Department of Chemical Engineering

Development of Systematic Technique for Energy and Property Integration in Batch Processes

Yeo Wan Sieng

This thesis is presented for the Degree of
Master of Philosophy
of
Curtin University

November 2013

Declaration

"To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university."

Signature: __________________

Name : YEO WAN SIENG

Date : 18th November 2013

To my beloved parents, brother, sisters and friends

Acknowledgement

Here, I would like to acknowledge counsel and valuable advice from my supervisor, Prof. Dr. Yudi Samyudia, Deputy Pro Vice-Chancellor, Curtin University, Sarawak Campus, for he has been very helpful in every way during the writing of this thesis. His support and guidance has made the time spent on this research work meaningful.

I extend my appreciation to Dr. Tan Yin Ling for her assistance and valuable advice on the information of my research. Without her motivation and helps, this research would not be able to finish on time.

My sincere thanks are also directed to Alex Goh Kwang Leng, Bridgid Chin Lai Fui and Regina for their kind assistance in giving supports and opinion during the proceeding of doing this research.

Finally, I would like to thank all those people around me including my family members and friends who have directly and indirectly contributed to the success of this research.

Abstract

The increasing consumption of energy, generation of waste as well as higher cost of fresh resources and waste treatment systems are the important driving forces for developing efficient, environmentally friendly and economic resource conservation techniques in the process industries. Process integration is being recognized as an useful systematic strategy for resource conservation and waste minimization. Up to date, less research works have been investigated on heat and property integration and these works are only focused on continuous processes. Since the application of batch processes is increasingly popular due to the development of technology-intensive industries such as pharmacy, fine chemistry and foods, it is necessary to consider both heat and property integration in batch processes simultaneously. In this thesis, a new mixed integer nonlinear programming (MINLP) mathematical model is introduced to synthesize a property-based heat integrated resource conservation networks (HIRCNs) for batch processes. A source-HEN-sink superstructure is constructed to embed all possible network configurations. Then, an MINLP model that consists of property-based resource conservation network (RCN) and heat exchanger network (HEN) models is developed. In the proposed model, the property-based RCN model is formulated based on supertargeting approach while HEN model is formulated via automated targeting method (ATM). The optimization objective is to minimize total annualized cost (TAC) for a batch process system. This includes the operating cost of fresh resources, hot and cold utilities as well as the capital cost of storage tanks. To demonstrate the proposed approach, three case studies were solved. Based on the optimized results, the proposed simultaneous targeting approach for property-based HIRCNs is more effective in term of TAC for HIRCNs than the presented sequential targeting approach.

List of Publication

W.S. Yeo, Y.L. Tan and Y. Samyudia. 2012. "Development of systematic technique for energy and property integration." *4^{th} International Conference on Chemical and Bioprocess Engineering, Kota Kinabalu, Sabah, 21 – 23 November 2012.*

W.S. Yeo, Y.L. Tan and Y. Samyudia. 2013. "Synthesis of Heat and Property Integration in Batch Processes." *Postgraduate colloquium for Environmental Research 2013, Genting Highland, Pahang, 28 – 29 June 2013.*

Table of Contents

List of Tables

List of Figures

Nomenclature

Parameters

A_0, A_1	Cost parameters for storage tank *s*
A_s	Annual fractional interest rate
C_m	Cost for fresh resources
C_{HU}	Cost for hot utility
C_{CU}	Cost for cold utility
Cp_i	Specific heat capacity of process source *i*
Cp_m	Specific heat capacity of fresh resource *m*
Cp_s	Specific heat capacity of the stored source
d	Cost parameter for storage tank *s*
F_i	Total amount of process source *i* in each batch cycle
F_j	Total amount of required sources that is needed for the process sink *j*
I	Chemical engineering plant cost index in the present time
I_b	Base cost index
N_b	Number of batches per year
NINTERVALS	Total number of time intervals

NSOURCES	Total number of process sources
NSINKS	Total number of process sinks
NSTORAGES	Total number of storage tanks
NFRESHS	Total number of fresh resources
T_i	Temperature of process source *i*
T_j	Temperature of process sink *j*
T_m	Temperature of fresh resource *m*
T_{Hot}^{ST}	Supply temperature for the hot stream
T_{Hot}^{TT}	Target temperature for the hot stream
T_{Cold}^{ST}	Supply temperature for the cold stream
T_{Cold}^{TT}	Target temperature for the cold stream
$t_{i,ST}$	Start time of process source *i*
$t_{i,ET}$	End time of process source *i*
$t_{j,ST}$	Start time of process sink *j*
$t_{j,ET}$	End time of process sink *j*
yr	Time life of the storage tanks

Variables

AF	Annualizing factor
C_s	Annualised capital cost for storage tanks

CST_s	Capacity of storage tank *s*
$CST_{s,t-1}$	Cumulative mass in the storage tank *s* in time interval *t*-1
$CST_{s,t}$	Cumulative mass in the storage tank *s* in time interval *t*
F_m	Total amount of fresh resource *m* which is required per each batch cycle
F^{waste}	Total amount of waste in each batch cycle
$F_{i,t}$	Fractional amount of process source *i* in time interval *t*
$F_{j,t}$	Fractional amount of process sink *j* in time interval *t*
$F_{i,j,t}$	Fractional amount of process source *i* that is sent to process sink *j* in time interval *t*
$F_{i,s,t}$	Fractional amount of process source *i* that is sent to the respective storage tank *s* in time interval *t*
$F_{i,s,t-1}$	Fractional amount of process source *i* which is sent to the respective storage tank *s* in time interval *t*-1
$F_{i,t}^{\text{waste}}$	Fractional amount of process source *i* that is sent to waste storage in time interval *t*
$F_{s,j,t}$	Amount from storage tank *s* that is sent to process sink *j* in time interval *t*
$F_{m,j,t}$	Amount from fresh resource *m* that is sent to process sink *j* in time interval *t*
$F_{s,t}^{\text{waste}}$	Amount from storage tank *s* that is sent to waste storage in time interval *t*

$H_{k,t}$ — Residual heat load within temperature interval k in time interval t

$H_{k-1,t}$ — Residual heat load within temperature interval k-1 in time interval t

$H_{0,t}$ — Heat load at the first temperature level in time interval t

$H_{n,t}$ — Heat load at the final temperature level in time interval t

$H_{i,j,k,t}$ — Heat load for the streams from process source i that are sent to process sink j within temperature interval k in time interval t

$H_{m,j,k,t}$ — Heat load for the streams from fresh resource m that are sent to process sink j within temperature interval k in time interval t

$H_{i,k,t}^{\text{waste}}$ — Heat load for the streams from process source i that are sent to waste storage within temperature interval k in time interval t

$H_{s,j,k,t}$ — Heat load for the streams from storage tank s that are sent to process sink j within temperature interval k in time interval t

$H_{s,k,t}^{\text{waste}}$ — Heat load for the streams from storage tank s that are sent to waste storage within temperature interval k in time interval t

$q_{\text{Hot}}^{\text{ST}}$ — Shifted supply temperature for the hot stream

$q_{\text{Hot}}^{\text{TT}}$ — Shifted target temperature for the hot stream

$q_{\text{Cold}}^{\text{ST}}$ — Shifted supply temperature for the cold stream

$q_{\text{Cold}}^{\text{TT}}$ — Shifted target temperature for the cold stream

t_t — End time of time interval t

t_{t-1} — Start time of time interval t

q_k	Temperature at upper limit of temperature interval k
q_{k-1}	Temperature at lower limit of temperature interval k
q_n	Lowest temperature
q_0	Highest temperature
Q_{H}	Total required external hot utility in each batch cycle
Q_{C}	Total required external cold utility in each batch cycle
$Q_{\mathrm{H},t}$	External hot utility that is required in time interval t
$Q_{\mathrm{C},t}$	External cold utility that is required in time interval t
x_l	Fractional distribution of stream l of the total mixture flow rate
$X_{i,t}, Y_{i,t}, Z_{i,t}$	Binary variable for process source i in time interval t
$X_{j,t}, Y_{j,t}, Z_{j,t}$	Binary variable for process sink j in time interval t

Indices and sets

$i \in I$	Process sources
$j \in J$	Process sinks
$k \in K$	Temperature levels
$m \in M$	Fresh resources
$t \in T$	Time intervals
$s \in S$	Storage tanks

Chapter 1 Introduction

1.1 BACKGROUND AND MOTIVATION

Global consumption of energy is expected to increase rapidly. Based on International Energy Outlook 2010 (IEO 2010) projections, total global consumption of energy is projected to increase by 49 percent from 495 quadrillion Btu in 2007 to 739 quadrillion Btu in 2035 (Figure 1.1). Energy consumption mainly occurs in four sectors, namely residential, industrial, commercial and transportation. Among these sectors, industrial sector consumes the highest amount of energy. In 2007, industrial sectors utilised 51 percent of global delivered energy and is expected to grow by an average rate of 1.3 percent annually (IEO 2010).

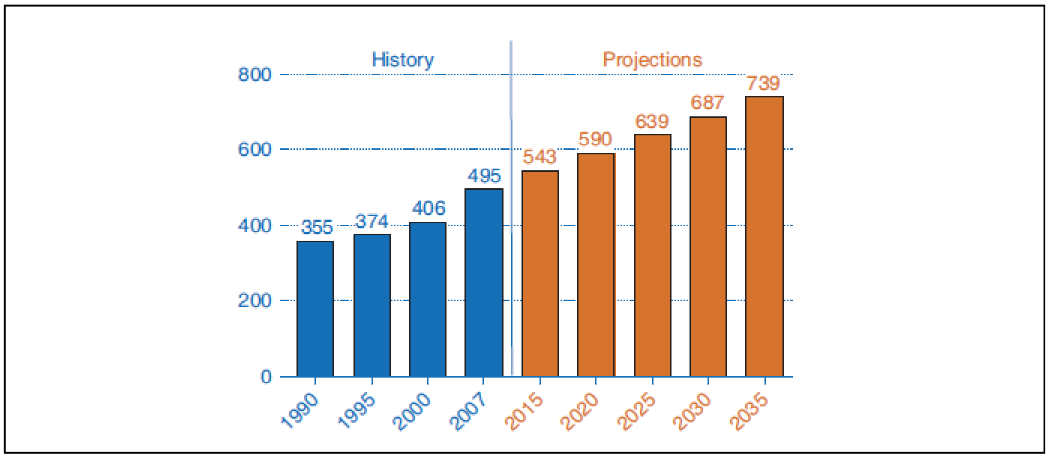

Figure 1.1: World marketed energy consumption, 1990-2035 (quadrillion Btu) (IEO 2010)

Other than energy consumption, the depletion of natural resources is growing dramatically and it is associated with a growing global population, economy and affluence (SERI *et al.*, 2009). A statistic was conducted on biomass, minerals, metals and fossil fuels extraction as presented in Figure 1.2. As shown, 58 billion tonnes of total global resource extraction in 2005 is expected to increase to more than 100 billion tonnes by year 2030 which is a 42 percent increase over 25 years. Furthermore, United Nations Environment Programme (UNEP 2011) has reported that humanity could devour an estimated 140 billion tons of minerals, ores, fossil fuels and biomass per year which is three times its current appetite by year 2050 (UNEP 2011).

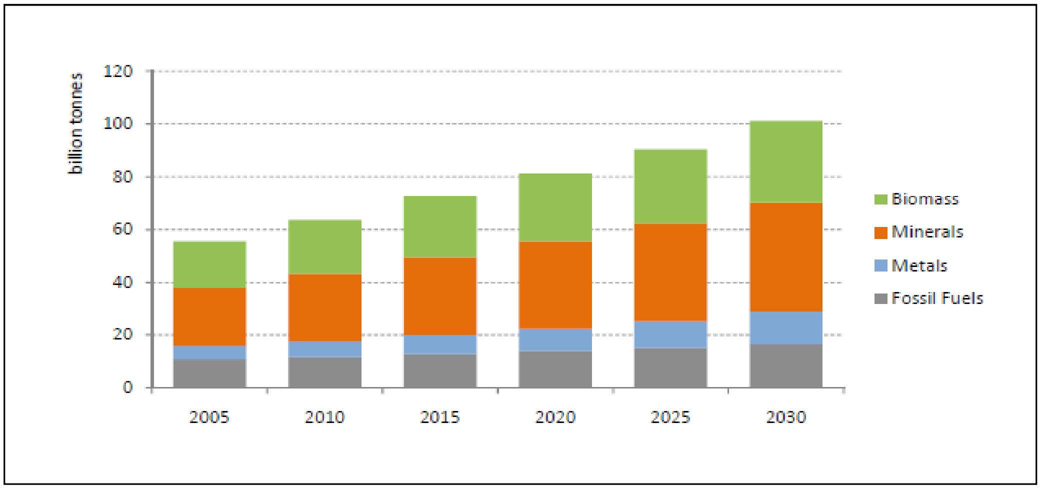

Figure 1.2: "Business-as-usual" scenario (if the world economy continues its current development path) on worldwide resource extraction, 2005 to 2030 (SERI *et al.*, 2009)

Therefore, high consumption of energy and natural resources has lead to increase cost of these resources. On the other hand, the current drive towards sustainability has resulted in stringent environmental regulations. All these have created a driving force for process industries to develop advanced technologies and practices to minimize the resources consumption and waste generation. One of the effective tools for resource conservation and waste reduction is via process integration. Process integration is defined as *a holistic approach to process design, retrofitting, and operation which emphasizes the unity of the process* (El-Halwagi 1997; 2006). Generally, process integration can be categorised as heat, mass and property integrations.

Heat integration is defined as *a systematic methodology that provides a fundamental understanding of energy utilization within the process and employs this understanding in identifying energy targets and optimizing heat-recovery and energy-utility systems*

(El-Halwagi 2006). Review on heat integration has been conducted by Linnhoff *et al.* (1982), Gundersen and Naess (1988), Linnhoff (1993), Furman and Sahinidis (2002), Smith (2005), Kemp (2007), Dunn and El-Halwagi (2003), Morar and Agachi (2010), Klemeš *et al.* (2010), and Fernández *et al.* (2012).

On the other hand, mass integration is defined as *a systematic methodology that provides a fundamental understanding of the global flow of mass within the process and employs this understanding in identifying performance targets and optimizing the generation and routing of species throughout the process* (El-Halwagi 2006). Mass integration started with the concept of a mass exchange network, which is a system of mass transfer units that employs a set of mass separating agents to selectively remove certain components from a set of rich process streams. Later, mass integration was established for resource conservation networks (RCNs). Extensive reviews on mass integration can be found in El-Halwagi (1998), El-Halwagi and Spriggs (1998), Dunn and El-Halwagi (2003), El-Halwagi (2006), El-Halwagi (2011) and Foo (2012).

However, mass integration techniques for material reuse and recycle are "chemo-centric" as mass integration techniques are only based on the composition of process streams without considering other properties or functionalities of the streams such as pH, colour, toxicity, TOC, BOD, viscosity and solubility (Shelley and El-Halwagi, 2000). Thus, property integration that took in the consideration of these properties was introduced. Property integration is defined as *a functionality-based, holistic approach to the allocation and manipulation of streams and processing units, which is based on the tracking, adjustment, assignment, and matching of functionalities throughout the process* (El-Halwagi *et al.*, 2004). Many methodologies have been developed for property-based RCNs.

Most of the works in the area of heat and property integrations have been studied separately. In many property-based RCNs, temperature is an important parameter. For example, solvent that is used for stripping purpose need to be heated or cooled to the desired temperature before entering the stripping unit. This shows that both of the temperature and quality of fresh resources are equally important. Thus, it is necessary to consider both heat and property integrations simultaneously. To date, limited works have been conducted in this area and yet these approaches are only applicable to continuous processes. Therefore, there is a need to develop a systematic approach to synthesize property-based heat integrated resource conservation networks (HIRCNs) for batch processes.

1.2 OBJECTIVE

The objective of this research is to develop a systematic approach to synthesize property-based HIRCNs for batch processes.

1.3 SCOPES OF RESEARCH

The scopes of this work have been outlined as follow:

a) A systematic approach to synthesize property-based HIRCNs for batch processes will be developed. The proposed superstructure will involve all process streams in heat exchanger network (HEN) as well as multiple storage tanks. The model formulation will consist of property-based RCN and HEN models which will be solve simultaneously to achieve the optimum fresh resources and hot and cold utilities targets.

b) Besides simultaneous targeting approach, the development of sequential targeting approach for property-based HIRCN will also be demonstrated. Then, the effectiveness of both approaches will be compared and analysed.

1.4 NOVELTY, CONTRIBUTION AND SIGNIFICANCE

The main contributions of this work are highlighted as follows:

a) Based on literature review conducted, this is the first work on property-based HIRCNs for batch processes. This work is established based on the basic concept of HEN and property-based RCNs. Furthermore, it is applicable to concentration-based HIRCNs.

b) This work presents a novel simultaneous targeting approach for property-based HIRCNs in batch processes.

c) Besides, sequential targeting approach has also been developed to synthesize property-based HIRCNs for batch processes.

1.5 SUMMARY OF THE THESIS

In this thesis, a new systematic approach to synthesize property-based HIRCNs for batch processes have been developed based on the basic concept of heat integration, property integration and batch processes. This thesis is organised as follows.

Chapter 2 provides a review of relevant literatures of this thesis. The development of heat integration and property integration are thoroughly reviewed.

The developed simultaneous and sequential targeting methodologies to synthesize property-based HIRCNs for batch processes are presented in Chapter 3.

Chapter 4 presents the problem statement and the model formulation to synthesize property-based HIRCNs for batch processes. The model formulation consists of two sections, namely property-based RCN and HEN models. Property-based RCN model is established based on supertargeting approach while HEN model is based on automated targeting method (ATM). The developed model is able to minimize the usage of heat and fresh resources for property-based HIRCNs batch processes.

In Chapter 5, the application of the developed simultaneous and sequential targeting methodologies in three case studies is presented. Moreover, analysis and discussions of the results are carried out in this chapter.

Chapter 6 concludes the thesis by encapsulating the contribution of this work and some recommendations of further works for property-based HIRCNs for batch processes. Figure 1.1 summarizes the thesis structure in a flow diagram.

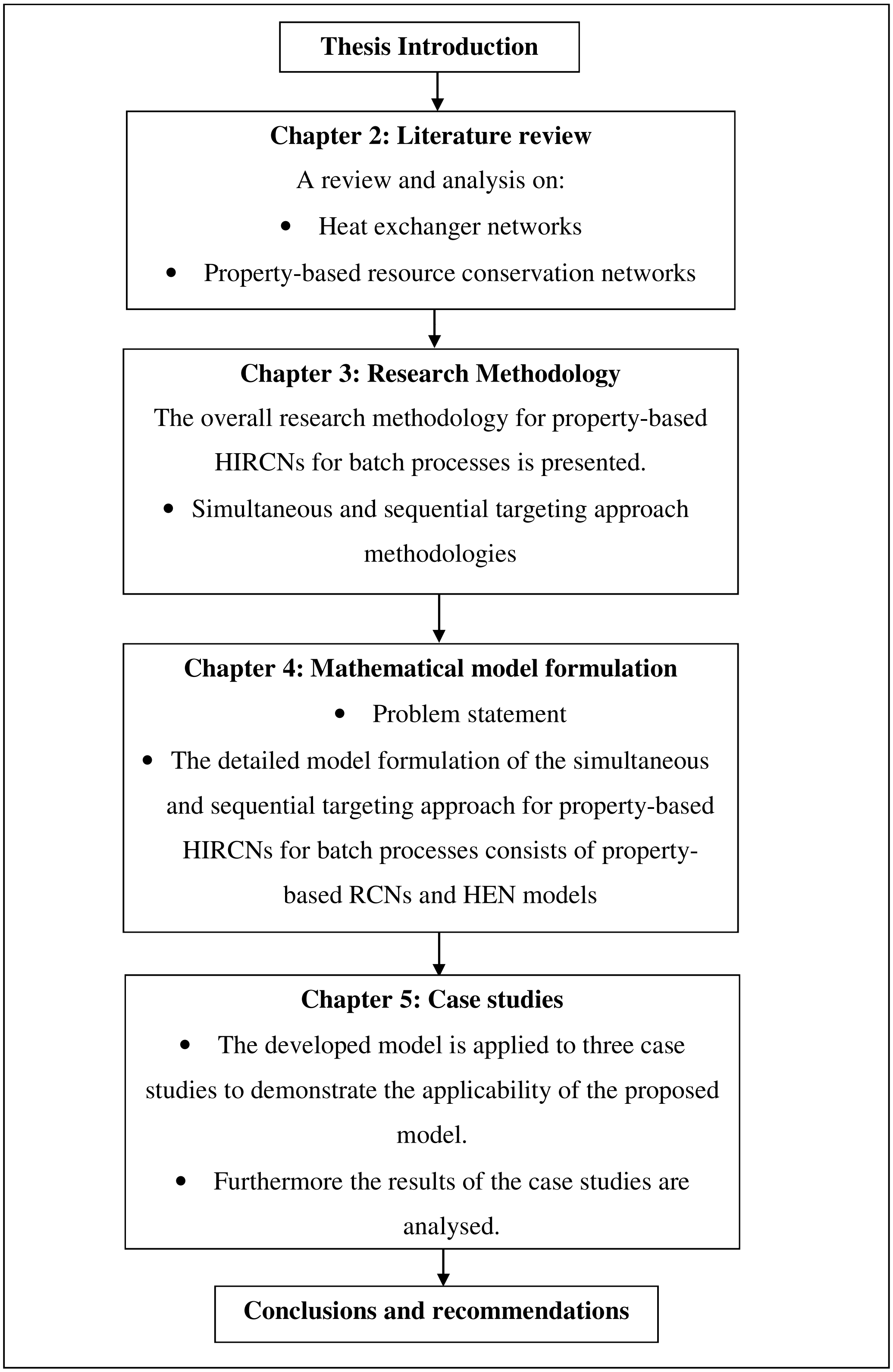

Figure 1.3: A flow diagram describing the thesis structure

Chapter 2 Literature Review

2.1 INTRODUCTION

Since 1970s, process integration has been recognized as an effective method to synthesize optimum heat exchanger network (HEN). In the late 1980s, process integration approach was extended to synthesize optimal mass exchange network in order to acquire the effective use of material. Over the past few decades, process integration has been established to synthesize different resource conservation network (RCN) for material reuses and recycles which includes water, hydrogen and property-based networks.

Generally, process integration can be divided into three categories such as heat, mass and property integrations. Most of the research works on heat, mass and property integrations have been conducted independently. Heat integration technique has been incorporated with water network synthesis, but less work has integrated heat integration with property-based RCN. These works for heat and property integrations are only focusing on continuous processes. Hence, the aim of this work is to develop a systematic technique to synthesize property-based heat-integrated resource conservation network (HIRCNs) for batch processes. This chapter therefore presents a critical review on the previous approaches for the synthesis of heat exchanger networks (HENs) (focusing on minimum utility targeting for batch processes), property-based resource conservation networks (RCNs) as well as property-based heat integrated resource conservation networks (HIRCNs).

Since many research works have been focused on HEN for continuous process with review being reported in Linhoff *et al*. (1982), Gundersen and Naess (1988), Furman and Sahinidis (2002) and Morar and Agachi (2010), only previous research works for HENs on batch processes are presented in this chapter. Then, the following sections are research works for property integration and property-based HIRCNs. Lastly, the research gap on the previous research techniques for the synthesis of HEN, RCN and HIRCNs is presented.

2.2 HEAT EXCHANGER NETWORKS FOR BATCH PROCESS SYSTEM

The current drive towards high-value, low tonnage products such as specialty chemicals, fine chemicals and pharmaceuticals have encouraged the batch process industry to figure out new methods to minimize energy consumption and maximize energy recovery. Thus, huge amount of research has been conducted in this area. .

These methodologies for HEN synthesis can be distinguished as sequential and simultaneous synthesis approaches. Sequential synthesis approach divides the HEN synthesis problem into minimum utility consumption and minimum number of heat exchanger units sub-problems. On the other hand, simultaneous approach obtains the optimal HEN structure based on total cost of utility consumption and heat exchanger units. Sequential approaches can be further divided into insight-based *pinch analysis* and mathematical programming techniques.

2.2.1 INSIGHT-BASED TECHNIQUE FOR SEQUENTIAL APPROACH

The original heat cascade analysis was established by Linnhoff (1979) to target the minimum hot and cold utilities consumption for continuous processes. Kemp and Deakin (1989a, 1989b,1989c) extended the heat cascade analysis called insight-based *pinch analysis* technique to batch processes. Kemp and Deakin (1989a, 1989b, 1989c) divided the process into a set of temperature and time intervals where heat is cascaded and with the involvement of heat storage. Time-temperature cascade tables for each time intervals were constructed to show heat recovery via direct heat exchange among the hot and cold streams and heat transfer between different time intervals using heat storage system. The developed technique was analyzed using various types of heat storage and a three-dimensional plot of the cumulative cascade was generated to visualize heat flows. The authors have proven that maximum energy recovery can be yielded by maximizing direct heat exchange and employing heat storage vessels. Then, the developed method was further investigated on producing heat exchanger network design and process rescheduling.

However, Kemp and Deakin (1989a, 1989b,1989c) did not consider the batch cycle time for rescheduling and heat storage is practically not used due to heat losses. Hence, Jung *et al*. (1994) enhanced the above work by developing a common technique for optimal rescheduling to maximize heat recovery and reduce batch cycle time when heat exchange occur in counter-current heat exchange mode. In this work, a heuristic rule which is so called the modified Hottest/ Highest is introduced to identify the optimal matching sequence of heat exchange in co-current heat exchange mode. Then, rescheduling opportunities are considered and followed by the minimization of the energy utility targets. The advantage of the proposed work is that the rescheduling procedure able to increase production and reduce indirect heat transfer by optimizing direct heat integration. Moreover, the developed heuristics are not complicated and require less computational effort.

Furthermore, the cascade analysis for heat integration that was presented by Kemp and Deakin (1989a, 1989b, 1989c) can be used to determine rescheduling opportunities after heat recovery. However, both heat integration and rescheduling were not performed simultaneously. Hence, Lee and Reklaitis (1995a, 1995b) developed a mathematical model that incorporates both rescheduling and heat integration simultaneously to determine the minimum usage of energy utilities. Firstly, the scheduling models were formulated for the NIS policy in single-product campaigns to maximize heat recovery. Then, a mixed integer linear programming (MILP) model was developed to obtain the operating schedule for optimal heat exchange among the batch streams in order to reduce the consumption of energy utilities. Later, the proposed model was extended to consider finite exchange times (the material transfer, heat exchange and processing times) and repeated batches. The results from the proposed studies have shown that more energy utilities can be saved when the heat exchange time is shorter.

2.2.2 MATHEMATICAL PROGRAMMING TECHNIQUE FOR SEQUENTIAL APPROACH

On the other hand, mathematical programming techniques based on sequential approach were established by Corominas *et al*. (1994), Zhao *et al.* (1998a), Halim and Srinivasan (2009) and Foo (2010). In Corominas *et al*. (1994) work, it is assumed that the process plants are operated in campaign mode in order to improve the flexibility of plant operation. Besides, the frequent change of product in batch sequences (product change over) problem was considered to obtain a feasible and optimal HEN design for multiproduct plants. This method also considered rescheduling aspects and designing a comprehensive HEN for all products (Macro-network) which includes the common matches between networks of campaigns of different products. Mathematical algorithms are used to acquire the best feasible matches and energy target of the multiproduct batch processes.

In 1998, Zhao *et al.* (1998a) developed a mathematical programming technique based on heat cascade analysis. This technique is used to optimize heat recovery cyclically operating in batch processes which includes no intermediate storage (NIS) policy with holding time. This method provides the matching pattern of multiple streams and it is adaptable in representing a number of batch processes. Thus, it creates more heat recovery opportunities with adequate results in an economic effective design. Later, Zhao *et al.* (1998b) utilized this systematic formulation to establish a three-step-design procedure for HEN design in batch or semi-continuous processes.

Furthermore, Halim and Srinivasan (2009) introduced a three-step mathematical programming method. In the approach, the scheduling problem was first solved to achieve an economic objective function. Later further constraints were added to the scheduling model in order to produce a set of alternative schedules via a stochastic search-based integer cut procedure. Lastly, heat integration that incorporates time average model and time slice model was applied to all generated schedules to demonstrate the minimum utility targets. The main benefit of this approach is its capability to solve complex problem.

2.2.3 AUTOMATED TARGETING METHOD FOR SEQUENTIAL APPROACH

Recently, an automated targeting method (ATM) which is the combination of both insight-based and mathematical programming technique was established for batch processes by Foo (2010). The author extended the ATM that was introduced for RCNs to solve both batch HEN problem. The proposed technique is able to identify the minimum external energy requirement, fresh resources and waste target for a batch network prior to the detailed design. Moreover, Foo (2010) also performed a two stages optimization where the usage of external fresh resource was minimized followed by minimizing the heat storage capacity. The main advantage of the ATM is its ability to determine the minimum cost target which is not possible for conventional insight-based technique.

2.2.4 SIMULTANEOUS APPROACH

On the other hand, many works have been established in the simultaneous synthesis method for HEN synthesis. These works on simultaneous approach for the synthesis of HEN in batch processes include Vaklieva-Bancheva *et al*. (1996), Bozan *et al*. (2001), Adonyi *et al.* (2003), Majozi (2006), Chen and Chang (2009), Liu *et al*. (2011) and Holczinger *et al.* (2012).

Vaklieva-Bancheva *et al*. (1996) demonstrated a MILP formulation which adopts the production campaigns and designs the HEN simultaneously in order to minimize the total capital and operating cost. The authors took into account a more general processing scheme where each product needs only a subset of the equipment stages. In the developed technique, binary variables and different types of constraints were added to the nonlinear objective function to transform the function in a linear form and then the MILP model was solved to obtain the minimum total cost network while fulfilling the given product demands. Nevertheless, the main disadvantage of the developed formulation is that transfer time has been ignored in every batch stages.

Bozan *et al.* (2001) reported that the model that was developed by Corominas *et al*. (1994) is only applicable for strict multiproduct plants. Every product in these strict plants must employ the same equipment item and the sequences of equipment items must also be the same for each product. On the other hand, the approach that was introduced by Vaklieva-Bancheva *et al*. (1996) can be applied to large multiproduct plants. However, it is difficult to understand, not practical in industry and unable to be solve through computer. Consequently, Bozan *et al*. (2001) developed a computerized and integrated technique that based on mathematical programming approach for multipurpose batch plants to identify the minimum operating cost of hot and cold utilities as well as the annualised capital cost of heat exchangers while the predetermined product demands in a given period of time were satisfied. Since the proposed technique took into account the most important factor which affects the plant operation like penalty of time, it can be applied to those problems that need sensitivity analysis.

On the other hand, Adonyi *et al.* (2003) developed an algorithm based on the S-graph approach that considers scheduling and heat integration simultaneously to obtain the minimum consumption of energy utilities for a given time horizon while fulfilling a constraint on the make span. According to Adonyi *et al.* (2003), Corominas *et al*. (1994) solved the scheduling problem before heat integration and this creates the possibility to cause worse results from heat integration. As a result, the obtained results are not an optimal result. However, the developed S-graph technique is able to perform both heat integration and scheduling simultaneously as well as minimize the usage of energy utilities with a slightly rise in make span. The drawback of their approach is that they assumed heat exchangers are existed for all of the hot-cold stream pairs and only one-to-one matches of streams which may result more energy utilities are required to satisfy the heat demand of the system.

In addition, Majozi (2006) also presented a mathematical approach which is a continuous-time mathematical formulation for both multiproduct and multipurpose batch processes to determine the production schedule that is associated with the maximum heat exchange and optimum profit. In the proposed work, two scenarios where energy requirement is dependent based on batch size that is allowed to vary at different instances along the time horizon of interest and fixed batch sizes were considered. The proposed approach which is a MILP model is more superior to pinch analysis as pinch analysis is unable to consider production scheduling, maximum heat recovery and optimal production simultaneously. Compared to the discrete-time formulation, this approach needs less computational work due to fewer amount of binary variables are generated. Since this approach is not restricted by time constraints, the developed model able to perform the repeated stage of the batch processes. Moreover, the objective function that was added to the developed model can be transformed into various forms (the maximal profit, or lowest operating cost etc.). Therefore, optimization on several objectives can be made at the same time.

Chen and Chang (2009) further extended the work from Majozi (2006) to a more complete framework which is the continuous resource-task network (RTN) to consider direct heat recovery and short-term batch scheduling in order to maximize the overall profit. In the proposed technique, a new MILP model for handling the short-term and periodic scheduling problems with direct heat integration in batch processes is developed to determine the optimal profit per cycle time. A set of adjustable parameter was established to handle the change of the starting time of HEN which increases the flexibility of the developed model.

Liu *et al.* (2011) introduced a new method to synthesize heat directly exchanged network for batch processes based on Pseudo-T-H diagram approach to achieve direct heat recovery within the operation process of the batch processes while remaining the original flow-sheet. A nonlinear programming (NLP) model was developed to obtain global optimization by minimizing the total annual cost which involves the operation cost for energy utilities and the capital cost for heat exchanger units simultaneously. In the developed work, a parallelized genetic-simulated annealing algorithm was used to obtain the global optimal solution.

Later, Holczinger *et al.* (2012) enhanced the simultaneous approach which was developed by Adonyi *et al.* (2003) in order to allow more heat recovery as well as minimize the consumption of energy utilities and the number of batches generated for each product. The authors developed a MILP model for a given problem and an S-graph based Branch-and-Bound algorithm was used for the discrete decisions making. Then, the interation between the MILP master problem and the S-graph framework reduces the usage of energy utilities for the batch plants. The proposed approach is able to consider the scheduling of both heat exchangers and processing equipments and allow one-to-many heat exchange.

2.3 PROPERTY-BASED RESOURCE CONSERVATION NETWORK

2.3.1 INSIGHT-BASED TECHNIQUE FOR RESOURCE CONSERVATION NETWORK

One of the most established areas in process integration is resource conservation networks (RCNs). However, RCNs techniques for material reuse and recycle are "chemo-centric" as these techniques are only based on the composition of process streams without considering other properties or functionalities of the streams (Shelley and El-Halwagi, 2000). Thus, property integration that took in the consideration of these properties was introduced.

Shelley and El-Halwagi (2000) first proposed the concept of property-based clustering technique. The developed clustering method is used to map the infinite-dimension problem into a two-dimensional domain by tracking functionality and properties of the complex hydrocarbon mixture as opposed to the individual components. In this work, important properties for conserved tracking of clusters and tailored lever-arm rules were developed. Lastly, these clusters were incorporated into mass integration framework to identify optimal strategies for the recovery and allocation of volatile organic compounds.

However, the proposed clustering technique by Shelley and El-Halwagi (2000) can only be used for processes that involve condensation devices. Moreover, the developed ternary presentation only maps the infinite points from the property domain to the cluster domain which may neglect some feasible regions. Furthermore, this technique does not provide a direct optimization way to determine optimal blends.

Thus, El-Halwagi *et al*. (2004) improved the above limitations by establishing component-less design and visualization tools. The authors implemented property interception network to improve the properties of process sources before entering the process sinks. The authors also developed mathematical expressions which describe the actual shape of the feasibility region on the cluster domain and these expressions are incorporated with rigorous optimization rules to define the optimal blends, allocation strategies and tasks of property-modifying devices. Lastly, the proposed graphical tools work together with the derived rules to determine optimal allocation strategies and minimum extent of required interception.

On the other hand, Qin *et al*. (2004) established algebraic tools for processes involving more than three properties. The authors developed constraint-reduction algorithm with the help of the mathematical structure of the problem to provide rigorous bounds on the flexible region. Moreover, the proposed approach can determine the optimal fractional contributions of any number of sources for mixing problems and minimum cost of external fresh resources.

Kazantzi and El-Halwagi (2005) and Foo *et al.* (2006) incorporated the concept of property integration into pinch analysis techniques for concentration-based RCN. Kazantzi and El-Halwagi (2005) developed a property-based pinch diagram to determine the targets for the minimum use of fresh resources and waste discharge. In the proposed work, the diagram was modified to include property operators in order to track properties of the process streams and the developed technique applies segregation, mixing and recycles/ reuses strategies. On the other hand, Foo *et al.* (2006) introduced a graphical method called property surplus diagram and a numerical method named property cascade analysis to determine the minimum usage of external fresh resources and waste produced in a property-based RCN.

2.3.2 AUTOMATED TARGETING METHOD FOR RESOURCE CONSERVATION NETWORK

Besides the insight-based techniques, an ATM which is a combination of insight-based and mathematical approach is developed for property-based RCN by Ng *et al.* (2009a). The authors established a mathematical model to obtain the minimum usage of fresh resource or the total operating cost of fresh resources for a property-based RCN. Moreover, the author's work also considered process modification, interception processes and bilateral property integration problem.

The above ATM is only focusing on property-based RCN. Thus, Ng *et al.* (2010) extended the aforementioned ATM to concentration and property-based total RCN, in which material recycle/ reuse and waste interception are addressed simultaneously. The authors considered a total RCN that consists of resource pre-treatment, material reuse, regeneration or interception, and waste treatment for final discharge. The optimization-based technique is proposed to minimize the consumption of fresh resource or total cost of a concentration- or property-based total RCNs. Ng *et al.* (2010) reported that the concept of insight-based targeting approach is limited to single impurity or property problem but the ATM can handle different impurities and properties for the reuse/ recycle and interception/ waste treatment networks.

2.3.3 MATHEMATICAL PROGRAMMING TECHNIQUE FOR COMPOSITION-BASED RESOURCE CONSERVATION NETWORK

The aforementioned approaches only considered property integration in the presented works. However, there are some research works on simultaneous mass and property integration. Ponce-Ortega *et al.* (2009) introduced an MINLP model to optimize the direct recycle networks together with waste water treatment in order to minimize the TAC of the system that involves the cost for fresh sources, the piping cost for process integration and the waste water treatment cost. The MINLP model considers both process integration and wastewater treatment process simultaneously while fulfilling a set of composition- and property-based constraints as well as environmental constraints to achieve economic saving. The proposed model was developed based on disjunctive programming and incorporated with waste treatment technologies to meet environmental regulations. Furthermore, the proposed approach proved that simultaneous consideration of process and environmental constraints minimizes the TAC of the system compared to a traditional sequential approach.

Nevertheless, the above approach that was presented by Ponce-Ortega *et al*. (2009) did not consider the involvement of property interceptors which may enhance the direct recycle and reuse strategy. Thereafter, Ponce-Ortega *et al.* (2010) presented a mathematical programming model for the direct recycle and reuse of mass and property integration network that includes property interceptors while satisfying process and environmental constraints in order to minimize the TAC. Based on a superstructure with the presence of property interception network, the mathematical model is formulated in such a way that most of the nonlinearities of the model are excluded. However, the developed model consists of bilinear terms and makes it a non-convex MINLP problem. Hence, a relaxation approach is adopted to handle the set of bilinear terms in order to obtain global optimal result.

Later, Nápoles-Rivera *et al*. (2010) presented a mathematical programming model for the recycle and reuse of mass and property integration networks that considers both process and environmental constraints as well as consists of in-plant property interceptors within the structure of the network to optimize the design of the network. The TAC that includes the fresh resources cost, the piping cost and the annualized property treatment system cost can be minimized by using the proposed model. In the proposed work, a relaxation approach was used to remove most of the nonlinearities of the system and remaining bilinear term in order to obtain a global optimal solution. The results from the case study have shown that the proposed model is applicable to a large size problem. Moreover, the authors have also shown that simultaneous consideration of process integration and property treatment systems is superior to sequential method.

2.3.4 RESOURCE CONSERVATION NETWORK FOR BATCH PROCESSES

The above stated property-based RCN works were established for continuous processes. There are also some methods that focus on batch processes. Grooms *et al.* (2005) formulated a mixed integer nonlinear programming (MINLP) mathematical model to synthesize and schedule a hybrid (steady-state and dynamic) property-interception and allocation network based on the defined properties. The developed model is used to minimize the total annualized cost (TAC) of the system which includes the operating costs of the steady-state MSA used in the absorber and the generating cost of the unsteady-state adsorbers as well as the capital costs of the fixed-bed columns. In their work, a source-interception-sink representation was constructed to determine optimal allocation and interception of sources while satisfying the constraints of the sinks at minimum cost. Besides optimizing unit sizes and network configuration, the proposed model is able to determinethe optimum regeneration scheduling for the fixed beds and the optimum distribution of flows to each unit in order to minimize both operating and capital costs of the property-interception and allocation network.

Later, Ng *et al*. (2008) developed an optimization mathematical programming approach to synthesize a cost-effective batch concentration-based RCN with incorporating property interceptors to optimize the reuse/ recycle of process streams and minimize the usage of external fresh resources. In this work, a source-tank-interception-tank-sink representation was developed to embed all possible network configurations and it was used to formulate the optimization mathematical model for the cost-effective batch water network. Moreover, storage tanks were involved in the concentration-based RCN to allow mixing, storage and dispatch of the reused/ recycled process streams to the interception devices or sinks. Nevertheless, their techniques may require a large number of tanks and it is not suitable for large scale highdimensional problems due to large number of process sources and sinks may create difficulty for their inspection step in eliminating the un-required tanks.

Chen *et al.* (2010) established a generic mathematical model for property-based RCN employed in batch and continuous processes with consideration of waste treatment. Their mathematical formulation incorporated with time index to reflect time dimension of batch processes (if present) and continuous processes are treated as a special case of batch processes. Moreover, the mathematical model was developed for the overall framework of property-based RCN which includes material recycle/ reuse, interception and waste treatment. Interception devices were added to the RCN to adjust stream properties for further reuse or recycle while storage system was also included to override the time constraint for batch processes.

Foo (2010) introduced the first work on ATM for batch processes. The author formulated a mathematical optimization model based on the ATM for batch process integration problem to determine the minimum resource and waste targets with the presence of storage system. For property-based integration, the mathematical model was developed based on property cascade diagrams that are performed across all property intervals in all time intervals to obtain overall minimum flow or cost targets. Moreover, the storage tanks are included in the network design to improve the material recovery across the time intervals. The author has successfully applied the developed mathematical model to a case study for property-based integration problem.

2.4 PROPERTY-BASED HEAT INTEGRATED RESOURCE CONSERVATION NETWORKS

The foregoing researches on mass and property integration have not considered the influence of temperature in the RCN. Thus, Kheireddine *et al.* (2011) extended the area of mass and property integrations with temperature effect to generate a cost-effective direct-recycle network. In the developed study, a NLP model that fulfils the process and environmental constraints was developed to minimize the TAC which includes the costs for fresh resources, waste discharge and pipeline. Moreover, the authors also took into account the heat of mixing and the interdependence properties in the proposed work. This is the first work that incorporated simultaneously on both mass and property integrations with the consideration of thermal effect in the direct recycle network.

However, the model presented by Kheireddine *et al.* (2011) did not include any temperature interceptors (e.g. heat exchangers, heaters or coolers) in the direct recycle networks. In some cases, temperatures of the process streams are required to be adjusted before entering process sinks or discharging as waste. Besides, there are other limitations in the previous studies for process integration where the characterization of the mixing operators for key environmental properties, temperature dependence of the properties and the simultaneous consideration of the heat, mass and property effects have been ignored. Hence, Rojas-Torres *et al*. (2012) introduced a systematic method that considers mixing operators for environmental properties, temperature dependence of the properties and the effects of heat, mass, and property to synthesize the regeneration recycling water networks. Moreover, mixing rules and a new set of property operators are introduced to tackle the properties of the streams and fulfil the process and environmental constraints. Also, the proposed technique considers thermal-environmental constraints due to the presence of temperature interceptors.

Nevertheless, the aforementioned work on property-based HIRCN did not considered heat recovery among the process streams via heat integration. Therefore, Tan *et al.* (2013) presented a technique that synthesizes the concentration- and property-based HIRCNs with the consideration of varying process parameters to minimize the usage of fresh resources and energy utilities. The proposed technique is a MINLP model for HIRCNs which consists of concentration- and property-based RCN model as well as HEN model. Since the proposed MINLP model is nonlinear, a discretization method is introduced to transform the proposed MINLP model into MILP.

2.5 THE RESEARCH GAP

From the literature review presented above, it is observed that the current property-based HIRCN only focus on continuous processes. The development of property-based HIRCN for batch processes has not been established yet. Due to the development of technology-intensive industries such as pharmacy, fine chemistry and foods, methodologies for batch processes RCN is in demand. Thus, a new systematic methodology for property-based HIRCN for batch processes should be proposed in order to cater for this need. The objective of this work is to develop a systematic approach to synthesize property-based HIRCNs for batch processes.

The main challenge of this work is to develop methods that consider both HEN and property-based RCN simultaneously. Besides, the formulation of the stream parameterisation models to allocate process sources and sinks in the respective time intervals is another challenging task. Furthermore, incorporation of storage system in property-based HIRCNs to further minimize the usage of the external fresh resources and energy utilities is another challenge of this work. With this methodology established, minimisation of the raw materials (fresh resources) and energy utilities in batch process industries can be achieved simultaneously. The proposed technique is also applicable to concentration-based HIRCNs.

Chapter 3 Research Methodology

3.1 INTRODUCTION

This chapter describes a systematic methodology for the synthesis of property-based heat integrated resource conservation networks (HIRCNs) for batch processes that has been developed in this work. The overview of the developed methodology is presented in Figure 3.1.

3.2 RESEARCH METHODOLOGY

With the literature review conducted and research gap identified in Chapter 2, the property-based HIRCNs for batch processes problem to be addressed in this work is formulated. The problem is constructed by incorporating heat exchanger network (HEN) into property-based resource conservation network (RCN) for batch processes. In order to illustrate the problem formulated, a source-HEN-sink superstructure is presented. This superstructure is then used to develop the optimization mathematical model for property-based HIRCNs in batch processes, which consists of HEN and property-based RCN models.

In this work, automated targeting method (ATM) which is a combination of insight-based and mathematical optimization techniques is chosen to formulate the HEN model. ATM is selected because it is able to obtain the minimum energy utilities for a HEN. Besides, ATM for HEN has advantage of being capable to visualize heat load cascade across all the temperature levels which cannot be done using other HEN mathematical models. Moreover, ATM for HEN is able to integrate with the property-based RCN model to determine the minimum total annual cost of HIRCNs.

On the other hand, supertargeting approach is used to formulate the property-based RCN model. This method is selected rather than the ATM because the same type of flow rate variable is needed to integrate the HEN model with the property-based RCN model. In the HEN model based on ATM, the flow rate variable needed in this model is all the hot and cold streams which mainly consist of the fresh resources streams supply to each sink, the reuse/recycle of each source to each sink and the discharge of each source as waste. This same type of flow rate variable is also used in the super-targeting model for property-based RCN. However, in ATM for property-based RCN, total flow rate variable of each sink and source are used. Thus, supertargeting approach is used to generate the property-based RCN model. As a result, the developed technique to synthesize HIRCNs is a hybrid of ATM and supertargeting approach. Note that incorrect results of the HIRCNs will be obtained if the total flow rate variable in ATM for property-based RCN is replaced by the same type of flow rate variable in HEN model (the fresh resources streams supply to each sink, the reuse/recycle of each source to each sink and the discharge of each source as waste). This is mainly due to incorrect mass balance for each sink and source when this type of flow rate variable is used.

Besides, the developed mathematical model for property-based HIRCNs is demonstrated via simultaneous targeting approach and sequential targeting approach. Simultaneous targeting approach is where the developed HIRCN model is solved with consideration of storage system to minimize the total annualized cost which includes the cost of fresh resources, external hot and cold utilities as well as the cost of storage tanks. For sequential targeting approach which is so called two stages optimization, the developed HIRCN model is solved with consideration of storage system using where the minimization of total annual operating cost of fresh resources is first considered followed by minimizing the costs of external hot and cold utilities. Then, the annualised capital cost of storage tanks is determined.

Lastly, the HIRCNs configurations for both of the developed simultaneous and sequential targeting approaches are constructed. To achieve a complete HIRCN configuration, the RCN structure and HEN design are needed. The RCN result is obtained directly from the proposed model solution which is generated by Extended

LINGO version 11.0 with Global Solver. Based on the result solution, a revised source-HEN-sink superstructure for each time interval which is the network configuration of the HIRCNs is drawn. Nevertheless, for HEN, the proposed model solution provides the external hot and cold utilities without any HEN design. Thus, the HEN design is synthesized based on the classical pinch design method (Linnhoff and Flower, 1978; Linnhoff *et al*. 1982; Smith 2005). This step also verifies the hot utility and cold utility targets obtained in the proposed model solution.

In order to illustrate the applicability and usefulness of the developed model, three case studies are solved. Furthermore, both simultaneous and sequential targeting approaches are applied into these case studies and compared with the solution from the proposed model. Note that both simultaneous and sequential approaches are developed for the synthesis of property-based HIRCNs.

3.3 SUMMARY

A systematic research methodology for property-based HIRCN for batch processes has been presented. The problem statement, mathematical model as well as case studies are presented in detail in Chapter 4 to Chapter 5 respectively.

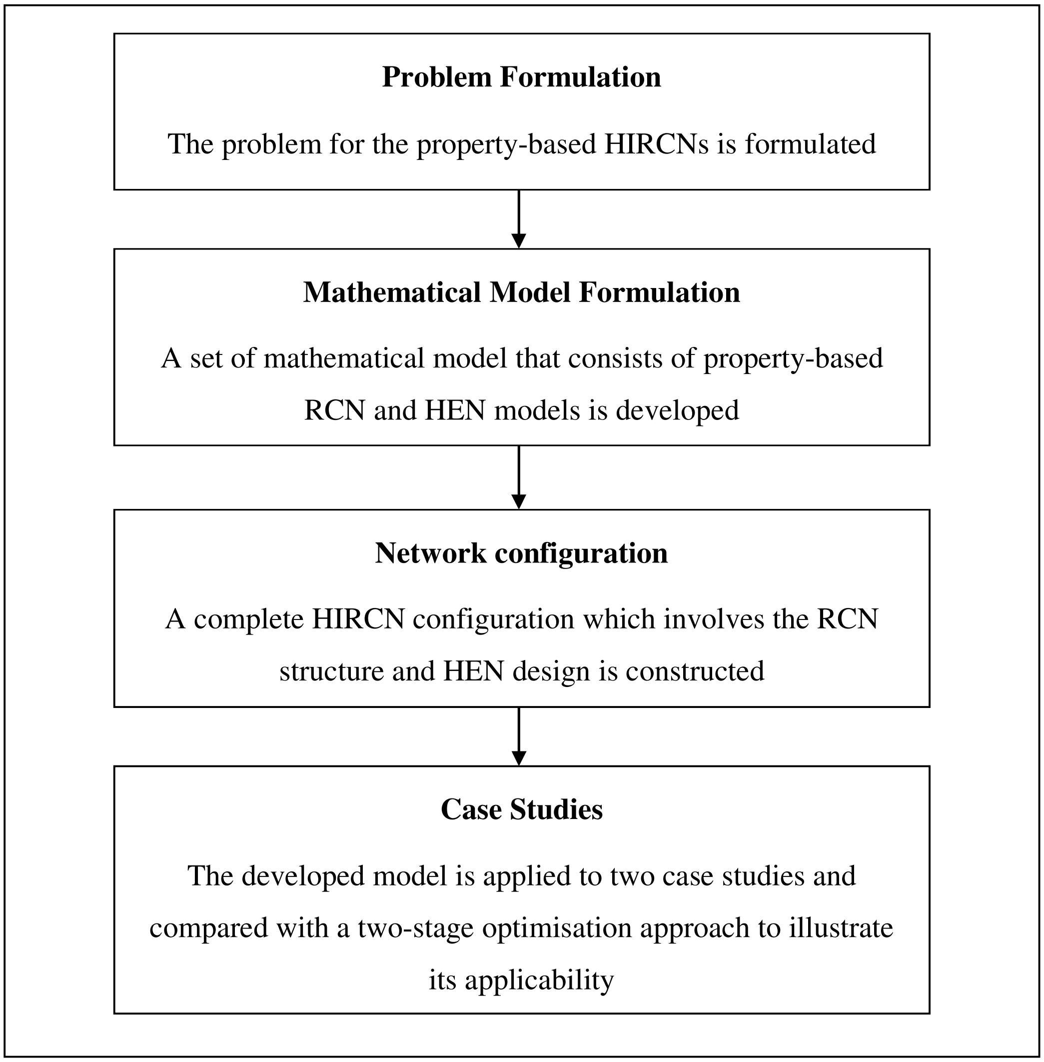

Figure 3.1: Overview of the developed methodology in this work

Chapter 4 Mathematical Model Formulation

4.1 INTRODUCTION

This chapter presents the property-based heat integrated resource conservation networks (HIRCNs) for batch processes problem to be addressed in this work. Firstly, the problem is formulated by incorporating heat exchanger network (HEN) into property-based resource conservation network (RCN) for batch processes.

Next, mathematical model to synthesize a property-based HIRCNs with minimum total annualised cost (TAC) for batch processes is presented. This model is divided into four sections, namely data conversion, property-based RCN, HEN as well as objective function. The first section lays out the equations needed to convert raw data of batch processes into the desired limiting data for the RCN and HEN models. Secondly, the property-based RCN model which is based on supertargeting approach (Rabie and El-Halwagi 2008; Shoaib *et al*. 2008; Chen *et al*. 2008) is presented. Binary variables are incorporated in the property-based RCN to locate the existence of process sinks and sources in each time interval. Then, HEN model via automated targeting method (ATM) (Foo 2010) is shown in the third section. Note that the property-based RCN and HEN models shown are developed for each time interval in the HIRCN. The fourth section presents the objective function of the HIRCN which takes the form of TAC (the operating cost of fresh resources, hot and cold utilities as well as the capital cost of storage tanks).

4.2 PROBLEM STATEMENT

The problem to be addressed in this work can be stated as follows.

Given is a batch process with τ batch cycle and a set of time intervals, INTERVALS = $\{t|t = 1,...,\text{NINTERVALS}\}$. The batch process is associated with a set of process sources, SOURCES = {i|i = 1,...,NSOURCES} and a set of process sinks, SINKS = {j|j = 1,..., NSINKS}. Process sources may be used for reuse/recycle to process sinks or be discharged as waste. Each process source i, has a fixed flow rate (F_i), property operator (ψ_i), temperature (T_i) as well as a start time ($t_{i,\mathrm{ST}}$), and end time ($t_{i,\mathrm{ET}}$). Process sinks are equipments that can accept process sources. Each sink, j requires a flow rate (F_j), temperature (T_j), property operator (ψ_j) and has a start time ($t_{j,\mathrm{ST}}$) and end time ($t_{j,\mathrm{ET}}$). The allowable property constraint for each process sink j is stated as:

$$\psi_j^{\min} \le \psi_j \le \psi_j^{\max} \tag{4.1}$$

where $\psi_j^{\min}$ and $\psi_j^{\max}$ are the lower and upper limits of the acceptable property operator of process sink j.

Moreover, there is a set of external fresh resources, FRESH = {m|m = 1,...,NFRESHS} that may be purchased to supplement the sinks. Each fresh resource has a given temperature (T_m) and property operator (ψ_m). The flow rate of fresh resource m is to be determined as part of the solution. Furthermore, a set of storage tanks, STORAGES = {s|s = 1,...,NSTORAGES} is available to store each individual process sources from one interval to another. It is assumed that the properties and temperature of process streams remain unchanged in the storage tanks.

When various process streams are mixed, a general mixing rule is required to specify all the potential mixing patterns among these individual properties and it is expressed as below (Shelley and El-Halwagi, 2000).

$$\psi(\overline{p}) = \sum_l x_l \psi_l \tag{4.2}$$

where $\psi(\bar{p})$ and ψ_l are the operators on mixture property $\bar{p}$ and property of stream l respectively; while x_l is the fractional distribution of stream l of the total mixture flow rate.

In the HIRCNs, all streams are intended for the HEN. These streams are to be heated (cold streams) or cooled (hot streams) before entering process sinks. The classification of hot and cold streams is pre-defined as the supply and target temperatures of these streams are directly extracted from the given source and sink temperature limiting data. The flow rate and temperatures of these streams are to be obtained simultaneously within the HIRCN. Heat is exchanged between the hot streams and cold streams. Besides, external hot (Q_H) and cold (Q_C) utilities are available to fulfil the heating and cooling requirement after maximum energy recovery. Note that after HEN the properties and flow rate of process streams remain the same. Figure 4.1 shows the superstructure of source-HEN-sink for this problem. As shown in Figure 4.1, process source, i and stored source in storage tank, s are sent to process sink j for reuse or discharged as waste. Fresh resource is only sent to process sink, j whenever it is needed. Storage tanks are added to the superstructure to enhance the material recovery. Besides, a HEN is placed in the superstructure to allow heat exchange among all the hot and cold streams in order to obtain the energy recovery. The superstructure of source-HEN-sink is design to achieve the objective of this work. The objective of this work is to synthesize a property-based HIRCNs for batch processes of minimum cost, which includes the operating cost for the fresh sources and hot and cold utilities as well as the capital cost of storage tanks.

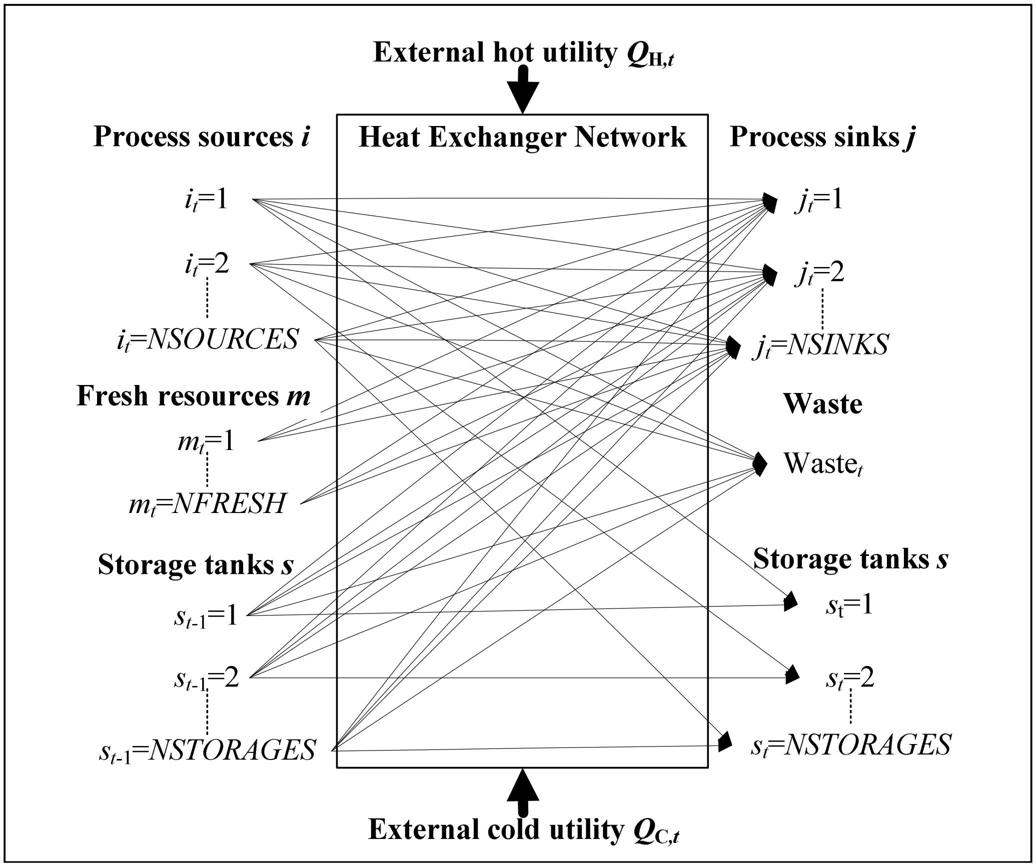

Figure 4.1: Superstructure of source-HEN-sink for property-based HIRCNs for batch processes

The following assumptions are used in this study:

(i) Counter-current heat exchange mode is assumed for heat recovery as it is most efficient (Zhao *et al.* 1998a).

