided into small network problems based on heuristics and is solved sequentially to determine the minimum utility cost, minimum number of heat exchangers in the network and minimum investment cost of the network. The major limitation with this approach is that the resulting step depends on the solution of the previous steps. So, to solve these types of optimization problems in a simultaneous fashion, a mixed integer nonlinear programming (MINLP) problem for the synthesis of HEN was proposed by [40]. In this approach a stage-wise super-structure representation is considered for the formulation of HEN model. The design of superstructure for a HEN model is derived as follows,

- For a given set of hot ($\mathbf{N_H}$) and cold ($\mathbf{N_C}$) process streams the number of stages is generally fixed to be maximum{($\mathbf{N_H}$), ($\mathbf{N_C}$)}.

- In each stage, the streams are split such that each hot stream is matched with each cold stream; later the outlet streams of the heat exchangers corresponding to a particular stream are mixed. These mixed streams acts as the input streams in the next stage.
- For these problems, only the initial and final temperature of the process streams are specified, all the intermediate temperatures in between each stage are calculated variables.

A representation of the super-structure for a two hot and two cold streams for a two stage problem is shown in the Figure 4.2. In this structure, there are a maximum of eight heat exchangers, with maximum of four heat exchangers in each stage. The assumptions in this structure formulation are splitting of streams at each stage are allowed, isothermal mixing of streams at the end of each stage, not more than one heat-exchanger is permissible on each split stream and utilities are present at the end of the super structure. For a superstructure design, the base or the operating conditions of each heat exchanger in the network, is determined by assigning modular sizes of heat exchangers available.

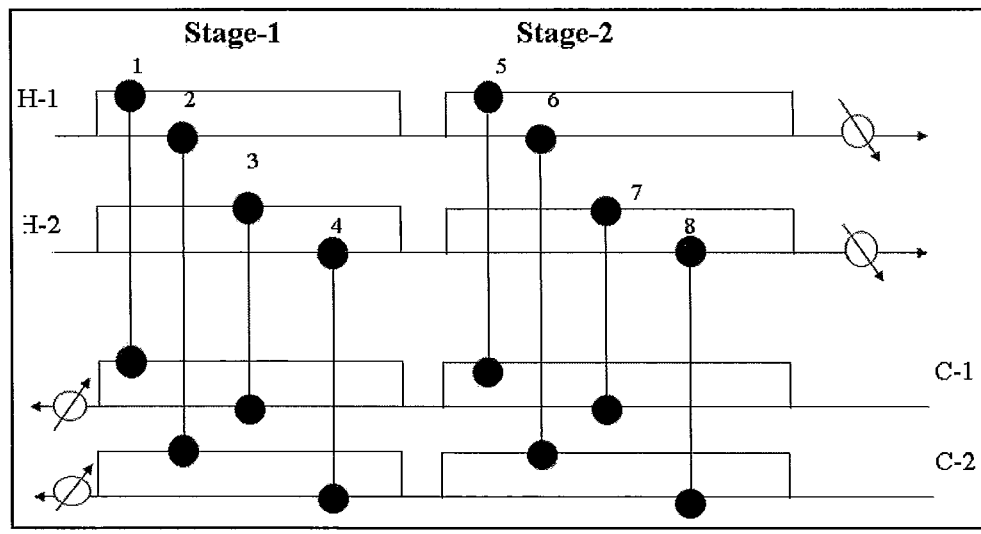


Figure 4.2: Two stage superstructure developed based on Yee and Grossman [40].

4.3.2 Initialization of HEN Population Members

In the initialization, random population members (NP) of HEN's are generated. These population members are termed as target population members. Each target population member p has the following optimization variables associated with it,

1). HEN structure representation $x(S)_T^K$

2). Area of the heat-exchangers in the structure $x(A)_T^K$

3). Split-fractions of the cold streams $x(\alpha_c)_T^K$

4). Split-fraction of the hot streams $x(\alpha_h)_T^K$

HEN Structure Representation

The HEN structure is represented in an array $x(S)_T^K$, which denotes the presence or non-presence of a heat-exchanger in a network. This form of representation of HEN is similar to the approach proposed earlier by Yerramsetty et al. [23]. During the initialization process, random numbers are generated between 0 and 1 to denote the heat-exchanger presence on the hot streams. The length of the array gives the maximum number of heat-exchangers (MHE) that can placed on the hot streams assuming that maximum of one heat-exchanger can be placed on each hot stream split. MHE is equal to the product of maximum number of hot splits (MHS), maximum number of cold splits (MCS), and number of stages (ST).

$$MHE = MHS \text{ x } MCS \text{ x } ST \tag{4.9}$$

For example, in a 2 cold stream and 2 hot stream network problem, the maximum number of cold splits (MCS) is 2, the maximum number of hot splits (MHS) is 2 and number of stages (ST) is 2, then the maximum number of heat-exchangers (MHE) in the network or the length of the array of HEN network is equal to 8 (2x2x2).

In the design, splitting of streams is allowed in each stage but isothermal mixing of the streams at the end of each stage is assumed. Thus, the super-structure network does not allow more than one heat-exchanger on a particular split in each stage. In each individual population member, the value of each element in the array $x(S)_T^K$, represents the presence of heat-exchanger in the network. If value is between 0 and 0.5 it denotes absence of heat exchanger and if it is greater than 0.5 and less than 1 it denotes the presence of heat exchanger.

$$x(S)_T^K = rand\{0,1\} \tag{4.10}$$

Where,

K = 1.... MHE

T - Denotes target population.

An example of an array $x(S)_T^K$ for 2 cold streams, 2 hot streams and 2 stage network is presented below.

Example:

$x(S)_T^K = \{0.134, 0.689, 0.768, 0.266, 0.894, 0.768, 0.943, 0.342\}$

An example entry of an array $x(S)_T^K$ generated by DE is shown above. The superstructure for the example problem is given by Figure 4.3. The network structure representation and the hot and cold stream splits corresponding to each heat exchanger in the network are presented in Figure 4.4. The final structure for the example problem is given by Figure 4.5.

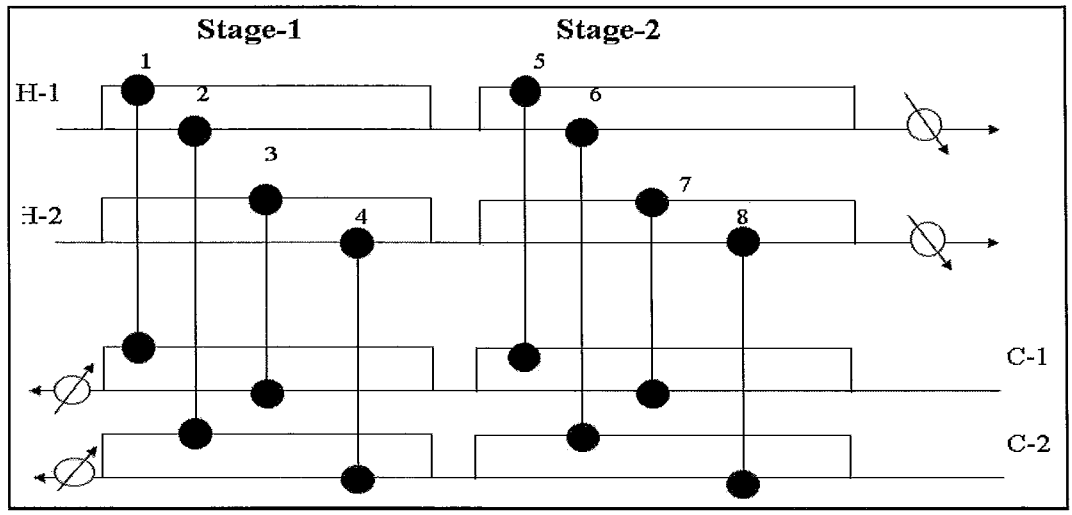


Figure 4.3: Two stage superstructure with maximum matches on the hot & cold streams.

	Stage-1				Stage-2			
HX	1	2	3	4	5	6	7	8
$x(S)_T$	0.134	0.689	0.768	0.266	0.894	0.768	0.943	0.342
Round($x(S)_T$)	0	1	1	0	1	1	1	0
$H_{i,j,k}$	$H_{1,1,1}$	$H_{1,1,2}$	$H_{1,2,1}$	$H_{1,2,2}$	$H_{2,1,1}$	$H_{2,1,2}$	$H_{2,2,1}$	$H_{2,2,2}$
$C_{i,j,k}$	$C_{1,1,1}$	$C_{1,2,1}$	$C_{1,1,2}$	$C_{1,2,2}$	$C_{2,1,1}$	$C_{2,2,1}$	$C_{2,1,2}$	$C_{2,2,2}$

Figure 4. 4: Sample output of DE operation to generate $x(S)_T^K$

Where,

HX - denotes the heat exchanger number

$x(S)_T^K$ - HEN structure array generated by DE.

$H_{i,j,k}$ / $C_{i,j,k}$ - Represents the corresponding hot / cold stream to the heat exchanger in $x(S)_T^K$.

i - Denotes stage number

j - Denotes the hot / cold stream number

k - Denotes the corresponding hot / cold stream split.

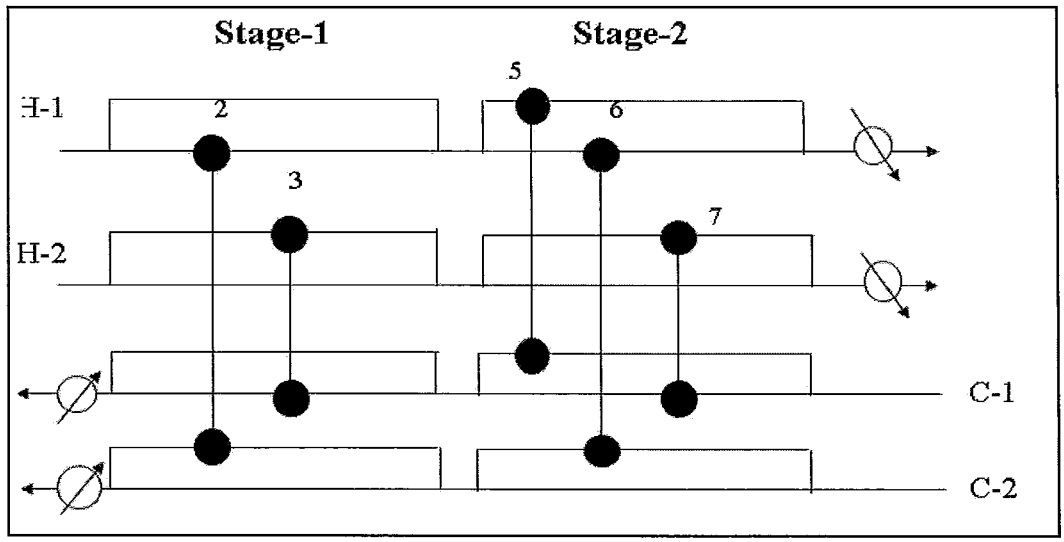


Figure 4.5: Resulting network structure after applying DE

Area of Heat Exchangers

The array $x(S)_T^K$ (network) generated above gives the network structure and the placement of the heat exchangers between the hot and cold process streams. For all the non-zero entries in $x(S)_T^K$, which represents the presence of a heat exchanger, area of the heat exchanger is assigned randomly from the available pool of modular sizes of heat exchangers. If $x(S)_T^K = 0$ it indicates non-presence of heat exchanger, so the area of the heat exchanger is zero. In this step, each modular heat exchanger available is assigned a size, based on the random number generator the corresponding area of the heat exchanger is assigned.

$$x(A)_T^K = rand\{0, MSE\} \tag{4.11}$$

Where,

$x(A)_T^K$ - Each nonzero entry in the array represents the size of a heat exchanger

MSE - Denotes the maximum number of modular heat exchangers available.

Example:

In a 2 cold stream, 2 hot stream and 2 stages network problem. The maximum number of heat exchangers (MHE) in the network is 8. Suppose, the modular heat exchangers sizes available are {10, 50, 100, 200, 500, 1000}. This implies that the maximum number of sizes (MSE) of heat exchangers available is 6. The size of each heat exchanger is represented with a number between 0 and 6, which are generated by a random number generator during DE initialization step. For evaluation purpose of HEN, the elements in the array $x(A)_T^K$ are mapped with the corresponding area of the heat exchanger from the available pool of heat exchangers.

For example say, the below $x(A)_T^K$ array is obtained after applying DE,

$x(A)_T^K$	1.134	2.689	5.768	4.266	1.894	0.768	2.943	3.342

Table 4.1: An example entry of array $x(A)_T^K$ generated from DE operation.

In this step $x(A)_T^K$ array is corrected with respect to the network structure $x(S)_T^K$ obtained in the earlier step. For all the zero elements in integer value of ($x(S)_T^K$) the corresponding entries in $x(A)_T^K$ is set to zero.

Round(x(S)T)	0	1	1	0	1	1	1	0
$x(A)_T^K$	0	2.689	5.768	0	1.894	0.768	2.943	0

Table 4.2: Corrected entries of array $x(A)_T^K$ generated from DE operation

Later, for evaluation of HEN, $x(A)_T^K$ is encoded with the pool of heat exchangers available. The following vector $x(A_E)_T^K$ is obtained.

$x(A_E)_T^K$	0	100	1000	0	50	10	100	0

Table 4.3: Corrected entries $x(A)_T^K$ mapped to generate $x(A_E)_T^K$ for evaluation

Split Fractions of Hot and Cold Streams

In HEN synthesis process, the other set of optimizing variables that are optimized to obtain a cost optimal HEN network are the split fractions of hot and cold streams. These optimizing variables determine the heat capacity flow rates of the streams across each heat exchanger in the network. The split fractions of the hot and cold streams are represented by $x(\alpha_h)_T^K$ and $x(\alpha_c)_T^K$ respectively. In the initialization step, the split fractions of the hot and cold streams are assigned by a random number generator by ensuring that a zero split fraction is assigned for cases where no heat exchanger is present. These split fractions are later corrected by normalization procedure to ensure that the sum of split fractions of a particular stream add up to 1.

$$x(\alpha_c)_T^K = rand\{0,1\} \tag{4.12}$$

$$x(\alpha_h)_T^K = rand\{0,1\} \tag{4.13}$$

Where,

$x(\alpha_c)_T^K$ - Split fractions of the cold streams

$x(\alpha_h)_T^K$ - Split fractions of the hot streams

Example:

For example, in a 2 cold stream, 2 hot stream and 2 stages network problem. The maximum number of cold splits per cold stream is 2 and the maximum number of hot splits per stream is 2. In the initialization step say $x(\alpha_c)_T^K$ and $x(\alpha_h)_T^K$ obtained after applying DE

$x(\alpha_h)_T^K$	0.245	0.345	0.342	0.693	0.752	0.743	0.576	0.981
$x(\alpha_c)_T^K$	0.657	0.482	0.581	0.422	0.589	0.231	0.859	0.132

Table 4.4: Example entry of hot and cold split fractions from DE operation

For all the zero elements in ($x(S)_T^K$) the corresponding entries in $x(\alpha_c)_T^K$ and $x(\alpha_h)_T^K$ are set to zero and the entries are corrected such that the sum of the splits add up to 1. The following corrected vector is obtained.

Round(x(S)T)	0	1	1	0	1	1	1	0
$x(\alpha_h)_T^K$	0	1	1	0	0.503	0.497	1	0
$x(\alpha_c)_T^K$	0	1	1	0	0.406	1	0.594	0

Table 4.5: Corrected entries of hot and cold split fractions from DE operation.

The final corrected entries of HEN structure $x(S)_T^K$, area of the heat exchangers $x(A)_T^K$, split fractions of hot $x(\alpha_h)_T^K$ and cold streams $x(\alpha_c)_T^K$ of a single population member is presented below. The same process is repeated to initialize the rest of the NP – 1 population members.

Round(x(S)T)	0	1	1	0	1	1	1	0
$x(A)_T^K$	0	100	1000	0	50	10	100	0
$x(\alpha_h)_T^K$	0	1	1	0	0.503	0.497	1	0
$x(\alpha_c)_T^K$	0	1	1	0	0.406	1	0.594	0

Table 4.6: Final corrected population member from DE operation

Evaluation of Initial Population Members

In this step, the initial populations generated above are evaluated through the simulation algorithm discussed in the earlier chapter to determine the global cost of the network. To calculate the global cost of the network, the HEN simulation is solved in 6 phases, the equation types representing are

- **Mass Balance**: The first phase is to solve linear mass balance equations to determine the mass flow rate across the HEN.
- **Propagate Phase**: The second phase involves the calculation of physical properties (specific heat capacity, density) across the network.
- **After Mass Balance** – This phase involves the calculation of intermediate and nonlinear function variables (mc_p, Φ_{flow}).
- **Energy Balance** - The fourth phase involves solving the linear energy balance equations to determine the temperatures, heat duties of the exchanger, the hot and cold utilities across the entire network, in this step the set of equations representing the energy model are linear with respect to temperatures as the flow rates are determined in the first step.
- **Utility Balance**- This phase involves the calculation of the area of the hot and cold utilities
- **Objective function** – The final phase involves the objective function calculation, in this phase capital cost of individual units i.e., heat exchangers, hot utilities and cold utilities along with operating cost of the hot and cold utilities is calculated to determine the global cost of the network.

In this step, the initial or target populations are evaluated to calculate the global cost of the network.

4.3.3 Mutation of HEN Population Members

In the mutation step, the initial or target population generated in the previous step undergo mutation operation to generate a noisy random vector termed as mutated population members. To generate these population members, three random individuals a, b and c are selected from initial or target population members for each corresponding variable. From these random vectors, the scaled difference of a,b is added to a third vector c to obtain the mutated population member. Thus, this process is repeated for NP population members. Generation of mutated population members for HEN structures

$x(S)_T^K$, area of the heat-exchangers $x(A)_T^K$, split-fractions of the hot and cold streams $x(\alpha_h)_T^K$, $x(\alpha_c)_T^K$ is discussed in the following sections.

Mutation of HEN Members

The three random individuals a, b and c vectors representing HEN structures $x(S)_T^K$, area of the heat-exchangers $x(A)_T^K$, split-fractions of the hot and cold streams $x(\alpha_h)_T^K$, $x(\alpha_c)_T^K$ individually are combined by applying the mutation rule to generate new populations of HEN structures $m(S)_T^K$, area of the heat-exchangers $m(A)_T^K$, split-fractions of the hot and cold streams $m(\alpha_h)_T^K$, $m(\alpha_c)_T^K$.

$$m(X)_T = x(X)_c + F(x(X)_a - x(X)_b) \tag{4.14}$$

Where $X = S, A, \alpha_c, \alpha_h$, the above equations are valid for all cases of X.

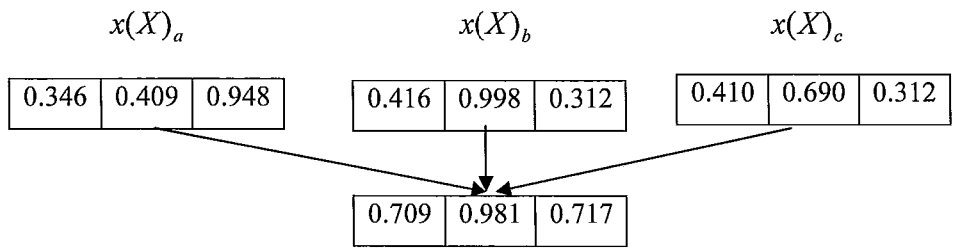


Figure 4. 6: Representation of mutation operation for a single variable

In cases, when the resulting noisy vector crosses the minimum and maximum bound of the corresponding variable, then the variable is reset by a random number generator within the minimum and maximum bounds.

