NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for naturvitenskap og teknologi
Institutt for kjemisk prosessteknologi

SPECIALIZATION PROJECT 2012

TKP 45

PROJECT TITLE:

Optimal Operation of Parallel Systems

By

Stian Aaltvedt

Supervisor for the project: Sigurd Skogestad & Johannes Jäschke

Date: 2012-12-07

Norwegian University of Science and Technology
Department of Chemical Engineering

Specialization project fall 2012

Optimal Operation of Parallel Systems

Author:
Stian Aaltvedt

Supervisors:
Prof. Sigurd Skogestad
Post.doc Johannes Jäschke

December 7, 2012

Summary

The aim of this report has been to investigate the self optimizing variables for control of steady state parallel heat exchange systems derived by Jäschke (Jaeschke 2012). The objective of the operation is to maximize the final outlet temperature subject to minimization of heat exchanger operating costs.

For three cases of different heat exchanger networks optimal operation by use of the Jäschke Temperature control variable was tested against the results from optimal design.

The optimal operation resulted in a difference in outlet temperature of less than 0.1 % from the optimal design, and thereby confirmed that the Jäschke Temperature is a good control variable for the three steady state cases studied in this report.

The heat exchanger operating costs' impacts on optimal design of the three cases was also investigated. Among expected trends of decreasing outlet temperature and heat exchanger size, the results showed that it is more economically favorable to exploit the capacity of hot heat exchangers and rather exclude colder heat exchangers to form a bypass.

All simulations were done using MATLAB and the build-in function `fmincon`.

Contents

List of Figures

List of Tables

1 Introduction

In a modern industrial and technological world where energy serves as one of the most essential parameters, there are greater and greater requirements for all production processes to be sustainable to future generations of our planet. In the chemical industry, especially including today's great oil and gas production, an overall goal of using the available energy sources in the most efficient way can be satisfied by optimal heat recovery from different parts of the given process. In terms of process control, this can be implemented using self-optimizing control. The idea of self-optimizing control is to achieve near-optimal control by keeping certain variables or variable combinations constant. Jäschke (Jaeschke 2012) has proposed a self-optimizing control variable for parallel heat exchange systems. The basic idea is to achieve good and tight control by keeping one certain constraint active, the Jäschke Temperature. Applied on different heat exchanger networks with parallel branches, the overall goal is to gain as high downstream temperature as possible, that is, to maximize the outlet temperature from the heat exchanger network.

The issue of saving energy strongly relates to the issue of saving money. For any industrial process the supervisory control tears down to the simple, but from time to time so complicated requirement of keeping the costs low. Great heat integration and heat exchanger utility costs money. With a trade off between outlet temperature and heat exchanger size, both requirements are met to whatever objective of any process.

The work done in this report takes on to these issues. First it is investigated whether the Jäschke Temperature controlled variable meet the criteria of keeping different parallel systems close to optimum. This is done by comparing the outlet temperature from optimal design to the optimal operation given by the Jäschke Temperature controlled variable. Secondly, a comprehensive analysis on the impacts of increased heat exchanger costs on the optimal design was done.

2 Principles of Heat Transfer

With heat exchange the overall goal is to transfer heat from a hot source to a cold source (Skogestad 2003). The heat transfer process can be carried out by three different mechanisms (Geankoplis 2003):

- Conduction heat transfer
- Convection heat transfer
- Radiation heat transfer

For most industrial processes where heat is transfered from one fluid to another through a solid wall, conduction is the main mechanism for heat transfer. This heat transfer is conducted in a heat *exchanger*, where a stream of hot fluid meets a stream of cold fluid, which is to be heated by the hot fluid. The most effective way of heat transfer is done through a *counter current* heat exchanger, shown in Figure 2.1. Here, Q $[kW]$ represents the transfered heat and T_h and T_c [°C] are the hot and cold streams, respectively.

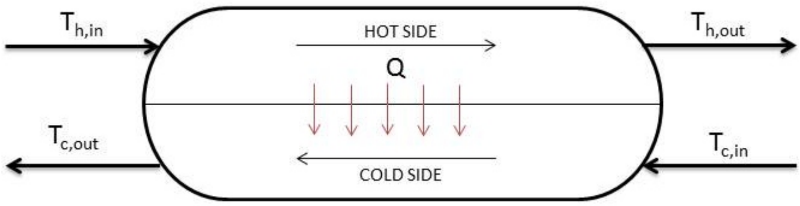

Figure 2.1: The counter current heat exchanger

2.1 Model and Energy Equations

From Figure 2.1 and for the ideal case with constant inlet temperatures ($T_{h,in}$ and $T_{c,in}$), the heat Q transfered form hot to cold side can be expressed by the heat exchanger model (Skogestad 2003):

$$Q = UA\Delta T_{LM} \tag{2.1}$$

Where U is the over all heat coefficient $[kW/°\text{C}m^2]$ and A is the total area of the heat exchanger $[m^2]$. The overall heat coefficient U is the reciprocal of the overall resistance to heat transfer, which is the sum of several individual resistances, given by Sinnot & Towler (Sinnott & Towler 2009)

$$\frac{1}{U} = \frac{1}{h} + \frac{1}{h_d} + \frac{d\ln(\frac{d}{d_i})}{2k_w} + \frac{d}{d_i}\frac{1}{h_i d} + \frac{d}{di}\frac{1}{h_i} \tag{2.2}$$

where
h = outside heat transfer coefficient $[kW/°\text{C}m^2]$
h_i = inside heat transfer coefficient $[kW/°\text{C}m^2]$
h_d = outside dirt coefficient (fouling factor) $[\text{k}W/°\text{C}m^2]$

$h_i d$ = inside dirt coefficient (fouling factor) $[kW/^{\circ}\mathrm{C}m^2]$
k_w = Thermal conductivity of the tube wall material $[kW/^{\circ}\mathrm{C}m]$
d_i = Tube inside diameter $[m]$
d = Tube outside diameter $[m]$

However, for most ideal cases the fouling factors and tube wall thermal conductivity is neglected. The overall heat coefficient can thereby be written as (Incorpera & DeWitt 2007):

$$U = \frac{h_c h_h}{h_c + h_h} \tag{2.3}$$

Here, h_c and h_h represents the heat transfer coefficients for cold and hot side, respectively.

The ΔT_{LM} therm is the logarithmic mean temperature difference (LMTD). For a counter current heat exchanger it is given by (Skogestad 2003)

$$\Delta T_{LM} = \frac{(T_{h,in} - T_{c,out}) - (T_{h,out} - T_{c,in})}{\ln(\frac{T_{h,in} - T_{c,out}}{T_{h,out} - T_{c,in}})} = \frac{\theta_1 - \theta_2}{\ln(\frac{\theta_1}{\theta_2})} \tag{2.4}$$

where θ_1 is the temperature difference on the hot side, and θ_2 the difference on the cold side for ideal counter current streams.

The energy balance for the ideal counter current heat exchanger in Figure 2.1 is (Skogestad 2003)

$$Q = m_c Cp_c(T_{c,out} - T_{c,in}) \tag{2.5}$$

$$Q = m_h Cp_h(T_{h,in} - T_{h,out}) \tag{2.6}$$

Cp_c, Cp_h and m_c, m_h represents the heat capacities $[kW/kgK]$ and the mass flows $[kg/sec]$ for the cold and hot fluid, respectively. The heat capacities are assumed to be constant. From now on, the product $Cp_c m_c$ and $Cp_h m_h$ will be written as w_c and w_h, respectively.

From the principle of energy- and mass conservation the correlation between Equation 2.1, 2.5 and 2.6 are

$$Q = UA\Delta T_{LM} = w_c(T_{c,out} - T_{c,in}) = w_h(T_{h,in} - T_{h,out}) \tag{2.7}$$

2.2 Approximations

Skogestad (Skogestad 2003) states that the logarithmic mean temperature difference (LMTD) (Equation 2.4) can be approximated to an arithmetic mean temperature difference (AMTD). If $\frac{1}{1.4} < \frac{\theta_1}{\theta_2}$<1.4, i.e the temperature difference between the cold and hot side are fairly constant, the error of using AMTD instead of LMTD is less than 1%.

The arithmetic mean temperature difference, AMTD is given as

$$\Delta T_{AM} = \frac{\Theta_1 + \Theta_2}{2} \tag{2.8}$$

Underwood (Underwood 1933) approximates the LMTD with the Underwood approximation:

$$\Delta T_{UN} = \left(\frac{\Theta_1^{\frac{1}{3}} + \Theta_2^{\frac{1}{3}}}{2} \right)^3 \tag{2.9}$$

In addition, this study is carried out assuming steady state. The following assumptions will then also be used:

- Constant heat capacities, that is w_0 for the cold stream and w_{h_i} for the hot stream in heat exchanger i are assumed to be constant
- Constant inlet temperature T_0
- Constant hot stream temperatures T_{hi} in heat exchanger i
- Single-phase flow, no phase transfers during heating transfer

3 Optimal Operation of Heat Exchanger Networks

For any process, the overall goal is to maximize the future income of the plant (Jensen & Skogestad 2008). For all heat exchanger networks, the usefulness comes down to the heat integration and energy recovery as well as the costs of the heat exchangers. Obtaining the highest possible outlet temperature while at the same time operating with reasonable heat exchangers duties are the two contradictory factors for cost-effective heat transfer. Therefore, the objective function J for systems like this includes the total heat exchanger duty and the outlet temperature.

3.1 Self-optimizing Control

Self-optimizing control is when near-optimal operation is achieved with constant setpoints for the controlled variables. Self-optimized control has the advantage that it don't need re-optimization when disturbances are present.
From Skogestad (Skogestad 2004) the goal of an optimization problem is to minimize an objective function subject to its given constraints:

$$\text{minimize } J(x, u_t, d) \tag{3.1}$$

$$\begin{aligned} \text{subject to equality constraints: } & g(x, u_t, d) = 0 \\ \text{subject to inequality constrains: } & h(x, u_t, d) \geq 0 \end{aligned} \tag{3.2}$$

where J is the objective function, x the state variables, u_t is the manipulated variables and d the disturbances. The manipulated variables also denotes the systems degrees of freedom (DOFs). The equality constraints g includes the model equations, whereas the *inequality* constrains covers the physics of the system.
The task is to decide what to control with the degrees of freedom, u. If the states x are eliminated by use of the model equations g the remaining unconstrained problem is

$$min_u \; J(u, d) = J(u_{opt}, d) = J_{opt}(d) \tag{3.3}$$

Here, u_{opt} is to be found and $J_{opt}(d)$ is the optimal value of the objective function J.

The aim for self-optimizing control is to find a subset of the measured variables named c to keep constant at the optimal values c_{opt}. The ideal case would give a disturbance-insensitive c_{opt} to obtain optimal operation. However, in the real world there is a loss associated with keeping the controlled variable constant. Therefore, the goal is an operation as *close to* optimum as possible. The loss can be expressed as:

$$L(u, d) = J(u, d) - J_{opt}(d) \tag{3.4}$$

Skogestad (Skogestad 2000) presents the following guidelines for selecting controlled variables:

- c_{opt} should be insensitive to disturbances
- c should be easy to measure and control accurately
- c should be sensitive to change in the manipulated variables (degrees of freedom)

- For cases with more than one unconstrained degree of freedom, the selected controlled variables should be independent

Proposed by Halvorsen and Skogestad (Halvorsen & Skogestad 1997), an ideal self-optimizing variable is the gradient of the objective function J:

$$c_{ideal} = \frac{\partial J}{\partial u} \tag{3.5}$$

To ensure optimal operation for all disturbances, this gradient should be zero, but measurements of the gradient is usually not available. Therefore, computing it requires values of the unmeasured disturbances. To find the best suitable variables for approximations of the gradient, several mehtods can be used, including:

- Exact local method
- Direct evaluation of loss for all disturbances ("brute force")
- Maximum (scaled) gain method
- The null space methid

3.2 Self-optimizing Control Applied to Heat Exchanger Networks

Optimization problems on heat exchange networks are subject to a number of equality constraints. These are the heat exchange energy balances presented in Equation 2.7, Section 2.1, in addition to the total mass and energy balances of the network.
For optimal operation the Jäschke Temperature (Jaeschke 2012) also serves as an equality constraint. More detailed equality constraints for each case are given in Section 4 on case studies.

Inequality constraints includes the ΔT_{min}. For the case of heat exchangers, a common and simple approach to this challenge is to specify the minimum temperature approach (ΔT_{min}) in each heat exchanger. The ΔT_{min} constraint and its relationship with the hot and cold streams are given in Equation 3.6 and 3.7 below. A small value of ΔT_{min} means that a lot of energy is recovered, but it requires a large heat exchanger (Jensen & Skogestad 2008). If this inequality constraint is fully satisfied on both sides of the heat exchanger, the heat exchanger is said to be maximized under its constraints. From Equation 2.1, as the ΔT decreases against the constraint ΔT_{min}, the UA value of the heat exchanger needs to increase. In practice, this means that a perfect heat exchanger has an infinite area A. The definitions of ΔT_{min} is:

$$T_{h,in} - T_{c,out} = \Delta T_{min,hot} \tag{3.6}$$

$$T_{hi,out} - T_{c,in} = \Delta T_{min,cold} \tag{3.7}$$

From Figure 2.1 the hot and cold side of a heat exchanger are better illustrated, here in Figure 3.1.

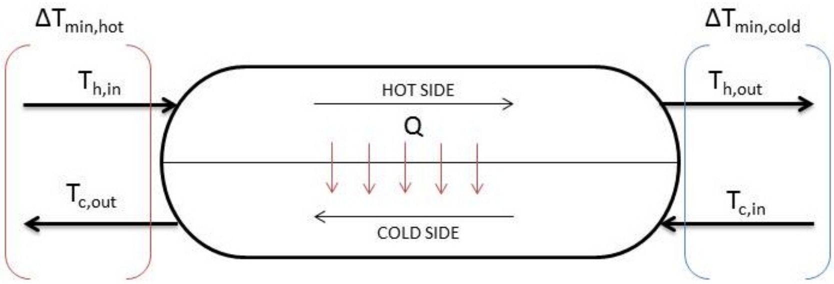

Figure 3.1: Hot and cold sides of a heat exchanger

To prevent from simulation errors due to infinite heat exchanger area or temperature cross, a ΔT_{min} of 0.5 was used for all cases studied in this report. The general inequality constraint matrix can then be written

$$h = \begin{pmatrix} T_{hi,in} - T_i - \Delta T_{min,hot,i} \\ T_{hi,out} - T_{i-1} - \Delta T_{min,cold,i} \\ \vdots \\ T_{hn,in} - T_n - \Delta T_{min,hot,n} \\ T_{hn,out} - T_{n-1} - \Delta T_{min,cold,n} \end{pmatrix} \geq 0 \tag{3.8}$$

Where the heat exchangers are numbered from i to n, and T_{i-1} and T_i are the cold streams entering and leaving heat exchanger i, respectively.

The main idea is that this specification should give a reasonable balance between minimizing operation costs (which is favored by a small ΔT_{min}) and minimizing the capital costs (which is favored by a large ΔT_{min}).
However, this method does not work as a complete optimization approach. A major part of start up- and maintenance costs are determined by *the size* of the heat exchanger. The minimum temperature approach does not take into account the area of each heat exchanger. A small value of ΔT_{min} means that a lot of energy is recovered, and is therefore favorable considering the energy demand, but this will also require a very large heat exchanger.

Jensen and Skogestad (Jensen & Skogestad 2008) states that the total annualized costs are divided into operation costs and capital costs. The challenge of cost estimations at an early design stage is that detailed equipment- and cost data are not available.

The detailed optimal design based on minimizing total annualized costs are (Jensen & Skogestad 2008):

$$min_u(J_{operation} + J_{capital}) \tag{3.9}$$

Where u is the degrees of freedom and includes all the equipment data and operating variables.
For a simplified case with heat exchangers the capital costs $J_{capital}$ can be expressed as a function of each heat exchangers area:

$$J_{capital} = c_0 \sum_i A_i^n \tag{3.10}$$

Here, c_0 and n is cost factors. A_i is the area of the heat exchanger i.

c_0 is defined as cost of heat exchanger duty relative to the costs of supplied heat. For example, if $c_0 = 2$, it means that the sizing costs of the heat exchanger is 2 times as expensive as the costs of heat supply, i.e. the costs of increasing the outlet temperature.

The operation variables has the general cost term and consists of several factors:

$$J_{operation} = \sum_i p_{F\,i} F_i - \sum_j p_{P\,j} P_j + \sum_k p_{Q\,k} Q_k + \sum_{s,l} p_{W\,s,l} W_{s,l} \tag{3.11}$$

where F_i are feeds, P_j are products, Q_k are utilities (energy), $W_{s,l}$ are the mechanical work and the p's are the respective prices. However, for a heat exchanger problem Equation 3.11 can be simplified to only include the supplied heat, that is

$$J_{opeation} = \sum_i p_{Q\,i} Q_i \tag{3.12}$$

which states that the operating costs are mostly determined by the heat supply in each heat exchanger. The transfered heat resembles the outlet temperature of the heat exchanger network, and for the optimization problem Equation 3.12 can be written in terms of the final outlet temperature T_{end}:

$$J_{operation} = -T_{end} \tag{3.13}$$

This states that the aim of the operation is to maximize T_{end}, and thereby minimize the negative value $-T_{end}$ in equation 3.12. The objective function (the total cost function) to be minimized then becomes

$$J = J_{operation} + J_{capital} = -T_{end} + C_0 \sum_i A_i^n \tag{3.14}$$

4 Case Studies

The aim of this analysis has been to investigate how different cost factors (c_0) and inlet stream heat capacity (w_0) affect the steady state self-optimizing control. The resulting optimized design will be used for further investigation and applied to obtain the optimal operation of the system. This was done by use of the Jäschke Temperatures from Jäschkes work on parallel heat exchanger systems (Jaeschke 2012). Jäschkes work was subjected to patent application as this report was carried out. The idea of Jäschke is to implement a much more easy controlled variable. With the Jäschke Temperature the stream split w_1 (or w_2) serves as the systems only manipulated variable, which will determine the optimal operation of the system.
For each case optimal operation based on the optimal design were determined for four different scenarios with different cost factors and inlet heat capacities (w_0).

In addition, a series of investigations were done on the cost factors impacts on optimal design. The resulting optimizes process variables were plotted against the cost function.

4.1 Case 1: Two heat exchangers in parallel

The following Figure 4.1 and Table 4.1 shows the network of two heat exchangers in parallel and the respective initial parameters.

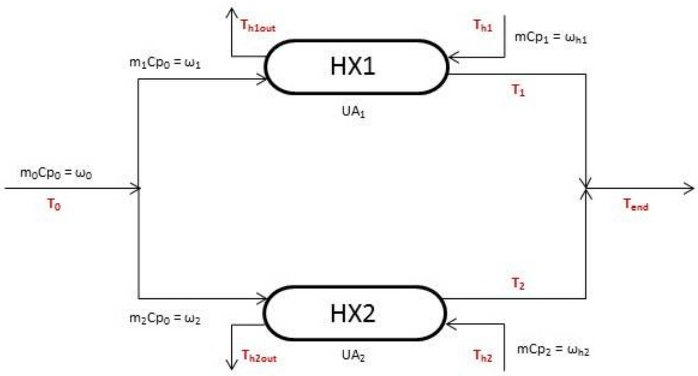

Figure 4.1: Case 1: Two heat exchangers in parallel

Table 4.1: Initial parameters Case 1 (two heat exchangers in parallel)

Inlet cold stream temperature, T_0	130 °C
Heat capacity cold stream, w_0	95 $kW/$°C
Hot stream temperature HX1, T_{h1}	203 °C
Heat capacity hot stream HX1, w_{h1}	60 $kW/$°C
Hot stream temperature HX2, T_{h2}	248 °C
Heat capacity hot stream HX2, w_{h2}	65 $kW/$°C

For this case the Underwood approximation (Underwood 1933) given in Section 2.2 was used. Thus, the heat exchanger model given in Equation 2.1 is now:

$$Q = UA\Delta T_{UN} \tag{4.1}$$

The total mass balance of the system is

$$w_0 = w_1 + w_2 \tag{4.2}$$

From this the overall energy balance becomes:

$$w_0 T_{end} = w_1 T_1 + w_2 T_2 \tag{4.3}$$

Including the three model equations given in Equation 2.7 in Section 2 the equality constraints for optimal design of case 1 is:

$$g_1 = \begin{pmatrix} Q_1 - (w_1(T_1 - T_0)) \\ Q_1 + (w_{h_1}(Th_{1,out} - Th_1)) \\ Q_1 + (UA_1 \Delta T_{UN,1}) \\ \\ Q_2 - (w_2(T_2 - T_0)) \\ Q_2 + (w_{h_2}(Th_{2,out} - Th_2)) \\ Q_2 + (UA_2 \Delta T_{UN,2}) \\ \\ w_1 + w_2 - w_0 \\ w_1 T_1 + w_2 T_2 - w_0 T_{end} \end{pmatrix} = 0 \tag{4.4}$$

When writing all the temperatures with reference to the inlet temperature T_0:

$$\begin{aligned} \Delta T_0 &= 0 \\ \Delta T_1 &= T_1 - T_0 \\ \Delta T_2 &= T_2 - T_0 \\ \Delta Th_1 &= Th_1 - T_0 \\ \Delta Th_2 &= Th_2 - T_0 \end{aligned} \tag{4.5}$$

After some algebra we end up with the self-optimizing variable from Jäschkes work (Jaeschke 2012) for a two heat exchangers in parallel system:

$$c = \frac{\Delta T_1^2}{\Delta T_{h,1}} - \frac{\Delta T_2^2}{\Delta T_{h,2}} \tag{4.6}$$

Or, more simplified by abuse of the delta-notation:

$$c = \frac{T_1^2}{T_{h,1}} - \frac{T_2^2}{T_{h,2}} \tag{4.7}$$

From Jäschke (Jaeschke 2012) this constraint should be an equality constraint and take the value 0 for optimal operation. For determination of optimal operation this control variable is the last constraint to be included in g_1 as the last satisfied equality constraint.

4.2 Case 2: Two heat exchangers in series parallel with one heat exchanger

The following Figure 4.2 and Table 4.2 shows the network of two heat exchangers in series parallel with one heat exchanger and the respective initial parameters.

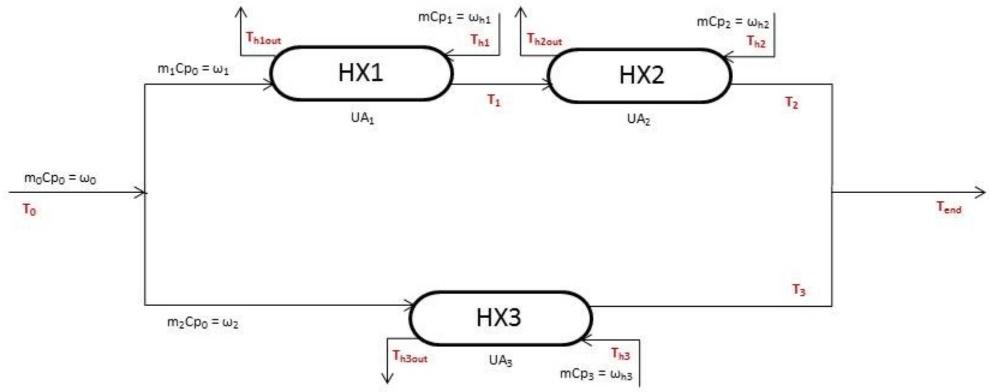

Figure 4.2: Cae 2: Two heat exchangers in series parallel with one heat exchanger

Table 4.2: Initial parameters Case 2 (one heat exchangers in parallel with two in series)

Inlet cold stream temperature, T_0	130 °C
Heat capacity cold stream, w_0	95 kW/°C
Hot stream temperature HX1 T_{h1}	203 °C
Heat capacity hot stream HX1, w_{h1}	60 kW/°C
Hot stream temperature HX2 T_{h2}	255 °C
Heat capacity hot stream HX2, w_{h2}	27 kW/°C
Hot stream temperature HX3 T_{h3}	248 °C
Heat capacity hot stream HX2, w_{h3}	65 kW/°C

Also for this case, the Underwood approximation was used (Underwood 1933). That means that Equation 4.1 and 4.2 applies for this case as well. The energy balance for this case is:

$$w_0 T_{end} = w_1 T_1 + w_2 T_3 \tag{4.8}$$

Again including the three model equations given in Equation 2.7 in Section 2 the equality constraints for optimal design of case 2 is:

$$g_2 = \begin{pmatrix} Q_1 - (w_1(T_1 - T_0)) \\ Q_1 + (w_{h_1}(Th_{1,out} - Th_1)) \\ Q_1 + (UA_1 \Delta T_{UN,1}) \\ \\ Q_2 - (w_1(T_2 - T_1)) \\ Q_2 + (w_{h_2}(Th_{2,out} - Th_2)) \\ Q_2 + (UA_2 \Delta T_{UN,2}) \\ \\ Q_3 - (w_2(T_3 - T_0)) \\ Q_3 + (w_{h_3}(Th_{3,out} - Th_3)) \\ Q_3 + (UA_3 \Delta T_{UN,3}) \\ \\ w_1 + w_2 - w_0 \\ w_1 T_1 + w_2 T_3 - w_0 T_{end} \end{pmatrix} = 0 \tag{4.9}$$

Again, by writing all the temperatures with reference to the inlet temperature T_0:

$$\begin{aligned} \Delta T_0 &= 0 \\ \Delta T_1 &= T_1 - T_0 \\ \Delta T_2 &= T_2 - T_0 \\ \Delta T_3 &= T_3 - T_0 \\ \Delta Th_1 &= Th_1 - T_0 \\ \Delta Th_2 &= Th_2 - T_0 \\ \Delta Th_3 &= Th_3 - T_0 \end{aligned} \tag{4.10}$$

Also here the delta-notation is omitted for the case of readability.