(ii) Only single impurity or property is considered in the synthesis of HIRCNs problems.

(iii) Each storage tank is assumed to maintain at the desired temperature with external heating or cooling system.

4.3 DATA CONVERSION

The raw data of batch processes are usually given in total amount in each batch cycle as well as the start time and end time for each process sinks and sources. However, the desired limiting data for the RCN and HEN models developed in this work are the fractional amount of each process sinks and sources in each time interval. Thus, data conversion for process sinks and sources is needed.

The fractional amount of process sources *i* and sink *j* in time interval *t* can be achieved via Equation 4.3 and 4.4 respectively.

$$F_{i,t} = \left[\frac{F_i}{t_{i,\mathrm{ET}} - t_{i,\mathrm{ST}}} \right] \Delta t_t \quad \forall i \in I, t \in T \tag{4.3}$$

$$F_{j,t} = \left[\frac{F_j}{t_{j,\mathrm{ET}} - t_{j,\mathrm{ST}}} \right] \Delta t_t \quad \forall j \in J, t \in T \tag{4.4}$$

where F_i and F_j are the total amounts of process source *i* and process sink *j* in each batch cycle respectively, while $t_{i,\mathrm{ET}}$, $t_{i,\mathrm{ST}}$, $t_{j,\mathrm{ET}}$ and $t_{j,\mathrm{ST}}$ are the end and start times of the process source *i* and process sink *j* respectively; Δt_t is the duration of time interval *t* and can be achieved via Equation 4.5.

$$\Delta t_t = t_t - t_{t-1} \quad \forall t \in T \tag{4.5}$$

where t_{t-1} and t_t represent the start and end times of time interval *t*.

In order to define t_{t-1} and t_t, the time interval of the process sinks and sources are arranged in ascending order, from the lowest time interval $t = 0$ to the highest time interval $t = T$, as presented in Figure 4.2.

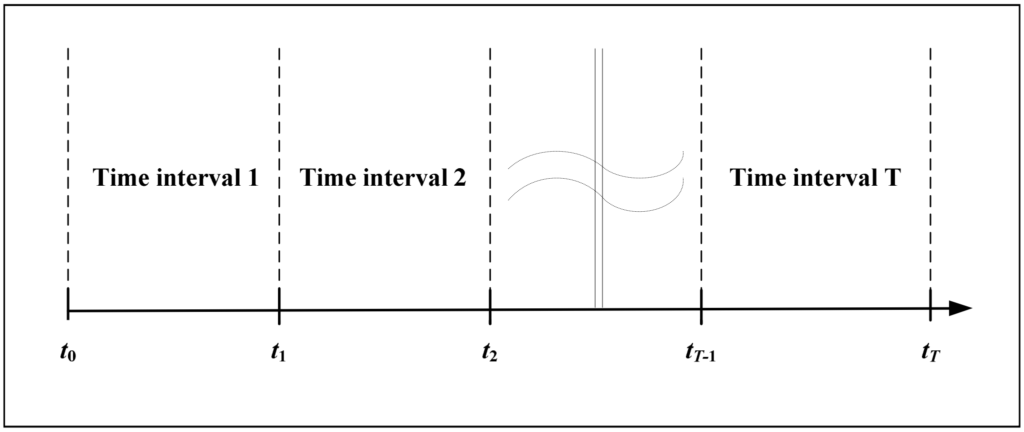

Figure 4.2: A diagram describing time intervals

As shown, the first time interval starts at t_0 and ends at t_1. Thus, t_0 and t_1 are the start and end times of time interval $t = 1$ respectively. For the second time interval, t_1 and t_2 are the start and end times respectively. The same definition is repeated for the rest of the time intervals.

4.4 PROPERTY-BASED RESOURCE CONSERVATION NETWORK

The following sub-sections present the models for stream parameterisation and property-based RCN. In stream parameterisation model, binary variables are used to parameterise the existence of process sinks and sources in each time interval. The model for property-based RCN is derived based on the source-HEN-sink superstructure in Figure 4.1 and mainly consists of mass balances for various process sources, process sinks and storage tanks.

Stream parameterisation

In this section, stream parameterisation models are introduced to allocate process sources and sinks in the respective time intervals respectively. The binary variables used to parameterise the process source *i* in each time interval are given by the constraints in Equations 4.6 to 4.8.

$$Z_{i,t} = \begin{cases} 1, & if \;\; t_t > t_{i,ST} \\ 0, & if \;\; t_t \le t_{i,ST} \end{cases} \quad \forall i \in I, t \in T \tag{4.6}$$

$$X_{i,t} = \begin{cases} 1, & if \;\; t_{i,ET} > \mathrm{t}_{\mathrm{t\text{-}1}} \\ 0, & if \;\; t_{i,ET} \le t_{t-1} \end{cases} \quad \forall i \in I, t \in T \tag{4.7}$$

$$Y_{i,t} = X_{i,t} Z_{i,t} \;\; \forall i \in I, t \in T \tag{4.8}$$

where $Z_{i,t}$, $X_{i,t}$ and $Y_{i,t}$ are binary variables for process source *i* in time interval *t*.

To demonstrate how Equations 4.6 to 4.8 determine the existence of process sources in a time interval, let us consider the following three scenarios for the existence of Process Source 1 (SR1) in a batch process with three time intervals where SR1 only appears in the second time interval (Figure 4.3).

1. Scenario 1 examines the presence of SR1 in time interval 1 (t_1 to t_2). Equations 4.6 and 4.7 determine that integer $Z_{SR1,1}$ = 0 and $X_{SR1,1}$ = 1. Therefore, $Y_{SR1,1}$ in Equation 4.8 is zero and this indicates that SR1 does not exist in time interval t_1 to t_2.

2. Scenario 2 investigates the existence of SR1 in time interval 2 (t_2 to t_3). Integers $Z_{SR1,2}$ and $X_{SR1,2}$ are equal to 1based on Equations 4.6 and 4.7 respectively. This denotes that SR1 exists in time interval t_2 to t_3 as $Y_{SR1,2}$ = 1 based on Equation 4.8.

3. Scenario 3 determines the existence of SR1 in time interval 3 (t_3 to t_4). Based on Equation 4.6 and 4.7, integers $Z_{SR1,3}$ = 1 and $X_{SR1,3}$ = 0. Thus, from Equation 4.8, $Y_{SR1,3}$ = 0 which reflects that SR1 exists in time interval t_3 to t_4.

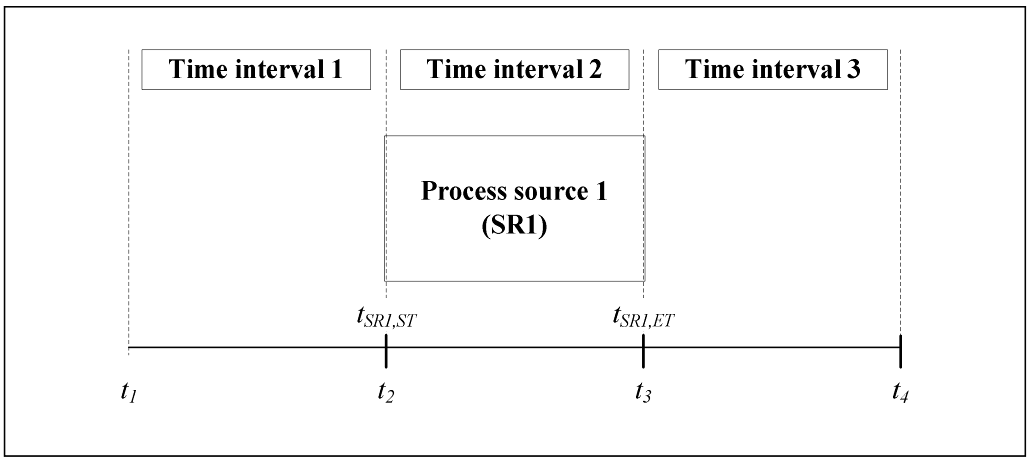

Figure 4.3: SR1 in a batch process with three time intervals

On the other hand, Equations 4.9 to 4.11 are used to define the existence of process sink *j* in each time intervals.

$$Z_{j,t} = \begin{cases} 1, & if \ \ t_t > t_{j,ST} \\ 0, & if \ \ t_t \le t_{j,ST} \end{cases} \quad \forall j \in J, t \in T \tag{4.9}$$

$$X_{j,t} = \begin{cases} 1, & if \ \ t_{j,ET} > t_{t-1} \\ 0, & if \ \ t_{j,ET} \le t_{t-1} \end{cases} \quad \forall j \in J, t \in T \tag{4.10}$$

$$Y_{j,t} = X_{j,t}Z_{j,t} \quad \forall j \in J, t \in T \tag{4.11}$$

where $Z_{j,t}$, $X_{j,t}$ and $Y_{j,t}$ are binary variables for process sink *j* in time interval *t*.

Similar to process source, three scenarios based on a batch process with three time intervals (Figure 4.4) are analyzed to explain how Equations 4.9 to 4.11 parameterise Process Sink 1 (SK1) in each time interval.

1. Scenario 1 determines the existence of SK1 in time interval 1 (t_1 to t_2). From Equations 4.9 and 4.10, $Z_{SK1,1}$ = 0 and $X_{SK1,1}$ = 1. Thus, $Y_{SK1,1}$ = 0 (Equation 4.11) which shows that SK1 does not exist in time interval t_1 to t_2.

2. Scenario 2 defines the presence of SK1 in time interval 2 (t_2 to t_3). From Equations 4.9 and 4.10, integers $Z_{SK1,2}$ and $X_{SK1,2}$ are equal to 1. Consequently, from Equation 4.11, $Y_{SK1,2}$ = 1 which indicates that SK1 presents in time interval t_2 to t_3.

3. Scenario 3 identifies the appearance of SK1 in time interval 3 (t_3 to t_4). Based on Equations 4.9 and 4.10, $Z_{SR1,3}$ = 1 and $X_{SR1,3}$ = 0. As a result, from Equation 4.11, $Y_{SK1,3}$ = 0 which reflects that SR1 does not exist in time interval t_3 to t_4. Note that these stream parameterisation models for process source *i* and sink *j* shown are developed for each time interval in the HIRCN.

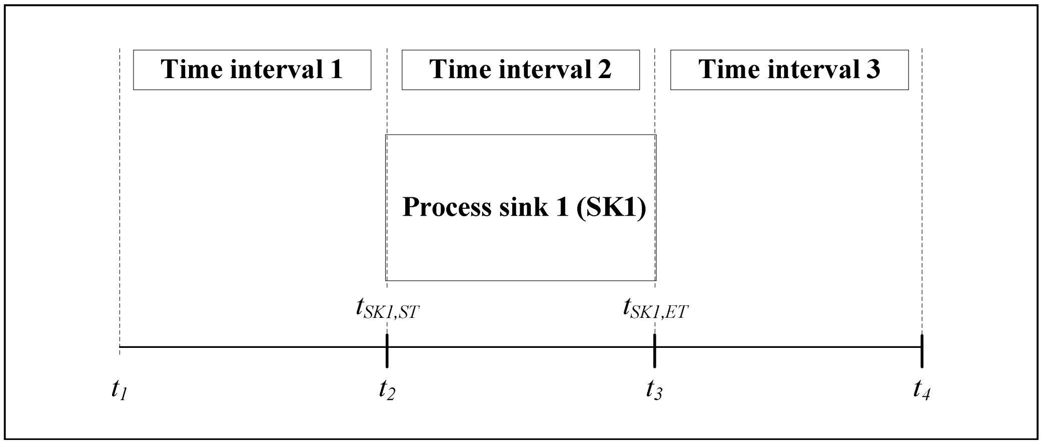

Figure 4.4: SK1 in a batch process with three time intervals

Material balances process sources

Process source *i* may be segregated and sent to process sink *j*, the respective storage tank*s* and/or be discharged as waste. Equation 4.12 describes the flow balance on the splitting of process source *i* in time interval *t*.

$$F_{i,t}Y_{i,t} = \sum_{j \in J} F_{i,j,t}Y_{j,t} + F_{i,s,t} + F_{i,t}^{\text{waste}} \quad \forall i \in I, t \in T, s = i \qquad (4.12)$$

where $F_{i,j,t}$, $F_{i,s,t}$, and $F_{i,t}^{\text{waste}}$ are the fractional amounts of process source *i* that are sent to process sink *j*, the respective storage tank *s* and waste storage in time interval *t* respectively.

Material balances for process sinks

Process sink *j* may accept flow from process source *i*, storage tank *s* and/or fresh resource *m*. Equation 4.13 presents the flow balance at the mixing point before sink *j* in time interval *t*.

$$F_{j,t}Y_{j,t} = \sum_{i \in I} F_{i,j,t}Y_{i,t} + \sum_{s \in S} F_{s,j,t} + \sum_{m \in M} F_{m,j,t} \quad \forall j \in J, t \in T \tag{4.13}$$

where $F_{s,j,t}$ and $F_{m,j,t}$ are the amounts from storage tank *s* and fresh resource *m* that are sent to sink *j* in time interval *t* respectively.

In addition to flow balance, property operator balance has to be considered for each sink *j*. Equation 4.14 shows the property operator balance for sink *j* in time interval *t*.

$$F_{j,t}Y_{j,t}\psi_j = \sum_{i \in I} F_{i,j,t}Y_{i,t}\psi_i + \sum_{s \in S} F_{s,j,t}\psi_s + \sum_{m \in M} F_{m,j,t}\psi_m \quad \forall j \in J, t \in T \tag{4.14}$$

where ψ_i, ψ_m and ψ_s are the property operators of source *i*, fresh resource *m* and the stored source, respectively.

Note that each process source *i* has its own designated storage tank *s*. Thus, the property operator of process source *i* in the respective storage tank *s* remains unchanged and is represented by the Equation 4.15.

$$\begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_s \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_i \end{bmatrix} \quad \forall i \in I \tag{4.15}$$

Material balances for fresh resource

Fresh resource m is sent to process sink j if it is needed but not to waste. The total amount of fresh resource m which is required for each batch cycle can be obtained via Equation 4.16.

$$F_m = \sum_{t \in T} \sum_{j \in J} F_{m,j,t} \quad \forall m \in M \tag{4.16}$$

Material balances for waste

Waste is flow from source i and storage tank s which are unable to reuse/recycle to process sink j. The total amount of waste in each batch cycle can be determined by Equation 4.17.

$$F^{\text{waste}} = \sum_{t \in T} \sum_{i \in I} F_{i,t}^{\text{waste}} + \sum_{t \in T} \sum_{s \in S} F_{s,t}^{\text{waste}} \tag{4.17}$$

where $F_{s,t}^{\text{waste}}$ is the amount from storage tank s that to waste storage in time interval t.

Material balances for storage tanks

Figure 4.5 shows the schematic representation for storage tank s. The input stream consists of the respective process source i from previous time interval t-1 while the output streams may be sent to process sink j or be discharged as waste in time interval t. Furthermore, some mass may be accumulated from the previous time interval as well as in the current time interval. The developed model considers storing the waste and not discharging the waste directly is to create more opportunities for energy recovery.

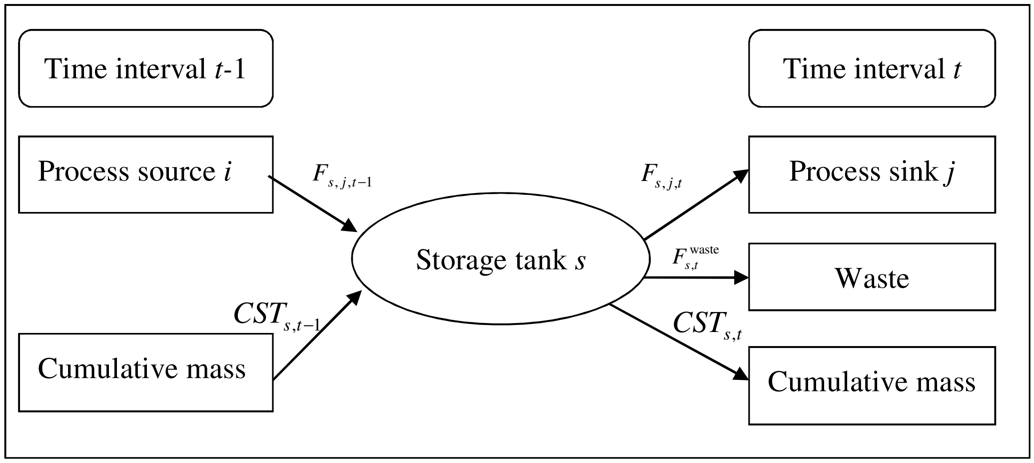

Figure 4.5: Schematic representation for storage tank *s*

The material balance for storage tank *s* is defined by Equation 4.18.

$$F_{i,s,t-1} + CST_{s,t-1} = \sum_{j \in J} F_{s,j,t} + F_{s,t}^{\text{waste}} + CST_{s,t} \quad \forall s \in S, t \in T, i \in I, s = i \tag{4.18}$$

$F_{i,s,t-1}$ represents the amount of process source *i* which is sent to the respective storage tank *s* in time interval *t*-1 while $CST_{s,t-1}$ and $CST_{s,t}$ are the cumulative mass in storage tank *s* in time interval *t*-1 and *t* respectively. Note that the presented model is also applicable to single batch process where the first cumulative mass in storage tank *s* will always take a value of zero as no stored source in storage tank *s* at the beginning stage.

4.5 HEAT EXCHANGERS NETWORK

In this study, the ATM that was developed by Foo (2010) is used to model the targeting for HEN. The targeting task of HEN is performed in every time intervals to ensure the overall targets for the given problem.

Firstly, process sources *i* and process sinks *j* are allocated in the respective time intervals to allow the performance of the HEN targeting task in every time interval. In each time interval, all possible streams that are connected from process sources and sinks to fresh resources, storage tanks and waste are identified based on the source-HEN-sink superstructure. These streams are either hot or cold streams. Hot streams are to be cooled and cold streams are to be heated. In this study, these hot and cold streams are pre-defined directly based on the supply and target temperatures of the given process source, process sink, fresh resource and waste temperature limiting data. For example, a stream that is segregated from a process source with a given temperature of 50°C to a process sink which has a given temperature of 80°C is categorized as a cold stream with supply and target temperatures of 50°C and 80°C respectively. The heat capacity for each hot and cold stream is assumed to be constant. In additional, external hot and cold utilities are available to fulfil the process requirement after maximising the energy recovery between the hot and cold streams.

To conduct ATM, cascade diagram for HEN as illustrated in Figure 4.6 is first generated for each time interval to determine the minimum external hot and cold utilities. As shown, a total of *n* temperatures is arranged vertically in descending order, from the highest temperature (q_o) to the lowest temperature (q_n). These temperatures are the shifted supply and target temperatures of all the hot and cold streams which are obtained by shifting the supply and target temperatures with a minimum temperature different (ΔT_{min}). This is important to ensure feasible heat transfer between the hot and cold streams.

For hot streams, the supply and target temperatures are shifted by subtracting $\frac{\Delta T_{min}}{2}$ as shown in Equations 4.19 and 4.20.

$$q_{Hot}^{ST} = T_{Hot}^{ST} - \frac{\Delta T_{min}}{2} \tag{4.19}$$

$$q_{\mathrm{Hot}}^{\mathrm{TT}} = T_{\mathrm{Hot}}^{\mathrm{TT}} - \frac{\Delta \mathrm{T}_{\mathrm{min}}}{2} \qquad (4.20)$$

where $T_{\mathrm{Hot}}^{\mathrm{ST}}$ and $T_{\mathrm{Hot}}^{\mathrm{TT}}$ are the supply and target temperatures for the hot stream respectively.

For cold streams, the supply and target temperatures are shifted by adding $\frac{\Delta \mathrm{T}_{\mathrm{min}}}{2}$ as shown by the following equations,

$$q_{Cold}^{ST} = T_{Cold}^{ST} + \frac{\Delta \mathrm{T}_{\mathrm{min}}}{2} \qquad (4.21)$$

$$q_{Cold}^{TT} = T_{Cold}^{TT} + \frac{\Delta \mathrm{T}_{\mathrm{min}}}{2} \qquad (4.22)$$

where T_{Cold}^{ST} and T_{Cold}^{TT} are the supply and target temperatures for the cold stream respectively.

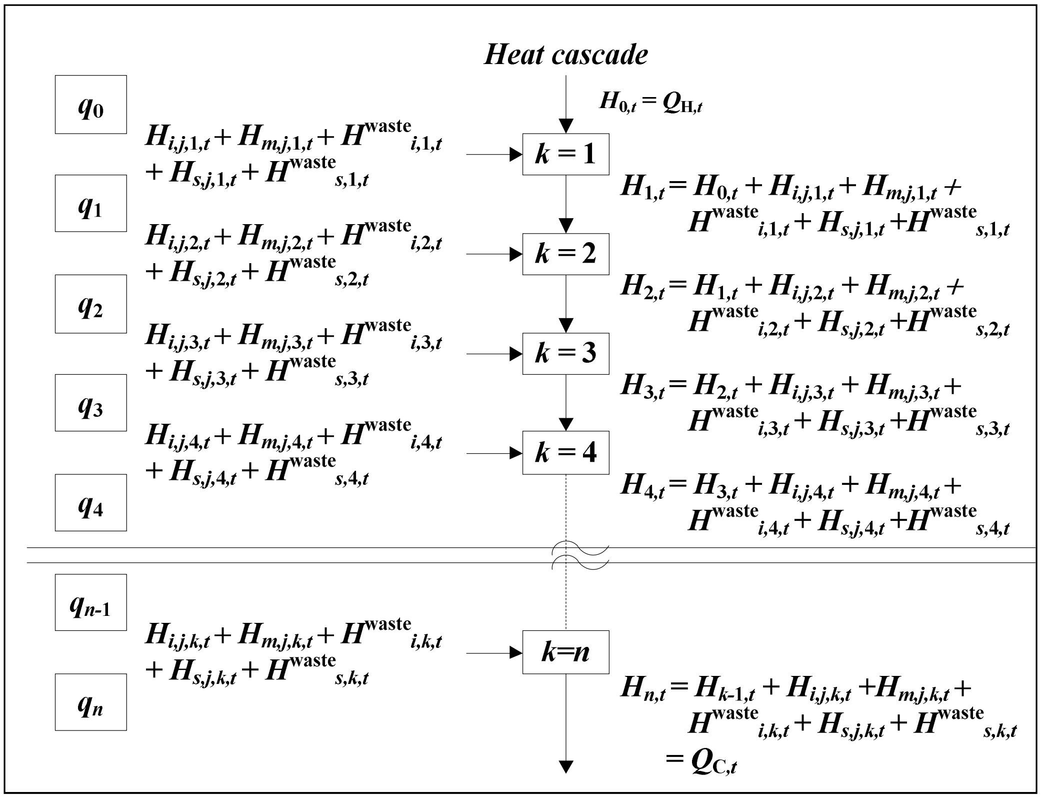

Figure 4.6: Generic cascade diagram for each time internal in a batch heat exchange network

As shown in Figure 4.6, a heat cascade is carried out across all temperature intervals ($q_k - q_{k-1}$) in all time intervals. In each time interval, hot streams that act as sources of heat transfer the heat to cold streams which act as sinks of heat in the same temperature interval to obtain heat recovery. In order to achieve heat balance in each temperature interval, any heat surplus in one temperature interval is cascaded down to the next temperature interval with lower temperatures. To performed heat cascade, the heat balance equations within each temperature interval k are developed to determine the interval heat flows of the streams based on the ratio of the duration of the time interval to that of the stream. Generally, the overall heat balance equation for each temperature interval k in time interval t can be expressed as:

$$H_{k,t} = H_{k-1,t} + \sum_{i\in I}\sum_{j\in J} H_{i,j,k,t} + \sum_{m\in M}\sum_{j\in J} H_{m,j,k,t} + \sum_{i\in I} H_{i,k,t}^{\text{waste}} + \sum_{s\in S}\sum_{j\in J} H_{s,j,k,t} + \sum_{s\in S} H_{s,k,t}^{\text{waste}} \quad \forall k\in K, t\in T \tag{4.23}$$

where $H_{k,t}$ and $H_{k-1,t}$ are the residual heat load within temperature interval *k* and *k*-1 in time interval *t*; $H_{i,j,k,t}$ and $H_{i,k,t}^{\text{waste}}$ are the heat loads for the streams from process source *i* that are sent to process sink *j* and waste within temperature interval *k* in time interval *t*; $H_{m,j,k,t}$ is the heat load for the streams from fresh resource *m* that is sent to process sink *j* within temperature interval *k* in time interval *t*; $H_{s,j,k,t}$ and $H_{s,k,t}^{\text{waste}}$ are the heat loads for the streams from storage tank *s* that are sent to process sink *j* and waste within temperature interval *k* in time interval *t*.

All of the above heat loads for the streams within temperature interval *k* in time interval *t* are determined from the product of the mass flow and the specific heat capacity of the streams as well as the temperature different of temperature interval *k* and are shown in Equations 4.24 – 4.29.

Heat balance for the streams from process source *i* to process sink *j*

Equation 4.24 presents the heat balance for the streams from process source *i* that are sent to process sink *j* within temperature interval *k* in time interval *t*.

$$H_{i,j,k,t} = \sum_{i\in I}\sum_{j\in J} F_{i,j,t}\mathrm{Cp}_i \Delta q_k \quad \forall k\in K, t\in T \tag{4.24}$$

where Cp_i is the specific heat capacity of process source *i*.

Heat balance for streams from fresh resource *m* to process sink *j*

The heat balance for the streams from fresh resource *m* that are sent to process sink *j* within temperature interval *k* in time interval *t* can be determined by Equation 4.25.

$$H_{m,j,k,t} = \sum_{m\in M}\sum_{j\in J} F_{m,j,t}\mathrm{Cp}_m \Delta q_k \quad \forall k \in K, t \in T \tag{4.25}$$

where Cp_m is the specific heat capacity of fresh resource *m*.

Heat balance for streams from process source *i* to waste storage

Equation 4.26 describes the heat balance for the streams from process source *i* that are sent to waste storage within temperature interval *k* in time interval *t*.

$$H_{i,k,t}^{\mathrm{waste}} = \sum_{i\in I} F_{i,t}^{\mathrm{waste}}\mathrm{Cp}_i \Delta q_k \quad \forall k \in K, t \in T \tag{4.26}$$

Heat balance for streams from stored source in storage tank*s* to process sink *j*

The heat balance for the streams from stored source in storage tank *s* that are sent to process sink *j* within temperature interval *k* in time interval *t* can be defined as follows.

$$H_{s,j,k,t} = \sum_{s\in S}\sum_{j\in J} F_{s,j,t}\mathrm{Cp}_s \Delta q_k \quad \forall k \in K, t \in T \tag{4.27}$$

where Cp_s is the specific heat capacity of the stored source.

As mentioned earlier, each process source *i* has its own respective storage tank *s*. Thus, the specific heat capacity of the process source *i*, Cp_i remains the same and it can be represented by the following equation.

$$\begin{bmatrix} Cp_1 \\ Cp_2 \\ \vdots \\ Cp_s \end{bmatrix} = \begin{bmatrix} Cp_1 \\ Cp_2 \\ \vdots \\ Cp_i \end{bmatrix} \quad \forall i \in I \tag{4.28}$$

Heat balance for streams from stored source in storage tank*s* to waste storage

The heat balance for the streams from stored source in storage tank *s* that are sent to waste storage within temperature interval *k* in time interval *t* can be achieved via Equation 4.29.

$$H_{s,k,t}^{\text{waste}} = \sum_{s \in S} F_{s,t}^{\text{waste}} Cp_s \Delta q_k \quad \forall k \in K, t \in T \tag{4.29}$$

Since the temperature range for each temperature interval *k* is difference, hence Equation 4.30 is used to determine the temperature different for each temperature interval *k*, Δq_k.

$$\Delta q_k = q_k - q_{k-1} \quad \forall k \in K \tag{4.30}$$

According to Foo (2010), the heat flow has to be a positive value to ensure the feasible of heat transfer. Therefore, Equation 4.31 is added to ensure the residual heat load is a non-negative value.

$$H_{k,t} \geq 0 \quad \forall k \in K, t \in T \tag{4.31}$$

In the heat cascade diagram (Figure 4.6), the external hot utility will be placed at the first temperature level. On the other hand, the external cold utility will be allocated at the lowest temperature level in the heat cascade diagram. Thus, the minimum hot and cold utilities for each time intervals are defined as follow,

$$Q_{\mathrm{H},t} = H_{0,t} \quad \forall t \in T \tag{4.32}$$

$$Q_{\mathrm{C},t} = H_{n,t} \quad \forall t \in T \tag{4.33}$$

where $H_{0,t}$ and $H_{n,t}$ are the heat loads at the first and final temperature levels in time interval t.

Equations 4.34 and 4.35 are included to determine the total required external hot (Q_{H}) and cold (Q_{C}) utilities for all time intervals in each batch cycle.

$$Q_{\mathrm{H}} = \sum_{t \in T} Q_{\mathrm{H},t} \tag{4.34}$$

$$Q_{\mathrm{C}} = \sum_{t \in T} Q_{\mathrm{C},t} \tag{4.35}$$

4.6 OBJECTIVE FUNCTION

This section presents the objective function of the proposed model which is to minimize the total annualised cost (TAC) of HIRCNs. The TAC for simultaneous targeting approach includes the total annual operating cost for fresh resources, hot and cold utilities as well as the annualised capital cost for storage tanks. It is expressed as follow,

$$\min \mathrm{TAC} = \mathrm{N_b}\left(\mathrm{C_m} F_m + \mathrm{C_{HU}} Q_\mathrm{H} + \mathrm{C_{CU}} Q_\mathrm{C}\right) + \sum_{s \in S} C_s \quad (4.36)$$

where C_m, C_{HU} and C_{CU} are the costs for fresh resources, hot and cold utilities. C_s is the annualised capital cost for storage tank*s*, and $\mathrm{N_b}$ is the number of batches per year.

The annualised capital cost for storage tank*s*, C_s (Seider *et al*. 2004) is given as,

$$C_s = \frac{\mathrm{I}}{\mathrm{I_b}}[\mathrm{A_0}(\mathrm{A_1} CST_s)^{\mathrm{d}}]AF \quad \forall s \in S \quad (4.37)$$

where A_0, A_1, and d are the cost parameters for storage tank *s* based on the chemical engineering plant cost index of 394. I and I_b are the chemical engineering plant cost index in the present time and base cost index. 10% of freeboard is assumed for storage tank *s* (Seider *et al*. 1999). It is also assumed that the cone-roofed carbon steel storage tanks are utilized. Therefore, A_0, A_1 and d are taken as 210, 1.10 and 0.51 respectively (Seider *et al.* 2004). CST_s is the capacity of storage tank *s* and *AF* is the annualizing factor.

The annualisation factor can be calculated by the following equation.

$$AF = \frac{\mathrm{A_s}(1+\mathrm{A_s})^{\mathrm{yr}}}{(1+\mathrm{A_s})^{\mathrm{yr}}-1} \tag{4.38}$$

where $\mathrm{A_s}$ is the annual fractional interest rate and yr is the time life of the storage tanks.

The capacity of storage tank *s*can be determined based on the input flow rate balance equation of the storage tank in time interval *t*-1 or the output flow rate balance of the storage tank in time interval *t* (refer to Figure 4.5). In this model, the output of storage tank is used to identify the capacity of storage tank *s* and is represent by Equation 4.39.

$$CST_s \geq \sum_{j\in J} F_{s,j,t} + F_{s,t}^{\mathrm{waste}} + CST_{s,t} \quad \forall s \in S, t \in T \tag{4.39}$$

Moreover, a positive value is taking into consideration for the capacity of each storage tank *s* (Foo 2010; Nun *et al*. 2011), as shown in Equation 4.40.

$$CST_s \geq 0 \quad \forall s \in S \tag{4.40}$$

On the other hand, the objective functions of the developed model for sequential targeting approach that are minimizing the TOC of fresh resource for property-based HIRCNs first and following by minimizing the cost of hot and cold utilities are shown in Equations 4.41 and 4.42:

$$\min \mathrm{TOC} = \mathrm{N_b C_m} F_m \tag{4.41}$$

$$\min \mathrm{TOC} = \mathrm{N_b}(\mathrm{C_{HU}} Q_{\mathrm{H}} + \mathrm{C_{CU}} Q_{\mathrm{C}}) \tag{4.42}$$

4.7 SUMMARY

The problem statement for this study has been clearly described in this chapter. Moreover, this chapter also describes the formulation of the mathematical model based on the proposed source-HEN-sink superstructure (Figure 4.1) that synthesizes property-based HIRCNs for batch processes. The proposed model is a set of mixed integer nonlinear programming model which is used to minimize the TAC for the HIRCNs. In this work, super-targeting approach is used to formulate the property-based RCN model while ATM is applied to develop the HEN model. This developed model is applied to three case studies to illustrate its effectiveness and capability and is presented in Chapter 5.

Chapter 5 Case Studies

5.1 INTRODUCTION

In this chapter, the developed property-based heat integrated resource conservation networks (HIRCNs) mathematical model for batch processes described in Chapter 5 is applied to three case studies. Each case study is solved using three different scenarios. The results of these scenarios are compared and analyzed. Besides, the HIRCNs involving the resource conservation network (RCN) and heat exchanger network (HEN) design for each case study are also presented. All these case studies are solved using Extended LINGO version 11.0 with Global Solver.

5.1.1 SCENARIOS

Three different scenarios are solved for each case study and these scenarios are explained as follow,

1. Scenario 1 - the developed HIRCN model is solved without the consideration of storage system to minimize the total annual operating cost (TOC) for fresh resources, external hot and cold utilities.

2. Scenario 2 - the developed HIRCN model is solved with consideration of storage system to minimize the total annualized cost (TAC) which includes the cost of fresh resources, external hot and cold utilities as well as the cost of storage tanks.

3. Scenario 3 - the developed HIRCN model is solved with consideration of storage system using two stages optimization where the minimization of TOC of fresh resources is first considered followed by minimizing the costs of

external hot and cold utilities. Then, the annualised capital cost of storage tanks is determined.

5.1.2 COSTING AND STREAM PARAMETERS

The following are the costing and stream parameters that are used to solve the case studies,

1. The capital cost of storage tank *s* is annualized with 10% of annual fractional interest rate over 6 years. Therefore, the annualizing factor, *AF* which is calculated by Equation 4.38 gives a value of 0.229.

2. The latest chemical engineering (CE) plant cost index of 572.7 is used. The update factor equals to latest CE plant cost index divides by CE plant base cost index.

3. The costs of fresh resource, C_m, hot utility, C_{HU}, and cold utility, C_{CU} are taken as $ 0.001/kg (Ng *et al*. 2010), $ 0.017/kW.hr and $ 0.006/kW.hr (Isafiade and Fraser, 2010).

4. The minimum temperature difference (ΔT_{min}) is assumed as 10^{o}C.

5. The specific heat capacities of all streams (Cp_i, Cp_s and Cp_m) are assumed as 4.2 kJ/kg.^{o}C.

6. The density of process streams is assumed as 1000 kg/m^3 as majority is water.

5.2 CASE STUDY 1

Case Study 1 is a case study modified from Example 3 in Wang and Smith (1995) with the limiting data shown in Table 5.1. There are three process sources and three process sinks. The concentration and supply temperature of the fresh resource are 0 ppm and 20ºC, respectively. The waste is discharged at 30ºC. It is assumed that the cycle time for a batch cycle is 2 hours and the annual operating days is 330 days. Hence, the number of batches per year, N_b is 3,960.

Table 5.1: Limiting data for Case Study 1

Sink	Amount (ton)	Concentration (ppm)	Temperature (ºC)	Time (hr)	
				Start	End
SK1	100	100	80	0	1.5
SK2	40	0	50	0.5	2.0
SK3	25	100	40	1.5	2.0
Source	**Amount (ton)**	**Concentration (ppm)**	**Temperature (ºC)**	**Time (hr)**	
				Start	**End**
SR1	100	400	40	0.5	2.0
SR2	40	200	60	0	1.5
SR3	25	200	80	1.0	2.0

Firstly, the time intervals for sources and sinks are arranged in ascending order. Based on the data presented in Table 5.1, the time intervals are t_0 = 0 hr, t_1 = 0.5 hr, t_2 = 1.0 hr, t_3 = 1.5 hr and t_4 = 2 hr. Four time intervals are found for this case study which is time interval 1, 2, 3 and 4. Figure 5.1 describes the time intervals for Case Study 1.

Next, stream parameterisation model described in Section 4.4 of Chapter 4 is initially solved to identify the existence of process sources and sinks in each time interval. After that, the sources and sinks are allocated in the respective time intervals. Then a revised source-HEN-sink superstructure for each time interval is established. Figure 5.2 shows the revised source-HEN-sink superstructure for each time interval in Case Study 1.

Based on Figure 5.2, all the hot and cold streams in each time interval can be defined and is classified in Table 5.2. As mentioned in Section 4.5, all the hot and cold streams are pre-defined based on the given limiting data of the process source, process sink, fresh resource and waste. For example, the $F_{\mathrm{SR2,SK1,1}}$ stream that is connected between process source SR2 and process sink SK1 at time interval 1 is categorized as a cold stream with supply and target temperatures of 60°C and 80°C respectively. With the data in Table 5.2, the HEN cascade diagram for each time interval is constructed (Figures 5.3 to 5.6). Figures 5.3 to 5.6 are generated based on Equations 4.23– 4.30 and 4.32 – 4.33. Besides, the temperatures that are arranged vertically in Figures 5.3 to 5.6 are the shifted supply and target temperatures for hot and cold streams that are determined via Equations 4.19 – 4.22.

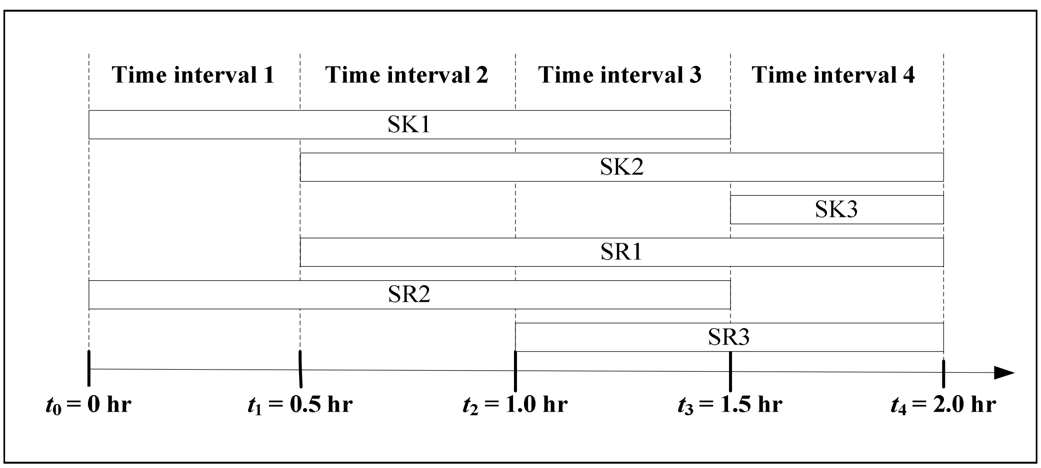

Figure 5.1: A diagram describing time intervals for Case Study 1

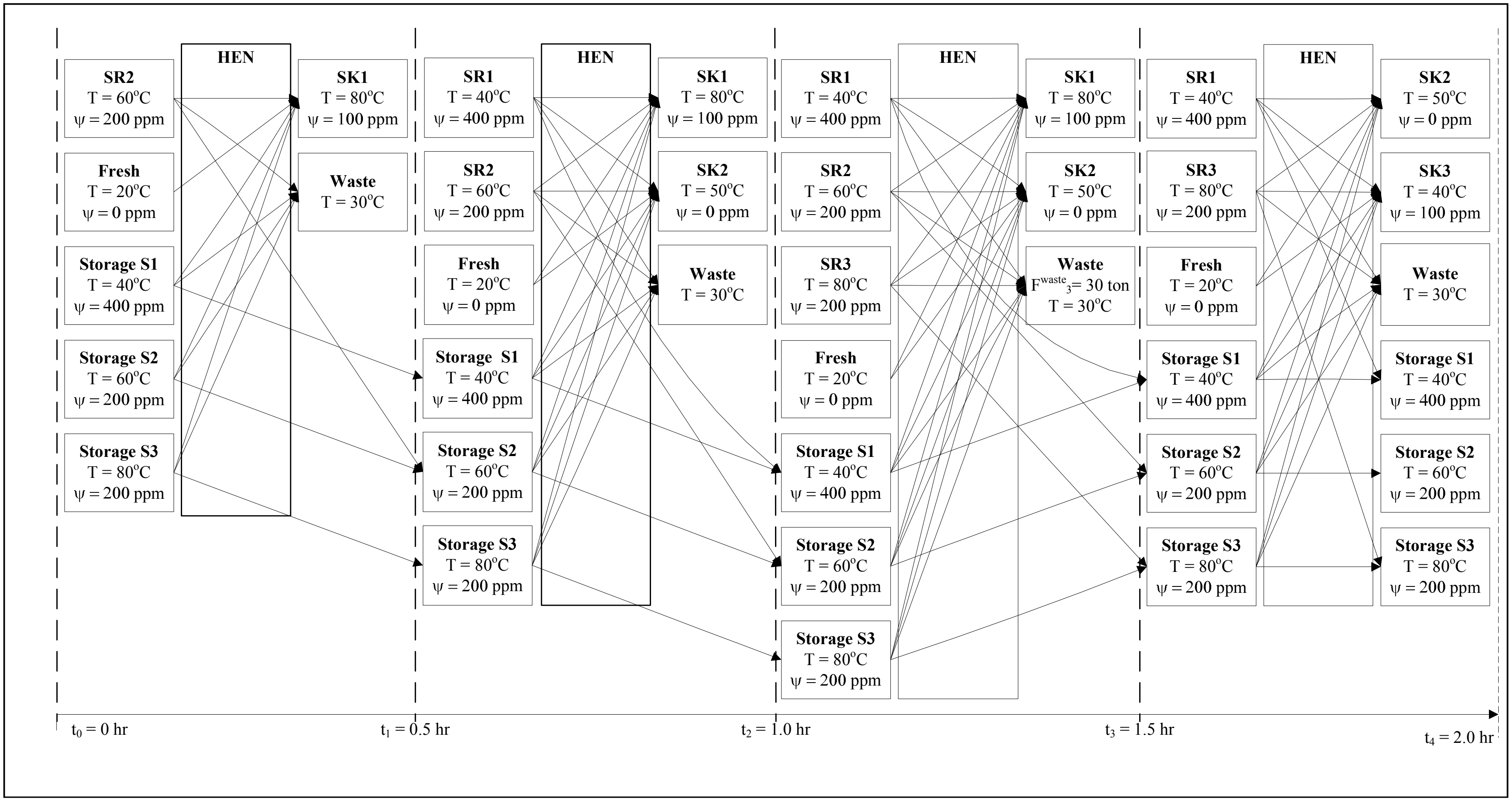

Figure 5.2: Revised source-HEN-sink superstructure for each time intervals in Case Study 1

Table 5.2: Extracted hot and cold streams for Case Study 1

Time interval	Cold streams	Hot streams
Time interval 1 (0 – 0.5 hr)	$F_{SR2,SK1,1}$, $F_{FR,SK1,1}$, $F_{S1,SK1,1}$, $F_{S2,SK1,1}$	$F^{waste}_{SR2,1}$, $F^{waste}_{S1,1}$, $F^{waste}_{S2,1}$, $F^{waste}_{S3,1}$
Time interval 2 (0.5 – 1.0 hr)	$F_{SR1,SK1,2}$, $F_{SR1,SK2,2}$, $F_{SR2,SK1,2}$, $F_{FR,SK1,2}$, $F_{FR,SK2,2}$, $F_{S1,SK2,2}$, $F_{S1,SK1,2}$, $F_{S2,SK1,2}$	$F^{waste}_{SR1,2}$, $F_{SR2,SK2,2}$, $F^{waste}_{SR2,2}$, $F^{waste}_{S1,2}$, $F^{waste}_{S2,2}$, $F_{S2,SK2,2}$, $F_{S3,SK2,2}$, $F^{waste}_{S3,2}$
Time interval 3 (1.0 - 1.5 hr)	$F_{SR1,SK1,3}$, $F_{SR1,SK2,3}$, $F_{SR2,SK1,3}$, $F_{FR,SK1,3}$, $F_{FR,SK2,3}$, $F_{S1,SK1,3}$, $F_{S1,SK2,3}$, $F_{S2,SK1,3}$	$F^{waste}_{SR1,3}$, $F_{SR2,SK2,3}$, $F^{waste}_{SR2,3}$, $F_{SR3,SK2,3}$, $F^{waste}_{SR3,3}$, $F^{waste}_{S1,3}$, $F_{S2,SK2,3}$, $F^{waste}_{S2,3}$, $F_{S3,SK2,3}$, $F^{waste}_{S3,3}$
Time interval 4 (1.5 – 2.0 hr)	$F_{SR1,SK2,4}$, $F_{FR,SK2,4}$, $F_{FR,SK3,4}$, $F_{S1,SK2,4}$	$F^{waste}_{SR1,4}$, $F_{SR3,SK2,4}$, $F_{SR3,SK3,4}$, $F^{waste}_{SR3,4}$, $F^{waste}_{S1,4}$, $F_{S2,SK2,4}$, $F_{S2,SK3,4}$, $F^{waste}_{S2,4}$, $F_{S3,SK2,4}$, $F_{S3,SK3,4}$, $F^{waste}_{S3,4}$

Heat cascade

85°C $\quad H_{0,1} = Q_{H,1}$

$k = 1$

75°C $\quad H_{1,1} = H_{0,1} + H_{SR2,SK1,1,1} + H_{FR,SK1,1,1} + H_{S1,SK1,1,1} + H_{S2,SK1,1,1}$

$k = 2$

65°C $\quad H_{2,1} = H_{1,1} + H_{SR2,SK1,2,1} + H_{FR,SK1,2,1} + H_{S1,SK1,2,1} + H_{S2,SK1,2,1} + H^{waste}_{S3,2,1}$

$k = 3$

55°C $\quad H_{3,1} = H_{2,1} + H_{FR,SK1,3,1} + H_{S1,SK1,3,1} + H^{waste}_{S3,3,1}$

$k = 4$

45°C $\quad H_{4,1} = H_{3,1} + H^{waste}_{SR2,4,1} + H_{FR,SK1,4,1} + H_{S1,SK1,4,1} + H^{waste}_{S2,4,1} + H^{waste}_{S3,4,1}$

$k = 5$

35°C $\quad H_{5,1} = H_{4,1} + H^{waste}_{SR2,5,1} + H_{FR,SK1,5,1} + H^{waste}_{S2,5,1} + H^{waste}_{S3,5,1}$

$k = 6$

25°C $\quad H_{6,1} = H_{5,1} + H^{waste}_{SR2,6,1} + H_{FR,SK1,6,1} + H^{waste}_{S1,6,1} + H^{waste}_{S2,6,1} + H^{waste}_{S3,6,1} = Q_{C,1}$

Figure 5.3: Heat cascade diagram for time interval 1 in Case Study 1

Heat cascade

85°C $H_{0,2} = Q_{H,2}$

$k = 1$

75°C $H_{1,2} = H_{0,2} + H_{SR1,SK1,1,2} + H_{SR2,SK1,1,2} + H_{FR,SK1,1,2} + H_{S1,SK1,1,2} + H_{S2,SK1,1,2}$

$k = 2$

65°C $H_{2,2} = H_{1,2} + H_{SR1,SK1,2,2} + H_{SR2,SK1,2,2} + H_{FR,SK1,2} + H_{S1,SK1,2,2} + H_{S2,SK1,2,2} + H_{S3,SK2,2,2} + H^{waste}_{S3,2,2}$

$k = 3$

55°C $H_{3,2} = H_{2,2} + H_{SR1,SK1,3,2} + H_{FR,SK1,3,2} + H_{S1,SK1,3,2} + H_{S3,SK2,3,2} + H^{waste}_{S3,3,2}$

$k = 4$

45°C $H_{4,2} = H_{3,2} + H_{SR1,SK1,4,2} + H_{SR1,SK2,4,2} + H_{SR2,SK2,4,2} + H^{waste}_{SR2,4,2} + H_{FR,SK1,4,2} + H_{FR,SK2,4,2} + H_{S1,SK1,4,2} + H_{S1,SK2,4,2} + H_{S2,SK2,4,2} + H^{waste}_{S2,4,2} + H_{S3,SK2,4,2} + H^{waste}_{S3,4,2}$

$k = 5$

35°C $H_{5,2} = H_{4,2} + H^{waste}_{SR2,5,2} + H_{FR,SK1,5,2} + H_{FR,SK2,5,2} + H^{waste}_{S2,5,2} + H^{waste}_{S3,5,2}$

$k = 6$

25°C $H_{6,2} = H_{5,2} + H^{waste}_{SR1,6,2} + H^{waste}_{SR2,6,2} + H_{FR,SK1,6,2} + H_{FR,SK2,6,2} + H^{waste}_{S1,6,2} + H^{waste}_{S2,6,2} + H^{waste}_{S3,6,2} = Q_{C,2}$

Figure 5.4: Heat cascade diagram for time interval 2 in Case Study 1

Heat cascade

85°C $H_{0,3} = Q_{H,3}$

$k = 1$

75°C $H_{1,3} = H_{0,3} + H_{SR1,SK1,1,3} + H_{SR2,SK1,1,3} + H_{FR,SK1,1,3} + H_{S1,SK1,1,3} + H_{S2,SK1,1,3}$

$k = 2$

65°C $H_{2,3} = H_{1,3} + H_{SR1,SK1,2,3} + H_{SR2,SK1,2,3} + H_{SR3,SK2,2,3} + H^{waste}_{SR3,2,3} + H_{FR,SK1,2,3} + H_{S1,SK1,2,3} + H_{S2,SK1,2,3} + H_{S3,SK2,2,3} + H^{waste}_{S3,2,3}$

$k = 3$

55°C $H_{3,3} = H_{2,3} + H_{SR1,SK1,3,3} + H_{SR3,SK2,3,3} + H^{waste}_{SR3,3,3} + H_{FR,SK1,3,3} + H_{S1,SK1,3,3} + H_{S3,SK2,3,3} + H^{waste}_{S3,3,3}$

$k = 4$

45°C $H_{4,3} = H_{3,3} + H_{SR1,SK1,4,3} + H_{SR1,SK2,4,3} + H_{SR2,SK2,4,3} + H^{waste}_{SR2,4,3} + H_{SR3,SK2,4,3} + H^{waste}_{SR3,4,3} + H_{FR,SK1,4,3} + H_{FR,SK2,4,3} + H_{S1,SK1,4,3} + H_{S1,SK2,4,3} + H_{S2,SK2,4,3} + H^{waste}_{S2,4,3} + H_{S3,SK2,4,3} + H^{waste}_{S3,4,3}$

$k = 5$

35°C $H_{5,3} = H_{4,3} + H^{waste}_{SR2,5,3} + H^{waste}_{SR3,5,3} + H_{FR,SK1,5,3} + H_{FR,SK2,5,3} + H^{waste}_{S2,5,3} + H^{waste}_{S3,5,3}$

$k = 6$

25°C $H_{6,3} = H_{5,3} + H^{waste}_{SR1,6,3} + H^{waste}_{SR2,6,3} + H^{waste}_{SR3,6,3} + H_{FR,SK1,6,3} + H_{FR,SK2,6,3} + H^{waste}_{S1,6,3} + H^{waste}_{S2,6,3} + H^{waste}_{S3,6,3} = Q_{C,3}$

Figure 5.5: Heat cascade diagram for time interval 3 in Case Study 1

Heat cascade

75°C $H_{0,4} = Q_{H,4}$

$k = 1$

55°C $H_{1,4} = H_{0,4} + H_{\mathrm{SR3,SK2,1,4}} + H_{\mathrm{SR3,SK3,1,4}} + H^{\mathrm{waste}}_{\mathrm{SR3,1,4}} + H_{\mathrm{S3,SK2,1,4}} + H_{\mathrm{S3,SK3,1,4}} + H^{\mathrm{waste}}_{\mathrm{S3,1,4}}$

$k = 2$

45°C $H_{2,4} = H_{1,4} + H_{\mathrm{SR1,SK2,2,4}} + H_{\mathrm{SR3,SK2,2,4}} + H_{\mathrm{SR3,SK3,2,4}} + H^{\mathrm{waste}}_{\mathrm{SR3,2,4}} + H_{\mathrm{FR,SK2,2,4}} + H_{\mathrm{S1,SK2,2,4}} + H_{\mathrm{S2,SK2,2,4}} + H_{\mathrm{S2,SK3,2,4}} + H^{\mathrm{waste}}_{\mathrm{S2,2,4}} + H_{\mathrm{S3,SK2,2,4}} + H_{\mathrm{S3,SK3,2,4}} + H^{\mathrm{waste}}_{\mathrm{S3,2,4}}$

$k = 3$

35°C $H_{3,4} = H_{2,4} + H_{\mathrm{SR3,SK3,3,4}} + H^{\mathrm{waste}}_{\mathrm{SR3,3,4}} + H_{\mathrm{FR,SK2,3,4}} + H_{\mathrm{FR,SK3,3,4}} + H_{\mathrm{S2,SK3,3,4}} + H^{\mathrm{waste}}_{\mathrm{S2,3,4}} + H_{\mathrm{S3,SK3,3,4}} + H^{\mathrm{waste}}_{\mathrm{S3,3,4}}$

$k = 4$

25°C $H_{4,4} = H_{3,4} + H^{\mathrm{waste}}_{\mathrm{SR1,4,4}} + H^{\mathrm{waste}}_{\mathrm{SR3,4,4}} + H_{\mathrm{FR,SK2,4,4}} + H_{\mathrm{FR,SK3,4,4}} + H^{\mathrm{waste}}_{\mathrm{S1,4,4}} + H^{\mathrm{waste}}_{\mathrm{S2,4,4}} + H^{\mathrm{waste}}_{\mathrm{S3,4,4}} = Q_{C,4}$

Figure 5.6: Heat cascade diagram for time interval 4 in Case Study 1

Scenario 1

For Scenario 1, the objective function is to minimize the TOC of fresh resource as well as hot and cold utilities in property-based HIRCNs without storage system across all time intervals.

The objective function is given as follows:

$$\min \mathrm{TOC} = \mathrm{N_b}(\mathrm{C_m} F_m + \mathrm{C_{HU}} Q_{\mathrm{H}} + \mathrm{C_{CU}} Q_{\mathrm{C}}) \tag{5.1}$$

Since storage system is not present in Scenario 1, process source will not be sent to any storage tank. Thus, additional constraint is added as follow:

$$F_{i,s,t} = 0 \quad \forall i \in I, s \in S, s = i, t \in T \tag{5.2}$$

Equation 5.1 is solved subject to constraints in Equations 4.3 - 4.18, 4.23 – 4.26, 4.30 – 4.35 and 5.2, yields the minimum TOC of $ 747,098/year. The obtained fresh resources as well as the hot and cold utilities are 107.5 ton/batch, 4,647 kWh/batch and 360 kWh/batch respectively. Figures 5.7 and 5.8 show the optimal HIRCNs and HEN for Scenario 1 in Case Study 1.

The HEN is synthesized using the classical pinch design method (Linnhoff and Flower, 1978; Linnhoff *et al.* 1982; Smith 2005) which is used to verify the obtained hot and cold utilities. The LINGO code formulations and solutions for Scenario 1 of Case Study 1 are presented in Appendix A.1.1.

Scenario 2

For Scenario 2, the optimization objective is set to minimize TAC for property-based HIRCNs with storage system which includes the TOC of fresh resources, hot and cold utilities as well as the annualised capital cost for storage tanks. Therefore, Equation 4.36 is solved subject to constraints in Equations 4.3 - 4.18, 4.23 – 4.35 and 4.37 – 4.40. The optimal HIRCNs and HEN for Scenario 2 in Case Study 1 are shown in Figures 5.9 and 5.10 respectively. The minimum TAC for Scenario 2 is determined as $ 702,208/year while the requirements of fresh resource, hot and cold utilities as well as storage tanks for Scenario 2 are 102.5 ton/batch, 4,229 kWh/batch, 0 kWh/batch, 16.66 tons, 5.42 tons and 6.67 tons respectively. Appendix A.1.2 shows the LINGO code formulations and solutions for Scenario 2 of Case Study 1.

