Development and Testing of an Optimization Model at Selected Eskom Sites for an Integrated Water Solution

Volume 2: Cooling Tower Model Development

Report to the
Water Research Commission

by

Vincent Gololo and Thokozani Majozi*

School of Chemical and Metallurgical Engineering, University of the Witwatersrand
* email: Thokozani.Majozi@wits.ac.za

WRC Report No. 2289/2/16
ISBN 978-1-4312-0818-0

July 2016

Obtainable from

Water Research Commission
Private Bag X03
GEZINA, 0031

orders@wrc.org.za or download from www.wrc.org.za

DISCLAIMER

This report has been reviewed by the Water Research Commission (WRC) and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the WRC, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

SUMMARY

Cooling water systems are used to remove excess heat from a chemical process to the atmosphere. The primary components of these systems are the cooling tower and the heat exchanger network. There is a strong interaction between these individual components, thus their performances are interrelated. Most published research in this area has focused mainly on optimization of the individual components, i.e. optimization of heat exchanger network or optimization of the cooling towers. This approach does not optimize the cooling water system as a whole. Previous research work in which a holistic approach was used is limited to cooling water systems with single cooling water source.

This work presents a technique for integrated optimisation of complex cooling water systems. The system under consideration consists of multiple cooling towers each supplying a set of heat exchangers. A superstructural approach is employed to explore all possible combinations between the heat exchangers and the cooling towers. The cooling water reuse opportunities within the heat exchanger networks are also explored. A detailed mathematical model consisting of the cooling towers and the heat exchanger networks model is developed. Two practical scenarios are considered and the mathematical formulations for Case I and II yield nonlinear programing (NLP) and mixed integer nonlinear programming (MINLP) structure respectively.

Although the reuse/recycle philosophy offers a good debottlenecking opportunity, the topology of the associated cooling water network is more complex, hence prone to higher pressure drop than the conventional parallel design. This is due to an increased network pressure drop associated with additional reuse/recycle streams. Therefore, it is essential to consider pressure drop during the synthesis of cooling water networks where the reuse/recycle philosophy is employed. The ongoing research in this area is only limited to cooling water networks consisting of a single cooling water source. The common technique used is mathematical optimization using either superstructural or non superstructural approach.

This work further presents a mathematical technique for pressure drop optimization in cooling water systems consisting of multiple cooling towers. The proposed technique is based on the Critical Path Algorithm and the superstructural approach. The Critical Path Algorithm is used to select the cooling water network with minimum pressure drop whilst the superstructural approach allows for cooling water reuse. The technique which was previously used in a cooling water network with single source is modified and applied in a cooling water network with multiple sources. The mathematical formulation is developed considering two cases. Both cases yield mixed integer nonlinear programming (MINLP) models. The cooling tower model is also used to predict the exit condition of the cooling tower given the inlet conditions from the cooling water network model.

The results show up to 29% decrease in total circulating cooling water flowrate when the cooling water system is debottlenecked without considering pressure drop. Consequently, the overall cooling towers effectiveness was improved by up to 5%. When considering pressure drop the results showed up to 26% decrease in total circulating water flowrate.

REFERENCE GROUP

The project team wishes to thank the following people for their contributions to the project.

Dr Valerie Naidoo	Water Research Commission
Dr Jo Burgess	Water Research Commission
Mr Gerhard Gericke	Eskom
Dr Tzu-Hua Jenny Huang	Sasol
Ms Ritva Mühlbauer	Anglo American
Dr Ndeke Musee	Council for Scientific and Industrial Research
Ms Kerry Slatter	Kalao Solutions
Ms Dorian Mokhonoana	Eskom
Mr Enos Sitabule	Sasol
Prof Ochieng Aoyi	Vaal University of Technology

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS and ACRONYMS

a, a_{fi}	Surface area per unit volume (m^2/m^3)
A	Area (m^2)
A_{fr}	Frontal area (m^2)
CC	Cycles of concentration
CP	Specific heat capacity times flowrate (kJ/ (s.^{o}C))
c_p	Specific heat capacity (J/ (kg.^{o}C))
CW	Cooling water
CWR	Cooling water return
F	Flowrate (kg/s)
H	Enthalpy (J/kg)
Q	Rate of heat transfer (W)
h	Heat transfer coefficient (W/ (m^2.s))
HP	High pressure steam
HX	Heat exchanger
K, K_a	Mass transfer coefficient (kg/ (m^2.s))
Le_f	Lewis factor
LP	Low pressure steam
m	Flowrate (kg/s)
T	Temperature (K)
V	Volume (m^3)
w	Humidity (kg/kg)
z	Cooling tower height (m)
ρ	Density (kg/m^3)
η	Efficiency
ε	Effectiveness
δ	Enthalpy correction factor (kJ/kg)
a_d , d_{da} , ATD , b_{db} , x,y,z,n	Cooling towers fill parameters/constants

SUBSCRIPTS

a	air
B	blowdown
c	cold
CT	cooling tower
E	evaporation
h	hot
in	inlet
M	makeup
ma	moist air
min	minimum
max	maximum
out	outlet
s	saturation
v	vapor
w	water
wb	wet bulb

This page was left blank intentionally

1 INTRODUCTION

1.1 Background

Cooling water systems use the mechanism of evaporative cooling to remove heat energy from the process to the atmosphere. They are mainly classified into two categories, open and closed loop cooling water systems. The systems consist of cooling towers, pumping system and cooling water network. Cooling water is pumped from the cooling source through the piping system to the cooling water using operations. Heat energy is then transferred from the process into the cooling water. In closed loop cooling water systems the heated water is recycled back to the cooling source where it is cooled and recycled back to the cooling water using operations. However, in open loop cooling water systems the heated water is discarded. Closed loop cooling water systems are more popular in industrial application because they have minimal environmental impact.

Most of the research work in this area has focus on optimization of individual components of the system. Bernier (1994) studied the influence of the water inlet temperature on the cooling tower performance. Khan et al. (2002), Kröger and Kloppers (2004) and Naphon (2005) looked at the heat and mass transfer characteristics of the cooling tower. Rezaei et al. (2010) presented a mathematical model for hybrid cooling circuit. The authors attempted to reduce water consumption by combining evaporative cooling with dry cooling. A similar study was conducted by Tarighaleslami et al. (2010) who evaluated the economic benefits of using hybrid cooling circuits. The authors used heuristics method to find the best combination of the dry and evaporative cooling. Recent work was conducted by Papaefthimiou (2012) who studied the effect of ambient air condition on the thermal performance of the cooling tower. Feng et al. (2005), Ponce-Ortega et al. (2007) and Majozi and Moodley (2008) presented a mathematical technique for the optimization of cooling water systems focusing on cooling water networks. All the above cited work focused on individual components of the cooling water systems thus neglecting the interaction between the cooling towers and the cooling water network.

A more holistic approach was used by Kim and Smith (2001) who developed the graphical technique to debottleneck the cooling water systems with single cooling source. The cooling source was debottlenecked by using the cooling water reuse/recycle philosophy. The authors also derived a cooling tower model to study the interaction of the cooling source and the cooling water network. Panjeshahi and Ataei (2008) extended the work of Kim and Smith (2001) on cooling water system design by incorporating a comprehensive cooling tower model and the ozone treatment for circulating cooling water. Ponce-Ortega et al. (2010) presented a full mathematical model for synthesis of cooling water networks that was based on a stage wise superstructural approach. This work was also limited to cooling water systems with single source. A different approach was taken by Gololo and Majozi (2011) who developed a mathematical technique for cooling water systems with multiple cooling sources. The authors used two different platforms to model the cooling towers and the cooling water network. An iterative procedure was used to link the two platforms thus allowing an interaction between the two components. Although this approach seems holistic, global optimality cannot be guaranteed because two platforms were used to model different components of the cooling water systems. Rubio-Castro et al. (2012) also developed an integrated mathematical technique for synthesis of recirculating cooling water systems consisting of multiple cooling towers. The authors used simple equations to predict the performance of the cooling towers thus compromising the accuracy results of the results.

Although the reuse/recycle philosophy offers a good debottlenecking opportunity, the topology of the associated cooling water network is more complex, therefore prone to higher pressure drop than the conventional parallel design. Debottlenecked cooling water network in which pressure drop is ignored may give misleading information required for sizing circulating water pump. Kim and Smith (2003) attempted to address this challenge by developing a mathematical technique for retrofit design in which the network pressure drop was optimized. The authors used a graphical technique to target the minimum circulating cooling water flowrate and mathematical technique to synthesize a cooling water network. This work was limited to cooling water systems with a single cooling source and the cooling tower model was not included.

Several authors addressed the issue of network pressure drop in other utility systems. Price and Majozi (2010) presented a mathematical technique for pressure drop optimization in steam systems. The authors adopted a similar technique used by Kim and Smith (2003) to minimize steam network pressure drop where condensate reuse philosophy is employed. Hung and Kim (2012) also used similar approach by Kim and Smith (2003) however, their work focused on mass integration problems. The authors synthesized water network that gives the optimal fresh water consumption, location and capacity of pumps.

1.2 Basis and objectives of this study

Existing holistic techniques on synthesis and optimization of complex cooling water systems are limited to the cooling water systems with single cooling source. These debottlenecking techniques employ the cooling water reuse/recycle philosophy thus, requiring additional piping for new streams. The network topology also consists of series and parallel combination of the heat exchangers. Consequently, this could results in an increased cooling water network pressure drop. Thus, it is essential to optimize pressure drop during synthesis of cooling water network where reuse/recycle philosophy is employed. The on-going research in this area is only limited to cooling water networks consisting of single source. Hence, it is imperative to develop a holistic technique which is applicable for both multiple and single source cooling water systems.

The main aim of this study was to develop a holistic methodology for analysis, synthesis and optimization of cooling water systems with multiple cooling towers which takes into account the effect of network pressure drop. The proposed technique which is an extension of the work by Gololo and Majozi (2011), debottleneck the cooling towers by reducing the circulating water flowrate while maintaining the minimum cooling water network pressure drop. Furthermore, the work of Gololo and Majozi (2011) is improved by developing an integrated model for the cooling water systems in a single platform.