With some algebraic steps the self-optimized variable from Jäschkes work is (Jaeschke 2012):

$$c = \left(\frac{T_{h,2} - T_1}{T_{h,1}} - 1 \right) \frac{T_1^2}{T_{h,2} - T_2} + \frac{T_2^2}{T_{h,2} - T_1} - \frac{T_3^2}{T_{h,3}} \tag{4.11}$$

For optimal operation this is to be included in the equality constraint g_2 and take the value 0.

4.3 Case 3: Three heat exchanger in series parallel with two heat exchangers in series

The following Figure 4.3 and Table 4.3 shows the network of three heat exchanger in series parallel with two heat exchangers in series and the respective initial parameters.

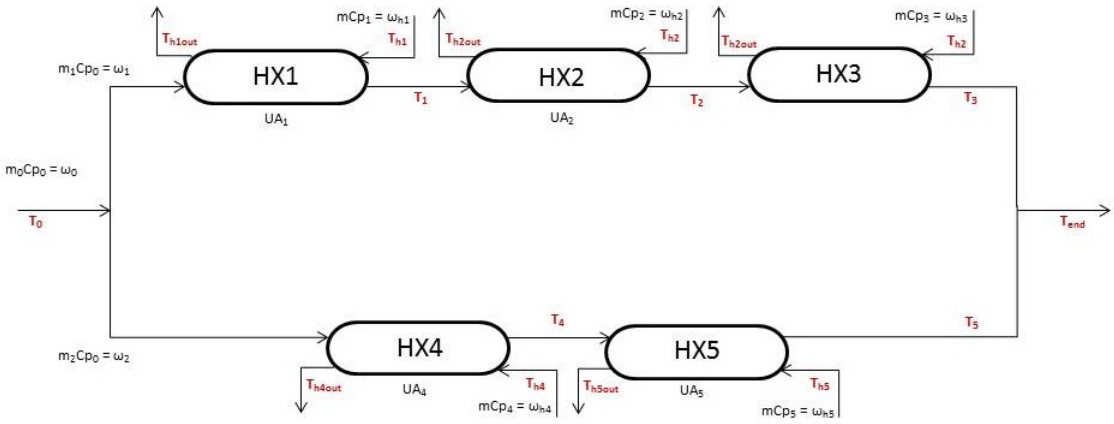

Figure 4.3: Case 3: Three heat exchanger in series parallel with two heat exchangers in series

Table 4.3: Initial parameters Case 3 (three heat exchanger in series parallel with two heat exchangers in series)

Inlet cold stream temperature, T_0	130 °C
Heat capacity cold stream, w_0	150 $kW/$°C
Hot stream temperature HX1, T_{h1}	190 °C
Heat capacity hot stream HX1, w_{h1}	50 $kW/$°C
Hot stream temperature HX2, T_{h2}	203 °C
Heat capacity hot stream HX2, w_{h2}	30 $kW/$°C
Hot stream temperature HX3, T_{h3}	220 °C
Heat capacity hot stream HX3, w_{h3}	15 $kW/$°C
Hot stream temperature HX4, T_{h4}	220 °C
Heat capacity hot stream HX4, w_{h4}	70 $kW/$°C
Hot stream temperature HX5, T_{h5}	248 °C
Heat capacity hot stream HX5, w_{h5}	20 $kW/$°C

The Underwood approximation was used (Underwood 1933). That means that Equation 4.1 and 4.2 also applies to this case. The topology gives the energy balance:

$$w_0 T_{end} = w_1 T_3 + w_2 T_5 \tag{4.12}$$

Including the three model equations given in Equation 2.7 in Section 2 the equality constraints for optimal design of case 3 is:

$$g_3 = \begin{pmatrix} Q_1 - (w_1(T_1 - T_0)) \\ Q_1 + (w_{h_1}(Th_{1,out} - Th_1)) \\ Q_1 + (UA_1 \Delta T_{UN,1}) \\ \\ Q_2 - (w_1(T_2 - T_1)) \\ Q_2 + (w_{h_2}(Th_{2,out} - Th_2)) \\ Q_2 + (UA_2 \Delta T_{UN,2}) \\ \\ Q_3 - (w_1(T_3 - T_2)) \\ Q_3 + (w_{h_3}(Th_{3,out} - Th_3)) \\ Q_3 + (UA_3 \Delta T_{UN,3}) \\ \\ Q_4 - (w_2(T_4 - T_0)) \\ Q_4 + (w_{h_4}(Th_{4,out} - Th_4)) \\ Q_4 + (UA_4 \Delta T_{UN,4}) \\ \\ Q_5 - (w_2(T_5 - T_4)) \\ Q_5 + (w_{h_5}(Th_{5,out} - Th_5)) \\ Q_5 + (UA_5 \Delta T_{UN,5}) \\ \\ w_1 + w_2 - w_0 \\ w_1 T_1 + w_2 T_5 - w_0 T_{end} \end{pmatrix} = 0 \tag{4.13}$$

By writing all the temperatures with reference to the inlet temperature T_0:

$$\begin{gathered} \Delta T_0 = 0 \\ \Delta T_1 = T_1 - T_0 \\ \Delta T_2 = T_2 - T_0 \\ \Delta T_3 = T_3 - T_0 \\ \Delta T_4 = T_4 - T_0 \\ \Delta T_5 = T_5 - T_0 \\ \Delta Th_1 = Th_1 - T_0 \\ \Delta Th_2 = Th_2 - T_0 \\ \Delta Th_3 = Th_3 - T_0 \\ \Delta Th_4 = Th_4 - T_0 \\ \Delta Th_5 = Th_5 - T_0 \end{gathered} \tag{4.14}$$

Also here the delta-notation is omitted for the case of readability.

The self-optimized variable from Jäschkes work for this case is (Jaeschke 2012):

$$c = \frac{T_1^2(T_2T_{h,3} - T_{h,1}T_{h,3} - T_{h,2}T_{h,3} + T_3T_{h,2} - T_{h,1}T_3) + T_2^2T_{h,1}(T_3 - T_1 - T_{h,3} + T_{h,2}) + T_3^2T_{h,1}(T_1 - T_{h,2})}{T_{h,1}(-T_{h,1}T_{h,3} - T_1T_2 + T_1T_{h,3} + T_{h,2}T_2)} - \left(\frac{T_{h,5} - T_4}{T_{h,4}} - 1\right)\frac{T_4^2}{T_{h,5} - T_4} + \frac{T_5^2}{T_{h,5} - T_5} \tag{4.15}$$

For optimal operation this is to be included in the equality constraint g_3 and take the value 0.

4.4 Test Descriptions

Each systems design was optimized and optimal operation was determined for 4 different case scenarios.

4.4.1 Case 1 Scenarios

The following 4 scenarios applies to case 1 with two heat exchangers in parallel:

1. Scenario: $w_0 = 130\ kW/°\text{C}$ and $c_0 = 2$
2. Scenario: $w_0 = 130\ kW/°\text{C}$ and $c_0 = 4$
3. Scenario: $w_0 = 95\ kW/°\text{C}$ and $c_0 = 2$
4. Scenario: $w_0 = 95\ kW/°\text{C}$ and $c_0 = 4$

In addition, a series of plots showing the impacts on optimal design and process variables were made for the case with $w_0 = 95$ and 130 $kW/°\text{C}$ and c_0 varying from 1 to 5.

4.4.2 Case 2 Scenarios

The following 4 scenarios applies to case 2 with two heat exchangers in series parallel with one heat exchanger:

1. Scenario: $w_0 = 160\ kW/°\text{C}$ and $c_0 = 2$
2. Scenario: $w_0 = 160\ kW/°\text{C}$ and $c_0 = 4$
3. Scenario: $w_0 = 95\ kW/°\text{C}$ and $c_0 = 2$
4. Scenario: $w_0 = 95\ kW/°\text{C}$ and $c_0 = 4$

A series of plots showing the impacts on optimal design and process variables were made for the case with $w_0 = 95$ and 160 $kW/°\text{C}$ and c_0 varying from 1 to 5.

4.4.3 Case 3 Scenarios

The following 4 scenarios applies to case 3 with three heat exchangers in series parallel with two heat exchangers in series:

1. Scenario: $w_0 = 180\ kW/°\text{C}$ and $c_0 = 2$
2. Scenario: $w_0 = 180\ kW/°\text{C}$ and $c_0 = 4$
3. Scenario: $w_0 = 150\ kW/°\text{C}$ and $c_0 = 2$
4. Scenario: $w_0 = 150\ kW/°\text{C}$ and $c_0 = 4$

A series of plots showing the impacts on optimal design and process variables were made for the case with $w_0 = 150$ and 180 $kW/°\text{C}$ and c_0 varying from 1 to 5.

For case 3 an additional investigation was done with optimal operation. By use of the Jäschke Temperature control variable with the optimized UA values from the four scenarios in case 3 the optimal operation was determined for a case with a different set of hot stream temperatures. The aim was to investigate the systems behavior with a hot exchanger followed by a cooler heat exchanger.
With the same heat capacities as originally given in Table 4.3, the new hot stream temperatures for this last investigation only was:

- $Th_1 = 205°C$
- $Th_2 = 203°C$
- $Th_3 = 220°C$
- $Th_4 = 220°C$
- $Th_5 = 248°C$

5 Results

5.1 Case 1: Two heat exchangers in parallel

A selection of the results from optimized design and the corresponding results from optimal operation obtained from the Jäschke Temperature constraint (Jaeschke 2012) are given in the following Tables 5.1 and 5.2. For the sake of the reports readability the optimized UA values used to obtain optimal operation are given in Appendix A.

Complete simulation results are given Appendix A.

Table 5.1: A selection of design and operating results for case study 1, scenario 1

	Optimal design	Optimal operation
T_{end} [°C]	199.08	199.04
w_1 [%]	30.3	28.3
w_2 [%]	69.6	71.7

Table 5.2: A selection of design and operating results for case study 1, scenario 4

	Optimal design	Optimal operation
T_{end} [°C]	205.97	205.68
w_1 [%]	24.8	22.1
w_2 [%]	75.2	77.9

From these results the Jäschke Temperature controlled variable gives almost the same outlet temperatures (T_{end}) as the optimized values. Only a slight difference is seen in the stream split (w_1 and w_2). The heat exchangers was observed to vary in size from $UA = 1$ - 10 $^{kWm^2}/_{°C}$ where heat exchanger 2 on the lower path was significant larger than the 1st heat exchanger on the upper path.
For complete results for all scenarios, including several other optimized and operation variables please see Appendix A.

For the cost factor c_0 spanning from 1 - 5 a series of plots were made showing the impacts on optimal design. Two of them are given here, displaying the outlet temperature T_{end} and the split (w_1 and w_2) as a function of c_0. These are given in Figure 5.1 and 5.2, respectively.
Several other plots are given in Appendix A.

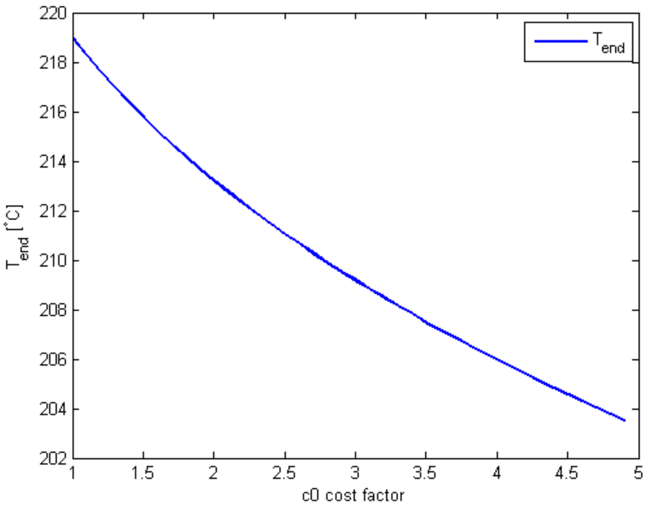

Figure 5.1: T_{end} as a function of cost factor c_0 for case 1 with inlet heat capacity $w_0 = 95\ kW/°C$

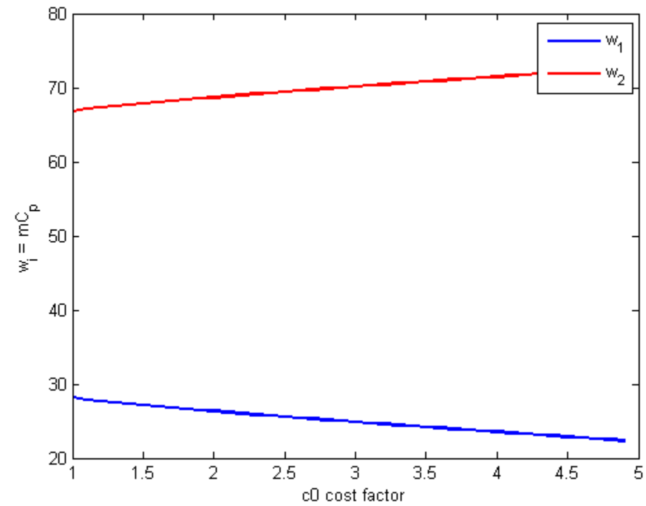

Figure 5.2: w_1 and w_2 [$kW/°C$] as a function of cost factor c_0 for case 1 with inlet heat capacity $w_0 = 95\ kW/°C$

The results shows good correlation with expected behavior. As the cost factor increase, i.e. the costs of operating the heat exchangers increase relative to the costs of increasing the outlet temperature, T_{end} is seen to decrease and the stream split favors w_2 through the hottest heat exchanger (HX2). Results with the same trend was observed for several other process variables including UA_1 and UA_2. Refer to Appendix A.

5.2 Case 2: Two heat exchangers in series parallel with one heat exchanger

A selection of the results from optimized design and the corresponding results from optimal operation obtained from the Jäschke Temperature constraint are given in the following Tables 5.3 and 5.4. For the sake of the reports readability the optimized UA values used to obtain optimal operation are given in Appendix B.
Complete simulation results are given Appendix B.

Table 5.3: A selection of design and operating results for case study 2, scenario 1

	Optimal design	Optimal operation
T_{end} [°C]	201.43	201.40
w_1 [%]	47.1	45.2
w_2 [%]	52.9	54.8

Table 5.4: A selection of design and operating results for case study 2, scenario 4

	Optimal design	Optimal operation
T_{end} [°C]	217.28	217.11
w_1 [%]	42.0	37.8
w_2 [%]	58.0	62.2

As for the first case, the optimal operation given by the Jäschke Temperature controlled variable gives a T_{end} very close to the value from optimal design. For both optimal design and operation the inlet stream are divided such that the major part follows w_2 through heat exchanger 3. However, optimal operation results in an even more unevenly distribution in w_2s favor.

The analysis of the dependency on cost factor for case study 2 shows the same trend for a number of process variables as showed in Figure 5.1 and 5.2 in case study 1.

Worth noticing is some results from the case with initial heat capacity w_0 of 160 $kW/°C$ (as in scenario 1 and 2). The results for stream split (w_1 and w_2) and the size of heat exchanger 2 (UA_2) are shown in Figure 5.3 and 5.4 below, respectively.

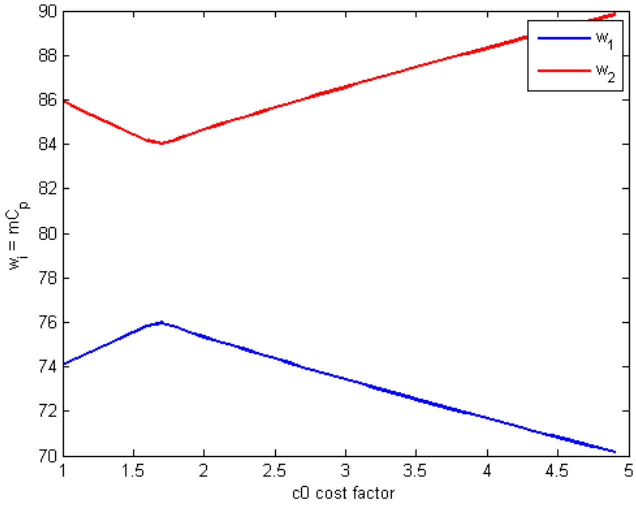

Figure 5.3: w_1 and w_2 $[kW/°\text{C}]$ as a function of cost factor c_0 for case study 2 with inlet heat capacity $w_0 = 160\ kW/°\text{C}$

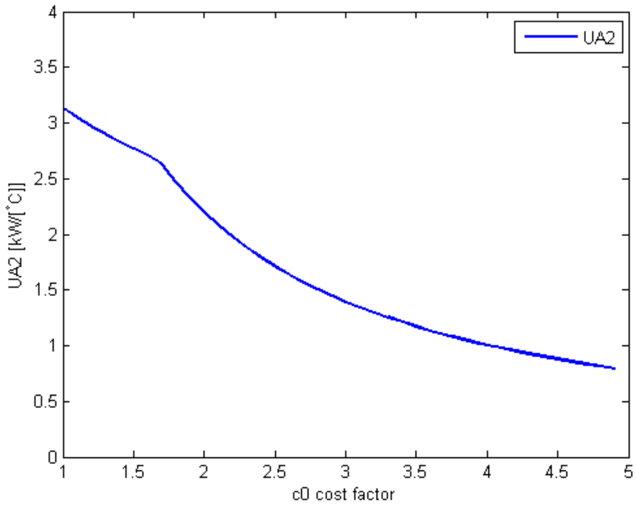

Figure 5.4: UA_2 $[kW/°\text{C}]$ as a function of cost factor c_0 for case study 2 with inlet heat capacity $w_0 = 160\ kW/°\text{C}$

These results shows that at with an inlet heat capacity of 160 $kW/°\text{C}$, the system experiences a shift around $c_0 = 1.7$. The stream fraction for the upper path (w_1) *increase* up to a cost factor of approximately 1.7, before it tips and start *decreasing*. From this point the system follows the same pattern that was discovered in the previous case.
Along with this increment of w_1, the size of the second heat exchanger (UA_2) experiences a much slighter decrease than the other heat exchangers (See Appendix B complete results). This strengthens the observation seen with the stream split. In addition, from Figure B.6 in Appendix B it is seen that the outlet temperature from heat exchanger 2, $Th2_{out}$ is decreasing for the same cost factor interval from 1 - 1.7. This means that more heat is transfered in heat exchanger 2 on the upper path w_1 during this interval, which also supports the first observations.

5.3 Case 3: Three heat exchangers in series parallel with two heat exchangers in series

Selected results from optimized design and the corresponding results from optimal operation obtained from the Jäschke Temperature constraint (Jaeschke 2012) are given in the following Tables 5.5 and 5.6. For the sake of the reports readability the optimized UA values used to obtain optimal operation are given in Appendix C.
More comprehensive simulation results are also given Appendix C.

Table 5.5: A selection of design and operating results for case study 3, scenario 1

	Optimal design	Optimal operation
T_{end} [°C]	182.01	181.93
w_1 [%]	34.7	30.6
w_2 [%]	65.3	69.4

Table 5.6: A selection of design and operating results for case study 3, scenario 4

	Optimal design	Optimal operation
T_{end} [°C]	180.20	180.12
w_1 [%]	25.5	22.0
w_2 [%]	74.5	78.0

For the case of three heat exchanger in series parallel with two heat exchangers in series the implementation of the Jäschke Temperature controlled variable also gave good results.

The same general trends were also observed for a number of process variables for this case as for the first and second case (Section 5.1 and 5.2) for increasing cost factor c_0. More interesting behavior was seen for certain units and variables, including especially heat exchanger size and stream split. Figure 5.5 and 5.6 shows the trends of UA_1 and UA_4, the first heat exchangers on upper and lower path, respectively.

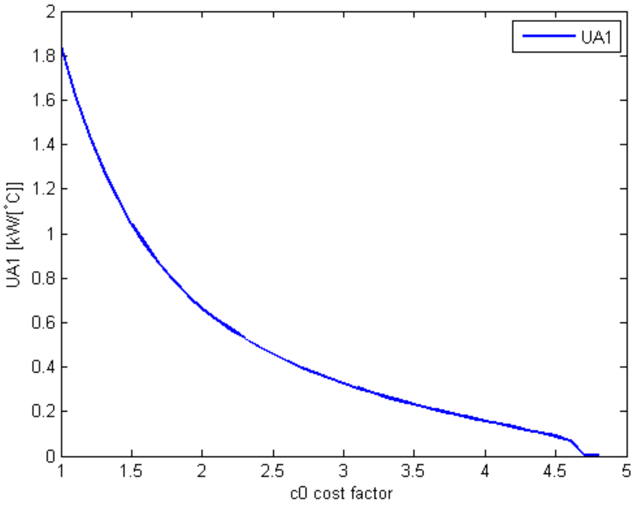

Figure 5.5: UA_1 $[kW/°\text{C}]$ as a function of cost factor c_0 for case study 3 with inlet heat capacity $w_0 = 180\ kW/°\text{C}$

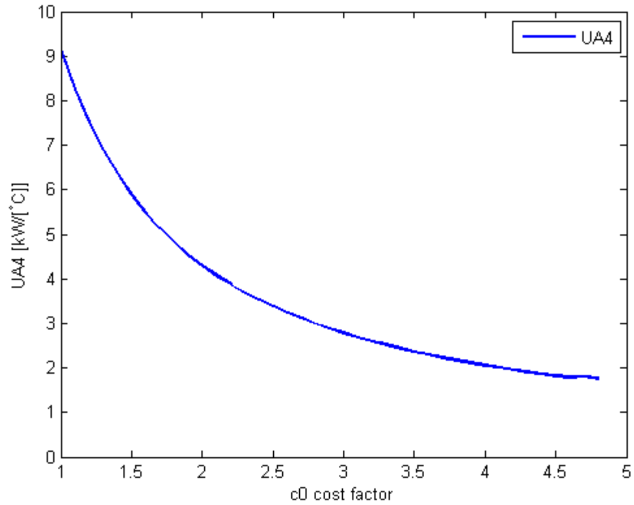

Figure 5.6: UA_4 $[kW/°\text{C}]$ as a function of cost factor c_0 for case study 3 with inlet heat capacity $w_0 = 180\ kW/°\text{C}$

Heat exchanger 1 (Figure 5.5) is the coolest operating heat exchanger in the whole network, with an inlet hot stream temperature of 190 °C. In the case of expensive heat exchangers (i.e. high c_0), heat exchanger 1 is removed completely from the network as its UA value approaches zero as the cost factor exceeds 4.5. This is also supported by the resulting UA_1 outlet temperature T_1 as it approaches the inlet temperature T_0 for the same c_0. See Appendix C.
On the other hand, heat exchanger 4 is the hottest in the network with highest heat capacity and second warmest hot stream. UA_4 shows the same trend as the other heat exchangers but has a significantly larger size.

A correlating observation is found in the stream split at the same inlet conditions. Figure 5.7

shows the stream split (w_1 and w_2) as a function of cost function c_0.

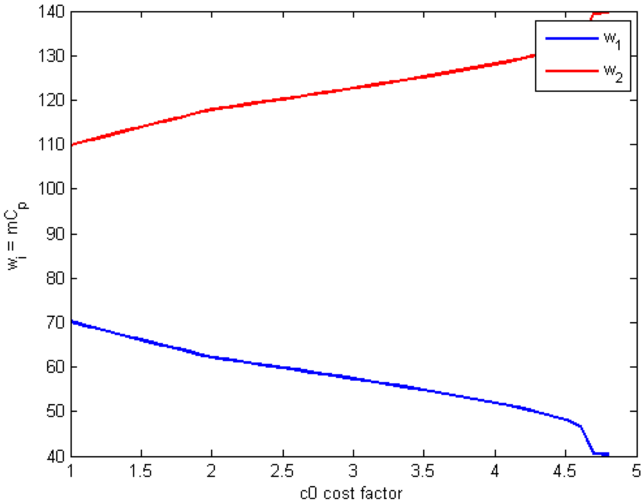

Figure 5.7: w_1 and w_2 $[kW/°\text{C}]$ as a function of cost factor c_0 for case study 3 with inlet heat capacity $w_0 = 180$ $kW/°\text{C}$

At the same point as UA_1 goes to zero and $T_1 = T_0$, i.e the point of no heat transfer in heat exchanger 1, a constant stream split is seen, with $w_2 = 140$ and $w_1 = 40$ $kW/°\text{C}$.

The additional investigation of the systems behavior with the other set of hot stream temperatures given in Section 4.4.3 gave the following results, where only a selection of process variables are listed and the rest are given in Appendix C.

Table 5.7: A selection of design and operating results for case study 3, scenario 1, with the additional set of hot stream temperatures

	Optimal design	Optimal operation
T_{end} [°C]	183.39	183.27
w_1 [%]	38.2	33.6
w_2 [%]	61.8	66.4

Table 5.8: A selection of design and operating results for case study 3, scenario 4, with the additional set of hot stream temperatures

	Optimal design	Optimal operation
T_{end} [°C]	180.63	180.54
w_1 [%]	27.3	23.4
w_2 [%]	72.7	76.6

Again, the Jäschke Temperature control variable shows good results.
Compared with the results from Table 5.5 and 5.6 with the original hot stream temperatures, the case with a hot heat exchanger followed by a slightly cooler heat exchanger (205 vs 203 °C, respectively) doesn't seem to have that much of an effect on the outlet temperature T_{end}.

6 Discussion and Further Work

The analysis has showed good results for the concept of using the Jäschke Temperature as a controlled variable for different heat exchanger networks. For all studied cases the optimal operation was within a 0.1% difference from the optimal design. The reason for why the Jäschke Temperature constraint always gave a little lower outlet temperature T_{end} than the optimal value has to do with the fundamental basement of the arithmetic mean temperature difference (AMTD) approximation (Equation 2.8) used in the derivation of the Jäschke Temperature. Even though the Jäschke Temperature gave good results on the outlet temperature, it showed different internal system behavior. The stream split differed by splitting the streams such that more of the cold fluid was sent in the direction of hottest heat exchangers, which in all cases was the lower path w_2. Therefore, the heat exchangers were used in a different way that gave various cold stream temperatures in the outlet of each heat exchanger (T_i).