4.3.4 Recombination of HEN Population Members

In this step, the mutated population members and the target population members are mated to generate a new population called trial population. The recombination is done on each variable based on crossover probability CR, a user specified constant parameter that decides whether the value to be copied from either mutant or target population member into trial population. The details of this operation is discussed for HEN structures $x(S)_T^K$, area of the heat-exchangers $x(A)_T^K$, split-fractions of the hot and cold streams $x(\alpha_h)_T^K$, $x(\alpha_c)_T^K$

In the process of recombination of HEN, a random number is generated between 0 and 1 and is compared with CR specified by the user. If the random number is less than CR, the value from $m(S)_T^K$ of the mutant population is copied into the trial vector $t(S)_T^K$ else the value from $x(S)_T^K$ of the target population is copied. This operation is repeated for each variable in a population member.

$$t(S)_T^K = \begin{cases} m(S)_T^K \ldots\ldots\ldots\ldots if\,(rand(0,1) < CR) \\ \\ x(S)_T^K \ldots\ldots\ldots\ldots elseif\,(rand(0,1) > CR) \end{cases} \tag{4.15}$$

Where,

K = 1.....MHE

T = 1.....NP

The same process is repeated to generate trial vector of area of the heat-exchangers $t(A)_T^K$ and split-fractions of the hot and cold streams $t(\alpha_h)_T^K$, $t(\alpha_c)_T^K$ these trial population members are compared with the corresponding trial vector $t(S)_T^K$ for each individual population member to correct any inconsistencies in the area of the heat-exchangers $t(A)_T^K$, and split-fractions of the hot and cold streams $t(\alpha_h)_T^K$, $t(\alpha_c)_T^K$. These corrected

population members are evaluated individually using the HEN simulation algorithm to determine the global cost of each network.

4.3.5 Selection of HEN Population Members

In the selection step of DE, the global cost of the network of each individual in trial population is compared with the corresponding population member in the target population. Based on the principle of least cost vector survival, the network with the minimum cost survives and participates in the next generation.

If GC ($t(S)_T^K$, $t(A)_T^K$, $t(\alpha_h)_T^K$, $t(\alpha_c)_T^K$) < GC ($x(S)_T^K$, $x(A)_T^K$, $x(\alpha_h)_T^K$, $x(\alpha_c)_T^K$)

{

$$x(S)_T^K = t(S)_T^K$$

$$x(A)_T^K = t(A)_T^K$$

$$x(\alpha_h)_T^K = t(\alpha_h)_T^K$$

$$x(\alpha_c)_T^K = t(\alpha_c)_T^K$$

}

else

{

$$x(S)_T^K = x(S)_T^K$$

$$x(A)_T^K = x(A)_T^K$$

$$x(\alpha_h)_T^K = x(\alpha_h)_T^K$$

$$x(\alpha_c)_T^K = x(\alpha_c)_T^K$$

}

Where,

GC	- Global cost of the network
t	- Trial population members

x	- Target population members
K	= 1.....MHE
T	= 1.....NP

The above steps discussed in 4.3, 4.4, 4.5 form one generation of DE. This procedure of mutation, crossover and selection of new population members are repeated until maximum of generations are reached or a convergence criterion is met. A flow chart for HEN synthesis methodology is presented in the Figure 4.7. This approach is successfully applied on four case studies the details of results are explained in the following sections.

Flow Chart of HEN Synthesis Algorithm

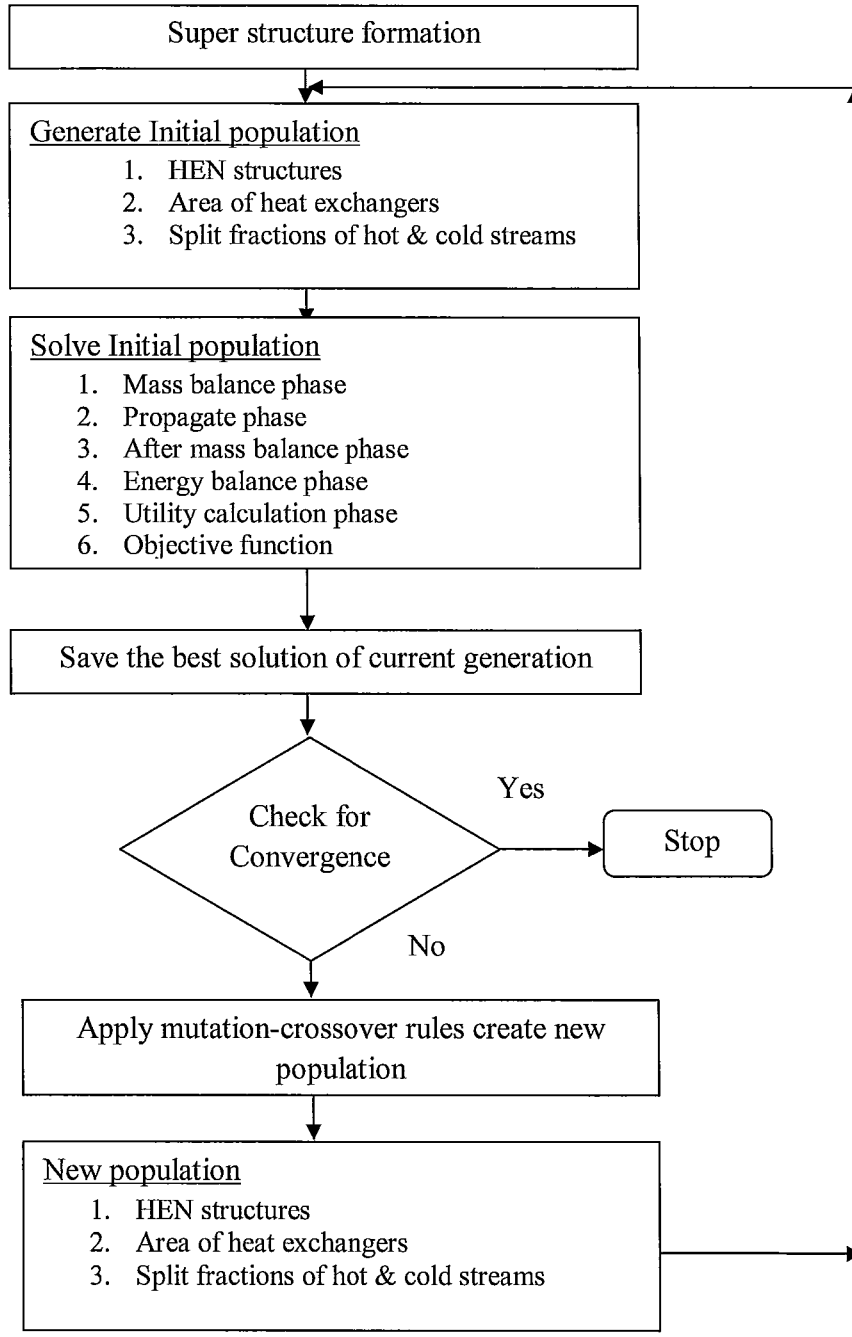


Figure 4.7: Flow-chart of the HEN synthesis methodology

4.4 Application of HEN Synthesis Methodology

The proposed HEN synthesis methodology is applied on four problems in the literature. The results of the current approach are compared with the results earlier published in terms of the network structure, global cost of the networks, area of the heat exchangers and computational time. All the case studies presented in this chapter are implemented using AEG on a computer with the following configuration (AMD Athlon 64 CPU (2.4 Ghz), with 3 GB RAM and XP- Microsoft).

4.4.1 Case Study 1

In the first case study, there are 4 process streams, which consist of 2 hot streams and 2 cold streams. The input conditions and the necessary data for the optimization are mentioned in the Table 4.7. This problem was recently solved by Ravaganani et al [30].

Stream	T_{in} (°C)	T_{out} (°C)	F (kWC^{-1})	h ($kW^{-1}m^2$ K)	Cost (US \$ kW^{-1} $year^{-1}$)
H1	175	45	10	2.615	
H2	125	65	40	1.333	
C1	20	155	20	0.917	
C2	40	112	15	0.166	
Steam	180	179		5.000	110
Water	15	25			10

Table 4.7: Problem data for case study 1

Note: Cost of heat exchangers = 1200[Area (m^2)] $^{0.57}$ (US\$ $year^{-1}$)

Optimization variables	16
Mutation factor (F)	0.85
Crossover factor (CR)	0.8
Population size	20
Crossover strategy	Uniform, Exponential

Table 4.8: DE Optimization parameters for case study 1

Modular sizes (m^2) of heat exchangers considered for the current study are {15, 50, 100, 120, 150, 200, 270, 280, 290, 300}. The superstructure for this problem is designed based on the following specifications; number of stages is 1, maximum of 2 hot splits, maximum of 2 cold splits and maximum of 4 heat exchangers in the network. This case study is implemented for two scenarios in one scenario uniform crossover strategy is applied and in second scenario exponential crossover strategy is applied.

The final optimal network solution obtained with the current approach resulted with an annual cost of US$ 115738.09 which is lower than all the earlier solutions. When uniform crossover strategy is applied it took almost 200 generations to reach the optimum solution but in the case when exponential crossover strategy is applied it took 100 generations to reach the optimum. The final optimal network solution obtained with both the approaches is same. Finally, exponential crossover strategy proved to be a better crossover strategy over uniform crossover for this case study.

In the current solution, even the small utilities of sizes less than 1 (m^2) are accounted but in the solutions obtained by Ravagnani et al. [30] they are neglected. The computational time required for the network synthesis is approximately 12 minutes. The number of equations in each phase is presented in Table 4.9.

Equation Phase	Number of equations
Mass balance	60
Propagate phase	124
After mass balance	52
Energy balance	88
Utility calculation phase	4
Objective function	18

Table 4.9: Number of equations in each phase for case study 1

The final design obtained with the current approach is shown in Figure 4.8. The results of the current approach are compared with that of the Ravagnani et al. [30] results in Table 4.10. The convergence rate profiles are shown in Figure 4.9 and Figure 4.10.

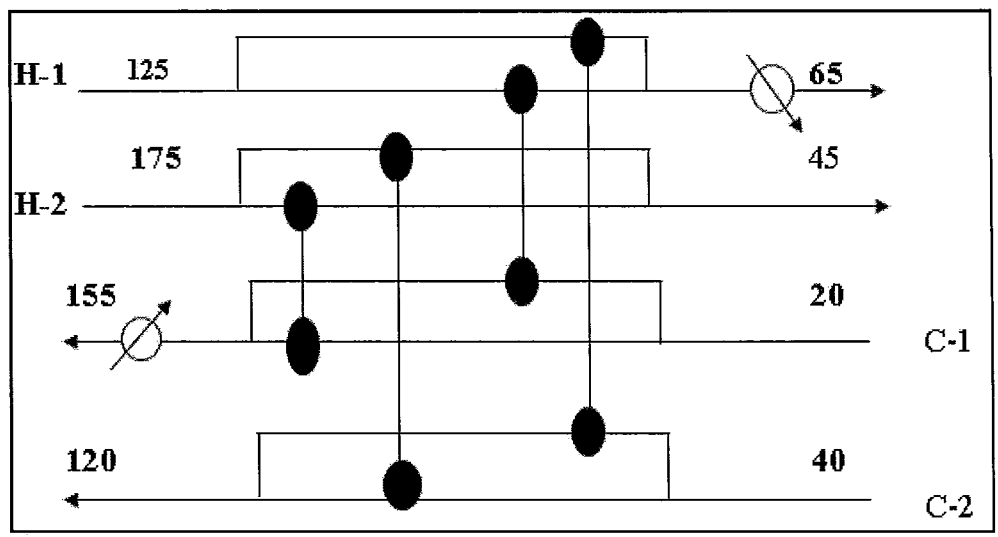


Figure 4.8: The final cost optimal network for case study 1 in this work

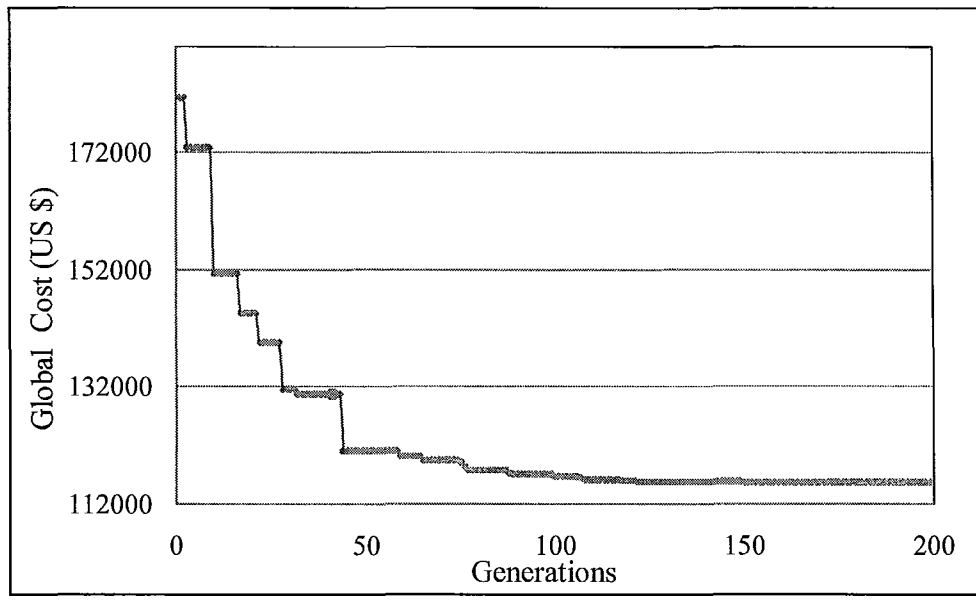


Figure 4.9: Convergence rate profile for case study 1 (Uniform crossover strategy)

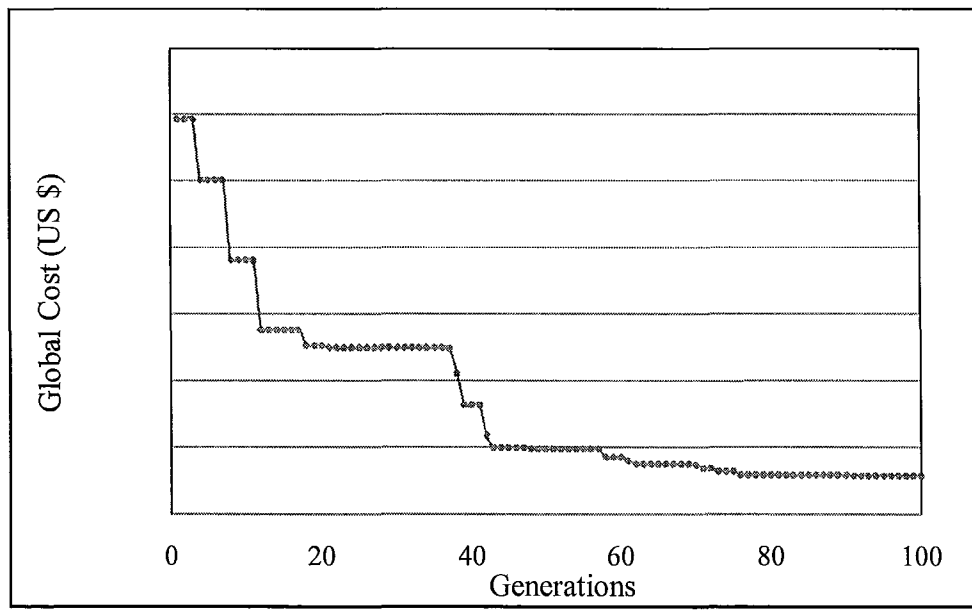


Figure 4.10: Convergence rate profile for case study 1 (Exponential crossover strategy)

This work Total Capital Cost (US $)		
Area of the HE (m^2)		**IC (US $)**
HE-1	150	20871.48
HE-2	150	20871.48
HE-3	15	5617.63
HE-4	300	30984.21
Area of HU (m^2)		**IC (US $)**
HU-1	10.86	4673.38
HU-2	0.03	163.05
Area of CU (m^2)		**IC (US $)**
CU-1	0.01	100.91
CU-2	3.28	2360.84
Total investment cost (US $)		85642.98
Total Energy Cost ($)		
Energy CU (kW)		**OC (US $)**
CU-1	257.06	28277.05
CU-2	0.33	35.96
Energy HU (kW)		**OC (US $)**
HU-1	0.41	4.09
HU-2	177.8	1778
Total energy cost (US $)		30095.11
Global Cost (US $)		115738.09

Ravaganani [30] Total Capital Cost (US $)		
Area of the HE (m^2)		**IC (US $)**
HE-1	52.78	11507.71
HE-2	202.23	24746.47
HE-3	147.81	20697.24
HE-4	291.74	30495.02
Area of HU (m^2)		**IC (US $)**
HU-1	8.85	4158.52
HU-2	0	0
Area of CU (m^2)		**IC (US $)**
CU-1	0	0
CU-2	3.04	2261.62
Total investment cost (US $)		93866.58
Total Energy Cost (US $)		
Energy CU (kW)		**OC (US $)**
CU-1	200	22000
CU-2	0	0
Energy HU (kW)		**OC (US $)**
HU-1	120.32	1203.2
HU-2	0	0
Total energy cost (US $)		23203.2
Global Cost (US $)		117069.78

Table 4.10: Comparison of the results in current work & Ravagnani's [30] work

Note: HE - represents heat exchanger, HU- Hot utility, CU- Cold utility, IC- Investment cost, OC- Operating cost

4.4.2 Case Study 2

In the second case study, there are 5 process streams, which consist of 3 hot streams and 2 cold streams. The input conditions and the necessary data for the optimization are mentioned in the Table 4.11. This problem was originally solved by Zamora and Grossmann [19] and recently solved by Petterson [15].

Stream	T_{in} (°C)	T_{out} (°C)	F (kWC^{-1})	h ($kW^{-1}m^2$ K)	Cost (US \$ kW^{-1} yr^{-1})
H1	159	77	2.285	0.10	
H2	267	80	0.204	0.04	
H3	343	90	0.538	0.50	
C1	26	127	0.933	0.01	
C2	118	265	1.961	0.50	
Steam	300	299			110
Water	20	60			10

Table 4.11: Problem data for case study 2

Note: Cost of heat exchangers = 7400 + 80[Area (m^2)] (US\$ $year^{-1}$)

Optimization variables	72
Mutation factor (F)	0.85
Crossover factor (CR)	0.8
Population size	20
Crossover strategy	Exponential

Table 4.12: DE Optimization parameters for case study 2

Modular sizes (m^2) of heat exchangers considered for the current study are {2, 3, 5, 10, 40, 50, 60, 80, 85, 90}. The superstructure for this problem is designed based on the following specifications, 3 stages, maximum of 2 hot splits, maximum of 3 cold splits and maximum of 18 heat exchangers in the network.

The final optimal network solution obtained with the current approach resulted with an annual cost of US$ 80,097.04 which is lower than all the earlier solutions. The final design obtained with the current approach is same as the one proposed by Petterson [15] but the key difference is, in the current solution all the heat exchangers are of modular sizes whereas the area of the heat exchangers in Petterson's [15] network are of continuous sizes. The computational time required for the network synthesis is approximately 37.7 minutes. The number of equations in each phase is presented in Table 4.13.

Equation Phase	**Number of equations**
Mass balance	199
Propagate phase	403
After mass balance	202
Energy balance	286
Utility calculation phase	5
Objective function	34

Table 4.13: Number of equations in each phase for case study 2

This problem is also solved by Zamora and Grossman[19], Lin and Miller [2] the final design in these works is same. The key point to be noted in these approaches is, splitting of the streams were not allowed but in the current approach splitting was allowed and the current methodology performed better in comparison to the previous solutions. The final design obtained with the current approach is shown in Figure 4.11. The results of the current approach are compared with that of the Petterson's [15] results in Table 4.14.