Problem statement

The problem addressed in this paper can be stated as follows:

Given,

- a set of cooling towers with their dedicated set of cooling water using operations
- the cooling water using operations with their limiting temperatures and heat duties
- the limiting temperatures for each cooling tower fill
- the dimensions for each cooling tower
- the coefficient of performance correlation for each cooling tower

Determine,

1. the minimum circulating water flowrate for cooling water system with multiple cooling towers.
2. the minimum cooling water network pressure drop for a cooling water system with multiple cooling towers whilst maintaining the minimum amount of circulating cooling water flowrate.

1.3 Report scope

The scope of this research is to develop a mathematical technique for analysis, synthesis and optimization of cooling water systems with multiple cooling water sources. The holistic cooling water systems model consisting of cooling tower model and the cooling water network model is developed. This gives the opportunity to study the interaction between the cooling towers and the cooling water network. The model debottlenecks the cooling towers whilst maintaining minimum cooling water network pressure drop. The critical path algorithm is used to determine the cooling water network pressure drop.

1.4 Report structure

This report is divided into five chapters as follows:

- Chapter 1 introduces the report by giving a detailed description of the cooling water systems and the brief discussion of the latest research in the area. The basis and objectives of this research are then presented followed by the research scope
- Chapter 2 gives a detailed literature study on process integration technique and their applications in utility systems. The main focus will be on cooling water systems. Both graphical and mathematical techniques are presented
- Chapter 3 presents the detailed review on the work done by Gololo and Majozi (2011). This chapter demonstrates the benefit of a comprehensive approach to cooling water system design. This forms basis for the current work
- Chapter 4 presents the development of the mathematical model. Detailed derivations and descriptions of all constraints used in the model are given
- Chapter 5 shows the case study used to illustrate the robustness of the proposed model. The results are also presented and discussed in this section
- Chapter 6 presents conclusions derived from the results
- Chapter 7 presents recommendations for future work

2 LITERATURE REVIEW

2.1 Introduction

Implementation of stringent environmental laws and shortage of resources has forced industries to consider various means of minimizing waste and fresh utilities consumption. Process integration techniques have played a major role in this regard. These techniques can be classified into mass and heat integration. The following section gives a detailed literature review on various mass and heat integration techniques and their applications. The section starts by giving an overview of heat integration techniques followed by mass integration techniques. Both graphical and mathematical techniques are presented and their setbacks are also highlighted. Various techniques which were developed for utility systems will also be presented. The main focus will be on studies concerning synthesis and optimization techniques for cooling water systems.

2.2 Heat integration

Manufacturing and processing industries involve streams which require cooling or heating. In most conversional designs external utilities are used to satisfy all process heating or cooling requirements. Various studies have been conducted to minimize the use of external utilities. In general authors try to integrate streams that require heating (heat sink) with streams that require cooling (heat source). This approach reduces the overall external utility consumption. Linnhoff and coworkers (Linnhoff & Flower, 1978; Linnhoff *et al.*, 1979; Flower & Linnhoff, 1980; Linnhoff & Hindmarsh, 1983; Linnhoff *et al.*, 1982) developed a graphical technique for energy savings. This technique familiar known as pinch analysis, gives the minimum target for cold and hot utility. Linnhoff and Flower (1978) developed a temperature interval method that targets the maximum energy recovery. This method was based on a fixed minimum approach temperature ($\Delta T_{\min}$), i.e. the minimum thermodynamic temperature difference between the cold and the hot stream. For more complicated problems the authors further developed a rigorous approach called problem table method. This method identifies the location of the minimum approach temperature called pinch point. A more simplified graphical targeting procedure can also be used. This procedure starts by plotting the cold stream and hot streams composite curve shown in Figure 1. The curves are then moved closer to each other in a horizontal direction until minimum allowable temperature is reached. The region where the cold and the hot composite curves overlap represents possible integration of the coolers and the heaters. The overshoots at the bottom and top represent the minimum external cooling and heating duties. It is clear at this stage that by minimizing ΔT more energy is recovered and the utilities costs are minimized however, the exchange area is increased. Thus, a balance needs to be reached between the capital and the operational cost.

Linnhoff & Hindmarsh (1983) developed a more rigorous procedure for heat exchanger network synthesis that achieves the target with minimum number of units. The minimum number of units can be calculated from the equation (2-1) introduced by Hohmann (1971).

$$U_{\min} = N - 1 \tag{2-1}$$

This method can also be used for processes consisting of multiple utility levels. The concept of grand composite curve (GCC) based on problem table method was introduced to incorporate multiple utility levels. Figure 2 shows an example of GCC where various levels of steams are used to satisfy the heat requirement.

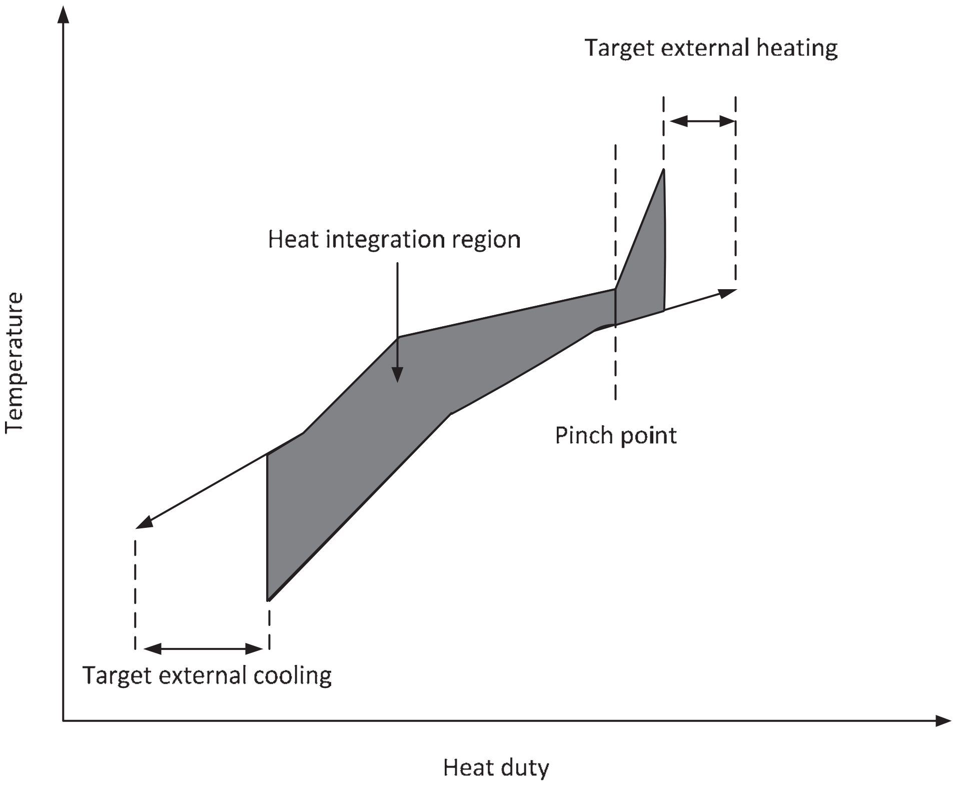

***Figure 1**: Composite curve*

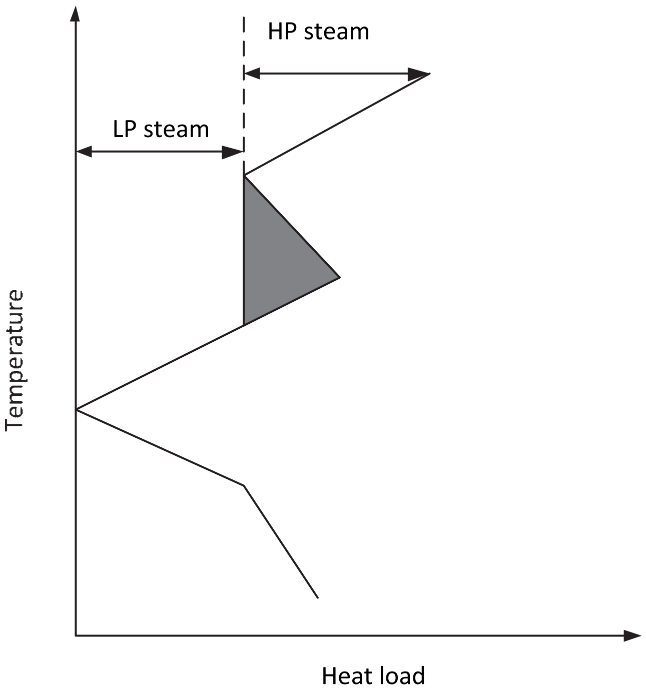

***Figure 2**: Grand composite curve*

Application of pinch analysis technique could be found in both grassroots and retrofit design with the benefits ranging from 6% to 60% for energy savings and 30% for capital savings (Linnhoff, *et* *al.*, 1982). This technique was later adopted for mass exchange problems and water using networks. Although this technique gives plausible results, it does have some limitations. The technique cannot handle additional practical constraints like pressure drop constraints, cost functions or flowsheet layout constraints. The technique can also be time consuming particularly when large practical problems are involved. To overcome these challenges several authors used mathematical optimization techniques to solve heat integration problems.

Yee *et al.* (1990a) presented a mathematical technique for energy and area targeting of the heat exchanger networks. The authors used stage-wise superstructural approach to develop a nonlinear programming (NLP) models for energy and area targeting. Figure 3 shows the superstructure used by Yee *et al.* (1990a). The following two scenarios were considered:

Case A. Simultaneous targeting for energy and area

Case B. Area targeting at a fixed energy target

A two-step approach was used to solve this model. This approach does not guarantee global optimality. Yee and Grossmann (1990) improved the work of Yee *et al.* (1990a) by developing a mixed integer nonlinear programming (MINLP) model that simultaneously targets the heat exchanger area whilst minimizing utility cost and fixed cost for number of units. The setback of this model lies in its complexity. MINLP models are generally more difficult to solve thus global optimality cannot be guaranteed. Yee and Grossmann (1990b) incorporated the process flowsheet to simultaneously optimize the heat exchanger network and the process. In this work the process flowrates and temperatures are not fixed. However, the heat transfer coefficients were assumed to be constant. The work of Yee and Grossmann (1990) and Yee and Grossmann (1990b) also assume fixed location for the utilities.