Some interesting results appeared in the simulations. The results from case 2 presented in Section 5.2 gave "unexpected" trends considering both the expectations and the general results from the first case. With inlet heat capacity of 160 $kW/°$C, for cost factors ranging from 1 to about 1.6 (low heat exchanger operating and maintenance costs) the stream split trend showed increment of w_1 from 74 to 78 and decrement of w_2 from 86 to 84 $kW/°$C. However, the overall split is still in the favor of the lower path w_2. The topology in case 2 includes a second heat exchanger in the upper path. This heat exchanger (HX2) operates at a the highest temperature in the network (255 °C) but compared to the other heating fluids, the heating fluid in HX2 has a low heat capacity of only 27 $kW/°$C (Table 4.2). When it's cheap to run heat exchangers, i.e. at low c_0, it will, according to the results, be optimal to use this heat exchanger due to its high temperature. As the costs increase (above $c_0 = 1.6$) the low heat capacity will be a limiting factor for the heat integration in heat exchanger 2. As a result of this, the trend in $Th2_{out}$, w_1 and w_2 are reversed.

Considering the objective function the results indicated that, especially at high cost factors, the most profitable region of operation was when the heat capacity of the cold stream was close to the heat capacity of the hot streams in the heat exchangers. It is not shown to be optimal with exact match of heat capacities, but rather a trade of including both the topology setup and the economic effect of contribution from each heat exchanger. This can be related back to the results presented for case 3 in Section 5.3. From Figure 5.7 on optimal stream split at inlet heat capacity w_0 of 180 $kW/°$C it was seen that the two different paths obtained a constant stream split at a c_0 value of about 4.5. From this point w_1 and w_2 took the value 40 and 140 $kW/°$C, respectively (Table 4.3). Also, at this point the optimal design showed total exclusion of heat exchanger 1. Remaining on the upper path are heat exchanger 2 and 3, with hot stream heat capacities of 30 and 15 $kW/°$C, respectively. On the lower path (w_2) heat exchanger 4 and 5 are still present with their respective hot stream heat capacities of 70 and 20 $kW/°$C. The lower path has the hottest heat exchangers, so obviously the major part of w_0 is sent through heat exchanger 4 and 5. This results in a significantly greater heat exchanger size (UA) for HX4 and 5, compared to HX2 and 3. This is the main reason for the shut down of heat exchanger 1, as it is the coldest and by then are the poorest contributing unit of the network. Due to the implementation costs of a heat exchanger, the bottom line is that it is more economically to shut a heat exchanger down and rather use the capacity of hotter heat exchangers, than running a small heat exchanger with a small size and duty.

Continuing with the heat capacities and stream split in case 3, the optimized design gave w_2 constant at 140 $kW/°$C at high cost factors. This is obviously the right thing to do since this is the path with best heat exchangers. However, at high costs the size of the heat exchangers matters a lot to the objective function. Sending too much of the cold fluid through the lower path would result in two very big heat exchangers, which would be very expensive. Therefore, it is more economically to let a fraction big enough to match the heat capacity of heat exchanger 2 and 3 go though the *upper* path. w_0 is constant at 40 $kW/^{\circ}$C and the heat capacities of HX2 and 3 are 30 and 15 $kW/^{\circ}$C, respectively. The total available heat capacity on the upper path is hence 45 $kW/^{\circ}$C, and letting w_0 as 40 $kW/^{\circ}$C approach the total heat capacity of 45 $kW/^{\circ}$C is economically favorable since the best heat exchange occurs when the heat capacities of the cold and hot stream equals each other. The same results also apply for the first case and the second case. However, observations like these are mainly seen in scenarios where the inlet heat capacity is low enough to allow for splits like this as they often results in a great load on one specific heat exchanger or one of the paths.

For every case the heat integration was far from perfect, in the sense of no active constraints on the ΔT_{min} constraint (Section 3). Good heat integration requires big heat exchangers, which again leads to high heat exchanger costs. However, the heat integration is better at low cost factors, but still far from good in an environmental aspect.

These results gave good a indication for the use of Jäschke Temperature as a control variable on the steady state optimal operation of different heat exchanger networks. In order for this control method to get further acceptance in different industries dynamic simulations will be needed. A dynamic investigation of the method will indicate whether the control configuration gives an acceptable process control for different and changing process conditions with unexpected and random disturbances.

7 Conclusions

Three cases with different heat exchanger networks were optimized for 4 different process scenarios at steady state with the given objective function. By use of The Jäschke Temperature control variable (Jaeschke 2012) the optimal operation was tested against an optimal design.

The resulting optimal operation showed very good correlation with the optimal design, with an error of less than 0.1 %.

The investigation of cost factor's impacts on optimal design showed in general expected trends for all process variables. In cases with expensive heat exchanger operation lower outlet temperatures and smaller heat exchangers were observed.

The results also showed that it is more economically to use more of the capacity of the hot heat exchangers and rather shut down colder heat exchangers.

For further investigation on whether the Jäschke Temperature can work as a good control variable for parallel systems dynamic simulations will be needed.

References

Geankoplis, C. J. (2003), *Transport Processes and Separation Process Principles*, 4 edn, Pearson Education, Inc.

Halvorsen, I. J. & Skogestad, S. (1997), Indirect on-line optimization through setpoint control, Prepared for presentation at the AIChe 1997 Annual Meeting, Los Angeles, CA.

Incorpera, F. P. & DeWitt, D. P. (2007), *Introduction to Heat Transfer*, 2nd edition edn, John Wiley and sons, New York.

Jaeschke, J. (2012), United Kingdom Patent Application No. 1207770.7: Parallel Heat Exchanger Control.

Jensen, J. B. & Skogestad, S. (2008), 'Problems with specifying δt_{min} in the design of processes with heat exchangers', *Industrial and Engineering Chemistry Research (ACS Publications)* **47**(9), 3071–3075.

Sinnott, R. & Towler, G. (2009), *Chemical Engineering Design*, 5 edn, Coulson and Richardson's Chemical Engineering Series.

Skogestad, S. (2000), 'Plantwide control: the search for the self-optimizing control structure', *Journal of Process Control* **10**(5), 487–507.

Skogestad, S. (2003), *Prosessteknikk*, 2 edn, Tapir Akademiske Forlag.

Skogestad, S. (2004), 'Near-optimal operation by self-optimizing control: from process control and marathin running and business systems', *Computers and Chemical Engineering* (29), 127–137.

Underwood, A. J. V. (1933), 'Graphical computation of logarithmic mean temperature difference', *Industrial Chemist and Chemical Manufacturer* (9), 167–170.

A Complete Simulation Results Case 1

Optimized values for the 4 different scenarios are given in Table A.1 below:

Table A.1: Optimized process variables and optimal design the 4 different scenarios for case 1

(a) Scenario 1

$w_0 = 130$, $C_0 = 2$	
T_{end}	199.08 °C
T_1	185.88 °C
T_2	204.84 °C
w_1	39.47 $kW/°C$
w_2	90.52 $kW/°C$
UA_1	3.13 $kWm^2/°C$
UA_2	7.92 $kWm^2/°C$

(b) Scenario 2

$w_0 = 130$, $C_0 = 4$	
T_{end}	192.30 °C
T_1	179.27 °C
T_2	197.25 °C
w_1	35.84 $kW/°C$
w_2	94.11 $kW/°C$
UA_1	1.23 $kWm^2/°C$
UA_2	3.76 $kWm^2/°C$

(c) Scenario 3

$w_0 = 95$, $C_0 = 2$	
T_{end}	213.20 °C
T_1	196.98 °C
T_2	219.4 °C
w_1	26.32 $kW/°C$
w_2	68.68 $kW/°C$
UA_1	3.11 $kWm^2/°C$
UA_2	9.39 $kWm^2/°C$

(d) Scenario 4

$w_0 = 95$, $C_0 = 4$	
T_{end}	205.97 °C
T_1	189.80 °C
T_2	211.29 °C
w_1	23.52 $kW/°C$
w_2	71.48 $kW/°C$
UA_1	1.23 $kWm^2/°C$
UA_2	4.51 $kWm^2/°C$

Optimal operation results using the Jäschke temperature are given below in Table A.2

Table A.2: Optimal operation for the 4 different conditions for case 1, using the Jäschke Temperature equality constraint

(a) Scenario 1: $UA_1 = 3.13$, $UA_2 = 7.92$

$w_0 = 130$, $C_0 = 2$	
T_{end}	199.04 °C
T_1	187.8 °C
T_2	203.5 °C
w_1	36.8 $kW/°C$
w_2	93.2 $kW/°C$

(b) Scenario 2: $UA_1 = 1.225$, $UA_2 = 3.76$

$w_0 = 130$, $C_0 = 4$	
T_{end}	192.25 °C
T_1	181.7 °C
T_2	195.8 °C
w_1	32.75 $kW/°C$
w_2	94.11 $kW/°C$

(c) Scenario 3: $UA_1 = 3.11$, $UA_2 = 9.39$

$w_0 = 95$, $C_0 = 2$	
T_{end}	213.16 °C
T_1	199.18 °C
T_2	217.96 °C
w_1	24.25 $kW/°C$
w_2	70.70 $kW/°C$

(d) Scenario 4: $UA_1 = 1.23$, $UA_2 = 4.51$

$w_0 = 95$, $C_0 = 4$	
T_{end}	205.68 °C
T_1	192.68 °C
T_2	209.7 °C
w_1	21.01 $kW/°C$
w_2	73.99 $kW/°C$

For $w_0 = 95$ and 130 $kW/°C$ and c_0 varying from 1 - 5 several plots were made. The results are given in the following figures A.1 - A.10

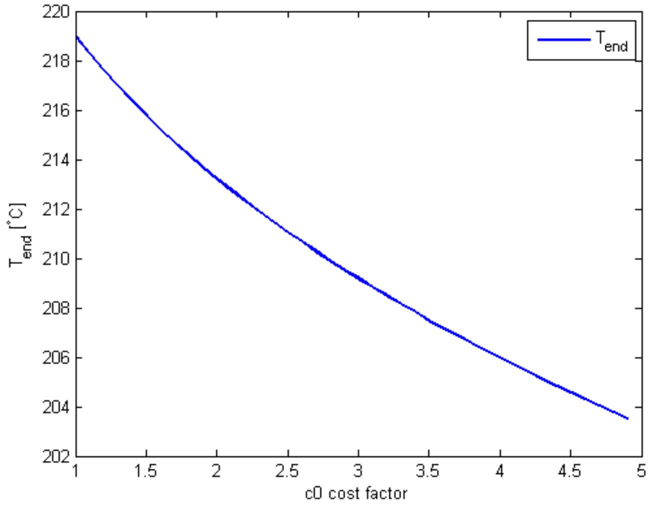

Figure A.1: Cost factor c_0 impacts on outlet temperature T_{end} for case 1 at $w_0 = 95\ kW/°\mathrm{C}$

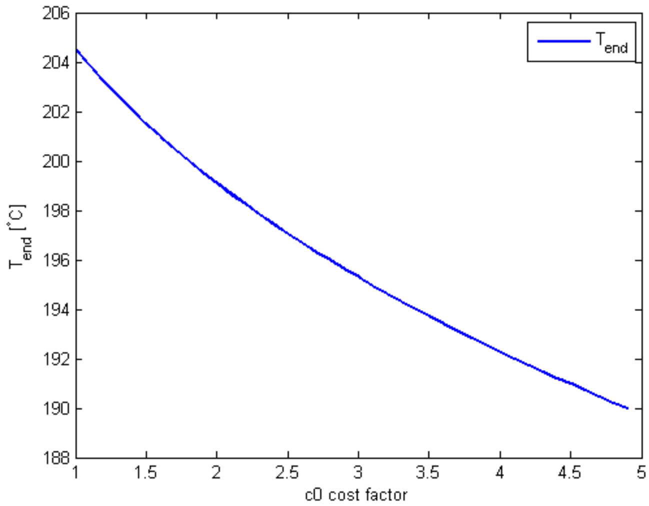

Figure A.2: Cost factor c_0 impacts on outlet temperature T_{end} for case 1 at $w_0 = 130\ kW/°\mathrm{C}$

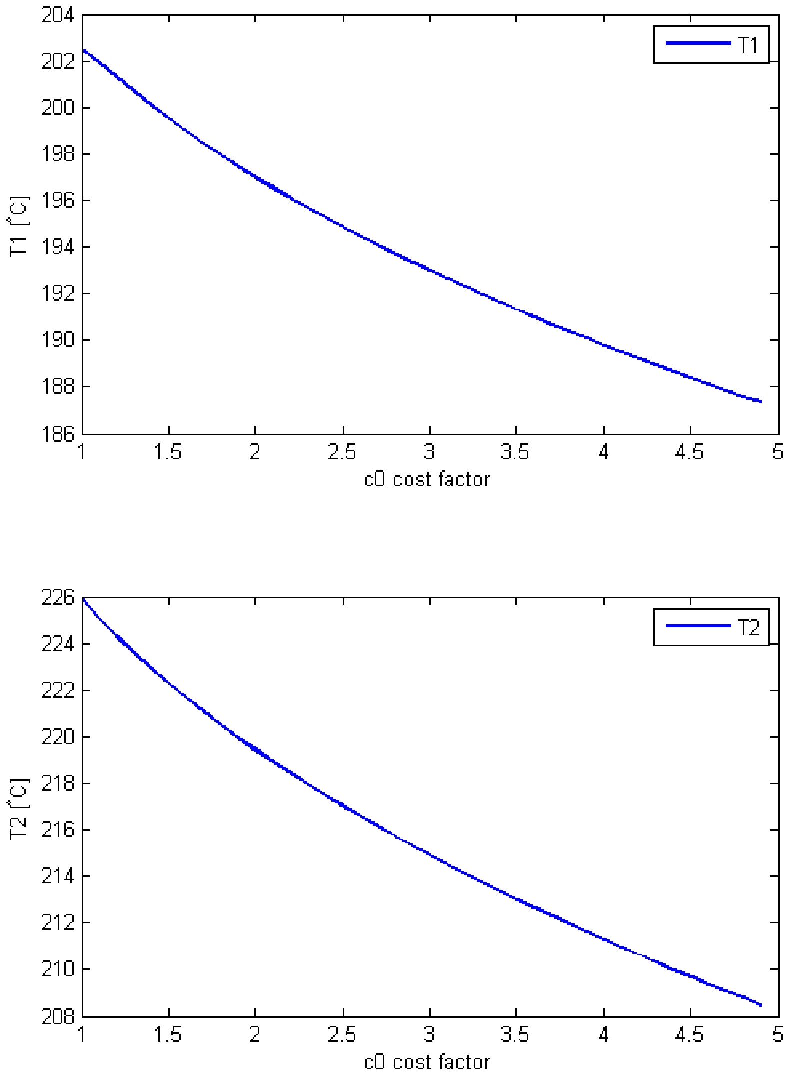

Figure A.3: Cost factor c_0 impacts on temperatures T_1 and T_2 for case 1 at $w_0 = 95\ kW/°\mathrm{C}$

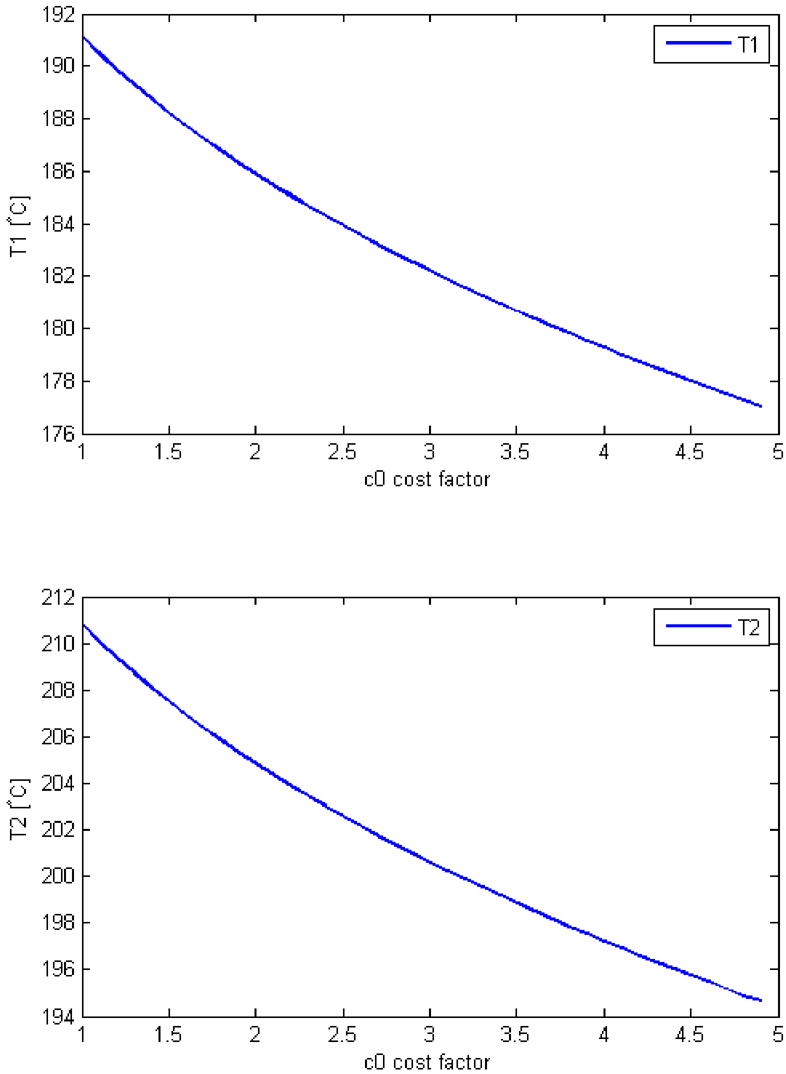

Figure A.4: Cost factor c_0 impacts on temperatures T_1 and T_2 for case 1 at $w_0 = 130\ kW/°\text{C}$

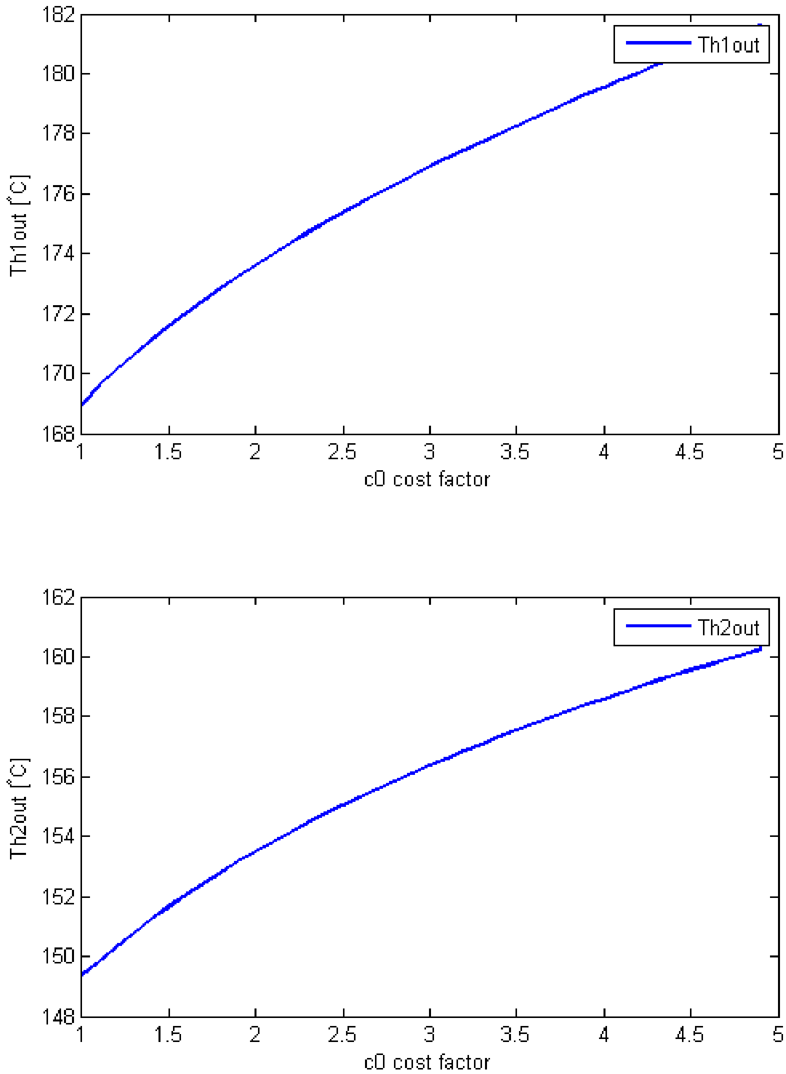

Figure A.5: Cost factor c_0 impacts on temperatures $Th1_{out}$ and $Th2_{out}$ for case 1 at $w_0 = 95$ $kW/°$C

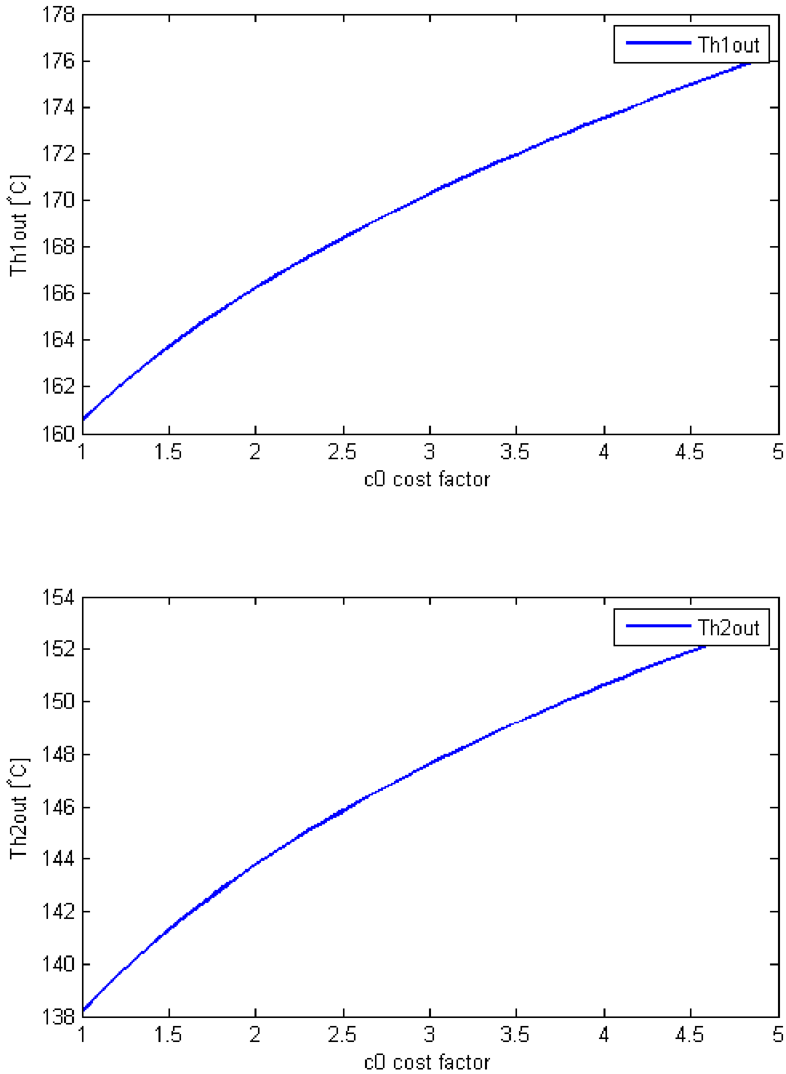

Figure A.6: Cost factor c_0 impacts on temperatures $Th1_{out}$ and $Th2_{out}$ for case 1 at $w_0 = 130$ $kW/°$C

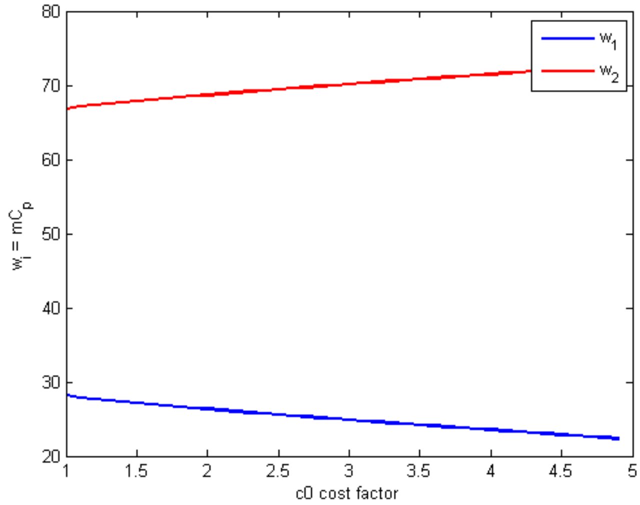

Figure A.7: Cost factor c_0 impacts on stream splits w_1 and w_2 for case 1 at $w_0 = 95\ kW/°$C

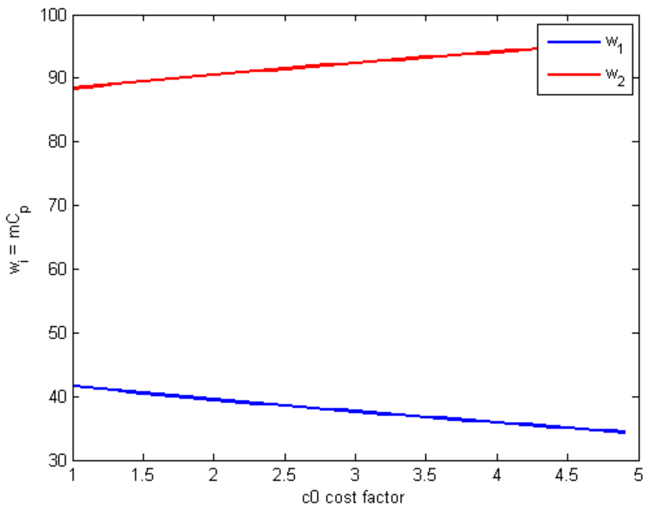

Figure A.8: Cost factor c_0 impacts on stream splits w_1 and w_2 for case 1 at $w_0 = 130\ kW/°$C