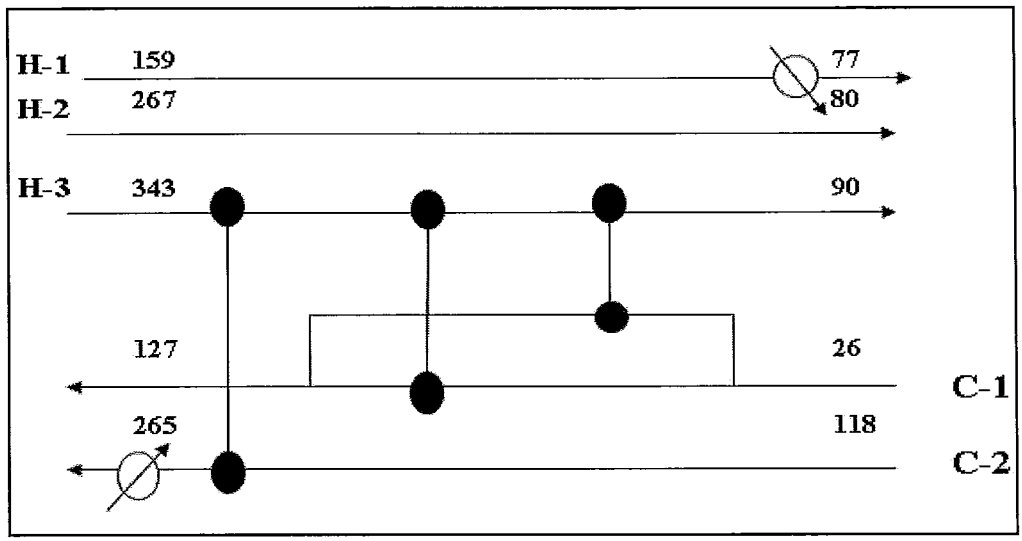


Figure 4.11: The final cost optimal network for case study 2 in this work

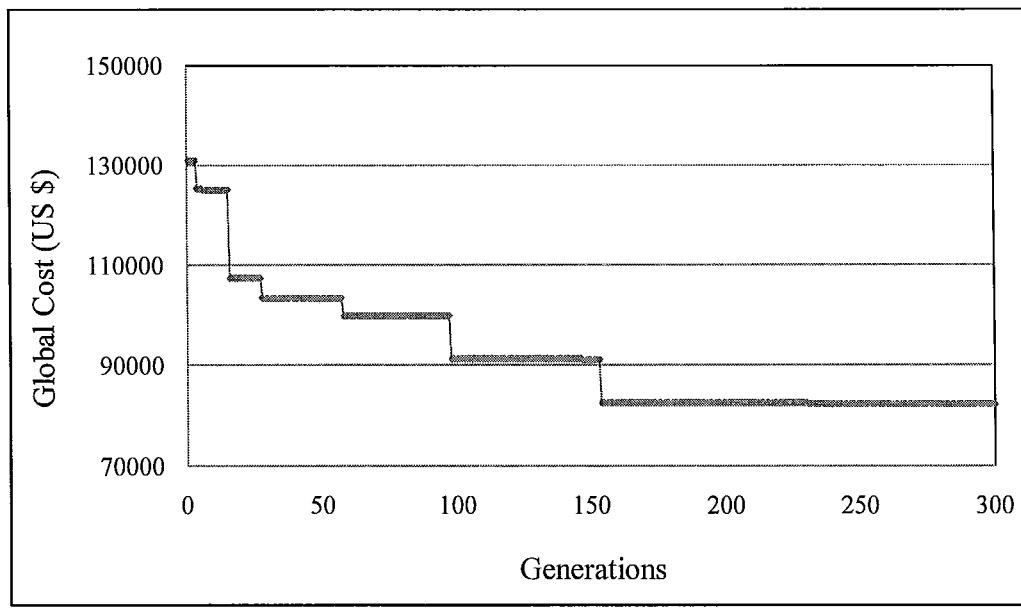


Figure 4.12: Convergence rate profile (Global cost Vs. Generations) for case study 2

This work Total Capital Cost (US $)		
Area of the HE (m^2)		**IC (US $)**
HE-6	3	7640
HE-8	55	11800
HE-9	82	13960
Area of HU (m^2)		**IC (US $)**
HU-1	0	0
HU-2	59.61	12169.11
Area of CU (m^2)		**IC (US $)**
CU-1	36.95	10356.05
CU-2	0	0
CU-3	0	0
Total investment cost (US $)		55925.17
Total Energy Cost (US $)		
Energy CU (kW)		**OC (US $)**
CU-1	187.81	1878.1
CU-2	0	0
CU-2	0	0
Energy HU (kW)		**OC (US $)**
HU-1	202.71	22298.18
HU-2	0	0
Total energy cost (US $)		24171.88
Global Cost (US $)		80,097.04

Petterson [15] Total Capital Cost (US$)		
Area of the HE (m^2)		**IC (US $)**
HE-6	2.62	7609.22
HE-8	50.75	11460.13
HE-9	84.27	14181.83
Area of HU (m^2)		**IC (US $)**
HU-1	0	0
HU-2	60.43	12234.52
Area of CU (m^2)		**IC (US $)**
CU-1	36.94	10355.46
CU-2	0	0
CU-3	0	0
Total investment cost (US $)		55841.16
Total Energy Cost (US $)		
Energy CU (kW)		**OC (US $)**
CU-1	187.37	1873.7
CU-2	0	0
CU-2	0	0
Energy HU (kW)		**OC (US $)**
HU-1	208.24	22906.40
HU-2	0	0
Total energy cost (US $)		24780.1
Global Cost (US $)		80,621.27

Table 4.14: Comparison of the results with the current approach & Petteron's [15]

Note: HE - represents heat exchanger, HU- Hot utility, CU- Cold utility, IC- Investment cost, OC- Operating cost

4.3 Case Study 3

In the third case study, there are 8 process streams, which consist of 7 hot streams and 1 cold stream [32]. The input conditions and the necessary data for the optimization are mentioned in the Table 4.15.

Stream	T_{in} (°F)	T_{out} (°F)	M x 10^3 (lbs)	C_p (BTU / h °F)	Cost (US \$ $kBTU^{-1}$ yr^{-1})
H1	376	197	211	0.616908	
H2	481	257	195	0.631868	
H3	508	266	97.3	0.637035	
H4	560	399	140.5	0.676378	
H5	639	490	358.0	0.764876	
H6	678	330	296.9	0.729763	
H7	680	584	196.5	0.732516	
C1	90	484	862.5	0.592725	
Steam	540	539			12.76
Water	100	180			5.24

Table 4.15: Problem data for case study 3

Note : Cost of heat exchangers = 35[Area (ft^2)] $^{0.6}$ (US\$ $year^{-1}$), U = 150 BTU / ft^2 °F for all the matches, except for matches involving steam U = 200 BTU / ft^2 °F

The superstructure for this problem is designed based on the following specifications, maximum of 1 hot split, maximum of 7 cold splits. This problem is studied for two scenarios, in one scenario the number of stages is fixed to 3 and in another scenario the number of stages is fixed to 4. Modular sizes (ft^2) of heat exchangers considered for the current study are {200, 500, 800, 1500, 1800, 2000, 2370, 2900, 3000, 3200, 4500, 6000, 9600}.

Scenario 1 (with 3 stages)

The final optimal network obtained with the current approach resulted with an annual cost of US$ 91653.23, which is lower than the cost of the network in [32]. The capital cost with the current approach is less than the cost of the network in [32]. However, in the current study utility cost has increased in comparison to [32]. The observation made in this case study is that for almost the same global cost there are multiple optimal solutions for the network. The final design obtained in the current study is shown in Figure 4.13. The results of the current approach are compared with that of the Goyal's [32] in Table 4.17. Area of the heat exchangers and utilities obtained in this scenario are mentioned in Table 4.19. The computational time required for the network synthesis is approximately 136.4 minutes. The number of equations in each phase is presented in Table 4.18.

Optimization variables	84
Mutation factor (F)	0.85
Crossover factor (CR)	0.8
Population size	30
Crossover strategy	Uniform

Table 4.16: DE Optimization parameters for case study 3 (Scenario 1)

	This work (US $)	**Goyal (US $)**
Heat exchanger capital cost	48518.07	92150.3
Hot utilities capital cost	35.60	0
Hot utilities energy cost	2149.33	0
Cold utilities capital cost	1421.18	0
Cold utilities energy cost	39529.04	0
Global cost	91653.22	92150.3

Table 4.17: Comparison of the results with the current approach (Scenario 1) & Goyal's [32]

Equation Phase	Number of equations
Mass balance	256
Propagate phase	520
After mass balance	253
Energy balance	364
Utility calculation phase	8
Objective function	53

Table 4.18: Number of equations in each phase for case study 3 (Scenario 1)

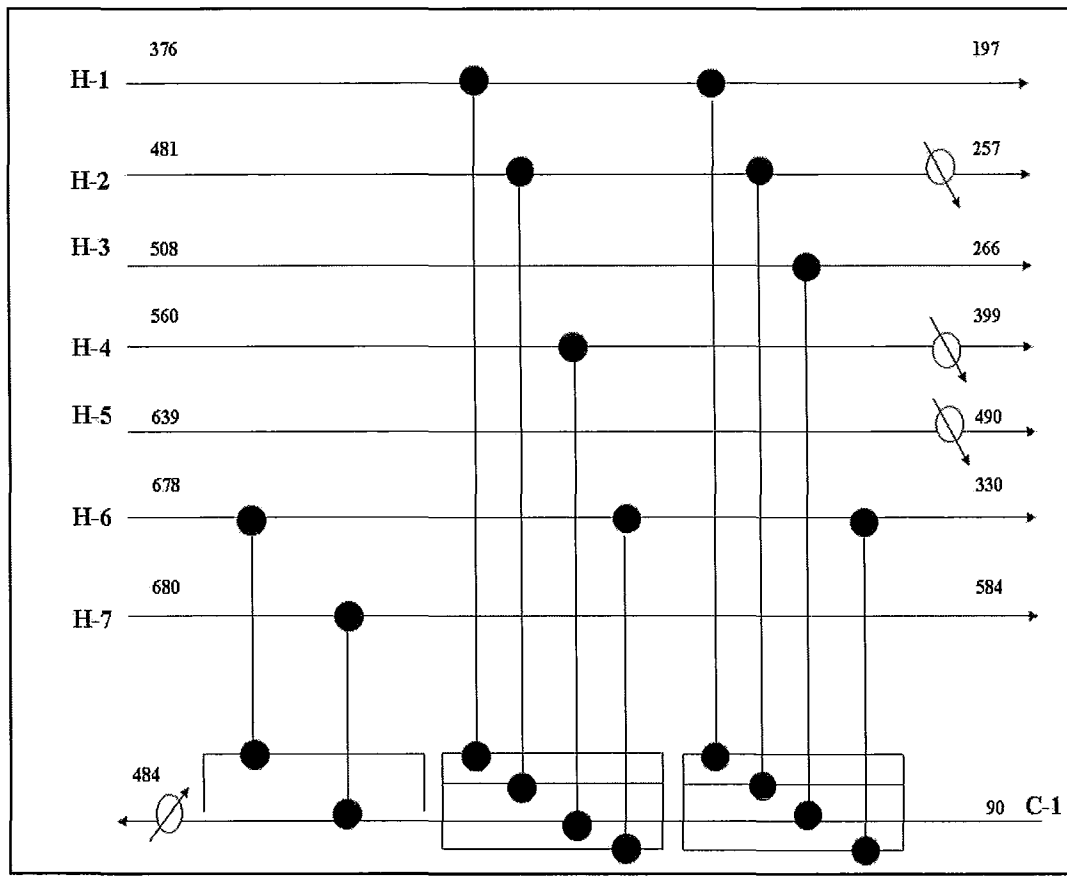


Figure 4.13: The final cost optimal network for case study 3 (Scenario 1) in this work

Area of the heat exchangers (ft^2) in scenario-1 (with 3-stages)						Area of the utilities (ft^2)	
Stage-1		**Stage-2**		**Stage-3**		CU-01	0
HX-01	0	HX-08	3200	HX-15	2900	CU-02	77.15
HX-02	0	HX-09	2350	HX-16	1800	CU-03	0
HX-03	0	HX-10	0	HX-17	2000	CU-04	140.34
HX-04	0	HX-11	3200	HX-18	0	CU-05	29.48
HX-05	0	HX-12	0	HX-19	0	CU-06	0
HX-06	9600	HX-13	9600	HX-20	3200	CU-07	0
HX-07	2350	HX-14	0	HX-21	0	HU-01	10.82

Table 4.19: Area of the heat exchangers and the utilities for Case study 3 (Scenario 1)

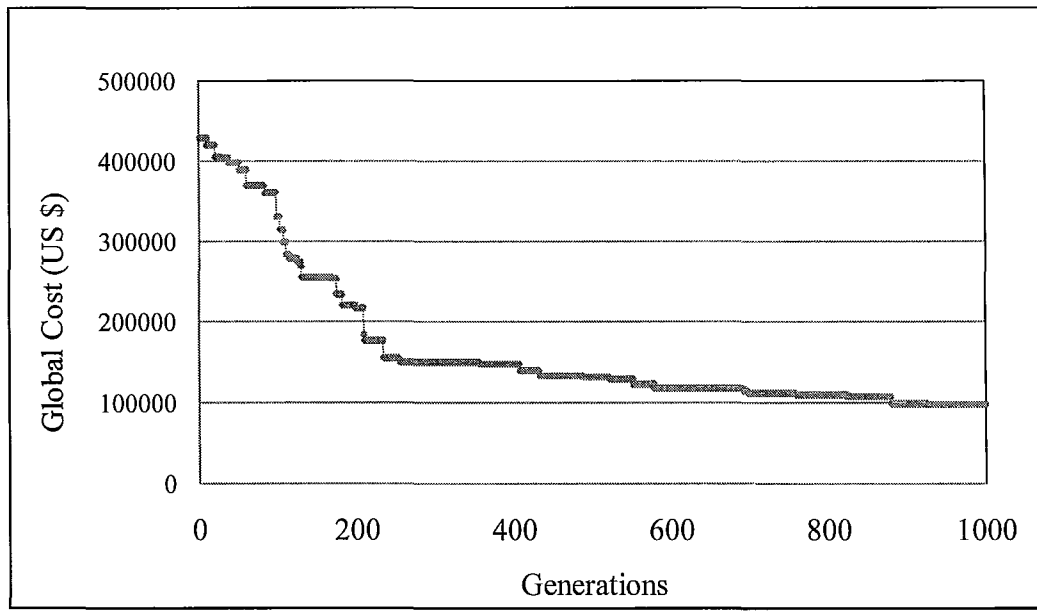


Figure 4.14: Convergence rate profile (Global cost Vs. Generations) for case study 3 (Scenario 1)

Scenario 2 (with 4 stages)

The final optimal network in this scenario resulted with an annual cost of US$ 160031.6973, which is higher than the solution obtained in the scenario 1. In comparison with scenario 1 we may say that the optimal number of stages is 3 for this problem. The final design obtained in the current study is shown in Figure 4.14. The results of the current approach are compared with that of the Goyal's [32] in Table 4.21. The computational time required for the network synthesis is approximately 171.4 minutes. The number of equations in each phase is presented in Table 4.22. Area of the heat exchangers and area of the utilities obtained in this scenario are mentioned in Table 4.23.

Optimization variables	112
Mutation factor (F)	0.85
Crossover factor (CR)	0.8
Population size	30
Crossover strategy	Uniform

Table 4.20: DE Optimization parameters for case study 3 (Scenario 2)

	This work (US $)	**Goyal (US $)**
Heat exchanger capital cost	65466.81	92150.3
Hot utilities capital cost	35.60	0
Hot utilities energy cost	562.11	0
Cold utilities capital cost	3268.80	0
Cold utilities energy cost	90698.36	0
Global cost	160031.69	92150.3

Table 4.21: Comparison of the results with the current approach (Scenario 2) & Goyal's network [32]

Equation Phase	Number of equations
Mass balance	328
Propagate phase	664
After mass balance	332
Energy balance	464
Utility calculation phase	8
Objective function	50

Table 4.22: Number of equations in each phase for case study 3 (Scenario 2)

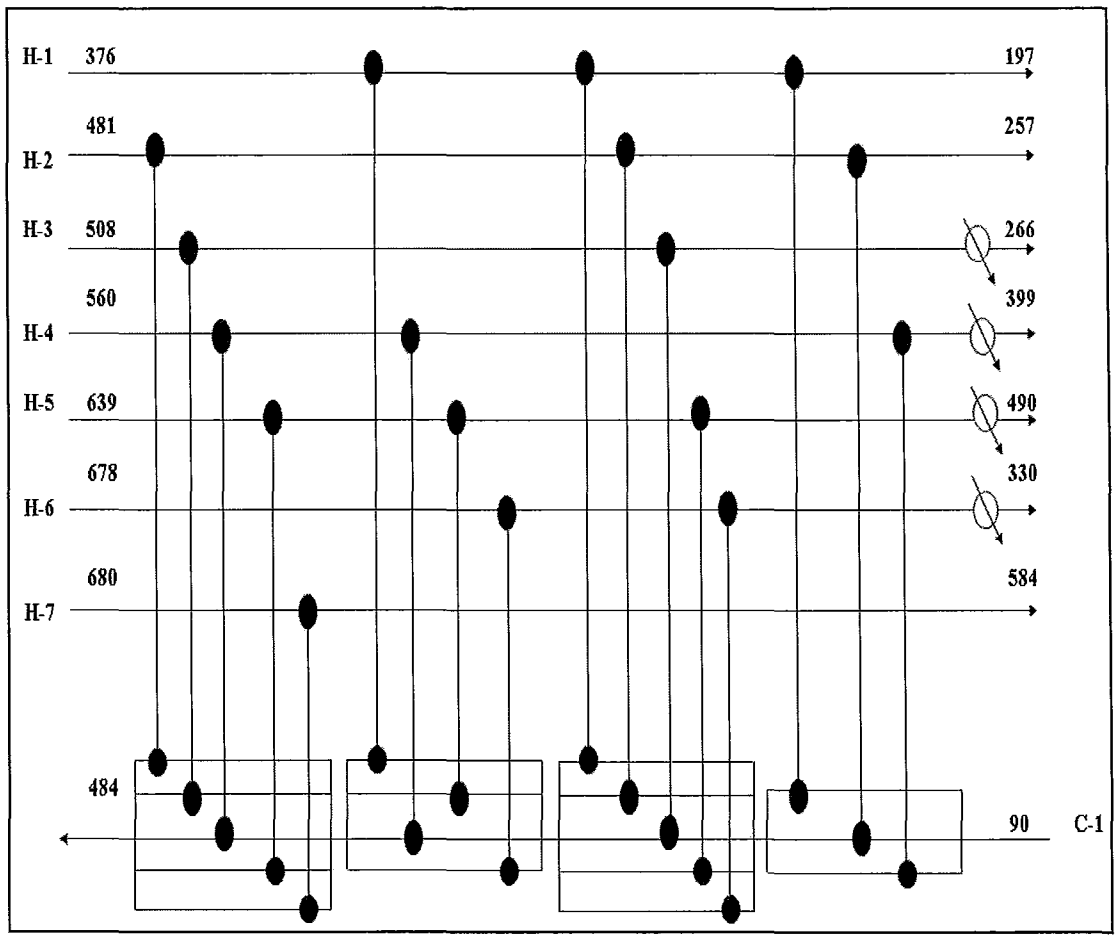


Figure 4.15: Case study-3 final design with 4- stages

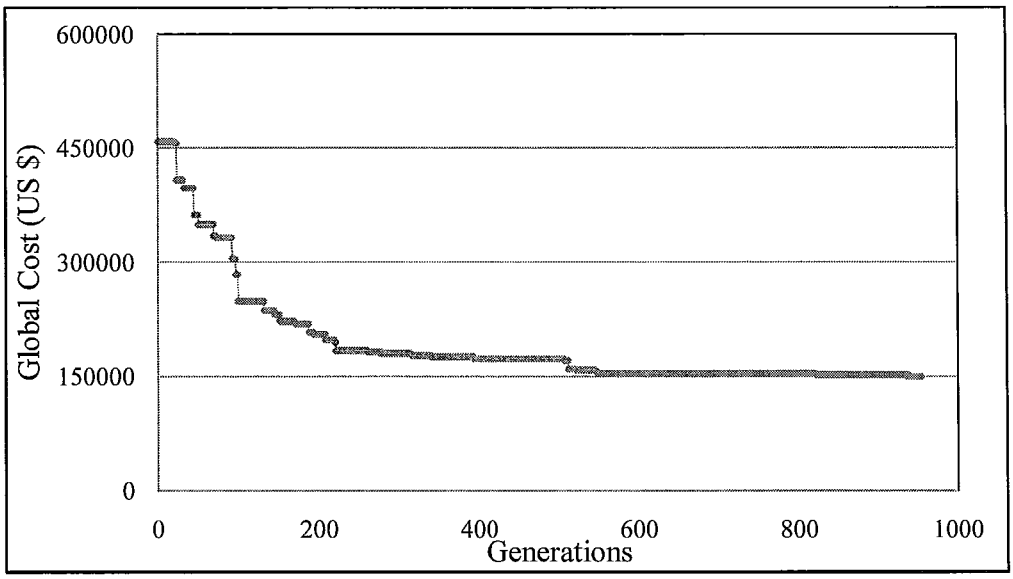


Figure 4.16: Convergence rate profile (Global cost Vs. Generations) for case study 3 (Scenario 2)

Area of the heat exchangers (ft^2) in scenario-1 (with 3-stages)								**Area of the utilities (ft^2)**	
Stage-1		**Stage-2**		**Stage-3**		**Stage-4**		CU-01	0
HX-01	0	HX-08	2370	HX-15	4500	HX-22	800	CU-02	0
HX-02	1800	HX-09	0	HX-16	800	HX-23	1800	CU-03	164.7
HX-03	3200	HX-10	0	HX-17	3200	HX-24	0	CU-04	212.5
HX-04	9600	HX-11	1500	HX-18	0	HX-25	500	CU-05	90.83
HX-05	6000	HX-12	500	HX-19	2000	HX-26	0	CU-06	73.58
HX-06	0	HX-13	2000	HX-20	6000	HX-27	0	CU-07	0
HX-07	2000	HX-14	0	HX-21	0	HX-28	0	HU-01	0

Table 4.23: Area of the heat exchangers and the utilities for case study 3 (Scenario 2)

4.4.4 Case Study 4

In the fourth case study, there are 37 process streams, which consist of 21 hot streams and 16 cold streams [17]. The input conditions for hot streams and cold streams are mentioned in Table 4.26 and Table 4.27 respectively.