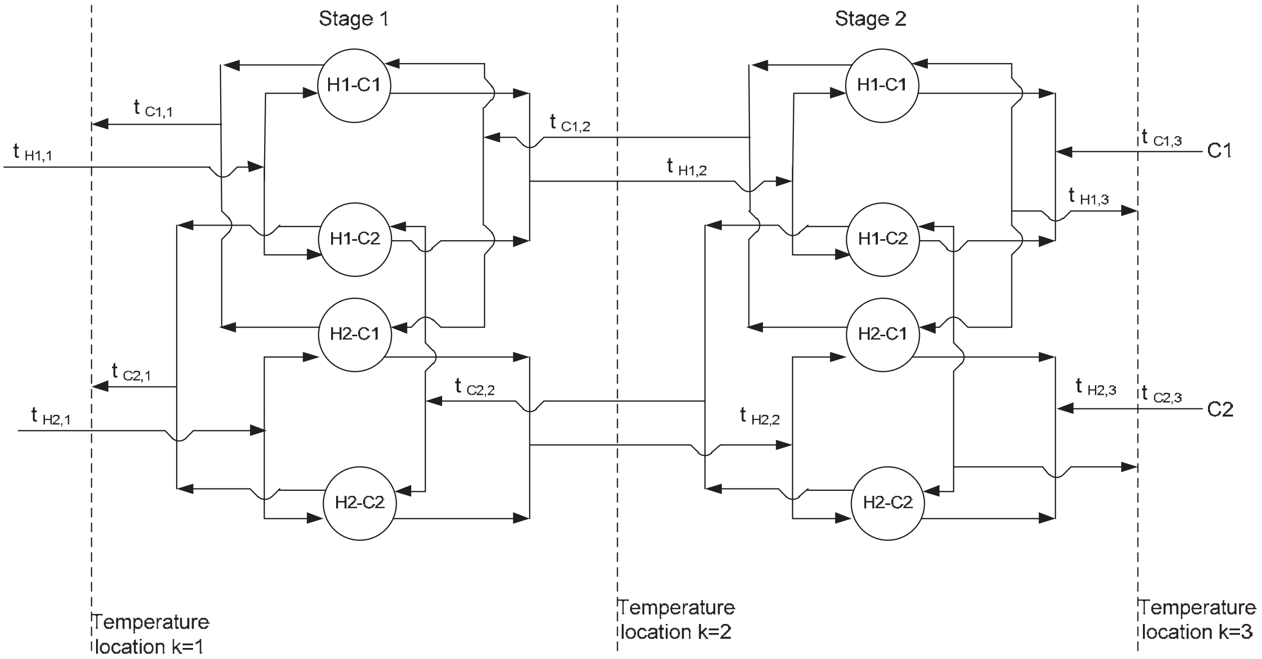

Figure 3*: Stage wise superstructure*

Asante and Zhu (1996) developed heat exchanger network optimization technique by combining both pinch based graphical technique and the mathematical programming technique. This technique offers minimum topology modification thus it can be easily applied in large industrial problems. Although this technique is computationally inexpensive, it does not consider all possible topology modifications. Shenoy *et al.* (1998) took a different approach by looking at heat exchanger network with multiple levels utilities. The authors used pinch analysis to select the cheapest available levels of utilities. Isafiade and Fraser (2008) and Ponce-Ortega *et al.* (2010a) presented a mathematical technique which optimizes the location of heat exchanger network's hot and cold utilities. The authors considered a problem where multiple levels for both cold and hot utilities are available. The problems were formulated as MINLP. Most heat integration techniques assume constant stream heat transfer coefficients or constant pressure drop. This is not entirely true when the stream flowrate is changed. In reality there is always a pressure drop or heat transfer area limit. Polley *et al.* (1990) used expression shown in equation (2-2) to relate the pressure drop and the heat transfer coefficient. The authors minimized the area by varying heat transfer coefficient. In this case stream pressure drop was fixed.

$$\Delta P = KAh^{m} \tag{2-2}$$

Nie and Zhu (1999) looked at the pressure drop for the heat exchanger networks. Given that retrofit design sometimes requires an additional area, the authors identified several options that can be optimized. These options include area distribution, shell arrangement and heat transfer enhancement however, all these options have pressure drop implications. The authors developed the mathematical model that synthesizes the heat exchanger network with the most cost effective options. The authors used the equations (2-3) and (2-4) to calculate the pressure for the heat exchanger. In the equations pressure drop is depended on the velocity of the stream and the pipe dimensions.

$$\begin{gathered} \Delta P = K_{1,t}Av^{2.8} + K_{2,t}v^{2} \\ h = K_{h,t}v^{0.8} \end{gathered} \tag{2-3}$$

$$\begin{gathered} \Delta P = K_{1,s}Av^{1.83} + K_{2,s}v^{2.83} + K_{3,s}v^{3} \\ h = K_{h,t}v^{0.52} \end{gathered} \tag{2-4}$$

Frausto-Hernandez *et al.* (2003) improved the work of Yee and Grossmann (1990b) by incorporating the pressure drop and calculating heat transfer coefficient. The authors used the expression similar to equations (2-3) and (2-4) to predict the heat transfer coefficient given the allowable pressure drop. Recent work by Huang and Chang (2012) shows a further improvement on heat exchanger network synthesis by accounting for the effect of flowrate on heat transfer coefficient and the efficiency for each heat exchanger. Although the authors considered the pressure drop in their models, they only focused on equipment pressure drop not the entire network pressure drop.

2.3 Mass Integration

Using the same principle as described above, pinch analysis was further applied in mass integration and wastewater minimization problems. El-Halwagi and Manousiouthakis (1989) used the principle of pinch analysis to develop a graphical technique for synthesis of mass exchange network. The procedure starts by targeting the minimum cost of external mass separating agents and the minimum number of units. The design with minimum number of units is then synthesized to balance the operating cost and the capital cost. This work was later automated by El-Halwagi and Manousiouthakis (1990) using two-step approach. The authors started by formulating a LP model to target the cost of mass separating agents. The MINLP model was then formulated to minimize the annualized total cost of the network. This work was extended by El-Halwagi and Manousiouthakis (1990) by including regeneration. Wang and Smith (1994a) developed a method for wastewater minimization in water using networks. The authors used principles of pinch analysis to target the minimum wastewater as shown in Figure 4. This technique is known as water pinch. The network that achieves the target was then synthesized using two methods. The first method was based on maximum concentration driving forces and the second method minimizes the number of water sources. This technique can be applied in both single and multi-component problems.

Kuo and Smith (1998; 1998b) improved the targeting method by Wang and Smith (1994a). The authors developed a technique called water main method to improve the network synthesis. The authors also included regeneration. The design was carried out by dividing water using operations into two groups. Group I was supplied by freshwater and Group II was supplied by regenerated water. For further freshwater reduction, operations could be moved from one group to the other.

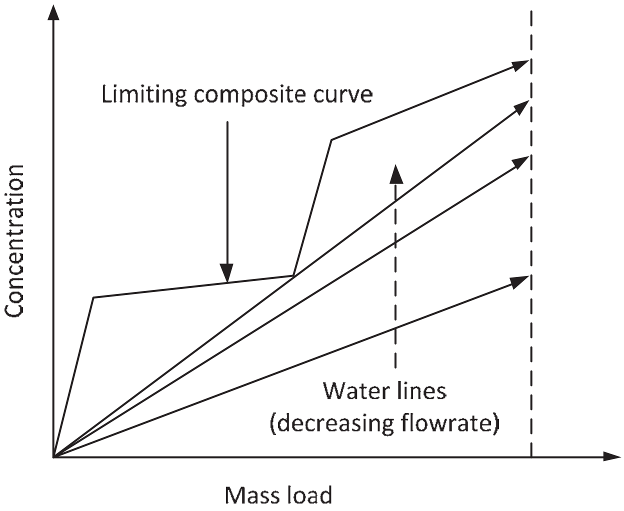

***Figure 4**: Limiting composite curve and water supply line*

Wang and Smith (1994b) applied the principles of pinch analysis in a slightly different environment. They presented a methodology for distributed effluent systems design for both single and multiple contaminants systems. The design procedure starts by plotting the composite curve for all effluent streams. The treatment line was then drawn against the composite curve to target minimum treatment flowrate. The distributed effluent system was then designed using the grouping rule which could be summarized as follows:

- All effluent streams with the concentration above pinch must pass through the treatment process

- All effluent streams with the concentration located at pinch partially bypass the treatment process
- All effluent streams with the concentration below pinch must bypass the treatment process

The setbacks of graphical technique in mass integration problems are similar to those given in heat integration problems.

Lakshmanan and Biegler (1996) developed mathematical technique to solve a more complex mass integration problem. The authors simultaneously synthesized reaction, mixing and separation networks. The model was formulated as MINLP thus global optimality cannot be guaranteed. Bagajewicz *et al.* (1998) and Savulescu *et al.* (2005) attempted a more holistic approach to process synthesis problems by simultaneously solving mass exchanger network and heat exchanger network problem. Bagajewicz *et al.* (1998) used a superstructural approach to develop a mathematical model that minimizes the total annualized cost. Savulescu *et al.* (2005) used a pinch based graphical approach to develop a technique that minimizes fresh water and energy consumption. Chen and Hung (2007) adopted the stage wise superstructural approach by Yee and Grossmann (1990a) to solve heat / mass exchange network problem. The authors solved a MINLP problem by minimizing the total annualized costs. Dong et al. (2008) increased the complexity of water allocation and heat exchanger network by including water treatment in the model. The MINLP model was developed from a state-space superstructure proposed by Bagajewcz et al. (1998). Hung and Kim (2012) solved the problem of water allocation networks by considering pressure drop and pumping arrangements. The authors used critical path algorithm to calculate the overall network pressure. This technique was developed by Kim and Smith (2003) for cooling water systems. In this work the authors assumed constant unit pressure drop thus only piping pressure drop were calculated.