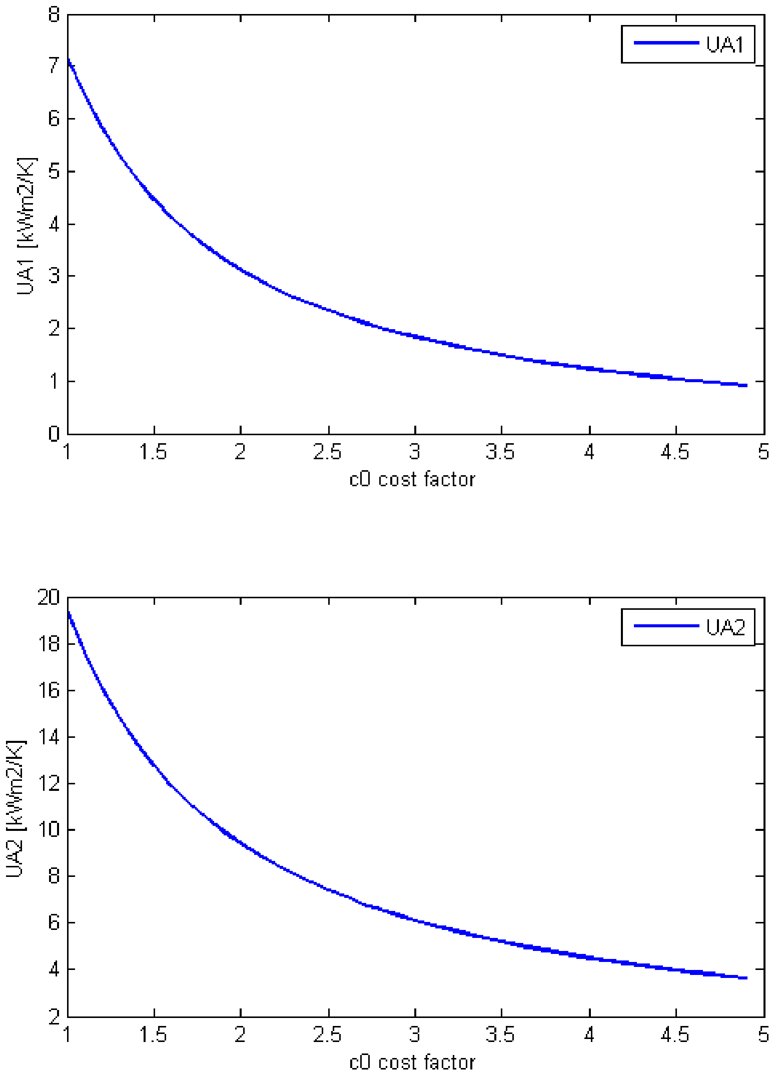

Figure A.9: Cost factor c_0 impacts on heat exchanger size UA for case 1 at $w_0 = 95\ kW/°C$

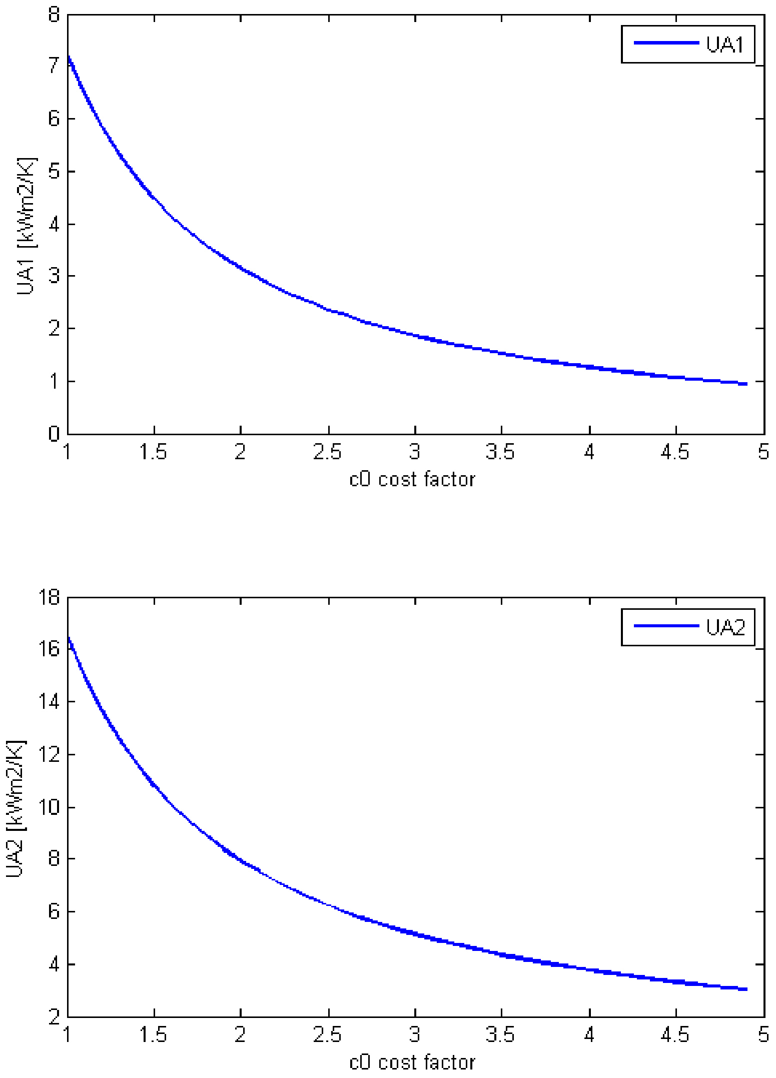

Figure A.10: Cost factor c_0 impacts on heat exchanger size UA for case 1 at $w_0 = 130\ kW/°\mathrm{C}$

B Complete Simulation Results Case 2

Optimized values for the 4 different scenarios are given in Table B.1 below:

Table B.1: Optimized process variables and optimal design from the 4 different scenarios for case 2

(a) Scenario 1

$w_0 = 160$, $C_0 = 2$	
T_{end}	201.43 °C
T_1	165.17 °C
T_2	196.37 °C
T_3	205.93 °C
w_1	75.34 $kW/°C$
w_2	84.65 $kW/°C$
UA_1	1.93 $kWm^2/°C$
UA_2	2.19 $kWm^2/°C$
UA_3	6.07 $kWm^2/°C$

(b) Scenario 2

$w_0 = 160$, $C_0 = 4$	
T_{end}	194.11 °C
T_1	158.53 °C
T_2	189.86 °C
T_3	197.56 °C
w_1	71.67 $kW/°C$
w_2	88.33 $kW/°C$
UA_1	0.76 $kWm^2/°C$
UA_2	1.01 $kWm^2/°C$
UA_3	2.86 $kWm^2/°C$

(c) Scenario 3

$w_0 = 95$, $C_0 = 2$	
T_{end}	225.01 °C
T_1	175.21 °C
T_2	217.24 °C
T_3	231.30 °C
w_1	42.53 $kW/°C$
w_2	52.47 $kW/°C$
UA_1	1.28 $kWm^2/°C$
UA_2	2.85 $kWm^2/°C$
UA_3	7.74 $kWm^2/°C$

(d) Scenario 4

$w_0 = 95$, $C_0 = 4$	
T_{end}	217.28 °C
T_1	165.65 °C
T_2	210.95 °C
T_3	221.86 °C
w_1	39.92 $kW/°C$
w_2	55.08 $kW/°C$
UA_1	0.47 $kWm^2/°C$
UA_2	1.33 $kWm^2/°C$
UA_3	3.75 $kWm^2/°C$

Table B.2: Optimal operation for the 4 different scenarios for case 2, using the Jäschke temperature equality constraint

(a) Scenario 1: $UA_1 = 1.93$, $UA_2 = 2.19$, $UA_3 = 6.07$

$w_0 = 160$, $C_0 = 2$	
T_{end}	201.40 °C
T_1	166.10 °C
T_2	197.84 °C
T_3	204.34 °C
w_1	72.36 $kW/°C$
w_2	87.64 $kW/°C$

(b) Scenario 2: $UA_1 = 0.76$, $UA_2 = 1.01$, $UA_3 = 2.86$

$w_0 = 160$, $C_0 = 4$	
T_{end}	194.10 °C
T_1	159.20 °C
T_2	191.12 °C
T_3	196.37 °C
w_1	69.20 $kW/°C$
w_2	90.80 $kW/°C$

(c) Scenario 3 $UA_1 = 1.28$, $UA_2 = 2.85$, $UA_3 = 7.74$

$w_0 = 95$, $C_0 = 2$	
T_{end}	225.01 °C
T_1	175.21 °C
T_2	217.24 °C
T_3	231.30 °C
w_1	42.53 $kW/°C$
w_2	52.47 $kW/°C$

(d) Scenario 4 $UA_1 = 0.47$, $UA_2 = 1.33$, $UA_3 = 3.75$

$w_0 = 95$, $C_0 = 4$	
T_{end}	217.11 °C
T_1	167.82 °C
T_2	214.75 °C
T_3	218.54 °C
w_1	35.95 $kW/°C$
w_2	59.04 $kW/°C$

For $w_0 = 95$ and 160 $kW/°C$ and c_0 varying from 1 - 5 several plots were made. The results are given in the following figures B.1 - B.10

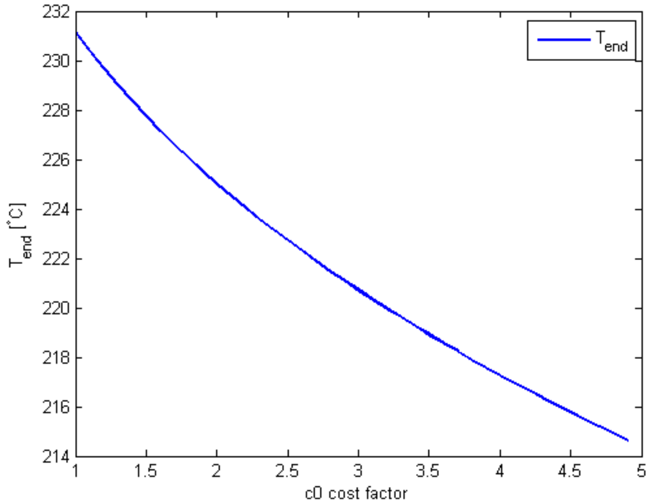

Figure B.1: Cost factor c_0 impacts on outlet temperature T_{end} for case 2 at $w_0 = 95$ $kW/°$C

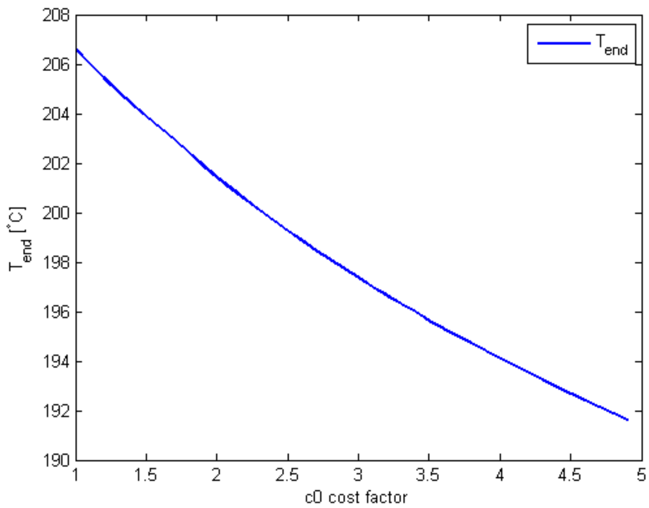

Figure B.2: Cost factor c_0 impacts on outlet temperature T_{end} for case 2 at $w_0 = 160$ $kW/°$C

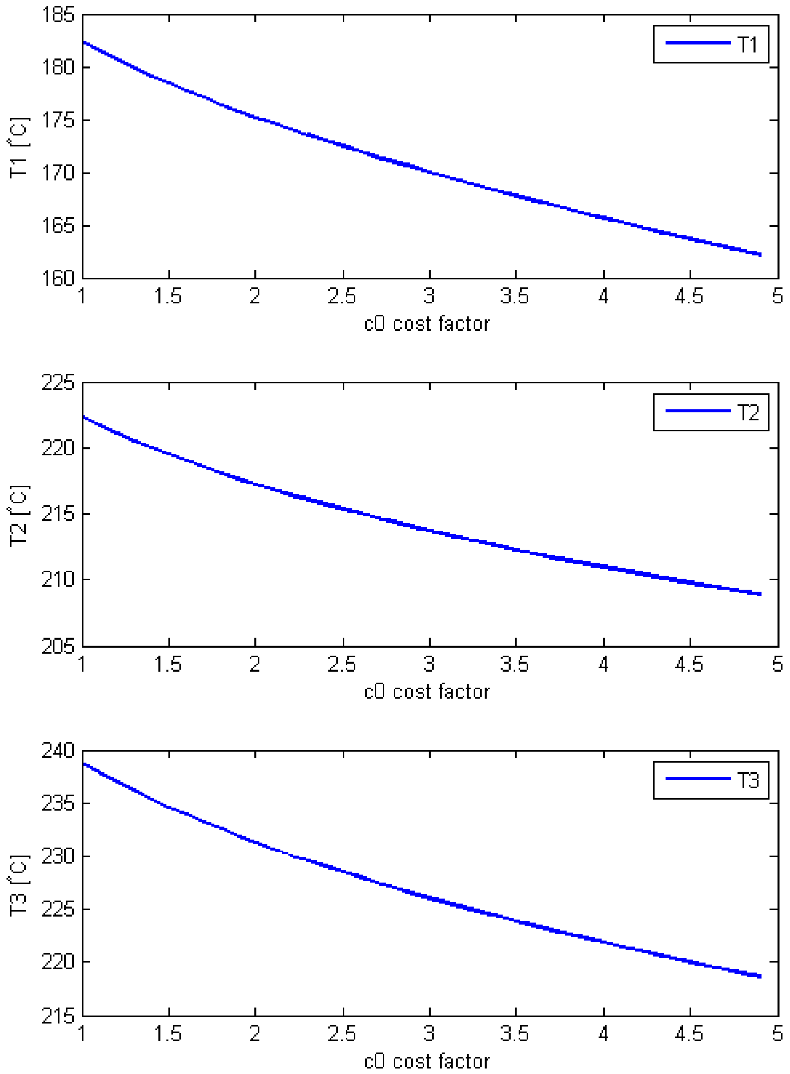

Figure B.3: Cost factor c_0 impacts on temperatures T_1 and T_2 for case 2 at $w_0 = 95\ kW/°\mathrm{C}$

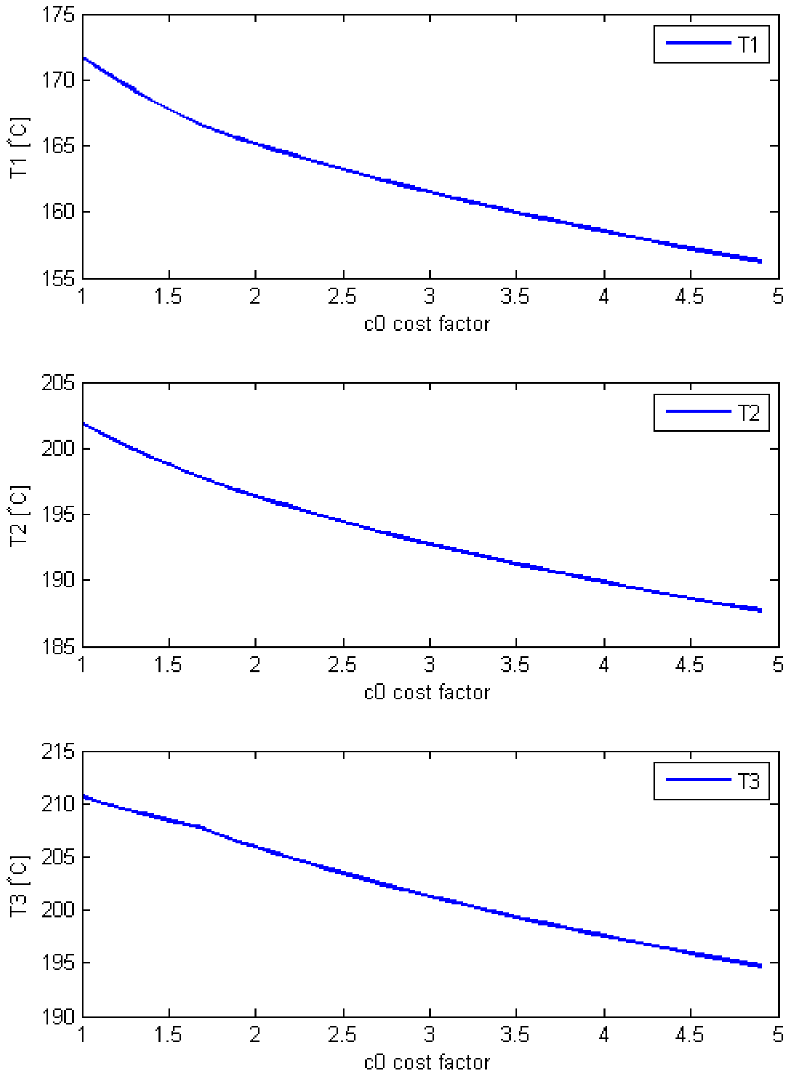

Figure B.4: Cost factor c_0 impacts on temperatures T_1 and T_2 for case 2 at $w_0 = 160\ kW/°\mathrm{C}$

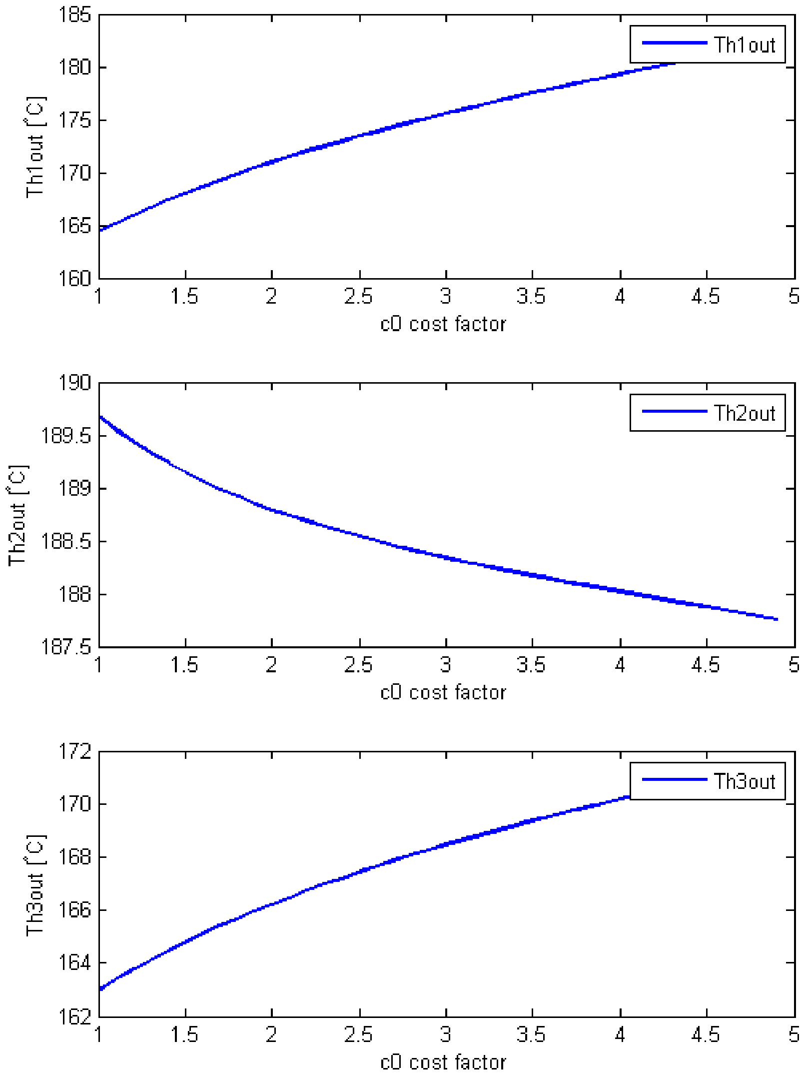

Figure B.5: Cost factor c_0 impacts on temperatures $Th1_{out}$ and $Th2_{out}$ for case 2 at $w_0 = 95$ $kW/°$C

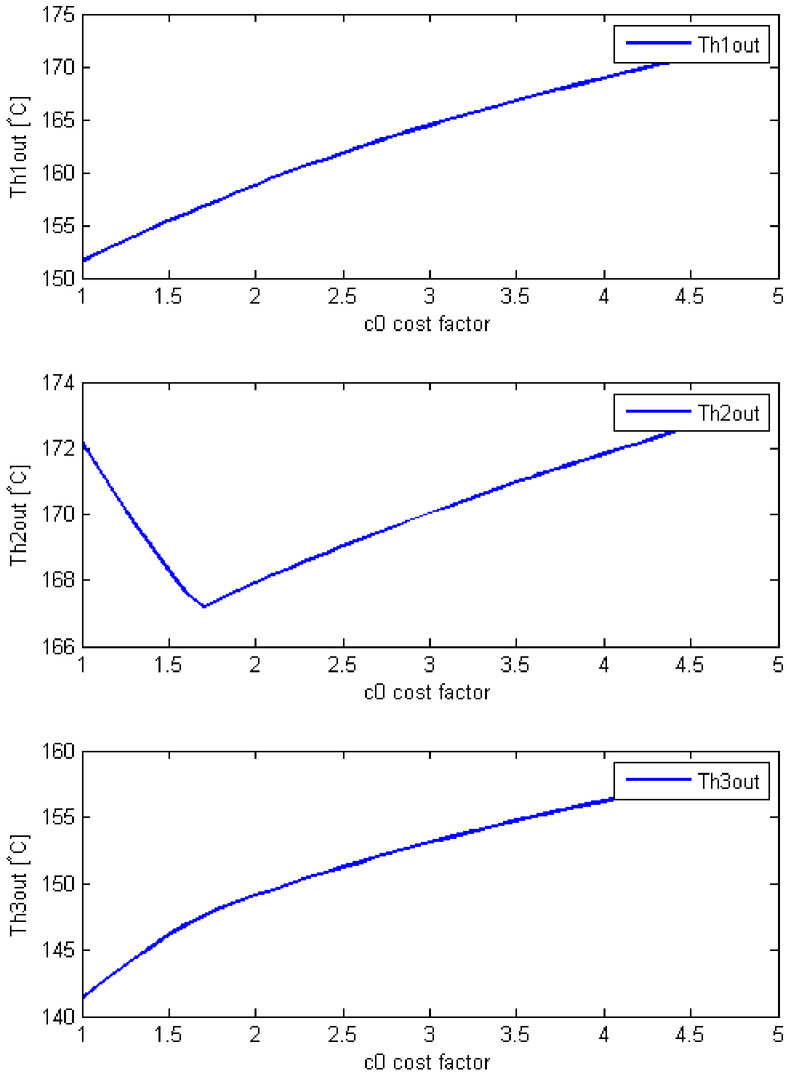

Figure B.6: Cost factor c_0 impacts on temperatures $Th1_{out}$ and $Th2_{out}$ for case 2 at $w_0 = 160$ $kW/°\text{C}$

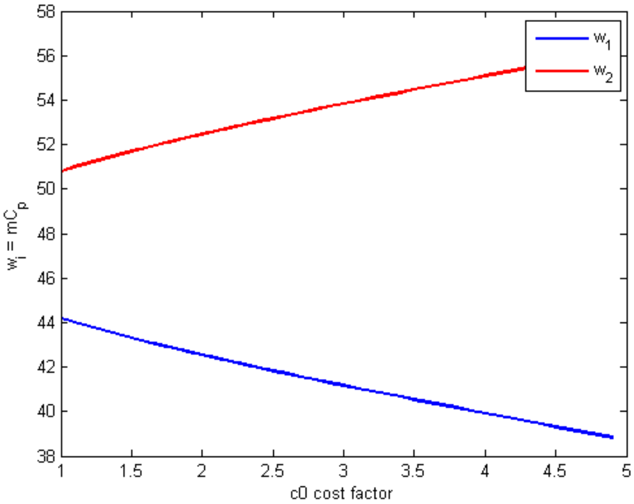

Figure B.7: Cost factor c_0 impacts on stream splits w_1 and w_2 for case 2 at $w_0 = 95\ kW/°\mathrm{C}$

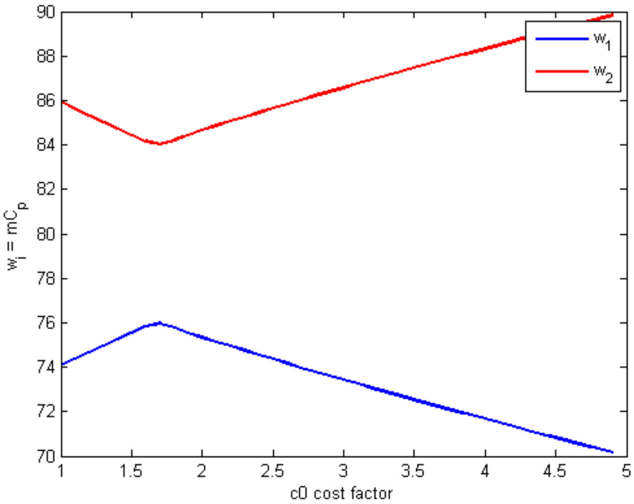

Figure B.8: Cost factor c_0 impacts on stream splits w_1 and w_2 for case 2 at $w_0 = 160\ kW/°\mathrm{C}$