The superstructure for this problem is designed based on the following specifications, maximum of 16 hot splits per one hot stream, maximum of 21 cold splits per one cold stream and the number of stages is fixed to 1.

Optimization variables	1344
Mutation factor (F)	0.85
Crossover factor (CR)	0.95
Population size	10
Crossover strategy	Uniform

Table 4.24: DE Optimization parameters for case study 4

Equation Phase	**Number of equations**
Mass balance	5894
Propagate phase	8878
After mass balance	3172
Energy balance	4402
Utility calculation phase	37
Objective function	416

Table 4.25: Number of equations in each phase for case study 4

Stream	T_{in} (°C)	T_{out} (°C)	F (kWC^{-1})	Cost (US \$ kW^{-1} $year^{-1}$)
H1	175	150	196.92	
H2	168	55	6.787611	
H3	190	175	1053.067	
H4	160	135	256.08	
H5	102	40	651.8871	
H6	170	40	6.115385	
H7	98.4	40	37.94521	
H8	97.4	40	39.89547	
H9	98.4	40	33.9726	
H10	40	25	43.46667	
H11	25	10	46.13333	
H12	104.7	40	8.85626	
H13	80	40	21	
H14	40	25	22.6	
H15	25	-7.4	47.9321	
H16	57.5	40	20.45714	
H17	0	-11.7	155.4701	
H18	-96.2	-99	188.9286	
H19	-28.6	-30.8	3040.909	
H20	44.7	40	1129.149	
H21	976	300	37.30769	
Steam	540	539		110

Table 4.26: Hot streams data for case study 4.

Stream	T_{in} (°C)	T_{out} (°C)	F (kWC^{-1})	Cost (US \$ kW^{-1} $year^{-1}$)
C2	10	160	32.82	
C3	170	170.5	31592	
C4	120	120.5	12804	
C5	170	170.5	27334	
C6	170	222	30.8462	
C7	68	90	35.3636	
C8	-15	-14.5	2896	
C9	122	122.5	772	
C10	2.4	2.6	8125	
C11	68	68.5	6482	
C12	-5	-4.5	6524	
C13	52	53	4666	
C14	175	196	300	
C15	40	99	110	
C16	-33.9	79.9	16.4763	
Water	100	180		10

Table 4.27: Cold streams data for case study 4

Note: Cost of heat exchangers = $1200[\text{Area } (m^2)]^{0.57}$ (US\$ $year^{-1}$), the heat transfer coefficients are all simplified as 0.8 W m^{-2} K^{-1}

Modular sizes (m^2) of heat exchangers considered in scenario 1 are {10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 4500}. The final optimal network in this scenario resulted with an annual cost of US\$ 2869431.79, which is higher than the cost reported in [17]. The results of the current approach are compared with that of the [17] in Table 4.31. In the current approach the capital cost is more as compared to the [17] results but the operating cost is less with the current approach as compared to [17]. Since, the operating cost is less in the current approach on long run this model would be advantageous. However some of the

limitations indentified in the current solution are, they are more than 30 heat exchangers of smaller sizes less than 20 m^2. The reason for such large number of small size heat exchangers is because in the current solution the numbers of utilities are decreased in compensation with small size heat exchangers.

The computational time required for the network synthesis is approximately 32 hours. The reason for such huge amount of computational time is because of the size of the problem. In [17] splitting of streams was not allowed, but in the current approach splitting is allowed, so the problem size in [17] is small as compared to the current approach. The number of equations and the computational time to solve each phase for a single population member is presented in Table 4.28. In Table 4.28 we could clearly see that the nonlinear function calculation phases are taking huge time compared to linear balance equations. So, the possible improvement for the current simulation algorithm is to minimize the computational time for nonlinear function calculations. Area of the heat exchangers and area of the utilities obtained in this scenario are mentioned in Table 4.29 and Table 4.30 respectively. The results obtained in the current approach are compared with that of [17] in Table 4.31

Equation Phase	**Number of equations**	**Computational time (sec) approximately**
Mass balance	5894	0.5
Propagate phase	8878	0.5
After mass balance	3172	2
Energy balance	4402	0.5
Utility calculation phase	37	1
Objective function	416	1

Table 4.28: Number of equations and computational time to solve each phase for case study 4 (Scenario 1)

Area of the heat exchangers (m^2) on each hot stream for case study 4 (scenario 1)											
H-1		**H-2**		**H-3**		**H-4**		**H-5**		**H-6**	
HX-01	0	HX-17	0	HX-33	0	HX-49	0	HX-65	20	HX-81	0
HX-02	0	HX-18	0	HX-34	4500	HX-50	0	HX-66	0	HX-82	0
HX-03	100	HX-19	0	HX-35	0	HX-51	0	HX-67	100	HX-83	40
HX-04	50	HX-20	20	HX-36	300	HX-52	0	HX-68	0	HX-84	0
HX-05	0	HX-21	10	HX-37	10	HX-53	40	HX-69	0	HX-85	0
HX-06	0	HX-22	0	HX-38	0	HX-54	50	HX-70	700	HX-86	10
HX-07	0	HX-23	0	HX-39	0	HX-55	0	HX-71	0	HX-87	20
HX-08	90	HX-24	0	HX-40	0	HX-56	0	HX-72	50	HX-88	80
HX-09	20	HX-25	0	HX-41	20	HX-57	100	HX-73	0	HX-89	30
HX-10	0	HX-26	0	HX-42	0	HX-58	200	HX-74	50	HX-90	0
HX-11	60	HX-27	40	HX-43	0	HX-59	0	HX-75	80	HX-91	0
HX-12	0	HX-28	10	HX-44	0	HX-60	60	HX-76	0	HX-92	0
HX-13	600	HX-29	50	HX-45	0	HX-61	20	HX-77	0	HX-93	0
HX-14	300	HX-30	30	HX-46	0	HX-62	60	HX-78	0	HX-94	0
HX-15	10	HX-31	0	HX-47	0	HX-63	0	HX-79	0	HX-95	0
HX-16	40	HX-32	0	HX-48	0	HX-64	30	HX-80	0	HX-96	100

H-7		**H-8**		**H-9**		**H-10**		**H-11**	
HX-97	70	HX-113	70	HX-129	0	HX-145	30	HX-161	20
HX-98	0	HX-114	0	HX-130	0	HX-146	60	HX-162	0
HX-99	30	HX-115	50	HX-131	80	HX-147	0	HX-163	0
HX-100	40	HX-116	0	HX-132	0	HX-148	40	HX-164	10
HX-101	0	HX-117	0	HX-133	0	HX-149	0	HX-165	100
HX-102	20	HX-118	200	HX-134	60	HX-150	0	HX-166	0
HX-103	0	HX-119	0	HX-135	0	HX-151	40	HX-167	0
HX-104	30	HX-120	0	HX-136	0	HX-152	0	HX-168	20
HX-105	0	HX-121	0	HX-137	0	HX-153	0	HX-169	20
HX-106	50	HX-122	20	HX-138	0	HX-154	40	HX-170	0
HX-107	0	HX-123	0	HX-139	500	HX-155	300	HX-171	0
HX-108	80	HX-124	0	HX-140	300	HX-156	0	HX-172	0
HX-109	0	HX-125	0	HX-141	50	HX-157	0	HX-173	0
HX-110	0	HX-126	500	HX-142	0	HX-158	0	HX-174	0
HX-111	0	HX-127	0	HX-143	0	HX-159	20	HX-175	0
HX-112	0	HX-128	0	HX-144	0	HX-160	0	HX-176	20

H-12		H-13		H-14		H-15		H-16	
HX-177	0	HX-193	0	HX-209	0	HX-225	0	HX-241	0
HX-178	90	HX-194	0	HX-210	0	HX-226	70	HX-242	0
HX-179	0	HX-195	20	HX-211	0	HX-227	0	HX-243	0
HX-180	0	HX-196	300	HX-212	0	HX-228	0	HX-244	0
HX-181	30	HX-197	50	HX-213	0	HX-229	0	HX-245	30
HX-182	40	HX-198	80	HX-214	0	HX-230	70	HX-246	0
HX-183	0	HX-199	30	HX-215	80	HX-231	0	HX-247	0
HX-184	0	HX-200	10	HX-216	0	HX-232	0	HX-248	0
HX-185	20	HX-201	0	HX-217	10	HX-233	40	HX-249	70
HX-186	10	HX-202	0	HX-218	0	HX-234	10	HX-250	0
HX-187	0	HX-203	0	HX-219	40	HX-235	40	HX-251	0
HX-188	30	HX-204	0	HX-220	40	HX-236	0	HX-252	20
HX-189	30	HX-205	30	HX-221	0	HX-237	40	HX-253	50
HX-190	0	HX-206	0	HX-222	20	HX-238	30	HX-254	0
HX-191	0	HX-207	0	HX-223	20	HX-239	0	HX-255	0
HX-192	80	HX-208	0	HX-224	0	HX-240	0	HX-256	0

H-17		H-18		H-19		H-20		H-21	
HX-257	0	HX-273	0	HX-289	20	HX-305	10	HX-321	900
HX-258	0	HX-274	0	HX-290	0	HX-306	10	HX-322	100
HX-259	0	HX-275	0	HX-291	0	HX-307	0	HX-323	20
HX-260	0	HX-276	0	HX-292	0	HX-308	0	HX-324	100
HX-261	0	HX-277	0	HX-293	50	HX-309	0	HX-325	20
HX-262	0	HX-278	20	HX-294	0	HX-310	100	HX-326	0
HX-263	0	HX-279	100	HX-295	0	HX-311	0	HX-327	0
HX-264	0	HX-280	0	HX-296	40	HX-312	0	HX-328	0
HX-265	0	HX-281	100	HX-297	0	HX-313	30	HX-329	40
HX-266	0	HX-282	0	HX-298	0	HX-314	0	HX-330	40
HX-267	0	HX-283	500	HX-299	0	HX-315	20	HX-331	0
HX-268	20	HX-284	0	HX-300	0	HX-316	0	HX-332	0
HX-269	0	HX-285	0	HX-301	0	HX-317	0	HX-333	0
HX-270	10	HX-286	0	HX-302	0	HX-318	80	HX-334	0
HX-271	0	HX-287	0	HX-303	0	HX-319	60	HX-335	0
HX-272	0	HX-288	50	HX-304	0	HX-320	0	HX-336	400

Table 4.29: Area of the heat exchangers for case study 4 (Scenario 1)

Area of the hot / cold utilities (m^2) for scenario 1 (Case study 4)					
CU-H1	44.85	CU-H17	141.25	HU-C1	0
CU-H2	0	CU-H18	154.16	HU-C2	0
CU-H3	26.77	CU-H19	306.95	HU-C3	0
CU-H4	0	CU-H20	126.36	HU-C4	17.61
CU-H5	746.6	CU-H21	36.98	HU-C5	0
CU-H6	0			HU-C6	0
CU-H7	21.82			HU-C7	0
CU-H8	0			HU-C8	0
CU-H9	0			HU-C9	0
CU-H10	0			HU-C10	0
CU-H11	95.80			HU-C11	0
CU-H12	0			HU-C12	0
CU-H13	0			HU-C13	39.13
CU-H14	15.00			HU-C14	0
CU-H15	137.5			HU-C15	0
CU-H16	0			HU-C16	0

Table 4.30: Area of the utilities for case study 4 (Scenario 1)

	This work (US $)	GA & SA [17] (US $)	GA [17] (US $)
HX CAPITAL COST	1272751.213	636505.9298	NA
HU CAPITAL COST	12541.31821	40809.29541	NA
CU CAPITAL COST	154082.8794	519790.1985	NA
HU ENERGY COST	1093411.48	1052615	NA
CU ENERGY COST	336644.9049	517636.2	NA
GLOBAL COST ($)	2869431.79	2767356.62	3188169.4

Table 4.31: Comparison of the results with the current approach for case study 4 (Scenario 1) and [17]

4.5 Chapter summary

In this chapter, we described the details of HEN synthesis methodology with differential evolution optimization for designing HENs using standard, modular sizes of the heat exchangers. We implemented the proposed methodology on a small scale problem to a large scale problem and tested the performance of the method varying the number of stages, the number of modular sizes of heat exchangers and different crossover strategies.

Chapter 5

Steady State Data Reconciliation

This chapter deals with the topic of data reconciliation of HEN. First a short overview of variable classification for partially measured systems, data reconciliation formulation using QR factorization and parameter observability analysis are discussed. Next, framework for data reconciliation and parameter estimation of HEN combined with the simulation algorithm and QR factorization is presented with application on two case studies. The chapter concludes with a summary and discussion. All case studies in this chapter are implemented in Matlab, and solved by fmincon.

5.1 Introduction

In process industries, accurate process measurements are very important for proper understanding of the process behavior. However, these measurements may contain random errors or systematic errors due to the improper functioning or failure of the process sensors. Due to which these measurements do not obey the mass and energy balances of the process. So, data reconciliation methodology is applied to adjust these measurements, such that they obey the mass and energy balances of the process model. In the data reconciliation problem, the objective is to minimize the error between the measurements and the corresponding variables values in the process model subject to the constraints (mass and energy balance equations) of the system.

For partially measured systems, the data reconciliation problem is decomposed into two sub problems and the constraints of the problem are rewritten in terms of the measured and unmeasured variables. The unmeasured variables are eliminated from the constraints by using projection matrix [10]. There are several methods available for the construction of the projection matrix [10]. However the most efficient method is QR factorization [31,

34], which was first applied to data reconciliation by Swartz C.L.E [9] to decompose and solve data reconciliation problems.

In this chapter, a formulation for data reconciliation combined with QR factorization and HEN simulation algorithm is presented. This method is successfully implemented on a process system. The HEN simulation algorithm combined with parameter observabaility analysis is extended for data reconciliation and estimation of heat exchanger heat transfer coefficients.

5.2 Variable Classification

The major step involved in data reconciliation for partially measured systems is variable classification. Classification of variables, provide valuable information about the results obtained from data reconciliation. It gives information whether a solution is unique; in case of a measurement failure it provides information whether all the unmeasured variables are still observable.

The variables of a process can be divided into measured, unmeasured or fixed variables. Measured variables can be either redundant or non-redundant and unmeasured variables can be either observable or unobservable. Figure 5.1 gives the representation of variable classification. Some of the definitions of the variables are given below, for detailed summary on this topic the reader is encouraged to refer [11]

Definition of redundant measurement: A measured variable is redundant if it is observable even if that variable is not measured [11].

Definition of observable variable: An unmeasured variable in a steady state process is observable, if it can be determined uniquely from the available measurements which are consistent with all the given constraints. If the unmeasured variable is not so determinable then the variable is unobservable [11].

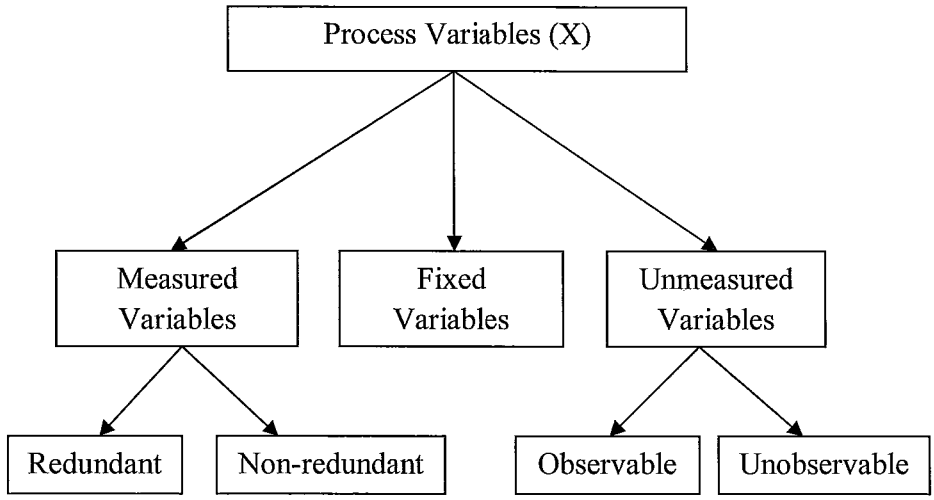


Figure 5.1: Scheme for variable classification, adapted from [12]

The above classification of variables is necessary for defining a data reconciliation problem and for selection of measurements for data reconciliation and parameter estimation problems. In real time optimization (RTO), an appropriate selection of measurements is required to make the parameters of the model observable. Improper selection of measurements may result in some parameters to be unobservable and in the end; solution of the reconciliation problem will result in a non-unique solution.

5.3 Data Reconciliation Formulation for Linear System

The equation representing the measurements of model is described as

$$Y = X + \varepsilon \tag{5.1}$$

Where,

Y Vector of n measurements

X Vector of true values corresponding to the measured variables

ε Vector of unknown random errors

In this problem, a set of estimates $\hat{X}$ are to be determined for X such that they satisfy the given constraints for a problem. The data reconciliation formulation for a linear system with all the variables measured is mentioned below,

$$\begin{aligned} \underset{X}{Min} \quad & (Y-X)\Sigma^{-1}(Y-X) \\ S.T \quad & AX = 0 \end{aligned} \tag{5.2}$$

Where,

Y: Vector of n measurements

X: Vector of true values corresponding to the measured variables

Σ: Covariance matrix.