The main challenge with mathematical optimization technique is the solution procedure. Most problems are formulated as NLP or MINLP. Generally, these nonconvex problem are difficult to initialize thus a systematic solution procedure is required. Sherali and Alameddine (1992) proposed reformulation-linearization technique to relax nonconvex bilinear terms. This technique was based on concave over and under estimator by McCormick (1976). The reformulation-linearization technique is not a direct linearization technique thus it does not always gives a feasible solution. Quesada and Grossmann (1995) embedded the reformulation-linearization technique inside branch and bound procedure to obtain global optimal solution. Wicaksono and Karimi (2008) used a piecewise relaxation procedure which was based on partitioning of the search space. Consequently, the relaxed problem is formulated as MILP rather than LP.

Doyle and Smith (1997) proposed an initialization procedure in which the model is linearized by fixing the outlet concentration for all operations. The solution for the linearized model is then used as a starting point for the nonlinear model. Teles et al. (2008) also applied the linearization strategy proposed by Doyle and Smith (1997). Salvelski and Bagajewicz (2000) propose necessary conditions for optimal water using networks with contamination. These conditions are generally optimal values of variables from the model thus one can determine the optimal values of certain variables prior optimization. The authors showed rigorous mathematical proofs to support their claims.

2.4 Utility Systems

The work on heat integration techniques has focused on targeting of external utilities and synthesizing the heat exchanger networks that achieves the target. The utility systems also require systematic synthesis procedure that ensures optimum operation. This section gives a brief overview of steam systems optimization and the detailed review of cooling water systems.

Steam Systems

Coetzee and Majozi (2008) developed a graphical technique based on pinch analysis to minimize the steam flowrate. Using the composite curve in Figure 5 the authors were able to minimize steam flowrate by reusing condensate. The authors used mathematical programming technique to synthesize the steam network that meet the target. The model was formulated as LP problem. The authors further developed a MILP model for simultaneous targeting and synthesis of steam systems. This approach results in a lower return condensate temperature thus compromising the boiler efficiency.

Price and Majozi (2010) recognized the tradeoff between condensate reuse and boiler efficiency. The authors formulated a mathematical model that maintains boiler efficiency at a fixed value whilst compromising the target or fixed target whilst compromising the boiler efficiency. The authors further showed that boiler efficiency can be improved by pre heating the boiler feed water.

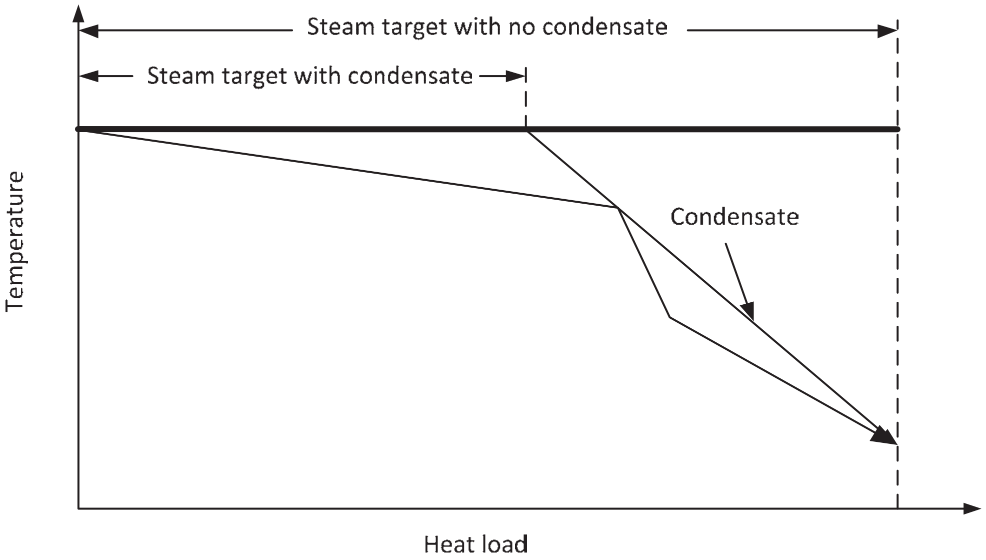

Figure 5*: Targeting using latent and sensible heat*

Price and Majozi (2010) also studied steam systems with multiple pressure levels steam. They further considered pressure drop in the heat exchanger network. The authors used the pressure minimization technique by Kim and Smith (2003) to synthesize the steam network with minimum pressure drop. The model was formulated as MINLP problem.

Cooling Water Systems

2.4.1.1 Cooling Towers

Cooling towers uses the mechanism of evaporative cooling to remove waste heat from the water to the atmosphere. They can be classified into mechanical draft and natural draft cooling towers. The mechanisms of heat transfer takes place in a cooling tower through evaporative cooling and convective heat transfer. Evaporative cooling contributes 62, 5% of the total heat transfers (Khan *et al.,* 2003). This review focuses only on mechanical draft cooling towers

Mechanism of evaporative cooling

When dry air at constant temperature comes into direct contact with water, the water vapor will escape into the air stream. This process is possible when the vapour pressure of water out of the liquid is greater than it is in the unsaturated air. The water vapour carries with it the latent heat of vaporization thus if this energy is not replaced the water temperature will decrease. In a cooling tower this process can continue until the air is saturated or the vapor pressure of water in the liquid is equal to the vapor pressure in the air (Kern, 1950). The nature of evaporative cooling allows the water temperature to drop below the inlet air dry bulb temperature. The limit to which the water temperature can drop is determined by the wet bulb temperature. Several authors derived the mathematical models that describe the thermal behavior of the cooling tower.

Cooling Tower Models

The prediction of cooling tower thermal performance dates back to 1925 by Merkel. Several authors used Merkel's theory to derive a cooling tower model. Bernier (1994) developed a cooling tower model based on a shower type cooling tower. The author derived the model by considering mass and energy characteristics of falling water droplets. The model assumes no packing for the cooling tower, constant Lewis factor and water flowrate. The author further used Merkel's theory to predict the coefficient of performance $\frac{KaV}{m_w}$ as shown in equation (2-5).

$$\frac{KaV}{m_w} = x\left[\frac{m_w}{m_a}\right]^y \tag{2-5}$$

The values of *x* and *y* parameters could be determined experimentally for a given cooling tower packing. The experimental work completed by the author showed a good approximation for $\frac{KaV}{m_w}$. The coefficient of correlation for the regression was in magnitude of 0.99. The author showed that the performance of the cooling tower can be improved by decreasing the inlet water flowrate while increasing the inlet temperature.

Richardson and Coulson (1996) suggested similar correlation as shown below.

$$Ka \quad \alpha \quad {m_w}^{1-n}{m_a}^{n} \tag{2-6}$$

where n varies from 0.4 to 0.8.

Milosavljevic *et al.* (2001) developed a one dimensional mathematical model for a counter flow cooling tower under assumption that the Lewis factor is unity throughout the cooling tower packing. Fisenko *et al.* (2004) and Qi *et al.* (2007) also derived a one dimensional mathematical model for a counter flow mechanical draft cooling tower, however the authors considered heat transfer, mass transfer and dynamic equations of a falling water droplet. Fisenko *et al.* (2004) used equation (2-7) to evaluate the cooling tower efficiency.

$$\eta = \frac{T_{w,in} - T_{w,out}}{T_{w,in} - T_{wb}} \tag{2-7}$$

Khan *et al.* (2004) studied the influence of fouling on cooling tower performance. The authors fouling model predicted 6% loss in effectiveness through fouling. Qureshi and Zubair (2006) also developed a cooling tower model that considers the heat and mass transfer in the spray zone, the packing and the rain zone. The authors also modeled fouling in the cooling tower fill and used equation (2-8) to evaluate the cooling tower effectiveness.

$$\varepsilon = \frac{H_{a,out} - H_{a,in}}{H_{s,w} - H_{a,in}} \tag{2-8}$$

The cooling tower effectiveness was earlier studied by Jaber and Webb (1961) who developed the effectiveness-NTU for cooling towers by adapting the definition of the effectiveness for heat exchangers. The authors considered two scenarios.

Case I. Water capacity rate (m_{cap}) is less than the air capacity rate (m_a). In this case the effectiveness was given as follows:

$$\varepsilon = \frac{m_w c_{pw}(T_{wi} - T_{wo})}{m_{cap}(H_{masi} - H_{mai})} \tag{2-9}$$

Kröger (2004) further showed that the effectiveness can be expressed in terms of enthalpies as shown in equation (2-10).

$$\varepsilon = \frac{H_{masi} - H_{maso}}{H_{masi} - H_{mai}} \tag{2-10}$$

If the exit water temperature is equal to the inlet air wet bulb temperature, then the effectiveness can be expressed as:

$$\varepsilon = \frac{T_{wi} - T_{wo}}{T_{wi} - T_{wb}} \tag{2-11}$$

Case II. Water capacity rate (m_{cap}) is greater than the air capacity rate (m_a).In this case the effectiveness was given as follows:

$$\varepsilon = \frac{m_w c_{pw}(T_{wi} - T_{wo})}{m_a (H_{masi} - \delta - H_{mai})} \tag{2-12}$$

where δ is the correction factor for nonlinearity of $H_{mas} vs \ T_w$ given in equation (2-13).

$$\delta = \frac{H_{maso} + H_{masi} - 2H_{mas}}{4} \tag{2-13}$$

where i_{mas} is the enthalpy of saturated air at mean water temperature.

Kröger (2004) developed a model for a cooling tower by considering a control volume as shown Figure 6.