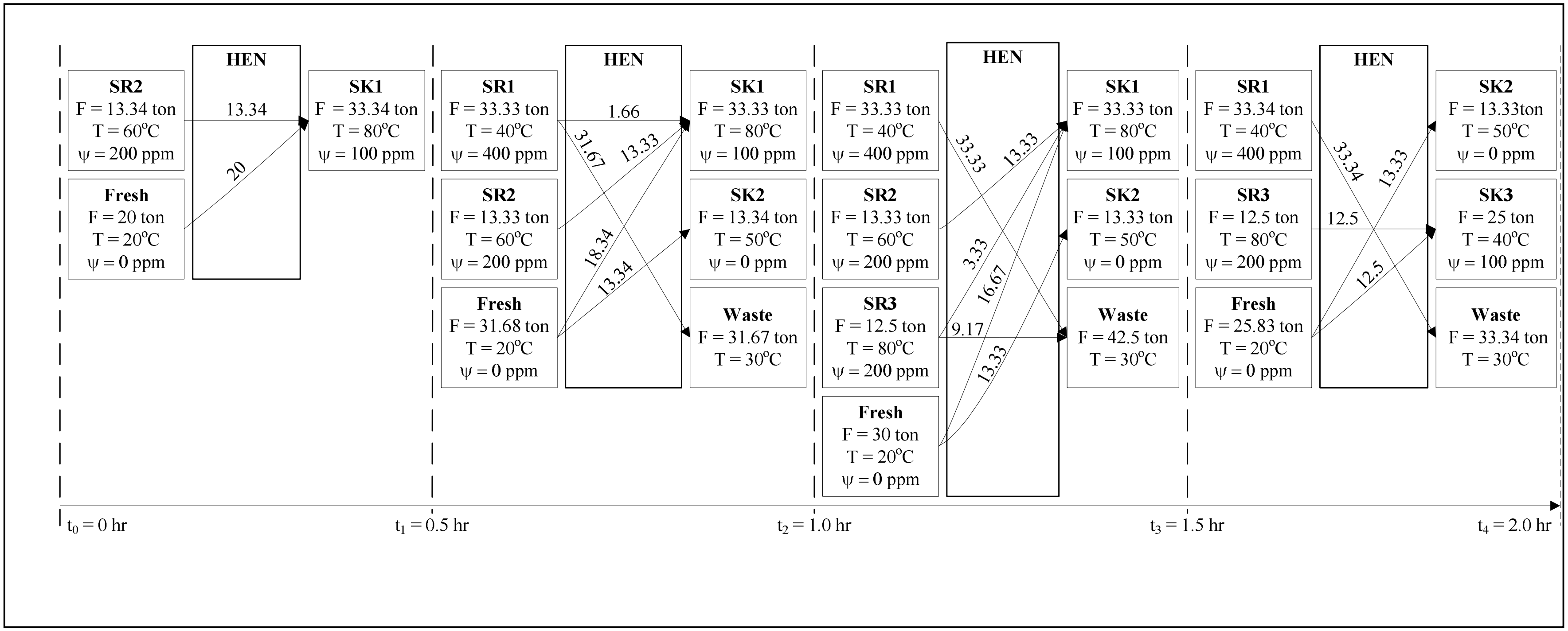

Figure 5.7: Property-based HIRCNs for Scenario 1 of Case Study 1 (All flow terms given in tons)

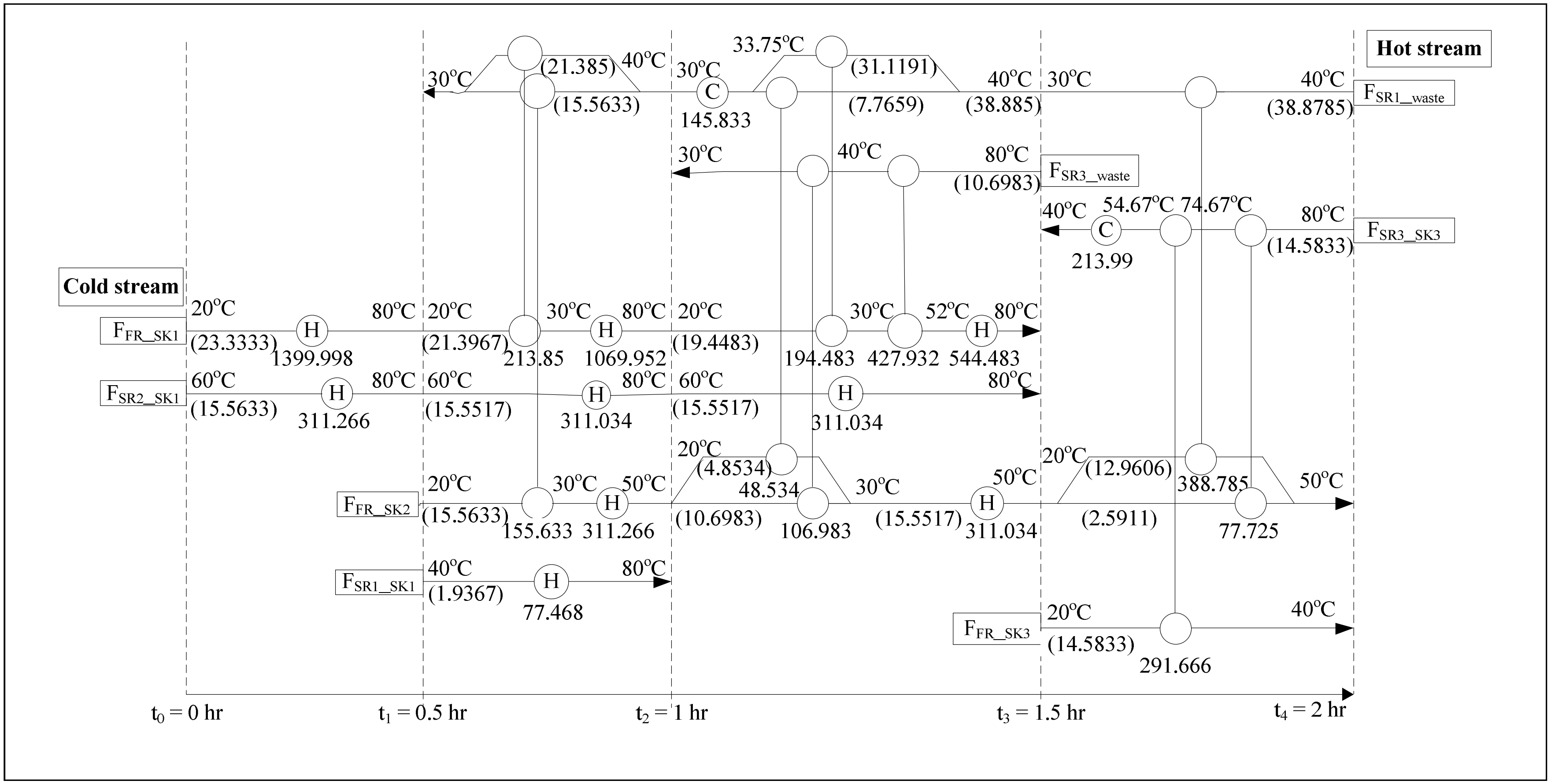

Figure 5.8: HEN for Scenario 1 of Case Study 1 (All heat terms given in kilowatts of heat, kWh; values shown in parentheses represent F*Cp, kWh/°C)

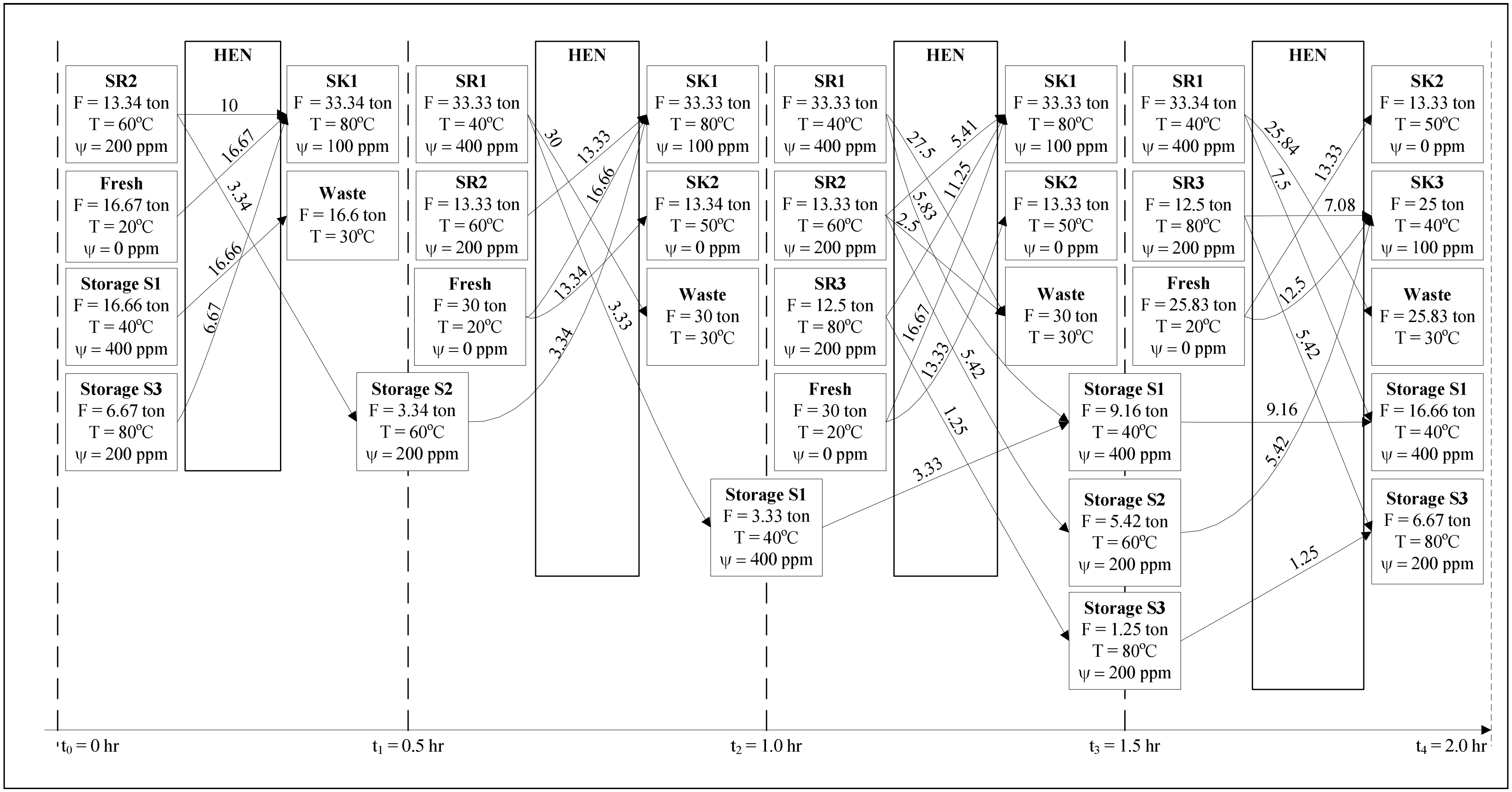

Figure 5.9: Property-based HIRCNs for Scenario 2 of Case Study 1 (All flow terms given in tons)

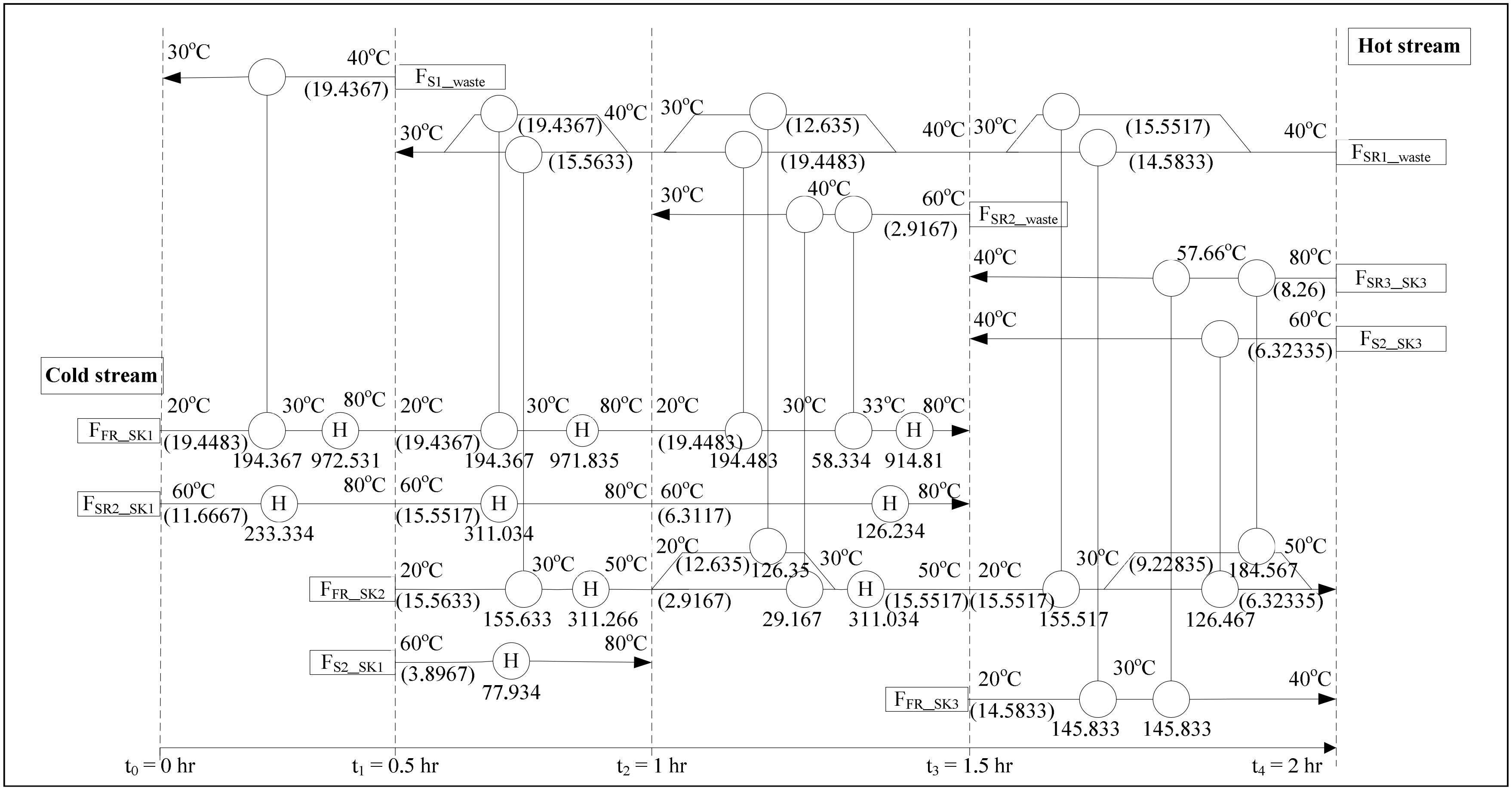

Figure 5.10: HEN for Scenario 2 of Case Study 1 (All heat terms given in kilowatts of heat, kWh; values shown in parentheses represent F*Cp, kWh/°C)

Scenario 3

In Scenario 3, Case Study 1 is solved via a two-stage optimization approach. In first stage, Equation 4.41 is solved subject to the constraints given in Equations 4.3 - 4.18 which yields F_m of 102.5 ton/batch. This is then added as a new constraint to the problem in the second stage, i.e.,

$$F_m = 102.5 \tag{5.3}$$

Then, solving Equation 4.42 subject to constraints in Equations 4.3 – 4.18, 4.23 – 4.35, 4.37 – 4.40 and 5.3 yields Q_{H} and Q_{C} of 4,229 kWh/batch and 0 kWh/batch respectively. Moreover, three storage tanks with the capacities of 14.17 tons, 9.17 tons and 5.42 tons are needed. The optimal HIRCNs and HEN for Scenario 3 of Case Study 1 are shown in Figures 5.11 and 5.12 respectively. The LINGO code formulations and solutions for Scenario 3 of Case Study 1 are provided in Appendix A.2.

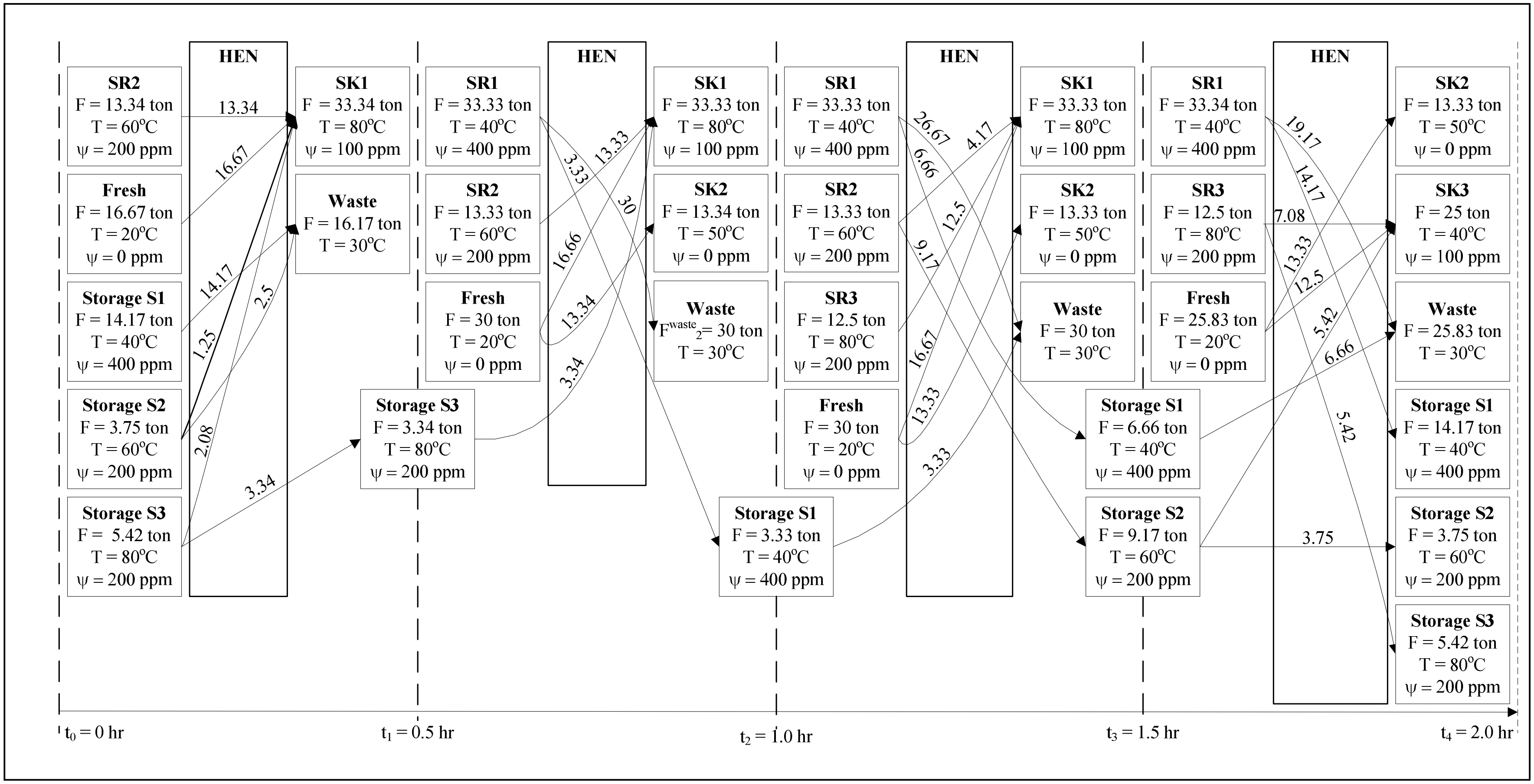

Figure 5.11: Property-based HIRCNs for Scenario 3 of Case Study 1

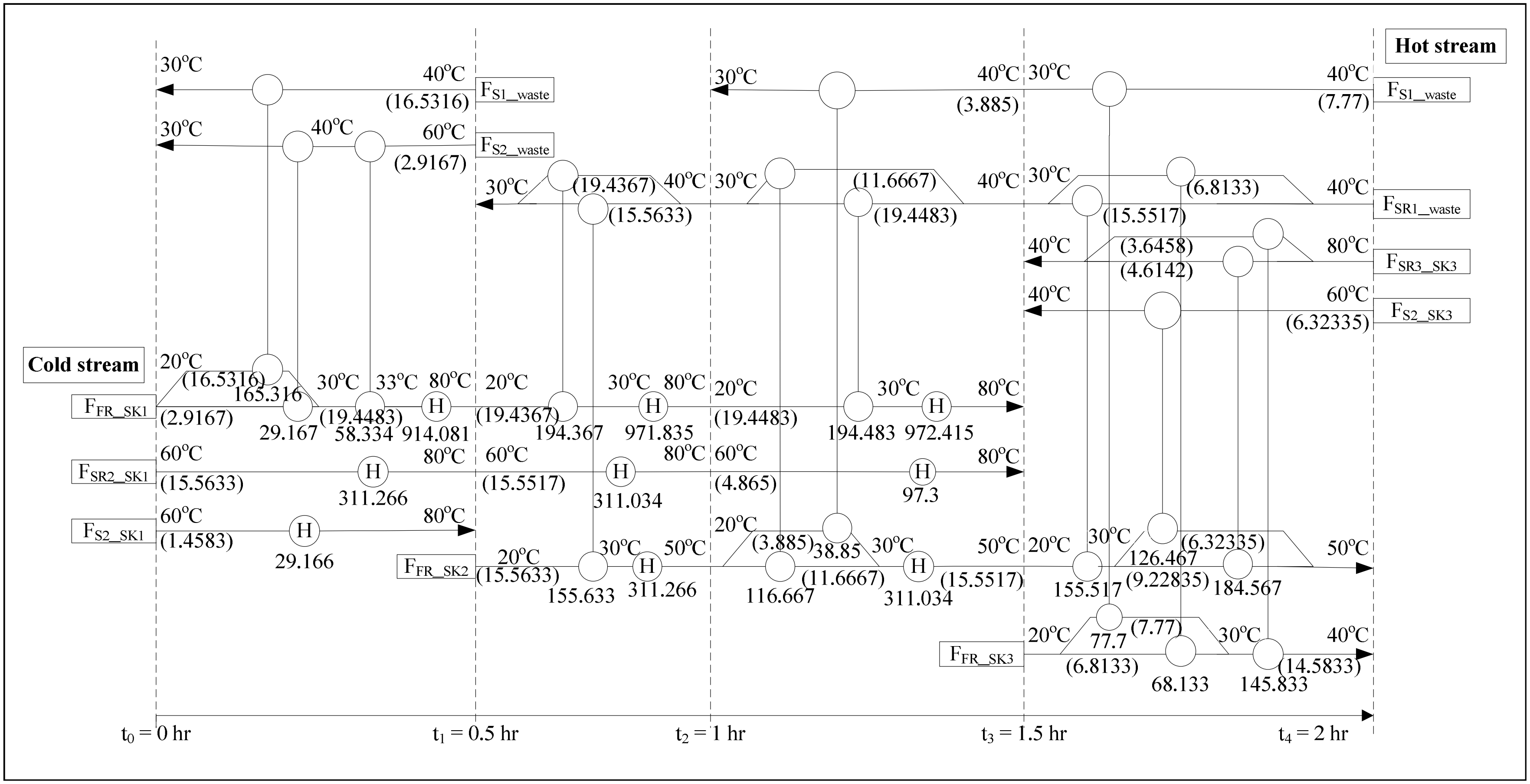

Figure 5.12: HEN for Scenario 3 of Case Study 1 (All heat terms given in kilowatts of heat, kWh; values shown in parentheses represent F*Cp, kWh/°C)

Comparisons between Scenarios 1, 2 and 3 for Case Study 1

Results for Scenarios 1, 2 and 3 of Case Study 1 are summarized and presented in Table 5.3. As shown, Scenario 2 obtained the lowest TAC which is $702,208/year followed by Scenario 3 ($702,372/year) and Scenario 1 ($747,098/year). The TAC for Scenario 2 is 6.39% and 0.02% lesser than the TAC for Scenarios 1 and 3 respectively. With the presence of storage system, Scenarios 2 and 3 require 4.87% lesser fresh resource, 9.88% lesser hot utility and 100% lesser cold utility compared to Scenario 1. Then, Scenarios 2 and 3 only require $11,600/year and $11,764/year respectively for the capital costs of storage tanks. However, the application of storage system is able to save $44,890/year of the TAC for Scenario 2 and $44,726/year of the TAC for Scenario 3. On the other hand, Scenarios 2 and 3 required the same amount of fresh resources and external energy utilities which are 102.5 tons of fresh resource, 4229 kWh/batch of hot utility and 0 kWh/batch of cold utility. However, the annual capital cost of storage tanks for Scenario 3 is 1.41% higher than Scenario 2. It can be seen that the capacities of storage tanks S1 and S3 for Scenario 2 are 2.5 tons and 1.25 tons higher than Scenario 3. However, the capacity of storage tank S2 for Scenario 2 is 3.75 tons lower than Scenario 3.

Table 5.3: Summary of results for Scenarios 1, 2 and 3 of Case Study 1

Concept	**Scenario 1**	**Scenario 2**	**Scenario 3**
Fresh resource, F_m (ton/batch)	107.5	102.5	102.5
Hot utility, Q_{H} (kWh/batch)	4,647	4,229	4,229
Cold utility, Q_{C} (kWh/batch)	360	0	0
Capacity for storage tank S1, CST_{S1} (tons)	-	16.67	14.17
Capacity for storage tank S2, CST_{S2} (tons)	-	5.42	9.17
Capacity for storage tank S3, CST_{S3} (tons)	-	6.67	5.42
Annual capital cost of storage tanks ($/year)	-	11,600	11,764
Total operating cost, TOC ($/year)	747,098	690,608	690,608
Total annualized cost, TAC ($/year)	747,098	702,208	702,372

5.3 CASE STUDY 2

Case Study 2 is a modified case study which is adopted from Bagajewicz *et al*. (2002). In this case study, there are three process sources and three process sinks, with the limiting data shown in Table 5.4. The concentration and supply temperature of fresh resource is 0 ppm and 20°C respectively. The waste must be released at 30°C. A batch cycle of 4 hours and the annual operating days of 330 days are assumed. Thus, the number of batches per year, N_b is 1,980.

Table 5.4: Limiting data for Case Study 2

Sink	Amount (ton)	Concentration (ppm)	Temperature (ºC)	Time (hr)	
				Start	End
SK1	360	50	100	1	3
SK2	144	50	75	0	4
SK3	600.12	800	100	1	3
Source	**Amount (ton)**	**Concentration (ppm)**	**Temperature (ºC)**	**Time (hr)**	
				Start	**End**
SR1	360	100	100	0	2
SR2	144	800	75	2	4
SR3	600.12	1100	100	1	3

Based on the limiting data in Table 5.4, the time intervals for sources and sinks are arranged in ascending order (t_0 = 0 hr, t_1 = 1 hr, t_2 = 2 hr, t_3 = 3 hr and t_4 = 4 hr). Four time intervals which are time intervals 1, 2, 3 and 4 are found and these time intervals are shown in Figure 5.13.

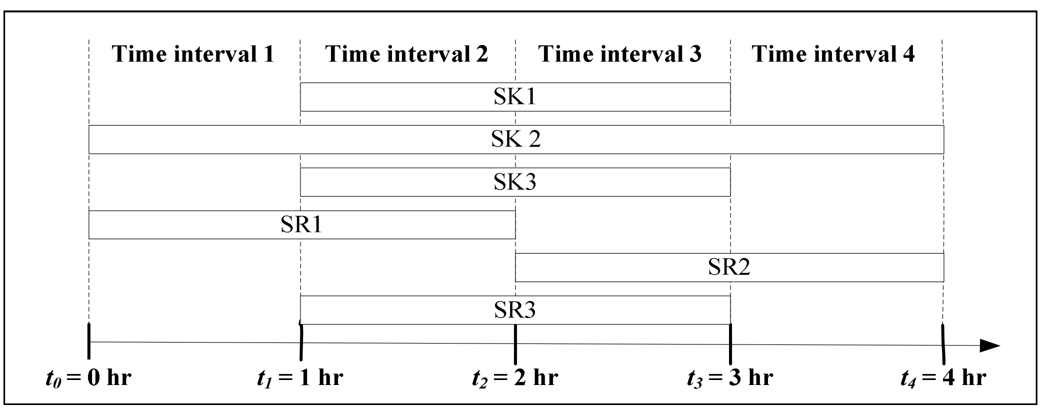

Figure 5.13: A diagram describing time intervals for Case Study 2

The stream parameterisation model is first solved to determine the existence of process sources and sinks in the corresponding time interval. Then, possible source-HEN-sink superstructure in each time interval as shown in Figure 5.14 is found. Similar to Case Study 1, the process streams in each time intervals are categorized as hot or cold streams according to the supply and target temperature of the given fresh resource, waste, process sources and sinks. These extracted hot and cold streams are summarized in Table 5.5. Next, these streams are used to establish the HEN cascade diagram for each time interval as shown in Figures 5.15 to 5.18. The procedure to construct Figures 5.15 to 5.18 is similar to that in Case Study 1 where Equations 4.23 – 4.30 and 4.32 – 4.33 are used. On the other hand, the temperatures that are presented in descending order in Figures 5.15 to 5.18 are shifted supply and target temperatures for hot and cold streams.

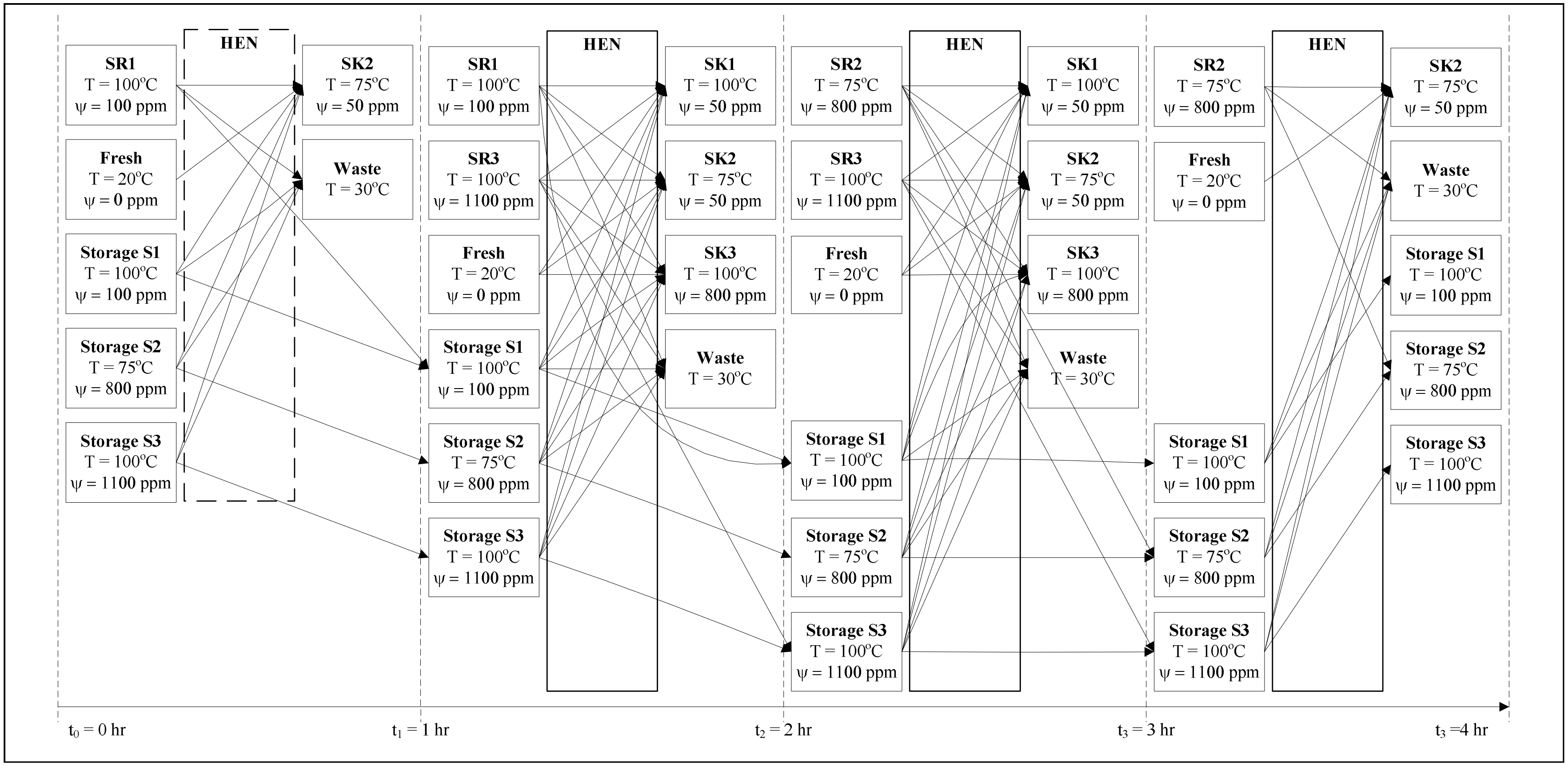

Figure 5.14: Revised source-HEN-sink superstructure for each time intervals in Case Study 2

Table 5.5: Extracted hot and cold streams for Case Study 2

Time interval	Cold streams	Hot streams
Time interval 1 **(0 – 0.5 hr)**	$F_{FR,SK2,1}$	$F_{SR1,SK2,1}$, $F^{waste}_{SR1,1}$, $F_{S1,SK2,1}$, $F^{waste}_{S1,1}$, $F^{waste}_{S2,1}$, $F_{S3,SK2,1}$, $F^{waste}_{S3,1}$
Time interval 2 **(0.5 – 1.0 hr)**	$F_{FR,SK1,2}$, $F_{FR,SK2,2}$, $F_{FR,SK3,2}$, $F_{S2,SK1,2}$, $F_{S2,SK3,2}$	$F_{SR1,SK2,2}$, $F^{waste}_{SR1,2}$, $F_{SR3,SK2,2}$, $F^{waste}_{SR3,2}$, $F_{S1,SK2,2}$, $F^{waste}_{S1,2}$, $F_{S2,SK1,2}$, $F_{S2,SK3,2}$, $F^{waste}_{S2,2}$, $F_{S3,SK2,2}$, $F^{waste}_{S3,2}$
Time interval 3 **(1.0 - 1.5 hr)**	$F_{SR2,SK1,3}$, $F_{SR2,SK3,3}$, $F_{FR,SK1,3}$, $F_{FR,SK2,3}$, $F_{FR,SK3,3}$, $F_{S2,SK1,3}$, $F_{S2,SK3,3}$	$F^{waste}_{SR2,3}$, $F_{SR3,SK2,3}$, $F^{waste}_{SR3,3}$, $F_{S1,SK2,3}$, $F^{waste}_{S1,3}$, $F^{waste}_{S2,3}$, $F_{S3,SK2,3}$, $F^{waste}_{S3,3}$
Time interval 4 **(1.5 – 2.0 hr)**	$F_{FR,SK2,4}$	$F^{waste}_{SR2,4}$, $F_{S1,SK2,4}$, $F^{waste}_{S1,4}$, $F^{waste}_{S2,4}$, $F_{S3,SK2,4}$, $F^{waste}_{S3,4}$

Heat cascade

95°C ↓ $H_{0,1} = Q_{H,1}$

$k = 1$

80°C ↓ $H_{1,1} = H_{0,1} + H_{SR1,SK2,1,1} + H^{waste}_{SR1,1,1} + H_{S1,SK2,1,1} + H^{waste}_{S1,1,1} + H^{waste}_{S3,1,1} + H_{S3,SK2,1,1}$

$k = 2$

70°C ↓ $H_{2,1} = H_{1,1} + H_{SR1,SK2,2,1} + H^{waste}_{SR1,2,1} + H_{FR,SK2,2,1} + H_{S1,SK2,2,1} + H^{waste}_{S1,2,1} + H_{S3,SK2,2,1} + H^{waste}_{S3,2,1}$

$k = 3$

25°C ↓ $H_{3,1} = H_{2,1} + H^{waste}_{SR1,3,1} + H_{FR,SK2,3,1} + H^{waste}_{S1,3,1} + H^{waste}_{S2,3,1} + H^{waste}_{S3,3,1} = Q_{C,1}$

Figure 5.15: Heat cascade diagram for time interval 1 in Case Study 2

Heat cascade

105°C $H_{0,2} = Q_{H,2}$

$k = 1$

95°C $H_{1,2} = H_{0,2} + H_{FR,SK1,1,2} + H_{FR,SK3,1,2} + H_{S2,SK1,1,2} + H_{S2,SK3,1,2}$

$k = 2$

80°C $H_{2,2} = H_{1,2} + H_{SR1,SK2,2,2} + H^{waste}_{SR1,2,2} + H_{SR3,SK2,2} + H^{waste}_{SR3,2,2} + H_{FR,SK1,2,2} + H_{FR,SK3,2,2} + H_{S1,SK2,2,2} + H^{waste}_{S1,2,2} + H_{S2,SK1,2,2} + H_{S2,SK3,2,2} + H_{S3,SK2,2,2} + H^{waste}_{S3,2,2}$

$k = 3$

70°C $H_{3,2} = H_{2,2} + H_{SR1,SK2,3,2} + H^{waste}_{SR1,3,2} + H_{SR3,SK2,3,2} + H^{waste}_{SR3,3,2} + H_{FR,SK1,3,2} + H_{FR,SK2,3,2} + H_{FR,SK3,3,2} + H_{S1,SK2,3,2} + H^{waste}_{S1,3,2} + H_{S3,SK2,3,2} + H^{waste}_{S3,3,2}$

$k = 4$

25°C $H_{4,2} = H_{3,2} + H^{waste}_{SR1,4,2} + H^{waste}_{SR3,4,2} + H_{FR,SK1,4,2} + H_{FR,SK2,4,2} + H_{FR,SK3,4,2} + H^{waste}_{S1,4,2} + H^{waste}_{S2,4,2} + H^{waste}_{S3,4,2} = Q_{C,2}$

Figure 5.16: Heat cascade diagram for time interval 2 in Case Study 2

Heat cascade

105°C $H_{0,3} = Q_{H,3}$

$k = 1$

95°C $H_{1,3} = H_{0,3} + H_{SR2,SK1,1,3} + H_{SR2,SK3,1,3} + H_{S2,SK1,1,3} + H_{FR,SK1,1,3} + H_{FR,SK3,1,3} + H_{S2,SK3,1,3}$

$k = 2$

80°C $H_{2,3} = H_{1,3} + H_{SR2,SK1,2,3} + H_{SR2,SK3,2,3} + H_{SR3,SK2,2,3} + H^{waste}_{SR3,2,3} + H_{S2,SK1,2,3} + H_{FR,SK1,2,3} + H^{waste}_{S1,2,3} + H_{S1,SK2,2,3} + H_{FR,SK3,2,3} + H_{S2,SK3,2,3} + H_{S3,SK2,2,3} + H^{waste}_{S3,2,3}$

$k = 3$

70°C $H_{3,3} = H_{2,3} + H_{SR3,SK2,3,3} + H^{waste}_{SR3,3,3} + H_{FR,SK1,3,3} + H_{FR,SK2,3,3} + H_{FR,SK3,3,3} + H_{S1,SK2,3,3} + H^{waste}_{S1,3,3} + H_{S3,SK2,3,3} + H^{waste}_{S3,3,3}$

$k = 4$

25°C $H_{4,3} = H_{3,3} + H^{waste}_{SR2,4,3} + H^{waste}_{SR3,4,3} + H_{FR,SK1,4,3} + H_{FR,SK2,4,3} + H_{FR,SK3,4,3} + H^{waste}_{S1,4,3} + H^{waste}_{S2,4,3} + H^{waste}_{S3,4,3,3} = Q_{C,3}$

Figure 5.17: Heat cascade diagram for time interval 3 in Case Study 2

Heat cascade

95°C $H_{0,4} = Q_{H,4}$

$k = 1$

80°C $H_{1,4} = H_{0,4} + H_{S1,SK2,1,4} + H^{waste}_{S1,1,4} + H_{S3,SK2,1,4} + H^{waste}_{S3,1,4}$

$k = 2$

70°C $H_{2,4} = H_{1,4} + H_{FR,SK2,2,4} + H_{S1,SK2,2,4} + H^{waste}_{S1,2,4} + H_{S3,SK2,2,4} + H^{waste}_{S3,2,4}$

$k = 3$

25°C $H_{3,4} = H_{2,4} + H^{waste}_{SR2,3,4} + H_{FR,SK2,3,4} + H^{waste}_{S1,3,4} + H^{waste}_{S2,3,4} + H^{waste}_{S3,3,4} = Q_{C,4}$

Figure 5.18: Heat cascade diagram for time interval 4 in Case Study 2

Scenario 1

The TOC of fresh resource as well as hot and cold utilities in the property-based HIRCNs without storage system are minimized in Scenario 1. The optimization objective in Equation 5.1 is solved subject to constraints given by Equations 4.3 - 4.18, 4.23 – 4.26, 4.30 – 4.35 and 5.2, yields the minimum TOC of $ 1,714,563/year. The required fresh resource, hot and cold utilities are determined as 444.51 ton/batch, 19,676 kWh/batch and 14,490 kWh/batch respectively. The optimal network configurations of property-based HIRCNs and HEN for Scenario 1 of Case Study 2 are shown in Figures 5.19 and 5.20 respectively. The LINGO code formulation and solutions for Scenario 1 in Case Study 2 are provided in Appendix B.1.1.

Scenario 2

The optimization objective for Scenario 2 is set to minimize the TAC for property-based HIRCNs with storage system. The TAC involves the TOC of fresh resource, hot and cold utilities as well as the annualised capital cost for storage tanks. Hence, Equation 4.36 is solved subject to constraints in Equations 4.3 - 4.18, 4.23 – 4.35 and 4.37 – 4.40 yields the TAC of $ 724,070/year. The requirements of fresh resource, hot utility, cold utility and storage tanks for Scenario 2 are 293.19 ton/batch, 3,421 kWh/batch, 0 kWh/batch 175.56 tons and 67.56 tons respectively. Figures 5.21 and 5.22 show the optimal network configurations of property-based HIRCNs and HEN for Scenario 2 in Case Study 2. The LINGO code formulation and solutions for Scenario 2 of Case Study 2 are illustrated in Appendix B.1.2.

Scenario 3

The optimization objective of Scenario 3 is initially set to minimize the TOC of fresh resource for property-based HIRCNs in order to achieve the first stage optimization. Therefore, Equation 4.40 is solved subject to constraints in Equations 4.3 to 4.18 yields the minimum required fresh resource of 278.22 ton/batch. The obtained fresh resource target is then embedded as an additional constraint (Equation 5.4).

$$F_m = 278.22 \tag{5.4}$$

In second stage optimization, the objective function is set to minimize the TOC of hot and cold utilities. Thus, Equation 4.42 is solved subject to constraints in Equations 4.3 - 4.18, 4.23 - 4.35, 4.37 – 4.40 and 5.4 yields 4,636 kWh/batch of hot utility and 1,391 kWh/batch of cold utility. Besides, 186.67 tons, 72 tons and 19.39 tons of storage tanks are obtained. The optimal network configurations of property-based HIRCNs and HEN for Scenario 3 in Case Study 2 are shown in Figures 5.23 and 5.24. Appendix B.2 shows the LINGO code formulation and solutions for Scenario 3 of Case Study 2.

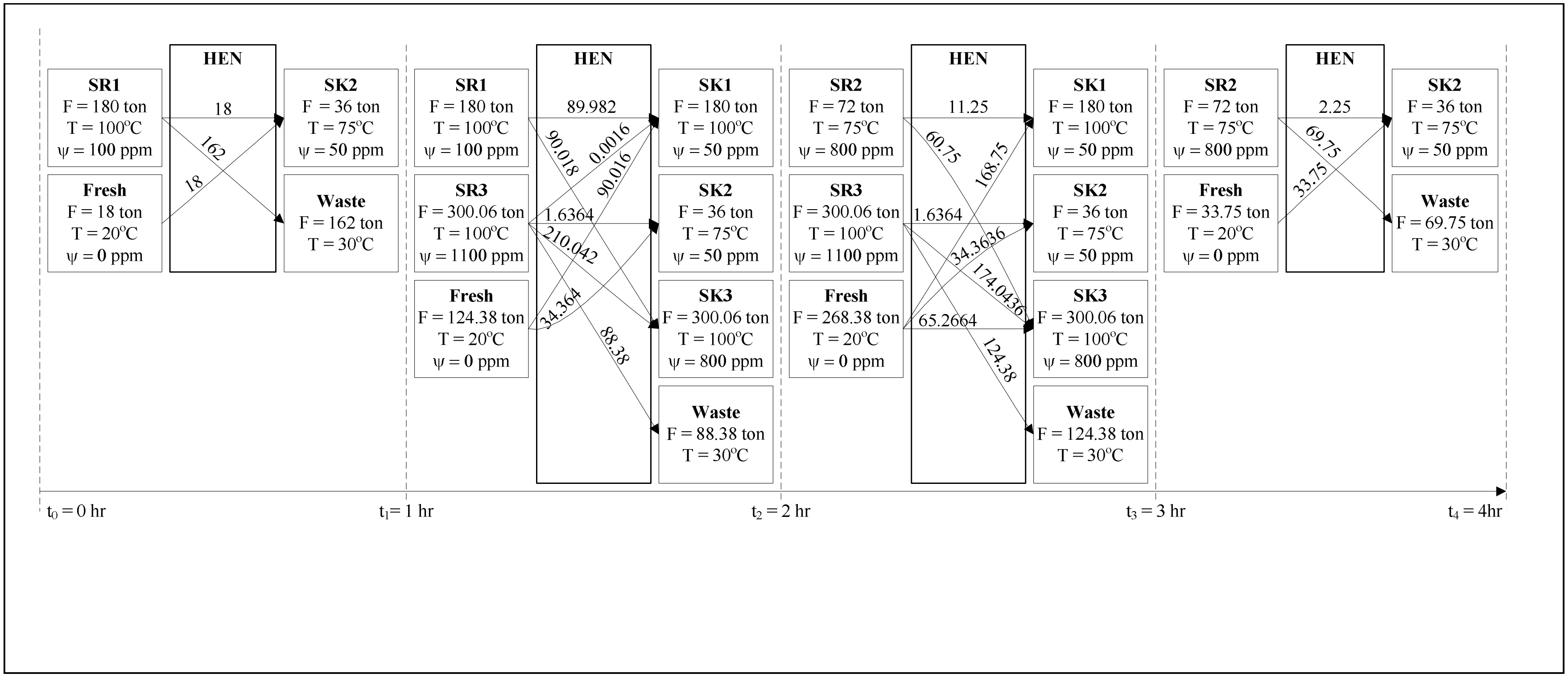

Figure 5.19: Property-based HIRCNs for Scenario 1 of Case Study 2 (All flow terms given in tons)

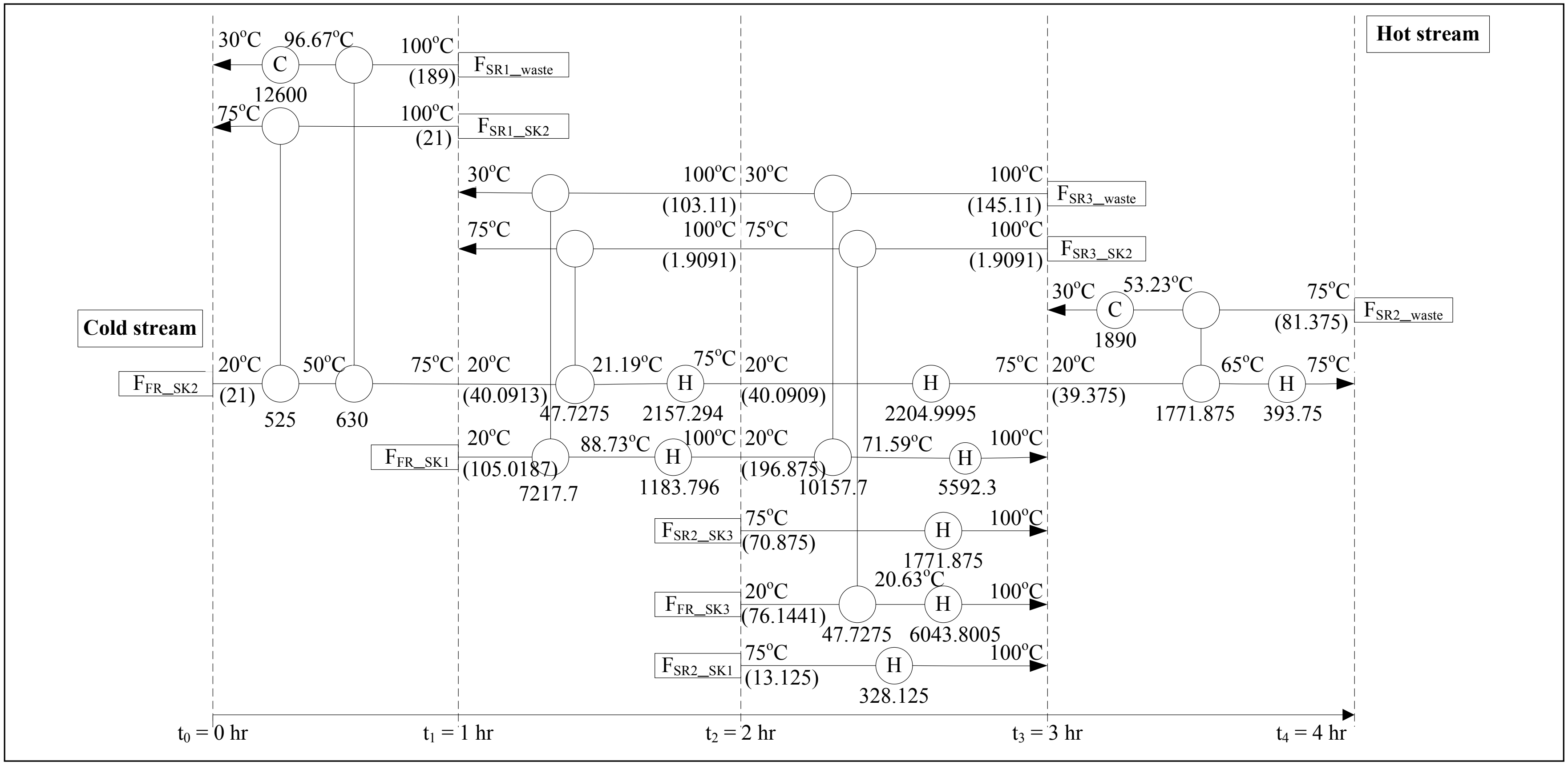

Figure 5.20: HEN for Scenario 1 of Case Study 2 (All heat terms given in kilowatts of heat, kWh; values shown in parentheses represent F*Cp, kWh/°C)

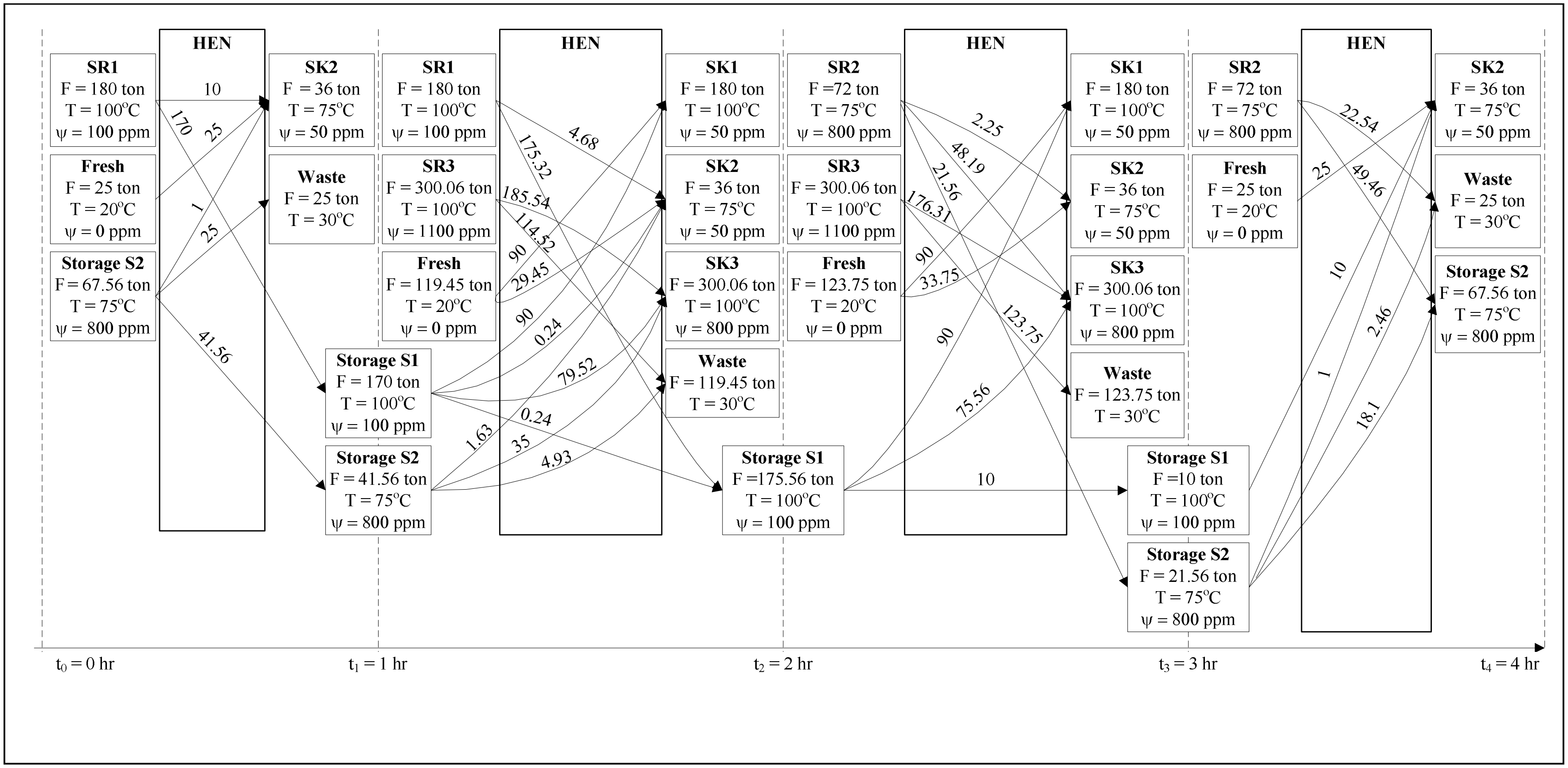

Figure 5.21: Property-based HIRCNs for Scenario 2 of Case Study 2 (All flow terms given in tons)

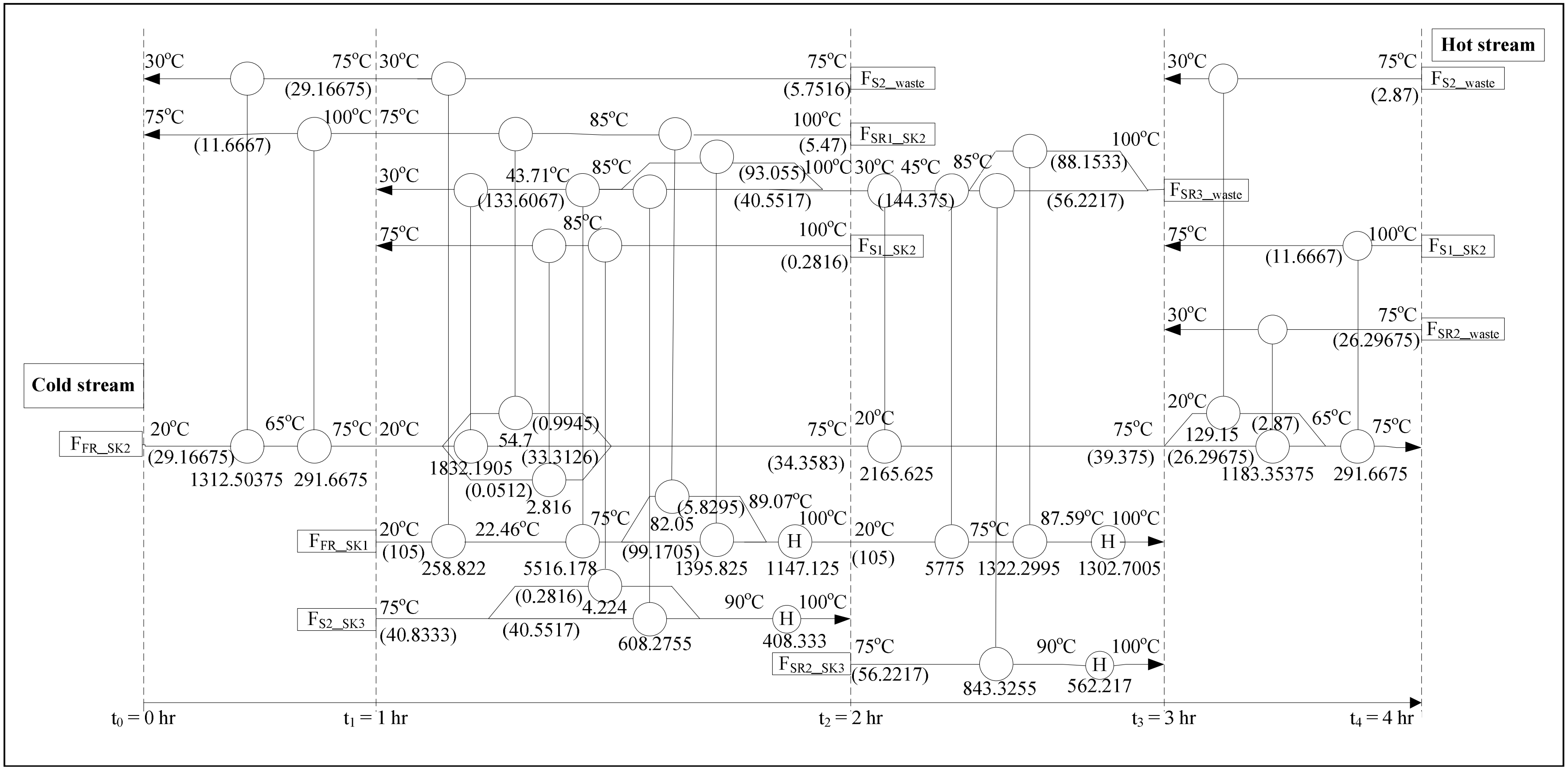

Figure 5.22: HEN for Scenario 2 of Case Study 2 (All heat terms given in kilowatts of heat, kWh; values shown in parentheses represent F*Cp, kWh/ºC)

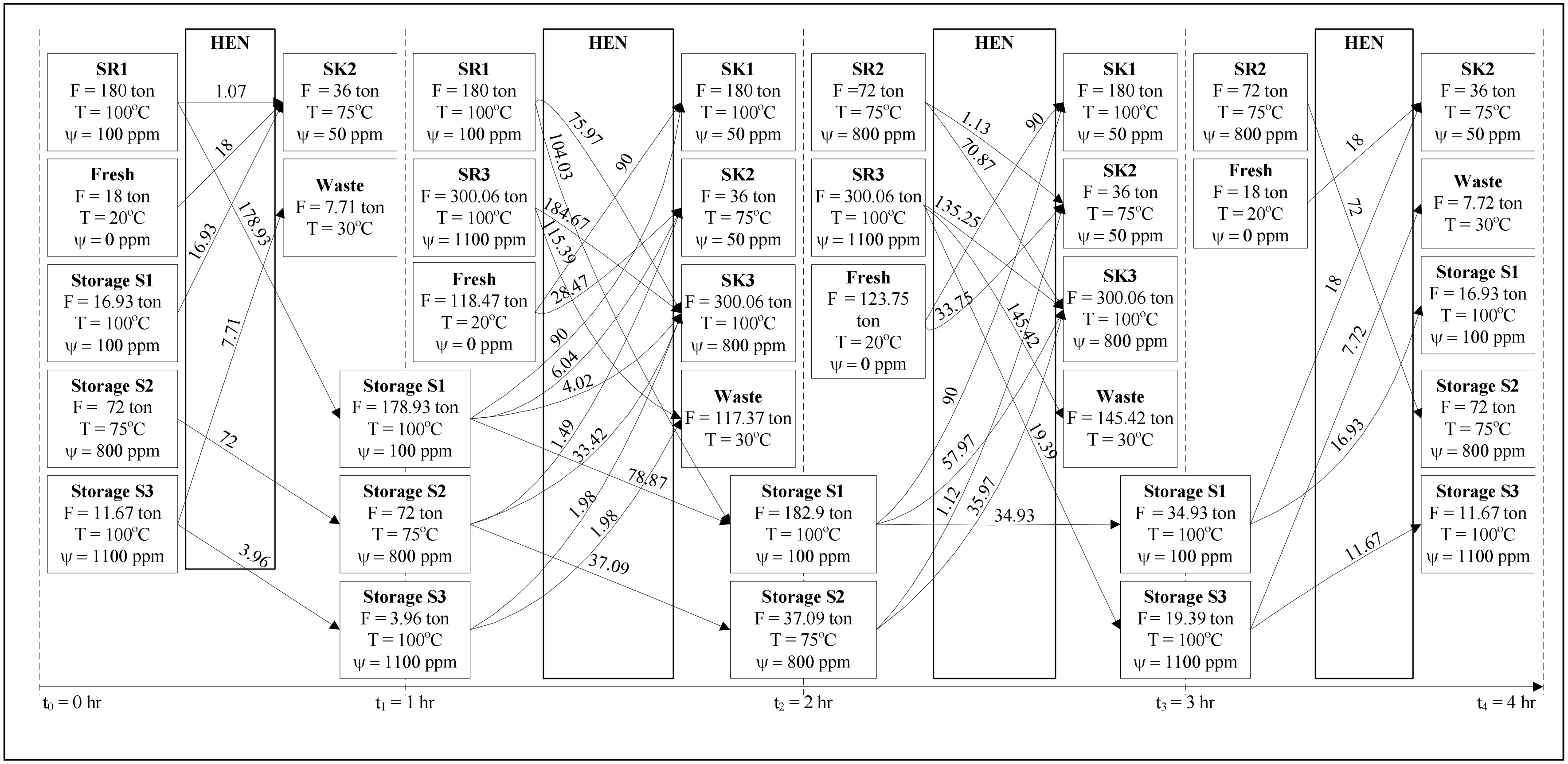

Figure 5.23: Property-based HIRCNs for Scenario 3 of Case Study 2 (All flow terms given in tons)

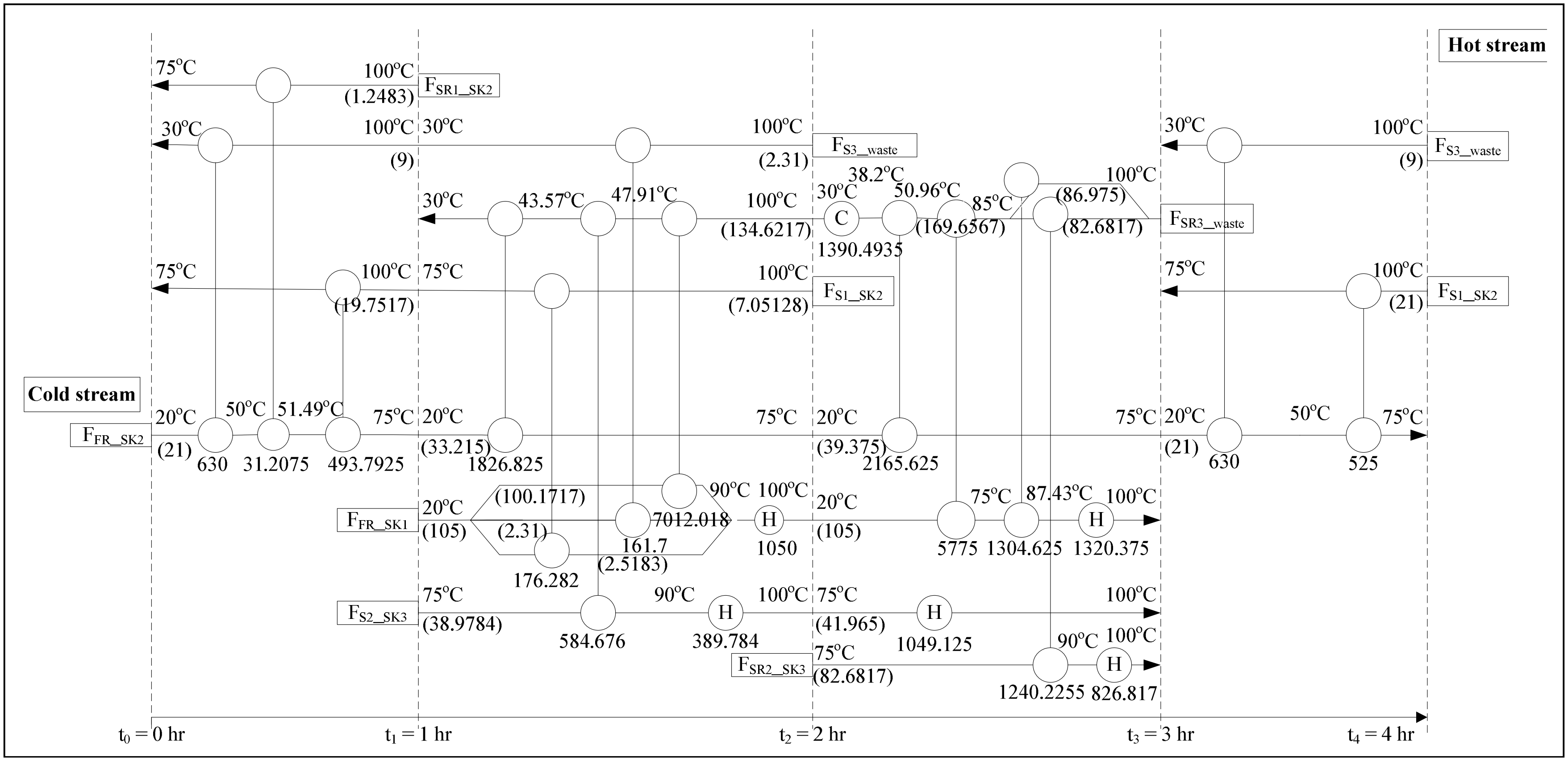

Figure 5.24: HEN for Scenario 3 of Case Study 2 (All heat terms given in kilowatts of heat, kWh; values shown in parentheses represent F*Cp, kWh/°C)

Comparisons between Scenarios 1, 2 and 3 for Case Study 2

Table 5.6 shows results for Scenarios 1, 2 and 3 of Case Study 2. Notice that Scenario 2 obtained the lowest TAC which is $724,070/year. The TAC for Scenarios 1 and 3 is $ 1,714,563/year and $ 758,488/year respectively. TAC for Scenario 2 is around 136.8% and 4.75% lesser than the TAC for Scenarios 1 and 3 respectively. As can be seen, Scenario 2 uses 51.61% lesser fresh resource, 475.15% lesser hot utility and 100% lesser cold utility compared to Scenario 1. It was also found that Scenario 3 requires 59.77% lesser fresh resource, 324.42% lesser hot utility and 941.7% lesser cold utility compared to Scenario 1. This has shown that the consumption of fresh resources, external hot and cold utilities can be reduced with the presence of storage system in property-based HIRCNs. Note that $28,410/year and $35,046/year of the capital costs for storage tanks are added to the TAC of Scenarios 2 and 3 respectively. By contrast, the results for Scenarios 2 and 3 correspond to $ 1,018,903/year and $ 991,121/year in reductions in TAC. On the other hand, Scenario 2 needs 14.97 ton/batch more of external fresh resource than Scenario 3. But the resultant hot and cold utilities for Scenario 2 are 35.52% and 100% lesser than Scenario 3. Moreover, Scenario 2 only requires storage tanks S1 and S2 with the capacities of 175.56 tons and 67.56 tons respectively. However, Scenario 3 needs storage tanks S1, S2 and S3 with the capacities of 186.67 tons, 72 tons and 19.39 tons respectively. This has shown that Scenario 2 requires lesser numbers of storage tanks compared to Scenario 3. Moreover, the sizes of the storage tanks S1 and S2 for Scenario 2 are smaller than the sizes of the storage tanks S1 and S2 for Scenario 3. Thus, the annual capital cost of storage tanks for Scenario 2 is 23.36% lesser than the annual capital cost of storage tanks for Scenario 3. As a conclusion, it is better to simultaneously consider the operating cost of fresh resource, hot and cold utilities as well as the annualized capital cost of storage tanks for the property-based HIRCNs.