AX: Set of linear equations of the problem

A: A matrix of dimension m x n (Incidence matrix of the constraints)

The objective of this minimization problem is to adjust the variables X, for the given process model $AX = 0$, such that the deviation from the measured variables Y is minimized.

5.3.1 Linear Systems with Measured and Unmeasured Variables

In systems that are partially measured are generally solved by decomposing the data reconciliation problem into two sub-problems [26, 10]. In the first sub-problem, the measured variables are reconciled and in the second sub-problem the observable unmeasured variables are calculated. This methodology is efficient when compared to

solving for all the variables simultaneously [36]. The problem formulation and data reconciliation for partially measured systems is discussed below.

Let the variables be classified into two sets, one set is a vector X of measured variables and the other set a vector U of unmeasured variables. The objective function remains the same as described by equation 5.2 but the constraints are redefined in terms of the measured and unmeasured variables and are rewritten as,

$$A_x X + A_u U = 0 \tag{5.3}$$

Where,

A_x Columns of the matrix correspond to the measured variables (m x n)

A_u Columns of the matrix correspond to the unmeasured variables (m x p).

In the equation 5.3 the unmeasured variables U and the matrix A_u are eliminated by matrix projection method [10] to solve the data reconciliation problem in terms of measured variables. In order to eliminate the matrix A_u from equation 5.3, it is multiplied by a projection matrix P, which has the property,

$$PA_u = 0 \tag{5.4}$$

Thus the reduced constraints with only the measured variables are obtained by multiplying the equation 5.3 with projection matrix P.

$$PA_x X = 0 \tag{5.5}$$

In the current study QR factorization method is used for the construction of projection matrix. For details on QR factorization method [36] is referred by the author. In this

method, the matrix A_u is decomposed as the product of orthogonal matrix Q and an upper triangular matrix R:

$$A_u = Q.R = [Q_1.Q_2].\begin{bmatrix} R_1 \\ 0 \end{bmatrix} \tag{5.6}$$

Where,

Q (m x m) Orthogonal matrix $Q^T Q = I$

R_1 (p x p) Non-singular upper triangular matrix. It has maximal rank in R matrix.

A_u (m x p) Unmeasured variables matrix. All the columns of the matrix are linearly independent

In the equation 5.7 Q can be partitioned as Q_1 and Q_2 based on the dimensions of R_1. Multiplying equation 5.7 with $[Q_1 \quad Q_2]^T$ on both the sides we get

$$\begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}.A_u = \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}.[Q_1.Q_2].\begin{bmatrix} R_1 \\ 0 \end{bmatrix} \tag{5.7}$$

Since, Q is orthogonal matrix ($Q^T Q = I$). Equation 5.7 becomes,

$$\begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}.A_u = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \tag{5.8}$$

From the equation 5.8 we can clearly say that Q_2^T is the required projection matrix P. (Since, from equation 5.8 $Q_2^T A_u = 0$). The, reduced data reconciliation problem is given in the following form [36],

$$\begin{aligned} &\underset{X}{Min} && (Y-X)\Sigma^{-1}(Y-X) \\ &S.T && PA_x X = 0 \end{aligned} \tag{5.9}$$

The solution to this problem is given by

$$\hat{X} = Y - \Sigma(PA_x)^T \left[(PA_x)\Sigma(PA_x)^T\right]^{-1}(PA_x)Y \tag{5.10}$$

The solution of the minimization problem is further used to calculate the unmeasured variables using the equation 5.11, where the estimates of X from equation 5.10 are used in 5.11.

$$U = -R_1^{-1}Q_1^T A_x X \tag{5.11}$$

This approach is used in this research to solve the data reconciliation problem in 3 phases, in the first phase reconciliation of the flows using linear mass balance equations, and in the second phase calculate the Φ_{flow} factor for each heat exchanger in the network and in the third phase reconcile the temperatures using the linear energy balance equations.

5.3.2 Data reconciliation Methodology with QR Factorization

Step 1: Reconciliation of measured flows

$$\begin{aligned} &\underset{m_m}{Min} && (m_T - m_m)^T \Sigma^{-1}(m_T - m_m) \\ &S.T && P_m A_m m_m = 0 \end{aligned} \tag{5.12}$$

Where,

m_T - True measurements of flows

m_m - Measured values of flows.

P_m - Projection matrix obtained from QR factorization of A_u.

A_m - The columns of A_m correspond to measured variables (flows).

A_u - The columns of A_u correspond to unmeasured variables (flows)

Σ - Covariance matrix of the measurements

Step 2: Estimation of unmeasured flows from the reconciled flows. The solution for this problem is given by

$$\hat{m}_u = -R_{m1}^{-1} Q_{m1}^T A_m m_m \tag{5.13}$$

Where,

$\hat{m}_u$ - Estimated values of unmeasured flows

Q_{m1} - Orthogonal matrix obtained from QR factorization of A_u

R_{m1} - Nonsingular upper triangular matrix obtained from QR factorization of A_u

Step 3: Intermediate calculations between the mass balance phase and energy balance phase. Φ_{flow} and product of mass flow rate and specific heat capacity of inlet streams to heat exchangers are calculated in this phase

Step 4: Reconciliation of measured temperatures

$$\underset{m_m}{Min} \quad (T_T - T_m)^T \Sigma^{-1} (T_T - T_m)$$

$$S.T \quad P_T B_m T_m = 0 \tag{5.14}$$

Where,

T_T - True measurements of temperatures

T_m - Measured values of temperatures.

P_T - Projection matrix obtained from QR factorization of B_u.

B_m - The columns of B_m correspond to measured variables (temperatures).

B_u - The columns of B_u correspond to unmeasured variables (temperatures).

Σ - Covariance matrix of the measurements

Step 5: Estimation of unmeasured variables from the reconciled temperatures. The solution for this problem is given by

$$\hat{T}_u = -R_{T1}^{-1} Q_{T1}^T B_m T_m \tag{5.15}$$

Where,

$\hat{T}_u$ - Estimated values of unmeasured variables

Q_{T1} - Orthogonal matrix obtained from QR factorization of B_u

R_{T1} - Non-singular upper triangular matrix obtained from QR factorization of B_u

5.4 Parameter Observability Analysis

In data reconciliation problems, parameter observability analysis is used to detect ill-posed problems due to improper selection of measurements. It provides information whether the parameters are observable or unobservable with measurements chosen for data reconciliation and parameter estimation. Parameter observability analysis based on LU decomposition is used in the current research work. For, the implementation of this method [12] is referred by the author.

Let us consider a set of linear constraints:

$$AX = 0 \tag{5.16}$$

Where,

X : Denotes the process variable vector (n x 1)

A: A matrix of dimension m x n (Incidence matrix of the constraints)

The process variable vector X can be partitioned again into two vectors, measured variables vector X_x (r x 1) and unmeasured variables vector X_u (($n-r$) x 1). The unmeasured variables vector consists of state variables x and parameters θ (p x1)

$$X_u = \begin{bmatrix} x \\ \theta \end{bmatrix} \tag{5.17}$$

Now based on the measured and unmeasured variables the constraints of the equation can be partitioned and can be rewritten as

$$A_x X_x + A_u X_u = 0 \tag{5.18}$$

From the definition of observability [11], the observable variables are those variables which can be uniquely determined for a given set of measurements for a data reconciliation problem. As, a first step the observability is tested by checking the column rank of matrix A_u. If all the unmeasured variables X_u are observable in equation 5.18 then matrix A_u is a full column rank matrix ($rank(A_u) = n - r$).

If $rank(A_u) < n - r$, then all the unmeasured variables are not observable [11], in such cases, the unmeasured variables that are not observable for the given set of measurements are to be determined. So, to handle this problem observability analysis based on LU decomposition is used in this research to detect ill-posed problems and the unobservability of variables. In this research LU decomposition of matrix A_u is implemented using FUNCTION "LU" in MATLAB7.

$$E_1.(A_U).E_2 = L.U = L.\begin{bmatrix} U_1 & U_2 \\ 0 & 0 \end{bmatrix}.\begin{bmatrix} X_{u1} \\ X_{u2} \end{bmatrix} \tag{5.19}$$

Where,

E_1, E_2	Permutation matrices with permutes rows and columns of A_U
L	Unit low triangular matrix
U	Upper triangular matrix, portioned into matrices U_1 and U_2
U_1	Maximal nonsingular matrix in U
X_{u1}, X_{u2}	Permuted unmeasured variables corresponding to U_1 and U_2

In equation 5.19 if all the unmeasured variables are observable then matrix U_2 does not exist. If some unmeasured variables are unobservable, then the observable and unobservable variables can be identified using equation 5.20 [11]

$$\Gamma = U_1^{-1} U_2 \tag{5.20}$$

If the rows of matrix Γ have non-zero entries, then the corresponding unmeasured variables in X_{u1} are related to X_{u2} and those variables are unobservable. The remaining variables in X_{u1} are observable [11].

5.5 Application of Data Reconciliation

The proposed data reconciliation approach is applied to a case study available in the literature. In this case study the problem is successively solved in five steps; in step-1 reconcile measured flows using fmincon(SQP) in Matlab. In step-2 calculate unmeasured flows with the results obtained in step-1 using equation 5.13. In step-3 calculate the intermediate calculation phase (Φ factor). In step-4 reconcile measured temperatures using fmincon (SQP) in Matlab and in step-5 calculate the unmeasured temperatures with the results obtained in step-5 using equation 5.15.

5.5.1 Case study 1

This problem was earlier addressed by Swartz C.L.E [9]. In this study three scenarios are implemented on this model and the results obtained in the current study are compared with results in the literature [9]. In this case study the equations of the model are given in matrix form in MATLAB and optimized using fmincon in Matlab.

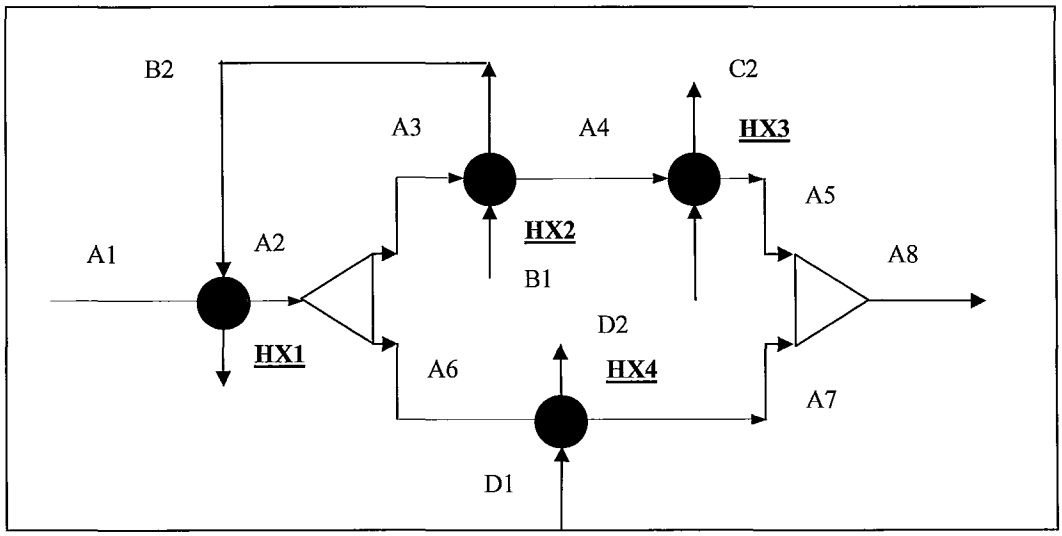


Figure 5.2: Process network flow diagram for case study 1, adapted from [9]

Run 1

STREAM	VBL	TAGNAME	STD. DEV.	MEASURED	CALCULATED (Current Study)	Literature [9]
A1	FLOW	FA1	20.00	1000.00	957.05	963.63
	TEMP	TA1	0.75	466.33	465.55	466.33
A2	FLOW				957.05	963.63
	TEMP				481.37	481.91
A3	FLOW	FA3	8.03	401.70	396.14	407.86
	TEMP	TA3	0.75	481.78	481.37	481.81
A4	FLOW				396.14	407.86
	TEMP	TA4	0.75	530.09	529.77	530.09
A5	FLOW				396.14	407.86
	TEMP	TA5	0.75	616.31	615.41	615.51
A6	FLOW	FA6	11.05	552.70	560.91	555.77
	TEMP				481.37	481.91
A7	FLOW				560.91	555.77
	TEMP	TA7	0.75	619.00	617.04	617.76
A8	FLOW				957.05	963.63
	TEMP	TA8	0.75	614.92	616.36	616.81
B1	FLOW	FB1	5.06	253.20	249.01	253.20
	TEMP	TB1	0.75	618.11	617.05	618.11
B2	FLOW				249.01	253.20
	TEMP				541.45	543.9
B3	FLOW				249.01	253.20
	TEMP				481.74	486.5
C1	FLOW	FC1	6.16	308.10	298.17	308.10
	TEMP	TC1	0.75	694.99	693.84	694.9
C2	FLOW				298.17	308.10
	TEMP				582.12	594.8
D1	FLOW				669.49	689.42
	TEMP	TD1	0.75	667.84	667.66	668.02
D2	FLOW	FD2	13.6	680.10	669.49	689.42
	TEMP	TD2	0.75	558.34	556.05	558.17

Table 5.1: Simulation results for case study 1 (run 1)

Results summary: Run 1

In this case study the reconciliation problem is carried out in two steps, in the first step reconciliation of flows and in the second step reconciliation of temperatures. The results obtained with the current approach and the earlier approach is presented in Table 5.1. In the current approach the number of iterations it took to converge is 10 iterations but in earlier approach it took 4 iterations to converge.

From the observation of the simulation results presented in Table 5.1, there are slight deviations in the final values of the flows obtained with the current approach and the earlier results but the temperatures could converge close to the earlier results. The convergence results for this case study are presented in Table 5.2 and Table 5.3.

Reconciliation of Flows:

Number of Iterations	6
Measured variables	6
Unmeasured variables	9
Number of equations	10

Table 5.2: Convergence results for reconciliation of flows case study 1 (run 1)

Reconciliation of Temperatures

Number of Iterations	4
Measured variables	10
Unmeasured variables	5
Number of equations	11

Table 5.3: Convergence results for reconciliation of temperatures for case study 1 (run 1)

Run 2

STREAM	VBL	TAGNAME	STD. DEV.	MEASURED	CALCULATED (Current Study)	Literature [9]
A1	FLOW	FA1	20.00	1000.00	957.27	969.12
	TEMP	TA1	0.75	466.33	465.75	466.33
A2	FLOW				957.27	969.12
	TEMP				481.48	481.77
A3	FLOW	FA3	8.03	401.70	402.16	406.68
	TEMP	TA3	0.75	481.78	481.48	481.77
A4	FLOW				402.16	406.68
	TEMP	TA4	0.75	530.09	529.3	530.09
A5	FLOW				402.16	406.68
	TEMP	TA5	0.75	616.31	614.77	616.31
A6	FLOW	FA6	11.05	552.70	555.11	562.44
	TEMP				481.48	481.77
A7	FLOW				555.11	562.44
	TEMP	TA7	0.75	619.00	617.18	613.94
A8	FLOW				957.27	969.12
	TEMP	TA8	0.75	614.92	616.17	614.93
B1	FLOW	FB1	5.06	253.20	248.42	253.2
	TEMP	TB1	0.75	618.11	617.36	618.11
B2	FLOW				248.42	253.2
	TEMP				541.35	543.91
B3	FLOW				248.42	253.2
	TEMP				481.8	486.71
C1	FLOW	FC1	6.16	308.10	301.7	308.1
	TEMP	TC1	0.75	694.99	694.22	694.99
C2	FLOW				301.7	308.1
	TEMP				582.35	594.1
D1	FLOW				666.05	679.72
	TEMP	TD1	0.75	667.84	666.93	667.83
D2	FLOW	FD2	13.6	680.10	666.05	679.72
	TEMP	TD2	0.75	558.34	555.88	558.35

Table 5.4: Simulation results for case study 1 (run 2)

Results summary: Run 2

The results obtained with the current approach and the earlier approach is presented in Table 5.4. In the current approach the number of iterations it took to converge is 10 iterations but in earlier approach it took 4 iterations to converge. The convergence results for this case study are presented in Table 5.5 and Table 5.6.

Reconciliation of Flows:

Number of Iterations	6
Measured variables	6
Unmeasured variables	9
Number of equations	10

Table 5.5: Convergence results for reconciliation of flows for case study 1 (run 2)

Reconciliation of Temperatures:

Number of Iterations	4
Measured variables	9
Unmeasured variables	6
Number of equations	11

Table 5.6: Convergence results for reconciliation of temperatures for case study 1 (run 2)

Run 3

STREAM	VBL	TAGNAME	STD. DEV.	MEASURED	CALCULATED (Current Study)	Literature [9]
A1	FLOW	FA1	20.00	1000.00	957.27	969.12
	TEMP	TA1	0.75	466.33	465.76	466.33
A2	FLOW				957.27	969.12
	TEMP				481.49	481.77
A3	FLOW	FA3	8.03	401.70	402.16	406.68
	TEMP	TA3	0.75	481.78	481.49	481.77
A4	FLOW				402.16	406.68
	TEMP	TA4	0.75	530.09		
A5	FLOW				402.16	406.68
	TEMP	TA5	0.75	616.31	614.77	616.31
A6	FLOW	FA6	11.05	552.70	555.11	562.44
	TEMP				481.49	481.77
A7	FLOW				555.11	562.44
	TEMP	TA7	0.75	619.00	617.18	613.94
A8	FLOW				957.27	969.12
	TEMP	TA8	0.75	614.92	616.17	614.93
B1	FLOW	FB1	5.06	253.20	248.42	253.2
	TEMP	TB1	0.75	618.11	617.38	618.11
B2	FLOW				248.42	253.2
	TEMP					
B3	FLOW				248.42	253.2
	TEMP					
C1	FLOW	FC1	6.16	308.10	301.7	308.1
	TEMP	TC1	0.75	694.99	694.21	694.99
C2	FLOW				301.7	308.1
	TEMP					
D1	FLOW				666.05	679.72
	TEMP	TD1	0.75	667.84	666.93	667.83
D2	FLOW	FD2	13.6	680.10	666.05	679.72
	TEMP	TD2	0.75	558.34	555.88	558.35

Table 5.7: Simulation results for case study 1 (run 3)

Results summary: Run 3

The results obtained with the current approach and the earlier approach is presented in Table 5.7. In the current study, the convergence results for this case study are presented in Table 5.8 and Table 5.9.

Reconciliation of Flows:

Number of Iterations	6
Measured variables	6
Unmeasured variables	9
Number of equations	10

Table 5.8: Convergence results for reconciliation of flows for case study 1 (run 3)

Reconciliation of Temperatures:

Number of Iterations	4
Measured variables	8
Unmeasured variables	7
Number of equations	11

Table 5.9: Convergence results for reconciliation of temperatures for case study 1 (run 3)

A framework for steady state data reconciliation using the HEN simulation algorithm combined with QR factorization is proposed. The key value that this work has added to the previous work on data reconciliation using QR factorization is, the constraints in this approach are linear, while the constraints in the previous work are nonlinear. In addition, in the current approach the data reconciliation is solved in two steps in first step reconciliation of mass flow rates is implemented and in the second step reconciliation of temperatures. Both the approaches of data reconciliation proved to be functioning well for the case study analyzed in this research. The current approach of data reconciliation can be used in the cases when the mass flow rates and the temperatures variables are observable independently. In cases, when they are to be calculated simultaneously, they can be solved as per the earlier published method using SQP.