The following assumptions were made:

- Interface water temperature is the same as the bulk temperature
- Air and water properties are the same at any horizontal cross section
- Heat and mass transfer area is identical

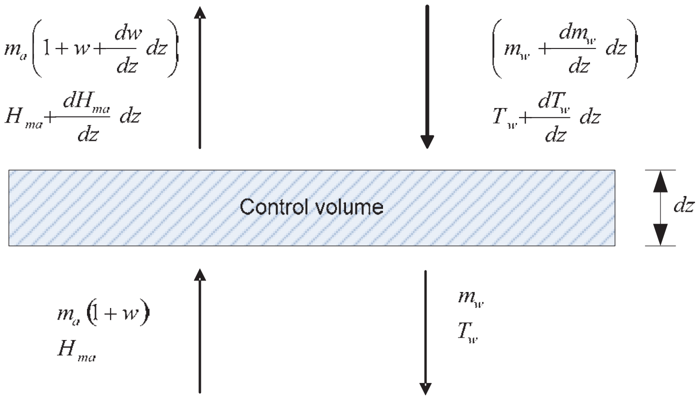
$m_a\left(1 + w + \frac{dw}{dz} dz\right)$
$H_{ma} + \frac{dH_{ma}}{dz} dz$
$\left(m_w + \frac{dm_w}{dz} dz\right)$
$T_w + \frac{dT_w}{dz} dz$
Control volume
dz
$m_a(1 + w)$
H_{ma}
m_w
T_w

***Figure 6**: Control volume*

The governing equations that predict the thermal performance of a cooling tower are given by equations (2-14), (2-15) and (2-16). Equations (2-14) and (2-15) define the mass and energy balance for the control volume, respectively. Equation (2-16) defines the air enthalpy change for the control volume.

$$\frac{dm_w}{dz} = m_a \frac{dw}{dz} \tag{2-14}$$

$$\frac{dT_w}{dz} = \frac{m_a}{c_{pw} m_w}\left(\frac{1}{c_{pw}} \frac{dH_a}{dz} - T_w \frac{dw}{dz}\right) \tag{2-15}$$

$$\frac{dH_a}{dz} = \frac{Ka_{fi} A_{fi}}{m_a}\left(Le_f (H_{as} - H_a) + (1 - Le_f) H_v (w_s - w)\right) \tag{2-16}$$

In equation (2-16) a_{fi} is the wetted area divided by the corresponding volume of the fill and A_{fr} is a frontal area. The Lewis factor, Le_f appearing in equation (2-16) is the relationship between the heat transfer coefficient and the mass transfer coefficient, i.e. $\frac{h}{Kc_{pma}} = Le_f$. Lewis factor appears in many governing heat and mass transfer equations. A number of authors such as Bernier (2004) and Milosavljevic *et al.* (2001) assumed the Lewis factor to be unity. Klopper and Kröger (2005) used expression given in equation (2-17) to predict the value of Lewis factor. The authors studied the influence of Lewis factor on the performance prediction of a wet cooling tower. Their findings were that the influence of Lewis factor diminishes when the inlet ambient air is relatively hot and humid.

$$Le_f = 0.866^{0.667}\left(\frac{w_s + 0.622}{w + 0.622} - 1\right) / \ln\left(\frac{w_s + 0.622}{w + 0.622}\right) \qquad (2\text{-}17)$$

They further elaborated that increasing Lewis factor increases heat rejection, decreases water outlet temperature and decreases water evaporation rate.

Cooling Tower Performance and Optimization

Bernier (1994), Kim and Smith (2001), Lemouri *et al.* (2007) showed that the cooling tower effectiveness can be improved by increasing the inlet temperature whilst decreasing the inlet flowrate. Kim and Smith (2001) used Figure 7 to emphasize this observation.

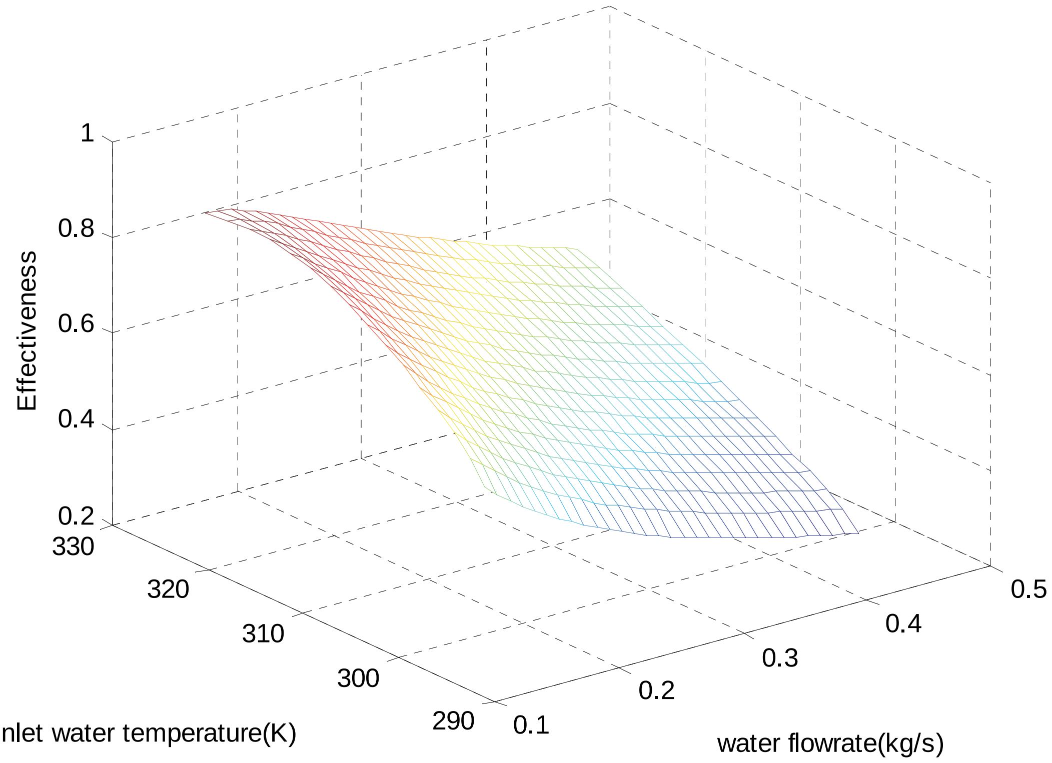

***Figure 7:** Cooling tower performance*

Khan *et al.* (2003), Naphon (2005), Gharagheizi *et al.* (2007) and Lemouari *et al.* (2007), Yingjian *et al.* (2011) studied the influence of water to air ratio on the cooling tower performance. The authors showed that higher water to air ratio decreases the thermal performance of the cooling tower. This suggests that it is better to operate the cooling tower with higher air flowrate.

Khan and Zubair (2001) showed that increasing the inlet wet bulb temperature decreases the performance of a cooling tower. These results were confirmed by Khan *et al.* (2004) who showed that an increase in atmospheric pressure results in a decrease in cooling tower effectiveness. Papaefthimiou *et al.* (2012) studied the influence of ambient condition on cooling tower performance. The authors suggested that optimum thermal performance of the cooling tower is achieved when the inlet air has low humidity.

Although it is important and necessary to study the thermal performance the cooling tower, its performance is influence by the cooling water network. Thus, it is imperative to study the cooling water system using a more holistic approach.

2.4.1.2 Optimization of Cooling Water Systems

Castro *et al.* (2000) developed a mathematical technique for synthesize of cooling water systems with minimum operating cost. The model consists of the cooling water network, pumping system and the cooling tower. The model also takes into account the piping and heat exchanger pressure drops. The authors used a simplified equation to predict the outlet conditions of the cooling tower. The model was applied to a case study to evaluate the effect of cooling tower outlet temperature and the climatic conditions. The highest cost was observed during a month with highest humidity and lower temperatures. The authors did not analyze the performance of the cooling tower or the cooling tower efficiency or effectiveness.

Cortinovis *et al.* (2009a) also developed a mathematical model for cooling water systems which takes into account the cooling tower, pumping and cooling water network. The authors developed a cooling tower model to predict the outlet conditions of the cooling tower. The Lewis factor was assumed to be unity across the cooling tower packing. This model also neglect evaporation thus, blowdown and makeup were ignored. The model was validated using experimental data. This was later improved by Cortinovis *et al*. (2009b) who incorporated makeup and blowdown by calculating the evaporation using equation (2-18). This equation was derived by assuming that the heat transfer between the air and water is through evaporation only. The authors optimized the cooling water system by minimizing the total cost.

$$Evaporation = 0.0008F_w\left(T_{w,in} - T_{w,out}\right) \tag{2-18}$$

where F_w is the circulating cooling water flowrate, $T_{w,in}$ and $T_{w,out}$ are inlet and outlet cooling tower temperatures respectively.

The model was applied in a case study to evaluate the effect of change in heat load, makeup water temperature and the air temperature on cost. The results showed that an increase in heat load of the process requires optimum performance of the cooling tower.

Kim & Smith (2001) showed that by exploring the opportunities for cooling water reuse, the circulating water flowrate can be reduced, thus debottlenecking the cooling tower. However, this option was not considered by the authors.

Several authors also showed that the cooling water systems can be debottlenecked by employing the cooling water reuse-recycle philosophy. This approach reduces the circulating water flowrate thus increasing the cooling source return temperature. Bernier (1994) suggested that the cooling tower effectiveness can be improved by minimizing the inlet flowrate whilst increasing the inlet temperature. Kim and Smith (2001) developed a methodology for grassroots design of a cooling water system with one cooling source taking into account the cooling tower performance. The authors used graphical technique to target the minimum circulating water flowrate as shown in Figure 8. The cooling tower and cooling water network were examined discretely. The cooling tower model was developed to study the interaction of the cooling tower with the cooling water network.