Table 5.6: Summary of results for Scenarios 1, 2 and 3 of Case Study 2

Concept	Scenario 1	Scenario 2	Scenario 3
Fresh resource, F_m (ton/batch)	444.51	293.19	278.22
Hot utility, Q_H (kWh/batch)	19,676	3,421	4,636
Cold utility, Q_C (kWh/batch)	14,490	0	1,391
Capacity for storage tank S1, CST_{S1} (tons)	-	175.56	186.67
Capacity for storage tank S2, CST_{S2} (tons)	-	67.56	72
Capacity for storage tank S3, CST_{S3} (tons)	-	0	19.39
Annual capital cost of storage tanks ($/ year)	-	28,410	35,046
Total operating cost, TOC ($/year)	1,714,563	695,660	723,442
Total annualized cost, TAC ($/year)	1,714,563	724,070	758,488

5.4 CASE STUDY 3

Case Study 3 is a case study modified from Case Study in Kheireddine *et al*. (2011) with the limiting data shown in Table 5.7. There are three process sources and three process sinks. The fresh resource 1 (FR1) and 2 (FR2) with the vapour pressures of 3 kpa and 6 kpa are available to fulfil the requirements of sinks. FR1 and FR2 have temperatures of 25^{o}C and 35^{o}C, respectively. In this case study, the unit costs for FR1 and FR2 are $ 0.00132/kg and $ 0.00088/kg. The waste is discharged at 30^{o}C. It is assumed that the cycle time for a batch cycle is 4 hours and the annual operating days is 330 days. Hence, the number of batches per year, N_b is 1,980.

Table 5.7: Limiting data for Case Study 3

Sink	Amount (kg)	Vapour pressure (kpa)	Temperature (°C)	Time (hr)	
				Start	End
SK1	5,436	15 - 35	85	1	3
SK2	3,986	10 - 25	50	0	2
SK3	3,381	13 - 40	65	1	4
Source	**Amount (kg)**	**Vapour pressure (kpa)**	**Temperature (°C)**	**Time (hr)**	
				Start	**End**
SR1	10,983	38	75	0	3
SR2	1,766	25	65	1	2
SR3	5,940	7	40	0	4

First of all, the time intervals for sources and sinks are arranged in ascending order. Based on the data displayed in Table 5.7, the time intervals are t_0 = 0 hr, t_1 = 1 hr, t_2 = 2 hr, t_3 = 3 hr and t_4 = 4 hr. Four time intervals are found for this case study which is time interval 1, 2, 3 and 4. Figure 5.25 describes the time intervals for Case Study 3.

Similar to earlier case studies, stream parameterisation model described in Section 4.4 of Chapter 4 is solved for Case Study 3 first to determine the existence of process sources and sinks in each time interval. Then, the sources and sinks are arranged in the respective time intervals. A revised source-HEN-sink superstructure for each time interval in Case Study 3 as shown in Figure 5.26 is constructed.

Based on Figure 5.26, all the hot and cold streams in each time interval are pre-defined. Table 5.8 shows the pre-defined hot and cold streams in each time interval. By using the data in Figure 5.26 and Table 5.8, the HEN cascade diagram for each time interval is drawn (Figures 5.27 to 5.30). Figures 5.27 to 5.30 are established based on Equations 4.23 – 4.30 and 4.32 – 4.33. In addition, the temperatures that are arranged vertically in Figures 5.27 to 5.30 are the shifted supply and target temperatures for hot and cold streams.

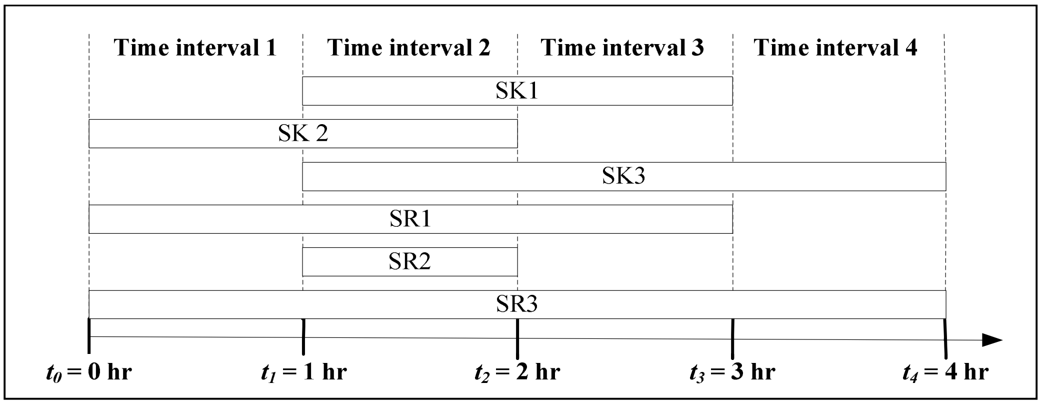

Figure 5.25: A diagram describing time intervals for Case Study 3

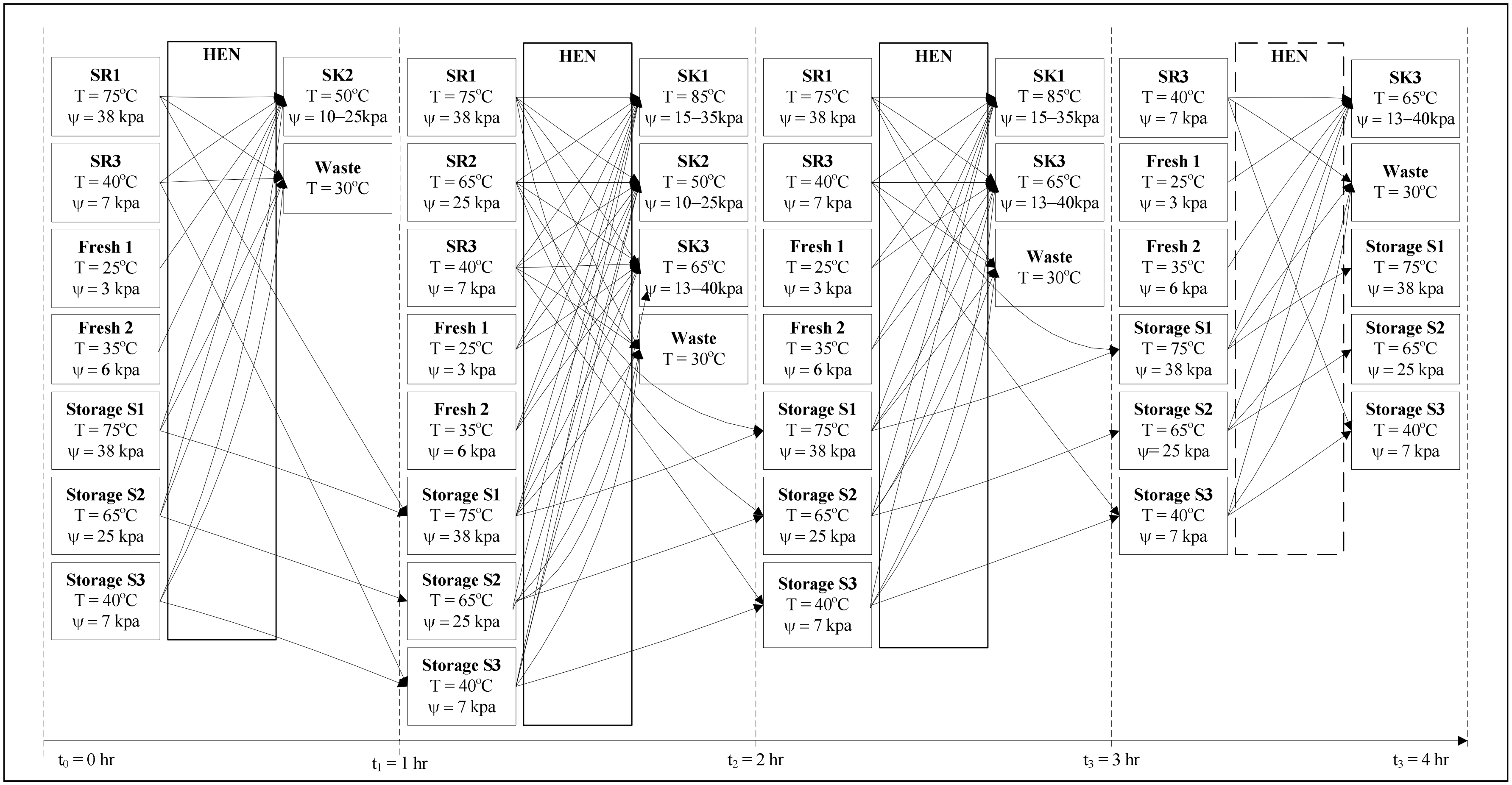

Figure 5.26: Revised source-HEN-sink superstructure for each time intervals in Case Study 3

Table 5.8: Extracted hot and cold streams for Case Study 3

Time interval	Cold streams	Hot streams
Time interval 1 (0 – 1 hr)	$F_{SR3,SK2,1}$, $F_{FR1,SK2,1}$, $F_{FR2,SK2,1}$, $F_{S3,SK2,1}$	$F_{SR1,SK2,1}$, $F^{waste}_{SR1,1}$, $F^{waste}_{SR3,1}$, $F_{S1,SK2,1}$, $F^{waste}_{S1,1}$, $F_{S2,SK2,1}$, $F^{waste}_{S2,1}$, $F^{waste}_{S3,1}$
Time interval 2 (1 – 2 hr)	$F_{SR1,SK1,2}$, $F_{SR2,SK1,2}$, $F_{SR3,SK1,2}$, $F_{SR3,SK2,2}$, $F_{SR3,SK3,2}$, $F_{FR1,SK1,2}$, $F_{FR1,SK2,2}$, $F_{FR1,SK3,2}$, $F_{FR2,SK1,2}$, $F_{FR2,SK2,2}$, $F_{FR2,SK3,2}$, $F_{S1,SK1,2}$, $F_{S2,SK1,2}$, $F_{S3,SK1,2}$, $F_{S3,SK2,2}$, $F_{S3,SK3,2}$	$F_{SR1,SK2,2}$, $F_{SR1,SK3,2}$, $F^{waste}_{SR1,2}$, $F_{SR2,SK2,2}$, $F^{waste}_{SR2,2}$, $F^{waste}_{SR3,2}$, $F_{S1,SK2,2}$, $F_{S1,SK3,2}$, $F^{waste}_{S1,2}$, $F_{S2,SK2,2}$, $F^{waste}_{S2,2}$, $F^{waste}_{S3,2}$
Time interval 3 (2 – 3 hr)	$F_{SR1,SK1,3}$, $F_{SR3,SK1,3}$, $F_{SR3,SK3,3}$, $F_{FR1,SK1,3}$, $F_{FR1,SK3,3}$, $F_{FR2,SK1,3}$, $F_{FR2,SK3,3}$, $F_{S1,SK1,3}$, $F_{S2,SK1,3}$, $F_{S3,SK1,3}$, $F_{S3,SK3,3}$	$F_{SR1,SK3,3}$, $F^{waste}_{SR1,3}$, $F^{waste}_{SR3,3}$, $F_{S1,SK3,3}$, $F^{waste}_{S1,3}$, $F^{waste}_{S2,3}$, $F^{waste}_{S3,3}$
Time interval 4 (3 – 4hr)	$F_{SR3,SK3,4}$, $F_{FR1,SK3,4}$, $F_{FR2,SK3,4}$, $F_{S3,SK3,4}$	$F^{waste}_{SR3,4}$, $F_{S1,SK3,4}$, $F^{waste}_{S1,4}$, $F^{waste}_{S2,4}$, $F^{waste}_{S3,4}$

Heat cascade

70°C $\downarrow$ $H_{0,1} = Q_{H,1}$

$k = 1$

60°C $H_{1,1} = H_{0,1} + H_{\mathrm{SR1,SK2,1,1}} + H^{\mathrm{waste}}_{\mathrm{SR1,1,1}} + H_{\mathrm{S1,SK2,1,1}} + H^{\mathrm{waste}}_{\mathrm{S1,1,1}}$

$k = 2$

55°C $H_{2,1} = H_{1,1} + H_{\mathrm{SR1,SK2,2,1}} + H^{\mathrm{waste}}_{\mathrm{SR1,2,1}} + H_{\mathrm{S1,SK2,2,1}} + H^{\mathrm{waste}}_{\mathrm{S1,2,1}} + H_{\mathrm{S2,SK2,2,1}} + H^{\mathrm{waste}}_{\mathrm{S2,2,1}}$

$k = 3$

45°C $H_{3,1} = H_{2,1} + H_{\mathrm{SR1,SK2,3,1}} + H^{\mathrm{waste}}_{\mathrm{SR1,3,1}} + H_{\mathrm{SR3,SK2,3,1}} + H_{\mathrm{FR1,SK2,3,1}} + H_{\mathrm{FR2,SK2,3,1}} + H_{\mathrm{S1,SK2,3,1}} + H^{\mathrm{waste}}_{\mathrm{S1,3,1}} + H_{\mathrm{S2,SK2,3,1}} + H^{\mathrm{waste}}_{\mathrm{S2,3,1}} + H_{\mathrm{S3,SK2,3,1}}$

$k = 4$

40°C $H_{4,1} = H_{3,1} + H^{\mathrm{waste}}_{\mathrm{SR1,4,1}} + H_{\mathrm{FR1,SK2,4,1}} + H_{\mathrm{FR2,SK2,4,1}} + H^{\mathrm{waste}}_{\mathrm{S1,4,1}} + H^{\mathrm{waste}}_{\mathrm{S2,4,1}}$

$k = 5$

35°C $H_{5,1} = H_{4,1} + H^{\mathrm{waste}}_{\mathrm{SR1,5,1}} + H_{\mathrm{FR1,SK2,5,1}} + H^{\mathrm{waste}}_{\mathrm{S1,5,1}} + H^{\mathrm{waste}}_{\mathrm{S2,5,1}}$

$k = 6$

30°C $H_{6,1} = H_{5,1} + H^{\mathrm{waste}}_{\mathrm{SR1,6,1}} + H^{\mathrm{waste}}_{\mathrm{SR3,6,1}} + H_{\mathrm{FR1,SK2,6,1}} + H^{\mathrm{waste}}_{\mathrm{S1,6,1}} + H^{\mathrm{waste}}_{\mathrm{S2,6,1}} + H^{\mathrm{waste}}_{\mathrm{S3,6,1}}$

$k = 7$

25°C $H_{7,1} = H_{6,1} + H^{\mathrm{waste}}_{\mathrm{SR1,7,1}} + H^{\mathrm{waste}}_{\mathrm{SR3,7,1}} + H^{\mathrm{waste}}_{\mathrm{S1,7,1}} + H^{\mathrm{waste}}_{\mathrm{S2,7,1}} + H^{\mathrm{waste}}_{\mathrm{S3,7,1}} = Q_{C,1}$

Figure 5.27: Heat cascade diagram for time interval 1 in Case Study 3

Heat cascade

90°C: $H_{0,2} = Q_{H,2}$

$k = 1$

80°C: $H_{1,2} = H_{0,2} + H_{SR1,SK1,1,2} + H_{SR2,SK1,1,2} + H_{SR3,SK1,1,2} + H_{FR1,SK1,1,2} + H_{FR2,SK1,1,2} + H_{S3,SK1,1,2} + H_{S3,SK3,1,2} + H_{S1,SK1,1,2} + H_{S2,SK1,1,2}$

$k = 2$

70°C: $H_{2,2} = H_{1,2} + H_{SR2,SK1,2,2} + H_{SR3,SK1,2,2} + H_{FR1,SK1,2,2} + H_{S3,SK1,2,2} + H_{S3,SK3,2,2} + H_{FR2,SK1,2,2} + H_{S2,SK1,2,2}$

$k = 3$

60°C: $H_{3,2} = H_{2,2} + H_{SR1,SK2,3,2} + H_{SR1,SK3,3,2} + H^{waste}_{SR1,3,2} + H_{SR3,SK1,3,2} + H_{SR3,SK3,3,2} + H_{FR1,SK1,3,2} + H_{FR1,SK3,3,2} + H_{FR2,SK1,3,2} + H_{FR2,SK3,3,2} + H_{S3,SK1,3,2} + H_{S3,SK3,3,2} + H_{S1,SK2,3,2} + H_{S1,SK3,3,2} + H^{waste}_{S1,3,2}$

$k = 4$

55°C: $H_{4,2} = H_{3,2} + H_{SR1,SK2,4,2} + H^{waste}_{SR1,4,2} + H_{SR2,SK2,4,2} + H^{waste}_{SR2,4,2} + H_{SR3,SK1,4,2} + H_{SR3,SK3,4,2} + H_{FR1,SK1,4,2} + H_{FR1,SK3,4,2} + H_{FR2,SK1,4,2} + H_{FR2,SK3,4,2} + H_{S1,SK2,4,2} + H^{waste}_{S1,4,2} + H_{S2,SK2,4,2} + H^{waste}_{S2,4,2} + H_{S3,SK1,4,2} + H_{S3,SK3,4,2} +$

$k = 5$

45°C: $H_{5,2} = H_{4,2} + H_{SR1,SK2,5,2} + H^{waste}_{SR1,5,2} + H_{SR2,SK2,5,2} + H^{waste}_{SR2,5,2} + H_{SR3,SK1,5,2} + H_{SR3,SK2,5,2} + H_{SR3,SK3,5,2} + H_{FR1,SK1,5,2} + H_{FR1,SK2,5,2} + H_{FR1,SK3,5,2} + H_{FR2,SK1,5,2} + H_{FR2,SK2,5,2} + H_{FR2,SK3,5,2} + H_{S1,SK2,5,2} + H^{waste}_{S1,5,2} + H_{S2,SK2,5,2} + H^{waste}_{S2,5,2} + H_{S3,SK1,5,2} + H_{S3,SK2,5,2} + H_{S3,SK3,5,2}$

$k = 6$

40°C: $H_{6,2} = H_{5,2} + H^{waste}_{SR1,6,2} + H^{waste}_{SR2,6,2} + H_{FR1,SK1,6,2} + H_{FR1,SK2,6,2} + H_{FR1,SK3,6,2} + H_{FR2,SK1,6,2} + H_{FR2,SK2,6,2} + H_{FR2,SK3,6,2} + H^{waste}_{S1,6,2} + H^{waste}_{S2,6,2}$

$k = 7$

35°C: $H_{7,2} = H_{6,2} + H^{waste}_{SR1,7,2} + H^{waste}_{SR2,7,2} + H_{FR1,SK1,7,2} + H_{FR1,SK2,7,2} + H_{FR1,SK3,7,2} + H^{waste}_{S1,7,2} + H^{waste}_{S2,7,2}$

$k = 8$

30°C: $H_{8,2} = H_{7,2} + H^{waste}_{SR1,8,2} + H^{waste}_{SR2,8,2} + H^{waste}_{SR3,8,2} + H_{FR1,SK1,8,2} + H_{FR1,SK2,8,2} + H_{FR1,SK3,8,2} + H^{waste}_{S1,8,2} + H^{waste}_{S2,8,2} + H^{waste}_{S3,8,2}$

$k = 9$

25°C: $H_{9,2} = H_{8,2} + H^{waste}_{SR1,9,2} + H^{waste}_{SR2,9,2} + H^{waste}_{SR3,9,2} + H^{waste}_{S1,9,2} + H^{waste}_{S2,9,2} + H^{waste}_{S3,9,2} = Q_{C,2}$

Figure 5.28: Heat cascade diagram for time interval 2 in Case Study 3

Heat cascade

90°C: $H_{0,3} = Q_{H,3}$

$k = 1$

80°C: $H_{1,3} = H_{0,3} + H_{SR1,SK1,1,3} + H_{SR3,SK1,1,3} + H_{FR1,SK1,1,3} + H_{FR2,SK1,1,3} + H_{S1,SK1,1,3} + H_{S2,SK1,1,3} + H_{S3,SK1,1,3}$

$k = 2$

70°C: $H_{2,3} = H_{1,3} + H_{SR3,SK1,2,3} + H_{FR1,SK1,2,3} + H_{FR2,SK1,2,3} + H_{S2,SK1,2,3} + H_{S3,SK1,2,3}$

$k = 3$

60°C: $H_{3,3} = H_{2,3} + H_{SR1,SK3,3,3} + H^{waste}_{SR1,3,3} + H_{SR3,SK1,3,3} + H_{SR3,SK3,3,3} + H_{FR1,SK1,3,3} + H_{FR1,SK3,3,3} + H_{FR2,SK1,3,3} + H_{FR2,SK3,3,3} + H_{S1,SK3,3,3} + H^{waste}_{S1,3,3} + H_{S3,SK1,3,3} + H_{S3,SK3,3,3}$

$k = 4$

45°C: $H_{4,3} = H_{3,3} + H^{waste}_{SR1,4,3} + H_{SR3,SK1,4,3} + H_{SR3,SK3,4,3} + H_{FR1,SK1,4,3} + H_{FR1,SK3,4,3} + H_{FR2,SK1,4,3} + H_{FR2,SK3,4,3} + H^{waste}_{S1,4,3} + H^{waste}_{S2,4,3} + H_{S3,SK1,4,3} + H_{S3,SK3,4,3}$

$k = 5$

40°C: $H_{5,3} = H_{4,3} + H^{waste}_{SR1,5,3} + H_{FR1,SK1,5,3} + H_{FR1,SK3,5,3} + H_{FR2,SK1,5,3} + H_{FR2,SK3,5,3} + H^{waste}_{S1,5,3} + H^{waste}_{S2,5,3}$

$k = 6$

35°C: $H_{6,3} = H_{5,3} + H^{waste}_{SR1,6,3} + H_{FR1,SK1,6,3} + H_{FR1,SK3,6,3} + H^{waste}_{S1,6,3} + H^{waste}_{S2,6,3}$

$k = 7$

30°C: $H_{7,3} = H_{6,3} + H^{waste}_{SR1,7,3} + H^{waste}_{SR3,7,3} + H_{FR1,SK1,7,3} + H_{FR1,SK3,7,3} + H^{waste}_{S1,7,3} + H^{waste}_{S2,7,3} + H^{waste}_{S3,7,3}$

$k = 8$

25°C: $H_{8,3} = H_{7,3} + H^{waste}_{SR1,8,3} + H^{waste}_{SR3,8,3} + H^{waste}_{S1,8,3} + H^{waste}_{S2,8,3} + H^{waste}_{S3,4,8,3} = Q_{C,3}$

Figure 5.29: Heat cascade diagram for time interval 3 in Case Study 3

Heat cascade

70°C $H_{0,4} = Q_{H,4}$

$k = 1$

60°C $H_{1,4} = H_{0,4} + H_{SR3,SK3,1,4} + H_{FR1,SK3,1,4} + H_{FR2,SK3,1,4} + H_{S1,SK3,1,4} + H^{waste}{}_{S1,1,4} + H_{S3,SK3,1,4}$

$k = 2$

45°C $H_{2,4} = H_{1,4} + H_{SR3,SK3,2,4} + H_{FR1,SK3,2,4} + H_{FR2,SK3,2,4} + H^{waste}{}_{S1,2,4} + H^{waste}{}_{S2,2,4} + H_{S3,SK3,2,4}$

$k = 3$

40°C $H_{3,4} = H_{2,4} + H_{FR1,SK3,3,4} + H_{FR2,SK3,3,4} + H^{waste}{}_{S1,3,4} + H^{waste}{}_{S2,3,4}$

$k = 4$

35°C $H_{4,4} = H_{3,4} + H_{FR1,SK3,4,4} + H^{waste}{}_{S1,4,4} + H^{waste}{}_{S2,4,4}$

$k = 5$

30°C $H_{5,4} = H_{4,4} + H^{waste}{}_{SR3,5,4} + H_{FR1,SK3,5,4} + H^{waste}{}_{S1,5,4} + H^{waste}{}_{S2,5,4} + H^{waste}{}_{S3,5,4}$

$k = 6$

25°C $H_{6,4} = H_{5,4} + H^{waste}{}_{SR3,6,4} + H^{waste}{}_{S1,6,4} + H^{waste}{}_{S2,6,4} + H^{waste}{}_{S3,6,4} = Q_{C,4}$

Figure 5.30: Heat cascade diagram for time interval 4 in Case Study 3

Scenario 1

In Scenario 1, the TOC of fresh resource as well as hot and cold utilities in the property-based HIRCNs without storage system are minimized. Similar to the previous case studies, the objective function in Equation 5.1 is solved subject to constraints given by Equations 4.3 - 4.18, 4.23 – 4.26, 4.30 – 4.35 and 5.2, yields the minimum TOC of $ 6,305.75/year. In this case study, only hot and cold utilities are required. The hot and cold utilities for this case study are obtained as 102.4283 kWh/batch and 240.5733 kWh/batch respectively. The optimal network configurations of property-based HIRCNs and HEN for Scenario 1 of Case Study 3 are displayed in Figures 5.31 and 5.32 respectively. The LINGO code formulation and solutions for Scenario 1 in Case Study 3 are given in Appendix C.1.1.

Scenario 2

The objective function for Scenario 2 is fixed to minimize the TAC for property-based HIRCNs with storage system. In this scenario, the TOC of fresh resources, hot and cold utilities as well as the annualised capital cost for storage tanks are minimized simultaneously. Thus, Equation 4.36 is solved subject to constraints in Equations 4.3 - 4.18, 4.23 – 4.35 and 4.37 – 4.40 yields the TAC of $ 5,937.92/year. The requirements of fresh resources (FR1 and FR2), hot utility, cold utility and storage tank for Scenario 2 are 0 kg/batch, 0 kg/batch, 69.55742 kWh/batch, 207.7024 kWh/batch and 805 kg respectively. Figures 6.33 and 6.34 show the optimal network configurations of property-based HIRCNs and HEN for Scenario 2 in Case Study 3. The LINGO code formulation and solutions for Scenario 2 of Case Study 3 are illustrated in Appendix C.1.2.

Scenario 3

Firstly, the optimization objective for Scenario 3 is set to minimize the TOC of fresh resources for property-based HIRCNs in order to achieve the first stage optimization. Hence, Equation 4.41 is solved subject to constraints in Equations 4.3 to 4.18 yields 0 kg/batch of FR1 and FR2. The obtained fresh resources of FR1 and FR2 target are then embedded as additional constraints (Equations 5.5 and 5.6).

$$F_{\mathrm{FR1}} = 0 \tag{5.5}$$

$$F_{\mathrm{FR2}} = 0 \tag{5.6}$$

For second stage optimization, the objective function is set to minimize the TOC of hot and cold utilities. Therefore, Equation 4.42 is solved subject to constraints in Equations 4.3 – 4.18, 4.23 – 4.35, 4.37 – 4.40, 5.7 and 5.8 yields 69.55742 kWh/batch of hot utility and 207.7024 kWh/batch of cold utility. Moreover, 3,199.824 kg, 696.4079 kg and 523.4818 kg of storage tanks are needed. The optimal network configurations of property-based HIRCNs and HEN for Scenario 3 in Case Study 3 are shown in Figures 5.35 and 5.36. Appendix C.2 shows the LINGO code formulation and solutions for Scenario 3 of Case Study 3.

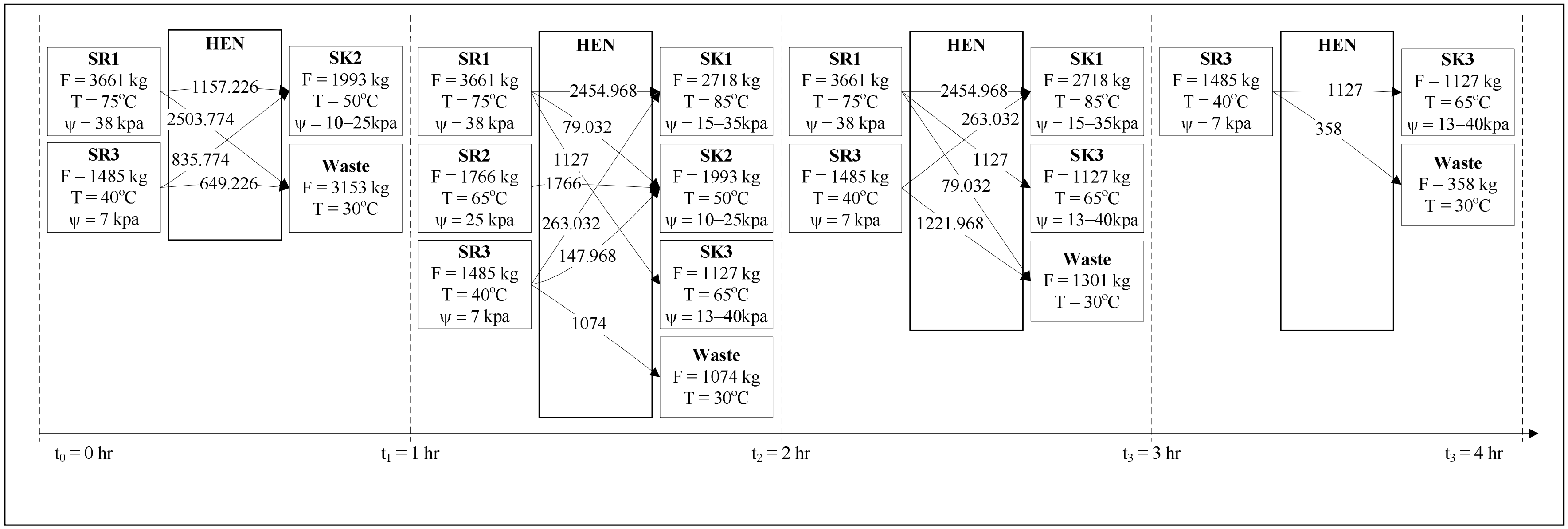

Figure 5.31: Property-based HIRCNs for Scenario 1 of Case Study 3 (All flow terms given in kg)

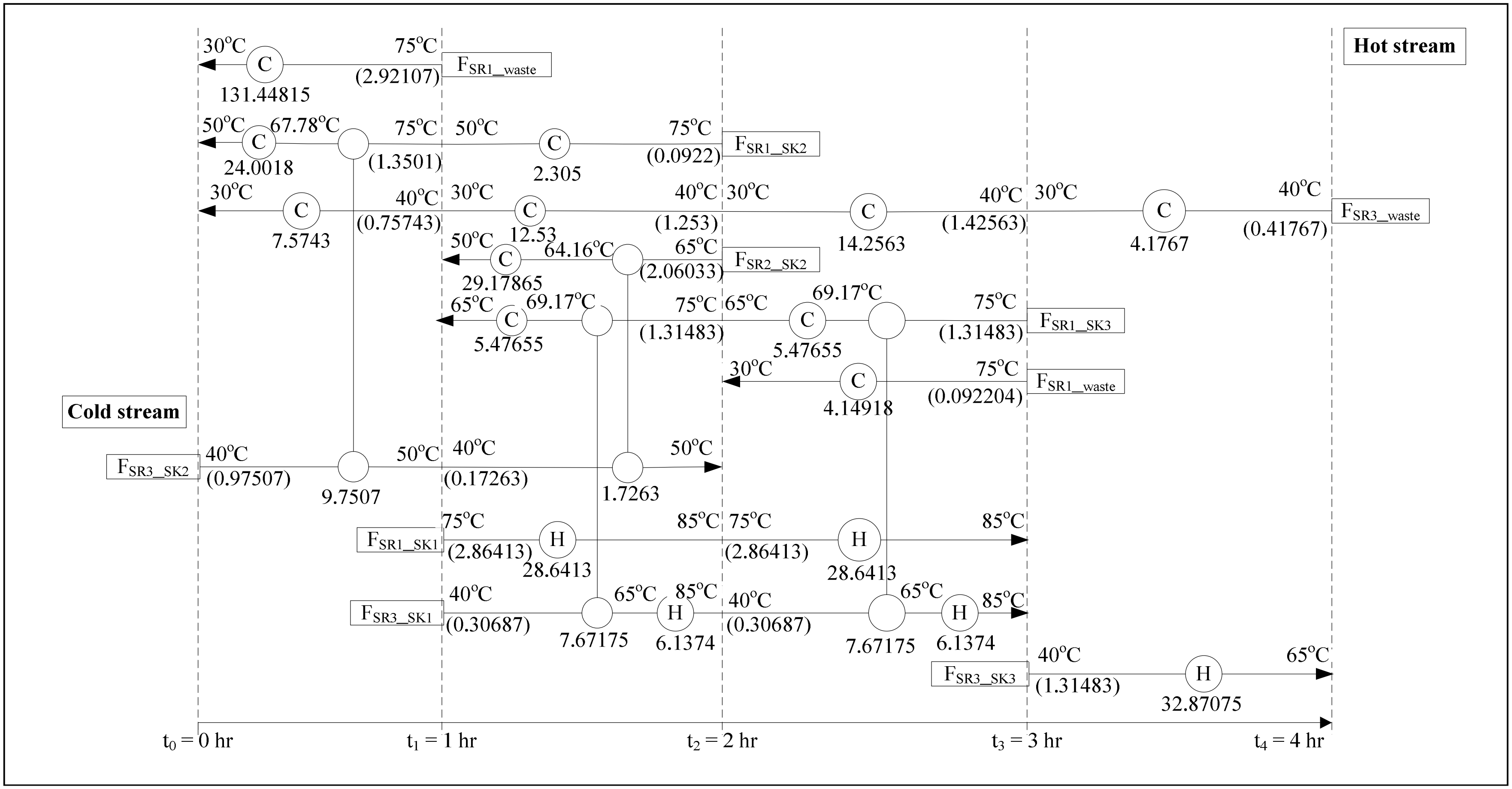

Figure 5.32: HEN for Scenario 1 of Case Study 3 (All heat terms given in kilowatts of heat, kWh; values shown in parentheses represent F*Cp, kWh/°C)

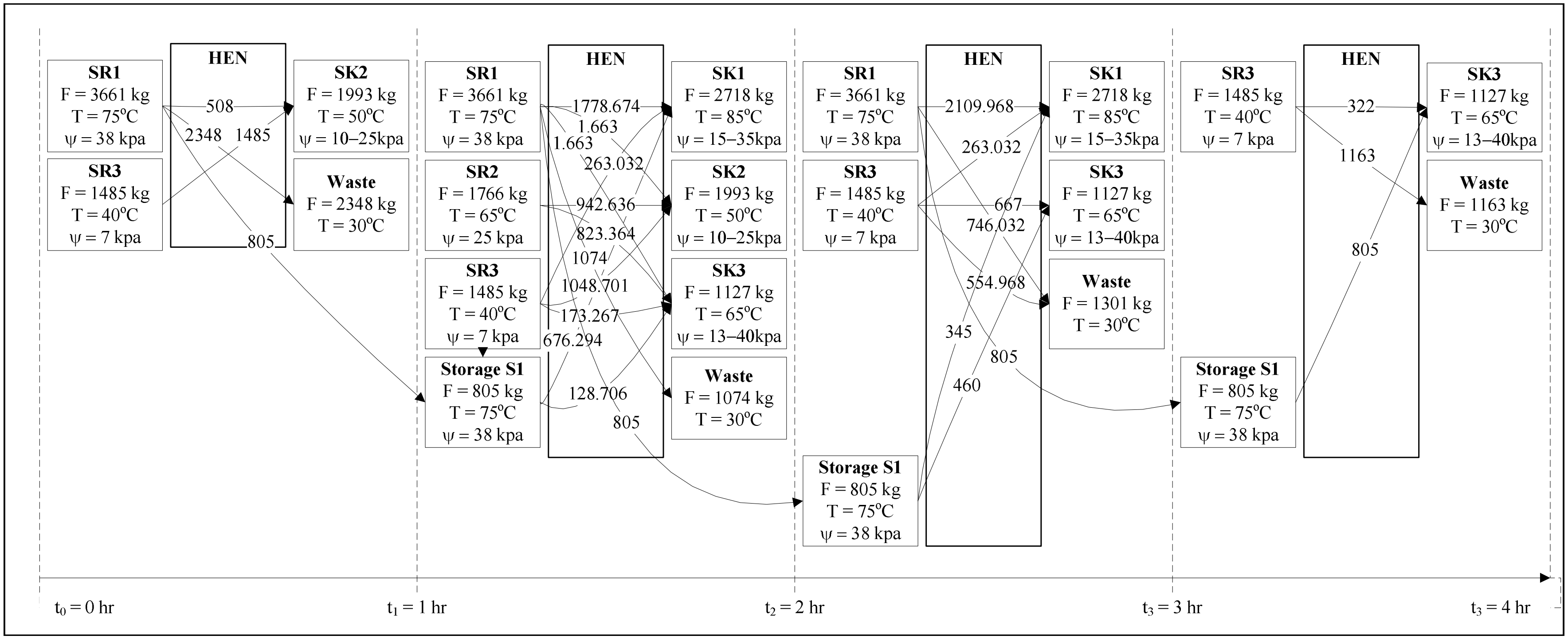

Figure 5.33: Property-based HIRCNs for Scenario 2 of Case Study 3 (All flow terms given in kg)

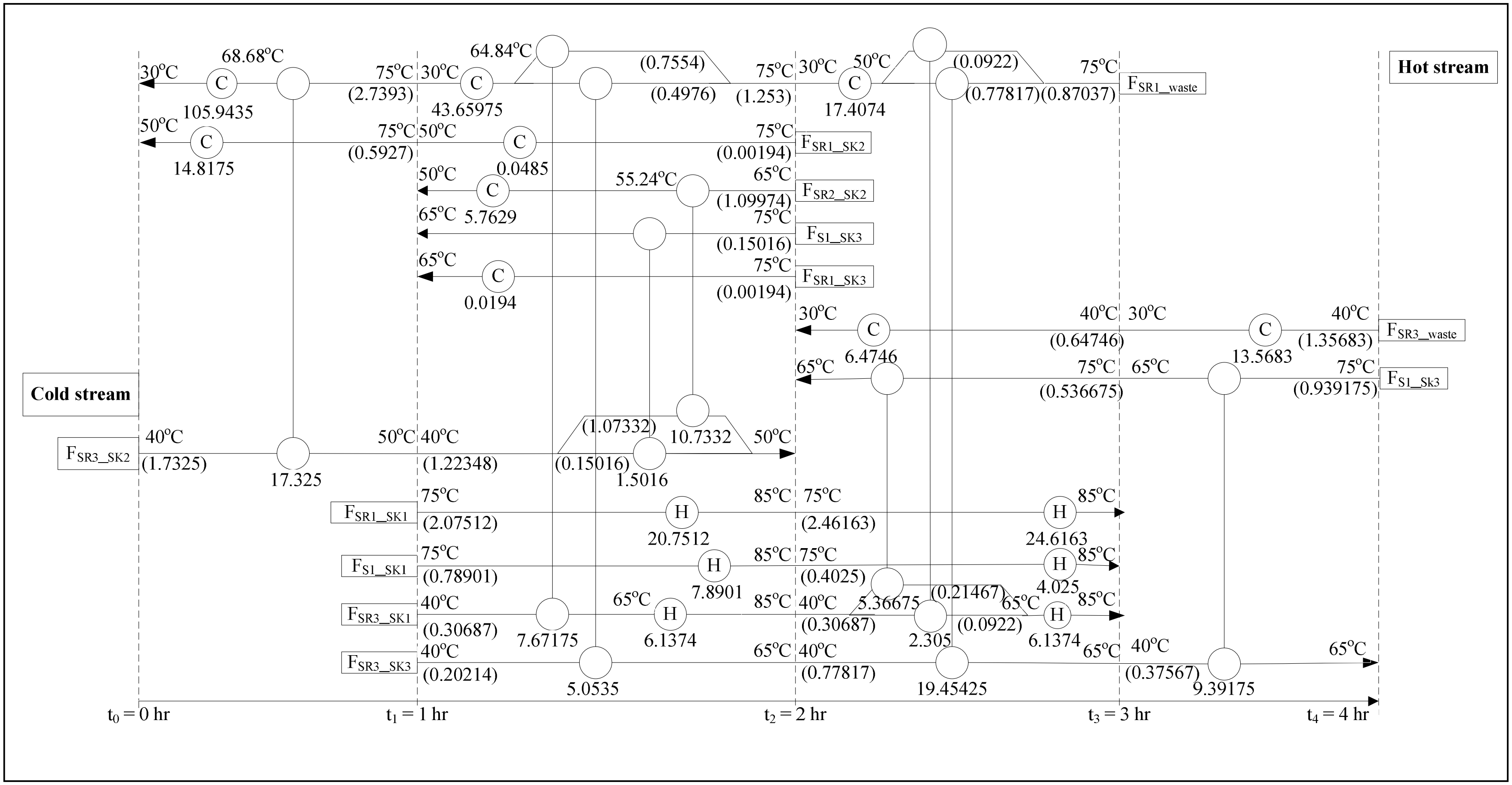

Figure 5.34: HEN for Scenario 2of Case Study 3 (All heat terms given in kilowatts of heat, kWh); values shown in parentheses represent F*Cp, kWh/ºC)

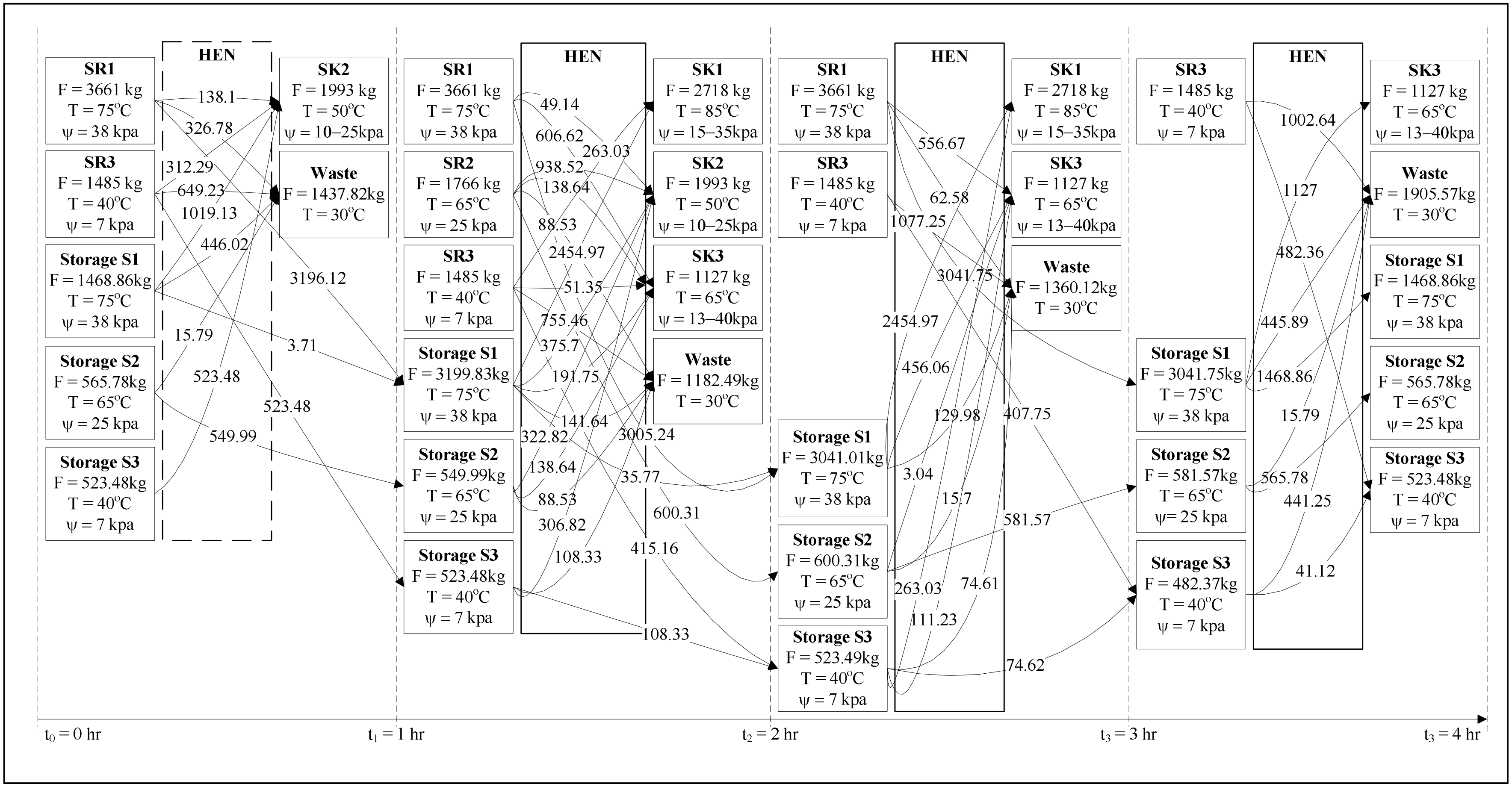

Figure 5.35: Property-based HIRCNs for Scenario 2 of Case Study 3 (All flow terms given in kg)

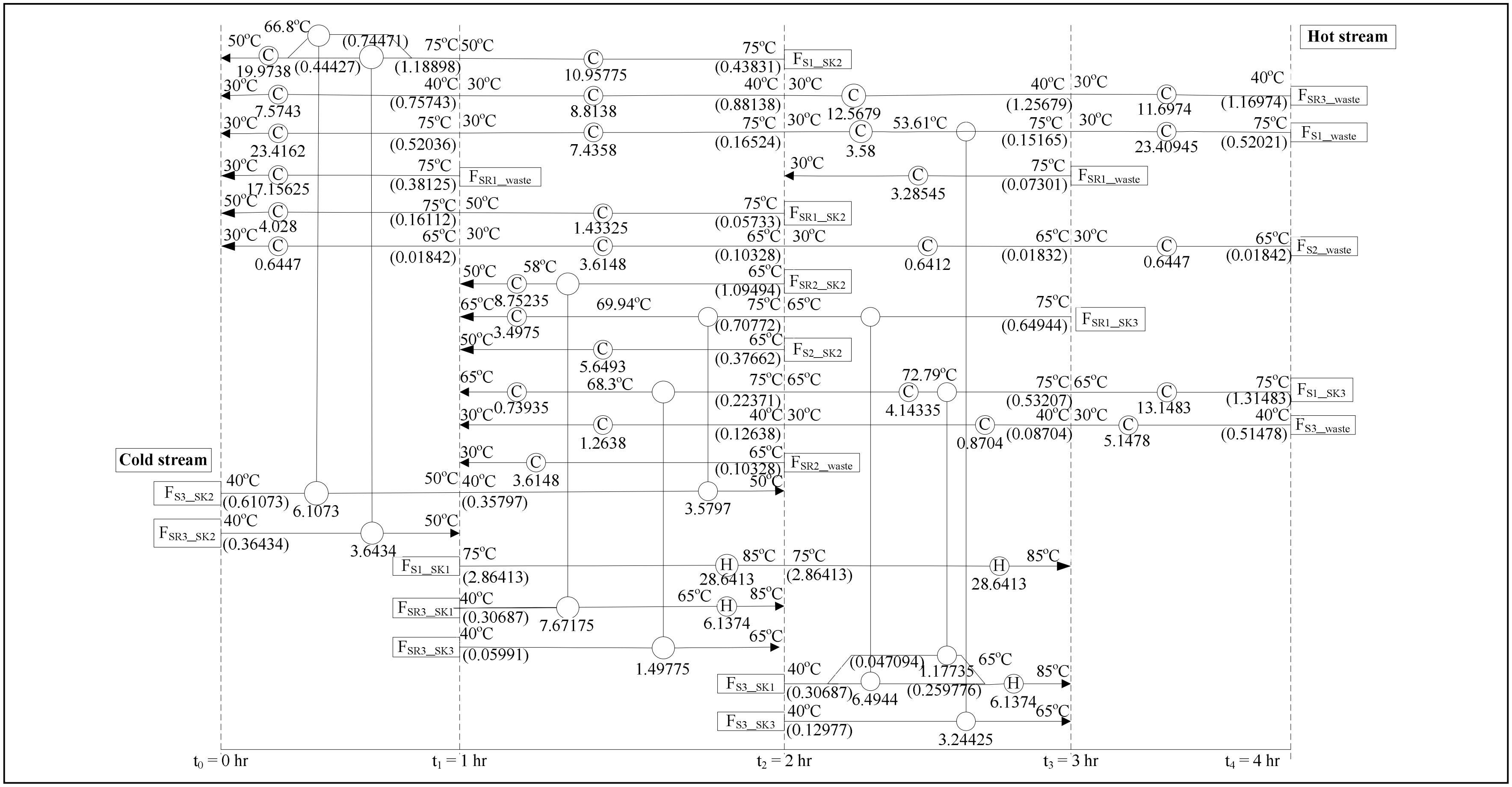

Figure 5.36: HEN for Scenario 3 of Case Study 3 (All heat terms given in kilowatts of heat, kWh; values shown in parentheses represent F*Cp, kWh/°C)

Comparisons between Scenarios 1, 2 and 3 for Case Study 3

Table 5.9 presents the summarized results for Scenarios 1, 2 and 3 of Case Study 3. It is noticed that Scenario 2 acquired the lowest TAC which is $ 5,937.92/year. Scenarios 3 has shown that it is able to obtain TAC of $ 9,046.52/year which is more than the TAC for Scenario 1 ($6,305.75/year). The TAC for Scenario 2 is around 6.19% and 52.35% lower than the TAC for Scenarios 1 and 3 respectively. Note that no fresh resource is required for these three scenarios as process sources have been recycled to process sinks in each time interval. Besides, both Scenarios 2 and 3 require 47.26% lesser hot utility and 15.83% lesser cold utility compared to Scenario 1. These results have proven that storage system has the ability to reduce the consumption of required external energy utility in property-based HIRCNs. Due to the presence of storage system, $ 1,129.11/year and $ 4,237.71/year of the total annualized capital costs for storage tanks are added to the TAC of Scenarios 2 and 3 respectively. However, the TAC of Scenario 2 is still lower than the TAC for Scenario 1. On the other hand, only one storage tank S1 with the capacity of 805 kg is employed for Scenario 2. But Scenario 3 requires storage tanks S1, S2 and S3 with the capacities of 3,199.824 kg, 696.4079 kg and 523.4818 kg respectively. This results show that Scenario 2 requires lesser total numbers and capacities sizes of storage tanks compared to Scenario 3. Consequently, the annual capital cost of storage tanks for Scenario 2 is 275.31% lower than the annual capital cost of storage tanks for Scenario 3. Therefore, it is better to take into account the operating cost of fresh resources, hot and cold utilities as well as the annualized capital cost of storage tanks for the property-based HIRCNs simultaneously.