5.5.2 Case Study 2

In this section, the application of the HEN simulation algorithm combined with observability analysis is demonstrated on data reconciliation and to identify, correct heat transfer coefficients for individual exchangers in the plant-model mismatch scenario. The process model example for this case study is taken from [32]

In this case study the plant (model-1) is perturbed by $\pm$ 10% deviations in the heat exchanger heat transfer coefficients (U) from the base value (U = 150 BTU / ft^2 °F) and model-1 is simulated. The obtained simulation results from model-1 are considered as the true measurements. These measurements are inputs to model-2 that has heat exchangers all with U = 150 BTU / ft^2 °F (base values) and the split-fractions are fixed with the one entered in the model-1 (α_1= 0.3936, α_2= 0.2289), refer Figure 5.4 for the corresponding streams. The model is run to minimize the deviations between the plant (model-1) measurements and measurements from the model-2 by changing heat exchanger heat coefficients (U) in model-2. In this case study only the reconciliation of temperature variables is taken account (assumed the flows are correct from model-1).

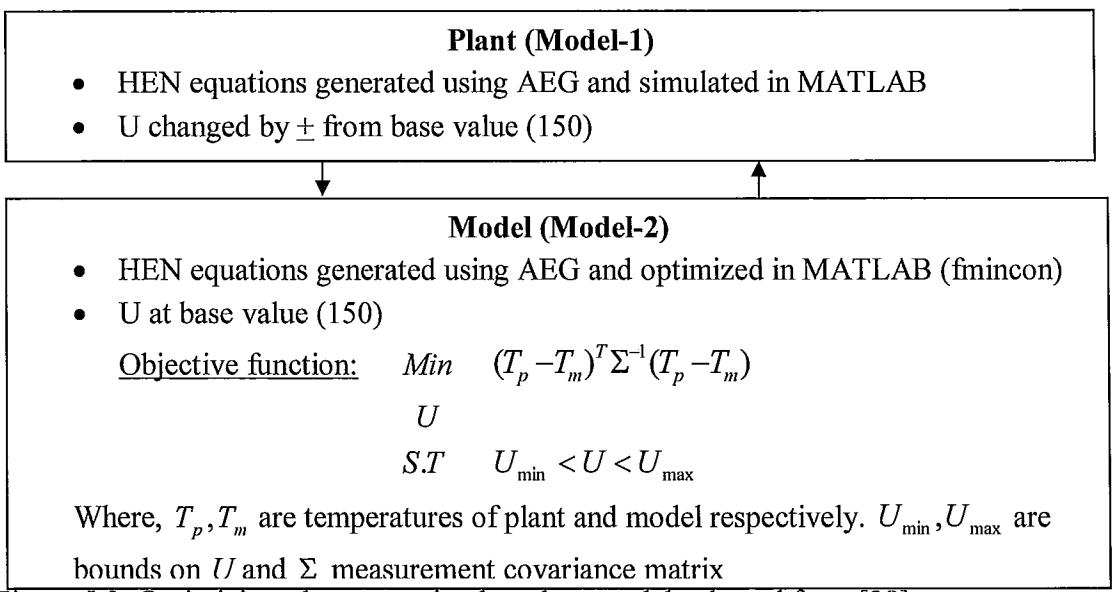


Figure 5.3: Optimizing plant operation based on model, adapted from [28]

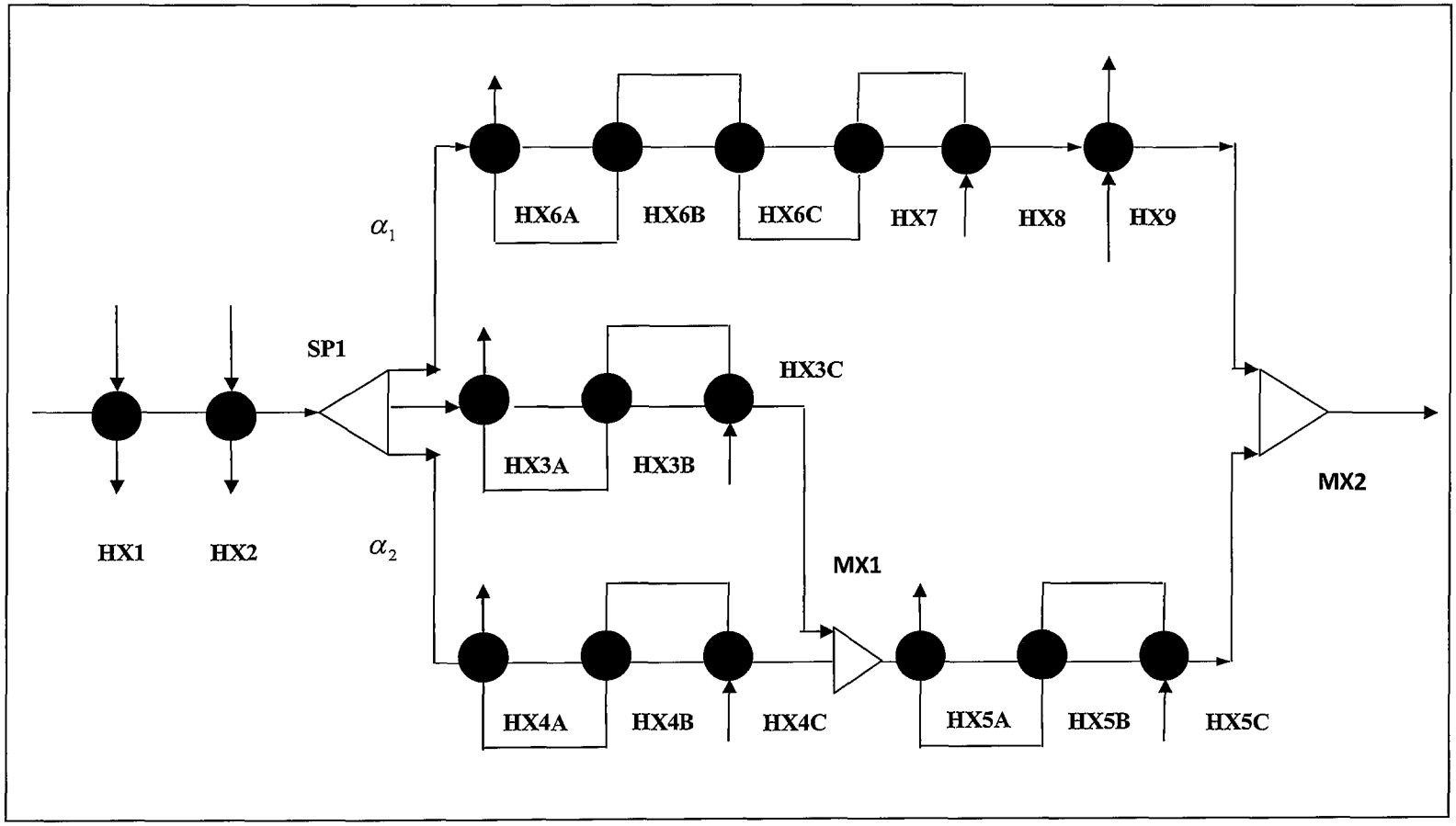


Figure 5.4: Process network flow diagram for case study 2, adapted from Goyal [32]

This case study was implemented for three scenarios with different measurements sets. All the equations for this case study are generated using AEG and the model was optimized in MATLAB using fmincon

Run 1

Measurements selected for run 1 are shown in Table 5.10, (The flow-sheet with the corresponding streams highlighted is presented in Figure 5.5). Observability analysis discussed in section 5.4 is applied on the energy balance equations, the following observations are made.

Let the energy balance equations be represented as s

$$Ax = b \tag{5.21}$$

1. Size (A) = 154 x 154
2. Rank (A) = 137 (So the minimum number of measurements required for this problem are 17, to make all the unmeasured variables observable).
3. The number of measurements considered in this case is 22 which are listed in Table 5.10: the matrix A is split into A_X and A_U based on the measured and unmeasured variables. By analyzing the matrix A_U, the following observations are made,
4. Size (A_U) = 154 x 132
5. Rank (A_U) = 125. This clearly shows the problem is ill-conditioned even though the number of measured variables (22) considered is more than required (17). So some of the unmeasured variables are unobservable. The unmeasured variables that are unobservable can be indentified from $\Gamma = U_1^{-1}U_2$. The results of observability analysis is presented in Table 5.12
6. It took 12 iterations to converge and the results of this case study are presented in Table 5.11.

Selected Measurements		
S.No	**Variable Name**	**Node Name**
1	HX1-HOT_IN_TEMP	HX1
2	HX1-COLD_IN_TEMP	HX1
3	HX2-HOT_IN_TEMP	HX2
4	HX3A-HOT_OUT_TEMP	HX3A
5	HX3A-COLD_IN_TEMP	HX3A
6	HX3B-COLD_IN_TEMP	HX3B
7	HX3C-HOT_IN_TEMP	HX3C
8	HX4A-HOT_OUT_TEMP	HX4A
9	HX4A-COLD_IN_TEMP	HX4A
10	HX4B-COLD_IN_TEMP	HX4B
11	HX4C-HOT_IN_TEMP	HX4C
12	HX5A-HOT_OUT_TEMP	HX5A
13	HX5A-COLD_OUT_TEMP	HX5A
14	HX5C-HOT_IN_TEMP	HX5C
15	HX5C-COLD_OUT_TEMP	HX5C
16	HX6A-HOT_OUT_TEMP	HX6A
17	HX6C-COLD_IN_TEMP	HX6C
18	HX6C-COLD_OUT_TEMP	HX6C
19	HX8-HOT_IN_TEMP	HX8
20	HX9-HOT_IN_TEMP	HX9
21	HX9-HOT_OUT_TEMP	HX9
22	MX2_COLD_OUT_TEMP	MX2

Table: 5.10: Measurements considered for case study 2 (run1)

S.No	Node Name	True Values of U	Estimated Values of U
1	HX1	160	161.1094
2	HX2	162	161.1599
3	HX3A	165	164.9971
4	HX3B	148	151.0013
5	HX3C	154	151.0013
6	HX4A	145	144.9989
7	HX4B	137	144.0009
8	HX4C	151	144.0009
9	HX5A	153	153.0002
10	HX5B	146	144.5007
11	HX5C	143	144.5007
12	HX6A	153	155.5006
13	HX6B	158	155.5006
14	HX6C	148	148.0013
15	HX7	136	153.2799
16	HX8	165	156.9316
17	HX9	156	155.9983

Table 5.11: True & estimated values of heat transfer coefficients for case study 2 (run1)

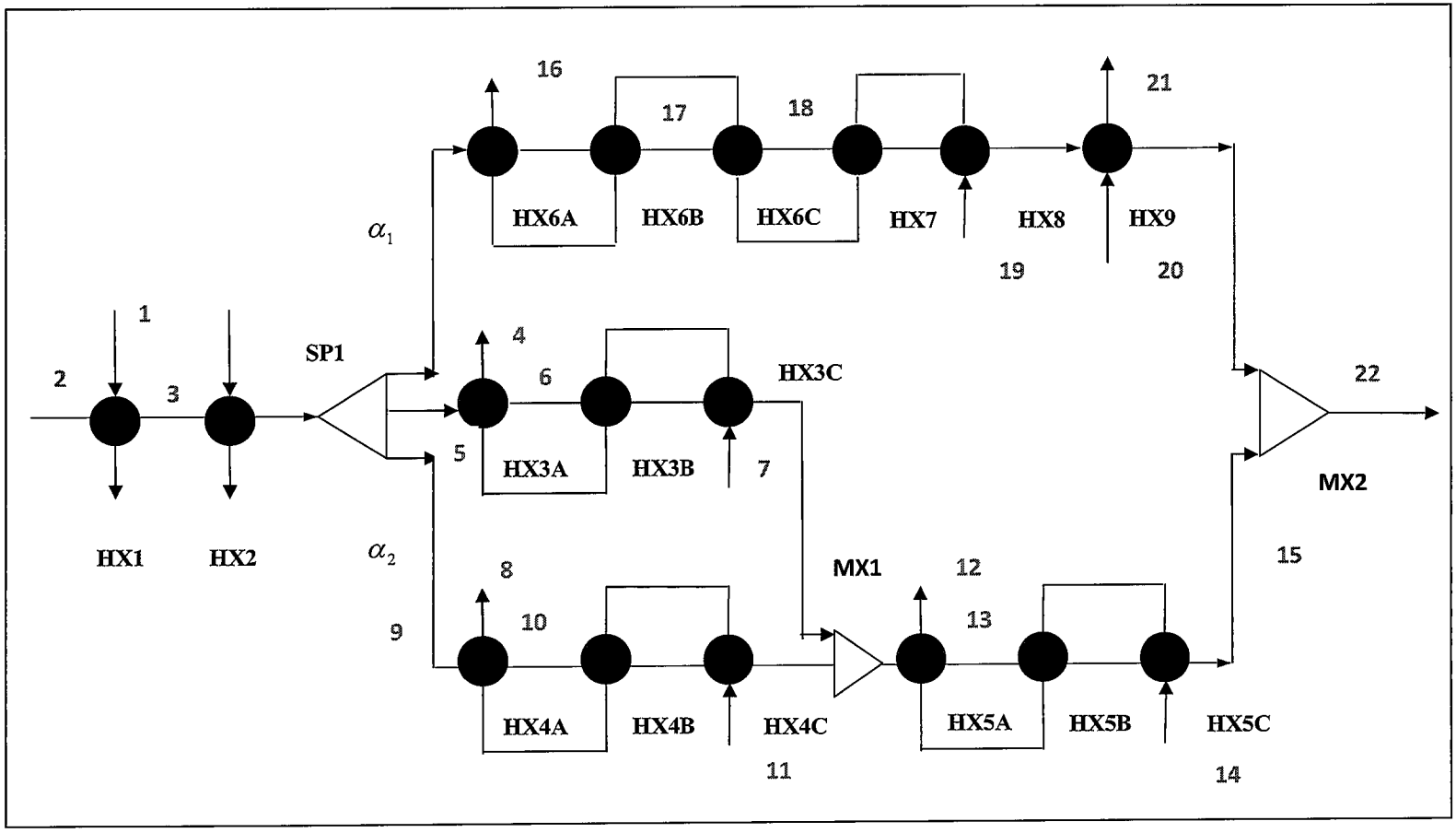


Figure 5.5: Measurements considered for case study 2 (run 1) adapted from Goyal [32]

$\Gamma = U_1^{-1}U_2$ (only non-zeros entries)							Variable Name	Node	Plant Value	Model Value
0	0	0	0	0	0	0	HX1-HOT_OUT_TEMP	HX1	176.8318	176.1725
0	0	0	0	4.1491	0	0	HX2-HOT_OUT_TEMP	HX2	234.5013	235.1978
0	0	0	0	-511225.2	0	0	DUTY_HOT	HX2	30372157.13	30286336.97
0	0	0	0	-511225.2	0	0	DUTY_COLD	HX2	30372157.13	30286336.97
0	0	0	0	-0.0183	0	0	OP	HX2	1.0849	1.0819
0	0	0	0	-511225.2	0	0	DUTY_OP	HX2	30372157.13	30286336.97
0	0	0	0	0	0	0.6492	MX2_COLD_IN2_TEMP	HX5C-MX2	547.388	547.388
0	0	0	0	0	0	0.4684	HX6A-HOT_IN_TEMP	HX6A	362.5652	363.4077
0	0	0	0	0	0	101482.44	DUTY_HOT	HX6A	11226710.3	11409245.93
0	0	0	0	0	0	-0.4957	HX6A-COLD_OUT_TEMP	HX6A	255.9078	256.8148
0	0	0	0	0	0	101482.44	DUTY_COLD	HX6A	11226710.34	11409245.93
0	0	0	0	0	0	0.0047	OP	HX6A	0.5239	0.5324
0	0	0	0	0	0	101482.44	DUTY_OP	HX6A	11226710.34	11409245.93
0	0	0	0	0	0	-0.4957	HX6B-COLD_IN_TEMP	HX6A-HX6B	255.9078	256.8148
0	0	0	0	0	0	0.9288	HX6B-HOT_IN_TEMP	HX6B	414.1454	414.1454
0	0	0	0	0	0	0.4684	HX6B-HOT_OUT_TEMP	HX6B	362.5652	363.4077
0	0	0	0	0	0	99766.089	DUTY_HOT	HX6B	11175718.22	10993182.7
0	0	0	0	0	0	99766.089	DUTY_COLD	HX6B	11175718.22	10993182.7
0	0	0	0	0	0	0.0047	OP	HX6B	0.5215	0.513
0	0	0	0	0	0	99766.089	DUTY_OP	HX6B	11175718.22	10993182.7
0	0	0	0	0	0	0.9288	HX6C-HOT_IN_TEMP	HX6C	460.6865	460.6865
0	0	0	0	0	0	0.9288	HX6C-HOT_OUT_TEMP	HX6C	414.1454	414.1454
0	0	0	0	0	0	1.258	HX7-HOT_IN_TEMP	HX7	495.8145	500.2584
0	0	0	0	0	0	0.9288	HX7-HOT_OUT_TEMP	HX7	460.6865	460.6865
0	0	0	0	0	0	71307.901	DUTY_HOT	HX7	7611081.281	8573917.997
0	0	0	0	0	0	0.3543	HX7-COLD_OUT_TEMP	HX7	399.3657	404.15
0	0	0	0	0	0	71307.901	DUTY_COLD	HX7	7611081.281	8573917.997
0	0	0	0	0	0	0.0046	OP	HX7	0.4912	0.5534
0	0	0	0	0	0	71307.901	DUTY_OP	HX7	7611081.281	8573917.997
0	0	0	0	0	0	0.3543	HX8-COLD_IN_TEMP	HX7-HX8	399.3657	404.15
0	0	0	0	0	0	1.258	HX8-HOT_OUT_TEMP	HX8	495.8145	500.2584
0	0	0	0	0	0	-272556.4	DUTY_HOT	HX8	18240187.24	17277350.27
0	0	0	0	0	0	-272556.4	DUTY_COLD	HX8	18240187.24	17277350.27
0	0	0	0	0	0	-0.009	OP	HX8	0.6038	0.5719
0	0	0	0	0	0	-272556.4	DUTY_OP	HX8	18240187.24	17277350.27

Table 5.12: Observability analysis output for case study 2 (run 1)

In this case study, from the degree of freedom analysis the minimum number of measurements required is 17 for all the unmeasured variables to be observable.

In this problem 22 measurements are considered, but from the observability analysis it is evident that some unmeasured variables are unobservable. These unobservable variables have been captured by $\Gamma = U_1^{-1}U_2$ matrix calculations. The non-zero entries of $\Gamma = U_1^{-1}U_2$ matrix and the corresponding plant and model measurements are shown in Table 5.12.

In addition, the reasons for the non-convergence of heat transfer coefficients of HX7 and HX8 are due to improper selections of measurements around these two heat exchangers. Since, no calculated variable temperatures (i.e., hot or cold outlet temperatures) are taken into account there are chances of getting struck at local optimal solutions.

Run 2

Measurements selected for run 2 are shown in Table 5.13, (The flow-sheet with the corresponding streams highlighted is presented in Figure 5.6). From the observability analysis on the energy balance equations the following observations are made.

1. The number of measurements considered in this case is 13 which are listed in Table 5.13: By analyzing the matrix A_U, the following observations are made,
2. Size (A_U) = 154 x 141
3. Rank (A_U) = 137. This clearly shows the problem is ill-conditioned the number of measured variables (13) considered is less than required (17). So some of the unmeasured variables are unobservable. The results of observability analysis is presented in Table 5.15
4. It took 15 iterations to converge and the results of this case study are presented in Table 5.14.