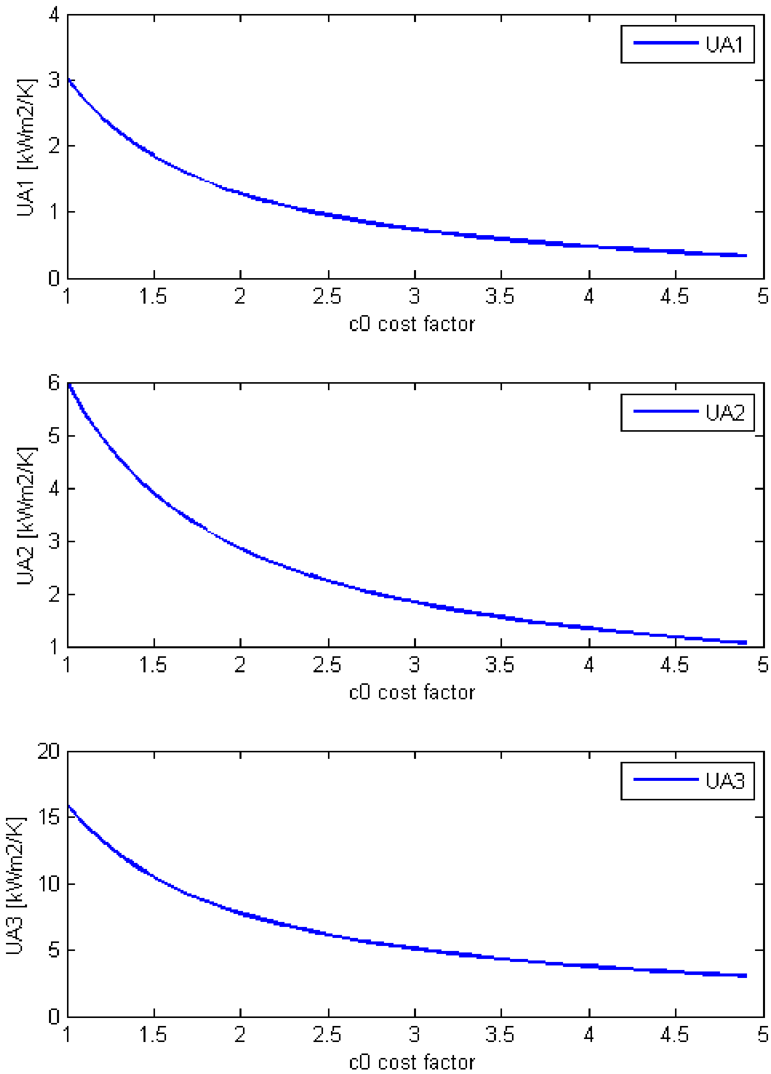

Figure B.9: Cost factor c_0 impacts on heat exchanger size UA for case 2 at $w_0 = 95\ kW/°C$

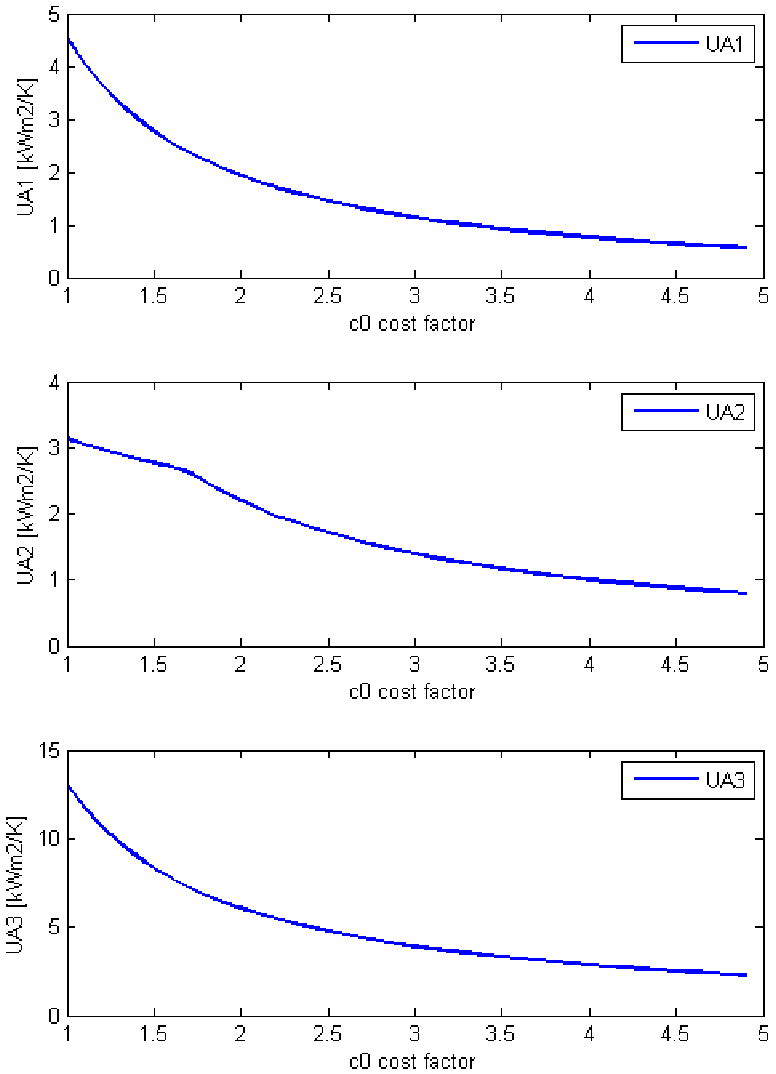

Figure B.10: Cost factor c_0 impacts on heat exchanger size UA for case 2 at $w_0 = 160\ kW/°C$

C Complete Simulation Results Case 3

Optimized values for the 4 different scenarios are given in Table C.1 below:

Table C.1: Optimized process variables and optimal design from the 4 different scenarios for case 3

(a) Scenario 1

$w_0 = 180$, $C_0 = 2$	
T_{end}	182.01 °C
T_1	150.30 °C
T_2	164.46 °C
T_3	174.42 °C
T_4	173.45 °C
T_5	186.05 °C
w_1	62.43 $kW/°C$
w_2	117.57 $kW/°C$
UA_1	0.67 $kWm^2/°C$
UA_2	0.81 $kWm^2/°C$
UA_3	0.63 $kWm^2/°C$
UA_4	4.30 $kWm^2/°C$
UA_5	1.31 $kWm^2/°C$

(b) Scenario 2

$w_0 = 180$, $C_0 = 4$	
T_{end}	174.61 °C
T_1	142.12 °C
T_2	156.94 °C
T_3	167.84 °C
T_4	165.88 °C
T_5	177.35 °C
w_1	51.90 $kW/°C$
w_2	128.10 $kW/°C$
UA_1	0.16 $kWm^2/°C$
UA_2	0.31 $kWm^2/°C$
UA_3	0.26 $kWm^2/°C$
UA_4	2.06 $kWm^2/°C$
UA_5	0.64 $kWm^2/°C$

(c) Scenario 3

$w_0 = 150$, $C_0 = 2$	
T_{end}	187.89 °C
T_1	151.98 °C
T_2	167.97 °C
T_3	179.31 °C
T_4	178.62 °C
T_5	192.06 °C
w_1	49.00 $kW/°C$
w_2	101.00 $kW/°C$
UA_1	0.52 $kWm^2/°C$
UA_2	0.79 $kWm^2/°C$
UA_3	0.70 $kWm^2/°C$
UA_4	4.62 $kWm^2/°C$
UA_5	1.54 $kWm^2/°C$

(d) Scenario 4

$w_0 = 150$, $C_0 = 2$	
T_{end}	180.20 °C
T_1	138.63 °C
T_2	159.16 °C
T_3	172.86 °C
T_4	170.36 °C
T_5	182.72 °C
w_1	38.29 $kW/°C$
w_2	111.71 $kW/°C$
UA_1	0.06 $kWm^2/°C$
UA_2	0.31 $kWm^2/°C$
UA_3	0.29 $kWm^2/°C$
UA_4	2.29 $kWm^2/°C$
UA_5	0.74 $kWm^2/°C$

Table C.2: Optimal operation for the 4 different scenarios for case 3, using the Jäschke temperature equality constraint

(a) Scenario 1: $UA_1 = 0.67$, $UA_2 = 0.81$, $UA_3 = 0.63$, $UA_4 = 4.30$, $UA_5 = 1.31$

$w_0 = 180$, $C_0 = 2$	
T_{end}	181.93 °C
T_1	152.19 °C
T_2	166.86 °C
T_3	177.19 °C
T_4	171.69 °C
T_5	184.02 °C
w_1	55.14 $kW/$°C
w_2	124.86 $kW/$°C

(b) Scenario 2: $UA_1 = 0.16$, $UA_2 = 0.31$, $UA_3 = 0.26$, $UA_3 = 2.06$, $UA_5 = 0.64$

$w_0 = 180$, $C_0 = 4$	
T_{end}	174.57 °C
T_1	143.57 °C
T_2	159.30 °C
T_3	170.69 °C
T_4	164.64 °C
T_5	175.88 °C
w_1	45.46 $kW/$°C
w_2	134.54 $kW/$°C

(c) Scenario 3 $UA_1 = 0.52$, $UA_2 = 0.79$, $UA_3 = 0.70$, $UA_4 = 4.62$, $UA_5 = 1.54$

$w_0 = 150$, $C_0 = 2$	
T_{end}	187.77 °C
T_1	154.20 °C
T_2	170.73 °C
T_3	182.48 °C
T_4	176.64 °C
T_5	189.86 °C
w_1	42.42 $kW/$°C
w_2	107.58 $kW/$°C

(d) Scenario 4 $UA_1 = 0.06$, $UA_2 = 0.31$, $UA_3 = 0.29$, $UA_4 = 2.29$, $UA_5 = 0.74$

$w_0 = 150$, $C_0 = 4$	
T_{end}	180.12 °C
T_1	139.52 °C
T_2	161.85 °C
T_3	176.15 °C
T_4	169.09 °C
T_5	181.24 °C
w_1	33.00 $kW/$°C
w_2	107.00 $kW/$°C

For the additional investigation with new hot stream temperatures, the results from optimization ans optimal operation with optimized design from the original case with the default hot stream temperatures are given in the following Table C.4. To get a better overview the new hot stream temperatures are cited here as well:

- Th1 = 205°C
- Th2 = 203°C
- Th3 = 220°C
- Th4 = 220°C
- Th5 = 248°C

Table C.3: Optimized process variables for the 4 different scenarios for case 3 with the new hot stream temperatures, with optimized design from the original case

(a) Scenario 1: $UA_1 = 0.67$, $UA_2 = 0.81$, $UA_3 = 0.63$, $UA_4 = 4.30$, $UA_5 = 1.31$

$w_0 = 180$, $C_0 = 2$	
T_{end}	183.39 °C
T_1	157.48 °C
T_2	167.74 °C
T_3	176.06 °C
T_4	175.11 °C
T_5	187.91 °C
w_1	68.70 $kW/°C$
w_2	111.30 $kW/°C$

(b) Scenario 2: $UA_1 = 0.16$, $UA_2 = 0.31$, $UA_3 = 0.26$, $UA_3 = 2.06$, $UA_5 = 0.64$

$w_0 = 180$, $C_0 = 4$	
T_{end}	175.52 °C
T_1	147.89 °C
T_2	159.29 °C
T_3	168.61 °C
T_4	167.05 °C
T_5	178.75 °C
w_1	57.47 $kW/°C$
w_2	122.53 $kW/°C$

(c) Scenario 3 $UA_1 = 0.52$, $UA_2 = 0.79$, $UA_3 = 0.70$, $UA_4 = 4.62$, $UA_5 = 1.54$

$w_0 = 150$, $C_0 = 2$	
T_{end}	189.07 °C
T_1	159.98 °C
T_2	171.13 °C
T_3	180.62 °C
T_4	180.25 °C
T_5	193.83 °C
w_1	54.03 $kW/°C$
w_2	95.97 $kW/°C$

(d) Scenario 4 $UA_1 = 0.06$, $UA_2 = 0.31$, $UA_3 = 0.29$, $UA_4 = 2.29$, $UA_5 = 0.74$

$w_0 = 150$, $C_0 = 4$	
T_{end}	180.63 °C
T_1	143.42 °C
T_2	160.52 °C
T_3	172.98 °C
T_4	171.05 °C
T_5	183.51 °C
w_1	40.96 $kW/°C$
w_2	109.04 $kW/°C$

Table C.4: Optimal operation of the 4 different scenarios for case 3 with the new hot stream temperatures, with optimized design from the original case

(a) Scenario 1: $UA_1 = 0.67$, $UA_2 = 0.81$, $UA_3 = 0.63$, $UA_4 = 4.30$, $UA_5 = 1.31$

$w_0 = 180$, $C_0 = 2$	
T_{end}	183.27 °C
T_1	160.05 °C
T_2	170.33 °C
T_3	178.91 °C
T_4	172.96 °C
T_5	185.47 °C
w_1	60.42 $kW/°C$
w_2	119.58 $kW/°C$

(b) Scenario 2: $UA_1 = 0.16$, $UA_2 = 0.31$, $UA_3 = 0.26$, $UA_3 = 2.06$, $UA_5 = 0.64$

$w_0 = 180$, $C_0 = 4$	
T_{end}	175.43 °C
T_1	149.90 °C
T_2	161.80 °C
T_3	171.60 °C
T_4	165.50 °C
T_5	176.90 °C
w_1	49.89 $kW/°C$
w_2	130.11 $kW/°C$

(c) Scenario 3 $UA_1 = 0.52$, $UA_2 = 0.79$, $UA_3 = 0.70$, $UA_4 = 4.62$, $UA_5 = 1.54$

$w_0 = 150$, $C_0 = 2$	
T_{end}	188.89 °C
T_1	163.21 °C
T_2	174.16 °C
T_3	183.92 °C
T_4	177.77 °C
T_5	191.11 °C
w_1	46.24 $kW/°C$
w_2	103.76 $kW/°C$

(d) Scenario 4 $UA_1 = 0.06$, $UA_2 = 0.31$, $UA_3 = 0.29$, $UA_4 = 2.29$, $UA_5 = 0.74$

$w_0 = 150$, $C_0 = 4$	
T_{end}	180.54 °C
T_1	145.17 °C
T_2	163.32 °C
T_3	176.36 °C
T_4	169.59 °C
T_5	181.82 °C
w_1	35.08 $kW/°C$
w_2	114.92 $kW/°C$

For $w_0 = 150$ and 180 $kW/°C$ and c_0 varying from 1 - 5 several plots were made. The results are given in the following figures C.1 - C.9

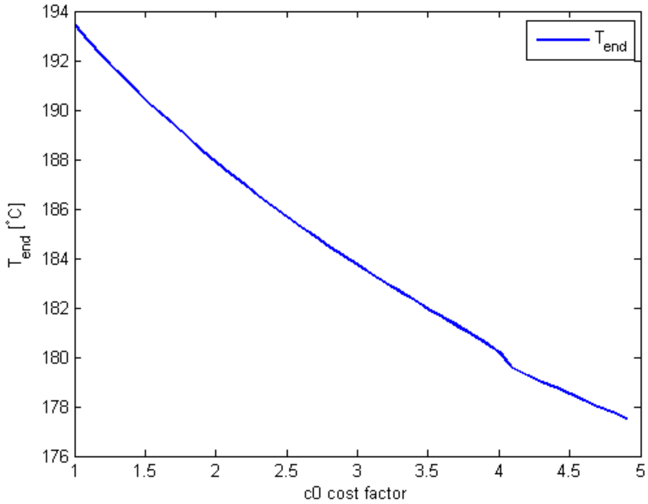

Figure C.1: Cost factor c_0 impacts on outlet temperature T_{end} for case 3 at $w_0 = 150\ kW/°\text{C}$

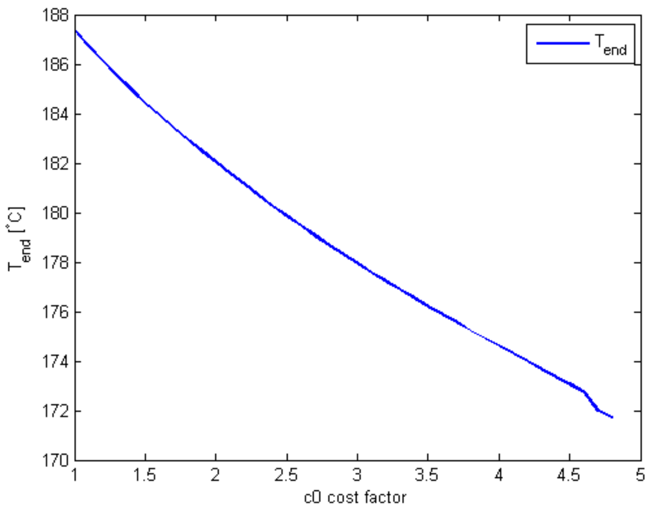

Figure C.2: Cost factor c_0 impacts on outlet temperature T_{end} for case 3 at $w_0 = 180\ kW/°\text{C}$

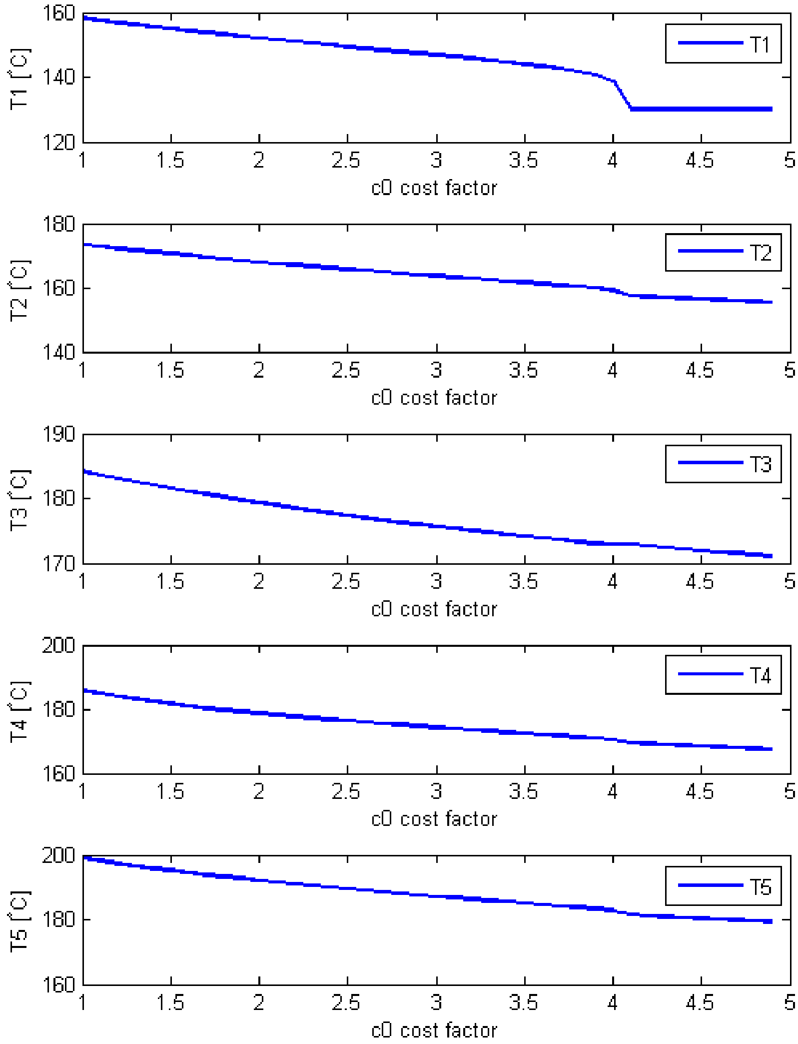

Figure C.3: Cost factor c_0 impacts on temperatures T_1 and T_2 for case 3 at $w_0 = 150\ kW/°$C

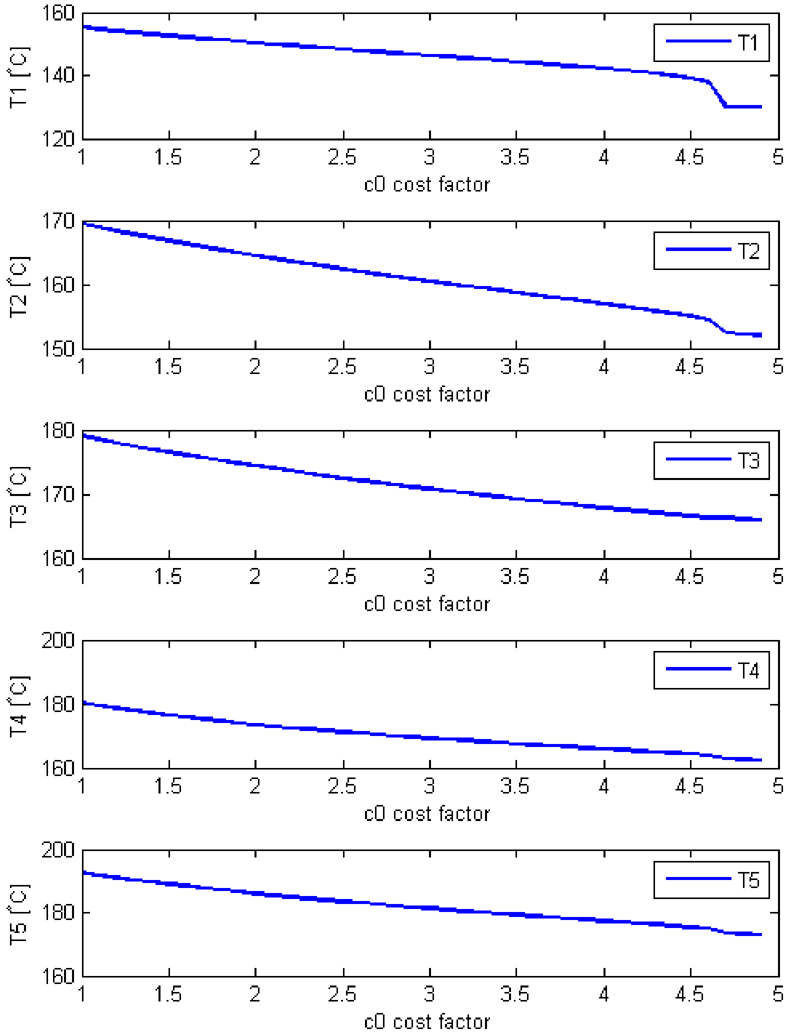

Figure C.4: Cost factor c_0 impacts on temperatures T_1 and T_2 for case 3 at $w_0 = 180\ kW/°\text{C}$

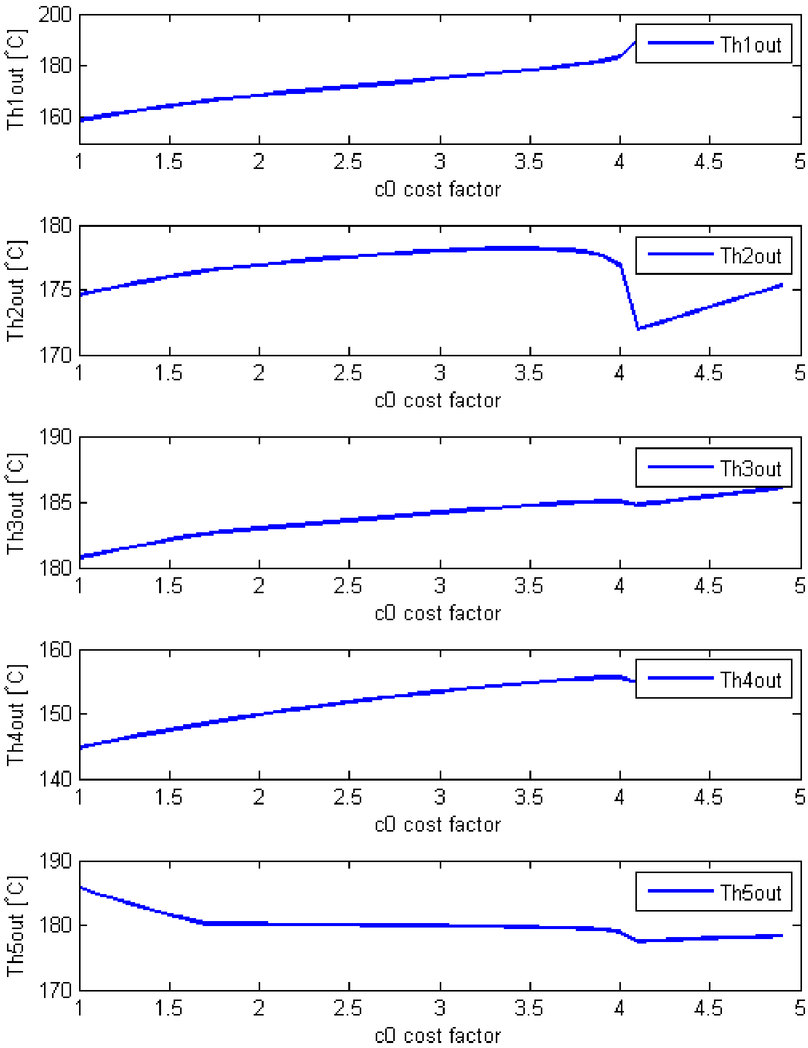

Figure C.5: Cost factor c_0 impacts on temperatures $Th1_{out}$ and $Th2_{out}$ for case 3 at $w_0 = 150$ $kW/°\mathrm{C}$