Table 5.9: Summary of results for Scenarios 1, 2 and 3 of Case Study 3

Concept	Scenario 1	Scenario 2	Scenario 3
Fresh resource, F_{FR1} (kg/batch)	0	0	0
Fresh resource, F_{FR2} (kg/batch)	0	0	0
Hot utility, Q_H (kWh/batch)	102.4283	69.5574	69.5574
Cold utility, Q_C (kWh/batch)	240.5733	207.7024	207.7024
Capacity for storage tank S1, CST_{S1} (kg)	-	805	3,199.824
Capacity for storage tank S2, CST_{S2} (kg)	-	0	696.4079
Capacity for storage tank S3, CST_{S3} (kg)	-	0	523.4818
Annual capital cost of storage tanks ($/year)	-	1,129.11	4,237.71
Total operating cost, TOC ($/year)	6,305.75	4,808.81	4,808.81
Total annualized cost, TAC ($/year)	6,305.75	5,937.92	9,046.52

5.5 CONCLUSIONS

In this chapter, the proposed model for synthesis of property-based HIRCNs is solved using three case studies with different scenarios. The results for all case studies indicate that Scenario 2 which is the simultaneous approach for property-based HIRCN model with storage system yields the best economic saving with respect to the approach without storage system (Scenario 1) and two stages optimization approach (Scenario 3).

Scenario 1 is analyzed to investigate the advantages of having storage system in the HIRCNs. The results showed that storage system does not only reduce the amount of fresh resources needed for the HIRCNs, but also further enhance the energy recovery via the process streams from storage tanks to process sinks and waste. As a result, the requirements of external hot and cold utilities for HIRCNs are also reduced.

On the other hand, Scenario 3 which is a sequential approach is investigated to distinguish the benefits of both simultaneous (Scenarios 1 and 2) and sequential methods. The results showed simultaneous approach is better than sequential method. The contribution of the present work and some recommendations of future works for property-based HIRCNs in batch processes are outlined in the following chapter.

Chapter 6 Conclusions and recommendations for future work

6.1 SUMMARY AND SIGNIFICANCE

The work presented in this thesis offers some major contributions in the area of heat integrated resource conservation networks (HIRCNs) for batch processes. A novel methodology has been developed for the synthesis of HIRCNs for batch processes. The model formulation is a hybrid of automated targeting method for heat exchange network (HEN) model with supertargeting approach for property-based resource conservation network model. It is used to determine the minimum total annualised cost (TAC) of a HIRCN which simultaneously optimise the fresh resources, hot and cold utilities as well as the size of the storage system. Besides, sequential approach for property-based HIRCNs have also been presented due to no previous study can be compared with the newly proposed method. Moreover, comparison of the proposed model with HIRCN without storage system as well as the developed sequential approach has been addressed and applied into two case studies. In all case studies, it has been concluded that the proposed model achieved the lowest TAC. Moreover, the results from one of the case studies showed that this model has the ability to reduce the number of required storage tanks for HIRCNs.

6.2 RECOMMENDATIONS FOR FUTURE WORK

The recommendations for future work are stated below:

a) The methodology developed in this work focused on targeting the operating cost (cost of fresh resources as well as hot and cold utilities) and the capital cost of storage tanks. The major drawback of this method is that the synthesis of HEN design is not solved simultaneously with the HIRCNs. Moreover, the capital costs of heat exchangers and piping system are not considered. This may create a significant impact on the total cost of the HIRCN. Hence, future study can include the synthesis of HEN design along with the capital costs of heat exchangers and piping system to determine an optimal HIRCN.

b) Scheduling of batch processes was not considered in the developed technique for property-based HIRCNs. Scheduling is also an important aspect in batch plants as it is used to minimize the time required to complete processing tasks (the makespan), optimize plant throughput and maximize profit or minimize production cost. Therefore, in future, it is necessary to consider scheduling while synthesize the property-based HIRCNs for batch processes.

c) In this study, environmental constraints are not taken into consideration as part of the model. Due to stricter environmental regulation, it is necessary to incorporate environmental constraints with waste treatment system in the proposed systematic technique. Therefore, the involvement of the waste treatment system or interception devices in the property-based HIRCNs to satisfy both process and environmental constraints can be further investigated.

d) The proposed approach for HIRCNs did not consider the existence of same quality process sources because it is assumed that all the process sources have different qualities. However, this option can reduce the number of required storage tanks. Thus, it can be included in further study of this work.

e) In this study, it is assumed that steady-state operation where no property or contaminant concentration changes in all process sources and storage tanks. However, in batch processes, the property in the process source is not fixed. Therefore, this issue can be considered in future study of this work.

References

Adonyi, Robert, Javier Romero, Luis Puigjaner, and Ferenc Friedler. 2003. "Incorporating Heat Integration in Batch Process Scheduling." *Applied Thermal Engineering* 23 (14): 1743-1762.

Bagajewicz, Miguel, Hernan Rodera, and Mariano Savelski. 2002. "Energy Efficient Water Utilization Systems in Process Plants." *Computers and Chemical Engineering* 26: 59-79.

Bozan, M., Fahir Borak, and Iihan Or. 2001. "A Computerized and Integrated Approach for Heat Exchanger Network Design in Multipurpose Batch Plants." *Chemical Engineering and Processing: Process Intensification* 40 (6): 511-524.

Chen, Cheng-Liang, Chia-Yuan Chang, and Jui-Yuan Lee. "Continuous-time formulation for the synthesis of water-using networks in batch plants." *Industrial& Engineering Chemistry Research* 47, no. 20 (2008): 7818-7832.

Chen, Cheng-Liang, Jui-Yuan Lee, Denny Kok Sum Ng, and Dominic Chwan Yee Foo. 2010. "A Unified Model of Property Integration for Batch and Continuous Processes." *AIChE Journal* 56 (7): 1845-1858. doi: 10.1002/aic.12116.

Chen, Cheng Liang, and Chia Yuan Chang. 2009. "A Resource-Task Network Approach for Optimal Short-Term/ Periodic Scheduling and Heat Integration in Multipurpose Batch Plants." *Applied Thermal Engineering* 29: 1195 - 1208.

Cheng, Kai-Fang, and Chuei-Tin Chang. 2007. "Integrated Water Network Designs for Batch Processes." *Industrial & engineering chemistry research* 46 (4): 1241-1253.

Corominas, J., Antonio Espuña, and Luis Puigjaner. 1994. "Method to Incorporate Energy Integration Considerations in Multiproduct Batch Processes." *Computers and Chemical Engineering* 18 (11): 1043-1055.

Dunn, Russell F., and Mahmoud M. El-Halwagi. 2003. "Process Integration Technology Review: Background and Applications in the Chemical Process Industry." *Journal of Chemical Technology and Biotechnology* 78 (9): 1011-1021.

El-Halwagi, Mahmoud M. 1998. "Pollution Prevention through Process Integration." *Clean Products and Processes* 1 (1): 5-19.

El-Halwagi, Mahmoud M. 2011. *Sustainable Design through Process Integration: Fundamentals and Applications to Industrial Pollution Prevention, Resource Conservation, and Profitability Enhancement*: Butterworth-Heinemann.

El-Halwagi, Mahmoud M., and H. Dennis Spriggs. 1998. "Solve Design Puzzles with Mass Integration." *Chemical Engineering Progress* 94: 25-44.

El-Halwagi, Mahmoud M., Frederico B. Gabriel, and Dustin Harell. 2003. "Rigorous Graphical Targeting for Resource Conversation Via Material Recycle/ Reuse Networks." *Industrial and Engineering Chemistry Research* 42 (19): 4319-4328. doi: 10.1021/ie030318a.

El-Halwagi, Mahmoud M., Ian M. Glasgow, and Xiaoyun Qin. 2004. "Property Integration: Componentless Design Techniques and Visualization Tools." *AIChE Journal* 50 (8): 1854-1869. doi: 10.1002/aic.10305.

Fernández, Inmaculada, Carlos Javier Renedo, Severiano Francisca Pérez, Alfredo Ortiz, and Mario Mañana. 2012. "A Review: Energy Recovery in Batch Processes." *Renewable and Sustainable Energy Reviews* 16 (4): 2260-2277.

Foo, Dominic Chwan Yee. 2010. "Automated Targeting Technique for Batch Process Integration." *Industrial and Engineering Chemistry Research* 49 (20): 9899-9916.

Foo, Dominic Chwan Yee. 2012. *Process Integration for Resource Conservation*. Boca Raton, Florida, US.: CRC Press.

Foo, Dominic Chwan Yee, Vasiliki Kazantzi, Mahmoud M El-Halwagi, and Zainuddin Abdul Manan. 2006. "Surplus Diagram and Cascade Analysis Technique for Targeting Property-Based Material Reuse Network." *Chemical Engineering Science* 61 (8): 2626-2642.

Furman, Kevin C., and Nikolaos Vasili Sahinidis. 2002. "A Critical Review and Annotated Bibliography for Heat Exchanger Network Synthesis in the 20th Century." *Industrial and Engineering Chemistry Research* 41 (10): 2335-2370.

Gundersen, Truls, and L. Naess. 1988. "The Synthesis of Cost Optimal Heat Exchanger Networks - an Industrial Review of the State of the Art." *Computers & Chemical Engineering* 12 (6): 503-530. doi: 10.1016/0098-1354(88)87002-9.

Grooms, Daniel, Vasiliki Kazantzi, and Mahmoud M. El-Halwagi. 2005. "Optimal Synthesis and Scheduling of Hybrid Dynamic/ Steady-State Property Integration Networks." *Computers and Chemical Engineering* 29 (11): 2318-2325.

Halim, Iskandar, and Rajagopalan Srinivasan. 2009. "Sequential Methodology for Scheduling of Heat-Integrated Batch Plants." *Industrial and Engineering Chemistry Research* 48 (18): 8551-8565.

Holczinger, Tibor, Máté Hegyháti, and Ferenc Friedler. 2012. "Simultaneous Heat Integration and Batch Process Scheduling." *Chemical Engineering Transactions* 29: 337-342. doi: 10.3303/CET1229057.

IEO, International Energy Outlook 2010. 2010. International Energy Outlook 2010. Accessed June 25, 2011, http://www.eia.gov/oiaf/ieo/pdf/0484(2010).pdf.

Isafiade, Adeniyi J., and Duncan M. Fraser. 2010. "Interval Based Minlp Superstructure Synthesis of Heat Exchanger Networks for Multi-Period Operations." *Chemical Engineering Research and Design* 88 (10): 1329-1341.

Jung, SooHee, In Beum Lee, Dae Ryook Yang, and Kun Soo Chang. 1994. "Synthesis of Maximum Energy Recovery Networks in Batch Processes." *Korean Journal of Chemical Engineering* 11 (3): 162-171.

Kazantzi, Vasiliki, and Mahmoud M. El-Halwagi. 2005. "Targeting Material Reuse via Property Integration." *Chemical Engineering Progress* 101 (8): 28-38.

Kemp, Ian C., and A. W. Deakin. 1989a. "The Cascade Analysis for Energy and Process Integration of Batch Processes. I: Calculation of Energy Targets." *Chemical Engineering Research and Design* 67 (5): 495-509.

Kemp, Ian C., and A. W. Deakin. 1989b. "The Cascade Analysis for Energy and Process Integration of Batch Processes. Ii: Network Design and Process Scheduling." *Chemical Engineering Research and Design* 67 (5): 510-516.

Kemp, Ian C., and A. W. Deakin. 1989c. "The Cascade Analysis for Energy and Process Integration of Batch Processes. Iii: A Case Study." *Chemical Engineering Research and Design* 67 (5): 517-525.

Kemp, Ian C.. 2007. *Pinch Analysis and Process Integration: A User Guide on Process Integration for the Efficient Use of Energy*. Amsterdam: Elsevier, Butterworth-Heinemann.

Kheireddine, Houssein, Younas Dadmohammadi, Chun Deng, Xiao Feng, and Mahmoud El-Halwagi. 2011. "Optimization of Direct Recycle Networks with the Simultaneous Consideration of Property, Mass, and Thermal Effects." *Industrial and Engineering Chemistry Research* 50 (7): 3754-3762.

Klemeš, Jiri, Ferenc Friedler, Igor Bulatov, and Petar Varbanov. 2010. *Sustainability in the Process Industry: Integration and Optimization–Integration and Optimization*: McGraw-Hill Professional.

Lee, B., and Gintaras V. Reklaitis. 1995a. "Optimal Scheduling of Cyclic Batch Processes for Heat Integration—I. Basic Formulation." *Computers and Chemical Engineering* 19 (8): 883-905.

Lee, B., and Gintaras V. Reklaitis. 1995b. "Optimal Scheduling of Cyclic Batch Processes for Heat Integration—Ii. Extended Problems." *Computers and Chemical Engineering* 19 (8): 907-931.

Linnhoff, Bodo. 1979. "Thermodynamic analysis in the design of process networks." PhD diss., University of Leeds.

Linnhoff, Bodo. 1993. "Pinch Analysis: A State-of-the-Art Overview: Techno-Economic Analysis." *Chemical Engineering Research and Design* 71 (5): 503-522.

Linnhoff, Bodo, David W. Townsend, D. Boland, Geoffrey Frederick Hewitt, B. E. A. Thomas, A. R. Guy, and R. H. Marshall. 1982. *A User Guide on Process Integration for the Efficient Use of Energy*. Rugby: IChemE.

Linnhoff, Bodo, and John R. Flower. 1978. "Synthesis of Heat Exchanger Networks: I. Systematic Generation of Energy Optimal Networks." *AIChE Journal* 24 (4): 633-642.

Liu, Lin-lin, Jian Du, Feng Xiao, Li Chen, and Ping-jing Yao. 2011. "Direct Heat Exchanger Network Synthesis for Batch Process with Cost Targets." *Applied Thermal Engineering* 31 (14): 2665-2675.

Majozi, Thokozani. 2006. "Heat Integration of Multipurpose Batch Plants Using a Continuous-Time Framework." *Applied Thermal Engineering* 26 (13): 1369 - 1377.

Morar, Mihaela, and Paul Serban Agachi. 2010. "Review: Important Contributions in Development and Improvement of the Heat Integration Techniques." *Computers and Chemical Engineering* 34 (8): 1171-1179.

Nápoles-Rivera, Fabricio, José María Ponce-Ortega, Mahmoud M El-Halwagi, and Arturo Jiménez-Gutiérrez. 2010. "Global Optimization of Mass and Property Integration Networks with in-Plant Property Interceptors." *Chemical Engineering Science* 65 (15): 4363-4377.

Ng, Denny Kok Sum, Dominic Chwan Yee Foo, Arwa Rabie, and Mahmoud M. El-Halwagi. 2008. "Simultaneous Synthesis of Property-Based Water Reuse/Recycle and Interception Networks for Batch Processes." *AIChE Journal* 54 (10): 2624-2632.

Ng, Denny Kok Sum, Dominic Chwan Yee Foo, Raymond R. Tan, Choon Hock Pau, and Yin Ling Tan. 2009a. "Automated Targeting for Conventional and Bilateral Property-Based Resource Conservation Network." *Chemical Engineering Journal* 149 (1): 87-101.

Ng, Denny Kok Sum, Dominic Chwan Yee Foo, and Raymond R. Tan. 2009b. "Automated Targeting Technique for Single-Impurity Resource Conservation Networks. Part 1: Direct Reuse/ Recycle." *Industrial & Engineering Chemistry Research* 48: 7637-7646.

Ng, Denny Kok Sum, Dominic Chwan Yee Foo, and Raymond R. Tan. 2009c. "Automated Targeting Technique for Single-Impurity Resource Conservation

Networks. Part 2: Single-Pass and Partitioning Waste-Interception Systems." *Industrial and Engineering Chemistry Research* 48 (16): 7647-7661.

Ng, Denny Kok Sum, Dominic Chwan Yee Foo, Raymond R. Tan, and Mahmoud El-Halwagi. 2010. "Automated Targeting Technique for Concentration-and Property-Based Total Resource Conservation Network." *Computers and Chemical Engineering* 34 (5): 825-845.

Nun, Sophanna, Dominic Chwan Yee Foo, Raymond R. Tan, Luis F. Razon, and Ryuichi Egashira. 2011. "Fuzzy Automated Targeting for Trade-Off Analysis in Batch Water Networks." *Asia-Pasific Journal of Chemical Engineering* 6 (3): 537 - 551.

Ponce-Ortega, Jose Maria, Ana Carolina Hortua, Mahmoud El-Halwagi, and Arturo Jimenez-Gutierrez. 2009. "A Property-Based Optimization of Direct Recycle Networks and Wastewater Treatment Processes." *AIChE Journal* 55 (9): 2329-2344.

Ponce-Ortega, José María, Mahmoud M El-Halwagi, and Arturo Jiménez-Gutiérrez. 2010. "Global Optimization for the Synthesis of Property-Based Recycle and Reuse Networks Including Environmental Constraints." *Computers and Chemical Engineering* 34 (3): 318-330.

Qin, Xiaoyun, Frederico B. Gabriel, Dustin Harell, and Mahmoud M. El-Halwagi. 2004. "Algebraic Techniques for Property Integration Via Componentless Design." *Industrial and Engineering Chemistry Research* 43 (14): 3792-3798.

Rabie, Arwa H., and Mahmoud M. El-Halwagi. 2008. "Synthesis and scheduling of optimal batch water-recycle networks." *Chinese Journal of Chemical Engineering* no. 16 (3):474-479.

Rojas-Torres, Ma Guadalupe, José María Ponce-Ortega, Medardo Serna-González, Fabricio Nápoles-Rivera, and Mahmoud M El-Halwagi. 2012. "Synthesis of Water Networks Involving Temperature-Based Property Operators and Thermal Effects." *Industrial and Engineering Chemistry Research* 52 (1): 442-461.

Seider, Warren D, J. D. Seader, and Daniel R. Lewin. 1999. *Process Design Synthesis, Analysis, and Evaluation*: Wiley New York.

Seider, Warren D., J. D. Seader, and Daniel R. Lewin. 2004. *Product and Process Design Principles*. Second ed. New Jersey, US: John Wiley and Sons, Inc.

SERI, GLOBAL 2000, and Friends of the Earth Europe. 2009. Overconsumption? Our Use of the World's Natural Resources. Accessed June 25, 2011, http://www.foeeurope.org/publications/2009/Overconsumption_Sep09.pdf.

Shelley, Mark D., and Mahmoud M El-Halwagi. 2000. "Component-Less Design of Recovery and Allocation Systems: A Functionality-Based Clustering Approach." *Computers and Chemical Engineering* 24 (9): 2081-2091.

Shoaib, Abeer Mohamed, Said M. Aly, Moustafa E. Awad, Dominic Chwan Yee Foo, and Mahmoud M. El-Halwagi. 2008. "A hierarchical approach for the synthesis of batch water network." *Computers & Chemical Engineering* no. 32 (3):530-539.

Smith, Robin. 2005. *Chemical Process: Design and Integration*. West Sussex, England: John Wiley & Sons Ltd..

Tan, Yin Ling, Denny Kok Sum Ng, Mahmoud M. El-Halwagi, Dominic Chwan Yee Foo, and Yudi Samyudia. 2013. "Synthesis of Heat Integrated Resource Conservation Networks with Varying Operating Parameters." *Industrial and Engineering Chemistry Research*.

UNEP, United National Environment Programme. 2011. Decoupling Rate of Resource Consumption from Economic Growth Rate - Europe Presents a Mixed Picture. Accessed March 28, 2013,http://www.unep.org/resourcepanel/metals_recycling/files/pdf/decoupling_press_release_for_Europe_green_week_May_2011.pdf.

Vaklieva-Bancheva, Natasha, Boyan Bonev Ivanov, N. Shah, and Costas C. Pantelides. 1996. "Heat Exchanger Network Design for Multipurpose Batch Plants." *Computers and Chemical Engineering* 20 (8): 989-1001.

Wang, Y. P., and Robin Smith. 1995. "Time Pinch Analysis." *Chemical engineering research & design* 73 (8): 905-914.

Zhao, X. G., Brian K. O'neill, J. R. Roach, and Richard M. Wood. 1998a. "Heat Integration for Batch Processes: Part 1: Process Scheduling Based on Cascade Analysis." *Chemical Engineering Research and Design* 76 (6): 685-699.

Zhao, X. G., Brian K. O'neill, J. R. Roach, and Richard M. Wood. 1998. "Heat Integration for Batch Processes: Part 2: Heat Exchanger Network Design." *Chemical Engineering Research and Design* 76 (6): 700-710.

Appendix A-Matching formulation code in LINGO and matching formulation solution from LINGO for Case Study 1

A.1 Simultaneous approach

A.1.1 without storage system (Scenario 1)

```
!Total_Cost_O = TOC;
min = Total_Cost_O;

!Equation 4.3;
FSR1A  =  FSR1*tA/(tSR1ET-tSR1ST);  FSR2A  =  FSR2*tA/(tSR2ET-tSR2ST);
FSR3A = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B  =  FSR1*tB/(tSR1ET-tSR1ST);  FSR2B  =  FSR2*tB/(tSR2ET-tSR2ST);
FSR3B = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C  =  FSR1*tC/(tSR1ET-tSR1ST);  FSR2C  =  FSR2*tC/(tSR2ET-tSR2ST);
FSR3C = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D  =  FSR1*tD/(tSR1ET-tSR1ST);  FSR2D  =  FSR2*tD/(tSR2ET-tSR2ST);
FSR3D = FSR3*tD/(tSR3ET-tSR3ST);

!Equation 4.4;
FSK1A  =  FSK1*tA/(tSK1ET-tSK1ST);  FSK2A  =  FSK2*tA/(tSK2ET-tSK2ST);
FSK3A = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B  =  FSK1*tB/(tSK1ET-tSK1ST);  FSK2B  =  FSK2*tB/(tSK2ET-tSK2ST);
FSK3B = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C  =  FSK1*tC/(tSK1ET-tSK1ST);  FSK2C  =  FSK2*tC/(tSK2ET-tSK2ST);
FSK3C = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D  =  FSK1*tD/(tSK1ET-tSK1ST);  FSK2D  =  FSK2*tD/(tSK2ET-tSK2ST);
FSK3D = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time  interval  1(0-0.5hr),  B=Time  interval  2(0.5-1.0hr),  C=Time
interval 3(1.0-1.5hr)& D=Time interval 4(1.5-2.0hr);
tA = t1 - t0; tB = t2 - t1; tC = t3 - t2; tD = t4 - t3;

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <= 1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <= 1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <= 1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <= 1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <= 1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <= 1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <= 1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <= 1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <= 1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <= 1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <= 1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <= 1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <= 1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <= 1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <= 1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <= 1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <= 1000*XSR2A;
```

```
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <= 1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <= 1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <= 1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <= 1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <= 1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <= 1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <= 1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <= 1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <= 1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <= 1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <= 1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <= 1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <= 1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <= 1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <= 1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <= 1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <= 1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <= 1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <= 1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <= 1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <= 1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <= 1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <= 1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <= 1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <= 1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <= 1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <= 1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <= 1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <= 1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <= 1000*XSK3C;
```

```
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <= 1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR_SK1A + FS1_SK1A + FS2_SK1A + FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR_SK2A + FS1_SK2A + FS2_SK2A + FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR_SK3A + FS1_SK3A + FS2_SK3A + FS3_SK3A;
```

```
FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR_SK1B + FS1_SK1B + FS2_SK1B + FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR_SK2B + FS1_SK2B + FS2_SK2B + FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR_SK3B + FS1_SK3B + FS2_SK3B + FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR_SK1C + FS1_SK1C + FS2_SK1C + FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR_SK2C + FS1_SK2C + FS2_SK2C + FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR_SK3C + FS1_SK3C + FS2_SK3C + FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR_SK1D + FS1_SK1D + FS2_SK1D + FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR_SK2D + FS1_SK2D + FS2_SK2D + FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR_SK3D + FS1_SK3D + FS2_SK3D + FS3_SK3D;

!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR_SK1A*OFFR +
                    FS1_SK1A*OSR1 + FS2_SK1A*OSR2 + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR_SK2A*OFFR +
                    FS1_SK2A*OSR1 + FS2_SK2A*OSR2 + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR_SK3A*OFFR +
                    FS1_SK3A*OSR1 + FS2_SK3A*OSR2 + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR_SK1B*OFFR +
                    FS1_SK1B*OSR1 + FS2_SK1B*OSR2 + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR_SK2B*OFFR +
                    FS1_SK2B*OSR1 + FS2_SK2B*OSR2 + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR_SK3B*OFFR +
                    FS1_SK3B*OSR1 + FS2_SK3B*OSR2 + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR_SK1C*OFFR +
                    FS1_SK1C*OSR1 + FS2_SK1C*OSR2 + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR_SK2C*OFFR +
                    FS1_SK2C*OSR1 + FS2_SK2C*OSR2 + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR_SK3C*OFFR +
                    FS1_SK3C*OSR1 + FS2_SK3C*OSR2 + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR_SK1D*OFFR +
                    FS1_SK1D*OSR1 + FS2_SK1D*OSR2 + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR_SK2D*OFFR +
                    FS1_SK2D*OSR1 + FS2_SK2D*OSR2 + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
```

```
                    FSR3_SK3D*YSR3D*OSR3 + FFR_SK3D*OFFR +
                    FS1_SK3D*OSR1 + FS2_SK3D*OSR2 + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR = FFR_SK1A + FFR_SK2A + FFR_SK3A + FFR_SK1B + FFR_SK2B +
            FFR_SK3B + FFR_SK1C + FFR_SK2C + FFR_SK3C + FFR_SK1D +
            FFR_SK2D + FFR_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FSR1_WasteB
              + FSR2_WasteB + FSR3_WasteB + FSR1_WasteC + FSR2_WasteC
              + FSR3_WasteC + FSR1_WasteD + FSR2_WasteD + FSR3_WasteD
              + FS1_WasteA + FS2_WasteA + FS3_WasteA + FS1_WasteB +
              FS2_WasteB + FS3_WasteB + FS1_WasteC + FS2_WasteC +
              FS3_WasteC + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.26, 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + CP*(85-65)*(- FSR2_SK1A - FFR_SK1A)*(1000/3600);
H2A = H1A + CP*(65-55)*(- FFR_SK1A)*(1000/3600);
H3A = H2A + CP*(55-25)*(FSR2_WasteA - FFR_SK1A)*(1000/3600);
H3A = QCA;

H0B = QHB;
H1B = H0B + CP*(85-65)*(- FSR1_SK1B - FSR2_SK1B -
      FFR_SK1B)*(1000/3600);
H2B = H1B + CP*(65-55)*(- FSR1_SK1B - FFR_SK1B)*(1000/3600);
H3B = H2B + CP*(55-45)*(- FSR1_SK1B - FSR1_SK2B + FSR2_SK2B +
      FSR2_WasteB - FFR_SK1B - FFR_SK2B)*(1000/3600);
H4B = H3B + CP*(45-35)*(FSR2_WasteB - FFR_SK1B -
```

```
      FFR_SK2B)*(1000/3600);
H5B = H4B + CP*(35-25)*(FSR1_WasteB + FSR2_WasteB - FFR_SK1B -
      FFR_SK2B)*(1000/3600);
H5B = QCB;

H0C = QHC;
H1C = H0C + CP*(85-75)*(-FSR1_SK1C - FSR2_SK1C -
      FFR_SK1C)*(1000/3600);
H2C = H1C + CP*(75-65)*(-FSR1_SK1C - FSR2_SK1C + FSR3_SK2C +
      FSR3_WasteC - FFR_SK1C)*(1000/3600);
H3C = H2C + CP*(65-55)*(-FSR1_SK1C + FSR3_SK2C + FSR3_WasteC -
      FFR_SK1C)*(1000/3600);
H4C = H3C + CP*(55-45)*(-FSR1_SK1C - FSR1_SK2C + FSR2_SK2C +
      FSR2_WasteC + FSR3_SK2C + FSR3_WasteC - FFR_SK1C -
      FFR_SK2C)*(1000/3600);
H5C = H4C + CP*(45-35)*(FSR2_WasteC + FSR3_WasteC - FFR_SK1C -
      FFR_SK2C)*(1000/3600);
H6C = H5C + CP*(35-25)*(FSR1_WasteC + FSR2_WasteC + FSR3_WasteC -
      FFR_SK1C - FFR_SK2C)*(1000/3600);
H6C = QCC;

H0D = QHD;
H1D = H0D + CP*(75-55)*(FSR3_SK2D + FSR3_SK3D +
      FSR3_WasteD)*(1000/3600);
H2D = H1D + CP*(55-45)*(-FSR1_SK2D + FSR3_SK2D + FSR3_SK3D +
      FSR3_WasteD - FFR_SK2D)*(1000/3600);
H3D = H2D + CP*(45-35)*(FSR3_SK3D + FSR3_WasteD - FFR_SK2D -
      FFR_SK3D)*(1000/3600);
H4D = H3D + CP*(35-25)*(FSR1_WasteD + FSR3_WasteD - FFR_SK2D -
      FFR_SK3D)*(1000/3600);
H4D = QCD;

!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0; H5B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0; H5C>=0; H6C>=0;
H0D>=0; H1D>=0; H2D>=0; H3D>=0; H4D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 5.1;
Total_Cost_O = Cm*Total_FFR*Nb*1000 + Total_QH*CHU*Nb +
               Total_QC*CCU*Nb;

!Equation 5.2;
FSR1_S1A =0; FSR1_S1B=0; FSR1_S1C=0; FSR1_S1D=0;
FSR2_S2A =0; FSR2_S2B=0; FSR2_S2C=0; FSR2_S2D=0;
FSR3_S3A =0; FSR3_S3B=0; FSR3_S3C=0; FSR3_S3D=0;

!Data;
FSR1 = 100;  FSR2 = 40; FSR3 = 25;
FSK1 = 100;  FSK2 = 40; FSK3 = 25;
OSR1 = 400;  OSR2 = 200; OSR3 = 200; OFFR = 0;
OSK1 = 100;   OSK2 = 0;  OSK3 = 100;
tSR1ST = 0.5; tSR1ET = 2.0;
```

```
tSR2ST = 0; tSR2ET = 1.5;
tSR3ST = 1.0; tSR3ET = 2.0;
tSK1ST = 0; tSK1ET = 1.5;
tSK2ST = 0.5; tSK2ET = 2.0;
tSK3ST = 1.5; tSK3ET = 2.0;
Nb = 3960;
Cm = 0.001;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
t0 = 0; t1 = 0.5; t2 = 1; t3 = 1.5; t4 = 2.0;
End
```

```
  Global optimal solution found.
  Objective value:                              747098.0
  Objective bound:                              747098.0
  Infeasibilities:                              0.000000
  Extended solver steps:                               1
  Total solver iterations:                           138
```

Variable	Value	Reduced Cost
TOTAL_COST_O	747098.0	0.000000
FSR2A	13.33333	0.000000
FSR1B	33.33333	0.000000
FSR2B	13.33333	0.000000
FSR1C	33.33333	0.000000
FSR2C	13.33333	0.000000
FSR3C	12.50000	0.000000
FSR1D	33.33333	0.000000
FSR3D	12.50000	0.000000
FSK1A	33.33333	0.000000
FSK1B	33.33333	0.000000
FSK2B	13.33333	0.000000
FSK1C	33.33333	0.000000
FSK2C	13.33333	0.000000
FSK2D	13.33333	0.000000
FSK3D	25.00000	0.000000
ZSR1A	0.000000	0.000000
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	1.000000	0.000000
ZSR2B	1.000000	-118712.2
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	0.000000
ZSR3A	0.000000	0.000000
ZSR3B	0.000000	0.000000
ZSR3C	1.000000	-71610.04
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	1.000000	0.000000
XSR1D	1.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	-118712.2
XSR2C	1.000000	0.000000
XSR2D	0.000000	0.000000
XSR3A	1.000000	0.000000

XSR3B	1.000000	0.000000
XSR3C	1.000000	-71610.04
XSR3D	1.000000	0.000000
YSR1A	0.000000	0.000000
YSR1B	1.000000	4466.033
YSR1C	1.000000	9240.000
YSR1D	1.000000	9240.000
YSR2A	1.000000	-210320.2
YSR2B	1.000000	0.000000
YSR2C	1.000000	-29182.91
YSR2D	0.000000	0.000000
YSR3A	0.000000	0.000000
YSR3B	0.000000	0.000000
YSR3C	1.000000	0.000000
YSR3D	1.000000	20789.99
ZSK1A	1.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	0.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	0.000000	0.000000
ZSK3C	0.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	1.000000	0.000000
XSK2D	1.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	1.000000	0.000000
YSK1A	1.000000	383768.2
YSK1B	1.000000	262537.1
YSK1C	1.000000	215765.0
YSK1D	0.000000	0.000000
YSK2A	0.000000	0.000000
YSK2B	1.000000	70048.00
YSK2C	1.000000	70048.00
YSK2D	1.000000	41712.00
YSK3A	0.000000	0.000000
YSK3B	0.000000	0.000000
YSK3C	0.000000	0.000000
YSK3D	1.000000	21780.01
FSR2_SK1A	13.33333	0.000000
FSR1_SK1B	1.666667	0.000000
FSR1_WASTEB	31.66667	0.000000
FSR2_SK1B	13.33333	0.000000
FSR1_WASTEC	33.33333	0.000000
FSR2_SK1C	13.33333	0.000000
FSR3_SK1C	3.333333	0.000000
FSR3_WASTEC	9.166667	0.1570802E-02

FSR1_WASTED	33.33333	0.000000
FSR3_SK3D	12.50000	0.000000
FFR_SK1A	20.00000	0.000000
FFR_SK1B	18.33333	0.000000
FFR_SK2B	13.33333	0.000000
FFR_SK1C	16.66667	0.000000
FFR_SK2C	13.33333	0.000000
FFR_SK2D	13.33333	0.000000
FFR_SK3D	12.50000	0.000000
TOTAL_FFR	107.5000	0.000000
TOTAL_WASTE	107.5000	0.000000
H0A	1711.111	0.000000
QHA	1711.111	0.000000
H1A	933.3333	0.000000
H2A	700.0000	0.000000
QCA	0.000000	0.000000
H0B	1769.444	0.000000
QHB	1769.444	0.000000
H1B	991.6667	0.000000
H2B	758.3333	0.000000
H3B	369.4444	0.000000
QCB	0.000000	0.000000
H0C	1166.667	0.000000
QHC	1166.667	0.000000
H1C	816.6667	0.000000
H2C	573.6111	0.000000
H3C	486.1111	0.000000
H4C	243.0556	0.000000
H6C	145.8333	0.000000
QCC	145.8333	0.000000
QHD	0.000000	0.000000
H1D	291.6667	0.000000
H2D	281.9444	0.000000
H3D	126.3889	0.000000
H4D	213.8889	0.000000
QCD	213.8889	0.000000
TOTAL_QH	4647.222	0.000000
TOTAL_QC	359.7222	0.000000

Note: The flow and heat terms which are not shown are equal to zero.

A.1.2 with storage system (Scenario 2)

```
!Total_Cost = TAC;
min = Total_Cost;

!Equation 4.3;
FSR1A  =  FSR1*tA/(tSR1ET-tSR1ST);  FSR2A  =  FSR2*tA/(tSR2ET-tSR2ST);
FSR3A = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B  =  FSR1*tB/(tSR1ET-tSR1ST);  FSR2B  =  FSR2*tB/(tSR2ET-tSR2ST);
FSR3B = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C  =  FSR1*tC/(tSR1ET-tSR1ST);  FSR2C  =  FSR2*tC/(tSR2ET-tSR2ST);
FSR3C = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D  =  FSR1*tD/(tSR1ET-tSR1ST);  FSR2D  =  FSR2*tD/(tSR2ET-tSR2ST);
FSR3D = FSR3*tD/(tSR3ET-tSR3ST);
```

```
!Equation 4.4;
FSK1A  =  FSK1*tA/(tSK1ET-tSK1ST);  FSK2A  =  FSK2*tA/(tSK2ET-tSK2ST);
FSK3A = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B  =  FSK1*tB/(tSK1ET-tSK1ST);  FSK2B  =  FSK2*tB/(tSK2ET-tSK2ST);
FSK3B = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C  =  FSK1*tC/(tSK1ET-tSK1ST);  FSK2C  =  FSK2*tC/(tSK2ET-tSK2ST);
FSK3C = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D  =  FSK1*tD/(tSK1ET-tSK1ST);  FSK2D  =  FSK2*tD/(tSK2ET-tSK2ST);
FSK3D = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time  interval  1(0-0.5hr),  B=Time  interval  2(0.5-1.0hr),  C=Time
interval 3(1.0-1.5hr)& D=Time interval 4(1.5-2.0hr);
tA = t1 - t0; tB = t2 - t1; tC = t3 - t2; tD = t4 - t3;

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <= 1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <= 1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <= 1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <= 1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <= 1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <= 1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <= 1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <= 1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <= 1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <= 1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <= 1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <= 1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <= 1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <= 1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <= 1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <= 1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <= 1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <= 1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <= 1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <= 1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <= 1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <= 1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <= 1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <= 1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);
```

```
@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <= 1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <= 1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <= 1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <= 1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <= 1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <= 1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <= 1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <= 1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <= 1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <= 1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <= 1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <= 1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <= 1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <= 1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <= 1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <= 1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <= 1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <= 1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <= 1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <= 1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <= 1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <= 1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <= 1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <= 1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);
```

```
!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR_SK1A + FS1_SK1A + FS2_SK1A + FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR_SK2A + FS1_SK2A + FS2_SK2A + FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR_SK3A + FS1_SK3A + FS2_SK3A + FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR_SK1B + FS1_SK1B + FS2_SK1B + FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR_SK2B + FS1_SK2B + FS2_SK2B + FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR_SK3B + FS1_SK3B + FS2_SK3B + FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR_SK1C + FS1_SK1C + FS2_SK1C + FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR_SK2C + FS1_SK2C + FS2_SK2C + FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR_SK3C + FS1_SK3C + FS2_SK3C + FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR_SK1D + FS1_SK1D + FS2_SK1D + FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR_SK2D + FS1_SK2D + FS2_SK2D + FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR_SK3D + FS1_SK3D + FS2_SK3D + FS3_SK3D;
```

```
!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR_SK1A*OFFR +
                    FS1_SK1A*OSR1 + FS2_SK1A*OSR2 + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR_SK2A*OFFR +
                    FS1_SK2A*OSR1 + FS2_SK2A*OSR2 + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR_SK3A*OFFR +
                    FS1_SK3A*OSR1 + FS2_SK3A*OSR2 + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR_SK1B*OFFR +
                    FS1_SK1B*OSR1 + FS2_SK1B*OSR2 + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR_SK2B*OFFR +
                    FS1_SK2B*OSR1 + FS2_SK2B*OSR2 + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR_SK3B*OFFR +
                    FS1_SK3B*OSR1 + FS2_SK3B*OSR2 + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR_SK1C*OFFR +
                    FS1_SK1C*OSR1 + FS2_SK1C*OSR2 + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR_SK2C*OFFR +
                    FS1_SK2C*OSR1 + FS2_SK2C*OSR2 + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR_SK3C*OFFR +
                    FS1_SK3C*OSR1 + FS2_SK3C*OSR2 + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR_SK1D*OFFR +
                    FS1_SK1D*OSR1 + FS2_SK1D*OSR2 + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR_SK2D*OFFR +
                    FS1_SK2D*OSR1 + FS2_SK2D*OSR2 + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR_SK3D*OFFR +
                    FS1_SK3D*OSR1 + FS2_SK3D*OSR2 + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR = FFR_SK1A + FFR_SK2A + FFR_SK3A + FFR_SK1B + FFR_SK2B +
            FFR_SK3B + FFR_SK1C + FFR_SK2C + FFR_SK3C + FFR_SK1D +
            FFR_SK2D + FFR_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FSR1_WasteB
            + FSR2_WasteB + FSR3_WasteB + FSR1_WasteC + FSR2_WasteC +
            FSR3_WasteC + FSR1_WasteD + FSR2_WasteD + FSR3_WasteD +
            FS1_WasteA + FS2_WasteA + FS3_WasteA + FS1_WasteB +
            FS2_WasteB + FS3_WasteB + FS1_WasteC + FS2_WasteC +
            FS3_WasteC + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
```

```
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + CP*(85-75)*(- FSR2_SK1A - FFR_SK1A - FS1_SK1A -
      FS2_SK1A)*(1000/3600);
H2A = H1A + CP*(75-65)*(- FSR2_SK1A - FFR_SK1A - FS1_SK1A - FS2_SK1A
      + FS3_WasteA)*(1000/3600);
H3A = H2A + CP*(65-55)*(- FFR_SK1A - FS1_SK1A +
      FS3_WasteA)*(1000/3600);
H4A = H3A + CP*(55-45)*(FSR2_WasteA - FFR_SK1A - FS1_SK1A +
      FS2_WasteA + FS3_WasteA)*(1000/3600);
H5A = H4A + CP*(45-35)*(FSR2_WasteA - FFR_SK1A + FS2_WasteA +
      FS3_WasteA)*(1000/3600);
H6A = H5A + CP*(35-25)*(FSR2_WasteA - FFR_SK1A + FS1_WasteA +
      FS2_WasteA + FS3_WasteA)*(1000/3600);
H6A = QCA;

H0B = QHB;
H1B = H0B + CP*(85-75)*(- FSR1_SK1B - FSR2_SK1B - FFR_SK1B -
      FS2_SK1B - FS1_SK1B)*(1000/3600);
H2B = H1B + CP*(75-65)*(- FSR1_SK1B - FSR2_SK1B - FFR_SK1B -
      FS2_SK1B - FS1_SK1B + FS3_SK2B + FS3_WasteB)*(1000/3600);
H3B = H2B + CP*(65-55)*(- FSR1_SK1B - FFR_SK1B - FS1_SK1B + FS3_SK2B
      + FS3_WasteB)*(1000/3600);
H4B = H3B + CP*(55-45)*(- FSR1_SK1B - FSR1_SK2B + FSR2_SK2B +
      FSR2_WasteB - FFR_SK1B - FFR_SK2B + FS2_SK2B + FS2_WasteB -
      FS1_SK1B - FS1_SK2B + FS3_SK2B + FS3_WasteB)*(1000/3600);
H5B = H4B + CP*(45-35)*(FSR2_WasteB - FFR_SK1B - FFR_SK2B +
      FS2_WasteB + FS3_WasteB)*(1000/3600);
H6B = H5B + CP*(35-25)*(FSR1_WasteB + FSR2_WasteB - FFR_SK1B -
      FFR_SK2B + FS2_WasteB + FS1_WasteB + FS3_WasteB)*(1000/3600);
H6B = QCB;

H0C = QHC;
H1C = H0C + CP*(85-75)*(- FSR1_SK1C - FSR2_SK1C - FFR_SK1C -
      FS1_SK1C - FS2_SK1C)*(1000/3600);
```

```
H2C = H1C + CP*(75-65)*(- FSR1_SK1C - FSR2_SK1C + FSR3_SK2C +
      FSR3_WasteC - FFR_SK1C - FS1_SK1C - FS2_SK1C + FS3_SK2C +
      FS3_WasteC)*(1000/3600);
H3C = H2C + CP*(65-55)*(- FSR1_SK1C + FSR3_SK2C + FSR3_WasteC -
      FFR_SK1C - FS1_SK1C + FS3_SK2C + FS3_WasteC)*(1000/3600);
H4C = H3C + CP*(55-45)*(- FSR1_SK1C - FSR1_SK2C + FSR2_SK2C +
      FSR2_WasteC + FSR3_SK2C + FSR3_WasteC - FFR_SK1C
      - FFR_SK2C - FS1_SK1C - FS1_SK2C + FS2_SK2C + FS2_WasteC +
      FS3_SK2C + FS3_WasteC)*(1000/3600);
H5C = H4C + CP*(45-35)*(FSR2_WasteC + FSR3_WasteC - FFR_SK1C -
      FFR_SK2C + FS2_WasteC + FS3_WasteC)*(1000/3600);
H6C = H5C + CP*(35-25)*(FSR1_WasteC + FSR2_WasteC + FSR3_WasteC -
      FFR_SK1C - FFR_SK2C + FS1_WasteC + FS2_WasteC +
      FS3_WasteC)*(1000/3600);
H6C = QCC;

H0D = QHD;
H1D = H0D + CP*(75-55)*(FSR3_SK2D + FSR3_SK3D + FSR3_WasteD +
      FS3_SK2D + FS3_SK3D + FS3_WasteD)*(1000/3600);
H2D = H1D + CP*(55-45)*(- FSR1_SK2D + FSR3_SK2D + FSR3_SK3D +
      FSR3_WasteD - FFR_SK2D - FS1_SK2D + FS2_SK2D + FS2_SK3D +
      FS2_WasteD + FS3_SK2D + FS3_SK3D +  FS3_WasteD)*(1000/3600);
H3D = H2D + CP*(45-35)*(FSR3_SK3D + FSR3_WasteD - FFR_SK2D -
      FFR_SK3D + FS2_SK3D + FS2_WasteD + FS3_SK3D +
      FS3_WasteD)*(1000/3600);
H4D = H3D + CP*(35-25)*(FSR1_WasteD + FSR3_WasteD - FFR_SK2D -
      FFR_SK3D + FS1_WasteD + FS2_WasteD + FS3_WasteD)*(1000/3600);
H4D = QCD;

!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0; H4A>=0; H5A>=0; H6A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0; H5B>=0; H6B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0; H5C>=0; H6C>=0;
H0D>=0; H1D>=0; H2D>=0; H3D>=0; H4D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 4.36;
Total_Cost    =    Cm*Total_FFR*Nb*1000    +    Total_QH*CHU*Nb    +
                   Total_QC*CCU*Nb + CS1 + CS2 + CS3;

!Equation 4.37;
!UC = value for unit conversion;
CS1 = (I/Ib)*(A0*(A1*UC*CSTS1*1000)^d)*AF;
CS2 = (I/Ib)*(A0*(A1*UC*CSTS2*1000)^d)*AF;
CS3 = (I/Ib)*(A0*(A1*UC*CSTS3*1000)^d)*AF;

!Equation 4.39;
CSTS1>= FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA + CSTS1A;
CSTS1>= FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB + CSTS1B;
CSTS1>= FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC + CSTS1C;
CSTS1>= FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD + CSTS1D;

CSTS2>= FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA + CSTS2A;
CSTS2>= FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB + CSTS2B;
```

```
CSTS2>= FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC + CSTS2C;
CSTS2>= FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD + CSTS2D;

CSTS3>= FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA + CSTS3A;
CSTS3>= FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB + CSTS3B;
CSTS3>= FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC + CSTS3C;
CSTS3>= FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD + CSTS3D;

!Equation 4.40;
CSTS1>=0; CSTS2>=0; CSTS3>=0;

!Equation 4.38;
AF = 0.229;

!Unit conversion;
!1 meter cube = 264.1721 gallon;
UC = 0.2642;

!Data;
FSR1 = 100;  FSR2 = 40; FSR3 = 25;
FSK1 = 100;  FSK2 = 40; FSK3 = 25;
OSR1 = 400;  OSR2 = 200; OSR3 = 200; OFFR = 0;
OSK1 = 100;   OSK2 = 0;  OSK3 = 100;
tSR1ST = 0.5; tSR1ET = 2.0;
tSR2ST = 0; tSR2ET = 1.5;
tSR3ST = 1.0; tSR3ET = 2.0;
tSK1ST = 0; tSK1ET = 1.5;
tSK2ST = 0.5; tSK2ET = 2.0;
tSK3ST = 1.5; tSK3ET = 2.0;
Nb = 3960;
Cm = 0.001;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
I = 572.7;
Ib = 394;
A0 = 210;
A1 = 1.1;
d = 0.51;
t0 = 0; t1 = 0.5; t2 = 1; t3 = 1.5; t4 = 2.0;
End

  Global optimal solution found.
  Objective value:                              702207.3
  Objective bound:                              702207.0
  Infeasibilities:                              0.000000
  Extended solver steps:                              52
  Total solver iterations:                         19243


                       Variable           Value        Reduced Cost
                     TOTAL_COST        702207.3            0.000000
                          FSR2A        13.33333            0.000000
                          FSR1B        33.33333            0.000000
                          FSR2B        13.33333            0.000000
                          FSR1C        33.33333            0.000000
                          FSR2C        13.33333            0.000000
                          FSR3C        12.50000            0.000000
                          FSR1D        33.33333            0.000000
```

FSR3D	12.50000	0.000000
FSK1A	33.33333	0.000000
FSK1B	33.33333	0.000000
FSK2B	13.33333	0.000000
FSK1C	33.33333	0.000000
FSK2C	13.33333	0.000000
FSK2D	13.33333	0.000000
FSK3D	25.00000	0.000000
ZSR1A	0.000000	0.000000
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	1.000000	0.000000
ZSR2B	1.000000	0.000000
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	-0.2063317E-01
ZSR3A	0.000000	-38978.60
ZSR3B	0.000000	-35805.02
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	1.000000	0.000000
XSR1D	1.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	1.000000	0.000000
XSR2D	0.000000	-20633.17
XSR3A	1.000000	-0.3897860E-01
XSR3B	1.000000	-0.3580502E-01
XSR3C	1.000000	0.000000
XSR3D	1.000000	0.000000
YSR1A	0.000000	0.000000
YSR1B	1.000000	9240.000
YSR1C	1.000000	9240.000
YSR1D	1.000000	9240.000
YSR2A	1.000000	-62210.41
YSR2B	1.000000	-62210.41
YSR2C	1.000000	-32763.09
YSR2D	0.000000	0.000000
YSR3A	0.000000	0.000000
YSR3B	0.000000	0.000000
YSR3C	1.000000	-68029.86
YSR3D	1.000000	-37823.43
ZSK1A	1.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	0.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	0.000000	0.000000
ZSK3C	0.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000

XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	1.000000	0.000000
XSK2D	1.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	1.000000	0.000000
YSK1A	1.000000	202135.5
YSK1B	1.000000	199434.7
YSK1C	1.000000	213801.7
YSK1D	0.000000	0.000000
YSK2A	0.000000	0.000000
YSK2B	1.000000	70048.00
YSK2C	1.000000	70048.00
YSK2D	1.000000	66300.31
YSK3A	0.000000	0.000000
YSK3B	0.000000	0.000000
YSK3C	0.000000	0.000000
YSK3D	1.000000	77947.17
FSR2_SK1A	10.00000	0.000000
FSR2_S2A	3.333333	0.1547489E-02
FSR1_WASTEB	30.00000	0.000000
FSR1_S1B	3.333333	0.000000
FSR2_SK1B	13.33333	0.000000
FSR1_WASTEC	27.50000	0.000000
FSR1_S1C	5.833333	0.000000
FSR2_SK1C	5.416667	0.000000
FSR2_WASTEC	2.500000	0.000000
FSR2_S2C	5.416667	0.000000
FSR3_SK1C	11.25000	0.000000
FSR3_S3C	1.250000	0.000000
FSR1_WASTED	25.83333	0.000000
FSR1_S1D	7.500000	0.000000
FSR3_SK3D	7.083333	0.000000
FSR3_S3D	5.416667	0.000000
FFR_SK1A	16.66667	0.000000
FS3_SK1A	6.666667	0.000000
FFR_SK1B	16.66667	0.000000
FS2_SK1B	3.333333	0.000000
FFR_SK2B	13.33333	0.000000
FFR_SK1C	16.66667	0.000000
FFR_SK2C	13.33333	0.000000
FFR_SK2D	13.33333	0.000000
FFR_SK3D	12.50000	0.000000
FS2_SK3D	5.416667	0.000000
TOTAL_FFR	102.5000	0.000000
TOTAL_WASTE	102.5000	0.000000
FS1_WASTEA	16.66667	0.000000
CSTS1C	3.333333	0.000000
CSTS1D	9.166667	0.000000
CSTS3D	1.250000	0.000000
H0A	1205.556	0.000000
QHA	1205.556	0.000000
H1A	894.4444	0.000000
H2A	583.3333	0.000000
H3A	388.8889	0.000000
H4A	194.4444	0.000000

```
        QCA        0.000000             0.000000
        H0B        1672.222             0.000000
        QHB        1672.222             0.000000
        H1B        1283.333             0.000000
        H2B        894.4444             0.000000
        H3B        700.0000             0.000000
        H4B        350.0000             0.000000
        QCB        0.000000             0.000000
        H0C        1351.389             0.000000
        QHC        1351.389             0.000000
        H1C        1093.750             0.000000
        H2C        836.1111             0.000000
        H3C        641.6667             0.000000
        H4C        320.8333             0.000000
        QCC        0.000000             0.000000
        QHD        0.000000             0.000000
        H1D        165.2778             0.000000
        H2D        155.5556             0.000000
        QCD        0.000000             0.000000
   TOTAL_QH        4229.167             0.000000
   TOTAL_QC        0.000000             0.000000
      CSTS1        16.66667             0.000000
      CSTS2        5.416667             0.000000
      CSTS3        6.666667             0.000000
```

Note: The flow and heat terms which are not shown are equal to zero.

A.2 Sequential approach with storage system (Scenario 3)

Part 1 Minimizing the cost for fresh resource

```
!Total_Cost_FR = Total cost for fresh resource;
min = Total_Cost_FR;

!Equation 4.3;
FSR1A  =  FSR1*tA/(tSR1ET-tSR1ST);  FSR2A  =  FSR2*tA/(tSR2ET-tSR2ST);
FSR3A  = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B  =  FSR1*tB/(tSR1ET-tSR1ST);  FSR2B  =  FSR2*tB/(tSR2ET-tSR2ST);
FSR3B  = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C  =  FSR1*tC/(tSR1ET-tSR1ST);  FSR2C  =  FSR2*tC/(tSR2ET-tSR2ST);
FSR3C  = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D  =  FSR1*tD/(tSR1ET-tSR1ST);  FSR2D  =  FSR2*tD/(tSR2ET-tSR2ST);
FSR3D  = FSR3*tD/(tSR3ET-tSR3ST);

!Equation 4.4;
FSK1A  =  FSK1*tA/(tSK1ET-tSK1ST);  FSK2A  =  FSK2*tA/(tSK2ET-tSK2ST);
FSK3A  = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B  =  FSK1*tB/(tSK1ET-tSK1ST);  FSK2B  =  FSK2*tB/(tSK2ET-tSK2ST);
FSK3B  = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C  =  FSK1*tC/(tSK1ET-tSK1ST);  FSK2C  =  FSK2*tC/(tSK2ET-tSK2ST);
FSK3C  = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D  =  FSK1*tD/(tSK1ET-tSK1ST);  FSK2D  =  FSK2*tD/(tSK2ET-tSK2ST);
FSK3D  = FSK3*tD/(tSK3ET-tSK3ST);
```

```
!Equation 4.5;
!A=Time interval 1(0-0.5hr), B=Time interval 2(0.5-1.0hr), C=Time
interval 3(1.0-1.5hr)& D=Time interval 4(1.5-2.0hr);
tA = t1 - t0; tB = t2 - t1; tC = t3 - t2; tD = t4 - t3;

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <= 1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <= 1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <= 1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <= 1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <= 1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <= 1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <= 1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <= 1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <= 1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <= 1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <= 1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <= 1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <= 1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <= 1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <= 1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <= 1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <= 1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <= 1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <= 1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <= 1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <= 1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <= 1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <= 1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <= 1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <= 1000*ZSK1A;
```

```
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <= 1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <= 1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <= 1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <= 1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <= 1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <= 1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <= 1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <= 1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <= 1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <= 1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <= 1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <= 1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <= 1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <= 1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <= 1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <= 1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <= 1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <= 1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <= 1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <= 1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <= 1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <= 1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <= 1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
```

```
               FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR_SK1A + FS1_SK1A + FS2_SK1A + FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR_SK2A + FS1_SK2A + FS2_SK2A + FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR_SK3A + FS1_SK3A + FS2_SK3A + FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR_SK1B + FS1_SK1B + FS2_SK1B + FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR_SK2B + FS1_SK2B + FS2_SK2B + FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR_SK3B + FS1_SK3B + FS2_SK3B + FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR_SK1C + FS1_SK1C + FS2_SK1C + FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR_SK2C + FS1_SK2C + FS2_SK2C + FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR_SK3C + FS1_SK3C + FS2_SK3C + FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR_SK1D + FS1_SK1D + FS2_SK1D + FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR_SK2D + FS1_SK2D + FS2_SK2D + FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR_SK3D + FS1_SK3D + FS2_SK3D + FS3_SK3D;

!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR_SK1A*OFFR +
                    FS1_SK1A*OSR1 + FS2_SK1A*OSR2 + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR_SK2A*OFFR +
                    FS1_SK2A*OSR1 + FS2_SK2A*OSR2 + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR_SK3A*OFFR +
                    FS1_SK3A*OSR1 + FS2_SK3A*OSR2 + FS3_SK3A*OSR3;
```

```
FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR_SK1B*OFFR +
                    FS1_SK1B*OSR1 + FS2_SK1B*OSR2 + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR_SK2B*OFFR +
                    FS1_SK2B*OSR1 + FS2_SK2B*OSR2 + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR_SK3B*OFFR +
                    FS1_SK3B*OSR1 + FS2_SK3B*OSR2 + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR_SK1C*OFFR +
                    FS1_SK1C*OSR1 + FS2_SK1C*OSR2 + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR_SK2C*OFFR +
                    FS1_SK2C*OSR1 + FS2_SK2C*OSR2 + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR_SK3C*OFFR +
                    FS1_SK3C*OSR1 + FS2_SK3C*OSR2 + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR_SK1D*OFFR +
                    FS1_SK1D*OSR1 + FS2_SK1D*OSR2 + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR_SK2D*OFFR +
                    FS1_SK2D*OSR1 + FS2_SK2D*OSR2 + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR_SK3D*OFFR +
                    FS1_SK3D*OSR1 + FS2_SK3D*OSR2 + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR = FFR_SK1A + FFR_SK2A + FFR_SK3A + FFR_SK1B + FFR_SK2B +
            FFR_SK3B + FFR_SK1C + FFR_SK2C + FFR_SK3C + FFR_SK1D +
            FFR_SK2D + FFR_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FSR1_WasteB
            + FSR2_WasteB + FSR3_WasteB + FSR1_WasteC + FSR2_WasteC
            + FSR3_WasteC + FSR1_WasteD + FSR2_WasteD + FSR3_WasteD +
            FS1_WasteA  +  FS2_WasteA  +  FS3_WasteA  +  FS1_WasteB  +
            FS2_WasteB  +  FS3_WasteB  +  FS1_WasteC  +  FS2_WasteC  +
            FS3_WasteC + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;
```

```
FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equation 4.41;
Total_Cost_FR = Cm*Total_FFR*Nb*1000;

!Data;
FSR1 = 100;  FSR2 = 40; FSR3 = 25;
FSK1 = 100;  FSK2 = 40; FSK3 = 25;
OSR1 = 400;  OSR2 = 200; OSR3 = 200; OFFR = 0;
OSK1 = 100;   OSK2 = 0;  OSK3 = 100;
tSR1ST = 0.5; tSR1ET = 2.0;
tSR2ST = 0; tSR2ET = 1.5;
tSR3ST = 1.0; tSR3ET = 2.0;
tSK1ST = 0; tSK1ET = 1.5;
tSK2ST = 0.5; tSK2ET = 2.0;
tSK3ST = 1.5; tSK3ET = 2.0;
Nb = 3960;
Cm = 0.001;
t0 = 0; t1 = 0.5; t2 = 1; t3 = 1.5; t4 = 2.0;
End

  Local optimal solution found.
  Objective value:                              405900.0
  Objective bound:                              405900.0
  Infeasibilities:                             0.2842171E-13
  Extended solver steps:                               0
  Total solver iterations:                           132
```

Variable	Value	Reduced Cost
TOTAL_COST_FR	405900.0	0.000000
FSR2A	13.33333	0.000000
FSR1B	33.33333	0.000000
FSR2B	13.33333	0.000000
FSR1C	33.33333	0.000000
FSR2C	13.33333	0.000000
FSR3C	12.50000	0.000000
FSR1D	33.33333	0.000000
FSR3D	12.50000	0.000000
FSK1A	33.33333	0.000000
FSK1B	33.33333	0.000000
FSK2B	13.33333	0.000000
FSK1C	33.33333	0.000000
FSK2C	13.33333	0.000000
FSK2D	13.33333	0.000000
FSK3D	25.00000	0.000000
ZSR1A	0.000000	0.000000

ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	1.000000	0.000000
ZSR2B	1.000000	0.000000
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	0.000000
ZSR3A	0.000000	0.000000
ZSR3B	0.000000	0.000000
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	1.000000	0.000000
XSR1D	1.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	1.000000	0.000000
XSR2D	0.000000	0.000000
XSR3A	1.000000	0.000000
XSR3B	1.000000	0.000000
XSR3C	1.000000	0.000000
XSR3D	1.000000	0.000000
YSR1A	0.000000	0.000000
YSR1B	1.000000	0.000000
YSR1C	1.000000	0.000000
YSR1D	1.000000	0.000000
YSR2A	1.000000	0.000000
YSR2B	1.000000	0.000000
YSR2C	1.000000	0.000000
YSR2D	0.000000	0.000000
YSR3A	0.000000	0.000000
YSR3B	0.000000	0.000000
YSR3C	1.000000	0.000000
YSR3D	1.000000	0.000000
ZSK1A	1.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	0.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	0.000000	0.000000
ZSK3C	0.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	1.000000	0.000000
XSK2D	1.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	1.000000	0.000000

YSK1A	1.000000	66000.00
YSK1B	1.000000	66000.00
YSK1C	1.000000	66000.00
YSK1D	0.000000	0.000000
YSK2A	0.000000	0.000000
YSK2B	1.000000	52800.00
YSK2C	1.000000	52800.00
YSK2D	1.000000	52800.00
YSK3A	0.000000	0.000000
YSK3B	0.000000	0.000000
YSK3C	0.000000	0.000000
YSK3D	1.000000	49500.00
FSR2_SK1A	8.412698	0.000000
FSR2_S2A	4.920635	0.000000
FSR1_WASTEB	33.33333	0.000000
FSR2_SK1B	11.74603	0.000000
FSR2_WASTEB	1.587302	0.000000
FSR1_WASTEC	33.33333	0.000000
FSR2_SK1C	7.853175	0.000000
FSR2_WASTEC	0.1626984	0.000000
FSR2_S2C	5.317460	0.000000
FSR3_SK1C	8.813492	0.000000
FSR3_WASTEC	0.7500000	0.000000
FSR3_S3C	2.936508	0.000000
FSR1_WASTED	33.33333	0.000000
FSR3_SK3D	12.50000	0.000000
FFR_SK1A	16.66667	0.000000
FS1_SK1A	0.000000	3960.000
FS2_SK1A	5.317460	0.000000
FS3_SK1A	2.936508	0.000000
FFR_SK1B	16.66667	0.000000
FS2_SK1B	4.920635	0.000000
FFR_SK2B	13.33333	0.000000
FFR_SK1C	16.66667	0.000000
FFR_SK2C	13.33333	0.000000
FFR_SK2D	13.33333	0.000000
FFR_SK3D	12.50000	0.000000
TOTAL_FFR	102.5000	0.000000
TOTAL_WASTE	102.5000	0.000000
CSTS2D	5.317460	0.000000
CSTS3D	2.936508	0.000000

Note: The flow terms which are not shown are equal to zero.