Selected Measurements		
S.No	**Variable Name**	**Node Name**
1	HX1-HOT_OUT_TEMP	HX2
2	HX2-HOT_OUT_TEMP	HX3A
3	HX3B-HOT_OUT_TEMP	HX3B
4	HX3C-HOT_OUT_TEMP	HX3C
5	HX4A-HOT_OUT_TEMP	HX4A
6	HX4C-HOT_OUT_TEMP	HX4B
7	HX5A-HOT_OUT_TEMP	HX4C
8	HX5B-HOT_OUT_TEMP	HX5A
9	HX6A-HOT_OUT_TEMP	HX5B
10	HX6B-HOT_OUT_TEMP	HX6A
11	HX6C-HOT_OUT_TEMP	HX6B
12	HX7-HOT_OUT_TEMP	HX6C
13	HX9-HOT_OUT_TEMP	HX7

Table: 5.13 Measurements considered for case study 2 (run2)

S.No	Node Name	True Values of U	Estimated Values of U
1	HX1	160	160
2	HX2	162	162
3	HX3A	165	150.2236
4	HX3B	148	145.4831
5	HX3C	154	152.6613
6	HX4A	145	141.0003
7	HX4B	137	141.0003
8	HX4C	151	150.9998
9	HX5A	153	152.2341
10	HX5B	146	143.8974
11	HX5C	143	143.8974
12	HX6A	153	152.9972
13	HX6B	158	158.0005
14	HX6C	148	148.0003
15	HX7	136	153.2622
16	HX8	165	156.9371
17	HX9	156	156.0011

Table 5.14: True & estimated values of heat transfer coefficients for case study 2 (run 2)

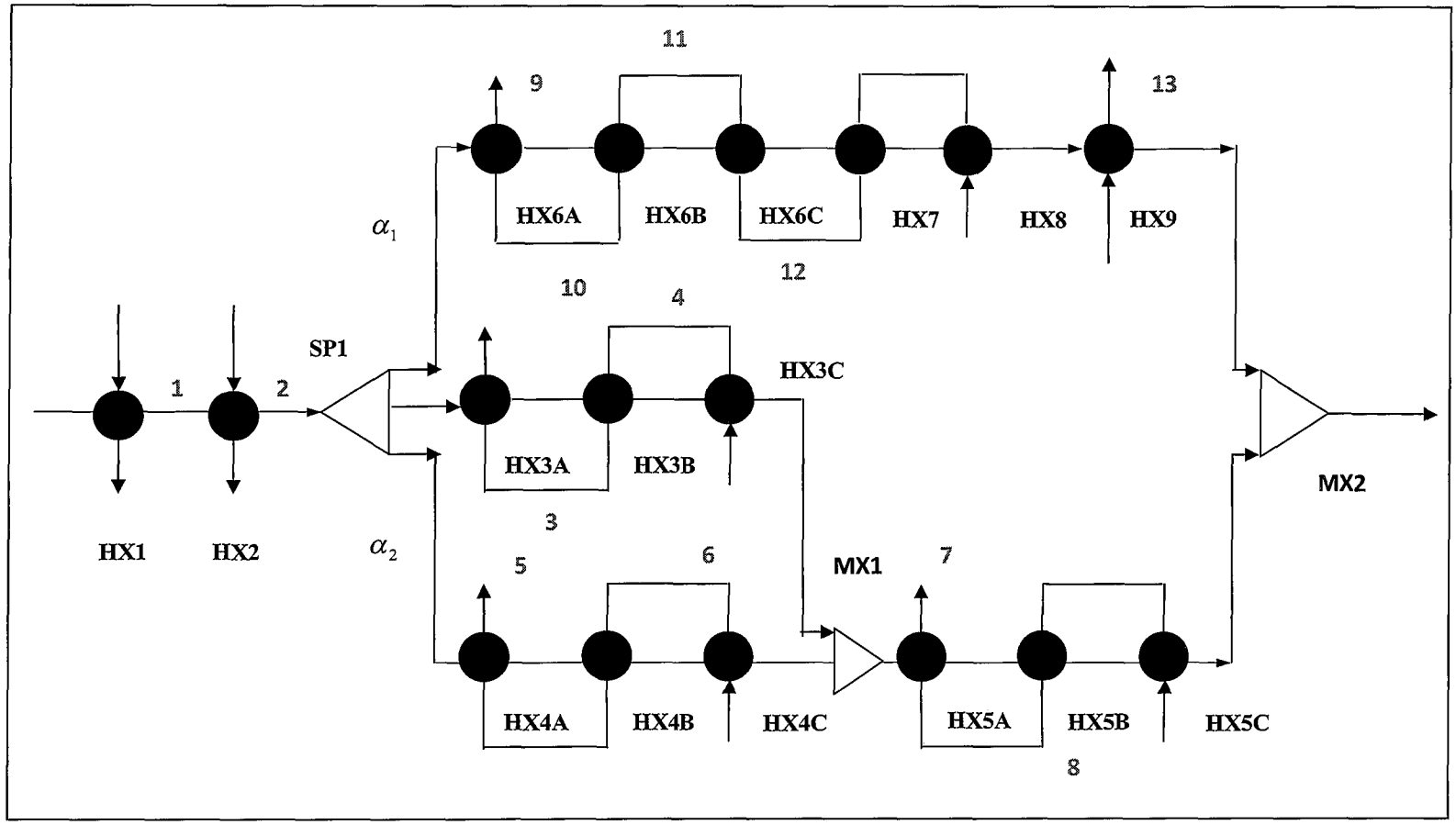


Figure 5.6: Measurements considered for case study 2 (run 2) adapted from Goyal [32]

$\Gamma = U_1^{-1}U_2$ (only non-zeros entries)					Variable Name	Node	Plant Value	Model Value
0.00	0.00	-1.04	0.00	0.00	HX3A-HOT_IN_TEMP	HX3A	263.14	259.9925
0.00	0.00	-64242.53	0.00	0.00	DUTY_HOT	HX3A	2739969	2544876
0.00	0.00	-1.55	0.00	0.00	HX3A-COLD_OUT_TEMP	HX3A	223.5456	221.8778
0.00	0.00	-64242.53	0.00	0.00	DUTY_COLD	HX3A	2739969	2544876
0.00	0.00	-0.01	0.00	0.00	OP	HX3A	0.2194	0.2038
0.00	0.00	-64242.53	0.00	0.00	DUTY_OP	HX3A	2739969	2544876
0.00	0.00	-1.55	0.00	0.00	HX3B-COLD_IN_TEMP	HX3A-HX3B	223.5456	221.8778
0.00	0.00	-1.04	0.00	0.00	DUTY_HOT	HX3B	4960852	5155945
0.00	0.00	64242.53	0.00	0.00	HX3B-COLD_OUT_TEMP	HX3B	265.9544	265.9544
0.00	0.00	-1.00	0.00	0.00	DUTY_COLD	HX3B	4960852	5155945
0.00	0.00	64242.53	0.00	0.00	OP	HX3B	0.3973	0.4129
0.00	0.00	0.01	0.00	0.00	DUTY_OP	HX3B	4960852	5155945
0.00	0.00	64242.53	0.00	0.00	HX3C-COLD_IN_TEMP	HX3B-HX3C	265.9544	265.9544
0.00	0.00	0.00	-0.34	0.00	HX4A-COLD_IN_TEMP	HX4A	200.1225	200.1225
0.00	0.00	0.00	32326.21	0.00	HX4A-COLD_OUT_TEMP	HX4A	230.8399	232.0343
0.00	0.00	0.00	-1.00	0.00	DUTY_COLD	HX4A	5928441	6158963
0.00	0.00	0.00	-0.83	0.00	OP	HX4A	0.6892	0.716
0.00	0.00	0.00	32326.21	0.00	DUTY_OP	HX4A	5928441	6158963
0.00	0.00	0.00	0.00	0.00	HX4B-COLD_IN_TEMP	HX4A-HX4B	230.8399	232.0343
0.00	0.00	0.00	32326.21	0.00	HX4B-HOT_IN_TEMP	HX4B	457.3888	457.3888
0.00	0.00	0.00	-0.83	0.00	DUTY_HOT	HX4B	7020133	7020133
0.00	0.00	0.00	0.00	0.00	HX4B-COLD_OUT_TEMP	HX4B	267.2137	268.4081
0.00	0.00	0.00	0.00	0.00	DUTY_COLD	HX4B	7020133	7020133
0.00	0.00	0.00	-0.83	0.00	OP	HX4B	0.8161	0.8161
0.00	0.00	0.00	0.00	0.00	HX4C-COLD_IN_TEMP	HX4B-HX4C	267.2137	268.4081
0.00	0.00	0.00	-0.83	0.00	DUTY_HOT	HX4C	9751255	9751255
0.00	0.00	0.00	-0.83	0.00	OP	HX4C	1.1336	1.1336
0.00	0.00	0.00	-0.83	0.00	DUTY_HOT	HX5A	20463165	20463165
0.00	0.00	0.00	0.00	0.00	HX5A-COLD_IN_TEMP	HX5A	331.1551	331.8988
0.00	0.00	0.00	0.00	0.00	HX5A-COLD_OUT_TEMP	HX5A	397.1703	397.914
0.00	0.00	-0.38	-0.52	0.00	DUTY_COLD	HX5A	20463165	20463165
0.00	0.00	-0.38	-0.52	0.00	OP	HX5A	0.4938	0.4938
0.00	0.00	0.00	0.00	0.00	HX5B-COLD_IN_TEMP	HX5A-HX5B	397.1703	397.914
0.00	0.00	0.00	0.00	0.00	HX5B-HOT_IN_TEMP	HX5B	548.7958	548.7479
0.00	0.00	-0.38	-0.52	0.00	DUTY_HOT	HX5B	21863797	21850670
0.00	0.00	-0.37	-0.51	0.00	HX5B-OLD_OUT_TEMP	HX5B	467.704	468.4053
0.00	0.00	-101746.6	-139753.7	0.00	DUTY_COLD	HX5B	21863797	21850670
0.00	0.00	-0.71	-0.97	0.00	OP	HX5B	0.5276	0.5273
0.00	0.00	-101746.6	-139753.7	0.00	DUTY_OP	HX5B	21863797	21850670
0.00	0.00	0.00	0.00	0.00	HX5C-COLD_IN_TEMP	HX5B-HX5C	467.704	468.4053
0.00	0.00	-101746.6	-139753.7	0.00	HX5C-HOT_IN_TEMP	HX5C	639	639
0.00	0.00	-0.71	-0.97	0.00	DUTY_HOT	HX5C	24700208	24713335

0.00	0.00	0.00	0.00	0.00	HX5C-COLD_OUT_TEMP	HX5C	547.388	548.1317
0.00	0.00	-0.37	-0.51	0.00	DUTY_COLD	HX5C	24700208	24713335
0.00	0.00	101746.6	139753.7	0.00	OP	HX5C	0.5961	0.5964
0.00	0.00	-0.38	-0.52	0.00	DUTY_OP	HX5C	24700208	24713335
0.00	0.00	101746.64	139753.77	0.00	MX2_COLD_IN2_TEMP	HX5C-MX2	547.388	548.1317
0.00	0.00	0.00	0.00	0.00	HX6A-HOT_IN_TEMP	HX6A	362.5652	362.5652
0.00	0.00	101746.64	139753.77	0.00	DUTY_HOT	HX6A	11226710	11226710
0.00	0.00	-0.38	-0.52	0.00	HX6A-COLD_IN_TEMP	HX6A	200.1225	200.1225
0.00	0.00	0.00	0.00	0.40	DUTY_COLD	HX7	8573030	8573317
0.00	0.00	0.00	0.00	86473.89	OP	HX7	0.5533	0.5533
0.00	0.00	0.00	0.00	-0.57	DUTY_OP	HX7	8573030	8573317
0.00	0.00	0.00	0.00	86473.89	HX8-COLD_IN_TEMP	HX7-HX8	404.1456	404.147
0.00	0.00	0.00	0.00	0.01	HX8-HOT_IN_TEMP	HX8	580	580
0.00	0.00	0.00	0.00	86473.89	DUTY_HOT	HX8	17278238	17277952
0.00	0.00	0.00	0.00	-0.57	HX8-COLD_OUT_TEMP	HX8	490.0008	490.0008
-1.00	0.00	0.00	0.00	0.00	DUTY_COLD	HX8	17278238	17277952
0.00	0.00	0.00	0.00	0.40	OP	HX8	0.572	0.5719
-216666.6	0.00	0.00	0.00	-86473.8	DUTY_OP	HX8	17278238	17277952
-1.08	0.00	0.00	0.00	-1.00	HX9-COLD_IN_TEMP	HX8-HX9	490.0008	490.0008
-216666.6	0.00	0.00	0.00	-86473.8	HX9-HOT_IN_TEMP	HX9	650	650
-0.01	0.00	0.00	0.00	0.00	HX9-HOT_OUT_TEMP	HX9	600.416	600.416
-216666.67	0.00	0.00	0.00	-86473.8	DUTY_HOT	HX9	7137089	7137089
-1.08	0.00	0.00	0.00	-1.00	HX9-COLD_OUT_TEMP	HX9	525.4649	525.4649
0.00	-143939.3	0.00	0.00	0.00	OP	HX9	0.7647	0.7647
-1.08	-0.72	0.00	0.00	-1.00	DUTY_OP	HX9	7137089	7137089
0.00	-143939.3	0.00	0.00	0.00	MX2_COLD_IN1_TEMP	HX9-MX2	525.4649	525.4649
0.00	-0.02	0.00	0.00	0.00	MX1_COLD_OUT_TEMP	MX1	331.1551	331.8988
0.00	-143939.3	0.00	0.00	0.00	MX2_COLD_OUT_TEMP	MX2	538.7578	539.2087

Table 5.15: Observability analysis output for case study 2 (run 2)

In this run, the number of measurements considered is 13, which is less than the minimum required for the system to be completely observable. This case study was implemented to check whether all the unobservable variables could be captured by $\Gamma = U_1^{-1}U_2$ matrix calculation. From these calculations, majority of the unobservable variables could be identified, but some of the non-zero entries in $\Gamma = U_1^{-1}U_2$ corresponded to certain entries for which the plant and the model values are same (Please check the highlighted entries in Table 5.12).

Run 3

Measurements selected for run 3 are shown in Table 5.16, (The flow-sheet with the corresponding streams highlighted is presented in Appendix C, Figure C.1). From the observability analysis on the energy balance equations the following observations are made.

1. The number of measurements considered in this case is 17 which are listed in Table 5.16: By analyzing the matrix A_U, the following observations are made,
2. Size (A_U) = 154 x 137
3. Rank (A_U) = 137. In this case the matrix A_U is a full rank matrix. So, all the unmeasured variables in this case are observable. For this case, since the matrix A_U is full rank matrix on LU decomposition U_2 does not exists.
4. It took 15 iterations to converge and the results of this case study are presented in Table 5.17.

Selected Measurements		
S.No	**Variable Name**	**Node Name**
1	HX1-HOT_OUT_TEMP	HX1
2	HX2-HOT_OUT_TEMP	HX2
3	HX3A-HOT_OUT_TEMP	HX3A
4	HX3B-HOT_OUT_TEMP	HX3B
5	HX3C-HOT_OUT_TEMP	HX3C
6	HX4A-HOT_OUT_TEMP	HX4A
7	HX4B-HOT_OUT_TEMP	HX4B
8	HX4C-HOT_OUT_TEMP	HX4C
9	HX5A-HOT_OUT_TEMP	HX5A
10	HX5B-HOT_OUT_TEMP	HX5B
11	HX5C-HOT_OUT_TEMP	HX5C
12	HX6A-HOT_OUT_TEMP	HX6A
13	HX6B-HOT_OUT_TEMP	HX6B
14	HX6C-HOT_OUT_TEMP	HX6C
15	HX7-HOT_OUT_TEMP	HX7
16	HX8-HOT_OUT_TEMP	HX8
17	HX9-HOT_OUT_TEMP	HX9

Table: 5.16 Measurements considered for case study 2 (run 2)

S.No	Node Name	True Values of U	Estimated Values of U
1	HX1	160	160.0001
2	HX2	162	162.0001
3	HX3A	165	165
4	HX3B	148	148.0007
5	HX3C	154	153.9999
6	HX4A	145	145.0014
7	HX4B	137	136.9999
8	HX4C	151	151.0001
9	HX5A	153	153.0031
10	HX5B	146	146.0007
11	HX5C	143	143.001
12	HX6A	153	153.0009
13	HX6B	158	158
14	HX6C	148	147.9979
15	HX7	136	136.0054
16	HX8	165	164.9991
17	HX9	156	156.0008

Table 5.17: True & estimated values of heat transfer coefficients for case study 2 (run 2)

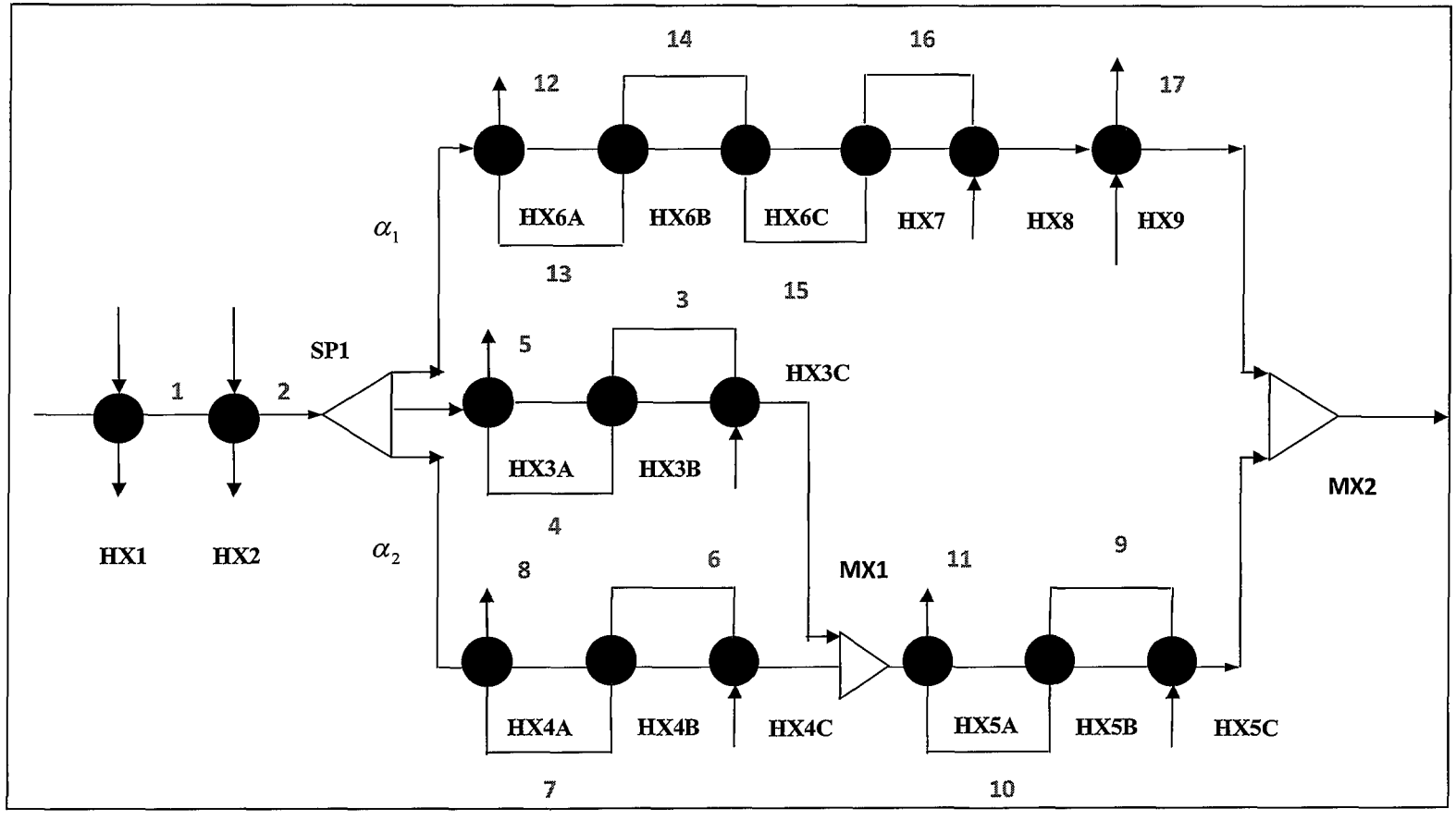


Figure 5.7: Measurements considered for case study 2 (run 3) adapted from Goyal [32]

In this run, all the calculated hot outlet temperatures of each individual heat exchanger is considered. From the observability analysis, it is clear that the matrix A_U is full rank matrix and on LU decomposition U_2 does not exist. Therefore all the unmeasured variables in this system are observable. From the optimization of this problem the heat transfer coefficients of the model converged to that of the plant.