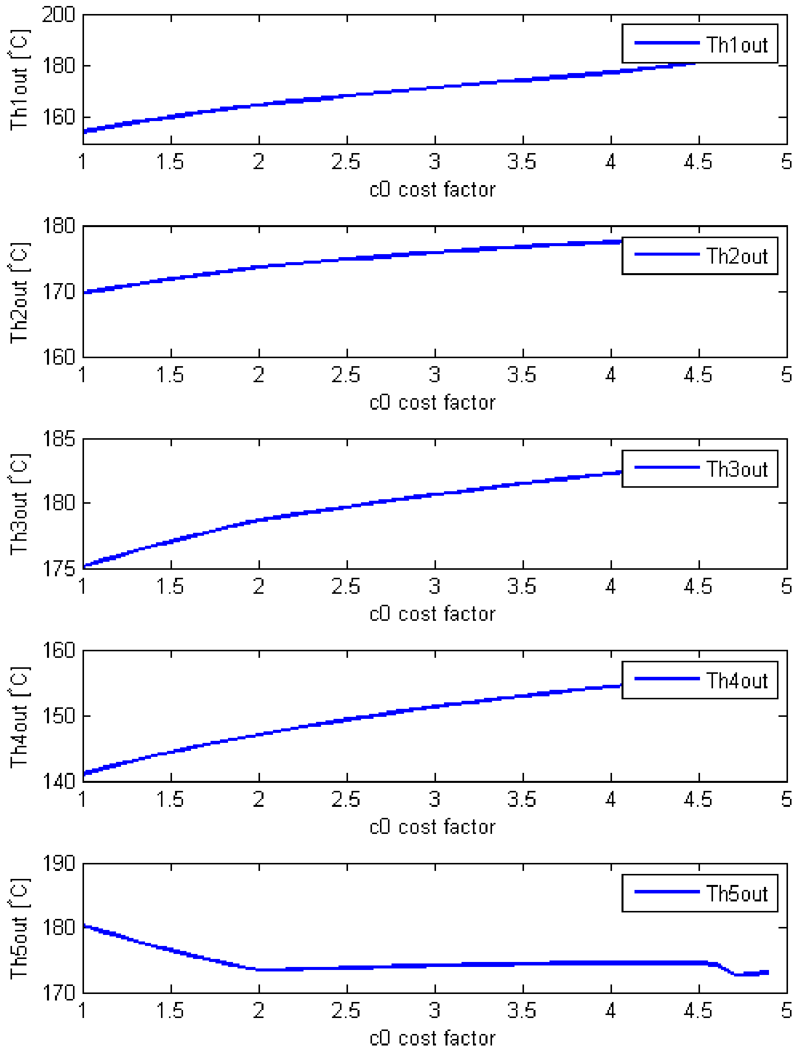

Figure C.6: Cost factor c_0 impacts on temperatures $Th1_{out}$ and $Th2_{out}$ for case 3 at $w_0 = 180$ $kW/°\text{C}$

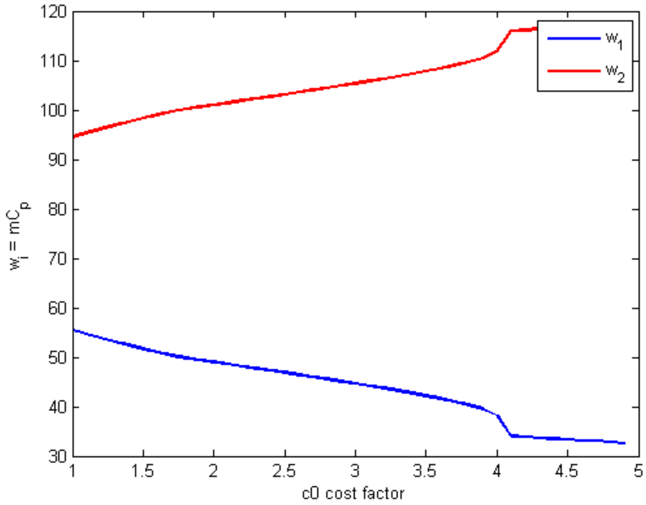

Figure C.7: Cost factor c_0 impacts on stream splits w_1 and w_2 for case 3 at $w_0 = 150\ kW/°C$

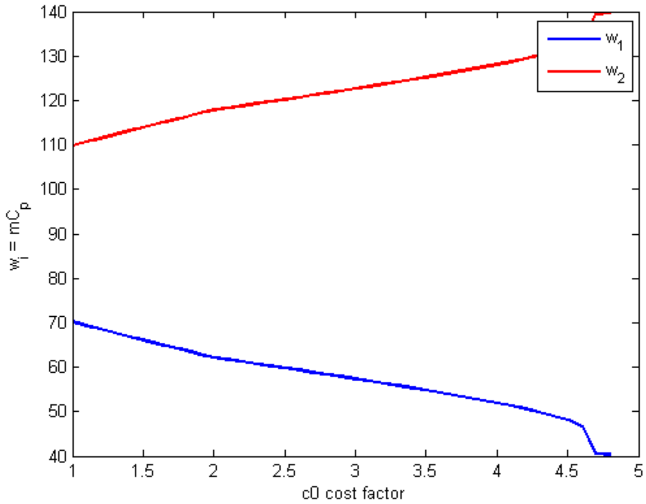

Figure C.8: Cost factor c_0 impacts on stream splits w_1 and w_2 for case 3 at $w_0 = 180\ kW/°C$

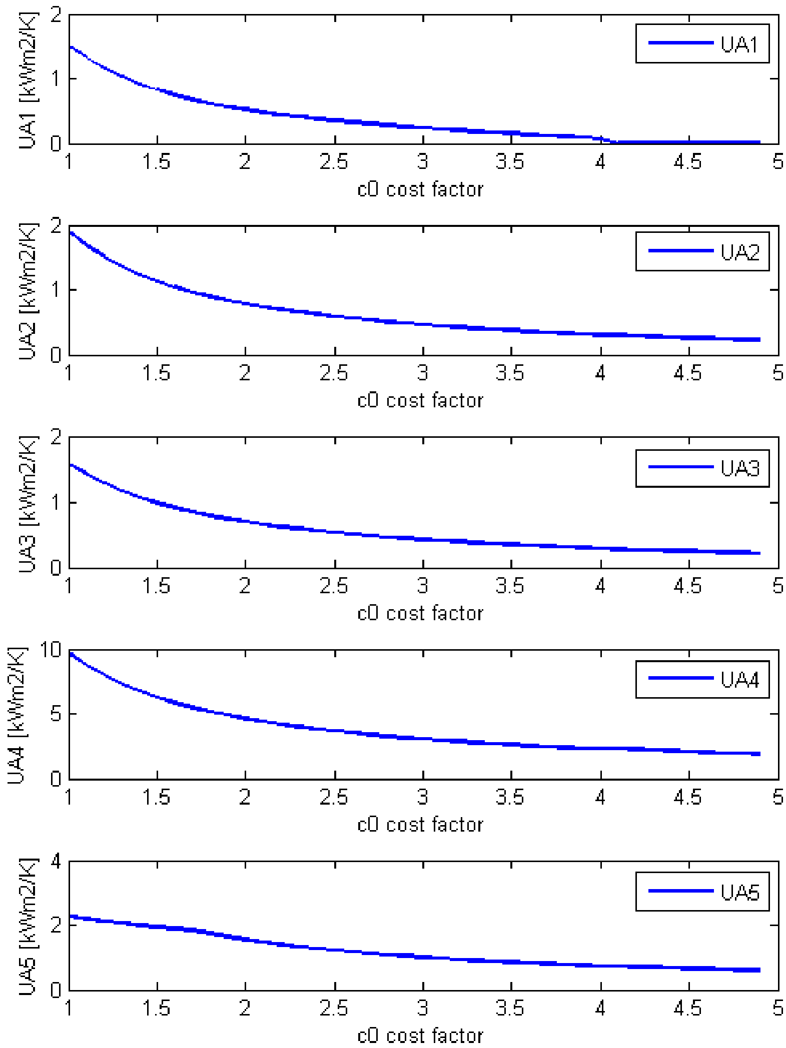

Figure C.9: Cost factor c_0 impacts on heat exchanger size UA for case 3 at $w_0 = 150\ kW/°\mathrm{C}$

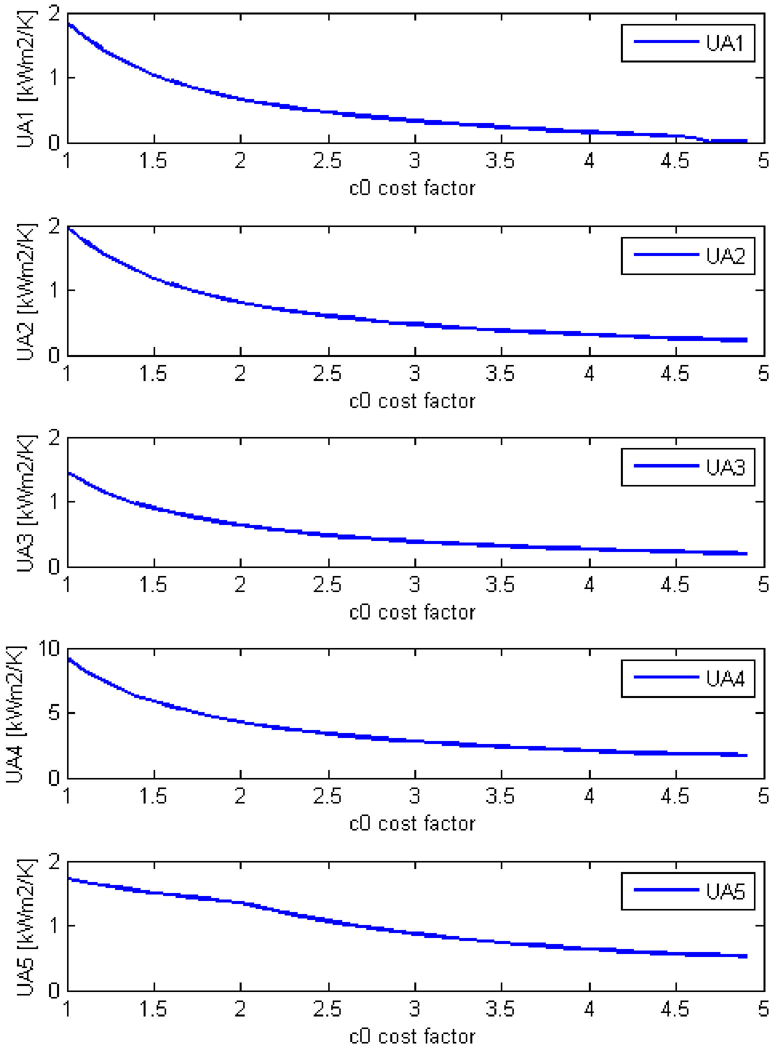

Figure C.10: Cost factor c_0 impacts on heat exchanger size UA for case 3 at $w_0 = 180\ kW/°\text{C}$

D Matlab Scripts

D.1 Case 1

`RunHEN_11_HXD.m`

```
%% Model to simulate a steady state HEN
%  Optimal Design

% Topology to be investigated:

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                1                  %
%        ----------0----------       %
%   ----|                    |----   %
%        ----------0----------       %
%                2                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clear all;
clc;

%% Parameters

par.w0  = 130;    %[J/K] w= miCpi
par.wh1 = 60;     %[J/K]
par.wh2 = 65;     %[J/K]

par.Th1 = 203;    %[degC]
par.Th2 = 248;    %[degC]
par.T0 = 130;     %[degC]

par.DeltaTmin = 0.5;      %[degC]
par.n = 0.65;             % Cost exponent
% par.c0 = 2.5;           %[$/m2]

par.sc.x = [200*ones(5,1);45;45;5000*ones(2,1);400*ones(2,1)];
par.sc.j = 200;

%% Optimization

%x0 = [Tend  T1   T2   Th1out  Th2out   w1     w2     Q1     Q2     UA1    UA2]
x0   = [207   186  227  144     146     60.5   60.5   3532   3592    27    27.6]';
x0 = x0./par.sc.x;

A = []; b = []; Aeq = []; Beq = [];
LB = 0.01*ones(11,1); UB = inf*ones(11,1);
options = optimset('Algorithm','interior-point','display','iter'...
    ,'MaxFunEvals',9000);

%% RESULTS
[x,J,exitflag] = fmincon(@(x)Object_11_HXD(x,par),x0,A,b,Aeq,Beq,LB,UB,...
```

```
    @(x)HEN_Constraints_11_HXD(x,par),options);
exitflag
x=x.*par.sc.x; % Unscale variables

Tend = x(1); T1 = x(2); T2 = x(3); Th1out = x(4); Th2out = x(5);
w1 = x(6); w2 = x(7); Q1 = x(8); Q2 = x(9); UA1 = x(10); UA2 = x(11);
T0 = par.T0; Th1 = par.Th1; Th2 = par.Th2;

display(['    T1          T2        Th1out        Th2out    [degC]'])
disp([T1 T2 Th1out Th2out])
display(['   Tend [degC] = '])
disp(Tend)
w1_ratio = w1/par.w0;
w2_ratio = w2/par.w0;
display(['w1 ratio     w2 ratio        [J/K]'])
disp([w1_ratio w2_ratio])
display(['        w1          w2'])
disp([w1 w2])
display(['       UA1         UA2       [Wm2/K]'])
disp([UA1 UA2])
display(['DeltaTmin'])
display(['      Hot1        Cold1         Hot2      Cold2'])
Thot1 = Th1-T1;
Tcold1 = Th1out-T0;
Thot2 = Th2-T2;
Tcold2 = Th2out-T0;
disp([Thot1 Tcold1 Thot2 Tcold2])
display(['   Q1          Q2'])
disp([Q1 Q2])

%% RESULTS WITH VARIATION OF C0
% Defining the initial points and steps
c0 = 1;
co_end = 5;
DeltaC0 = 0.1;
n = (co_end-c0)/DeltaC0;
c0_vec = [];

% Utilizing the resulting variables
T1_vec = [];
T2_vec = [];
Tend_vec = [];
w1_vec = [];
w2_vec = [];
UA1_vec = [];
UA2_vec = [];
Th1out_vec = [];
Th2out_vec = [];
Q1_vec = [];
Q2_vec = [];
DeltaTHX1_vec = [];
DeltaTHX2_vec = [];
exitflag_vec = [];
```

```
for i=1:n;
    [x,J,exitflag] = fmincon(@(x)Object_11_HXD(x,par,c0),x0,A,b,Aeq,Beq,...
        LB,UB,@(x)HEN_Constraints_11_HXD(x,par),options);
    x=x.*par.sc.x; % Unscale variables
    exitflag

Tend = x(1); T1 = x(2); T2 = x(3); Th1out = x(4); Th2out = x(5);
w1 = x(6); w2 = x(7); Q1 = x(8); Q2 = x(9); UA1 = x(10); UA2 = x(11);

Th1 = par.Th1; Th2 = par.Th2; T0 = par.T0;

Thot1 = Th1-T1;
Tcold1 = Th1out-T0;
DeltaTHX1 = Thot1/Tcold1;

Thot2 = Th2-T2;
Tcold2 = Th2out-T0;
DeltaTHX2 = Thot2/Tcold2;

T1_vec(i)= T1;
T2_vec(i) = T2;
Tend_vec(i) = Tend;
w1_vec(i) = w1;
w2_vec(i) = w2;
UA1_vec(i) = UA1;
UA2_vec(i) = UA2;
exitflag_vec(i) = exitflag;
Th1out_vec(i) = Th1out;
Th2out_vec(i) = Th2out;
Q1_vec(i) = Q1;
Q2_vec(i) = Q2;
DeltaTHX1_vec(i) = DeltaTHX1;
DeltaTHX2_vec(i) = DeltaTHX2;
c0_vec(i) = c0;

c0 = c0+DeltaC0;

end

% Plotting the results
figure(1)
plot(c0_vec,Tend_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T_{end}')
legend('T_{end}')

figure(2)
plot(c0_vec,exitflag_vec,'LineWidth',2)
xlabel('c0')
ylabel('exitflag')

figure(3)
plot(c0_vec,w1_vec,'b','LineWidth',2)
hold on
```

```
plot(c0_vec,w2_vec,'r','LineWidth',2)
xlabel('c0 cost factor')
ylabel('w_{i} = mC_{p}')
legend('w_{1}','w_{2}')

figure(4)
subplot(2,1,1)
plot(c0_vec,T1_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T1 [^{\circ}C]')
legend('T1')

subplot(2,1,2)
plot(c0_vec,T2_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T2 [^{\circ}C]')
legend('T2')

figure(5)
subplot(2,1,1)
plot(c0_vec,Th1out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th1out [^{\circ}C]')
legend('Th1out')

subplot(2,1,2)
plot(c0_vec,Th2out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th2out [^{\circ}C]')
legend('Th2out')

figure(6)
subplot(2,1,1)
plot(c0_vec,DeltaTHX1_vec)
xlabel('c0 cost factor')
ylabel('DeltaT HX1')
legend('DeltaT HX1')

subplot(2,1,1)
plot(c0_vec,DeltaTHX2_vec)
xlabel('c0 cost factor')
ylabel('DeltaT HX2')
legend('DeltaT HX2')

figure(7)
subplot(2,1,1)
plot(c0_vec,UA1_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA1 [kWm2/K]')
legend('UA1')

subplot(2,1,2)
plot(c0_vec,UA2_vec,'LineWidth',2)
xlabel('c0 cost factor')
```

```
ylabel('UA2 [kWm2/K]')
legend('UA2')
```

`RunHEN_11_HXD_DJT.m`

```
%% Model to simulate a steady state HEN with the Jaesckhe Temperatures
%  Optimal Operation

% Topology to be investigated:

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 1                 %
%         ---------0---------       %
%    ----|                   |----  %
%         ---------0---------       %
%                 2                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clear all;
clc;

%% Parameters

par.w0  = 95;   %[J/K] w= miCpi
par.wh1 = 60;   %[J/K]
par.wh2 = 65;   %[J/K]

par.Th1 = 203;  %[degC]
par.Th2 = 248;  %[degC]
par.T0 = 130;   %[degC]

par.UA1 = 1.23; % GIVEN FROM OPTIMAL DESIGN
par.UA2 = 4.51; % GIVEN FROM OPTIMAL DESIGN

par.DeltaTmin = 0.5; %[degC]
par.c0 = 4; %[$/m2]
par.n = 0.65;

par.sc.x = [200*ones(5,1);45;45;5000*ones(2,1)];
par.sc.j = 200;

%% Optimization

% x0 = [  Tend   T1 T2    Th1out  Th2out    w1 w2    Q1 Q2]

x0   = [227.92    203.55 248.55    149.6 154.2    43.5 51.4    3203   6099]';
x0 = x0./par.sc.x;

A = []; b = []; Aeq = []; Beq = [];
LB = 0.01*ones(9,1); UB = inf*ones(9,1);
options = optimset('display','iter','MaxFunEvals',9000,...
                   'TolCon',1e-10,'TolX',1e-10);
```

```
[x,J,exitflag] = fmincon(@(x)Object_11_HXD_DJT(x,par),x0,A,b,Aeq,Beq...
                    ,LB,UB,@(x)HEN_Constraints_11_HXD_fixedA(x,par),options);
exitflag
x=x.*par.sc.x; % Unscale variables

%% RESULTS
Tend = x(1); T1 = x(2); T2 = x(3); Th1out = x(4); Th2out = x(5); w1 = x(6);
w2 = x(7); Q1 = x(8); Q2 = x(9);
T0 = par.T0; Th1 = par.Th1; Th2 = par.Th2;
UA1 = par.UA1; UA2 = par.UA2;

display(['   T1          T2        Th1out        Th2out     [degC]'])
disp([T1 T2 Th1out Th2out])
display(['   Tend  [degC] =  '])
disp(Tend)
w1_ratio = w1/par.w0;
w2_ratio = w2/par.w0;
display(['w1 ratio     w2 ratio         [J/K]'])
disp([w1_ratio w2_ratio])
display(['        w1           w2'])
disp([w1 w2])
display(['       UA1          UA2       [Wm2/K]'])
disp([UA1 UA2])
display(['DeltaTmin'])
display(['      Hot1         Cold1          Hot2      Cold2'])
Thot1 = Th1-T1;
Tcold1 = Th1out-T0;
Thot2 = Th2-T2;
Tcold2 = Th2out-T0;
disp([Thot1 Tcold1 Thot2 Tcold2])
display(['   Q1          Q2'])
disp([Q1 Q2])

%% RESULTS JAESCHKE TEMPERATURE

% x0 = [Tend T1 T2 Th1out Th2out w1 w2 Q1 Q2]
x0   = [222.967 208.31 229.56 164.53 147.64 29.48 65.52 2308 6523]';
x0 = x0./par.sc.x;

A = []; B = []; Aeq = []; Beq = []; LB = 0*ones(9,1); UB = inf*ones(9,1);
options = optimset('display','iter','MaxFunEvals',9000...
                    ,'TolCon',1e-10,'TolX',1e-10);

[xDJT,fval,exitflag] = fmincon(@(x)Object_11_HXD_DJT(x,par)...
     ,x0,A,B,Aeq,Beq,LB,UB,@(x)HEN_Constraints_11_HXD_DJT2(x,par),options);
exitflag
xDJT = xDJT.*par.sc.x;

TendDJT = xDJT(1); T1DJT = xDJT(2); T2DJT = xDJT(3);
Th1outDJT= xDJT(4); Th2outDJT = xDJT(5);
w1DJT = xDJT(6); w2DJT = xDJT(7);
Q1DJT = xDJT(8); Q2DJT = xDJT(9);
```

```
T0DJT = par.T0; Th1DJT = par.Th1; Th2DJT = par.Th2;
UA1DJT = par.UA1; UA2DJT = par.UA2;

display(['    T1_DJT             T2_DJT         Th1out_DJT          Th2out_DJT    [degC]'])
disp([T1DJT T2DJT Th1outDJT Th2outDJT])
display(['   Tend_DJT [degC] =  '])
disp(TendDJT)
w1_ratio = w1/par.w0;
w2_ratio = w2/par.w0;
% display(['w1 ratio     w2 ratio         [J/K]'])
% disp([w1_ratio w2_ratio])
display(['        w1_DJT            w2_DJT'])
disp([w1DJT w2DJT])
display(['        UA1_DJT           UA2_DJT         [Wm2/K]'])
disp([UA1DJT UA2DJT])
display(['DeltaTmin_DJT'])
display(['       Hot1         Cold1           Hot2       Cold2'])
Thot1DJT = Th1DJT-T1DJT;
Tcold1DJT = Th1outDJT-T0DJT;
Thot2DJT = Th2DJT-T2DJT;
Tcold2DJT = Th2outDJT-T0DJT;
disp([Thot1DJT Tcold1DJT Thot2DJT Tcold2DJT])
display(['   Q1_DJT           Q2_DJT'])
disp([Q1DJT Q2DJT])
```

HEN_Constraints_11_HXD.m

```
% HEN_Constraints function
% Nonlinear constraints for optimizing a HEN
% Includes mass, energy and steady state balances

function [Cineq, Res] = HEN_Constraints_11_HXD(x,par)
x=x.*par.sc.x; % Unscale variables

% States
Tend = x(1); T1 = x(2); T2 = x(3); Th1out = x(4); Th2out = x(5);
w1 = x(6); w2 = x(7); Q1 = x(8); Q2 = x(9); UA1 = x(10); UA2 = x(11);

% Parameters
w0 = par.w0; wh1 = par.wh1; wh2 = par.wh2;
Th1 = par.Th1; Th2 = par.Th2; T0 = par.T0;
DeltaTmin = par.DeltaTmin;

%% INEQUALITY CONSTRAINTS
%%HX 1
  Cineq1 = -(Th1-T1-DeltaTmin);        % HOT SIDE HX1
  Cineq2 = -(Th1out-T0-DeltaTmin);     % COLD SIDE HX1

%%HX 2
  Cineq3 = -(Th2-T2-DeltaTmin);        % HOT SIDE HX2
  Cineq4 = -(Th2out-T0-DeltaTmin);     % COLD SIDE HX2

Cineq = [Cineq1;Cineq2;Cineq3;Cineq4];
```

```
%% MODEL EQUATIONS

%AMTD Approximation
% DeltaT1 = 0.5*((Th1out-T0)+(Th1-T1));
% DeltaT2 = 0.5*((Th2out-T0)+(Th2-T2));

%Underwood Approximation
  DeltaT1 = ((((Th1out-T0)^1/3)+((Th1-T1)^1/3))/2)^3;
  DeltaT2 = ((((Th2out-T0)^1/3)+((Th2-T2)^1/3))/2)^3;

Res = [ % Upper path, 1st HX
         Q1-(w1*(T1-T0));                    % Cold Stream, w1
         Q1+(par.wh1*(Th1out-Th1));          % Hot Stream, wh1
         Q1-(UA1*DeltaT1);                   % HX Design Equation

        % Lower path, 2nd HX
         Q2-(w2*(T2-T0));                    % Cold Stream, w2
         Q2+(par.wh2*(Th2out-Th2));          % Hot Stream, wh2
         Q2-(UA2*DeltaT2);                   % HX Design Equation

        % Mass balance
         w1+w2-w0;

        % Energy balance
         (w0*Tend)-(w1*T1)-(w2*T2)];


end
```

HEN_Constraints_11_HXD_DJT2.m

```
% HEN_Constraints function for the Jaeschke Temperature
% Nonlinear constraints for optimizing a HEN
% Includes mass, energy, steady state balances and the Jaeschke Temp


function [Cineq, Res] = HEN_Constraints_11_HXD_DJT2(x,par)
x=x.*par.sc.x; % Unscale variables

% States
Tend = x(1); T1 = x(2); T2 = x(3); Th1out = x(4); Th2out = x(5);
w1 = x(6); w2 = x(7); Q1 = x(8); Q2 = x(9);

% Parameters
w0 = par.w0; wh1 = par.wh1; wh2 = par.wh2;
Th1 = par.Th1; Th2 = par.Th2; T0 = par.T0;
UA1 = par.UA1; UA2 = par.UA2;
DeltaTmin = par.DeltaTmin;

% INEQUALITY CONSTRAINTS
%%HX 1
  Cineq1 = -(Th1-T1-DeltaTmin);        % HOT SIDE HX1
  Cineq2 = -(Th1out-T0-DeltaTmin);     % COLD SIDE HX1

%%HX 2
  Cineq3 = -(Th2-T2-DeltaTmin);        % HOT SIDE HX2
  Cineq4 = -(Th2out-T0-DeltaTmin);     % COLD SIDE HX2

Cineq = [Cineq1;Cineq2;Cineq3;Cineq4];

% MODEL EQUAITONS
%AMTD Approximation
% DeltaT1 = 0.5*((Th1out-T0)+(Th1-T1));
% DeltaT2 = 0.5*((Th2out-T0)+(Th2-T2));

%Underwood Approximation
  DeltaT1 = ((((Th1out-T0)^1/3)+((Th1-T1)^1/3))/2)^3;
  DeltaT2 = ((((Th2out-T0)^1/3)+((Th2-T2)^1/3))/2)^3;

%Jaeschke Temperatures
  DJT1 = (T1-T0)^2/(Th1-T0);
  DJT2 = (T2-T0)^2/(Th2-T0);

Res = [ % Upper path, 1st HX
          Q1-(w1*(T1-T0));                 % Cold Stream, w1
          Q1+(par.wh1*(Th1out-Th1));       % Hot Stream, wh1
          Q1-(UA1*DeltaT1);                % HX Design Equation

        % Lower path, 2nd HX
          Q2-(w2*(T2-T0));                 % Cold Stream, w2
          Q2+(par.wh2*(Th2out-Th2));       % Hot Stream, wh2
          Q2-(UA2*DeltaT2);                % HX Design Equation
```

```
        % Mass balance
         w1+w2-w0;

        % Energy balance
         (w0*Tend)-(w1*T1)-(w2*T2);

         % Jaeschke constraint
         DJT1 - DJT2];


end
```