Part 2 Minimizing the total operating cost

```
!Total_Cost_O = TOC;
min = Total_Cost_O;

!Equation 4.3;
FSR1A  =  FSR1*tA/(tSR1ET-tSR1ST);  FSR2A  =  FSR2*tA/(tSR2ET-tSR2ST);
FSR3A  = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B  =  FSR1*tB/(tSR1ET-tSR1ST);  FSR2B  =  FSR2*tB/(tSR2ET-tSR2ST);
FSR3B  = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C  =  FSR1*tC/(tSR1ET-tSR1ST);  FSR2C  =  FSR2*tC/(tSR2ET-tSR2ST);
FSR3C  = FSR3*tC/(tSR3ET-tSR3ST);
```

```
FSR1D  =  FSR1*tD/(tSR1ET-tSR1ST);  FSR2D  =  FSR2*tD/(tSR2ET-tSR2ST);
FSR3D  = FSR3*tD/(tSR3ET-tSR3ST);

!Equation 4.4;
FSK1A  =  FSK1*tA/(tSK1ET-tSK1ST);  FSK2A  =  FSK2*tA/(tSK2ET-tSK2ST);
FSK3A  = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B  =  FSK1*tB/(tSK1ET-tSK1ST);  FSK2B  =  FSK2*tB/(tSK2ET-tSK2ST);
FSK3B  = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C  =  FSK1*tC/(tSK1ET-tSK1ST);  FSK2C  =  FSK2*tC/(tSK2ET-tSK2ST);
FSK3C  = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D  =  FSK1*tD/(tSK1ET-tSK1ST);  FSK2D  =  FSK2*tD/(tSK2ET-tSK2ST);
FSK3D  = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time  interval  1(0-0.5hr),  B=Time  interval  2(0.5-1.0hr),  C=Time
interval 3(1.0-1.5hr)& D=Time interval 4(1.5-2.0hr);
tA = t1 - t0; tB = t2 - t1; tC = t3 - t2; tD = t4 - t3;

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <= 1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <= 1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <= 1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <= 1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <= 1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <= 1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <= 1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <= 1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <= 1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <= 1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <= 1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <= 1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <= 1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <= 1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <= 1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <= 1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <= 1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <= 1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <= 1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <= 1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <= 1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <= 1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <= 1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <= 1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
```

```
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <= 1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <= 1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <= 1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <= 1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <= 1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <= 1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <= 1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <= 1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <= 1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <= 1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <= 1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <= 1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <= 1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <= 1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <= 1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <= 1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <= 1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <= 1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <= 1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <= 1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <= 1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <= 1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <= 1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <= 1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);
```

```
@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR_SK1A + FS1_SK1A + FS2_SK1A + FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR_SK2A + FS1_SK2A + FS2_SK2A + FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR_SK3A + FS1_SK3A + FS2_SK3A + FS3_SK3A;
FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR_SK1B + FS1_SK1B + FS2_SK1B + FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR_SK2B + FS1_SK2B + FS2_SK2B + FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR_SK3B + FS1_SK3B + FS2_SK3B + FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR_SK1C + FS1_SK1C + FS2_SK1C + FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR_SK2C + FS1_SK2C + FS2_SK2C + FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR_SK3C + FS1_SK3C + FS2_SK3C + FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR_SK1D + FS1_SK1D + FS2_SK1D + FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR_SK2D + FS1_SK2D + FS2_SK2D + FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
```

```
              FFR_SK3D + FS1_SK3D + FS2_SK3D + FS3_SK3D;

!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR_SK1A*OFFR +
                    FS1_SK1A*OSR1 + FS2_SK1A*OSR2 + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR_SK2A*OFFR +
                    FS1_SK2A*OSR1 + FS2_SK2A*OSR2 + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR_SK3A*OFFR +
                    FS1_SK3A*OSR1 + FS2_SK3A*OSR2 + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR_SK1B*OFFR +
                    FS1_SK1B*OSR1 + FS2_SK1B*OSR2 + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR_SK2B*OFFR +
                    FS1_SK2B*OSR1 + FS2_SK2B*OSR2 + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR_SK3B*OFFR +
                    FS1_SK3B*OSR1 + FS2_SK3B*OSR2 + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR_SK1C*OFFR +
                    FS1_SK1C*OSR1 + FS2_SK1C*OSR2 + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR_SK2C*OFFR +
                    FS1_SK2C*OSR1 + FS2_SK2C*OSR2 + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR_SK3C*OFFR +
                    FS1_SK3C*OSR1 + FS2_SK3C*OSR2 + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR_SK1D*OFFR +
                    FS1_SK1D*OSR1 + FS2_SK1D*OSR2 + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR_SK2D*OFFR +
                    FS1_SK2D*OSR1 + FS2_SK2D*OSR2 + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR_SK3D*OFFR +
                    FS1_SK3D*OSR1 + FS2_SK3D*OSR2 + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR = FFR_SK1A + FFR_SK2A + FFR_SK3A + FFR_SK1B + FFR_SK2B +
            FFR_SK3B + FFR_SK1C + FFR_SK2C + FFR_SK3C + FFR_SK1D +
            FFR_SK2D + FFR_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FSR1_WasteB
            + FSR2_WasteB + FSR3_WasteB + FSR1_WasteC + FSR2_WasteC
            + FSR3_WasteC + FSR1_WasteD + FSR2_WasteD + FSR3_WasteD +
            FS1_WasteA  +  FS2_WasteA  +  FS3_WasteA  +  FS1_WasteB  +
            FS2_WasteB  +  FS3_WasteB  +  FS1_WasteC  +  FS2_WasteC  +
            FS3_WasteC + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
```

```
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + CP*(85-75)*(- FSR2_SK1A - FFR_SK1A - FS1_SK1A -
      FS2_SK1A)*(1000/3600);
H2A = H1A + CP*(75-65)*(- FSR2_SK1A - FFR_SK1A - FS1_SK1A - FS2_SK1A
      + FS3_WasteA)*(1000/3600);
H3A = H2A + CP*(65-55)*(- FFR_SK1A - FS1_SK1A +
      FS3_WasteA)*(1000/3600);
H4A = H3A + CP*(55-45)*(FSR2_WasteA - FFR_SK1A - FS1_SK1A +
      FS2_WasteA + FS3_WasteA)*(1000/3600);
H5A = H4A + CP*(45-35)*(FSR2_WasteA - FFR_SK1A + FS2_WasteA +
      FS3_WasteA)*(1000/3600);
H6A = H5A + CP*(35-25)*(FSR2_WasteA - FFR_SK1A + FS1_WasteA +
      FS2_WasteA + FS3_WasteA)*(1000/3600);
H6A = QCA;

H0B = QHB;
H1B = H0B + CP*(85-75)*(- FSR1_SK1B - FSR2_SK1B - FFR_SK1B -
      FS2_SK1B - FS1_SK1B)*(1000/3600);
H2B = H1B + CP*(75-65)*(- FSR1_SK1B - FSR2_SK1B - FFR_SK1B -
      FS2_SK1B - FS1_SK1B + FS3_SK2B + FS3_WasteB)*(1000/3600);
H3B = H2B + CP*(65-55)*(- FSR1_SK1B - FFR_SK1B - FS1_SK1B + FS3_SK2B
      + FS3_WasteB)*(1000/3600);
H4B = H3B + CP*(55-45)*(- FSR1_SK1B - FSR1_SK2B + FSR2_SK2B +
      FSR2_WasteB - FFR_SK1B - FFR_SK2B + FS2_SK2B + FS2_WasteB -
      FS1_SK1B - FS1_SK2B + FS3_SK2B +  FS3_WasteB)*(1000/3600);
H5B = H4B + CP*(45-35)*(FSR2_WasteB - FFR_SK1B - FFR_SK2B +
      FS2_WasteB + FS3_WasteB)*(1000/3600);
H6B = H5B + CP*(35-25)*(FSR1_WasteB + FSR2_WasteB - FFR_SK1B -
      FFR_SK2B + FS2_WasteB + FS1_WasteB + FS3_WasteB)*(1000/3600);
H6B = QCB;

H0C = QHC;
```

```
H1C = H0C + CP*(85-75)*(- FSR1_SK1C - FSR2_SK1C - FFR_SK1C -
      FS1_SK1C - FS2_SK1C)*(1000/3600);
H2C = H1C + CP*(75-65)*(- FSR1_SK1C - FSR2_SK1C + FSR3_SK2C +
      FSR3_WasteC - FFR_SK1C - FS1_SK1C - FS2_SK1C
      + FS3_SK2C + FS3_WasteC)*(1000/3600);
H3C = H2C + CP*(65-55)*(- FSR1_SK1C + FSR3_SK2C + FSR3_WasteC -
      FFR_SK1C - FS1_SK1C + FS3_SK2C + FS3_WasteC)*(1000/3600);
H4C = H3C + CP*(55-45)*(- FSR1_SK1C - FSR1_SK2C + FSR2_SK2C +
      FSR2_WasteC + FSR3_SK2C + FSR3_WasteC - FFR_SK1C
      - FFR_SK2C - FS1_SK1C - FS1_SK2C + FS2_SK2C + FS2_WasteC +
      FS3_SK2C + FS3_WasteC)*(1000/3600);
H5C = H4C + CP*(45-35)*(FSR2_WasteC + FSR3_WasteC - FFR_SK1C -
      FFR_SK2C + FS2_WasteC + FS3_WasteC)*(1000/3600);
H6C = H5C + CP*(35-25)*(FSR1_WasteC + FSR2_WasteC + FSR3_WasteC -
      FFR_SK1C - FFR_SK2C + FS1_WasteC + FS2_WasteC +
      FS3_WasteC)*(1000/3600);
H6C = QCC;

H0D = QHD;
H1D = H0D + CP*(75-55)*(FSR3_SK2D + FSR3_SK3D + FSR3_WasteD +
      FS3_SK2D + FS3_SK3D + FS3_WasteD)*(1000/3600);
H2D = H1D + CP*(55-45)*(- FSR1_SK2D + FSR3_SK2D + FSR3_SK3D +
      FSR3_WasteD - FFR_SK2D - FS1_SK2D + FS2_SK2D + FS2_SK3D +
      FS2_WasteD + FS3_SK2D + FS3_SK3D +  FS3_WasteD)*(1000/3600);
H3D = H2D + CP*(45-35)*(FSR3_SK3D + FSR3_WasteD - FFR_SK2D -
      FFR_SK3D + FS2_SK3D + FS2_WasteD + FS3_SK3D +
      FS3_WasteD)*(1000/3600);
H4D = H3D + CP*(35-25)*(FSR1_WasteD + FSR3_WasteD - FFR_SK2D -
      FFR_SK3D + FS1_WasteD + FS2_WasteD + FS3_WasteD)*(1000/3600);
H4D = QCD;

!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0; H4A>=0; H5A>=0; H6A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0; H5B>=0; H6B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0; H5C>=0; H6C>=0;
H0D>=0; H1D>=0; H2D>=0; H3D>=0; H4D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 4.37;
!UC = value for unit conversion;
CS1 = (I/Ib)*(A0*(A1*UC*CSTS1*1000)^d)*AF;
CS2 = (I/Ib)*(A0*(A1*UC*CSTS2*1000)^d)*AF;
CS3 = (I/Ib)*(A0*(A1*UC*CSTS3*1000)^d)*AF;

!Equation 4.39;
CSTS1>= FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA + CSTS1A;
CSTS1>= FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB + CSTS1B;
CSTS1>= FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC + CSTS1C;
CSTS1>= FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD + CSTS1D;

CSTS2>= FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA + CSTS2A;
CSTS2>= FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB + CSTS2B;
CSTS2>= FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC + CSTS2C;
CSTS2>= FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD + CSTS2D;
```

```
CSTS3>= FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA + CSTS3A;
CSTS3>= FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB + CSTS3B;
CSTS3>= FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC + CSTS3C;
CSTS3>= FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD + CSTS3D;

!Equation 4.40;
CSTS1>=0; CSTS2>=0; CSTS3>=0;

!Equation 4.38;
AF = 0.229;

!Unit conversion;
!1 meter cube = 264.1721 gallon;
UC = 0.2642;

!Additional constraint (Equation 5.3);
!Total_FFR = Fm;
Total_FFR = 102.5;

!Equation 4.42;
Total_Cost_O = Total_QH*CHU*Nb + Total_QC*CCU*Nb;

!Data;
FSR1 = 100;  FSR2 = 40; FSR3 = 25;
FSK1 = 100;  FSK2 = 40; FSK3 = 25;
OSR1 = 400;  OSR2 = 200; OSR3 = 200; OFFR = 0;
OSK1 = 100;   OSK2 = 0;  OSK3 = 100;
tSR1ST = 0.5; tSR1ET = 2.0;
tSR2ST = 0; tSR2ET = 1.5;
tSR3ST = 1.0; tSR3ET = 2.0;
tSK1ST = 0; tSK1ET = 1.5;
tSK2ST = 0.5; tSK2ET = 2.0;
tSK3ST = 1.5; tSK3ET = 2.0;
Nb = 3960;
Cm = 0.001;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
I = 572.7;
Ib = 394;
A0 = 210;
A1 = 1.1;
d = 0.51;
t0 = 0; t1 = 0.5; t2 = 1; t3 = 1.5; t4 = 2.0;
End

  Global optimal solution found.
  Objective value:                              284707.5
  Objective bound:                              284707.5
  Infeasibilities:                              0.000000
  Extended solver steps:                               1
  Total solver iterations:                           632


                       Variable           Value        Reduced Cost
                   TOTAL_COST_O        284707.5            0.000000
                          FSR2A        13.33333            0.000000
                          FSR1B        33.33333            0.000000
```

FSR2B	13.33333	0.000000
FSR1C	33.33333	0.000000
FSR2C	13.33333	0.000000
FSR3C	12.50000	0.000000
FSR1D	33.33333	0.000000
FSR3D	12.50000	0.000000
FSK1A	33.33333	0.000000
FSK1B	33.33333	0.000000
FSK2B	13.33333	0.000000
FSK1C	33.33333	0.000000
FSK2C	13.33333	0.000000
FSK2D	13.33333	0.000000
FSK3D	25.00000	0.000000
ZSR1A	0.000000	0.000000
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	1.000000	0.000000
ZSR2B	1.000000	0.000000
ZSR2C	1.000000	-29182.91
ZSR2D	1.000000	-0.1724800E-01
ZSR3A	0.000000	0.000000
ZSR3B	0.000000	-35805.00
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	1.000000	0.000000
XSR1D	1.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	1.000000	-29182.91
XSR2D	0.000000	-17248.00
XSR3A	1.000000	0.000000
XSR3B	1.000000	-0.3580500E-01
XSR3C	1.000000	0.000000
XSR3D	1.000000	0.000000
YSR1A	0.000000	0.000000
YSR1B	1.000000	9240.000
YSR1C	1.000000	9240.000
YSR1D	1.000000	9240.000
YSR2A	1.000000	-55440.06
YSR2B	1.000000	-55440.06
YSR2C	1.000000	0.000000
YSR2D	0.000000	0.000000
YSR3A	0.000000	0.000000
YSR3B	0.000000	0.000000
YSR3C	1.000000	-71610.04
YSR3D	1.000000	-33841.51
ZSK1A	1.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	0.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	0.000000	0.000000

ZSK3C	0.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	1.000000	0.000000
XSK2D	1.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	1.000000	0.000000
YSK1A	1.000000	112728.2
YSK1B	1.000000	112728.2
YSK1C	1.000000	136675.1
YSK1D	0.000000	-26179.76
YSK2A	0.000000	0.000000
YSK2B	1.000000	6776.095
YSK2C	1.000000	6776.095
YSK2D	1.000000	6776.057
YSK3A	0.000000	0.000000
YSK3B	0.000000	-19634.82
YSK3C	0.000000	-19634.82
YSK3D	1.000000	13359.72
FSR2_SK1A	13.33333	0.000000
FSR1_WASTEB	30.00000	0.000000
FSR1_S1B	3.333333	0.000000
FSR2_SK1B	13.33333	0.000000
FSR1_WASTEC	26.66667	0.000000
FSR1_S1C	6.666667	0.000000
FSR2_SK1C	4.166667	0.000000
FSR2_S2C	9.166667	0.000000
FSR3_SK1C	12.50000	0.000000
FSR1_WASTED	19.16667	0.000000
FSR1_S1D	14.16667	0.000000
FSR3_SK3D	7.083333	0.000000
FSR3_S3D	5.416667	0.000000
FFR_SK1A	16.66667	0.000000
FS2_SK1A	1.250000	0.000000
FS3_SK1A	2.083333	0.000000
FFR_SK1B	16.66667	0.000000
FS3_SK1B	3.333333	0.000000
FFR_SK2B	13.33333	0.000000
FFR_SK1C	16.66667	0.000000
FFR_SK2C	13.33333	0.000000
FFR_SK2D	13.33333	0.000000
FFR_SK3D	12.50000	0.000000
FS2_SK3D	5.416667	0.000000
TOTAL_FFR	102.5000	0.000000
TOTAL_WASTE	102.5000	0.000000
FS1_WASTEA	14.16667	0.000000
FS2_WASTEA	2.500000	0.000000
FS1_WASTEC	3.333333	0.000000
FS1_WASTED	6.666667	0.000000
CSTS3A	3.333333	0.000000
CSTS2D	3.750000	0.000000
H0A	1254.167	0.000000

```
            QHA        1254.167             0.000000
            H1A        889.5833             0.000000
            H2A        525.0000             0.000000
            H3A        330.5556             0.000000
            H4A        165.2778             0.000000
            QCA        0.000000             0.000000
            H0B        1594.444             0.000000
            QHB        1594.444             0.000000
            H1B        1244.444             0.000000
            H2B        894.4444             0.000000
            H3B        700.0000             0.000000
            H4B        350.0000             0.000000
            QCB        0.000000             0.000000
            H0C        1380.556             0.000000
            QHC        1380.556             0.000000
            H1C        1137.500             0.000000
            H2C        894.4444             0.000000
            H3C        700.0000             0.000000
            H4C        350.0000             0.000000
            QCC        0.000000             0.000000
            QHD        0.000000             0.000000
            H1D        165.2778             0.000000
            H2D        155.5556             0.000000
            QCD        0.000000             0.000000
       TOTAL_QH        4229.167             0.000000
       TOTAL_QC        0.000000             0.000000
            CS1        4874.480             0.000000
          CSTS1        14.16667             0.000000
            CS2        3903.998             0.000000
          CSTS2        9.166667             0.000000
            CS3        2985.280             0.000000
          CSTS3        5.416667             0.000000
```

Note: The flow and heat terms which are not shown are equal to zero.

Appendix B - Matching formulation code in LINGO and matching formulation solution from LINGO for Case Study 2

B.1 Simultaneous approach

B.1.1 without storage system (Scenario 1)

```
!Total_Cost_O = TOC;
min = Total_Cost_O;

!Equation 4.3;
FSR1A  =  FSR1*tA/(tSR1ET-tSR1ST);  FSR2A  =  FSR2*tA/(tSR2ET-tSR2ST);
FSR3A = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B  =  FSR1*tB/(tSR1ET-tSR1ST);  FSR2B  =  FSR2*tB/(tSR2ET-tSR2ST);
FSR3B = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C  =  FSR1*tC/(tSR1ET-tSR1ST);  FSR2C  =  FSR2*tC/(tSR2ET-tSR2ST);
FSR3C = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D  =  FSR1*tD/(tSR1ET-tSR1ST);  FSR2D  =  FSR2*tD/(tSR2ET-tSR2ST);
FSR3D = FSR3*tD/(tSR3ET-tSR3ST);
```

```
!Equation 4.4;
FSK1A  =  FSK1*tA/(tSK1ET-tSK1ST);  FSK2A  =  FSK2*tA/(tSK2ET-tSK2ST);
FSK3A = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B  =  FSK1*tB/(tSK1ET-tSK1ST);  FSK2B  =  FSK2*tB/(tSK2ET-tSK2ST);
FSK3B = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C  =  FSK1*tC/(tSK1ET-tSK1ST);  FSK2C  =  FSK2*tC/(tSK2ET-tSK2ST);
FSK3C = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D  =  FSK1*tD/(tSK1ET-tSK1ST);  FSK2D  =  FSK2*tD/(tSK2ET-tSK2ST);
FSK3D = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time interval 1(0-1hr), B=Time interval 2(1-2hr), C=Time interval
3(2-3hr)& D=Time interval 4(3-4hr);
tA = (t1 - t0); tB = (t2 - t1); tC = (t3 - t2); tD = (t4 - t3);

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <= 1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <= 1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <= 1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <= 1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <= 1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <= 1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <= 1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <= 1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <= 1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <= 1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <= 1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <= 1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <= 1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <= 1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <= 1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <= 1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <= 1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <= 1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <= 1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <= 1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <= 1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <= 1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <= 1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <= 1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);
```

```
@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <= 1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <= 1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <= 1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <= 1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <= 1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <= 1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <= 1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <= 1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <= 1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <= 1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <= 1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <= 1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <= 1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <= 1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <= 1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <= 1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <= 1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <= 1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <= 1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <= 1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <= 1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <= 1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <= 1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <= 1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);
```

```
!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR_SK1A + FS1_SK1A + FS2_SK1A + FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR_SK2A + FS1_SK2A + FS2_SK2A + FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR_SK3A + FS1_SK3A + FS2_SK3A + FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR_SK1B + FS1_SK1B + FS2_SK1B + FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR_SK2B + FS1_SK2B + FS2_SK2B + FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR_SK3B + FS1_SK3B + FS2_SK3B + FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR_SK1C + FS1_SK1C + FS2_SK1C + FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR_SK2C + FS1_SK2C + FS2_SK2C + FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR_SK3C + FS1_SK3C + FS2_SK3C + FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR_SK1D + FS1_SK1D + FS2_SK1D + FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR_SK2D + FS1_SK2D + FS2_SK2D + FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR_SK3D + FS1_SK3D + FS2_SK3D + FS3_SK3D;
```

```
!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR_SK1A*OFFR +
                    FS1_SK1A*OSR1 + FS2_SK1A*OSR2 + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR_SK2A*OFFR +
                    FS1_SK2A*OSR1 + FS2_SK2A*OSR2 + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR_SK3A*OFFR +
                    FS1_SK3A*OSR1 + FS2_SK3A*OSR2 + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR_SK1B*OFFR +
                    FS1_SK1B*OSR1 + FS2_SK1B*OSR2 + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR_SK2B*OFFR +
                    FS1_SK2B*OSR1 + FS2_SK2B*OSR2 + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR_SK3B*OFFR +
                    FS1_SK3B*OSR1 + FS2_SK3B*OSR2 + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR_SK1C*OFFR +
                    FS1_SK1C*OSR1 + FS2_SK1C*OSR2 + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR_SK2C*OFFR +
                    FS1_SK2C*OSR1 + FS2_SK2C*OSR2 + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR_SK3C*OFFR +
                    FS1_SK3C*OSR1 + FS2_SK3C*OSR2 + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR_SK1D*OFFR +
                    FS1_SK1D*OSR1 + FS2_SK1D*OSR2 + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR_SK2D*OFFR +
                    FS1_SK2D*OSR1 + FS2_SK2D*OSR2 + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR_SK3D*OFFR +
                    FS1_SK3D*OSR1 + FS2_SK3D*OSR2 + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR = FFR_SK1A + FFR_SK2A + FFR_SK3A + FFR_SK1B + FFR_SK2B +
            FFR_SK3B + FFR_SK1C + FFR_SK2C + FFR_SK3C + FFR_SK1D +
            FFR_SK2D + FFR_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FS1_WasteA +
            FS2_WasteA + FS3_WasteA + FSR1_WasteB + FSR2_WasteB +
            FSR3_WasteB + FS1_WasteB + FS2_WasteB + FS3_WasteB +
            FSR1_WasteC + FSR2_WasteC + FSR3_WasteC + FS1_WasteC +
            FS2_WasteC + FS3_WasteC + FSR1_WasteD + FSR2_WasteD +
            FSR3_WasteD + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
```

```
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.26, 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + (95-80)*CP*(FSR1_SK2A+FSR1_WasteA)*(1000/3600);
H2A = H1A + (80-70)*CP*(FSR1_SK2A+FSR1_WasteA -
      FFR_SK2A)*(1000/3600);
H3A = H2A + (70-25)*CP*(FSR1_WasteA-FFR_SK2A)*(1000/3600);
H3A = QCA;

H0B = QHB;
H1B = H0B + (105-95)*CP*(-FFR_SK1B - FFR_SK3B)*(1000/3600);
H2B = H1B + (95-80)*CP*(FSR1_SK2B+FSR1_WasteB+FSR3_SK2B+FSR3_WasteB-
      FFR_SK1B-FFR_SK3B)*(1000/3600);
H3B = H2B + (80-70)*CP*(FSR1_SK2B+FSR1_WasteB+FSR3_SK2B+FSR3_WasteB-
      FFR_SK1B-FFR_SK2B-FFR_SK3B)*(1000/3600);
H4B = H3B + (70-25)*CP*(FSR1_WasteB+FSR3_WasteB-FFR_SK1B-FFR_SK2B-
      FFR_SK3B)*(1000/3600);
H4B = QCB;

H0C = QHC;
H1C = H0C + (105-95)*CP*(-FSR2_SK1C-FSR2_SK3C-FFR_SK1C-
      FFR_SK3C)*(1000/3600);
H2C = H1C + (95-80)*CP*(-FSR2_SK1C-FSR2_SK3C+FSR3_SK2C+FSR3_WasteC-
      FFR_SK1C-FFR_SK3C)*(1000/3600);
H3C = H2C + (80-70)*CP*(FSR3_SK2C+FSR3_WasteC-FFR_SK1C-FFR_SK2C-
      FFR_SK3C)*(1000/3600);
H4C = H3C + (70-25)*CP*(FSR2_WasteC+FSR3_WasteC-FFR_SK1C-FFR_SK2C-
      FFR_SK3C)*(1000/3600);
H4C = QCC;

H0D = QHD;
H1D = H0D + (80-70)*CP*(-FFR_SK2D)*(1000/3600);
H2D = H1D + (70-25)*CP*(FSR2_WasteD-FFR_SK2D)*(1000/3600);
H2D = QCD;
```

```
!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0;
H0D>=0; H1D>=0; H2D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 5.1;
Total_Cost_O = Cm*Total_FFR*Nb*1000 + Total_QH*CHU*Nb +
            Total_QC*CCU*Nb;

!Equation 5.2;
FSR1_S1A =0; FSR1_S1B=0; FSR1_S1C=0; FSR1_S1D=0;
FSR2_S2A =0; FSR2_S2B=0; FSR2_S2C=0; FSR2_S2D=0;
FSR3_S3A =0; FSR3_S3B=0; FSR3_S3C=0; FSR3_S3D=0;

!Data;
FSR1 = 360;  FSR2 = 144; FSR3 = 600.12;
FSK1 = 360;  FSK2 = 144; FSK3 = 600.12;
OSK1 = 50; OSK2 = 50; OSK3 = 800; OFFR = 0;
OSR1 = 100; OSR2 = 800;  OSR3 = 1100;
tSR1ST = 0; tSR1ET = 2;
tSR2ST = 2; tSR2ET = 4;
tSR3ST = 1; tSR3ET = 3;
tSK1ST = 1; tSK1ET = 3;
tSK2ST = 0; tSK2ET = 4;
tSK3ST = 1; tSK3ET = 3;
Nb = 1980;
Cm = 0.001;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
t0 = 0; t1 = 1; t2 = 2; t3 = 3; t4 = 4;
End
```

```
  Global optimal solution found.
  Objective value:                              1714563.
  Objective bound:                              1714563.
  Infeasibilities:                              0.000000
  Extended solver steps:                               1
  Total solver iterations:                           179


                       Variable           Value        Reduced Cost
                          TOTAL_COST_O  1714563.            0.000000
                          FSR1A       180.0000            0.000000
                          FSR1B       180.0000            0.000000
                          FSR3B       300.0600            0.000000
                          FSR2C       72.00000            0.000000
                          FSR3C       300.0600            0.000000
                          FSR2D       72.00000            0.000000
                          FSK2A       36.00000            0.000000
                          FSK1B       180.0000            0.000000
                          FSK2B       36.00000            0.000000
```

FSK3B	300.0600	0.000000
FSK1C	180.0000	0.000000
FSK2C	36.00000	0.000000
FSK3C	300.0600	0.000000
FSK2D	36.00000	0.000000
ZSR1A	1.000000	0.000000
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	0.000000	0.000000
ZSR2B	0.000000	0.000000
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	0.000000
ZSR3A	0.000000	0.000000
ZSR3B	1.000000	0.000000
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	0.000000	0.000000
XSR1D	0.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	1.000000	0.000000
XSR2D	1.000000	0.000000
XSR3A	1.000000	0.000000
XSR3B	1.000000	0.000000
XSR3C	1.000000	0.000000
XSR3D	0.000000	0.000000
YSR1A	1.000000	185862.6
YSR1B	1.000000	-1766126.
YSR1C	0.000000	0.000000
YSR1D	0.000000	0.000000
YSR2A	0.000000	0.000000
YSR2B	0.000000	0.000000
YSR2C	1.000000	-418338.1
YSR2D	1.000000	46309.72
YSR3A	0.000000	0.000000
YSR3B	1.000000	-1405114.
YSR3C	1.000000	-1306154.
YSR3D	0.000000	0.000000
ZSK1A	0.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	1.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	1.000000	0.000000
ZSK3C	1.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000

XSK2C	1.000000	0.000000
XSK2D	1.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	0.000000	0.000000
YSK1A	0.000000	0.000000
YSK1B	1.000000	1343924.
YSK1C	1.000000	924966.0
YSK1D	0.000000	0.000000
YSK2A	1.000000	-6771.629
YSK2B	1.000000	149650.2
YSK2C	1.000000	149650.2
YSK2D	1.000000	56222.10
YSK3A	0.000000	0.000000
YSK3B	1.000000	2038006.
YSK3C	1.000000	1648766.
YSK3D	0.000000	0.000000
FSR1_SK2A	18.00000	0.000000
FSR1_WASTEA	162.0000	0.000000
FSR1_SK1B	89.98200	0.000000
FSR1_SK3B	90.01800	0.000000
FSR3_SK1B	0.1636364E-02	0.000000
FSR3_SK2B	1.636364	0.000000
FSR3_SK3B	210.0420	0.000000
FSR3_WASTEB	88.38000	0.000000
FSR2_SK1C	11.25000	0.000000
FSR2_SK3C	60.75000	0.000000
FSR3_SK2C	1.636364	0.000000
FSR3_SK3C	174.0436	0.000000
FSR3_WASTEC	124.3800	0.000000
FSR2_SK2D	2.250000	0.000000
FSR2_WASTED	69.75000	0.000000
FFR_SK2A	18.00000	0.000000
FFR_SK1B	90.01636	0.000000
FFR_SK2B	34.36364	0.000000
FFR_SK1C	168.7500	0.000000
FFR_SK2C	34.36364	0.000000
FFR_SK3C	65.26636	0.000000
FFR_SK2D	33.75000	0.000000
TOTAL_FFR	444.5100	0.000000
TOTAL_WASTE	444.5100	0.000000
QHA	0.000000	0.000000
H1A	3150.000	0.000000
H2A	5040.000	0.000000
H3A	12600.00	0.000000
QCA	12600.00	0.000000
H0B	3341.100	0.000000
QHB	3341.100	0.000000
H1B	2290.909	0.000000
H2B	2290.909	0.000000
H3B	1890.000	0.000000
QCB	0.000000	0.000000
H0C	15941.10	0.000000
QHC	15941.10	0.000000
H1C	12370.91	0.000000
H2C	9220.909	0.000000
H3C	7560.000	0.000000
QCC	0.000000	0.000000

```
                H0D        393.7500              0.000000
                QHD        393.7500              0.000000
                H2D        1890.000              0.000000
                QCD        1890.000              0.000000
           TOTAL_QH        19675.95              0.000000
           TOTAL_QC        14490.00              0.000000
```

Note: The flow and heat terms which are not shown are equal to zero.

B.1.2 with storage system (Scenario 2)

```
!Total_Cost = TAC;
min = Total_Cost;

!Equation 4.3;
FSR1A  =  FSR1*tA/(tSR1ET-tSR1ST);  FSR2A  =  FSR2*tA/(tSR2ET-tSR2ST);
FSR3A  = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B  =  FSR1*tB/(tSR1ET-tSR1ST);  FSR2B  =  FSR2*tB/(tSR2ET-tSR2ST);
FSR3B  = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C  =  FSR1*tC/(tSR1ET-tSR1ST);  FSR2C  =  FSR2*tC/(tSR2ET-tSR2ST);
FSR3C  = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D  =  FSR1*tD/(tSR1ET-tSR1ST);  FSR2D  =  FSR2*tD/(tSR2ET-tSR2ST);
FSR3D  = FSR3*tD/(tSR3ET-tSR3ST);

!Equation 4.4;
FSK1A  =  FSK1*tA/(tSK1ET-tSK1ST);  FSK2A  =  FSK2*tA/(tSK2ET-tSK2ST);
FSK3A  = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B  =  FSK1*tB/(tSK1ET-tSK1ST);  FSK2B  =  FSK2*tB/(tSK2ET-tSK2ST);
FSK3B  = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C  =  FSK1*tC/(tSK1ET-tSK1ST);  FSK2C  =  FSK2*tC/(tSK2ET-tSK2ST);
FSK3C  = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D  =  FSK1*tD/(tSK1ET-tSK1ST);  FSK2D  =  FSK2*tD/(tSK2ET-tSK2ST);
FSK3D  = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time interval 1(0-1hr), B=Time interval 2(1-2hr), C=Time interval
3(2-3hr)& D=Time interval 4(3-4hr);
tA = (t1 - t0); tB = (t2 - t1); tC = (t3 - t2); tD = (t4 - t3);

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <= 1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <= 1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <= 1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <= 1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <= 1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <= 1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <= 1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <= 1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <= 1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <= 1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <= 1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <= 1000*ZSR3D;
```

```
!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <= 1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <= 1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <= 1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <= 1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <= 1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <= 1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <= 1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <= 1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <= 1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <= 1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <= 1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <= 1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <= 1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <= 1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <= 1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <= 1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <= 1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <= 1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <= 1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <= 1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <= 1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <= 1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <= 1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <= 1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <= 1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <= 1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <= 1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <= 1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <= 1000*XSK2A;
```

```
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <= 1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <= 1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <= 1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <= 1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <= 1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <= 1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <= 1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;
```

```
!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR_SK1A + FS1_SK1A + FS2_SK1A + FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR_SK2A + FS1_SK2A + FS2_SK2A + FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR_SK3A + FS1_SK3A + FS2_SK3A + FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR_SK1B + FS1_SK1B + FS2_SK1B + FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR_SK2B + FS1_SK2B + FS2_SK2B + FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR_SK3B + FS1_SK3B + FS2_SK3B + FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR_SK1C + FS1_SK1C + FS2_SK1C + FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR_SK2C + FS1_SK2C + FS2_SK2C + FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR_SK3C + FS1_SK3C + FS2_SK3C + FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR_SK1D + FS1_SK1D + FS2_SK1D + FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR_SK2D + FS1_SK2D + FS2_SK2D + FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR_SK3D + FS1_SK3D + FS2_SK3D + FS3_SK3D;

!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR_SK1A*OFFR +
                    FS1_SK1A*OSR1 + FS2_SK1A*OSR2 + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR_SK2A*OFFR +
                    FS1_SK2A*OSR1 + FS2_SK2A*OSR2 + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR_SK3A*OFFR +
                    FS1_SK3A*OSR1 + FS2_SK3A*OSR2 + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR_SK1B*OFFR +
                    FS1_SK1B*OSR1 + FS2_SK1B*OSR2 + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR_SK2B*OFFR +
                    FS1_SK2B*OSR1 + FS2_SK2B*OSR2 + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR_SK3B*OFFR +
                    FS1_SK3B*OSR1 + FS2_SK3B*OSR2 + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR_SK1C*OFFR +
                    FS1_SK1C*OSR1 + FS2_SK1C*OSR2 + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR_SK2C*OFFR +
                    FS1_SK2C*OSR1 + FS2_SK2C*OSR2 + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR_SK3C*OFFR +
                    FS1_SK3C*OSR1 + FS2_SK3C*OSR2 + FS3_SK3C*OSR3;
```

```
FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR_SK1D*OFFR +
                    FS1_SK1D*OSR1 + FS2_SK1D*OSR2 + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR_SK2D*OFFR +
                    FS1_SK2D*OSR1 + FS2_SK2D*OSR2 + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR_SK3D*OFFR +
                    FS1_SK3D*OSR1 + FS2_SK3D*OSR2 + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR = FFR_SK1A + FFR_SK2A + FFR_SK3A + FFR_SK1B + FFR_SK2B +
            FFR_SK3B + FFR_SK1C + FFR_SK2C + FFR_SK3C + FFR_SK1D +
            FFR_SK2D + FFR_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FS1_WasteA +
            FS2_WasteA + FS3_WasteA + FSR1_WasteB + FSR2_WasteB +
            FSR3_WasteB + FS1_WasteB + FS2_WasteB + FS3_WasteB +
            FSR1_WasteC + FSR2_WasteC + FSR3_WasteC + FS1_WasteC +
            FS2_WasteC + FS3_WasteC + FSR1_WasteD + FSR2_WasteD +
            FSR3_WasteD + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + (95-80)*CP*(FSR1_SK2A + FSR1_WasteA + FS1_SK2A +
      FS1_WasteA + FS3_WasteA + FS3_SK2A)*(1000/3600);
H2A = H1A + (80-70)*CP*(FSR1_SK2A + FSR1_WasteA - FFR_SK2A +
      FS1_SK2A + FS1_WasteA + FS3_WasteA + FS3_SK2A)*(1000/3600);
```

```
H3A = H2A + (70-25)*CP*(FSR1_WasteA - FFR_SK2A + FS1_WasteA +
      FS2_WasteA + FS3_WasteA)*(1000/3600);
H3A = QCA;

H0B = QHB;
H1B = H0B + (105-95)*CP*(- FFR_SK1B - FFR_SK3B - FS2_SK1B -
      FS2_SK3B)*(1000/3600);
H2B = H1B + (95-80)*CP*(FSR1_SK2B + FSR1_WasteB + FSR3_SK2B +
      FSR3_WasteB - FFR_SK1B - FFR_SK3B + FS1_SK2B + FS1_WasteB
      - FS2_SK1B - FS2_SK3B + FS3_SK2B +FS3_WasteB)*(1000/3600);
H3B = H2B + (80-70)*CP*(FSR1_SK2B + FSR1_WasteB + FSR3_SK2B +
      FSR3_WasteB - FFR_SK1B - FFR_SK2B - FFR_SK3B + FS1_SK2B +
      FS1_WasteB + FS3_SK2B + FS3_WasteB)*(1000/3600);
H4B = H3B + (70-25)*CP*(FSR1_WasteB + FSR3_WasteB - FFR_SK1B -
      FFR_SK2B - FFR_SK3B + FS1_WasteB + FS2_WasteB +
      FS3_WasteB)*(1000/3600);
H4B = QCB;

H0C = QHC;
H1C = H0C + (105-95)*CP*(- FSR2_SK1C - FSR2_SK3C - FS2_SK1C-FFR_SK1C
      - FFR_SK3C - FS2_SK3C)*(1000/3600);
H2C = H1C + (95-80)*CP*(- FSR2_SK1C - FSR2_SK3C + FSR3_SK2C +
      FSR3_WasteC - FS2_SK1C - FFR_SK1C + FS1_WasteC + FS1_SK2C
      - FFR_SK3C - FS2_SK3C + FS3_SK2C +FS3_WasteC)*(1000/3600);
H3C = H2C + (80-70)*CP*(FSR3_SK2C + FSR3_WasteC -FFR_SK1C +
      FS1_WasteC - FFR_SK2C + FS1_SK2C - FFR_SK3C + FS3_SK2C +
      FS3_WasteC)*(1000/3600);
H4C = H3C + (70-25)*CP*(FSR2_WasteC + FSR3_WasteC - FFR_SK1C +
      FS1_WasteC - FFR_SK2C - FFR_SK3C + FS2_WasteC +
      FS3_WasteC)*(1000/3600);
H4C = QCC;

H0D = QHD;
H1D = H0D + (95-80)*CP*(FS1_SK2D + FS1_WasteD + FS3_SK2D +
      FS3_WasteD)*(1000/3600);
H2D = H1D + (80-70)*CP*(- FFR_SK2D + FS1_SK2D + FS1_WasteD +
      FS3_SK2D + FS3_WasteD)*(1000/3600);
H3D = H2D + (70-25)*CP*(FSR2_WasteD - FFR_SK2D + FS1_WasteD +
      FS2_WasteD + FS3_WasteD)*(1000/3600);
H3D = QCD;

!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0;
H0D>=0; H1D>=0; H2D>=0; H3D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 4.36;
Total_Cost = Cm*Total_FFR*Nb*1000 + Total_QH*CHU*Nb +
            Total_QC*CCU*Nb + CS1 + CS2 + CS3;

!Equation 4.37;
!UC = value for unit conversion;
```

```
CS1 = (I/Ib)*(A0*(A1*UC*CSTS1*1000)^d)*AF;
CS2 = (I/Ib)*(A0*(A1*UC*CSTS2*1000)^d)*AF;
CS3 = (I/Ib)*(A0*(A1*UC*CSTS3*1000)^d)*AF;

!Equation 4.39;
CSTS1>= FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA + CSTS1A;
CSTS1>= FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB + CSTS1B;
CSTS1>= FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC + CSTS1C;
CSTS1>= FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD + CSTS1D;

CSTS2>= FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA + CSTS2A;
CSTS2>= FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB + CSTS2B;
CSTS2>= FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC + CSTS2C;
CSTS2>= FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD + CSTS2D;

CSTS3>= FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA + CSTS3A;
CSTS3>= FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB + CSTS3B;
CSTS3>= FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC + CSTS3C;
CSTS3>= FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD + CSTS3D;

!Equation 4.40;
CSTS1>=0; CSTS2>=0; CSTS3>=0;

!Equation 4.38;
AF = 0.229;

!Unit conversion;
!1 meter cube = 264.1721 gallon;
UC = 0.2642;

!Data;
FSR1 = 360;  FSR2 = 144; FSR3 = 600.12;
FSK1 = 360;  FSK2 = 144; FSK3 = 600.12;
OSK1 = 50; OSK2 = 50; OSK3 = 800; OFFR = 0;
OSR1 = 100; OSR2 = 800;  OSR3 = 1100;
tSR1ST = 0; tSR1ET = 2;
tSR2ST = 2; tSR2ET = 4;
tSR3ST = 1; tSR3ET = 3;
tSK1ST = 1; tSK1ET = 3;
tSK2ST = 0; tSK2ET = 4;
tSK3ST = 1; tSK3ET = 3;
Nb = 1980;
Cm = 0.001;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
I = 572.7;
Ib = 394;
A0 = 210;
A1 = 1.1;
d = 0.51;
t0 = 0; t1 = 1; t2 = 2; t3 = 3; t4 = 4;
End

  Global optimal solution found.
  Objective value:                              724069.3
  Objective bound:                              724068.7
  Infeasibilities:                              0.000000
  Extended solver steps:                             121
```

```
Total solver iterations:                         32761


                                      Variable           Value
                                    TOTAL_COST        724069.3
                                         FSR1A        180.0000
                                         FSR1B        180.0000
                                         FSR3B        300.0600
                                         FSR2C        72.00000
                                         FSR3C        300.0600
                                         FSR2D        72.00000
                                         FSK2A        36.00000
                                         FSK1B        180.0000
                                         FSK2B        36.00000
                                         FSK3B        300.0600
                                         FSK1C        180.0000
                                         FSK2C        36.00000
                                         FSK3C        300.0600
                                         FSK2D        36.00000
                                         ZSR1A        1.000000
                                         ZSR1B        1.000000
                                         ZSR1C        1.000000
                                         ZSR1D        1.000000
                                         ZSR2A        0.000000
                                         ZSR2B        0.000000
                                         ZSR2C        1.000000
                                         ZSR2D        1.000000
                                         ZSR3A        0.000000
                                         ZSR3B        1.000000
                                         ZSR3C        1.000000
                                         ZSR3D        1.000000
                                         XSR1A        1.000000
                                         XSR1B        1.000000
                                         XSR1C        0.000000
                                         XSR1D        0.000000
                                         XSR2A        1.000000
                                         XSR2B        1.000000
                                         XSR2C        1.000000
                                         XSR2D        1.000000
                                         XSR3A        1.000000
                                         XSR3B        1.000000
                                         XSR3C        1.000000
                                         XSR3D        0.000000
                                         YSR1A        1.000000
                                         YSR1B        1.000000
                                         YSR1C        0.000000
                                         YSR1D        0.000000
                                         YSR2A        0.000000
                                         YSR2B        0.000000
                                         YSR2C        1.000000
                                         YSR2D        1.000000
                                         YSR3A        0.000000
                                         YSR3B        1.000000
                                         YSR3C        1.000000
                                         YSR3D        0.000000
                                         ZSK1A        0.000000
                                         ZSK1B        1.000000
                                         ZSK1C        1.000000
                                         ZSK1D        1.000000
```

ZSK2A	1.000000
ZSK2B	1.000000
ZSK2C	1.000000
ZSK2D	1.000000
ZSK3A	0.000000
ZSK3B	1.000000
ZSK3C	1.000000
ZSK3D	1.000000
XSK1A	1.000000
XSK1B	1.000000
XSK1C	1.000000
XSK1D	0.000000
XSK2A	1.000000
XSK2B	1.000000
XSK2C	1.000000
XSK2D	1.000000
XSK3A	1.000000
XSK3B	1.000000
XSK3C	1.000000
XSK3D	0.000000
YSK1A	0.000000
YSK1B	1.000000
YSK1C	1.000000
YSK1D	0.000000
YSK2A	1.000000
YSK2B	1.000000
YSK2C	1.000000
YSK2D	1.000000
YSK3A	0.000000
YSK3B	1.000000
YSK3C	1.000000
YSK3D	0.000000
FSR1_SK2A	10.00000
FSR1_S1A	170.0000
FSR1_SK2B	4.680276
FSR1_S1B	175.3197
FSR3_SK3B	185.5387
FSR3_WASTEB	114.5213
FSR2_SK2C	2.250000
FSR2_SK3C	48.18857
FSR2_S2C	21.56143
FSR3_SK3C	176.3100
FSR3_WASTEC	123.7500
FSR2_WASTED	22.54000
FSR2_S2D	49.46000
FFR_SK2A	25.00000
FS2_SK2A	1.000000
FFR_SK1B	90.00000
FS1_SK1B	90.00000
FFR_SK2B	29.44327
FS1_SK2B	0.2417044
FS2_SK2B	1.634752
FS1_SK3B	79.51659
FS2_SK3B	35.00470
FFR_SK1C	90.00000
FS1_SK1C	90.00000
FFR_SK2C	33.75000
FS1_SK3C	75.56143
FFR_SK2D	25.00000

```
     FS1_SK2D        10.00000
     FS2_SK2D        1.000000
    TOTAL_FFR        293.1933
  TOTAL_WASTE        293.1933
   FS2_WASTEA        25.00000
   FS2_WASTEB        4.921980
   FS2_WASTED        2.460000
       CSTS1B       0.2417044
       CSTS2A        41.56143
       CSTS1C        10.00000
       CSTS2D        18.10143
          QHA        0.000000
          H1A        175.0000
          QCA        0.000000
          H0B        1555.713
          QHB        1555.713
          H1B        97.32500
          QCB        0.000000
          H0C        1864.875
          QHC        1864.875
          H1C        252.6750
          QCC        0.000000
          QHD        0.000000
          H1D        175.0000
          QCD        0.000000
     TOTAL_QH        3420.588
     TOTAL_QC        0.000000
          CS1        17597.08
          CS2        10812.55
          CS3        0.000000
        CSTS1        175.5614
        CSTS2        67.56143
        CSTS3        0.000000
```

Note: The flow and heat terms which are not shown are equal to zero.

B.2 Sequential approach with storage system (Scenario 3)

Part 1 Minimizing the cost for fresh resource

```
!Total_Cost_FFR = Total cost for fresh resource;
min = Total_Cost_FFR;

!Equation 4.3;
FSR1A  =  FSR1*tA/(tSR1ET-tSR1ST);  FSR2A  =  FSR2*tA/(tSR2ET-tSR2ST);
FSR3A  = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B  =  FSR1*tB/(tSR1ET-tSR1ST);  FSR2B  =  FSR2*tB/(tSR2ET-tSR2ST);
FSR3B  = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C  =  FSR1*tC/(tSR1ET-tSR1ST);  FSR2C  =  FSR2*tC/(tSR2ET-tSR2ST);
FSR3C  = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D  =  FSR1*tD/(tSR1ET-tSR1ST);  FSR2D  =  FSR2*tD/(tSR2ET-tSR2ST);
FSR3D  = FSR3*tD/(tSR3ET-tSR3ST);
```

```
!Equation 4.4;
FSK1A  =  FSK1*tA/(tSK1ET-tSK1ST);  FSK2A  =  FSK2*tA/(tSK2ET-tSK2ST);
FSK3A  = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B  =  FSK1*tB/(tSK1ET-tSK1ST);  FSK2B  =  FSK2*tB/(tSK2ET-tSK2ST);
FSK3B  = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C  =  FSK1*tC/(tSK1ET-tSK1ST);  FSK2C  =  FSK2*tC/(tSK2ET-tSK2ST);
FSK3C  = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D  =  FSK1*tD/(tSK1ET-tSK1ST);  FSK2D  =  FSK2*tD/(tSK2ET-tSK2ST);
FSK3D  = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time interval 1(0-1hr), B=Time interval 2(1-2hr), C=Time interval
3(2-3hr)& D=Time interval 4(3-4hr);
tA = (t1 - t0); tB = (t2 - t1); tC = (t3 - t2); tD = (t4 - t3);

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <= 1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <= 1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <= 1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <= 1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <= 1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <= 1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <= 1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <= 1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <= 1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <= 1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <= 1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <= 1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <= 1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <= 1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <= 1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <= 1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <= 1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <= 1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <= 1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <= 1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <= 1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <= 1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <= 1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <= 1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);
```

```
@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <= 1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <= 1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <= 1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <= 1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <= 1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <= 1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <= 1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <= 1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <= 1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <= 1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <= 1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <= 1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <= 1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <= 1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <= 1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <= 1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <= 1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <= 1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <= 1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <= 1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <= 1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <= 1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <= 1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <= 1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);
```

```
!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR_SK1A + FS1_SK1A + FS2_SK1A + FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR_SK2A + FS1_SK2A + FS2_SK2A + FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR_SK3A + FS1_SK3A + FS2_SK3A + FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR_SK1B + FS1_SK1B + FS2_SK1B + FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR_SK2B + FS1_SK2B + FS2_SK2B + FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR_SK3B + FS1_SK3B + FS2_SK3B + FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR_SK1C + FS1_SK1C + FS2_SK1C + FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR_SK2C + FS1_SK2C + FS2_SK2C + FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR_SK3C + FS1_SK3C + FS2_SK3C + FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR_SK1D + FS1_SK1D + FS2_SK1D + FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR_SK2D + FS1_SK2D + FS2_SK2D + FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR_SK3D + FS1_SK3D + FS2_SK3D + FS3_SK3D;
```

```
!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR_SK1A*OFFR +
                    FS1_SK1A*OSR1 + FS2_SK1A*OSR2 + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR_SK2A*OFFR +
                    FS1_SK2A*OSR1 + FS2_SK2A*OSR2 + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR_SK3A*OFFR +
                    FS1_SK3A*OSR1 + FS2_SK3A*OSR2 + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR_SK1B*OFFR +
                    FS1_SK1B*OSR1 + FS2_SK1B*OSR2 + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR_SK2B*OFFR +
                    FS1_SK2B*OSR1 + FS2_SK2B*OSR2 + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR_SK3B*OFFR +
                    FS1_SK3B*OSR1 + FS2_SK3B*OSR2 + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR_SK1C*OFFR +
                    FS1_SK1C*OSR1 + FS2_SK1C*OSR2 + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR_SK2C*OFFR +
                    FS1_SK2C*OSR1 + FS2_SK2C*OSR2 + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR_SK3C*OFFR +
                    FS1_SK3C*OSR1 + FS2_SK3C*OSR2 + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR_SK1D*OFFR +
                    FS1_SK1D*OSR1 + FS2_SK1D*OSR2 + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR_SK2D*OFFR +
                    FS1_SK2D*OSR1 + FS2_SK2D*OSR2 + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR_SK3D*OFFR +
                    FS1_SK3D*OSR1 + FS2_SK3D*OSR2 + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR = FFR_SK1A + FFR_SK2A + FFR_SK3A + FFR_SK1B + FFR_SK2B +
            FFR_SK3B + FFR_SK1C + FFR_SK2C + FFR_SK3C + FFR_SK1D +
            FFR_SK2D + FFR_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FS1_WasteA +
              FS2_WasteA + FS3_WasteA + FSR1_WasteB + FSR2_WasteB +
              FSR3_WasteB + FS1_WasteB + FS2_WasteB + FS3_WasteB +
              FSR1_WasteC + FSR2_WasteC + FSR3_WasteC + FS1_WasteC +
              FS2_WasteC + FS3_WasteC + FSR1_WasteD + FSR2_WasteD +
              FSR3_WasteD + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
```

```
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;
FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equation 4.41;
Total_Cost_FFR = Cm*Total_FFR*Nb*1000;

!Data;
FSR1 = 360; FSR2 = 144; FSR3 = 600.12;
FSK1 = 360; FSK2 = 144; FSK3 = 600.12;
OSK1 = 50; OSK2 = 50; OSK3 = 800; OFFR = 0;
OSR1 = 100; OSR2 = 800; OSR3 = 1100;
tSR1ST = 0; tSR1ET = 2;
tSR2ST = 2; tSR2ET = 4;
tSR3ST = 1; tSR3ET = 3;
tSK1ST = 1; tSK1ET = 3;
tSK2ST = 0; tSK2ET = 4;
tSK3ST = 1; tSK3ET = 3;
Nb = 1980;
Cm = 0.001;
t0 = 0; t1 = 1; t2 = 2; t3 = 3; t4 = 4;
End

  Global optimal solution found.
  Objective value:                              550864.8
  Objective bound:                              550864.8
  Infeasibilities:                              0.000000
  Extended solver steps:                               1
  Total solver iterations:                           238
```

Variable	Value	Reduced Cost
TOTAL_COST_FFR	550864.8	0.000000
FSR1A	180.0000	0.000000
FSR1B	180.0000	0.000000
FSR3B	300.0600	0.000000
FSR2C	72.00000	0.000000
FSR3C	300.0600	0.000000
FSR2D	72.00000	0.000000
FSK2A	36.00000	0.000000

FSK1B	180.0000	0.000000
FSK2B	36.00000	0.000000
FSK3B	300.0600	0.000000
FSK1C	180.0000	0.000000
FSK2C	36.00000	0.000000
FSK3C	300.0600	0.000000
FSK2D	36.00000	0.000000
ZSR1A	1.000000	0.000000
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	-0.3240000
ZSR1D	1.000000	-0.3240000
ZSR2A	0.000000	-38880.00
ZSR2B	0.000000	-38880.00
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	-40095.00
ZSR3A	0.000000	0.000000
ZSR3B	1.000000	0.000000
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	0.000000	-324000.0
XSR1D	0.000000	-324000.0
XSR2A	1.000000	-0.3888000E-01
XSR2B	1.000000	-0.3888000E-01
XSR2C	1.000000	0.000000
XSR2D	1.000000	-40095.00
XSR3A	1.000000	0.000000
XSR3B	1.000000	0.000000
XSR3C	1.000000	0.000000
XSR3D	0.000000	0.000000
YSR1A	1.000000	-356399.9
YSR1B	1.000000	-648000.6
YSR1C	0.000000	0.000000
YSR1D	0.000000	0.000000
YSR2A	0.000000	0.000000
YSR2B	0.000000	0.000000
YSR2C	1.000000	-77760.08
YSR2D	1.000000	0.000000
YSR3A	0.000000	0.000000
YSR3B	1.000000	0.000000
YSR3C	1.000000	0.000000
YSR3D	0.000000	0.000000
ZSK1A	0.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	1.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	1.000000	0.000000
ZSK3C	1.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000

XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	1.000000	0.000000
XSK2D	1.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	0.000000	0.000000
YSK1A	0.000000	0.000000
YSK1B	1.000000	469768.2
YSK1C	1.000000	340200.0
YSK1D	0.000000	0.000000
YSK2A	1.000000	100439.9
YSK2B	1.000000	100440.0
YSK2C	1.000000	68040.00
YSK2D	1.000000	69255.00
YSK3A	0.000000	0.000000
YSK3B	1.000000	324064.8
YSK3C	1.000000	200912.5
YSK3D	0.000000	0.000000
FSR1_SK2A	18.00000	0.000000
FSR1_S1A	162.0000	0.000000
FSR1_SK1B	71.98200	0.000000
FSR1_SK2B	18.00000	0.000000
FSR1_SK3B	90.01800	0.000000
FSR3_SK3B	210.0420	0.000000
FSR3_S3B	90.01800	0.000000
FSR2_SK3C	72.00000	0.000000
FSR3_SK3C	109.2420	0.000000
FSR3_WASTEC	190.8180	0.000000
FSR2_S2D	72.00000	0.000000
FFR_SK2A	18.00000	0.000000
FFR_SK1B	90.00000	0.000000
FS1_SK1B	18.01800	0.000000
FFR_SK2B	18.00000	0.000000
FFR_SK1C	116.2145	0.000000
FS1_SK1C	61.16400	0.000000
FS3_SK1C	2.621455	0.000000
FFR_SK2C	18.00000	0.000000
FS1_SK2C	18.00000	0.000000
FS1_SK3C	46.81800	0.000000
FS2_SK3C	72.00000	0.000000
FFR_SK2D	18.00000	0.000000
FS1_SK2D	18.00000	0.6750000E-04
TOTAL_FFR	278.2145	0.000000
TOTAL_WASTE	278.2145	0.000000
FS3_WASTEC	87.39655	0.000000
CSTS1B	143.9820	0.000000
CSTS2A	72.00000	0.000000
CSTS2B	72.00000	0.000000
CSTS1C	18.00000	0.000000

Note: The flow terms which are not shown are equal to zero.