5.6 Chapter Summary

In this chapter, framework for steady state data reconciliation of HEN in two steps is developed, which involves reconciliation of flows in the first step and reconciliation of temperatures in the second step. This approach is successfully implemented on a case study from the literature.

In another case study data reconciliation combined with observability analysis is implemented to correct the heat transfer coefficients in plant-model mismatch scenario and several case studies are implemented to show the applicability of the current HEN simulation algorithm for different scenarios to detect ill-posed problems and indentify the unobservable variables for a given set of measurements.

Chapter 6

Conclusion

In this research a reformulated heat exchanger networks model that is linear and rigorous is developed. This has enabled development of a non iterative, rigorous HEN simulation algorithm. The simulation algorithm for a heat exchanger network comprises of multi-phase solutions of sets of linear equations. Since solution of linear equations is very quick, this new algorithm has enabled use of evolutionary optimization algorithms to synthesize heat exchanger networks by using modular sizes of the exchangers. A methodology for HEN synthesis via differential evolution optimization proposed to design HENs, based on standard or modular sizes of heat exchangers, shows that optimal results can be obtained by using modular heat exchanger sizes, without having custom size exchangers as has been traditionally done in the literature. A procedure for data reconciliation and correction of heat exchanger heat transfer coefficients in HEN is proposed. The procedure uses heat exchangers network models developed in this work. Several case studies show that the proposed methodology for DR and parameter estimation produces results comparable to previous research with the advantage of using simpler models.

In the below sections the key findings and observations made in this research are presented.

1. **HEN Synthesis via Differential Evolution**

   In this work, a methodology for designing HEN based on modular sizes of heat exchangers is developed. This method is applied to several case studies from a small size problem (16 optimization variables) to a large size problem (1344

optimization variables) to understand the applicability and the size of the problem this method could handle. The impact of changing different optimizing parameters like crossover strategies, number of stages, crossover factor, and mutation factor on these case studies are analyzed. These cases studies resulted in the following key findings,

In case study 1, two different crossover strategies were applied, in one scenario uniform crossover strategy is applied in another scenario exponential crossover strategy is applied. In both the cases the final optimum solution in terms of network structure and cost is same but the number of generations it took to reach the optimum is more in the case when uniform crossover strategy is applied.

In case study 3, the problem was analyzed for understanding the effect of number of stages on the optimum solution. The author's suspect was that with the increase in number of stages chances of getting better solutions was more. So, two scenarios were analyzed in this case study, in one scenario the number of stages was fixed to 3 and in another case study the number of stages was fixed to 4. However, the final optimal network with 3 stages resulted in a better solution compared to the network from 4 stages.

Case study 4, was implemented to understand the robustness and the maximum scale of the problem that is approach could handle. In this case study equal ratio of number of small, medium and large size heat exchangers are considered. The key point noticed was more number of small size heat exchangers were present in the final network, to compensate with a minimum utility requirement. Though the final cost of the network in this approach is more in comparison to the earlier results, the final network in this approach would be advantageous in long run as the operating cost is less in comparison to earlier results.

2. **Steady State Data Reconciliation**

A framework for steady state data reconciliation using the HEN simulation algorithm combined with QR factorization is proposed. The key value that is work has added to the previous work on data reconciliation using QR factorization is, the constraints in this approach are linear, while the constraints in the previous work are nonlinear. In addition, in the current approach the data reconciliation is solved in two steps in first step reconciliation of mass flow rates is implemented and in the second step reconciliation of temperatures. Both the approaches of data reconciliation proved to be functioning well for the case study analyzed in this research.

In another case study data reconciliation combined with observability analysis is implemented to correct the heat transfer coefficients in plant-model mismatch scenario and several case studies are implemented to show the applicability of the current HEN simulation algorithm for different scenarios to detect ill-posed problems and indentify the unobservable variables for a given set of measurements. In this study of data reconciliation, the author's view was to show the applicability of the proposed HEN simulation algorithm for various scenarios and this study demonstrates that the HEN simulation algorithm can be extended further for different case studies and applications.

6.1 Recommendations for Further Work

1. **HEN Simulation Algorithm:**

The behavior of Φ_{flow} factor with respect to change in mass flow rate of hot and cold streams is almost a linear function. If the minimum and maximum mass flow rate of hot and streams are fixed for a heat exchanger, then the nonlinear Φ_{flow} factor can be linearized between these limits to have a complete linear model for

HEN simulation. As a future extension, linear models of Φ_{flow} factor could be modeled and these models could be tested for different types of problems.

In the HEN simulation algorithm, the computational time required to compute the nonlinear functional calculation phase, namely the after mass balance phase is high as compared to the remaining phases. Further improvements can be made from the perspective of programming side to decrease the computational time for calculation of Φ_{flow} factor mainly the search time for identifying the value of a corresponding variable in the memory.

The HEN simulation algorithm can be extended to optimize heat-integrated crude distillation units to check the applicability and robustness of the method.

2. HEN Synthesis via DE

A strategy for selecting modular sizes of heat exchangers for a HEN synthesis is to be developed, which leads to reduction in random choices of heat exchangers and thus decrease in the computational time and the number of heat exchangers required for optimizing HEN synthesis problem.

The key point to be addressed in HEN synthesis methodology is to improve DE in terms of computational time. In the current study DE was implemented on a single processor in sequential manner. In order to reduce the computational time drastically, the HEN synthesis via DE can be parallelized and can be run on multiprocessors in a parallel programming environment.

In the current study, several case studies were implemented by changing different optimizing parameters like crossover factor, mutation factor, number of stages, crossover strategies and number of modular sizes of heat exchangers. However, in this work the exact impact and a proper strategy for deciding these optimizing parameters factors is not presented. In future work a strategy for deciding these

factors in an integrated perspective (simultaneous change in different optimizing parameters) can be considered.

3. Steady State Data Reconciliation

In this research, the application of the proposed data reconciliation approach is demonstrated on a case study from the literature. However, in this study many areas of data reconciliation are not addressed by the author due to time constraint. The current approach could be further extended to check for gross-error detections, measurements testing criteria, observability analysis.

In the current study, from the results obtained in case study 2, there are chances of further improvements in understanding the observability analysis of the system and measurements selection criteria.

After extensive experimental trials on the current data reconciliation framework, this approach could be extended to implement data reconciliation on heat integrated crude distillation units

List of References

1. B. Linnhoff and E. Hindmarsh. The pinch design method for heat exchanger networks. *Chemical Engineering Science,* 38(5): 745 - 763, 1983.

2. B. Lin and D.C. Miller. Solving Heat Exchanger Network Synthesis Problems with Tabu Search. *Computers and Chemical Engineering*, 28, 2287-2306, 2004.

3. B.V. Babu. Process plant simulation. Oxford university press. 2004

4. C.A. Floudas and A.R. Ciric, Strategies for overcoming uncertainties in heat exchanger network synthesis. *Computers and Chemical Engineering,* 13(10), 1133 – 1152, 1989.

5. C.A. Floudas, A.R. Ciric and I.E. Grossmann, Automatic Synthesis of Optimum Heat Exchanger Network Configurations. *AIChE J.* 32(2), 276-290, 1986.

6. C.A. Floudas and A.R. Ciric, Corrigendum-strategies for overcoming uncertainties in heat exchanger network synthesis. *Computers and Chemical Engineering,* 14(8), I, 1990.

7. Chen, J.J.J. Comments on improvements on a replacement for the logarithmic mean. *Chemical Engineering Science*, 42(10): 2488 - 2489, 1987.

8. Chong, Z. Dynamic Reoptimization and Control Under Shutdown Conditions. Master's Thesis, McMaster University. 2006.

9. C.L.E Swartz. Data Reconciliation for Generalized Flow Sheet Applications, American Chemical Society National meeting, Dallas, Texas, 1989.

10. C.M. Crowe, Y.A.G. Campos and A. Hrymark. Reconciliation of Process Flow Rates by Matrix Projection. I: Linear Case. AIChE Journal, 29: 881-888. 1983

11. C.M. Crowe. Observability and Redundancy of Process Data for Steady State Reconciliation Chemical Engineering Science, 44(12): 2909-2917, 1989.

12. Chunliang Zhang. Nonlinear Data Reconciliation and Parameter Estimation for Real-Time Optimization. Master's Thesis, McMaster University. August 1998.

13. D. Corne, M. Dorigo and F. Glover. New ideas in optimization. McGraw-Hill, Berkshire, England, UK, 1999.

14. D.E. Goldberg. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley, 1989.

15. F. Pettersson. Heat exchanger network design using geometric mean temperature difference. *Computers and Chemical Engineering,* 32: 1726 – 1734, 2008.

16. G. Athier, P. Floquet, L.Pibouleau and S.Domenech. Optimization of heat exchanger networks by coupled simulated annealing and NLP procedures. *Computers and Chemical Engineering Supplement,* 20: S13 – S18, 1996.

17. Hongmei Yu, Haipeng Fang, Pingjing Yao and Yi Yuan. A combined genetic algorithm / simulated annealing algorithm for large scale system energy integration. *Computers and Chemical Engineering,* 24: 2023 – 2035, 2000.

18. J.H. Holland. *Adaptation in natural and artificial systems.* Ann Arbor, Michigan: The University of Michingan Press, 1975.

19. J.M. Zamora and I.E. Grossmann. A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits. *Computers and Chemical Engineering,* 22(3): 367 – 384, 1998.

20. Kenneth Kundert. Sparse 1.4, an Open Source Sparse Linear Equation Solver (http://sparse.sourceforge.net/index.html), June 2003.

21. Ken Price, Rainer Storn and Jouni Lampinen, Differential evolution - a practical approach to global optimization. Springer, June 2005

22. Kevin C.Furman and N.V. Sahinidis. A critical review and annotated bibliography for heat exchanger network synthesis in the 20th century. *Industrial and Engineering Chemistry Research,* 41 (10): 2335 – 2370, 2002.

23. Krishna M. Yerramsetty and C.V.S. Murty. Synthesis of cost-optimal heat exchanger networks using differential evolution. *Computers and Chemical Engineering,* 32:, 1861 – 1876, 2008.

24. Lester Godwin. Differential evolution (DE) (http://www.icsi.berkeley.edu/~storn/code.html#c++c), 1998.

25. Luiz O. de Oliveira Filho, Eduardo M.Queiroz and André L.H. Costa. A matrix approach for steady-state simulation of heat exchanger networks. *Applied Thermal Engineering,* 27: 2385 – 2393, 2007.

26. Mah, R.S.H. Chemical Process Structures and Information Flows. Boston: Butterworths, 1990.

27. Mahalec. V. Private Communication, 2008

28. Mahalec. V. Process Modeling and Optimization, Real Time Optimization. Class notes, McMaster University, 2008

29. Massimiliano Errico, Sara Maccioni, Giuseppe Tola and Paola Zuddas. A deterministic algorithm for the synthesis of maximum energy recovery heat exchanger network. *Computers and Chemical Engineering,* 31: 773-781, 2007

30. M.A.S.S. Ravagnani, A.P. Silva, P.A. Arroyo and A.A. Constantino. Heat exchanger network synthesis and optimization using genetic algorithm. *Applied Thermal Engineering*, 25: 1003 – 1017, 2004

31. Nobel, B., and J.W.Daniel. *Applied Linear Algebra. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1977.*

32. Om. P. Goyal. Speedy analysis of heat-exchange processes by elimination of trial and error. *Chemical Age of India.* 26(6):, 465 – 470, June 1975.

33. Rakesh Angira and B.V. Babu. Optimization of process synthesis and design problems: A modified differential evolution appraoch. *Chemical Engineering Science,* 61: 4707 – 4721, 2006.

34. R.Ratnam and V.S. Patwardhan. Sensitivity analysis for heat exchanger networks. *Chemical Engineering Science,* 46(2): 451-458, 1991.

35. R. Siegel and J.R. Howell. Thermal radiation heat transfer (2nd ed.). Washington, DC: Hemisphere Publication Corporation.

36. Shankar Narasimhan, Cornelius Jordache. Data Reconciliation & Gross Error Detection". Gulf publishing company, 2000.

37. S.Frausto-Hernández, V. Rico-Ramírez and A. Jiménez-Gutiérrez. MINLP synthesis of heat exchanger networks considering pressure drop effects. *Computers and Chemical Engineering*, 27: 1143-1152, 2003.

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

39. S. Kirkpartrick, C.D. Gelatt and M.P. Vechhi. Optimization by simulated annealing. *Science,* 220(4598): 671- 680, 1983.

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

41. Vitaliy Feoktistov. *Differential evolution. In search of solutions.* Springer, Volume 5, 2005.

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

43. X. Yunan, Pibouleau and S. Domenech. Experiments in process synthesis via mixed-integer programming. *Chemical Engineering Process,* 25(2), 99-116, 1989.

Appendix

A.1. Derivation of Φ factor for heat exchanger model

The mathematical expression for Φ factor is derived for no change in stream phase, counter-current flow configuration, no change in average specific heat of the process streams.

The three equations of heat transfer for heat exchange equipment are

$$q = m_h.Cp_h(t_{h1} - t_{h2}) \tag{A.1}$$

$$q = m_c.Cp_c(t_{c2} - t_{c1}) \tag{A.2}$$

$$q = U.A.\frac{(\Delta t_1 - \Delta t_2)}{\ln\left(\frac{\Delta t_1}{\Delta t_2}\right)} \tag{A.3}$$

Substituting in equation (A.3), $\Delta t_1 = (\Delta t_\alpha - \Delta t_c)$, $\Delta t_2 = (\Delta t_\alpha - \Delta t_h)$

$$q = U.A.\frac{(\Delta t_h - \Delta t_c)}{\ln\left(\frac{\Delta t_\alpha - \Delta t_c}{\Delta t_\alpha - \Delta t_h}\right)} \tag{A.4}$$

Where,

$\Delta t_h = (\Delta t_{h1} - \Delta t_{h2})$
$\Delta t_c = (\Delta t_{c2} - \Delta t_{c1})$
$\Delta t_\alpha = (\Delta t_{h1} - \Delta t_{c1})$

Introducing new set of conditions in the system, the transformed variables are represented as follows,

$m_h \longrightarrow m_h^{'}$

$m_c \longrightarrow m_c^{'}$

$t_{h1} \longrightarrow t_{h1}^{'}$

$t_{c1} \longrightarrow t_{c1}^{'}$

$U \longrightarrow U^{'}$

$A \longrightarrow A^{'}$

$t_{h2} \longrightarrow t_{h2}^{'}$

$t_{c2} \longrightarrow t_{c2}^{'}$

$q \longrightarrow q^{'}$

Therefore with the set of new conditions the new equilibrium conditions of the system are given as

$$q^{'} = m_h^{'}.Cp_h(t_{h1}^{'} - t_{h2}^{'}) \tag{A.5}$$

$$q^{'} = m_c^{'}.Cp_c(t_{c2}^{'} - t_{c1}^{'}) \tag{A.6}$$

$$q^{'} = U^{'}.A^{'}.\frac{(\Delta t^{'}{}_h - \Delta t^{'}{}_c)}{\ln\left(\frac{\Delta t^{'}{}_\alpha - \Delta t^{'}{}_c}{\Delta t^{'}{}_\alpha - \Delta t^{'}{}_h}\right)} \tag{A.7}$$

Where,

$\Delta t^{'}{}_\alpha = (\Delta t^{'}{}_{h1} - \Delta t^{'}{}_{c1})$

$\Delta t^{'}{}_h = (\Delta t^{'}{}_{h1} - \Delta t^{'}{}_{h2})$

$\Delta t^{'}{}_c = (\Delta t^{'}{}_{c2} - \Delta t^{'}{}_{c1})$

Now, introducing an outward parameter Φ, it is defined later in the section.

$$q^{'} = \Phi.q \tag{A.8}$$

Form equation (A.1) and (A.5) we get,

$$m_h'.(t_{h1}' - t_{h2}') = \Phi.m_h.(t_{h1} - t_{h2}) \tag{A.9}$$

$$\Delta t'_h = \Phi.\left(\frac{m_h}{m'_h}\right).\Delta t_h \tag{A.10}$$

Similarly form equation (A.2) and (A.6) we get,

$$m_c'.(t_{c2}' - t_{c1}') = \Phi.m_c.(t_{c2} - t_{c1}) \tag{A.11}$$

$$\Delta t'_c = \Phi.\left(\frac{m_c}{m'_c}\right).\Delta t_c \tag{A.12}$$

Now, eliminating q & q' from equations (A.3) and (A.7) using equation (A.8) we get,

$$\frac{(U').(A').[\Delta t'_h - \Delta' t_c]}{\ln\left[\frac{\Delta t'_\alpha - \Delta t'_c}{\Delta t'_\alpha - \Delta t'_h}\right]} = \frac{\Phi.U.A.(\Delta t_h - \Delta t_c)}{\ln\left[\left(\frac{\Delta t_\alpha - \Delta t_c}{\Delta t_\alpha - \Delta t_h}\right)\right]} \tag{A.13}$$

Substituting value of $\Delta t'_h \& \Delta t'_c$ from equations (A.10) and (A.12) into (A.13) we get, the following form,

$$\frac{\left(\frac{U'}{U}\right)\left(\frac{A'}{A}\right)\left[\left(\frac{m_h}{m_h'}\right)\Delta t_h - \left(\frac{m_c}{m_c'}\right)\Delta t_c\right]}{\ln\left[\frac{\Delta t_\alpha' - \Phi.\left(\frac{m_c}{m_c'}\right)\Delta t_c}{\Delta t_\alpha' - \Phi.\left(\frac{m_h}{m_h'}\right)\Delta t_h}\right]} = \frac{(\Delta t_h - \Delta t_c)}{\ln\left(\frac{\Delta t_\alpha - \Delta t_c}{\Delta t_\alpha - \Delta t_h}\right)} \tag{A.14}$$

Re-arranging the terms in the equation in A.14, we get the following equation.

$$\frac{\Delta t'_\alpha - \Phi.\left(\frac{m_c}{m'_c}\right).\Delta t_c}{\Delta t'_\alpha - \Phi.\left(\frac{m_h}{m'_h}\right).\Delta t_h} = \left[\left(\frac{\Delta t_\alpha - \Delta t_c}{\Delta t_\alpha - \Delta t_h}\right)^X\right] = \left(\frac{\Delta t_1}{\Delta t_2}\right)^X \tag{A.15}$$

Where,

$$X = \frac{\left(\frac{U'}{U}\right)\left(\frac{A'}{A}\right)\left[\left(\frac{m_h}{m_h'}\right)\Delta t_h - \left(\frac{m_c}{m_c'}\right)\Delta t_c\right]}{(\Delta t_h - \Delta t_c)} \tag{A.16}$$

Now solving equation (A.13) for Φ, we get the following equation,

$$\Phi = \frac{\left(\frac{\Delta t_1}{\Delta t_2}\right)^X \Delta t'_\alpha - \Delta t'_\alpha}{\left(\frac{\Delta t_1}{\Delta t_2}\right)^X \cdot \left(\frac{m_h}{m'_h}\right).\Delta t_h - \left(\frac{m_c}{m'_c}\right).\Delta t_c} \tag{A.17}$$

Now, the Φ factor and the parameter X estimated in the above, equations (A.16) and (A.17) can be used to calculate the new changes in the system by using the base conditions of the process and the current conditions of the process.