Object_11_HXD.m

```
% Object function for optimal operation

function [J] = Object_11_HXD(x,par,c0)
x=x.*par.sc.x;
UA1=x(10);
UA2=x(11);
Tend = x(1)

% Cost function to be minimizes
J = (-Tend+c0*(UA1^par.n+UA2^par.n);

end
```

D.2 Case 2

`RunHEN_21_HXD.m`

```
%% Model to simulate a steady state HEN
%  Optimal Design

% Topology to be investigated:

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              1       2             %
%          ----0-------0----         %
%    ----|                 |----     %
%          --------0--------         %
%                  3                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clear all;
clc;

%% Parameters

par.w0  = 160;    %[kW/K]  w= miCpi
par.wh1 = 60;     %[kW/K]
par.wh2 = 27;     %[kW/K]
par.wh3 = 65;     %[kW/K]

par.Th1 = 203;   %[degC]
par.Th2 = 255;   %[degC]
par.Th3 = 248;   %[degC]
par.T0 = 130;    %[degC]

par.DeltaTmin = 0.5; %[degC]
par.n = 0.65;
% par.c0 = 1.4; %[$/m2]
par.sc.x = [200*ones(7,1);45;45;5000*ones(3,1);200*ones(3,1)];
par.sc.j = 200;

%% Optimization

% x0 = [ Tend T1   T2   T3  Th1out  Th2out   Th3out     w1    w2...
Q1     Q2      Q3        UA1      UA2      UA3]

x0   = [ 204  168  199  208  154     190      145      42.5    42.5...
           2896   2323   6808        3          2          9]';

x0 = x0./par.sc.x;
A = []; b = []; Aeq = []; Beq = [];
LB = 0.00*ones(15,1); UB = inf*ones(15,1);
options = optimset('Algorithm','interior-point','display','iter'...
                           ,'MaxFunEvals',9000,'TolCon',1e-10,'TolX',1e-10);
```

```
% [x,J,exitflag] = fmincon(@(x)Object_21_HXD(x,par),x0,A,b...
%,Aeq,Beq,LB,UB,@(x)HEN_Constraints_21_HXD(x,par),options);
% exitflag
% x=x.*par.sc.x; % Unscale variables

% %% FIXED INLET TEMEPRATURE
% Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); Th1out = x(5);
%Th2out = x(6); Th3out = x(7); w1 = x(8);
% w2 = x(9); Q1 = x(10); Q2 = x(11); Q3 = x(12); UA1 = x(13);
%UA2 = x(14); UA3 = x(15); % T0 = par.T0; Th1 = par.Th1;
% Th2 = par.Th2; Th3 = par.Th3;
%
% display(['    T1          T2          T3        Th1out     Th2out     Th3out
[degC]'])
% disp([T1 T2 T3   Th1out Th2out Th3out])
% display(['  Tend [degC] = '])
% disp(Tend)
% w1_ratio = w1/par.w0;
% w2_ratio = w2/par.w0;
% display(['w1 ratio     w2 ratio        [J/K]'])
% disp([w1_ratio w2_ratio])
% display(['       w1          w2'])
% disp([w1 w2])
% display(['      UA1         UA2         UA3       [Wm2/K]'])
% disp([UA1 UA2 UA3])
% display(['DeltaTmin'])
% display(['     Hot1       Cold1        Hot2     Cold2         Hot3     Cold3'])
% Thot1 = Th1-T1;
% Tcold1 = Th1out-T0;
% Thot2 = Th2-T2;
% Tcold2 = Th2out-T1;
% Thot3 = Th3-T3;
% Tcold3 = Th3out-T0;
% disp([Thot1 Tcold1 Thot2 Tcold2 Thot3 Tcold3])
% display(['  Q1               Q2           Q3'])
% disp([Q1 Q2 Q3])

%% RESULTS WITH VARIATION OF C0
% Defining the initial points and step
c0 = 1;
co_end = 5;
DeltaC0 = 0.1;
n = (co_end-c0)/DeltaC0;
c0_vec = [];

% Utilizing the resulting variables
T1_vec = [];
T2_vec = [];
T3_vec = [];
Tend_vec = [];
w1_vec = [];
w2_vec = [];
UA1_vec = [];
UA2_vec = [];
```

```
UA3_vec = [];
Th1out_vec = [];
Th2out_vec = [];
Th3out_vec = [];
Q1_vec = [];
Q2_vec = [];
Q3_vec = [];
exitflag_vec = [];
DeltaTHX1_vec = [];
DeltaTHX2_vec = [];
DeltaTHX3_vec = [];

for i=1:n;
    [x,J,exitflag] = fmincon(@(x)Object_21_HXD(x,par,c0),x0,A,b,...
Aeq,Beq,LB,UB,@(x)HEN_Constraints_21_HXD(x,par),options);
    x=x.*par.sc.x; % Unscale variables
    exitflag

Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); Th1out = x(5);
Th2out = x(6); Th3out = x(7); w1 = x(8);
w2 = x(9); Q1 = x(10); Q2 = x(11); Q3 = x(12);
UA1 = x(13); UA2 = x(14); UA3 = x(15);


w0 = par.w0;
wh1 = par.wh1; wh2 = par.wh2; wh3 = par.wh3;
Th1 = par.Th1; Th2 = par.Th2; Th3 = par.Th3;
T0 = par.T0;

Thot1 = Th1-T1;
Tcold1 = Th1out-T0;
DeltaTHX1 = Thot1/Tcold1;

Thot2 = Th2-T2;
Tcold2 = Th2out-T1;
DeltaTHX2 = Thot2/Tcold2;

Thot3 = Th3-T3;
Tcold3 = Th3out-T0;
DeltaTHX3 = Thot3/Tcold3;

T1_vec(i)= T1;
T2_vec(i) = T2;
T3_vec(i) = T3;
Tend_vec(i) = Tend;
w1_vec(i) = w1;
w2_vec(i) = w2;
UA1_vec(i) = UA1;
UA2_vec(i) = UA2;
UA3_vec(i) = UA3;
exitflag_vec(i) = exitflag;
Th1out_vec(i) = Th1out;
Th2out_vec(i) = Th2out;
Th3out_vec(i) = Th3out;
```

```
Q1_vec(i) = Q1;
Q2_vec(i) = Q2;
Q3_vec(i) = Q3;
DeltaTHX1_vec(i) = DeltaTHX1;
DeltaTHX2_vec(i) = DeltaTHX2;
DeltaTHX3_vec(i) = DeltaTHX3;
c0_vec(i) = c0;

c0 = c0+DeltaC0;

end

figure(1)
plot(c0_vec,Tend_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T_{end}')
legend('T_{end}')

figure(2)
plot(c0_vec,exitflag_vec)
xlabel('c0')
ylabel('exitflag')

figure(3)
plot(c0_vec,w1_vec,'b','LineWidth',2)
hold on
plot(c0_vec,w2_vec,'r','LineWidth',2)
xlabel('c0 cost factor')
ylabel('w_{i} = mC_{p}')
legend('w_{1}','w_{2}')

figure(4)
subplot(3,1,1)
plot(c0_vec,T1_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T1 [^{\circ}C]')
legend('T1')

subplot(3,1,2)
plot(c0_vec,T2_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T2 [^{\circ}C]')
legend('T2')

subplot(3,1,3)
plot(c0_vec,T3_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T3 [^{\circ}C]')
legend('T3')

figure(5)
subplot(3,1,1)
plot(c0_vec,Th1out_vec,'LineWidth',2)
xlabel('c0 cost factor')
```

```
ylabel('Th1out [^{\circ}C]')
legend('Th1out')

subplot(3,1,2)
plot(c0_vec,Th2out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th2out [^{\circ}C]')
legend('Th2out')

subplot(3,1,3)
plot(c0_vec,Th3out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th3out [^{\circ}C]')
legend('Th3out')

figure(6)
subplot(3,1,1)
plot(c0_vec,DeltaTHX1_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('DeltaT HX1')
legend('DeltaT HX1')

subplot(3,1,1)
plot(c0_vec,DeltaTHX2_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('DeltaT HX2')
legend('DeltaT HX2')

figure(7)
subplot(3,1,1)
plot(c0_vec,UA1_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA1 [kWm2/K]')
legend('UA1')

subplot(3,1,2)
plot(c0_vec,UA2_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA2 [kWm2/K]')
legend('UA2')

subplot(3,1,3)
plot(c0_vec,UA3_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA3 [kWm2/K]')
legend('UA3')
```

RunHEN_21_HXD_DJT2.m

```
%% Model to simulate a steady state HEN with the Jaeschke Temperature
% Optimal Operation

% Topology to be investigated:
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                1        2                %
%            ----0------0----              %
%     ----|                    |----       %
%            ---------0--------            %
%                   3                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clear all;
clc;

%% Parameters

par.w0  = 100;    %[kW/K]  w= miCpi
par.wh1 = 60;     %[kW/K]
par.wh2 = 27;     %[kW/K]
par.wh3 = 65;     %[kW/K]

par.Th1 = 203;   %[degC]
par.Th2 = 255;   %[degC]
par.Th3 = 248;   %[degC]
par.T0 = 130;    %[degC]

par.UA1 = 1.28; % GIVEN FROM OPTIMAL DESIGN
par.UA2 = 2.85; % GIVEN FROM OPTIMAL DESIGN
par.UA3 = 7.74; % GIVEN FROM OPTIMAL DESIGN

par.DeltaTmin = 0.5; %[degC]
par.c0 = 2; % Cost Factor
par.n = 0.65; % Cost Exponent
par.sc.x = [200*ones(7,1);45;45;5000*ones(3,1)]; %Scaling
par.sc.j = 200; % Scaling

%% Optimization

% x0 = [Tend T1 T2 T3 Th1out Th2out Th3out w1 w2 Q1 Q2 Q3]
x0   = [223 182 227 222 175 202 159 42.5 42.5 1679 1432 5804]';
x0 = x0./par.sc.x;

A = []; b = []; Aeq = []; Beq = [];
LB = 0.00*ones(12,1); UB = inf*ones(12,1);
options = optimset('Algorithm','interior-point','display',...
                   'iter','MaxFunEvals',9000);


[x,J,exitflag] = fmincon(@(x)Object_21_HXD_DJT(x,par)...
    ,x0,A,b,Aeq,Beq,LB,UB,@(x)HEN_Constraints_21_HXD_DJT(x,par),options);
exitflag
x=x.*par.sc.x; % Unscale variables

%% RESULTS
Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); Th1out = x(5);
Th2out = x(6); Th3out = x(7); w1 = x(8);
```

```
w2 = x(9); Q1 = x(10); Q2 = x(11); Q3 = x(12);
T0 = par.T0; Th1 = par.Th1; Th2 = par.Th2; Th3 = par.Th3;
UA1 = par.UA1; UA2 = par.UA2; UA3 = par.UA3;

display(['    T1          T2          T3        Th1out     Th2out     Th3out
[degC]'])
disp([T1 T2 T3   Th1out Th2out Th3out])
display(['  Tend [degC] = '])
disp(Tend)
w1_ratio = w1/par.w0;
w2_ratio = w2/par.w0;
display(['w1 ratio    w2 ratio       [J/K]'])
disp([w1_ratio w2_ratio])
display(['       w1          w2'])
disp([w1 w2])
display(['      UA1        UA2        UA3      [Wm2/K]'])
disp([UA1 UA2 UA3])
display(['DeltaTmin'])
display(['     Hot1       Cold1        Hot2     Cold2        Hot3     Cold3'])
Thot1 = Th1-T1;
Tcold1 = Th1out-T0;
Thot2 = Th2-T2;
Tcold2 = Th2out-T1;
Thot3 = Th3-T3;
Tcold3 = Th3out-T0;
disp([Thot1 Tcold1 Thot2 Tcold2 Thot3 Tcold3])
display(['  Q1              Q2           Q3'])
disp([Q1 Q2 Q3])

%% RESULTS JAESCHKE TEMPERATURE

% x0 = [Tend T1 T2 T3 Th1out Th2out Th3out w1 w2 Q1 Q2]
x0   = [217 165 210 221 180 192 167 36 5 1360 1687 5228]';
x0 = x0./par.sc.x;

A = []; B = []; Aeq = []; Beq = []; LB = 0*ones(12,1); UB = inf*ones(12,1);
options = optimset('Algorithm','interior-point','display',...
                   'iter','MaxFunEvals',9000);

[xDJT,fval,exitflag] = fmincon(@(x)Object_21_HXD_DJT(x,par)...
     ,x0,A,B,Aeq,Beq,LB,UB,@(x)HEN_Constraints_21_HXD_DJT(x,par),options);
exitflag
xDJT = xDJT.*par.sc.x;

TendDJT = xDJT(1); T1DJT = xDJT(2); T2DJT = xDJT(3); T3DJT = xDJT(4);
Th1outDJT= xDJT(5); Th2outDJT = xDJT(6); Th3outDJT = xDJT(7);
w1DJT = xDJT(8); w2DJT = xDJT(9);
Q1DJT = xDJT(10); Q2DJT = xDJT(11); Q3DJT = xDJT(12);

T0DJT = par.T0; Th1DJT = par.Th1; Th2DJT = par.Th2; Th3DJT = par.Th3;
UA1DJT = par.UA1; UA2DJT = par.UA2; UA3DJT = par.UA3;

display(['    T1_DJT          T2_DJT     T3_DJT       Th1out_DJT        Th2out_DJT
Th3out_DJT    [degC]'])
```

```
disp([T1DJT T2DJT T3DJT Th1outDJT Th2outDJT Th3outDJT])
display([' Tend_DJT [degC] = '])
disp(TendDJT)
w1_ratio = w1/par.w0;
w2_ratio = w2/par.w0;
display([' w1_DJT w2_DJT'])
disp([w1DJT w2DJT])
display([' UA1_DJT UA2_DJT UA3_DJT [Wm2/K]'])
disp([UA1DJT UA2DJT UA3DJT])
display(['DeltaTmin_DJT'])
display([' Hot1 Cold1 Hot2 Cold2 Hot3 Cold3'])
Thot1DJT = Th1DJT-T1DJT;
Tcold1DJT = Th1outDJT-T0DJT;
Thot2DJT = Th2DJT-T2DJT;
Tcold2DJT = Th2outDJT-T0DJT;
Thot3DJT = Th3DJT-T3DJT;
Tcold3DJT = Th3outDJT-T0;
disp([Thot1DJT Tcold1DJT Thot2DJT Tcold2DJT Thot3DJT Tcold3DJT])
display([' Q1_DJT Q2_DJT Q3_DJT'])
disp([Q1DJT Q2DJT Q3DJT])
```

HEN_Constraints_21_HXD.m

```
% HEN_Constraints function
% Nonlinear constraints for optimizing a HEN
% Includes mass, energy and steady state balances


function [Cineq, Res] = HEN_Constraints_21_HXD(x,par)
x=x.*par.sc.x; % Unscale variables

% States
Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); Th1out = x(5);
Th2out = x(6); Th3out = x(7); w1 = x(8);
w2 = x(9); Q1 = x(10); Q2 = x(11); Q3 = x(12); UA1 = x(13);
UA2 = x(14); UA3 = x(15);

% Parameters
w0 = par.w0; wh1 = par.wh1; wh2 = par.wh2; wh3 = par.wh3;
Th1 = par.Th1; Th2 = par.Th2; Th3 = par.Th3; T0 = par.T0;
DeltaTmin = par.DeltaTmin;

%% INEQUALITY CONSTRAINTS
%%HX 1
 Cineq1 = -(Th1-T1-DeltaTmin); % HOT SIDE HX1
 Cineq2 = -(Th1out-T0-DeltaTmin); % COLD SIDE HX1

%%HX 2
 Cineq3 = -(Th2-T2-DeltaTmin); % HOT SIDE HX2
 Cineq4 = -(Th2out-T1-DeltaTmin); % COLD SIDE HX2

%%HX 3
Cineq5 = -(Th3-T3-DeltaTmin); % HOT SIDE HX3
Cineq6 = -(Th3out-T0-DeltaTmin); % COLD SIDE HX3
```

```
Cineq = [Cineq1;Cineq2;Cineq3;Cineq4;Cineq5;Cineq6];

%% MODEL EQUATIONS

%AMTD Approximation
% DeltaT1 = 0.5*((Th1out-T0)+(Th1-T1));
% DeltaT2 = 0.5*((Th2out-T0)+(Th2-T2));

%Underwood Approximation
 DeltaT1 = ((((Th1out-T0)^1/3)+((Th1-T1)^1/3))/2)^3;
 DeltaT2 = ((((Th2out-T1)^1/3)+((Th2-T2)^1/3))/2)^3;
 DeltaT3 = ((((Th3out-T0)^1/3)+((Th3-T3)^1/3))/2)^3;

 %% (DESIGN EQUATIONS) EQUALITY CONSTRAINTS
Res = [ % Upper path, 1st HX
          Q1-(w1*(T1-T0));                  % Cold Stream, w1
          Q1+(par.wh1*(Th1out-Th1));        % Hot Stream, wh1
          Q1-(UA1*DeltaT1);                 % HX Design Equation


        % Upper path, 2nd HX
          Q2-(w1*(T2-T1));                  % Cold Stream, w2
          Q2+(par.wh2*(Th2out-Th2));        % Hot Stream, wh2
          Q2-(UA2*DeltaT2);                 % HX Design Equation


        % Lower path, 3rd HX
          Q3-(w2*(T3-T0));                  % Cold Stream, w3
          Q3+(par.wh3*(Th3out-Th3));        % Hot Stream, wh2
          Q3-(UA3*DeltaT3);                 % HX Design equation

        % Mass balance
          w1+w2-w0;

        % Energy balance
          (w0*Tend)-(w1*T2)-(w2*T3)];


end
```

HEN_Constraints_21_HXD_DJT.m

```
% HEN_Constraints function for the Jaeschke Temperature
% Nonlinear constraints for optimal operation of a (2,1) HEN
% Includes mass, energy and steady state balances and the Jaeschke Temp

function [Cineq, Res] = HEN_Constraints_21_HXD_DJT(x,par)
x=x.*par.sc.x; % Unscale variables

% States
Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); Th1out = x(5);
Th2out = x(6); Th3out = x(7); w1 = x(8);
w2 = x(9); Q1 = x(10); Q2 = x(11); Q3 = x(12);
```

```
% Parameters
w0 = par.w0; wh1 = par.wh1; wh2 = par.wh2; wh3 = par.wh3;
Th1 = par.Th1; Th2 = par.Th2; Th3 = par.Th3; T0 = par.T0;
UA1 = par.UA1; UA2 = par.UA2; UA3 = par.UA3;

DeltaTmin = par.DeltaTmin;

%% INEQUALITY CONSTRAINTS
%%HX 1
  Cineq1 = -(Th1-T1-DeltaTmin); % HOT SIDE HX1
  Cineq2 = -(Th1out-T0-DeltaTmin); % COLD SIDE HX1

%%HX 2
  Cineq3 = -(Th2-T2-DeltaTmin); % HOT SIDE HX2
  Cineq4 = -(Th2out-T1-DeltaTmin); % COLD SIDE HX2

%%HX 3
 Cineq5 = -(Th3-T3-DeltaTmin); % HOT SIDE HX3
 Cineq6 = -(Th3out-T0-DeltaTmin); % COLD SIDE HX3

 Cineq = [Cineq1;Cineq2;Cineq3;Cineq4;Cineq5;Cineq6];

%% MODEL EQUATIONS
%AMTD Approximation
% DeltaT1 = 0.5*((Th1out-T0)+(Th1-T1));
% DeltaT2 = 0.5*((Th2out-T0)+(Th2-T2));
% DeltaT3 = 0.5*((Th3out-T2)+(Th3-T3));


%Underwood Approximation
  DeltaT1 = ((((Th1out-T0)^1/3)+((Th1-T1)^1/3))/2)^3;
  DeltaT2 = ((((Th2out-T1)^1/3)+((Th2-T2)^1/3))/2)^3;
  DeltaT3 = ((((Th3out-T0)^1/3)+((Th3-T3)^1/3))/2)^3;

%Jaeschke Temperatures
DJT1 = (((Th2-T2)/(Th1-T0))-1)*((T1-T0)^2/(Th2-T2))+((T2-T0)^2)/(Th2-T1);
DJT2 = ((T3-T0)^2)/(Th3-T0);

%% DESIGN EQUATION (EQUALITY CONSTRAINTS)
Res = [ % Upper path, 1st HX
          Q1-(w1*(T1-T0));                    % Cold Stream, w1
          Q1+(par.wh1*(Th1out-Th1));          % Hot Stream, wh1
          Q1-(UA1*DeltaT1);                   % HX Design Equation


        % Upper path, 2nd HX
          Q2-(w1*(T2-T1));                    % Cold Stream, w2
          Q2+(par.wh2*(Th2out-Th2));          % Hot Stream, wh2
          Q2-(UA2*DeltaT2);                   % HX Design Equation


        % Lower path, 3rd HX
          Q3-(w2*(T3-T0));                    % Cold Stream, w3
```

```
        Q3+(par.wh3*(Th3out-Th3));          % Hot Stream, wh2
        Q3-(UA3*DeltaT3);                   % HX Design equation

       % Mass balance
        w1+w2-w0;

       % Energy balance
        (w0*Tend)-(w1*T2)-(w2*T3);

       % Jaeschke
         DJT1 - DJT2];


end
```

Object_21_HXD.m

```
% Object function for the (2,1) HEN

function [J] = Object_21_HXD(x,par,c0)
x = x.*par.sc.x;
UA1 = x(13);
UA2 = x(14);
UA3 = x(15);
Tend = x(1);
J = (-Tend+c0*(UA1^par.n+UA2^par.n+UA3^par.n)); % Cost function

end
```