Part 2 Minimizing the total operating cost

```
!Total_Cost_O = TOC;
min = Total_Cost_O;

!Equation 4.3;
FSR1A  =  FSR1*tA/(tSR1ET-tSR1ST);  FSR2A  =  FSR2*tA/(tSR2ET-tSR2ST);
FSR3A = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B  =  FSR1*tB/(tSR1ET-tSR1ST);  FSR2B  =  FSR2*tB/(tSR2ET-tSR2ST);
FSR3B = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C  =  FSR1*tC/(tSR1ET-tSR1ST);  FSR2C  =  FSR2*tC/(tSR2ET-tSR2ST);
FSR3C = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D  =  FSR1*tD/(tSR1ET-tSR1ST);  FSR2D  =  FSR2*tD/(tSR2ET-tSR2ST);
FSR3D = FSR3*tD/(tSR3ET-tSR3ST);

!Equation 4.4;
FSK1A  =  FSK1*tA/(tSK1ET-tSK1ST);  FSK2A  =  FSK2*tA/(tSK2ET-tSK2ST);
FSK3A = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B  =  FSK1*tB/(tSK1ET-tSK1ST);  FSK2B  =  FSK2*tB/(tSK2ET-tSK2ST);
FSK3B = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C  =  FSK1*tC/(tSK1ET-tSK1ST);  FSK2C  =  FSK2*tC/(tSK2ET-tSK2ST);
FSK3C = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D  =  FSK1*tD/(tSK1ET-tSK1ST);  FSK2D  =  FSK2*tD/(tSK2ET-tSK2ST);
FSK3D = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time interval 1(0-1hr), B=Time interval 2(1-2hr), C=Time interval
3(2-3hr)& D=Time interval 4(3-4hr);
tA = (t1 - t0); tB = (t2 - t1); tC = (t3 - t2); tD = (t4 - t3);

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <= 1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <= 1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <= 1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <= 1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <= 1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <= 1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <= 1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <= 1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <= 1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <= 1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <= 1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <= 1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <= 1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <= 1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <= 1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <= 1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <= 1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <= 1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <= 1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <= 1000*XSR2D;
1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <= 1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <= 1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <= 1000*XSR3C;
```

```
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <= 1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <= 1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <= 1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <= 1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <= 1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <= 1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <= 1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <= 1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <= 1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <= 1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <= 1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <= 1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <= 1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <= 1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <= 1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <= 1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <= 1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <= 1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <= 1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <= 1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <= 1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <= 1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <= 1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <= 1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <= 1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
```

```
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);
@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR_SK1A + FS1_SK1A + FS2_SK1A + FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR_SK2A + FS1_SK2A + FS2_SK2A + FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR_SK3A + FS1_SK3A + FS2_SK3A + FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR_SK1B + FS1_SK1B + FS2_SK1B + FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR_SK2B + FS1_SK2B + FS2_SK2B + FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR_SK3B + FS1_SK3B + FS2_SK3B + FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
```

```
               FFR_SK1C + FS1_SK1C + FS2_SK1C + FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR_SK2C + FS1_SK2C + FS2_SK2C + FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR_SK3C + FS1_SK3C + FS2_SK3C + FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR_SK1D + FS1_SK1D + FS2_SK1D + FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR_SK2D + FS1_SK2D + FS2_SK2D + FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR_SK3D + FS1_SK3D + FS2_SK3D + FS3_SK3D;

!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR_SK1A*OFFR +
                    FS1_SK1A*OSR1 + FS2_SK1A*OSR2 + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR_SK2A*OFFR +
                    FS1_SK2A*OSR1 + FS2_SK2A*OSR2 + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR_SK3A*OFFR +
                    FS1_SK3A*OSR1 + FS2_SK3A*OSR2 + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR_SK1B*OFFR +
                    FS1_SK1B*OSR1 + FS2_SK1B*OSR2 + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR_SK2B*OFFR +
                    FS1_SK2B*OSR1 + FS2_SK2B*OSR2 + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR_SK3B*OFFR +
                    FS1_SK3B*OSR1 + FS2_SK3B*OSR2 + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR_SK1C*OFFR +
                    FS1_SK1C*OSR1 + FS2_SK1C*OSR2 + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR_SK2C*OFFR +
                    FS1_SK2C*OSR1 + FS2_SK2C*OSR2 + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR_SK3C*OFFR +
                    FS1_SK3C*OSR1 + FS2_SK3C*OSR2 + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR_SK1D*OFFR +
                    FS1_SK1D*OSR1 + FS2_SK1D*OSR2 + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR_SK2D*OFFR +
                    FS1_SK2D*OSR1 + FS2_SK2D*OSR2 + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR_SK3D*OFFR +
                    FS1_SK3D*OSR1 + FS2_SK3D*OSR2 + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR = FFR_SK1A + FFR_SK2A + FFR_SK3A + FFR_SK1B + FFR_SK2B +
            FFR_SK3B + FFR_SK1C + FFR_SK2C + FFR_SK3C + FFR_SK1D +
            FFR_SK2D + FFR_SK3D;
```

```
!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FS1_WasteA +
            FS2_WasteA + FS3_WasteA + FSR1_WasteB + FSR2_WasteB +
            FSR3_WasteB + FS1_WasteB + FS2_WasteB + FS3_WasteB +
            FSR1_WasteC + FSR2_WasteC + FSR3_WasteC + FS1_WasteC +
            FS2_WasteC + FS3_WasteC + FSR1_WasteD + FSR2_WasteD +
            FSR3_WasteD + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + (95-80)*CP*(FSR1_SK2A + FSR1_WasteA + FS1_SK2A +
      FS1_WasteA + FS3_WasteA + FS3_SK2A)*(1000/3600);
H2A = H1A + (80-70)*CP*(FSR1_SK2A + FSR1_WasteA - FFR_SK2A +
      FS1_SK2A + FS1_WasteA + FS3_WasteA + FS3_SK2A)*(1000/3600);
H3A = H2A + (70-25)*CP*(FSR1_WasteA - FFR_SK2A + FS1_WasteA +
      FS2_WasteA + FS3_WasteA)*(1000/3600);
H3A = QCA;

H0B = QHB;
H1B = H0B + (105-95)*CP*(- FFR_SK1B - FFR_SK3B - FS2_SK1B -
      FS2_SK3B)*(1000/3600);
H2B = H1B + (95-80)*CP*(FSR1_SK2B + FSR1_WasteB + FSR3_SK2B +
      FSR3_WasteB - FFR_SK1B - FFR_SK3B + FS1_SK2B + FS1_WasteB
      - FS2_SK1B - FS2_SK3B + FS3_SK2B +FS3_WasteB)*(1000/3600);
H3B = H2B + (80-70)*CP*(FSR1_SK2B + FSR1_WasteB + FSR3_SK2B +
      FSR3_WasteB - FFR_SK1B - FFR_SK2B - FFR_SK3B + FS1_SK2B +
      FS1_WasteB + FS3_SK2B + FS3_WasteB)*(1000/3600);
H4B = H3B + (70-25)*CP*(FSR1_WasteB + FSR3_WasteB - FFR_SK1B -
      FFR_SK2B - FFR_SK3B + FS1_WasteB + FS2_WasteB +
      FS3_WasteB)*(1000/3600);
```

```
H4B = QCB;

H0C = QHC;
H1C = H0C + (105-95)*CP*(- FSR2_SK1C - FSR2_SK3C - FS2_SK1C-FFR_SK1C
      - FFR_SK3C - FS2_SK3C)*(1000/3600);
H2C = H1C + (95-80)*CP*(- FSR2_SK1C - FSR2_SK3C + FSR3_SK2C +
      FSR3_WasteC - FS2_SK1C - FFR_SK1C + FS1_WasteC + FS1_SK2C
      - FFR_SK3C - FS2_SK3C + FS3_SK2C +FS3_WasteC)*(1000/3600);
H3C = H2C + (80-70)*CP*(FSR3_SK2C + FSR3_WasteC -FFR_SK1C +
      FS1_WasteC - FFR_SK2C + FS1_SK2C - FFR_SK3C + FS3_SK2C +
      FS3_WasteC)*(1000/3600);
H4C = H3C + (70-25)*CP*(FSR2_WasteC + FSR3_WasteC - FFR_SK1C +
      FS1_WasteC - FFR_SK2C - FFR_SK3C + FS2_WasteC +
      FS3_WasteC)*(1000/3600);
H4C = QCC;

H0D = QHD;
H1D = H0D + (95-80)*CP*(FS1_SK2D + FS1_WasteD + FS3_SK2D +
      FS3_WasteD)*(1000/3600);
H2D = H1D + (80-70)*CP*(- FFR_SK2D + FS1_SK2D + FS1_WasteD +
      FS3_SK2D + FS3_WasteD)*(1000/3600);
H3D = H2D + (70-25)*CP*(FSR2_WasteD - FFR_SK2D + FS1_WasteD +
      FS2_WasteD + FS3_WasteD)*(1000/3600);
H3D = QCD;

!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0;
H0D>=0; H1D>=0; H2D>=0; H3D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 4.37;
!UC = value for unit conversion;
CS1 = (I/Ib)*(A0*(A1*UC*CSTS1*1000)^d)*AF;
CS2 = (I/Ib)*(A0*(A1*UC*CSTS2*1000)^d)*AF;
CS3 = (I/Ib)*(A0*(A1*UC*CSTS3*1000)^d)*AF;

!Equation 4.39;
CSTS1>= FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA + CSTS1A;
CSTS1>= FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB + CSTS1B;
CSTS1>= FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC + CSTS1C;
CSTS1>= FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD + CSTS1D;

CSTS2>= FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA + CSTS2A;
CSTS2>= FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB + CSTS2B;
CSTS2>= FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC + CSTS2C;
CSTS2>= FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD + CSTS2D;

CSTS3>= FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA + CSTS3A;
CSTS3>= FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB + CSTS3B;
CSTS3>= FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC + CSTS3C;
CSTS3>= FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD + CSTS3D;
```

```
!Equation 4.40;
CSTS1>=0; CSTS2>=0; CSTS3>=0;

!Equation 4.38;
AF = 0.229;

!Unit conversion;
!1 meter cube = 264.1721 gallon;
UC = 0.2642;

!Additional constraint (Equation 5.4);
Total_FFR = 278.2146;

!Equation 4.42;
Total_Cost_O = Total_QH*CHU*Nb + Total_QC*CCU*Nb;

!Data;
FSR1 = 360;  FSR2 = 144; FSR3 = 600.12;
FSK1 = 360;  FSK2 = 144; FSK3 = 600.12;
OSK1 = 50; OSK2 = 50; OSK3 = 800; OFFR = 0;
OSR1 = 100; OSR2 = 800;  OSR3 = 1100;
tSR1ST = 0; tSR1ET = 2;
tSR2ST = 2; tSR2ET = 4;
tSR3ST = 1; tSR3ET = 3;
tSK1ST = 1; tSK1ET = 3;
tSK2ST = 0; tSK2ET = 4;
tSK3ST = 1; tSK3ET = 3;
Nb = 1980;
Cm = 0.001;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
I = 572.7;
Ib = 394;
A0 = 210;
A1 = 1.1;
d = 0.51;
t0 = 0; t1 = 1; t2 = 2; t3 = 3; t4 = 4;
End
```

```
  Global optimal solution found.
  Objective value:                              172576.8
  Objective bound:                              172576.8
  Infeasibilities:                              0.000000
  Extended solver steps:                               1
  Total solver iterations:                          2284
```

Variable	Value	Reduced Cost
TOTAL_COST_O	172576.8	0.000000
FSR1A	180.0000	0.000000
FSR1B	180.0000	0.000000
FSR3B	300.0600	0.000000
FSR2C	72.00000	0.000000
FSR3C	300.0600	0.000000
FSR2D	72.00000	0.000000
FSK2A	36.00000	0.000000
FSK1B	180.0000	0.000000
FSK2B	36.00000	0.000000

FSK3B	300.0600	0.000000
FSK1C	180.0000	0.000000
FSK2C	36.00000	0.000000
FSK3C	300.0600	0.000000
FSK2D	36.00000	0.000000
ZSR1A	1.000000	-655602.6
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	0.000000	0.000000
ZSR2B	0.000000	0.000000
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	0.000000
ZSR3A	0.000000	0.000000
ZSR3B	1.000000	0.000000
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	-655602.6
XSR1B	1.000000	0.000000
XSR1C	0.000000	0.000000
XSR1D	0.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	1.000000	0.000000
XSR2D	1.000000	0.000000
XSR3A	1.000000	0.000000
XSR3B	1.000000	0.000000
XSR3C	1.000000	0.000000
XSR3D	0.000000	0.000000
YSR1A	1.000000	0.000000
YSR1B	1.000000	-1174155.
YSR1C	0.000000	-594990.0
YSR1D	0.000000	-594990.0
YSR2A	0.000000	0.000000
YSR2B	0.000000	0.000000
YSR2C	1.000000	8019.000
YSR2D	1.000000	8019.000
YSR3A	0.000000	0.000000
YSR3B	1.000000	83980.41
YSR3C	1.000000	75417.67
YSR3D	0.000000	0.000000
ZSK1A	0.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	1.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	1.000000	0.000000
ZSK3C	1.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000

XSK2C	1.000000	0.000000
XSK2D	1.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	0.000000	0.000000
YSK1A	0.000000	1559.250
YSK1B	1.000000	978440.8
YSK1C	1.000000	680946.4
YSK1D	0.000000	363700.9
YSK2A	1.000000	190401.6
YSK2B	1.000000	130375.3
YSK2C	1.000000	110409.6
YSK2D	1.000000	130902.7
YSK3A	0.000000	-36389.18
YSK3B	1.000000	493595.8
YSK3C	1.000000	237732.4
YSK3D	0.000000	-33419.18
FSR1_SK2A	1.072040	0.000000
FSR1_S1A	178.9280	0.000000
FSR1_SK3B	75.97437	0.000000
FSR1_S1B	104.0256	0.000000
FSR3_SK3B	184.6755	0.000000
FSR3_WASTEB	115.3845	0.000000
FSR2_SK2C	1.127661	0.000000
FSR2_SK3C	70.87234	0.9817500E-03
FSR3_SK3C	135.2484	0.000000
FSR3_WASTEC	145.4196	0.000000
FSR3_S3C	19.39202	0.000000
FSR2_S2D	71.99980	0.000000
FFR_SK2A	18.00000	0.000000
FS1_SK2A	16.92796	0.3305500E-02
FFR_SK1B	90.00000	0.000000
FS1_SK1B	90.00000	0.3305500E-02
FFR_SK2B	28.46460	0.000000
FS1_SK2B	6.040457	0.3305500E-02
FS2_SK2B	1.494943	0.000000
FS1_SK3B	4.021590	0.3305500E-02
FS2_SK3B	33.40679	0.000000
FS3_SK3B	1.981790	0.000000
FFR_SK1C	90.00000	0.000000
FS1_SK1C	90.00000	0.000000
FFR_SK2C	33.75000	0.000000
FS2_SK2C	1.122339	0.000000
FS1_SK3C	57.96358	0.000000
FS2_SK3C	35.97573	0.000000
FFR_SK2D	18.00000	0.000000
FS1_SK2D	18.00000	0.000000
TOTAL_FFR	278.2146	0.000000
TOTAL_WASTE	278.2146	0.000000
FS3_WASTEA	7.714286	0.000000
FS3_WASTEB	1.981790	0.000000
FS3_WASTED	7.714157	0.000000
CSTS1B	78.86591	0.000000
CSTS2A	71.99980	0.000000
CSTS2B	37.09806	0.000000
CSTS3A	3.963581	0.000000
CSTS1C	34.92796	0.000000
CSTS1D	16.92796	0.000000

CSTS3D	11.67787	0.000000
QHA	0.000000	0.000000
H1A	450.0000	0.000000
H2A	540.0000	0.000000
QCA	0.000000	0.000000
H0B	1439.746	0.000000
QHB	1439.746	0.000000
H3B	57.65891	0.000000
QCB	0.000000	0.000000
H0C	3196.559	0.000000
QHC	3196.559	0.000000
H1C	899.9978	0.000000
H3C	252.8123	0.000000
H4C	1390.467	0.000000
QCC	1390.467	0.000000
QHD	0.000000	0.000000
H1D	449.9978	0.000000
H2D	539.9963	0.000000
QCD	0.000000	0.000000
TOTAL_QH	4636.304	0.000000
TOTAL_QC	1390.467	0.000000
CS1	18156.18	0.000000
CSTS1	186.6655	0.000000
CS2	11169.16	0.000000
CSTS2	71.99980	0.000000
CS3	5720.968	0.000000
CSTS3	19.39202	0.000000

Note: The flow and heat terms which are not shown are equal to zero.

Appendix C - Matching formulation code in LINGO and matching formulation solution from LINGO for Case Study 3

C.1 Simultaneous approach

C.1.1 without storage system (Scenario 1)

```
!Total_Cost_O = TOC;
min = Total_Cost_O;

!Equation 4.3;
FSR1A = FSR1*tA/(tSR1ET-tSR1ST); FSR2A = FSR2*tA/(tSR2ET-tSR2ST);
FSR3A = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B = FSR1*tB/(tSR1ET-tSR1ST); FSR2B = FSR2*tB/(tSR2ET-tSR2ST);
FSR3B = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C = FSR1*tC/(tSR1ET-tSR1ST); FSR2C = FSR2*tC/(tSR2ET-tSR2ST);
FSR3C = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D = FSR1*tD/(tSR1ET-tSR1ST); FSR2D = FSR2*tD/(tSR2ET-tSR2ST);
FSR3D = FSR3*tD/(tSR3ET-tSR3ST);

!Equation 4.4;
FSK1A = FSK1*tA/(tSK1ET-tSK1ST); FSK2A = FSK2*tA/(tSK2ET-tSK2ST);
FSK3A = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B = FSK1*tB/(tSK1ET-tSK1ST); FSK2B = FSK2*tB/(tSK2ET-tSK2ST);
FSK3B = FSK3*tB/(tSK3ET-tSK3ST);
```

```
FSK1C = FSK1*tC/(tSK1ET-tSK1ST); FSK2C = FSK2*tC/(tSK2ET-tSK2ST);
FSK3C = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D = FSK1*tD/(tSK1ET-tSK1ST); FSK2D = FSK2*tD/(tSK2ET-tSK2ST);
FSK3D = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time interval 1(0-1hr), B=Time interval 2(1-2hr), C=Time interval
3(2-3hr)& D=Time interval 4(3-4hr);
tA = (t1 - t0); tB = (t2 - t1); tC = (t3 - t2); tD = (t4 - t3);

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <=
1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <=
1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <=
1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <=
1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <=
1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <=
1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <=
1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <=
1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <=
1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <=
1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <=
1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <=
1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <=
1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <=
1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <=
1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <=
1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <=
1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <=
1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <=
1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <=
1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <=
1000*XSR3A;
```

```
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <=
1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <=
1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <=
1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <=
1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <=
1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <=
1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <=
1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <=
1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <=
1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <=
1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <=
1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <=
1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <=
1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <=
1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <=
1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <=
1000*XSK1A;
```

```
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <=
1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <=
1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <=
1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <=
1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <=
1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <=
1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <=
1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <=
1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <=
1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <=
1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <=
1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;
```

```
FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR1_SK1A + FFR2_SK1A + FS1_SK1A + FS2_SK1A +
              FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR1_SK2A + FFR2_SK2A + FS1_SK2A + FS2_SK2A +
              FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR1_SK3A + FFR2_SK3A + FS1_SK3A + FS2_SK3A +
              FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR1_SK1B + FFR2_SK1B + FS1_SK1B + FS2_SK1B +
              FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR1_SK2B + FFR2_SK2B + FS1_SK2B + FS2_SK2B +
              FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR1_SK3B + FFR2_SK3B + FS1_SK3B + FS2_SK3B +
              FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR1_SK1C + FFR2_SK1C + FS1_SK1C + FS2_SK1C +
              FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR1_SK2C + FFR2_SK2C + FS1_SK2C + FS2_SK2C +
              FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR1_SK3C + FFR2_SK3C + FS1_SK3C + FS2_SK3C +
              FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR1_SK1D + FFR2_SK1D + FS1_SK1D + FS2_SK1D +
              FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR1_SK2D + FFR2_SK2D + FS1_SK2D + FS2_SK2D +
              FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR1_SK3D + FFR2_SK3D + FS1_SK3D + FS2_SK3D +
              FS3_SK3D;

!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR1_SK1A*OFFR1 +
```

```
                    FFR2_SK1A*OFFR2 + FS1_SK1A*OSR1 + FS2_SK1A*OSR2
                    + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR1_SK2A*OFFR1 +
                    FFR2_SK2A*OFFR2 + FS1_SK2A*OSR1 + FS2_SK2A*OSR2
                    + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR1_SK3A*OFFR1 +
                    FFR2_SK3A*OFFR2 + FS1_SK3A*OSR1 + FS2_SK3A*OSR2
                    + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR1_SK1B*OFFR1 +
                    FFR2_SK1B*OFFR2 + FS1_SK1B*OSR1 + FS2_SK1B*OSR2
                    + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR1_SK2B*OFFR1 +
                    FFR2_SK2B*OFFR2 + FS1_SK2B*OSR1 + FS2_SK2B*OSR2
                    + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR1_SK3B*OFFR1 +
                    FFR2_SK3B*OFFR2 + FS1_SK3B*OSR1 + FS2_SK3B*OSR2
                    + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR1_SK1C*OFFR1 +
                    FFR2_SK1C*OFFR2 + FS1_SK1C*OSR1 + FS2_SK1C*OSR2
                    + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR1_SK2C*OFFR1 +
                    FFR2_SK2C*OFFR2 + FS1_SK2C*OSR1 + FS2_SK2C*OSR2
                    + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR1_SK3C*OFFR1 +
                    FFR2_SK3C*OFFR2 + FS1_SK3C*OSR1 + FS2_SK3C*OSR2
                    + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR1_SK1D*OFFR1 +
                    FFR2_SK1D*OFFR2 + FS1_SK1D*OSR1 + FS2_SK1D*OSR2
                    + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR1_SK2D*OFFR1 +
                    FFR2_SK2D*OFFR2 + FS1_SK2D*OSR1 + FS2_SK2D*OSR2
                    + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR1_SK3D*OFFR1 +
                    FFR2_SK3D*OFFR2 + FS1_SK3D*OSR1 + FS2_SK3D*OSR2
                    + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR1 = FFR1_SK1A + FFR1_SK2A + FFR1_SK3A + FFR1_SK1B +
             FFR1_SK2B + FFR1_SK3B + FFR1_SK1C + FFR1_SK2C +
             FFR1_SK3C + FFR1_SK1D + FFR1_SK2D + FFR1_SK3D;

Total_FFR2 = FFR2_SK1A + FFR2_SK2A + FFR2_SK3A + FFR2_SK1B +
             FFR2_SK2B + FFR2_SK3B + FFR2_SK1C + FFR2_SK2C +
             FFR2_SK3C + FFR2_SK1D + FFR2_SK2D + FFR2_SK3D;
```

```
!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FS1_WasteA +
              FS2_WasteA + FS3_WasteA + FSR1_WasteB + FSR2_WasteB +
              FSR3_WasteB + FS1_WasteB + FS2_WasteB + FS3_WasteB +
              FSR1_WasteC + FSR2_WasteC + FSR3_WasteC + FS1_WasteC +
              FS2_WasteC + FS3_WasteC +FSR1_WasteD + FSR2_WasteD +
              FSR3_WasteD + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.26, 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + (70-55)*CP*(FSR1_SK2A+FSR1_WasteA)/3600;
H2A = H1A + (55-45)*CP*(FSR1_SK2A+FSR1_WasteA-FSR3_SK2A-FFR1_SK2A-
     FFR2_SK2A)/3600;
H3A = H2A + (45-40)*CP*(FSR1_WasteA-FFR1_SK2A-FFR2_SK2A)/3600;
H4A = H3A + (40-35)*CP*(FSR1_WasteA-FFR1_SK2A)/3600;
H5A = H4A + (35-30)*CP*(FSR1_WasteA+FSR3_WasteA-FFR1_SK2A)/3600;
H6A = H5A + (30-25)*CP*(FSR1_WasteA+FSR3_WasteA)/3600;
H6A = QCA;

H0B = QHB;
H1B = H0B + (90-80)*CP*(-FSR1_SK1B-FSR2_SK1B-FSR3_SK1B-FFR1_SK1B-
      FFR2_SK1B)/3600;
H2B = H1B + (80-70)*CP*(-FSR2_SK1B-FSR3_SK1B-FFR1_SK1B-
      FFR2_SK1B)/3600;
H3B = H2B + (70-60)*CP*(FSR1_SK2B+FSR1_SK3B+FSR1_WasteB-FSR3_SK1B-
      FSR3_SK3B-FFR1_SK1B-FFR1_SK3B-FFR2_SK1B-FFR2_SK3B)/3600;
H4B = H3B + (60-55)*CP*(FSR1_SK2B+FSR1_WasteB+FSR2_SK2B+FSR2_WasteB-
      FSR3_SK1B-FSR3_SK3B-FFR1_SK1B-FFR1_SK3B-FFR2_SK1B-FFR2_SK3B)
      /3600;
H5B = H4B + (55-45)*CP*(FSR1_SK2B+FSR1_WasteB+FSR2_SK2B+FSR2_WasteB-
```

```
      FSR3_SK1B-FSR3_SK2B-FSR3_SK3B-FFR1_SK1B-FFR1_SK2B-FFR1_SK3B-
      FFR2_SK1B-FFR2_SK2B-FFR2_SK3B)/3600;
H6B = H5B + (45-40)*CP*(FSR1_WasteB+FSR2_WasteB-FFR1_SK1B-FFR1_SK2B-
      FFR1_SK3B-FFR2_SK1B-FFR2_SK2B-FFR2_SK3B)/3600;
H7B = H6B + (40-35)*CP*(FSR1_WasteB+FSR2_WasteB-FFR1_SK1B-FFR1_SK2B-
      FFR1_SK3B)/3600;
H8B = H7B + (35-30)*CP*(FSR1_WasteB+FSR2_WasteB+FSR3_WasteB-
      FFR1_SK1B-FFR1_SK2B-FFR1_SK3B)/3600;
H9B = H8B + (30-25)*CP*(FSR1_WasteB+FSR2_WasteB+FSR3_WasteB)/3600;
H9B = QCB;

H0C = QHC;
H1C = H0C + (90-80)*CP*(-FSR1_SK1C-FSR3_SK1C-FFR1_SK1C-
      FFR2_SK1C)/3600;
H2C = H1C + (80-70)*CP*(-FSR3_SK1C-FFR1_SK1C-FFR2_SK1C)/3600;
H3C = H2C + (70-60)*CP*(FSR1_SK3C+FSR1_WasteC-FSR3_SK1C-FSR3_SK3C-
      FFR1_SK1C-FFR1_SK3C-FFR2_SK1C-FFR2_SK3C)/3600;
H4C = H3C + (60-45)*CP*(FSR1_WasteC-FSR3_SK1C-FSR3_SK3C-FFR1_SK1C-
      FFR1_SK3C-FFR2_SK1C-FFR2_SK3C)/3600;
H5C = H4C + (45-40)*CP*(FSR1_WasteC-FFR1_SK1C-FFR1_SK3C-FFR2_SK1C-
      FFR2_SK3C)/3600;
H6C = H5C + (40-35)*CP*(FSR1_WasteC-FFR1_SK1C-FFR1_SK3C)/3600;
H7C = H6C + (35-30)*CP*(FSR1_WasteC+FSR3_WasteC-FFR1_SK1C-
      FFR1_SK3C)/3600;
H8C = H7C + (30-25)*CP*(FSR1_WasteC+FSR3_WasteC)/3600;
H8C = QCC;

H0D = QHD;
H1D = H0D + (70-45)*CP*(-FSR3_SK3D-FFR1_SK3D-FFR2_SK3D)/3600;
H2D = H1D + (45-40)*CP*(-FFR1_SK3D-FFR2_SK3D)/3600;
H3D = H2D + (40-35)*CP*(-FFR1_SK3D)/3600;
H4D = H3D + (35-30)*CP*(FSR3_WasteD-FFR1_SK3D)/3600;
H5D = H4D + (30-25)*CP*(FSR3_WasteD)/3600;
H5D = QCD;

!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0; H4A>=0; H5A>=0; H6A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0; H5B>=0; H6B>=0; H7B>=0;
H8B>=0; H9B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0; H5C>=0; H6C>=0; H7C>=0;
H8C>=0;
H0D>=0; H1D>=0; H2D>=0; H3D>=0; H4D>=0; H5D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 5.1;
Total_Cost_O = Cm1*Total_FFR1*Nb + Cm2*Total_FFR2*Nb +
               Total_QH*CHU*Nb + Total_QC*CCU*Nb;

!Equation 5.2;
FSR1_S1A =0; FSR1_S1B=0; FSR1_S1C=0; FSR1_S1D=0;
FSR2_S2A =0; FSR2_S2B=0; FSR2_S2C=0; FSR2_S2D=0;
FSR3_S3A =0; FSR3_S3B=0; FSR3_S3C=0; FSR3_S3D=0;
```

```
!Data;
FSR1 = 10983;  FSR2 = 1766; FSR3 = 5940;
FSK1 = 5436;  FSK2 = 3986; FSK3 = 3381;
OSK1 = 35; OSK2 = 25; OSK3 = 40; OFFR1 = 3; OFFR2 = 6;
OSR1 = 38; OSR2 = 25;  OSR3 = 7;
tSR1ST = 0; tSR1ET = 3;
tSR2ST = 1; tSR2ET = 2;
tSR3ST = 0; tSR3ET = 4;
tSK1ST = 1; tSK1ET = 3;
tSK2ST = 0; tSK2ET = 2;
tSK3ST = 1; tSK3ET = 4;
Nb = 1980;
Cm1 = 0.00132;
Cm2 = 0.00088;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
t0 = 0; t1 = 1; t2 = 2; t3 = 3; t4 = 4;
End
```

```
  Global optimal solution found.
  Objective value:                              6305.745
  Objective bound:                              6305.745
  Infeasibilities:                              0.000000
  Extended solver steps:                               1
  Total solver iterations:                           150
```

Variable	Value	Reduced Cost
TOTAL_COST_O	6305.745	0.000000
FSR1A	3661.000	0.000000
FSR2A	1766.000	0.000000
FSR3A	1485.000	0.000000
FSR1B	3661.000	0.000000
FSR2B	1766.000	0.000000
FSR3B	1485.000	0.000000
FSR1C	3661.000	0.000000
FSR2C	1766.000	0.000000
FSR3C	1485.000	0.000000
FSR1D	3661.000	0.000000
FSR2D	1766.000	0.000000
FSR3D	1485.000	0.000000
FSK1A	2718.000	0.000000
FSK2A	1993.000	0.000000
FSK3A	1127.000	0.000000
FSK1B	2718.000	0.000000
FSK2B	1993.000	0.000000
FSK3B	1127.000	0.000000
FSK1C	2718.000	0.000000
FSK2C	1993.000	0.000000
FSK3C	1127.000	0.000000
FSK1D	2718.000	0.000000
FSK2D	1993.000	0.000000
FSK3D	1127.000	0.000000
ZSR1A	1.000000	0.000000
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000

ZSR2A	0.000000	0.000000
ZSR2B	1.000000	0.000000
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	0.000000
ZSR3A	1.000000	0.000000
ZSR3B	1.000000	0.000000
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	1.000000	0.000000
XSR1D	0.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	0.000000	0.000000
XSR2D	0.000000	0.000000
XSR3A	1.000000	0.000000
XSR3B	1.000000	0.000000
XSR3C	1.000000	0.000000
XSR3D	1.000000	0.000000
YSR1A	1.000000	2604.149
YSR1B	1.000000	3419.079
YSR1C	1.000000	3397.173
YSR1D	0.000000	0.000000
YSR2A	0.000000	0.000000
YSR2B	1.000000	1346.222
YSR2C	0.000000	0.000000
YSR2D	0.000000	0.000000
YSR3A	1.000000	437.4981
YSR3B	1.000000	167.8520
YSR3C	1.000000	126.8323
YSR3D	1.000000	-744.4083
ZSK1A	0.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	1.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	1.000000	0.000000
ZSK3C	1.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	0.000000	0.000000
XSK2D	0.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	1.000000	0.000000
YSK1A	0.000000	0.000000
YSK1B	1.000000	-2055.731
YSK1C	1.000000	-2055.731

YSK1D	0.000000	0.000000
YSK2A	1.000000	-1390.061
YSK2B	1.000000	-1478.944
YSK2C	0.000000	0.000000
YSK2D	0.000000	0.000000
YSK3A	0.000000	0.000000
YSK3B	1.000000	-1249.619
YSK3C	1.000000	-1249.619
YSK3D	1.000000	794.0282
FSR1_SK2A	1157.226	0.000000
FSR1_WASTEA	2503.774	0.000000
FSR3_SK2A	835.7742	0.000000
FSR3_WASTEA	649.2258	0.000000
FSR1_SK1B	2454.968	0.000000
FSR1_SK2B	79.03226	0.000000
FSR1_SK3B	1127.000	0.000000
FSR2_SK2B	1766.000	0.000000
FSR3_SK1B	263.0323	0.000000
FSR3_SK2B	147.9677	0.000000
FSR3_WASTEB	1074.000	0.000000
FSR1_SK1C	2454.968	0.000000
FSR1_SK3C	1127.000	0.000000
FSR1_WASTEC	79.03226	0.000000
FSR3_SK1C	263.0323	0.000000
FSR3_WASTEC	1221.968	0.000000
FSR3_SK3D	1127.000	0.000000
FSR3_WASTED	358.0000	0.000000
TOTAL_FFR1	0.000000	2.475745
TOTAL_FFR2	0.000000	0.000000
TOTAL_WASTE	5886.000	0.000000
QHA	0.000000	0.000000
H1A	64.06750	0.000000
H2A	97.02847	0.000000
H3A	111.6338	0.000000
H4A	126.2392	0.000000
H5A	144.6317	0.000000
H6A	163.0242	0.000000
QCA	163.0242	0.000000
H0B	34.77871	0.000000
QHB	34.77871	0.000000
H1B	3.068710	0.000000
H3B	11.00167	0.000000
H4B	20.23000	0.000000
H5B	36.96038	0.000000
H6B	36.96038	0.000000
H7B	36.96038	0.000000
H8B	43.22538	0.000000
H9B	49.49038	0.000000
QCB	49.49038	0.000000
H0C	34.77871	0.000000
QHC	34.77871	0.000000
H1C	3.068710	0.000000
H3C	11.00167	0.000000
H4C	7.781667	0.000000
H5C	8.242688	0.000000
H6C	8.703710	0.000000
H7C	16.29288	0.000000
H8C	23.88204	0.000000
QCC	23.88204	0.000000

```
H0D        32.87083        0.000000
QHD        32.87083        0.000000
H4D        2.088333        0.000000
H5D        4.176667        0.000000
QCD        4.176667        0.000000
TOTAL_QH   102.4283        0.000000
TOTAL_QC   240.5733        0.000000
```

Note: The flow and heat terms which are not shown are equal to zero.

C.1.2 with storage system (Scenario 2)

```
!Total_Cost = TAC;
min = Total_Cost;

!Equation 4.3;
FSR1A = FSR1*tA/(tSR1ET-tSR1ST); FSR2A = FSR2*tA/(tSR2ET-tSR2ST);
FSR3A = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B = FSR1*tB/(tSR1ET-tSR1ST); FSR2B = FSR2*tB/(tSR2ET-tSR2ST);
FSR3B = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C = FSR1*tC/(tSR1ET-tSR1ST); FSR2C = FSR2*tC/(tSR2ET-tSR2ST);
FSR3C = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D = FSR1*tD/(tSR1ET-tSR1ST); FSR2D = FSR2*tD/(tSR2ET-tSR2ST);
FSR3D = FSR3*tD/(tSR3ET-tSR3ST);

!Equation 4.4;
FSK1A = FSK1*tA/(tSK1ET-tSK1ST); FSK2A = FSK2*tA/(tSK2ET-tSK2ST);
FSK3A = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B = FSK1*tB/(tSK1ET-tSK1ST); FSK2B = FSK2*tB/(tSK2ET-tSK2ST);
FSK3B = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C = FSK1*tC/(tSK1ET-tSK1ST); FSK2C = FSK2*tC/(tSK2ET-tSK2ST);
FSK3C = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D = FSK1*tD/(tSK1ET-tSK1ST); FSK2D = FSK2*tD/(tSK2ET-tSK2ST);
FSK3D = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time interval 1(0-1hr), B=Time interval 2(1-2hr), C=Time interval
3(2-3hr)& D=Time interval 4(3-4hr);
tA = (t1 - t0); tB = (t2 - t1); tC = (t3 - t2); tD = (t4 - t3);

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <=
1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <=
1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <=
1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <=
1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <=
1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <=
1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <=
1000*ZSR2C;
```

```
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <=
1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <=
1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <=
1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <=
1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <=
1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <=
1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <=
1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <=
1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <=
1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <=
1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <=
1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <=
1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <=
1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <=
1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <=
1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <=
1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <=
1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);
```

```
!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <=
1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <=
1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <=
1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <=
1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <=
1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <=
1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <=
1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <=
1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <=
1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <=
1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <=
1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <=
1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <=
1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <=
1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <=
1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <=
1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <=
1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <=
1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <=
1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <=
1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <=
1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <=
1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <=
1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <=
1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
```

```
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR1_SK1A + FFR2_SK1A + FS1_SK1A + FS2_SK1A +
              FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR1_SK2A + FFR2_SK2A + FS1_SK2A + FS2_SK2A +
              FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR1_SK3A + FFR2_SK3A + FS1_SK3A + FS2_SK3A +
              FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR1_SK1B + FFR2_SK1B + FS1_SK1B + FS2_SK1B +
```

```
               FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR1_SK2B + FFR2_SK2B + FS1_SK2B + FS2_SK2B +
              FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR1_SK3B + FFR2_SK3B + FS1_SK3B + FS2_SK3B +
              FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR1_SK1C + FFR2_SK1C + FS1_SK1C + FS2_SK1C +
              FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR1_SK2C + FFR2_SK2C + FS1_SK2C + FS2_SK2C +
              FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR1_SK3C + FFR2_SK3C + FS1_SK3C + FS2_SK3C +
              FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR1_SK1D + FFR2_SK1D + FS1_SK1D + FS2_SK1D +
              FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR1_SK2D + FFR2_SK2D + FS1_SK2D + FS2_SK2D +
              FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR1_SK3D + FFR2_SK3D + FS1_SK3D + FS2_SK3D +
              FS3_SK3D;

!Equation 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR1_SK1A*OFFR1 +
                    FFR2_SK1A*OFFR2 + FS1_SK1A*OSR1 + FS2_SK1A*OSR2
                    + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR1_SK2A*OFFR1 +
                    FFR2_SK2A*OFFR2 + FS1_SK2A*OSR1 + FS2_SK2A*OSR2
                    + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR1_SK3A*OFFR1 +
                    FFR2_SK3A*OFFR2 + FS1_SK3A*OSR1 + FS2_SK3A*OSR2
                    + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR1_SK1B*OFFR1 +
                    FFR2_SK1B*OFFR2 + FS1_SK1B*OSR1 + FS2_SK1B*OSR2
                    + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR1_SK2B*OFFR1 +
                    FFR2_SK2B*OFFR2 + FS1_SK2B*OSR1 + FS2_SK2B*OSR2
                    + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR1_SK3B*OFFR1 +
                    FFR2_SK3B*OFFR2 + FS1_SK3B*OSR1 + FS2_SK3B*OSR2
                    + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR1_SK1C*OFFR1 +
                    FFR2_SK1C*OFFR2 + FS1_SK1C*OSR1 + FS2_SK1C*OSR2
                    + FS3_SK1C*OSR3;
```

```
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR1_SK2C*OFFR1 +
                    FFR2_SK2C*OFFR2 + FS1_SK2C*OSR1 + FS2_SK2C*OSR2
                    + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR1_SK3C*OFFR1 +
                    FFR2_SK3C*OFFR2 + FS1_SK3C*OSR1 + FS2_SK3C*OSR2
                    + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR1_SK1D*OFFR1 +
                    FFR2_SK1D*OFFR2 + FS1_SK1D*OSR1 + FS2_SK1D*OSR2
                    + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR1_SK2D*OFFR1 +
                    FFR2_SK2D*OFFR2 + FS1_SK2D*OSR1 + FS2_SK2D*OSR2
                    + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR1_SK3D*OFFR1 +
                    FFR2_SK3D*OFFR2 + FS1_SK3D*OSR1 + FS2_SK3D*OSR2
                    + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR1 = FFR1_SK1A + FFR1_SK2A + FFR1_SK3A + FFR1_SK1B +
             FFR1_SK2B + FFR1_SK3B + FFR1_SK1C + FFR1_SK2C +
             FFR1_SK3C + FFR1_SK1D + FFR1_SK2D + FFR1_SK3D;

Total_FFR2 = FFR2_SK1A + FFR2_SK2A + FFR2_SK3A + FFR2_SK1B +
             FFR2_SK2B + FFR2_SK3B + FFR2_SK1C + FFR2_SK2C +
             FFR2_SK3C + FFR2_SK1D + FFR2_SK2D + FFR2_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FS1_WasteA +
              FS2_WasteA + FS3_WasteA + FSR1_WasteB + FSR2_WasteB +
              FSR3_WasteB + FS1_WasteB + FS2_WasteB + FS3_WasteB +
              FSR1_WasteC + FSR2_WasteC + FSR3_WasteC + FS1_WasteC +
              FS2_WasteC + FS3_WasteC +FSR1_WasteD + FSR2_WasteD +
              FSR3_WasteD + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
```

```
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + (70-60)*CP*(FSR1_SK2A+FSR1_WasteA+FS1_SK2A+FS1_WasteA)
      /3600;
H2A = H1A + (60-55)*CP*(FSR1_SK2A+FSR1_WasteA+FS1_SK2A+FS1_WasteA
      +FS2_SK2A+FS2_WasteA)/3600;
H3A = H2A + (55-45)*CP*(FSR1_SK2A+FSR1_WasteA-FSR3_SK2A-FFR1_SK2A-
      FFR2_SK2A+FS1_SK2A+FS1_WasteA+FS2_SK2A+FS2_WasteA-FS3_SK2A)
      /3600;
H4A = H3A + (45-40)*CP*(FSR1_WasteA-FFR1_SK2A-FFR2_SK2A+FS1_WasteA+
      FS2_WasteA)/3600;
H5A = H4A + (40-35)*CP*(FSR1_WasteA-FFR1_SK2A+FS1_WasteA+FS2_WasteA)
      /3600;
H6A = H5A + (35-30)*CP*(FSR1_WasteA+FSR3_WasteA-FFR1_SK2A+FS1_WasteA
      +FS2_WasteA+FS3_WasteA)/3600;
H7A = H6A + (30-25)*CP*(FSR1_WasteA+FSR3_WasteA+FS1_WasteA+
      FS2_WasteA+FS3_WasteA)/3600;
H7A = QCA;

H0B = QHB;
H1B = H0B + (90-80)*CP*(-FSR1_SK1B-FSR2_SK1B-FSR3_SK1B-FFR1_SK1B-
      FS3_SK1B-FS3_SK3B-FFR2_SK1B-FS1_SK1B-FS2_SK1B)/3600;
H2B = H1B + (80-70)*CP*(-FSR2_SK1B-FSR3_SK1B-FFR1_SK1B-FS3_SK1B-
      FS3_SK3B-FFR2_SK1B-FS2_SK1B)/3600;
H3B = H2B + (70-60)*CP*(FSR1_SK2B+FSR1_SK3B+FSR1_WasteB-FSR3_SK1B-
      FSR3_SK3B-FFR1_SK1B-FS3_SK1B-FFR1_SK3B-FS3_SK3B-FFR2_SK1B-
      FFR2_SK3B+FS1_SK2B+FS1_SK3B+FS1_WasteB)/3600;
H4B = H3B + (60-55)*CP*(FSR1_SK2B+FSR1_WasteB+FSR2_SK2B+FSR2_WasteB-
      FSR3_SK1B-FSR3_SK3B-FFR1_SK1B-FS3_SK1B-FFR1_SK3B-FS3_SK3B-
      FFR2_SK1B+FS2_SK2B-FFR2_SK3B+FS1_SK2B+FS1_WasteB+FS2_WasteB)
      /3600;
H5B = H4B + (55-45)*CP*(FSR1_SK2B-FS3_SK2B+FSR1_WasteB+FSR2_SK2B+
      FSR2_WasteB-FSR3_SK1B-FSR3_SK2B-FSR3_SK3B-FFR1_SK1B-FFR1_SK2B-
      FS3_SK1B-FFR1_SK3B-FS3_SK3B-FFR2_SK1B+FS2_SK2B-FFR2_SK2B-
      FFR2_SK3B+FS1_SK2B+FS1_WasteB+FS2_WasteB)/3600;
H6B = H5B + (45-40)*CP*(FSR1_WasteB+FSR2_WasteB-FFR1_SK1B-FFR1_SK2B-
      FFR1_SK3B-FFR2_SK1B-FFR2_SK2B-FFR2_SK3B+FS1_WasteB+FS2_WasteB)
      /3600;
H7B = H6B + (40-35)*CP*(FSR1_WasteB+FSR2_WasteB-FFR1_SK1B-FFR1_SK2B-
      FFR1_SK3B+FS1_WasteB+FS2_WasteB)/3600;
H8B = H7B + (35-30)*CP*(FSR1_WasteB+FSR2_WasteB+FSR3_WasteB+
      FS3_WasteB-FFR1_SK1B-FFR1_SK2B-FFR1_SK3B+FS1_WasteB+
      FS2_WasteB)/3600;
H9B = H8B + (30-25)*CP*(FSR1_WasteB+FSR2_WasteB+FSR3_WasteB+
      FS1_WasteB+FS2_WasteB+FS3_WasteB)/3600;
H9B = QCB;

H0C = QHC;
H1C = H0C + (90-80)*CP*(-FSR1_SK1C-FSR3_SK1C-FFR1_SK1C-FFR2_SK1C-
      FS1_SK1C-FS2_SK1C-FS3_SK1C)/3600;
```

```
H2C = H1C + (80-70)*CP*(-FSR3_SK1C-FFR1_SK1C-FFR2_SK1C-FS2_SK1C-
      FS3_SK1C)/3600;
H3C = H2C + (70-60)*CP*(FSR1_SK3C+FSR1_WasteC-FSR3_SK1C-FSR3_SK3C-
      FFR1_SK1C-FFR1_SK3C-FFR2_SK1C-FFR2_SK3C+FS1_SK3C+FS1_WasteC
      -FS3_SK1C-FS3_SK3C)/3600;
H4C = H3C + (60-45)*CP*(FSR1_WasteC-FSR3_SK1C-FSR3_SK3C-FFR1_SK1C-
      FFR1_SK3C-FFR2_SK1C-FFR2_SK3C+FS1_WasteC+FS2_WasteC-FS3_SK1C
      -FS3_SK3C)/3600;
H5C = H4C + (45-40)*CP*(FSR1_WasteC-FFR1_SK1C-FFR1_SK3C-FFR2_SK1C-
      FFR2_SK3C+FS1_WasteC+FS2_WasteC)/3600;
H6C = H5C + (40-35)*CP*(FSR1_WasteC-FFR1_SK1C-FFR1_SK3C+FS1_WasteC
      +FS2_WasteC)/3600;
H7C = H6C + (35-30)*CP*(FSR1_WasteC+FSR3_WasteC-FFR1_SK1C-FFR1_SK3C
      +FS1_WasteC+FS2_WasteC+FS3_WasteC)/3600;
H8C = H7C + (30-25)*CP*(FSR1_WasteC+FSR3_WasteC+FS1_WasteC+
      FS2_WasteC+FS3_WasteC)/3600;
H8C = QCC;

H0D = QHD;
H1D = H0D + (70-60)*CP*(-FSR3_SK3D-FFR1_SK3D-FFR2_SK3D+FS1_SK3D+
      FS1_WasteD-FS3_SK3D)/3600;
H2D = H1D + (60-45)*CP*(-FSR3_SK3D-FFR1_SK3D-FFR2_SK3D+FS1_WasteD
      +FS2_WasteD-FS3_SK3D)/3600;
H3D = H2D + (45-40)*CP*(-FFR1_SK3D-FFR2_SK3D+FS1_WasteD+FS2_WasteD)
      /3600;
H4D = H3D + (40-35)*CP*(-FFR1_SK3D+FS1_WasteD+FS2_WasteD)/3600;
H5D = H4D + (35-30)*CP*(FSR3_WasteD-FFR1_SK3D+FS1_WasteD+FS2_WasteD+
      FS3_WasteD)/3600;
H6D = H5D + (30-25)*CP*(FSR3_WasteD+FS1_WasteD+FS2_WasteD+
      FS3_WasteD)/3600;
H6D = QCD;

!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0; H4A>=0; H5A>=0; H6A>=0; H7A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0; H5B>=0; H6B>=0; H7B>=0;
H8B>=0; H9B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0; H5C>=0; H6C>=0; H7C>=0;
H8C>=0;
H0D>=0; H1D>=0; H2D>=0; H3D>=0; H4D>=0; H5D>=0; H6D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 4.36;
Total_Cost = Cm1*Total_FFR1*Nb + Cm2*Total_FFR2*Nb + Total_QH*CHU*Nb
+ Total_QC*CCU*Nb + CS1 + CS2 +CS3;

!Equation 4.37;
!UC = value for unit conversion;
CS1 = (I/Ib)*(A0*(A1*UC*CSTS1)^d)*AF;
CS2 = (I/Ib)*(A0*(A1*UC*CSTS2)^d)*AF;
CS3 = (I/Ib)*(A0*(A1*UC*CSTS3)^d)*AF;

!Equation 4.38;
AF = 0.229;
```

```
!Equation 4.39;
CSTS1>= FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA + CSTS1A;
CSTS1>= FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB + CSTS1B;
CSTS1>= FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC + CSTS1C;
CSTS1>= FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD + CSTS1D;

CSTS2>= FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA + CSTS2A;
CSTS2>= FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB + CSTS2B;
CSTS2>= FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC + CSTS2C;
CSTS2>= FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD + CSTS2D;

CSTS3>= FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA + CSTS3A;
CSTS3>= FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB + CSTS3B;
CSTS3>= FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC + CSTS3C;
CSTS3>= FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD + CSTS3D;

!Equation 4.40;
CSTS1>=0; CSTS2>=0; CSTS3>=0;

!Unit conversion;
!1 meter cube = 264.1721 gallon;
UC = 0.2642;

!Data;
FSR1 = 10983;  FSR2 = 1766; FSR3 = 5940;
FSK1 = 5436;  FSK2 = 3986; FSK3 = 3381;
OSK1 = 35; OSK2 = 25; OSK3 = 40; OFFR1 = 3; OFFR2 = 6;
OSR1 = 38; OSR2 = 25;  OSR3 = 7;
tSR1ST = 0; tSR1ET = 3;
tSR2ST = 1; tSR2ET = 2;
tSR3ST = 0; tSR3ET = 4;
tSK1ST = 1; tSK1ET = 3;
tSK2ST = 0; tSK2ET = 2;
tSK3ST = 1; tSK3ET = 4;
Nb = 1980;
Cm1 = 0.00132;
Cm2 = 0.00088;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
I = 572.7;
Ib = 394;
A0 = 210;
A1 = 1.1;
d = 0.51;
t0 = 0; t1 = 1; t2 = 2; t3 = 3; t4 = 4;
End
```

```
  Global optimal solution found.
  Objective value:                              5937.921
  Objective bound:                              5937.921
  Infeasibilities:                              0.000000
  Extended solver steps:                              73
  Total solver iterations:                         23173


                       Variable           Value        Reduced Cost
                     TOTAL_COST        5937.921            0.000000
```

FSR1A	3661.000	0.000000
FSR2A	1766.000	0.000000
FSR3A	1485.000	0.000000
FSR1B	3661.000	0.000000
FSR2B	1766.000	0.000000
FSR3B	1485.000	0.000000
FSR1C	3661.000	0.000000
FSR2C	1766.000	0.000000
FSR3C	1485.000	0.000000
FSR1D	3661.000	0.000000
FSR2D	1766.000	0.000000
FSR3D	1485.000	0.000000
FSK1A	2718.000	0.000000
FSK2A	1993.000	0.000000
FSK3A	1127.000	0.000000
FSK1B	2718.000	0.000000
FSK2B	1993.000	0.000000
FSK3B	1127.000	0.000000
FSK1C	2718.000	0.000000
FSK2C	1993.000	0.000000
FSK3C	1127.000	0.000000
FSK1D	2718.000	0.000000
FSK2D	1993.000	0.000000
FSK3D	1127.000	0.000000
ZSR1A	1.000000	0.000000
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	0.000000	0.000000
ZSR2B	1.000000	0.000000
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	0.000000
ZSR3A	1.000000	0.000000
ZSR3B	1.000000	0.000000
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	1.000000	0.000000
XSR1D	0.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	0.000000	0.000000
XSR2D	0.000000	0.000000
XSR3A	1.000000	0.000000
XSR3B	1.000000	0.000000
XSR3C	1.000000	0.000000
XSR3D	1.000000	0.000000
YSR1A	1.000000	2424.184
YSR1B	1.000000	2901.957
YSR1C	1.000000	3332.112
YSR1D	0.000000	0.000000
YSR2A	0.000000	0.000000
YSR2B	1.000000	1565.634
YSR2C	0.000000	0.000000
YSR2D	0.000000	0.000000
YSR3A	1.000000	617.4630
YSR3B	1.000000	465.5616
YSR3C	1.000000	191.8939

YSR3D	1.000000	197.4952
ZSK1A	0.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	1.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	1.000000	0.000000
ZSK3C	1.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	0.000000	0.000000
XSK2D	0.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	1.000000	0.000000
YSK1A	0.000000	0.000000
YSK1B	1.000000	-2055.731
YSK1C	1.000000	-2055.731
YSK1D	0.000000	0.000000
YSK2A	1.000000	-1075.123
YSK2B	1.000000	-1104.227
YSK2C	0.000000	-721.7620
YSK2D	0.000000	0.000000
YSK3A	0.000000	0.000000
YSK3B	1.000000	-1103.344
YSK3C	1.000000	-1184.557
YSK3D	1.000000	-15.48904
FSR1_SK2A	508.0000	0.000000
FSR1_WASTEA	2348.000	0.000000
FSR1_S1A	805.0000	0.000000
FSR3_SK2A	1485.000	0.000000
FSR1_SK1B	1778.673	0.000000
FSR1_SK2B	1.663330	0.000000
FSR1_SK3B	1.663330	0.000000
FSR1_WASTEB	1074.000	0.000000
FSR1_S1B	805.0000	0.000000
FSR2_SK2B	942.6360	0.000000
FSR2_SK3B	823.3640	0.000000
FSR3_SK1B	263.0323	0.000000
FSR3_SK2B	1048.701	0.000000
FSR3_SK3B	173.2671	0.000000
FSR1_SK1C	2109.968	0.000000
FSR1_WASTEC	746.0323	0.000000
FSR1_S1C	805.0000	0.000000
FSR3_SK1C	263.0323	0.000000
FSR3_SK3C	667.0000	0.000000
FSR3_WASTEC	554.9677	0.000000
FSR3_SK3D	322.0000	0.000000
FSR3_WASTED	1163.000	0.000000

FS1_SK1B	676.2944	0.2310002E-06
FS1_SK3B	128.7056	0.4851005E-06
FS1_SK1C	345.0000	0.2310002E-06
FS1_SK3C	460.0000	0.4851005E-06
FS1_SK3D	805.0000	0.000000
TOTAL_FFR1	0.000000	0.000000
TOTAL_FFR2	0.000000	0.000000
TOTAL_WASTE	5886.000	0.000000
QHA	0.000000	0.000000
H1A	33.32000	0.000000
H2A	49.98000	0.000000
H3A	65.97500	0.000000
H4A	79.67167	0.000000
H5A	93.36833	0.000000
H6A	107.0650	0.000000
H7A	120.7617	0.000000
QCA	120.7617	0.000000
H0B	34.77871	0.000000
QHB	34.77871	0.000000
H1B	3.068710	0.000000
H3B	8.980218	0.000000
H4B	18.20855	0.000000
H5B	24.43038	0.000000
H6B	30.69538	0.000000
H7B	36.96038	0.000000
H8B	43.22538	0.000000
H9B	49.49038	0.000000
QCB	49.49038	0.000000
H0C	34.77871	0.000000
QHC	34.77871	0.000000
H1C	3.068710	0.000000
H3C	3.220000	0.000000
H5C	4.351855	0.000000
H6C	8.703710	0.000000
H7C	16.29288	0.000000
H8C	23.88204	0.000000
QCC	23.88204	0.000000
QHD	0.000000	0.000000
H1D	5.635000	0.000000
H5D	6.784167	0.000000
H6D	13.56833	0.000000
QCD	13.56833	0.000000
TOTAL_QH	69.55742	0.000000
TOTAL_QC	207.7024	0.000000
CS1	1129.114	0.000000
CS2	0.000000	0.000000
CS3	0.000000	0.000000
CSTS1	805.0000	0.000000
CSTS2	0.000000	0.2066883E+09
CSTS3	0.000000	0.2066883E+09

Note: The flow and heat terms which are not shown are equal to zero.