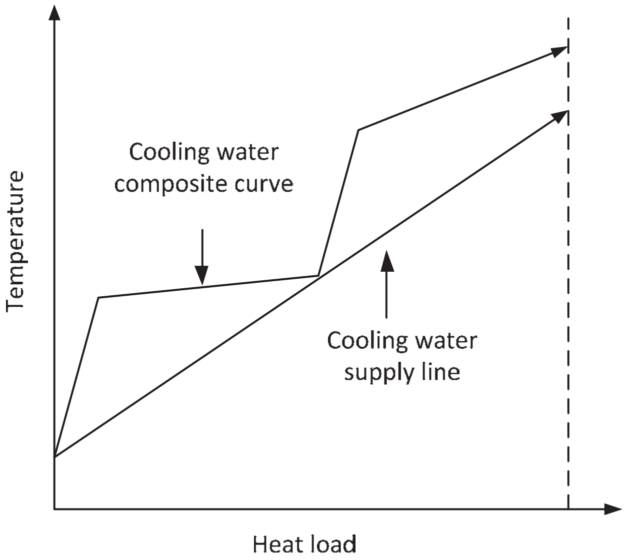

***Figure 8**: Targeting for cooling water systems*

The cooling water network was synthesized by the procedure proposed by Kuo and Smith (1998a). The water mains method by Kuo and Smith (1998a) is only applicable for cases were maximum reuse is allowed, i.e. the heat exchanger outlet temperature was allowed to go as high as possible. In most practical situations the return water temperature is constrained. The cooling water supply line for these situations does not make a pinch with the composite curve, which implies that the water mains method cannot be readily applied. The concept of pinch migration and temperature shift was introduced to handle problems where process pinch does not exist.

Kim and Smith (2003) extended their previous work of 2001 by incorporating pressure drop into their design. The authors presented a paper on retrofit design of cooling water systems using both graphical and mathematical programming techniques. The authors used graphical technique to target the minimum circulating water flowrate and mathematical technique to design a cooling water network.

The pressure drop of the network was calculated using critical path algorithm (CPA) technique. The total pressure drop includes the piping and equipment pressure drops. Equations (2-19) and (2-22) were used to calculate piping and heat exchanger pressure drop respectively. The author used the superstructure in Figure 9 to formulate the network pressure drop problem. The model was formulated as MINLP problem. This problem was solved in GAMS platform using OSL solver for MILP subproblems and DICOP++ for MINLP. The authors did not use any cooling tower model to predict the outlet cooling water conditions. They assume constant outlet cooling water conditions although the inlet conditions are changed. This work was also limited to one cooling source.

$$\Delta P(i) = N_{t1} V_t^{1.8} + N_{t2} V_{it}^{\ 2} \quad \forall i \in I \tag{2-19}$$

where $$N_{t1} = \frac{1.115567 \rho^{0.8} \mu^{0.8} n_{tp}^{\ 2.8} A}{\pi^{2.8} N_t^{2.8} d_o d_i^{4.8}} \tag{2-20}$$

$$N_{t2} = \frac{20 n_{tp}^{\ 3} \rho}{\pi^2 N_t^2 d_i^4} \tag{2-21}$$

$$\Delta P = N_p \frac{1}{V_p^{0.36}} \tag{2-22}$$

where V is the volumetric flowrate of any stream.

and $$N_p = \frac{188.318 \rho^{0.176} \mu^{0.2} L}{\pi^{1.8}} \tag{2-23}$$

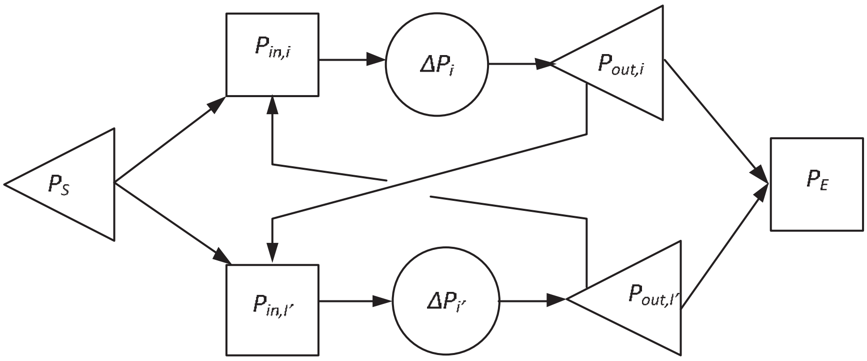

***Figure 9**: Pressure drop superstructure*

Kim and Smith (2004) further presented a systematic method for the design of cooling water system to reduce makeup water. The makeup water reduction was achieved by reusing water from wastewater generating processes. The authors assumed that wastewater was available at allowable contaminant concentration and with unlimited quantities. This approach is only applicable for processes where wastewater is available with minimum salt loading because high salt loading can exacerbate fouling on the cooling tower packing or heat exchangers surfaces.

Feng *et al.* (2005) used a superstructure shown in Figure 10 to synthesis the cooling water network with minimum circulating water flowrate. The superstructure was divided into two sections consisting of three mains, namely cooling water supply main, intermediate cooling water main and cooling water return main. Cooling water using operations can get water from any water main and return to any as long as the thermodynamic constraints are met. The model was formulated as MINLP where the objective function was minimization of circulating water flowrate. This problem was solved in LINGO platform. Although the authors claim that network topology is simpler when using this approach, the cooling water reuse opportunities are not fully explored. The authors further neglect the impact of changing cooling tower return flowrate and temperature.

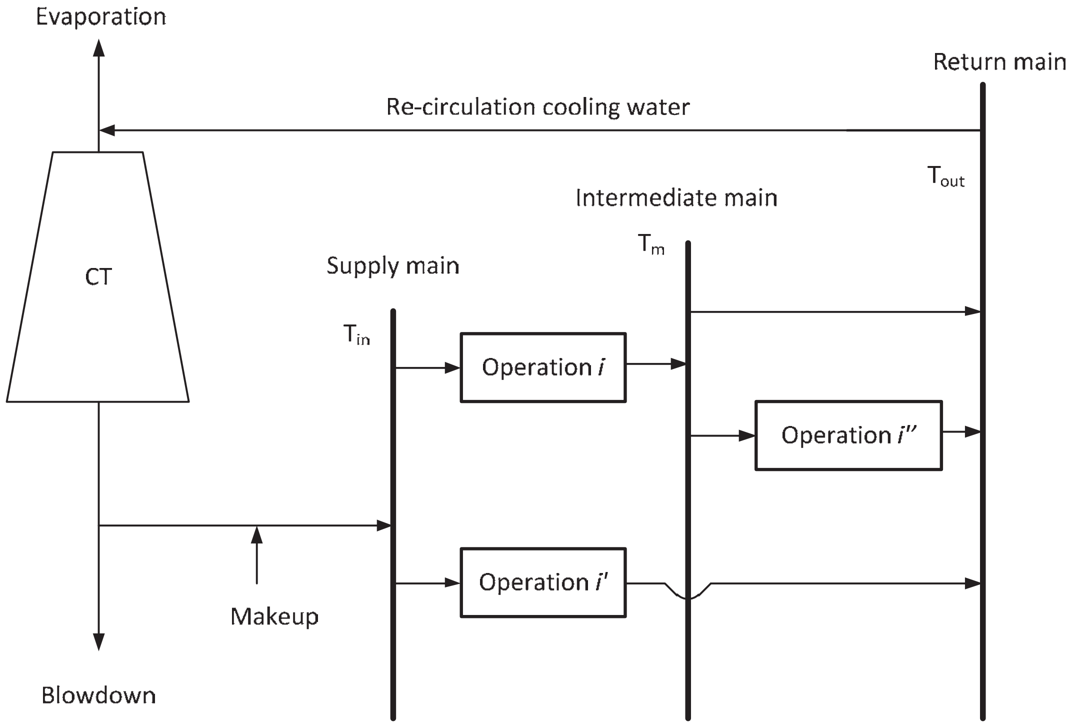

***Figure 10**: Superstructure for cooling water network with water mains*

Ponce-Ortega *et al.* (2007) presented a methodology for synthesis of cooling water networks with minimum cost. This method is based on a stage wise superstructural approach proposed by Yee *et al.* (1990a). However, in this work the outlet temperature for each cooler is a variable. The superstructure which is shown in Figure 11 allows for all possible cooling water reuse. Their mathematical model was formulated as mixed integer nonlinear programming. The model was solved GAMS platform using DICOPT solver. This approach does not take into account the cooling tower performance. The case study showed an increase in circulating water flowrate with a decrease in return temperature. This has a potential of compromising the cooling tower performance. Ponce-Ortega *et al.* (2010b) improved this work by incorporating the cooling tower model. The model also gives a detailed design for coolers and the cooling tower. However, the cooling tower model was based on simple equation thus, compromising the accuracy of the results. The model was solved using DICOPT in a GAMS platform. CPLEX solver was used for MIP subproblems and CONOPT solver was used for NLP subproblems. This method was limited to cooling water systems with single cooling source.