D.3 Case 3

`RunHEN_32_HXD.m`

```
%% Model to simulate a steady state HEN
%  Optimal Design

% Topology to be investigated:

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               1       2       3              %
%          ----0------0------0----             %
%    ----|                        |----   %
%          --------0-------0-------            %
%                   4       5                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clear all;
clc;

%% Parameters

par.w0  = 180;    %[kW/degC] w= miCpi
par.wh1 = 50;     %[kW/degC]
par.wh2 = 30;     %[kW/degC]
par.wh3 = 15;     %[kW/degC]
par.wh4 = 70;     %[kW/degC]
par.wh5 = 20;

par.Th1 = 190;  %[degC]
par.Th2 = 203;  %[degC]
par.Th3 = 220;  %[degC]
par.Th4 = 220;  %[degC]
par.Th5 = 248;  %[degC]
par.T0 = 130;   %[degC]

par.DeltaTmin = 0.00001; %[degC]
par.n = 0.65;
% par.c0 = 4; %[$/m2]
par.sc.x = [200*ones(11,1);100;100;1000*ones(5,1);5*ones(5,1)];
par.sc.j = 200;

%% Optimization

% x0 = [Tend T1 T2 T3 T4 T5 Th1out Th2out Th3out Th4out Th5out w1 w2...
%       [Q1 Q2 Q3 Q4 Q5 UA1 UA2 UA3 UA4 UA5]
x0   = [188 158 173 184 185 198 158 174 180 144 186 56 94...
        1564 854 590 5268 1233 1.52 1.90 1.59 9.72 2.18]';

x0 = x0./par.sc.x;
A = []; b = []; Aeq = []; Beq = [];
LB = 0.00*ones(23,1); UB = inf*ones(23,1);
```

```
options = optimset('Algorithm','interior-point','display','iter',...
                  'MaxFunEvals',9000,'TolCon',1e-8,'TolX',1e-8);


% [x,J,exitflag] = fmincon(@(x)Object_32_HXD(x,par),x0,A,b,Aeq,Beq,...
%                   LB,UB,@(x)HEN_Constraints_32_HXD(x,par),options);
% exitflag
% x=x.*par.sc.x; % Unscale variables

% %% RESULTS
% Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); T4 = x(5);
% T5 = x(6); Th1out = x(7); Th2out = x(8); Th3out = x(9);
% Th4out = x(10); Th5out = x(11); w1 = x(12);
% w2 = x(13); Q1 = x(14); Q2 = x(15); Q3 = x(16); Q4 = x(17);
% Q5 = x(18); UA1 = x(19); UA2 = x(20); UA3 = x(21);
% UA4 = x(22); UA5 = x(23); % T0 = par.T0; Th1 = par.Th1;
% Th2 = par.Th2; Th3 = par.Th3; Th4 = par.Th4; Th5 = par.Th5;
%
% display(['   T1          T2          T3          T4          T5      Th1out
Th2out     Th3out     Th4out     Th5out  [degC]'])
% disp([T1 T2 T3 T4 T5   Th1out Th2out Th3out Th4out Th5out])
% display(['   Tend  [degC] =  '])
% disp(Tend)
% w1_ratio = w1/par.w0;
% w2_ratio = w2/par.w0;
% display(['w1 ratio     w2 ratio        [J/K]'])
% disp([w1_ratio w2_ratio])
% display(['        w1          w2'])
% disp([w1 w2])
% display(['      UA1        UA2        UA3        UA4        UA5        [Wm2/K]'])
% disp([UA1 UA2 UA3 UA4 UA5])
% display(['DeltaTmin'])
% display(['      Hot1        Cold1          Hot2      Cold2          Hot3      Cold3'])
% Thot1 = Th1-T1;
% Tcold1 = Th1out-T0;
% Thot2 = Th2-T2;
% Tcold2 = Th2out-T1;
% Thot3 = Th3-T3;
% Tcold3 = Th3out-T0;
% disp([Thot1 Tcold1 Thot2 Tcold2 Thot3 Tcold3])
% display(['      Hot4        Cold4          Hot5      Cold5'])
% Thot4 = Th4-T4;
% Tcold4 = Th4out-T0;
% Thot5 = Th5-T5;
% Tcold5 = Th5out-T4;
% disp([Thot4 Tcold4 Thot5 Tcold5])
% display(['   Q1              Q2          Q3          Q4          Q5'])
% disp([Q1 Q2 Q3 Q4 Q5])

%% VARIATION AND TRENDS
% Defining the initial point and step
c0 = 1;
co_end = 5;
DeltaC0 = 0.1;
```

```
n = (co_end-c0)/DeltaC0;
c0_vec = [];

% Utilizing the resulting variables
T1_vec = [];
T2_vec = [];
T3_vec = [];
T4_vec = [];
T5_vec = [];
Tend_vec = [];
w1_vec = [];
w2_vec = [];
UA1_vec = [];
UA2_vec = [];
UA3_vec = [];
UA4_vec = [];
UA5_vec = [];
Th1out_vec = [];
Th2out_vec = [];
Th3out_vec = [];
Th4out_vec = [];
Th5out_vec = [];
Q1_vec = [];
Q2_vec = [];
Q3_vec = [];
Q4_vec = [];
Q5_vec = [];
exitflag_vec = [];
DeltaTHX1_vec = [];
DeltaTHX2_vec = [];
DeltaTHX3_vec = [];
DeltaTHX4_vec = [];
DeltaTHX5_vec = [];

for i=1:n;

    [x,J,exitflag] = fmincon(@(x)Object_32_HXD(x,par,c0),...
        x0,A,b,Aeq,Beq,LB,UB,@(x)HEN_Constraints_32_HXD(x,par),options);
        x=x.*par.sc.x; % Unscale variables
exitflag

% States
Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); T4 = x(5); T5 = x(6);
Th1out = x(7); Th2out = x(8); Th3out = x(9); Th4out = x(10);
Th5out = x(11); w1 = x(12); w2 = x(13); Q1 = x(14); Q2 = x(15);
Q3 = x(16); Q4 = x(17); Q5 = x(18); UA1 = x(19); UA2 = x(20);
UA3 = x(21); UA4 = x(22); UA5 = x(23); T0 = par.T0;
Th1 = par.Th1; Th2 = par.Th2; Th3 = par.Th3; Th4 = par.Th4; Th5 = par.Th5;

% Parameters
w0 = par.w0;
wh1 = par.wh1; wh2 = par.wh2; wh3 = par.wh3;

% Calculating the DeltaT's for each HX
```

```
Thot1 = Th1-T1;
Tcold1 = Th1out-T0;
DeltaTHX1 = Thot1/Tcold1;

Thot2 = Th2-T2;
Tcold2 = Th2out-T1;
DeltaTHX2 = Thot2/Tcold2;

Thot3 = Th3-T3;
Tcold3 = Th3out-T0;
DeltaTHX3 = Thot3/Tcold3;

Thot4 = Th4-T4;
Tcold4 = Th4out-T0;
DeltaTHX4 = Thot4/Tcold4;

Thot5 = Th5-T5;
Tcold5 = Th5out-T4;
DeltaTHX5 = Thot5/Tcold5;

% Inserting the calulcated value in the solution matrices
T1_vec(i)= T1;
T2_vec(i) = T2;
T3_vec(i) = T3;
T4_vec(i) = T4;
T5_vec(i) = T5;
Tend_vec(i) = Tend;
w1_vec(i) = w1;
w2_vec(i) = w2;
UA1_vec(i) = UA1;
UA2_vec(i) = UA2;
UA3_vec(i) = UA3;
UA4_vec(i) = UA4;
UA5_vec(i) = UA5,
exitflag_vec(i) = exitflag;
Th1out_vec(i) = Th1out;
Th2out_vec(i) = Th2out;
Th3out_vec(i) = Th3out;
Th4out_vec(i) = Th4out;
Th5out_vec(i) = Th5out;
Q1_vec(i) = Q1;
Q2_vec(i) = Q2;
Q3_vec(i) = Q3;
Q4_vec(i) = Q4;
Q5_vec(i) = Q5;
DeltaTHX1_vec(i) = DeltaTHX1;
DeltaTHX2_vec(i) = DeltaTHX2;
DeltaTHX3_vec(i) = DeltaTHX3;
DeltaTHX4_vec(i) = DeltaTHX4;
DeltaTHX5_vec(i) = DeltaTHX5;
c0_vec(i) = c0;

c0 = c0+DeltaC0;
```

```
end

% Plotting the results
figure(1)
plot(c0_vec,Tend_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T_{end}')
legend('T_{end}')

figure(2)
plot(c0_vec,exitflag_vec)
xlabel('c0')
ylabel('exitflag')

figure(3)
plot(c0_vec,w1_vec,'b','LineWidth',2)
hold on
plot(c0_vec,w2_vec,'r','LineWidth',2)
xlabel('c0 cost factor')
ylabel('w_{i} = mC_{p}')
legend('w_{1}','w_{2}')

figure(4)
subplot(5,1,1)
plot(c0_vec,T1_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T1 [^{\circ}C]')
legend('T1')

subplot(5,1,2)
plot(c0_vec,T2_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T2 [^{\circ}C]')
legend('T2')

subplot(5,1,3)
plot(c0_vec,T3_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T3 [^{\circ}C]')
legend('T3')

subplot(5,1,4)
plot(c0_vec,T4_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T4 [^{\circ}C]')
legend('T4')

subplot(5,1,5)
plot(c0_vec,T5_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('T5 [^{\circ}C]')
legend('T5')

figure(5)
```

```
subplot(5,1,1)
plot(c0_vec,Th1out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th1out [^{\circ}C]')
legend('Th1out')

subplot(5,1,2)
plot(c0_vec,Th2out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th2out [^{\circ}C]')
legend('Th2out')

subplot(5,1,3)
plot(c0_vec,Th3out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th3out [^{\circ}C]')
legend('Th3out')

subplot(5,1,4)
plot(c0_vec,Th4out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th4out [^{\circ}C]')
legend('Th4out')

subplot(5,1,5)
plot(c0_vec,Th5out_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('Th5out [^{\circ}C]')
legend('Th5out')

figure(6)
subplot(5,1,1)
plot(c0_vec,DeltaTHX1_vec)
xlabel('c0 cost factor')
ylabel('DeltaT HX1')
legend('DeltaT HX1')

subplot(5,1,1)
plot(c0_vec,DeltaTHX2_vec)
xlabel('c0 cost factor')
ylabel('DeltaT HX2')
legend('DeltaT HX2')

figure(7)
subplot(5,1,1)
plot(c0_vec,UA1_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA1 [kWm2/K]')
legend('UA1')

subplot(5,1,2)
plot(c0_vec,UA2_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA2 [kWm2/K]')
```

```
legend('UA2')

subplot(5,1,3)
plot(c0_vec,UA3_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA3 [kWm2/K]')
legend('UA3')

subplot(5,1,4)
plot(c0_vec,UA4_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA4 [kWm2/K]')
legend('UA4')

subplot(5,1,5)
plot(c0_vec,UA5_vec,'LineWidth',2)
xlabel('c0 cost factor')
ylabel('UA5 [kWm2/K]')
legend('UA5')
```

RunHEN_32_HXD_DJT.m

```
%% Model to simulate a steady state HEN with the Jaeschke Temperature
%  Optimal Operation

% Topology to be investigated:

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              1        2        3             %
%          -----0------0------0-----           %
%   -----|                          |-----     %
%          ---------0---------0-------         %
%                   4         5                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clear all;
clc;

%% Parameters

par.w0  = 150;    %[kW/degC] w= miCpi
par.wh1 = 50;     %[kW/degC]
par.wh2 = 30;     %[kW/degC]
par.wh3 = 15;     %[kW/degC]
par.wh4 = 70;     %[kW/degC]
par.wh5 = 20;     %[kW/degC]

par.Th1 = 205;  %[degC]
par.Th2 = 203;  %[degC]
par.Th3 = 220;  %[degC]
par.Th4 = 220;  %[degC]
par.Th5 = 248;  %[degC]
par.T0 = 130;   %[degC]
```

```
par.UA1 = 0.06; % GIVEN FROM OPTIMAL DESIGN [kWm2/degC]
par.UA2 = 0.31; % GIVEN FROM OPTIMAL DESIGN [kWm2/degC]
par.UA3 = 0.29; % GIVEN FROM OPTIMAL DESIGN [kWm2/degC]
par.UA4 = 2.29; % GIVEN FROM OPTIMAL DESIGN [kWm2/degC]
par.UA5 = 0.74; % GIVEN FROM OPTIMAL DESIGN [kWm2/degC]

par.DeltaTmin = 0.2; %[degC]
par.n = 0.65; % Cost exponent
par.c0 = 4; % Cost factor
par.sc.x = [200*ones(11,1);100;100;1000*ones(5,1)];


%% Optimization


% x0 = [Tend T1 T2 T3 T4 T5 Th1out Th2out Th3out Th4out Th5out...
%       [w1 w2 Q1 Q2 Q3 Q4 Q5]

% Scenario 180/4
% x0    = [  175       153     168      178      172     185          165
177       182           144          173          80    80       1550    790    518
5102     1519]';

% Scenario 150/4
% x0    = [  180       139     161      176      169     181          183
178       188           154          176          33    117       314    737    472
4574     1422]';

% Scenario 150/2
x0    = [187 151 167 179 178 192 168 176 182 149 180...
          75 75 1077 783 556 4911 1357]';

x0 = x0./par.sc.x;

A = []; b = []; Aeq = []; Beq = []
; LB = 0.00*ones(18,1); UB = inf*ones(18,1);

options = optimset('Algorithm','interior-point','display',...
    'iter','MaxFunEvals',9000,'TolCon',1e-12,'TolX',1e-12);

[x,J,exitflag] = fmincon(@(x)Object_32_HXD_DJT(x,par),...
    x0,A,b,Aeq,Beq,LB,UB,@(x)HEN_Constraints_32_HXD_DJT(x,par),options);
exitflag
x=x.*par.sc.x; % Unscale variables

%% RESULTS
% States
Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); T4 = x(5);
T5 = x(6); Th1out = x(7); Th2out = x(8); Th3out = x(9);
Th4out = x(10); Th5out = x(11); w1 = x(12); w2 = x(13);
Q1 = x(14); Q2 = x(15); Q3 = x(16); Q4 = x(17); Q5 = x(18);

% Parameters
```

```
T0 = par.T0; Th1 = par.Th1; Th2 = par.Th2; Th3 = par.Th3;
Th4 = par.Th4; Th5 = par.Th5; UA1 = par.UA1;
UA2 = par.UA2; UA3 = par.UA3; UA4 = par.UA4; UA5 = par.UA5;

display(['    T1          T2          T3          T4          T5     Th1out
Th2out     Th3out     Th4out     Th5out  [degC]'])
disp([T1 T2 T3 T4 T5   Th1out Th2out Th3out Th4out Th5out])
display(['  Tend  [degC] =  '])
disp(Tend)
w1_ratio = w1/par.w0;
w2_ratio = w2/par.w0;
display(['w1 ratio     w2 ratio        [J/K]'])
disp([w1_ratio w2_ratio])
display(['        w1          w2'])
disp([w1 w2])
display(['       UA1         UA2         UA3         UA4         UA5         [Wm2/K]'])
disp([UA1 UA2 UA3 UA4 UA5])
display(['DeltaTmin'])
display(['      Hot1        Cold1         Hot2      Cold2         Hot3      Cold3'])
Thot1 = Th1-T1;
Tcold1 = Th1out-T0;
Thot2 = Th2-T2;
Tcold2 = Th2out-T1;
Thot3 = Th3-T3;
Tcold3 = Th3out-T0;
disp([Thot1 Tcold1 Thot2 Tcold2 Thot3 Tcold3])
display(['      Hot4        Cold4         Hot5      Cold5'])
Thot4 = Th4-T4;
Tcold4 = Th4out-T0;
Thot5 = Th5-T5;
Tcold5 = Th5out-T4;
disp([Thot4 Tcold4 Thot5 Tcold5])
display(['  Q1              Q2          Q3          Q4         Q5'])
disp([Q1 Q2 Q3 Q4 Q5])


%% RESULTS JAESCHKE TEMPERATURE

% x0 = [Tend T1 T2 T3 Th1out Th2out Th3out w1 w2 Q1 Q2]
x0   = [182 150 164 174 173 186 165 176 182 145 170...
          53 124 1321 828 562 5190 1534]';

x0 = x0./par.sc.x;

A = []; B = []; Aeq = []; Beq = []; LB = 0*ones(18,1); UB = inf*ones(18,1);
options = optimset('Algorithm','interior-point','display',...
                   'iter','MaxFunEvals',9000);

[xDJT,fval,exitflag] = fmincon(@(x)Object_32_HXD_DJT(x,par),...
    x0,A,B,Aeq,Beq,LB,UB,@(x)HEN_Constraints_32_HXD_DJT(x,par),options);
exitflag
xDJT = xDJT.*par.sc.x;

TendDJT = xDJT(1); T1DJT = xDJT(2); T2DJT = xDJT(3); T3DJT = xDJT(4);
```

```
T4DJT = xDJT(5); T5DJT = xDJT(6);

Th1outDJT= xDJT(7); Th2outDJT = xDJT(8); Th3outDJT = xDJT(9);
Th4outDJT = xDJT(10); Th5outDJT = xDJT(11);

w1DJT = xDJT(12); w2DJT = xDJT(13);
Q1DJT = xDJT(14); Q2DJT = xDJT(15); Q3DJT = xDJT(16);
Q4DJT = xDJT(17); Q5DJT = xDJT(18);

T0DJT = par.T0; Th1DJT = par.Th1; Th2DJT = par.Th2; Th3DJT = par.Th3;
Th4DJT = par.Th4; Th5DJT = par.Th5;
UA1DJT = par.UA1; UA2DJT = par.UA2; UA3DJT = par.UA3;
UA4DJT = par.UA4; UA5DJT = par.UA5;

% Displaying the results
display(['    T1_DJT          T2_DJT     T3_DJT        T4_DJT        T5_DJT
Th1out_DJT         Th2out_DJT    Th3out_DJT        Th4out_DJT        Th5out_DJT
[degC]'])
disp([T1DJT T2DJT T3DJT T4DJT T5DJT Th1outDJT Th2outDJT Th3outDJT Th4outDJT Th5outDJT
display(['   Tend_DJT [degC] = '])
disp(TendDJT)
w1_ratio = w1/par.w0;
w2_ratio = w2/par.w0;
display(['        w1_DJT           w2_DJT'])
disp([w1DJT w2DJT])
display(['       UA1_DJT          UA2_DJT      UA3_DJT      UA4_DJT      UA5_DJT
[Wm2/K]'])

disp([UA1DJT UA2DJT UA3DJT UA4DJT UA5DJT])
display(['DeltaTmin_DJT'])
display(['      Hot1        Cold1          Hot2      Cold2       Hot3      Cold3'])
Thot1DJT = Th1DJT-T1DJT;
Tcold1DJT = Th1outDJT-T0DJT;
Thot2DJT = Th2DJT-T2DJT;
Tcold2DJT = Th2outDJT-T0DJT;
Thot3DJT = Th3DJT-T3DJT;
Tcold3DJT = Th3outDJT-T0;
disp([Thot1DJT Tcold1DJT Thot2DJT Tcold2DJT Thot3DJT Tcold3DJT])
display(['   Q1_DJT          Q2_DJT        Q3_DJT'])
disp([Q1DJT Q2DJT Q3DJT])
```

HEN_Constraints_32_HXD.m

```
% HEN_Constraints function
% Nonlinear constraints for optimizing a (3,2) HEN
% Includes mass, energy and steady state balances

function [Cineq, Res] = HEN_Constraints_32_HXD(x,par)
x=x.*par.sc.x; % Unscale variables

% States
Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); T4 = x(5);
T5 = x(6); Th1out = x(7); Th2out = x(8); Th3out = x(9);
Th4out = x(10); Th5out = x(11); w1 = x(12);
```

```
w2 = x(13); Q1 = x(14); Q2 = x(15); Q3 = x(16); Q4 = x(17);
Q5 = x(18); UA1 = x(19); UA2 = x(20); UA3 = x(21); UA4 = x(22);
UA5 = x(23);

% Parameters
w0 = par.w0;
wh1 = par.wh1; wh2 = par.wh2; wh3 = par.wh3; wh4 = par.wh4; wh5 = par.wh5;
Th1 = par.Th1; Th2 = par.Th2; Th3 = par.Th3; Th4 = par.Th4; Th5 = par.Th5;
T0 = par.T0;

DeltaTmin = par.DeltaTmin;


%% INEQUALITY CONSTRAINTS
%%HX 1
  Cineq1 = -(Th1-T1-DeltaTmin); % HOT SIDE HX1
  Cineq2 = -(Th1out-T0-DeltaTmin); % COLD SIDE HX1

%%HX 2
  Cineq3 = -(Th2-T2-DeltaTmin); % HOT SIDE HX2
  Cineq4 = -(Th2out-T1-DeltaTmin); % COLD SIDE HX2

%%HX 3
 Cineq5 = -(Th3-T3-DeltaTmin); % HOT SIDE HX3
 Cineq6 = -(Th3out-T2-DeltaTmin); % COLD SIDE HX3

%%HX4
 Cineq7 = -(Th4-T4-DeltaTmin); % HOT SIDE HX4
 Cineq8 = -(Th4out-T0-DeltaTmin); % COLD SIDE HX4

%%HX5
 Cineq9 = -(Th5-T5-DeltaTmin); % HOT SIDE HX5
 Cineq10 = -(Th5out-T4-DeltaTmin); % COLD SIDE HX5

 Cineq = [Cineq1;Cineq2;Cineq3;Cineq4;Cineq5;Cineq6;Cineq7;Cineq8;...
          Cineq9;Cineq10];

%% MODEL EQUATIONS

%Underwood Approximation
  DeltaT1 = ((((Th1out-T0)^1/3)+((Th1-T1)^1/3))/2)^3;
  DeltaT2 = ((((Th2out-T1)^1/3)+((Th2-T2)^1/3))/2)^3;
  DeltaT3 = ((((Th3out-T2)^1/3)+((Th3-T3)^1/3))/2)^3;
  DeltaT4 = ((((Th4out-T0)^1/3)+((Th4-T4)^1/3))/2)^3;
  DeltaT5 = ((((Th5out-T4)^1/3)+((Th5-T5)^1/3))/2)^3;

 %% DESIGN EQUATIONS (EQUALITY CONSTRAINTS)
 Res = [ % Upper path, 1st HX
         Q1-(w1*(T1-T0));                  % Cold Stream, w1
         Q1+(par.wh1*(Th1out-Th1));        % Hot Stream, wh1
         Q1-(UA1*DeltaT1);                 % HX Design Equation


       % Upper path, 2nd HX
```

```
        Q2-(w1*(T2-T1));                    % Cold Stream, w1
        Q2+(par.wh2*(Th2out-Th2));          % Hot Stream, wh2
        Q2-(UA2*DeltaT2);                   % HX Design Equation


      % Upper path, 3rd HX
        Q3-(w1*(T3-T2));                    % Cold Stream, w1
        Q3+(par.wh3*(Th3out-Th3));          % Hot Stream, wh3
        Q3-(UA3*DeltaT3);                   % HX Design equation

      % Lower path, 4th HX
        Q4-(w2*(T4-T0));                    % Cold stream, w2
        Q4+(par.wh4*(Th4out-Th4));          % Hot stream, wh4
        Q4-(UA4*DeltaT4);                   % HX design equation

      % Lower path, 5th HX
        Q5-(w2*(T5-T4));                    % Cold stream, w2
        Q5+(par.wh5*(Th5out-Th5));          % Hot stream, wh5
        Q5-(UA5*DeltaT5);                   % HX design equation

      % Mass balance
        w1+w2-w0;

      % Energy balance
        (w0*Tend)-(w1*T3)-(w2*T5)];


end
```

HEN_Constraints_32_HXD_DJT.m

```
% HEN_Constraints function for the Jaeschke Temperature
% Nonlinear constraints for optimizing a HEN
% Includes mass, energy and steady state balances and the Jaeschke temp

function [Cineq, Res] = HEN_Constraints_32_HXD_DJT(x,par)
x=x.*par.sc.x; % Unscale variables

% States
Tend = x(1); T1 = x(2); T2 = x(3); T3 = x(4); T4 = x(5);
T5 = x(6); Th1out = x(7); Th2out = x(8); Th3out = x(9);
Th4out = x(10); Th5out = x(11); w1 = x(12); w2 = x(13);
Q1 = x(14); Q2 = x(15); Q3 = x(16); Q4 = x(17); Q5 = x(18);

% Parameters
w0 = par.w0;
wh1 = par.wh1; wh2 = par.wh2; wh3 = par.wh3; wh4 = par.wh4; wh5 = par.wh5;
Th1 = par.Th1; Th2 = par.Th2; Th3 = par.Th3; Th4 = par.Th4; Th5 = par.Th5;
T0 = par.T0;
UA1 = par.UA1; UA2 = par.UA2; UA3 = par.UA3; UA4 = par.UA4; UA5 = par.UA5;

DeltaTmin = par.DeltaTmin;

%% INEQUALITY CONSTRAINTS
```

```
%%HX 1
  Cineq1 = -(Th1-T1-DeltaTmin); % HOT SIDE HX1
  Cineq2 = -(Th1out-T0-DeltaTmin); % COLD SIDE HX1

%%HX 2
  Cineq3 = -(Th2-T2-DeltaTmin); % HOT SIDE HX2
  Cineq4 = -(Th2out-T1-DeltaTmin); % COLD SIDE HX2

%%HX 3
 Cineq5 = -(Th3-T3-DeltaTmin); % HOT SIDE HX3
 Cineq6 = -(Th3out-T2-DeltaTmin); % COLD SIDE HX3

%%HX4
 Cineq7 = -(Th4-T4-DeltaTmin); % HOT SIDE HX4
 Cineq8 = -(Th4out-T0-DeltaTmin); % COLD SIDE HX4

%%HX5
 Cineq9 = -(Th5-T5-DeltaTmin); % HOT SIDE HX5
 Cineq10 = -(Th5out-T4-DeltaTmin); % COLD SIDE HX5

 Cineq = [Cineq1;Cineq2;Cineq3;Cineq4;Cineq5;Cineq6;Cineq7;Cineq8;..
          Cineq9;Cineq10];

%% MODEL EQUATIONS

%Underwood Approximation
  DeltaT1 = ((((Th1out-T0)^1/3)+((Th1-T1)^1/3))/2)^3;
  DeltaT2 = ((((Th2out-T1)^1/3)+((Th2-T2)^1/3))/2)^3;
  DeltaT3 = ((((Th3out-T2)^1/3)+((Th3-T3)^1/3))/2)^3;
  DeltaT4 = ((((Th4out-T0)^1/3)+((Th4-T4)^1/3))/2)^3;
  DeltaT5 = ((((Th5out-T4)^1/3)+((Th5-T5)^1/3))/2)^3;

 %Jaesckhe Temperatures
 DJT1 = ((T1-T0)^2*((T2-T0)*(Th3-T0)-(T3-T0)*(T2-T0)+(Th1-T0)*...
      (Th3-T0)-(Th2-T0)*(Th3-T0)+(T3-T0)*(Th2-T0)-(Th1-T0)*(T3-T0))+...
      (((T2-T0)^2)*(Th1-T0)*((T3-T0)-(T1-T0)-(Th3-T0)+(Th2-T0)))+...
      ((T3-T0)^2)*(Th1-T0)*((T1-T0)-(Th2-T0)))/((Th1-T0)*...
      ((-(Th2-T0)*(Th3-T0))-(T1-T0)*(T2-T0)+(T1-T0)*...
      (Th3-T0)+(Th2-T0)*(T2-T0)));
 DJT2 = (((Th5-T4)/(Th4-T0))-1)*((T4-T0)^2/(Th5-T4))+((T5-T0)^2)/(Th5-T4);

 %% DESIGN EQUATIONS (EQUALITY CONSTRAINTS)
 Res = [ % Upper path, 1st HX
          Q1-(w1*(T1-T0));                 % Cold Stream, w1
          Q1+(par.wh1*(Th1out-Th1));       % Hot Stream, wh1
          Q1-(UA1*DeltaT1);                % HX Design Equation


        % Upper path, 2nd HX
          Q2-(w1*(T2-T1));                 % Cold Stream, w1
          Q2+(par.wh2*(Th2out-Th2));       % Hot Stream, wh2
          Q2-(UA2*DeltaT2);                % HX Design Equation
```

```
    % Upper path, 3rd HX
     Q3-(w1*(T3-T2));                  % Cold Stream, w1
     Q3+(par.wh3*(Th3out-Th3));        % Hot Stream, wh3
     Q3-(UA3*DeltaT3);                 % HX Design equation

    % Lower path, 4th HX
     Q4-(w2*(T4-T0));                  % Cold stream, w2
     Q4+(par.wh4*(Th4out-Th4));        % Hot stream, wh4
     Q4-(UA4*DeltaT4);                 % HX design equation

    % Lower path, 5th HX
     Q5-(w2*(T5-T4));                  % Cold stream, w2
     Q5+(par.wh5*(Th5out-Th5));        % Hot stream, wh5
     Q5-(UA5*DeltaT5);                 % HX design equation

    % Mass balance
     (w1+w2-w0);

    % Energy balance
     (w0*Tend)-(w1*T3)-(w2*T5);

    % Jaeschke temperatures
      (DJT2 - DJT1)];

end
```

Object_32_HXD.m

```
% Object function for the (3,2) HEN

function [J] = Object_32_HXD(x,par,c0)
x = x.*par.sc.x;
UA1 = x(19);
UA2 = x(20);
UA3 = x(21);
UA4 = x(22);
UA5 = x(23);
Tend = x(1);

% Cost function
J = (-Tend+c0*(UA1^par.n+UA2^par.n+UA3^par.n+UA4^par.n+UA5^par.n));

end
```