C.2 Sequential approach with storage system (Scenario 3)

Part 1 Minimizing the cost for fresh resource

```
!Total_Cost_FR = Total cost for fresh resource;
min = Total_Cost_FR;

!Equation 4.3;
FSR1A = FSR1*tA/(tSR1ET-tSR1ST); FSR2A = FSR2*tA/(tSR2ET-tSR2ST);
FSR3A = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B = FSR1*tB/(tSR1ET-tSR1ST); FSR2B = FSR2*tB/(tSR2ET-tSR2ST);
FSR3B = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C = FSR1*tC/(tSR1ET-tSR1ST); FSR2C = FSR2*tC/(tSR2ET-tSR2ST);
FSR3C = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D = FSR1*tD/(tSR1ET-tSR1ST); FSR2D = FSR2*tD/(tSR2ET-tSR2ST);
FSR3D = FSR3*tD/(tSR3ET-tSR3ST);

!Equation 4.4;
FSK1A = FSK1*tA/(tSK1ET-tSK1ST); FSK2A = FSK2*tA/(tSK2ET-tSK2ST);
FSK3A = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B = FSK1*tB/(tSK1ET-tSK1ST); FSK2B = FSK2*tB/(tSK2ET-tSK2ST);
FSK3B = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C = FSK1*tC/(tSK1ET-tSK1ST); FSK2C = FSK2*tC/(tSK2ET-tSK2ST);
FSK3C = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D = FSK1*tD/(tSK1ET-tSK1ST); FSK2D = FSK2*tD/(tSK2ET-tSK2ST);
FSK3D = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time interval 1(0-1hr), B=Time interval 2(1-2hr), C=Time interval
3(2-3hr)& D=Time interval 4(3-4hr);
tA = (t1 - t0); tB = (t2 - t1); tC = (t3 - t2); tD = (t4 - t3);

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <=
1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <=
1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <=
1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <=
1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <=
1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <=
1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <=
1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <=
1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <=
1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <=
1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <=
1000*ZSR3C;
```

```
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <=
1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <=
1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <=
1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <=
1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <=
1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <=
1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <=
1000*XSR2B;
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <=
1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <=
1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <=
1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <=
1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <=
1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <=
1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <=
1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <=
1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <=
1000*ZSK1C;
```

```
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <=
1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <=
1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <=
1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <=
1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <=
1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <=
1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <=
1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <=
1000*ZSK3C;
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <=
1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <=
1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <=
1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <=
1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <=
1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <=
1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <=
1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <=
1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <=
1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <=
1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <=
1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <=
1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <=
1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
```

```
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;

FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR1_SK1A + FFR2_SK1A + FS1_SK1A + FS2_SK1A +
              FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR1_SK2A + FFR2_SK2A + FS1_SK2A + FS2_SK2A +
              FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR1_SK3A + FFR2_SK3A + FS1_SK3A + FS2_SK3A +
              FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR1_SK1B + FFR2_SK1B + FS1_SK1B + FS2_SK1B +
              FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR1_SK2B + FFR2_SK2B + FS1_SK2B + FS2_SK2B +
              FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR1_SK3B + FFR2_SK3B + FS1_SK3B + FS2_SK3B +
              FS3_SK3B;
```

```
FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR1_SK1C + FFR2_SK1C + FS1_SK1C + FS2_SK1C +
              FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR1_SK2C + FFR2_SK2C + FS1_SK2C + FS2_SK2C +
              FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR1_SK3C + FFR2_SK3C + FS1_SK3C + FS2_SK3C +
              FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR1_SK1D + FFR2_SK1D + FS1_SK1D + FS2_SK1D +
              FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR1_SK2D + FFR2_SK2D + FS1_SK2D + FS2_SK2D +
              FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
              FFR1_SK3D + FFR2_SK3D + FS1_SK3D + FS2_SK3D +
              FS3_SK3D;

!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR1_SK1A*OFFR1 +
                    FFR2_SK1A*OFFR2 + FS1_SK1A*OSR1 + FS2_SK1A*OSR2
                    + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR1_SK2A*OFFR1 +
                    FFR2_SK2A*OFFR2 + FS1_SK2A*OSR1 + FS2_SK2A*OSR2
                    + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR1_SK3A*OFFR1 +
                    FFR2_SK3A*OFFR2 + FS1_SK3A*OSR1 + FS2_SK3A*OSR2
                    + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR1_SK1B*OFFR1 +
                    FFR2_SK1B*OFFR2 + FS1_SK1B*OSR1 + FS2_SK1B*OSR2
                    + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR1_SK2B*OFFR1 +
                    FFR2_SK2B*OFFR2 + FS1_SK2B*OSR1 + FS2_SK2B*OSR2
                    + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR1_SK3B*OFFR1 +
                    FFR2_SK3B*OFFR2 + FS1_SK3B*OSR1 + FS2_SK3B*OSR2
                    + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR1_SK1C*OFFR1 +
                    FFR2_SK1C*OFFR2 + FS1_SK1C*OSR1 + FS2_SK1C*OSR2
                    + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR1_SK2C*OFFR1 +
                    FFR2_SK2C*OFFR2 + FS1_SK2C*OSR1 + FS2_SK2C*OSR2
                    + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR1_SK3C*OFFR1 +
                    FFR2_SK3C*OFFR2 + FS1_SK3C*OSR1 + FS2_SK3C*OSR2
```

```
                    + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR1_SK1D*OFFR1 +
                    FFR2_SK1D*OFFR2 + FS1_SK1D*OSR1 + FS2_SK1D*OSR2
                    + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR1_SK2D*OFFR1 +
                    FFR2_SK2D*OFFR2 + FS1_SK2D*OSR1 + FS2_SK2D*OSR2
                    + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR1_SK3D*OFFR1 +
                    FFR2_SK3D*OFFR2 + FS1_SK3D*OSR1 + FS2_SK3D*OSR2
                    + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR1 = FFR1_SK1A + FFR1_SK2A + FFR1_SK3A + FFR1_SK1B +
             FFR1_SK2B + FFR1_SK3B + FFR1_SK1C + FFR1_SK2C +
             FFR1_SK3C + FFR1_SK1D + FFR1_SK2D + FFR1_SK3D;

Total_FFR2 = FFR2_SK1A + FFR2_SK2A + FFR2_SK3A + FFR2_SK1B +
             FFR2_SK2B + FFR2_SK3B + FFR2_SK1C + FFR2_SK2C +
             FFR2_SK3C + FFR2_SK1D + FFR2_SK2D + FFR2_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FS1_WasteA +
              FS2_WasteA + FS3_WasteA + FSR1_WasteB + FSR2_WasteB +
              FSR3_WasteB + FS1_WasteB + FS2_WasteB + FS3_WasteB +
              FSR1_WasteC + FSR2_WasteC + FSR3_WasteC + FS1_WasteC +
              FS2_WasteC + FS3_WasteC +FSR1_WasteD + FSR2_WasteD +
              FSR3_WasteD + FS1_WasteD + FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
```

```
                    CSTS3A;

!Equation 4.41;
Total_Cost_FFR = Cm1*Total_FFR1*Nb + Cm2*Total_FFR2*Nb;

!Data;
FSR1 = 10983;  FSR2 = 1766; FSR3 = 5940;
FSK1 = 5436;  FSK2 = 3986; FSK3 = 3381;
OSK1 = 35; OSK2 = 25; OSK3 = 40; OFFR1 = 3; OFFR2 = 6;
OSR1 = 38; OSR2 = 25;  OSR3 = 7;
tSR1ST = 0; tSR1ET = 3;
tSR2ST = 1; tSR2ET = 2;
tSR3ST = 0; tSR3ET = 4;
tSK1ST = 1; tSK1ET = 3;
tSK2ST = 0; tSK2ET = 2;
tSK3ST = 1; tSK3ET = 4;
Nb = 1980;
Cm1 = 0.00132;
Cm2 = 0.00088;
t0 = 0; t1 = 1; t2 = 2; t3 = 3; t4 = 4;
End
```

```
  Global optimal solution found.
  Objective value:                              0.000000
  Objective bound:                              0.000000
  Infeasibilities:                              0.000000
  Extended solver steps:                               0
  Total solver iterations:                            74
```

Variable	Value	Reduced Cost
TOTAL_COST_FR	0.000000	0.000000
FSR1A	3661.000	0.000000
FSR2A	1766.000	0.000000
FSR3A	1485.000	0.000000
FSR1B	3661.000	0.000000
FSR2B	1766.000	0.000000
FSR3B	1485.000	0.000000
FSR1C	3661.000	0.000000
FSR2C	1766.000	0.000000
FSR3C	1485.000	0.000000
FSR1D	3661.000	0.000000
FSR2D	1766.000	0.000000
FSR3D	1485.000	0.000000
FSK1A	2718.000	0.000000
FSK2A	1993.000	0.000000
FSK3A	1127.000	0.000000
FSK1B	2718.000	0.000000
FSK2B	1993.000	0.000000
FSK3B	1127.000	0.000000
FSK1C	2718.000	0.000000
FSK2C	1993.000	0.000000
FSK3C	1127.000	0.000000
FSK1D	2718.000	0.000000
FSK2D	1993.000	0.000000
FSK3D	1127.000	0.000000
ZSR1A	1.000000	0.000000
ZSR1B	1.000000	0.000000

ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	0.000000	0.000000
ZSR2B	1.000000	0.000000
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	0.000000
ZSR3A	1.000000	0.000000
ZSR3B	1.000000	0.000000
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	1.000000	0.000000
XSR1D	0.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	0.000000	0.000000
XSR2D	0.000000	0.000000
XSR3A	1.000000	0.000000
XSR3B	1.000000	0.000000
XSR3C	1.000000	0.000000
XSR3D	1.000000	0.000000
YSR1A	1.000000	0.000000
YSR1B	1.000000	0.000000
YSR1C	1.000000	0.000000
YSR1D	0.000000	0.000000
YSR2A	0.000000	0.000000
YSR2B	1.000000	0.000000
YSR2C	0.000000	0.000000
YSR2D	0.000000	0.000000
YSR3A	1.000000	0.000000
YSR3B	1.000000	0.000000
YSR3C	1.000000	0.000000
YSR3D	1.000000	0.000000
ZSK1A	0.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	1.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	1.000000	0.000000
ZSK3C	1.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	0.000000	0.000000
XSK2D	0.000000	0.000000
XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	1.000000	0.000000
YSK1A	0.000000	0.000000

```
          YSK1B        1.000000             0.000000
          YSK1C        1.000000             0.000000
          YSK1D        0.000000             0.000000
          YSK2A        1.000000             0.000000
          YSK2B        1.000000             0.000000
          YSK2C        0.000000             0.000000
          YSK2D        0.000000             0.000000
          YSK3A        0.000000             0.000000
          YSK3B        1.000000             0.000000
          YSK3C        1.000000             0.000000
          YSK3D        1.000000             0.000000
      FSR1_SK2A        1157.226             0.000000
    FSR1_WASTEA        2503.774             0.000000
      FSR3_SK2A        477.7742             0.000000
       FSR3_S3A        1007.226             0.000000
      FSR1_SK1B        2454.968             0.000000
      FSR1_SK2B        1157.226             0.000000
       FSR1_S1B        48.80645             0.000000
       FSR2_S2B        1766.000             0.000000
      FSR3_SK1B        263.0323             0.000000
      FSR3_SK2B        835.7742             0.000000
      FSR3_SK3B        119.7742             0.000000
       FSR3_S3B        266.4194             0.000000
      FSR1_SK1C        327.5806             0.000000
    FSR1_WASTEC        3333.419             0.000000
      FSR3_SK1C        358.0000             0.000000
      FSR3_SK3C        1127.000             0.000000
      FSR3_SK3D        1127.000             0.000000
       FSR3_S3D        358.0000             0.000000
       FS3_SK2A        358.0000             0.000000
       FS3_SK3B        1007.226             0.000000
       FS2_SK1C        1766.000             0.000000
       FS3_SK1C        266.4194             0.000000
     TOTAL_FFR1        0.000000             0.000000
     TOTAL_FFR2        0.000000             0.000000
    TOTAL_WASTE        5886.000             0.000000
     FS1_WASTEC        48.80645             0.000000
```

Note: The flow and heat terms which are not shown are equal to zero.

Part 2 Minimizing the total operating cost

```
!Total_Cost_O = TOC;
min = Total_Cost_O;

!Equation 4.3;
FSR1A = FSR1*tA/(tSR1ET-tSR1ST); FSR2A = FSR2*tA/(tSR2ET-tSR2ST);
FSR3A = FSR3*tA/(tSR3ET-tSR3ST);
FSR1B = FSR1*tB/(tSR1ET-tSR1ST); FSR2B = FSR2*tB/(tSR2ET-tSR2ST);
FSR3B = FSR3*tB/(tSR3ET-tSR3ST);
FSR1C = FSR1*tC/(tSR1ET-tSR1ST); FSR2C = FSR2*tC/(tSR2ET-tSR2ST);
FSR3C = FSR3*tC/(tSR3ET-tSR3ST);
FSR1D = FSR1*tD/(tSR1ET-tSR1ST); FSR2D = FSR2*tD/(tSR2ET-tSR2ST);
FSR3D = FSR3*tD/(tSR3ET-tSR3ST);
```

```
!Equation 4.4;
FSK1A = FSK1*tA/(tSK1ET-tSK1ST); FSK2A = FSK2*tA/(tSK2ET-tSK2ST);
FSK3A = FSK3*tA/(tSK3ET-tSK3ST);
FSK1B = FSK1*tB/(tSK1ET-tSK1ST); FSK2B = FSK2*tB/(tSK2ET-tSK2ST);
FSK3B = FSK3*tB/(tSK3ET-tSK3ST);
FSK1C = FSK1*tC/(tSK1ET-tSK1ST); FSK2C = FSK2*tC/(tSK2ET-tSK2ST);
FSK3C = FSK3*tC/(tSK3ET-tSK3ST);
FSK1D = FSK1*tD/(tSK1ET-tSK1ST); FSK2D = FSK2*tD/(tSK2ET-tSK2ST);
FSK3D = FSK3*tD/(tSK3ET-tSK3ST);

!Equation 4.5;
!A=Time interval 1(0-1hr), B=Time interval 2(1-2hr), C=Time interval
3(2-3hr)& D=Time interval 4(3-4hr);
tA = (t1 - t0); tB = (t2 - t1); tC = (t3 - t2); tD = (t4 - t3);

!Equation 4.6;
1000*(ZSR1A - 1) + 0.001 <= (t1 - tSR1ST); (t1 - tSR1ST) <=
1000*ZSR1A;
1000*(ZSR1B - 1) + 0.001 <= (t2 - tSR1ST); (t2 - tSR1ST) <=
1000*ZSR1B;
1000*(ZSR1C - 1) + 0.001 <= (t3 - tSR1ST); (t3 - tSR1ST) <=
1000*ZSR1C;
1000*(ZSR1D - 1) + 0.001 <= (t4 - tSR1ST); (t4 - tSR1ST) <=
1000*ZSR1D;

1000*(ZSR2A - 1) + 0.001 <= (t1 - tSR2ST); (t1 - tSR2ST) <=
1000*ZSR2A;
1000*(ZSR2B - 1) + 0.001 <= (t2 - tSR2ST); (t2 - tSR2ST) <=
1000*ZSR2B;
1000*(ZSR2C - 1) + 0.001 <= (t3 - tSR2ST); (t3 - tSR2ST) <=
1000*ZSR2C;
1000*(ZSR2D - 1) + 0.001 <= (t4 - tSR2ST); (t4 - tSR2ST) <=
1000*ZSR2D;

1000*(ZSR3A - 1) + 0.001 <= (t1 - tSR3ST); (t1 - tSR3ST) <=
1000*ZSR3A;
1000*(ZSR3B - 1) + 0.001 <= (t2 - tSR3ST); (t2 - tSR3ST) <=
1000*ZSR3B;
1000*(ZSR3C - 1) + 0.001 <= (t3 - tSR3ST); (t3 - tSR3ST) <=
1000*ZSR3C;
1000*(ZSR3D - 1) + 0.001 <= (t4 - tSR3ST); (t4 - tSR3ST) <=
1000*ZSR3D;

!Equation 4.7;
1000*(XSR1A - 1) + 0.001 <= (tSR1ET - t0); (tSR1ET - t0) <=
1000*XSR1A;
1000*(XSR1B - 1) + 0.001 <= (tSR1ET - t1); (tSR1ET - t1) <=
1000*XSR1B;
1000*(XSR1C - 1) + 0.001 <= (tSR1ET - t2); (tSR1ET - t2) <=
1000*XSR1C;
1000*(XSR1D - 1) + 0.001 <= (tSR1ET - t3); (tSR1ET - t3) <=
1000*XSR1D;

1000*(XSR2A - 1) + 0.001 <= (tSR2ET - t0); (tSR2ET - t0) <=
1000*XSR2A;
1000*(XSR2B - 1) + 0.001 <= (tSR2ET - t1); (tSR2ET - t1) <=
1000*XSR2B;
```

```
1000*(XSR2C - 1) + 0.001 <= (tSR2ET - t2); (tSR2ET - t2) <=
1000*XSR2C;
1000*(XSR2D - 1) + 0.001 <= (tSR2ET - t3); (tSR2ET - t3) <=
1000*XSR2D;

1000*(XSR3A - 1) + 0.001 <= (tSR3ET - t0); (tSR3ET - t0) <=
1000*XSR3A;
1000*(XSR3B - 1) + 0.001 <= (tSR3ET - t1); (tSR3ET - t1) <=
1000*XSR3B;
1000*(XSR3C - 1) + 0.001 <= (tSR3ET - t2); (tSR3ET - t2) <=
1000*XSR3C;
1000*(XSR3D - 1) + 0.001 <= (tSR3ET - t3); (tSR3ET - t3) <=
1000*XSR3D;

!Equation 4.8;
YSR1A = XSR1A*ZSR1A; YSR1B = XSR1B*ZSR1B; YSR1C = XSR1C*ZSR1C;
YSR1D = XSR1D*ZSR1D;
YSR2A = XSR2A*ZSR2A; YSR2B = XSR2B*ZSR2B; YSR2C = XSR2C*ZSR2C;
YSR2D = XSR2D*ZSR2D;
YSR3A = XSR3A*ZSR3A; YSR3B = XSR3B*ZSR3B; YSR3C = XSR3C*ZSR3C;
YSR3D = XSR3D*ZSR3D;

@bin(XSR1A); @bin(XSR1B); @bin(XSR1C); @bin(XSR1D);
@bin(XSR2A); @bin(XSR2B); @bin(XSR2C); @bin(XSR2D);
@bin(XSR3A); @bin(XSR3B); @bin(XSR3C); @bin(XSR3D);

@bin(ZSR1A); @bin(ZSR1B); @bin(ZSR1C); @bin(ZSR1D);
@bin(ZSR2A); @bin(ZSR2B); @bin(ZSR2C); @bin(ZSR2D);
@bin(ZSR3A); @bin(ZSR3B); @bin(ZSR3C); @bin(ZSR3D);

@bin(YSR1A); @bin(YSR1B); @bin(YSR1C); @bin(YSR1D);
@bin(YSR2A); @bin(YSR2B); @bin(YSR2C); @bin(YSR2D);
@bin(YSR3A); @bin(YSR3B); @bin(YSR3C); @bin(YSR3D);

!Equation 4.9;
1000*(ZSK1A - 1) + 0.001 <= (t1 - tSK1ST); (t1 - tSK1ST) <=
1000*ZSK1A;
1000*(ZSK1B - 1) + 0.001 <= (t2 - tSK1ST); (t2 - tSK1ST) <=
1000*ZSK1B;
1000*(ZSK1C - 1) + 0.001 <= (t3 - tSK1ST); (t3 - tSK1ST) <=
1000*ZSK1C;
1000*(ZSK1D - 1) + 0.001 <= (t4 - tSK1ST); (t4 - tSK1ST) <=
1000*ZSK1D;

1000*(ZSK2A - 1) + 0.001 <= (t1 - tSK2ST); (t1 - tSK2ST) <=
1000*ZSK2A;
1000*(ZSK2B - 1) + 0.001 <= (t2 - tSK2ST); (t2 - tSK2ST) <=
1000*ZSK2B;
1000*(ZSK2C - 1) + 0.001 <= (t3 - tSK2ST); (t3 - tSK2ST) <=
1000*ZSK2C;
1000*(ZSK2D - 1) + 0.001 <= (t4 - tSK2ST); (t4 - tSK2ST) <=
1000*ZSK2D;

1000*(ZSK3A - 1) + 0.001 <= (t1 - tSK3ST); (t1 - tSK3ST) <=
1000*ZSK3A;
1000*(ZSK3B - 1) + 0.001 <= (t2 - tSK3ST); (t2 - tSK3ST) <=
1000*ZSK3B;
1000*(ZSK3C - 1) + 0.001 <= (t3 - tSK3ST); (t3 - tSK3ST) <=
1000*ZSK3C;
```

```
1000*(ZSK3D - 1) + 0.001 <= (t4 - tSK3ST); (t4 - tSK3ST) <=
1000*ZSK3D;

!Equation 4.10;
1000*(XSK1A - 1) + 0.001 <= (tSK1ET - t0); (tSK1ET - t0) <=
1000*XSK1A;
1000*(XSK1B - 1) + 0.001 <= (tSK1ET - t1); (tSK1ET - t1) <=
1000*XSK1B;
1000*(XSK1C - 1) + 0.001 <= (tSK1ET - t2); (tSK1ET - t2) <=
1000*XSK1C;
1000*(XSK1D - 1) + 0.001 <= (tSK1ET - t3); (tSK1ET - t3) <=
1000*XSK1D;

1000*(XSK2A - 1) + 0.001 <= (tSK2ET - t0); (tSK2ET - t0) <=
1000*XSK2A;
1000*(XSK2B - 1) + 0.001 <= (tSK2ET - t1); (tSK2ET - t1) <=
1000*XSK2B;
1000*(XSK2C - 1) + 0.001 <= (tSK2ET - t2); (tSK2ET - t2) <=
1000*XSK2C;
1000*(XSK2D - 1) + 0.001 <= (tSK2ET - t3); (tSK2ET - t3) <=
1000*XSK2D;

1000*(XSK3A - 1) + 0.001 <= (tSK3ET - t0); (tSK3ET - t0) <=
1000*XSK3A;
1000*(XSK3B - 1) + 0.001 <= (tSK3ET - t1); (tSK3ET - t1) <=
1000*XSK3B;
1000*(XSK3C - 1) + 0.001 <= (tSK3ET - t2); (tSK3ET - t2) <=
1000*XSK3C;
1000*(XSK3D - 1) + 0.001 <= (tSK3ET - t3); (tSK3ET - t3) <=
1000*XSK3D;

!Equation 4.11;
YSK1A = XSK1A*ZSK1A; YSK1B = XSK1B*ZSK1B; YSK1C = XSK1C*ZSK1C;
YSK1D = XSK1D*ZSK1D;
YSK2A = XSK2A*ZSK2A; YSK2B = XSK2B*ZSK2B; YSK2C = XSK2C*ZSK2C;
YSK2D = XSK2D*ZSK2D;
YSK3A = XSK3A*ZSK3A; YSK3B = XSK3B*ZSK3B; YSK3C = XSK3C*ZSK3C;
YSK3D = XSK3D*ZSK3D;

@bin(XSK1A); @bin(XSK1B); @bin(XSK1C); @bin(XSK1D);
@bin(XSK2A); @bin(XSK2B); @bin(XSK2C); @bin(XSK2D);
@bin(XSK3A); @bin(XSK3B); @bin(XSK3C); @bin(XSK3D);

@bin(ZSK1A); @bin(ZSK1B); @bin(ZSK1C); @bin(ZSK1D);
@bin(ZSK2A); @bin(ZSK2B); @bin(ZSK2C); @bin(ZSK2D);
@bin(ZSK3A); @bin(ZSK3B); @bin(ZSK3C); @bin(ZSK3D);

@bin(YSK1A); @bin(YSK1B); @bin(YSK1C); @bin(YSK1D);
@bin(YSK2A); @bin(YSK2B); @bin(YSK2C); @bin(YSK2D);
@bin(YSK3A); @bin(YSK3B); @bin(YSK3C); @bin(YSK3D);

!Equation 4.12;
FSR1A*YSR1A = FSR1_SK1A*YSK1A + FSR1_SK2A*YSK2A + FSR1_SK3A*YSK3A +
              FSR1_WasteA + FSR1_S1A;
FSR2A*YSR2A = FSR2_SK1A*YSK1A + FSR2_SK2A*YSK2A + FSR2_SK3A*YSK3A +
              FSR2_WasteA + FSR2_S2A;
FSR3A*YSR3A = FSR3_SK1A*YSK1A + FSR3_SK2A*YSK2A + FSR3_SK3A*YSK3A +
              FSR3_WasteA + FSR3_S3A;
```

```
FSR1B*YSR1B = FSR1_SK1B*YSK1B + FSR1_SK2B*YSK2B + FSR1_SK3B*YSK3B +
              FSR1_WasteB + FSR1_S1B;
FSR2B*YSR2B = FSR2_SK1B*YSK1B + FSR2_SK2B*YSK2B + FSR2_SK3B*YSK3B +
              FSR2_WasteB + FSR2_S2B;
FSR3B*YSR3B = FSR3_SK1B*YSK1B + FSR3_SK2B*YSK2B + FSR3_SK3B*YSK3B +
              FSR3_WasteB + FSR3_S3B;

FSR1C*YSR1C = FSR1_SK1C*YSK1C + FSR1_SK2C*YSK2C + FSR1_SK3C*YSK3C +
              FSR1_WasteC + FSR1_S1C;
FSR2C*YSR2C = FSR2_SK1C*YSK1C + FSR2_SK2C*YSK2C + FSR2_SK3C*YSK3C +
              FSR2_WasteC + FSR2_S2C;
FSR3C*YSR3C = FSR3_SK1C*YSK1C + FSR3_SK2C*YSK2C + FSR3_SK3C*YSK3C +
              FSR3_WasteC + FSR3_S3C;

FSR1D*YSR1D = FSR1_SK1D*YSK1D + FSR1_SK2D*YSK2D + FSR1_SK3D*YSK3D +
              FSR1_WasteD + FSR1_S1D;
FSR2D*YSR2D = FSR2_SK1D*YSK1D + FSR2_SK2D*YSK2D + FSR2_SK3D*YSK3D +
              FSR2_WasteD + FSR2_S2D;
FSR3D*YSR3D = FSR3_SK1D*YSK1D + FSR3_SK2D*YSK2D + FSR3_SK3D*YSK3D +
              FSR3_WasteD + FSR3_S3D;

!Equation 4.13;
FSK1A*YSK1A = FSR1_SK1A*YSR1A + FSR2_SK1A*YSR2A + FSR3_SK1A*YSR3A +
              FFR1_SK1A + FFR2_SK1A + FS1_SK1A + FS2_SK1A +
              FS3_SK1A;
FSK2A*YSK2A = FSR1_SK2A*YSR1A + FSR2_SK2A*YSR2A + FSR3_SK2A*YSR3A +
              FFR1_SK2A + FFR2_SK2A + FS1_SK2A + FS2_SK2A +
              FS3_SK2A;
FSK3A*YSK3A = FSR1_SK3A*YSR1A + FSR2_SK3A*YSR2A + FSR3_SK3A*YSR3A +
              FFR1_SK3A + FFR2_SK3A + FS1_SK3A + FS2_SK3A +
              FS3_SK3A;

FSK1B*YSK1B = FSR1_SK1B*YSR1B + FSR2_SK1B*YSR2B + FSR3_SK1B*YSR3B +
              FFR1_SK1B + FFR2_SK1B + FS1_SK1B + FS2_SK1B +
              FS3_SK1B;
FSK2B*YSK2B = FSR1_SK2B*YSR1B + FSR2_SK2B*YSR2B + FSR3_SK2B*YSR3B +
              FFR1_SK2B + FFR2_SK2B + FS1_SK2B + FS2_SK2B +
              FS3_SK2B;
FSK3B*YSK3B = FSR1_SK3B*YSR1B + FSR2_SK3B*YSR2B + FSR3_SK3B*YSR3B +
              FFR1_SK3B + FFR2_SK3B + FS1_SK3B + FS2_SK3B +
              FS3_SK3B;

FSK1C*YSK1C = FSR1_SK1C*YSR1C + FSR2_SK1C*YSR2C + FSR3_SK1C*YSR3C +
              FFR1_SK1C + FFR2_SK1C + FS1_SK1C + FS2_SK1C +
              FS3_SK1C;
FSK2C*YSK2C = FSR1_SK2C*YSR1C + FSR2_SK2C*YSR2C + FSR3_SK2C*YSR3C +
              FFR1_SK2C + FFR2_SK2C + FS1_SK2C + FS2_SK2C +
              FS3_SK2C;
FSK3C*YSK3C = FSR1_SK3C*YSR1C + FSR2_SK3C*YSR2C + FSR3_SK3C*YSR3C +
              FFR1_SK3C + FFR2_SK3C + FS1_SK3C + FS2_SK3C +
              FS3_SK3C;

FSK1D*YSK1D = FSR1_SK1D*YSR1D + FSR2_SK1D*YSR2D + FSR3_SK1D*YSR3D +
              FFR1_SK1D + FFR2_SK1D + FS1_SK1D + FS2_SK1D +
              FS3_SK1D;
FSK2D*YSK2D = FSR1_SK2D*YSR1D + FSR2_SK2D*YSR2D + FSR3_SK2D*YSR3D +
              FFR1_SK2D + FFR2_SK2D + FS1_SK2D + FS2_SK2D +
              FS3_SK2D;
FSK3D*YSK3D = FSR1_SK3D*YSR1D + FSR2_SK3D*YSR2D + FSR3_SK3D*YSR3D +
```

```
              FFR1_SK3D + FFR2_SK3D + FS1_SK3D + FS2_SK3D +
              FS3_SK3D;

!Equations 4.14 & 4.15;
FSK1A*YSK1A*OSK1 >= FSR1_SK1A*YSR1A*OSR1 + FSR2_SK1A*YSR2A*OSR2 +
                    FSR3_SK1A*YSR3A*OSR3 + FFR1_SK1A*OFFR1 +
                    FFR2_SK1A*OFFR2 + FS1_SK1A*OSR1 + FS2_SK1A*OSR2
                    + FS3_SK1A*OSR3;
FSK2A*YSK2A*OSK2 >= FSR1_SK2A*YSR1A*OSR1 + FSR2_SK2A*YSR2A*OSR2 +
                    FSR3_SK2A*YSR3A*OSR3 + FFR1_SK2A*OFFR1 +
                    FFR2_SK2A*OFFR2 + FS1_SK2A*OSR1 + FS2_SK2A*OSR2
                    + FS3_SK2A*OSR3;
FSK3A*YSK3A*OSK3 >= FSR1_SK3A*YSR1A*OSR1 + FSR2_SK3A*YSR2A*OSR2 +
                    FSR3_SK3A*YSR3A*OSR3 + FFR1_SK3A*OFFR1 +
                    FFR2_SK3A*OFFR2 + FS1_SK3A*OSR1 + FS2_SK3A*OSR2
                    + FS3_SK3A*OSR3;

FSK1B*YSK1B*OSK1 >= FSR1_SK1B*YSR1B*OSR1 + FSR2_SK1B*YSR2B*OSR2 +
                    FSR3_SK1B*YSR3B*OSR3 + FFR1_SK1B*OFFR1 +
                    FFR2_SK1B*OFFR2 + FS1_SK1B*OSR1 + FS2_SK1B*OSR2
                    + FS3_SK1B*OSR3;
FSK2B*YSK2B*OSK2 >= FSR1_SK2B*YSR1B*OSR1 + FSR2_SK2B*YSR2B*OSR2 +
                    FSR3_SK2B*YSR3B*OSR3 + FFR1_SK2B*OFFR1 +
                    FFR2_SK2B*OFFR2 + FS1_SK2B*OSR1 + FS2_SK2B*OSR2
                    + FS3_SK2B*OSR3;
FSK3B*YSK3B*OSK3 >= FSR1_SK3B*YSR1B*OSR1 + FSR2_SK3B*YSR2B*OSR2 +
                    FSR3_SK3B*YSR3B*OSR3 + FFR1_SK3B*OFFR1 +
                    FFR2_SK3B*OFFR2 + FS1_SK3B*OSR1 + FS2_SK3B*OSR2
                    + FS3_SK3B*OSR3;

FSK1C*YSK1C*OSK1 >= FSR1_SK1C*YSR1C*OSR1 + FSR2_SK1C*YSR2C*OSR2 +
                    FSR3_SK1C*YSR3C*OSR3 + FFR1_SK1C*OFFR1 +
                    FFR2_SK1C*OFFR2 + FS1_SK1C*OSR1 + FS2_SK1C*OSR2
                    + FS3_SK1C*OSR3;
FSK2C*YSK2C*OSK2 >= FSR1_SK2C*YSR1C*OSR1 + FSR2_SK2C*YSR2C*OSR2 +
                    FSR3_SK2C*YSR3C*OSR3 + FFR1_SK2C*OFFR1 +
                    FFR2_SK2C*OFFR2 + FS1_SK2C*OSR1 + FS2_SK2C*OSR2
                    + FS3_SK2C*OSR3;
FSK3C*YSK3C*OSK3 >= FSR1_SK3C*YSR1C*OSR1 + FSR2_SK3C*YSR2C*OSR2 +
                    FSR3_SK3C*YSR3C*OSR3 + FFR1_SK3C*OFFR1 +
                    FFR2_SK3C*OFFR2 + FS1_SK3C*OSR1 + FS2_SK3C*OSR2
                    + FS3_SK3C*OSR3;

FSK1D*YSK1D*OSK1 >= FSR1_SK1D*YSR1D*OSR1 + FSR2_SK1D*YSR2D*OSR2 +
                    FSR3_SK1D*YSR3D*OSR3 + FFR1_SK1D*OFFR1 +
                    FFR2_SK1D*OFFR2 + FS1_SK1D*OSR1 + FS2_SK1D*OSR2
                    + FS3_SK1D*OSR3;
FSK2D*YSK2D*OSK2 >= FSR1_SK2D*YSR1D*OSR1 + FSR2_SK2D*YSR2D*OSR2 +
                    FSR3_SK2D*YSR3D*OSR3 + FFR1_SK2D*OFFR1 +
                    FFR2_SK2D*OFFR2 + FS1_SK2D*OSR1 + FS2_SK2D*OSR2
                    + FS3_SK2D*OSR3;
FSK3D*YSK3D*OSK3 >= FSR1_SK3D*YSR1D*OSR1 + FSR2_SK3D*YSR2D*OSR2 +
                    FSR3_SK3D*YSR3D*OSR3 + FFR1_SK3D*OFFR1 +
                    FFR2_SK3D*OFFR2 + FS1_SK3D*OSR1 + FS2_SK3D*OSR2
                    + FS3_SK3D*OSR3;

!Equation 4.16;
Total_FFR1 = FFR1_SK1A + FFR1_SK2A + FFR1_SK3A + FFR1_SK1B +
             FFR1_SK2B + FFR1_SK3B + FFR1_SK1C + FFR1_SK2C +
```

```
             FFR1_SK3C + FFR1_SK1D + FFR1_SK2D + FFR1_SK3D;

Total_FFR2 = FFR2_SK1A + FFR2_SK2A + FFR2_SK3A + FFR2_SK1B +
             FFR2_SK2B + FFR2_SK3B + FFR2_SK1C + FFR2_SK2C +
             FFR2_SK3C + FFR2_SK1D + FFR2_SK2D + FFR2_SK3D;

!Equation 4.17;
Total_Waste = FSR1_WasteA + FSR2_WasteA + FSR3_WasteA + FS1_WasteA +
              FS2_WasteA + FS3_WasteA +
              FSR1_WasteB + FSR2_WasteB + FSR3_WasteB + FS1_WasteB +
              FS2_WasteB + FS3_WasteB +FSR1_WasteC + FSR2_WasteC +
              FSR3_WasteC + FS1_WasteC + FS2_WasteC + FS3_WasteC +
              FSR1_WasteD + FSR2_WasteD + FSR3_WasteD + FS1_WasteD +
              FS2_WasteD + FS3_WasteD;

!Equation 4.18;
FSR1_S1A + CSTS1A = FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB +
                    CSTS1B;
FSR2_S2A + CSTS2A = FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB +
                    CSTS2B;
FSR3_S3A + CSTS3A = FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB +
                    CSTS3B;

FSR1_S1B + CSTS1B = FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC +
                    CSTS1C;
FSR2_S2B + CSTS2B = FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC +
                    CSTS2C;
FSR3_S3B + CSTS3B = FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC +
                    CSTS3C;

FSR1_S1C + CSTS1C = FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD +
                    CSTS1D;
FSR2_S2C + CSTS2C = FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD +
                    CSTS2D;
FSR3_S3C + CSTS3C = FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD +
                    CSTS3D;

FSR1_S1D + CSTS1D = FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA +
                    CSTS1A;
FSR2_S2D + CSTS2D = FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA +
                    CSTS2A;
FSR3_S3D + CSTS3D = FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA +
                    CSTS3A;

!Equations 4.23 to 4.30 & 4.32 to 4.33;
H0A = QHA;
H1A = H0A + (70-60)*CP*(FSR1_SK2A+FSR1_WasteA+FS1_SK2A+FS1_WasteA)
      /3600;
H2A = H1A + (60-55)*CP*(FSR1_SK2A+FSR1_WasteA+FS1_SK2A+FS1_WasteA+
      FS2_SK2A+FS2_WasteA)/3600;
H3A = H2A + (55-45)*CP*(FSR1_SK2A+FSR1_WasteA-FSR3_SK2A-FFR1_SK2A-
      FFR2_SK2A+FS1_SK2A+FS1_WasteA+FS2_SK2A+FS2_WasteA-FS3_SK2A)
      /3600;
H4A = H3A + (45-40)*CP*(FSR1_WasteA-FFR1_SK2A-FFR2_SK2A+FS1_WasteA+
      FS2_WasteA)/3600;
H5A = H4A + (40-35)*CP*(FSR1_WasteA-FFR1_SK2A+FS1_WasteA+FS2_WasteA)
      /3600;
H6A = H5A + (35-30)*CP*(FSR1_WasteA+FSR3_WasteA-FFR1_SK2A+FS1_WasteA
      +FS2_WasteA+FS3_WasteA)/3600;
```

```
H7A = H6A + (30-25)*CP*(FSR1_WasteA+FSR3_WasteA+FS1_WasteA+
      FS2_WasteA+FS3_WasteA)/3600;
H7A = QCA;

H0B = QHB;
H1B = H0B + (90-80)*CP*(-FSR1_SK1B-FSR2_SK1B-FSR3_SK1B-FFR1_SK1B-
      FS3_SK1B-FS3_SK3B-FFR2_SK1B-FS1_SK1B-FS2_SK1B)/3600;
H2B = H1B + (80-70)*CP*(-FSR2_SK1B-FSR3_SK1B-FFR1_SK1B-FS3_SK1B-
      FS3_SK3B-FFR2_SK1B-FS2_SK1B)/3600;
H3B = H2B + (70-60)*CP*(FSR1_SK2B+FSR1_SK3B+FSR1_WasteB-FSR3_SK1B-
      FSR3_SK3B-FFR1_SK1B-FS3_SK1B-FFR1_SK3B-FS3_SK3B-FFR2_SK1B-
      FFR2_SK3B+FS1_SK2B+FS1_SK3B+FS1_WasteB)/3600;
H4B = H3B + (60-55)*CP*(FSR1_SK2B+FSR1_WasteB+FSR2_SK2B+FSR2_WasteB-
      FSR3_SK1B-FSR3_SK3B-FFR1_SK1B-FS3_SK1B-FFR1_SK3B-FS3_SK3B-
      FFR2_SK1B+FS2_SK2B-FFR2_SK3B+FS1_SK2B+FS1_WasteB+FS2_WasteB)
      /3600;
H5B = H4B + (55-45)*CP*(FSR1_SK2B-FS3_SK2B+FSR1_WasteB+FSR2_SK2B+
      FSR2_WasteB-FSR3_SK1B-FSR3_SK2B-FSR3_SK3B-FFR1_SK1B-FFR1_SK2B
      -FS3_SK1B-FFR1_SK3B-FS3_SK3B-FFR2_SK1B+FS2_SK2B-FFR2_SK2B-
      FFR2_SK3B+FS1_SK2B+FS1_WasteB+FS2_WasteB)/3600;
H6B = H5B + (45-40)*CP*(FSR1_WasteB+FSR2_WasteB-FFR1_SK1B-FFR1_SK2B-
      FFR1_SK3B-FFR2_SK1B-FFR2_SK2B-FFR2_SK3B+FS1_WasteB+FS2_WasteB)
      /3600;
H7B = H6B + (40-35)*CP*(FSR1_WasteB+FSR2_WasteB-FFR1_SK1B-FFR1_SK2B-
      FFR1_SK3B+FS1_WasteB+FS2_WasteB)/3600;
H8B = H7B + (35-30)*CP*(FSR1_WasteB+FSR2_WasteB+FSR3_WasteB+
      FS3_WasteB-FFR1_SK1B-FFR1_SK2B-
      FFR1_SK3B+FS1_WasteB+FS2_WasteB)/3600;
H9B = H8B + (30-25)*CP*(FSR1_WasteB+FSR2_WasteB+FSR3_WasteB+
      FS1_WasteB+FS2_WasteB+FS3_WasteB)/3600;
H9B = QCB;

H0C = QHC;
H1C = H0C + (90-80)*CP*(-FSR1_SK1C-FSR3_SK1C-FFR1_SK1C-FFR2_SK1C-
      FS1_SK1C-FS2_SK1C-FS3_SK1C)/3600;
H2C = H1C + (80-70)*CP*(-FSR3_SK1C-FFR1_SK1C-FFR2_SK1C-FS2_SK1C-
      FS3_SK1C)/3600;
H3C = H2C + (70-60)*CP*(FSR1_SK3C+FSR1_WasteC-FSR3_SK1C-FSR3_SK3C-
      FFR1_SK1C-FFR1_SK3C-FFR2_SK1C-FFR2_SK3C+FS1_SK3C+FS1_WasteC-
      FS3_SK1C-FS3_SK3C)/3600;
H4C = H3C + (60-45)*CP*(FSR1_WasteC-FSR3_SK1C-FSR3_SK3C-FFR1_SK1C-
      FFR1_SK3C-FFR2_SK1C-FFR2_SK3C+FS1_WasteC+FS2_WasteC-FS3_SK1C
      -FS3_SK3C)/3600;
H5C = H4C + (45-40)*CP*(FSR1_WasteC-FFR1_SK1C-FFR1_SK3C-FFR2_SK1C-
      FFR2_SK3C+FS1_WasteC+FS2_WasteC)/3600;
H6C = H5C + (40-35)*CP*(FSR1_WasteC-FFR1_SK1C-FFR1_SK3C+FS1_WasteC+
      FS2_WasteC)/3600;
H7C = H6C + (35-30)*CP*(FSR1_WasteC+FSR3_WasteC-FFR1_SK1C-FFR1_SK3C+
      FS1_WasteC+FS2_WasteC+FS3_WasteC)/3600;
H8C = H7C + (30-25)*CP*(FSR1_WasteC+FSR3_WasteC+FS1_WasteC+
      FS2_WasteC+FS3_WasteC)/3600;
H8C = QCC;

H0D = QHD;
H1D = H0D + (70-60)*CP*(-FSR3_SK3D-FFR1_SK3D-FFR2_SK3D+FS1_SK3D+
      FS1_WasteD-FS3_SK3D)/3600;
H2D = H1D + (60-45)*CP*(-FSR3_SK3D-FFR1_SK3D-FFR2_SK3D+FS1_WasteD+
      FS2_WasteD-FS3_SK3D)/3600;
H3D = H2D + (45-40)*CP*(-FFR1_SK3D-FFR2_SK3D+FS1_WasteD+FS2_WasteD)
```

```
      /3600;
H4D = H3D + (40-35)*CP*(-FFR1_SK3D+FS1_WasteD+FS2_WasteD)/3600;
H5D = H4D + (35-30)*CP*(FSR3_WasteD-FFR1_SK3D+FS1_WasteD+FS2_WasteD+
      FS3_WasteD)/3600;
H6D = H5D + (30-25)*CP*(FSR3_WasteD+FS1_WasteD+FS2_WasteD+
      FS3_WasteD)/3600;
H6D = QCD;

!Equation 4.31;
H0A>=0; H1A>=0; H2A>=0; H3A>=0; H4A>=0; H5A>=0; H6A>=0; H7A>=0;
H0B>=0; H1B>=0; H2B>=0; H3B>=0; H4B>=0; H5B>=0; H6B>=0; H7B>=0;
H8B>=0; H9B>=0;
H0C>=0; H1C>=0; H2C>=0; H3C>=0; H4C>=0; H5C>=0; H6C>=0; H7C>=0;
H8C>=0;
H0D>=0; H1D>=0; H2D>=0; H3D>=0; H4D>=0; H5D>=0; H6D>=0;

!Equation 4.34;
Total_QH = QHA + QHB + QHC + QHD;

!Equation 4.35;
Total_QC = QCA + QCB + QCC + QCD;

!Equation 4.37;
!UC = value for unit conversion;
CS1 = (I/Ib)*(A0*(A1*UC*CSTS1)^d)*AF;
CS2 = (I/Ib)*(A0*(A1*UC*CSTS2)^d)*AF;
CS3 = (I/Ib)*(A0*(A1*UC*CSTS3)^d)*AF;

!Equation 4.38;
AF = 0.229;

!Equation 4.39;
CSTS1>= FS1_SK1A + FS1_SK2A + FS1_SK3A + FS1_WasteA + CSTS1A;
CSTS1>= FS1_SK1B + FS1_SK2B + FS1_SK3B + FS1_WasteB + CSTS1B;
CSTS1>= FS1_SK1C + FS1_SK2C + FS1_SK3C + FS1_WasteC + CSTS1C;
CSTS1>= FS1_SK1D + FS1_SK2D + FS1_SK3D + FS1_WasteD + CSTS1D;

CSTS2>= FS2_SK1A + FS2_SK2A + FS2_SK3A + FS2_WasteA + CSTS2A;
CSTS2>= FS2_SK1B + FS2_SK2B + FS2_SK3B + FS2_WasteB + CSTS2B;
CSTS2>= FS2_SK1C + FS2_SK2C + FS2_SK3C + FS2_WasteC + CSTS2C;
CSTS2>= FS2_SK1D + FS2_SK2D + FS2_SK3D + FS2_WasteD + CSTS2D;

CSTS3>= FS3_SK1A + FS3_SK2A + FS3_SK3A + FS3_WasteA + CSTS3A;
CSTS3>= FS3_SK1B + FS3_SK2B + FS3_SK3B + FS3_WasteB + CSTS3B;
CSTS3>= FS3_SK1C + FS3_SK2C + FS3_SK3C + FS3_WasteC + CSTS3C;
CSTS3>= FS3_SK1D + FS3_SK2D + FS3_SK3D + FS3_WasteD + CSTS3D;

!Equation 4.40;
CSTS1>=0; CSTS2>=0; CSTS3>=0;

!Unit conversion;
!1 meter cube = 264.1721 gallon;
UC = 0.2642;

!Equation 4.42;
Total_Cost_O = Total_QH*CHU*Nb + Total_QC*CCU*Nb;

!Additional constraints (Equations 5.5 & 5.6);
Total_FFR1 =0;
```

```
Total_FFR2 =0;

!Data;
FSR1 = 10983;  FSR2 = 1766; FSR3 = 5940;
FSK1 = 5436;  FSK2 = 3986; FSK3 = 3381;
OSK1 = 35; OSK2 = 25; OSK3 = 40; OFFR1 = 3; OFFR2 = 6;
OSR1 = 38; OSR2 = 25;  OSR3 = 7;
tSR1ST = 0; tSR1ET = 3;
tSR2ST = 1; tSR2ET = 2;
tSR3ST = 0; tSR3ET = 4;
tSK1ST = 1; tSK1ET = 3;
tSK2ST = 0; tSK2ET = 2;
tSK3ST = 1; tSK3ET = 4;
Nb = 1980;
Cm1 = 0.00132;
Cm2 = 0.00088;
CHU = 0.017;
CCU = 0.006;
CP = 4.2;
I = 572.7;
Ib = 394;
A0 = 210;
A1 = 1.1;
d = 0.51;
t0 = 0; t1 = 1; t2 = 2; t3 = 3; t4 = 4;
End
```

```
  Global optimal solution found.
  Objective value:                              4808.807
  Objective bound:                              4808.807
  Infeasibilities:                              0.000000
  Extended solver steps:                               1
  Total solver iterations:                           219
```

Variable	Value	Reduced Cost
TOTAL_COST_O	4808.807	0.000000
FSR1A	3661.000	0.000000
FSR2A	1766.000	0.000000
FSR3A	1485.000	0.000000
FSR1B	3661.000	0.000000
FSR2B	1766.000	0.000000
FSR3B	1485.000	0.000000
FSR1C	3661.000	0.000000
FSR2C	1766.000	0.000000
FSR3C	1485.000	0.000000
FSR1D	3661.000	0.000000
FSR2D	1766.000	0.000000
FSR3D	1485.000	0.000000
FSK1A	2718.000	0.000000
FSK2A	1993.000	0.000000
FSK3A	1127.000	0.000000
FSK1B	2718.000	0.000000
FSK2B	1993.000	0.000000
FSK3B	1127.000	0.000000
FSK1C	2718.000	0.000000
FSK2C	1993.000	0.000000
FSK3C	1127.000	0.000000

FSK1D	2718.000	0.000000
FSK2D	1993.000	0.000000
FSK3D	1127.000	0.000000
ZSR1A	1.000000	0.000000
ZSR1B	1.000000	0.000000
ZSR1C	1.000000	0.000000
ZSR1D	1.000000	0.000000
ZSR2A	0.000000	0.000000
ZSR2B	1.000000	0.000000
ZSR2C	1.000000	0.000000
ZSR2D	1.000000	0.000000
ZSR3A	1.000000	0.000000
ZSR3B	1.000000	0.000000
ZSR3C	1.000000	0.000000
ZSR3D	1.000000	0.000000
XSR1A	1.000000	0.000000
XSR1B	1.000000	0.000000
XSR1C	1.000000	0.000000
XSR1D	0.000000	0.000000
XSR2A	1.000000	0.000000
XSR2B	1.000000	0.000000
XSR2C	0.000000	0.000000
XSR2D	0.000000	0.000000
XSR3A	1.000000	0.000000
XSR3B	1.000000	0.000000
XSR3C	1.000000	0.000000
XSR3D	1.000000	0.000000
YSR1A	1.000000	2497.097
YSR1B	1.000000	3408.105
YSR1C	1.000000	2953.860
YSR1D	0.000000	0.000000
YSR2A	0.000000	0.000000
YSR2B	1.000000	1346.220
YSR2C	0.000000	0.000000
YSR2D	0.000000	0.000000
YSR3A	1.000000	544.5504
YSR3B	1.000000	257.8166
YSR3C	1.000000	649.1340
YSR3D	1.000000	752.5298
ZSK1A	0.000000	0.000000
ZSK1B	1.000000	0.000000
ZSK1C	1.000000	0.000000
ZSK1D	1.000000	0.000000
ZSK2A	1.000000	0.000000
ZSK2B	1.000000	0.000000
ZSK2C	1.000000	0.000000
ZSK2D	1.000000	0.000000
ZSK3A	0.000000	0.000000
ZSK3B	1.000000	0.000000
ZSK3C	1.000000	0.000000
ZSK3D	1.000000	0.000000
XSK1A	1.000000	0.000000
XSK1B	1.000000	0.000000
XSK1C	1.000000	0.000000
XSK1D	0.000000	0.000000
XSK2A	1.000000	0.000000
XSK2B	1.000000	0.000000
XSK2C	0.000000	0.000000
XSK2D	0.000000	0.000000

XSK3A	1.000000	0.000000
XSK3B	1.000000	0.000000
XSK3C	1.000000	0.000000
XSK3D	1.000000	0.000000
YSK1A	0.000000	-1608.282
YSK1B	1.000000	-2019.274
YSK1C	1.000000	-2019.274
YSK1D	0.000000	-1608.282
YSK2A	1.000000	-1202.720
YSK2B	1.000000	-1504.549
YSK2C	0.000000	-966.8043
YSK2D	0.000000	-966.8043
YSK3A	0.000000	-702.9099
YSK3B	1.000000	-1224.012
YSK3C	1.000000	-806.3063
YSK3D	1.000000	-702.9106
FSR1_SK2A	138.1006	0.000000
FSR1_WASTEA	326.7836	0.000000
FSR1_S1A	3196.116	0.000000
FSR3_SK2A	312.2924	0.000000
FSR3_WASTEA	649.2258	0.000000
FSR3_S3A	523.4818	0.000000
FSR1_SK2B	49.14186	0.000000
FSR1_SK3B	606.6149	0.000000
FSR1_S1B	3005.243	0.000000
FSR2_SK2B	938.5182	0.000000
FSR2_SK3B	138.6416	0.000000
FSR2_WASTEB	88.52739	0.000000
FSR2_S2B	600.3128	0.000000
FSR3_SK1B	263.0323	0.3003000E-06
FSR3_SK3B	51.34879	0.000000
FSR3_WASTEB	755.4643	0.000000
FSR3_S3B	415.1546	0.000000
FSR1_SK3C	556.6668	0.000000
FSR1_WASTEC	62.57732	0.000000
FSR1_S1C	3041.756	0.000000
FSR3_WASTEC	1077.247	0.000000
FSR3_S3C	407.7533	0.000000
FSR3_WASTED	1002.638	0.000000
FSR3_S3D	482.3625	0.000000
FS1_SK2A	1019.125	0.2772003E-06
FS3_SK2A	523.4818	0.2772003E-06
FS1_SK1B	2454.968	0.2310002E-06
FS1_SK2B	375.6961	0.2772003E-06
FS2_SK2B	322.8164	0.2772003E-06
FS3_SK2B	306.8274	0.2772003E-06
FS1_SK3B	191.7531	0.4851005E-06
FS2_SK3B	138.6416	0.4851005E-06
FS1_SK1C	2454.968	0.2310002E-06
FS3_SK1C	263.0323	0.000000
FS1_SK3C	456.0595	0.4851005E-06
FS2_SK3C	3.042525	0.4851005E-06
FS3_SK3C	111.2312	0.4851005E-06
FS1_SK3D	1127.000	0.4851005E-06
TOTAL_FFR1	0.000000	0.000000
TOTAL_FFR2	0.000000	0.000000
TOTAL_WASTE	5886.000	0.000000
FS1_WASTEA	446.0244	0.000000
FS2_WASTEA	15.79057	0.000000

FS1_WASTEB	141.6389	0.000000
FS2_WASTEB	88.52739	0.000000
FS3_WASTEB	108.3272	0.000000
FS1_WASTEC	129.9844	0.000000
FS2_WASTEC	15.70375	0.000000
FS3_WASTEC	74.60918	0.000000
FS1_WASTED	445.8978	0.000000
FS2_WASTED	15.79057	0.000000
FS3_WASTED	441.2431	0.000000
CSTS1A	3.708433	0.000000
CSTS1B	35.76836	0.000000
CSTS2A	549.9854	0.000000
CSTS3B	108.3272	0.000000
CSTS2C	581.5665	0.000000
CSTS3C	74.60918	0.000000
CSTS1D	1468.858	0.000000
CSTS2D	565.7759	0.000000
CSTS3D	41.11936	0.000000
QHA	0.000000	0.000000
H1A	22.51706	0.000000
H2A	33.86770	0.000000
H3A	46.81829	0.000000
H4A	51.41845	0.000000
H5A	56.01861	0.000000
H6A	64.40591	0.000000
H7A	72.79322	0.000000
QCA	72.79322	0.000000
H0B	34.77871	0.000000
QHB	34.77871	0.000000
H1B	3.068710	0.000000
H3B	12.25541	0.000000
H4B	22.11657	0.000000
H5B	38.25925	0.000000
H6B	40.11829	0.000000
H7B	41.97734	0.000000
H8B	48.87517	0.000000
H9B	55.77300	0.000000
QCB	55.77300	0.000000
H0C	34.77871	0.000000
QHC	34.77871	0.000000
H1C	3.068710	0.000000
H3C	9.695286	0.000000
H4C	6.790321	0.000000
H5C	8.005203	0.000000
H6C	9.220085	0.000000
H7C	17.15413	0.000000
H8C	25.08817	0.000000
QCC	25.08817	0.000000
QHD	0.000000	0.000000
H1D	18.35047	0.000000
H2D	26.43002	0.000000
H3D	29.12320	0.000000
H4D	31.81639	0.000000
H5D	42.93221	0.000000
H6D	54.04803	0.000000
QCD	54.04803	0.000000
TOTAL_QH	69.55742	0.000000
TOTAL_QC	207.7024	0.000000
CS1	2282.423	0.000000

CSTS1	3199.824	0.000000
CS2	1048.678	0.000000
CSTS2	696.4079	0.000000
CS3	906.6122	0.000000
CSTS3	523.4818	0.000000

Note: The flow and heat terms which are not shown are equal to zero.