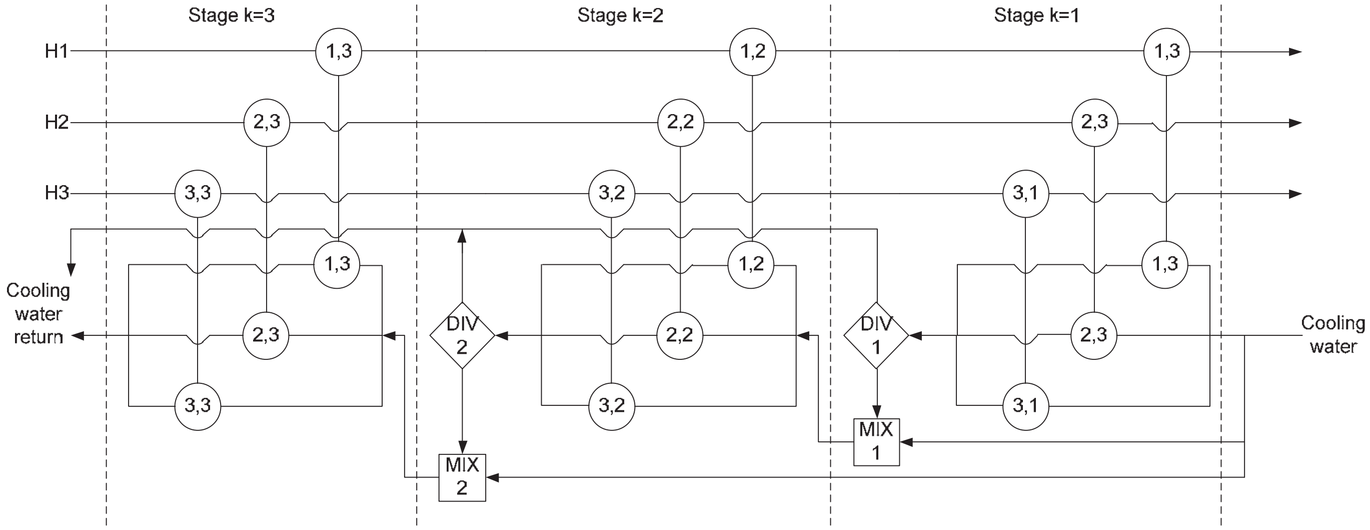

Figure 11*: Superstructure for cooling water network with three hot process streams*

Majozi and Nyathi (2007) derived a methodology for synthesis of cooling water system consisting of multiple cooling sources by combining graphical approach and mathematical programming. The authors used graphical technique to target the minimum cooling water flowrate and superstructural approach to synthesize the cooling water network that achieves the target. The mathematical model was developed based on the following two practical scenarios:

Case A. An unspecified cooling tower inlet temperature without a dedicated source and sink for any operation using the water. In this case cooling water using operation can receive water from any cooling tower and return to any cooling tower.

Case B. An unspecified cooling tower inlet temperature with dedicated source and sink. This means cooling water using operation can only receive water from one cooling tower and return water to the same cooling tower. However, the cooling water reuse between operations is still allowed.

The formulation in Case A was NLP whilst Case B was MINLP. The authors used the linearization technique by Savelski and Bagajewicz (2000) to linearize their model. Savelski and Bagajewicz (2000) stated that the condition for optimality exists when the outlet concentration is at its maximum allowable level. If this condition is satisfied, the water flowrate will be at its minimum. In the case of heat exchanger networks, Majozi and Nyathi (2007) demonstrated that the optimal solution exists if the outlet temperature is at its maximum allowable level. The cooling tower inlet temperature is normally controlled at a particular limit to minimize fouling however, in this work this limit was ignored. Majozi and Moodley (2008) improved the work of Majozi and Nyathi (2007) by considering the following two additional cases which takes into account the maximum limit of the cooling tower inlet temperature.

Case C. A specified cooling tower inlet temperature without a dedicated source and sink for any operation using the water. In this case cooling water using operation can receive water from any cooling tower and return to any cooling tower.

Case D. A specified cooling tower inlet temperature with dedicated source and sink. This means cooling water using operation can only receive water from one cooling tower and return water to the same cooling tower. However, the cooling water reuse between operations is still allowed.

Case C was formulated as NLP whilst Case D was formulated as MINLP. The authors used the reformulation-linearization technique by Sherali and Alameddine (1992) to relax their model. These problems were solved in GAMS platform. NLP subproblems where solved using CONOPT and MINLP subproblems where solved using DICOPT solver where CPLEX was used for MILP subproblems and CONOPT was used for NLP subproblems. The drawback of the work by Majozi and Nyathi (2007) and Majozi and Moodley (2008) was the exclusion of cooling tower model. Changing the cooling water return conditions (temperature and flowrate) will affect the thermal performance of a cooling tower thus optimization of cooling system without the cooling tower model will not yield true optimality.

Panjeshahi and Ataei (2008) extended the work of Kim and Smith (2001) exploring all feasible inlet cooling tower temperature above the approach. Figure 12 shows the actual feasible region above the approach temperature. The authors further incorporated ozone treatment to minimize makeup water flowrate.

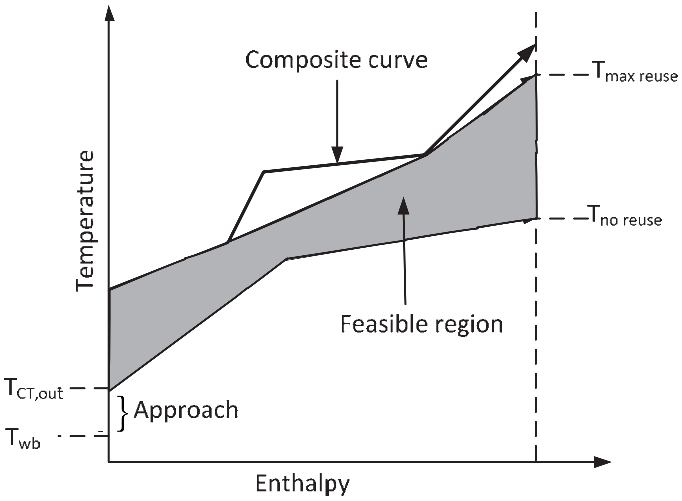

***Figure 12**: Identification of optimization constraints*

The inclusion of ozone treatment allows the authors to operate at an increased cycles of concentration, which resulted in a significant reduction in blowdown. Figure 13 shows change the effect of cycles of concentration on blowdown. The authors compared their proposed method with Kim and Smith (2001). Kim and Smith (2001) targeting was based on fixed cooling tower outlet temperature while in Panjeshahi and Ataei (2008) method the cooling tower outlet temperature was allowed to reach minimum approach temperature. The cycles of concentration in Kim and Smith (2001) design was 3, which resulted in higher blowdown as compared cycles of concentration of 15 proposed by Panjeshahi and Ataei (2008).

Panjeshahi *et al.* (2009) and Ataei *et al.* (2010) used similar design methods proposed by Panjeshahi and Ataei (2008), however Ataei *et al.* (2010) used the cooling tower model which accounted for rain zone and spray zone while Panjeshahi *et al.* (2009) applied pinch migration technique to solve problems where the cooling water line does not form pinch with the composite curve.

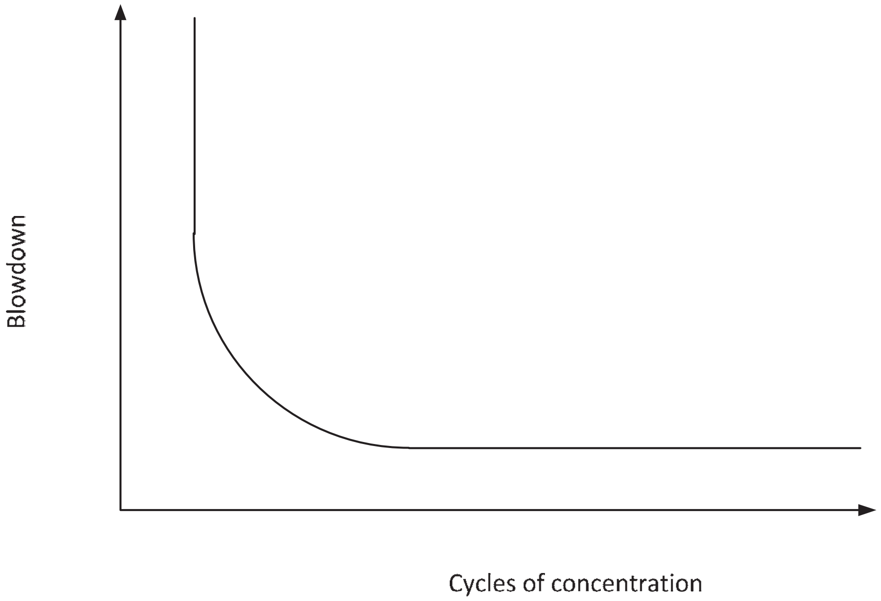

Figure 13*: Identification of optimization constraints*

A different approach was taken by Ataei *et al.* (2009a). They developed a technique for grassroots design of cooling water systems by incorporating regeneration. In this context regenerator is an air cooler thus waste heat from the process is dissipated into the atmosphere by the cooling tower and the air cooler. The authors adopted the proposed grouping technique by Wang and Smith (1994a) for mass integration problems to target the minimum circulating cooling water flowrate. Figure 14 shows the graphical targeting technique with regeneration. An introduction of air cooler result in a significant reduction in circulating cooling water flowrate however, there are cost implications associated with the operation and installation of this equipment.

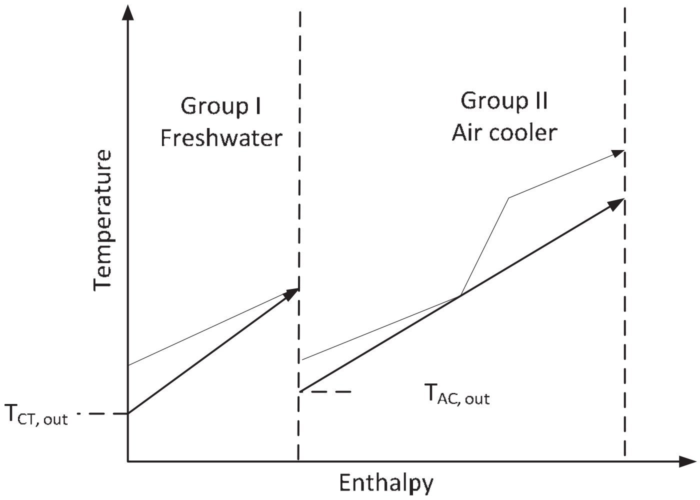

Figure 14*: Targeting with regeneration*

The work of Panjeshahi and Ataei (2008), Panjeshahi *et al.* (2009), Ataei *et al.* (2010) and Ataei *et al.* (2009a) was limited to the cooling water systems consisting of a single cooling tower while in practice there are many cases with multiple cooling towers.

Gololo and Majozi (2011) improved the method developed by Majozi and Moodley (2008) by adding the detailed cooling tower model to study the interaction of the cooling water network and the cooling towers. The authors used two different platforms to model the cooling towers and the cooling water network. An iterative procedure was used to link the two platforms thus allowing an interaction between the two components. Although this approach seems holistic, global optimality cannot be guaranteed because two platforms were used to model the overall different components of the cooling water systems. Rubio-Castro *et al.* (2012) also developed an integrated mathematical technique for synthesis of re-circulating cooling water systems consisting of multiple cooling towers. The authors used a modified superstructure proposed by Ponce-Ortega *et al.* (2007). In this work the superstructure allows for parallel and series combinations of cooling towers. The model was formulated as MINLP and the objective function was minimization of the total cost. This problem was solved using GAMS/DICOPT solver. There are some few setbacks with this approach.

- The authors used simple equations to predict the performance of the cooling towers thus compromising the results accuracy

- The superstructure allows for series connections of cooling towers. This means a second cooling tower will receive water a very low temperature thus compromising the effectiveness

- The maximum cooler outlet temperatures were based on the cooling tower packing maximum temperature. The cooler outlet temperature can be as high as possible as long as the thermodynamic constraint is not violated. Better results can be achieved by imposing tower packing temperature limit on the mixer feeding the cooling tower

- The blowdown charges were ignored thus resulting in a less environmentally friendly design

Picón-Núñez et *al.* (2007) studied the impact of network arrangement on the total heat transfer area. The authors analysed four network arrangements namely, parallel, series, series-parallel (for vertical heat transfer) and series-parallel (for minimum flowrate) compared the cost of various network arrangements. The results showed that series-parallel arrangement (for minimum water flowrate) exhibits the largest heat transfer area. This was mainly because the heat exchangers are operated at minimum approach temperature. The lowest cost was on a series arrangement. This arrangement allows for total cooling water reuse from one heat exchanger to the other. It is very rare to apply this approach in practice without violating the minimum approach temperature constraint.

Lee *et al.* (2013) developed a mathematical technique for synthesis and design of chilled water network. This superstructural approach was based on direct and indirect integration. The direct integration involves reuse/recycle of chilled water directly from one operation to the other. However, in the indirect integration the direct transfer of chilled water between operations is forbidden. The chilled water reuse/recycle takes place via intermediate main. The purpose of the intermediate main was to improve the network structure as explained by Feng *et al.* (2005). This work did not take into account the impact of chilled water network on the performance of the chiller.

Shenoy and Shenoy (2013) developed a methodology for targeting and design of cooling water networks. The technique called unified targeting algorithm was used to target the minimum flowrate and the cooling water network was synthesized using nearest neighbour algorithm. The authors illustrated that this method can be applied to problem with or without the cooling tower return temperature limit. This work was limited to the cooling water systems with single cooling source. The cooling tower performance was also ignored.

2.4.1.3 Optimization of Effluent Thermal Treatment Cooling Water Systems

Effluent cooling water systems are used to cool waste streams before being discharged to the environment. Generally waste streams are mixed and sent to the cooling source as shown in Figure 15(a). This approach results in high flowrate to the cooling source and has a potential to compromise the cooling source thermal performance. Kim *et al.* (2001) proposed a graphical technique to debottleneck the effluent cooling water systems. The authors adopted the methodology introduced by Wang and Smith (1994b) and Kuo and Smith (1997) on synthesis of distributed wastewater treatment systems. This technique involves targeting the minimum flowrate to the cooling source and synthesizing the network that achieves the target. Grouping strategy was then used to synthesis the distributed cooling water system. The grouping was carried out as follows: Effluent streams with temperatures above pinch belong to Group I. Those streams at a pinch temperature belong to Group II and streams with temperatures below pinch belong to Group III. Only Group I streams are cooled and Group III streams bypass the cooling source. Stream at pinch temperature are partially cooled. The bypass and cooled streams are the mixed to achieve the required temperature as shown in Figure 15(b). This approach assumes fixed cooling tower performance although the inlet water flowrate and temperature are changed.

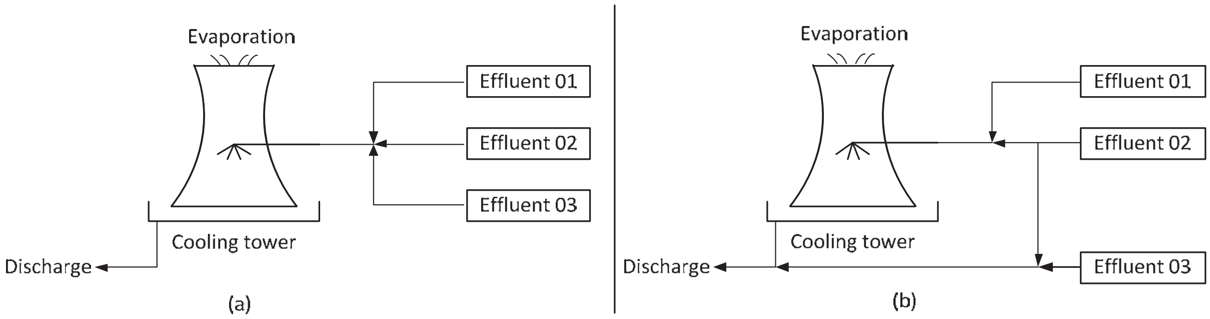

***Figure 15**: Effluent cooling water systems*

Ataei *et al.* (2009b) extended this work by including a detailed cooling tower model. The evaporative effect on the cooling tower supply flowrate was also considered, which resulted in a cooling water line without pinch. However, the grouping rule was based on problems with pinch, thus the concept of pinch migration developed by Kim & Smith (2001) was used to modify the composite curve to form a new pinch. The grouping rule was then used to design the distributed cooling system.

Rubio-Castro *et al.* (2010) used a mathematical programming technique to develop a comprehensive model that synthesizes the effluent cooling systems at minimum cost. The model was formulated as MINLP. The cooling tower model was used to predict the outlet conditions of the cooling tower. The model was solved in a GAMS platform using DICOPT solver. CONOPT was used for NLP subproblems and CPLEX was used for MILP subproblems. The cooling tower model used in this methodology is based on simplified equation. The model is does not used detailed differential equation thus, compromising the accuracy this method.

The presented research work on effluent cooling water systems is limited to systems consisting of one cooling source. In practice effluent cooling water systems may be designed with multiple cooling sources with their dedicated cooling water network. The work of Gololo and Majozi (2011) on optimization of cooling water systems with multiple cooling sources can be adopted for complex effluent cooling water systems. The superstructure proposed by Kuo and Smith (1997) for effluent waste water treatment can be adopted for effluent cooling water systems as shown in Figure 16. This would further reduce the total water flowrate to the cooling sources.

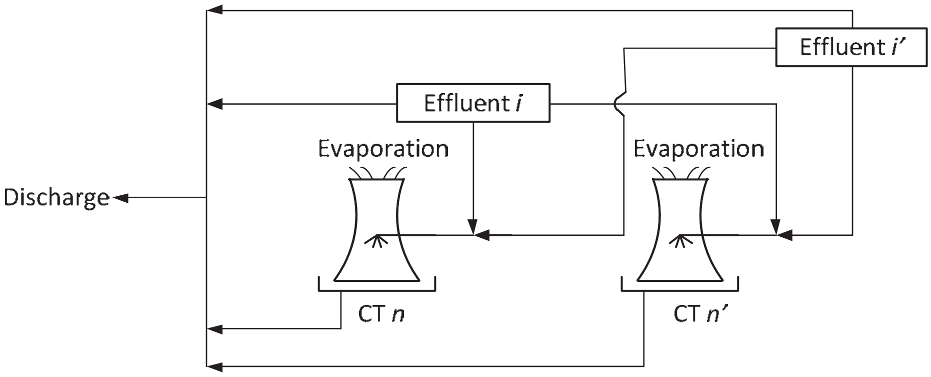

Figure 16*: Superstructure for effluent cooling water systems*

2.5 Conclusions

Process integration techniques for heat and mass integration problems have been presented. In heat integration problems the main aim is to minimize cold and hot external utilities whilst in mass integration the main aim is to minimize wastewater or freshwater consumption. These techniques can be classified into graphical and mathematical programming technique. The most common graphical technique is pinch analysis for heat integration problems. This technique was adapted for mass integrations problems. Although, this technique shows a potential for energy and cost savings, it does not offer more flexibility in practical applications. The technique cannot handle the cost functions, pressure drop, forbidden streams matching, etc. On the other hand mathematical technique offers more flexibility hence more practical constraints can be incorporated.

The main setback with mathematical programming technique is the formulation structure. Most practical problems are formulated as MINLP/NLP models. Generally, these types of formulation are difficult to solve. The authors are forced to derive solution procedures that will ensure a global optimal solution.

A detailed review on cooling water systems was also presented. Most published research in this area has focused mainly on optimization of individual components, i.e. optimization of heat exchanger network or optimization of the cooling towers. This approach does not optimize the cooling water system as a whole. Several attempts have been made to holistically optimize the cooling water systems. In most of these attempts the model for the cooling source and the model for cooling water network are built in different platforms. Although authors try to relate the two platforms in the solution procedure, global optimality is compromised. There is also insufficient research work in literature on cooling water network with multiple cooling towers.

Another setback with the available techniques is that optimization and synthesis cooling water systems involves the exploitation of recycle/reuse opportunities to debottleneck the cooling sources. This approach results in an additional streams being added to the network. The network also consists of series and parallel combinations of cooling water using operations. This network topology has a potential for higher pressure drop relative to the conventional parallel combinations. Kim and Smith (2003) attempted to address this problem however their work was limited to the cooling water systems with single source.

3 BACKGROUND ON COMPLEX COOLING WATER SYSTEMS

3.1 Introduction

This chapter presents a review on the work done by Gololo and Majozi (2011) on synthesis and optimization of cooling water systems consisting of multiple cooling towers. This work forms basis and motivation for the current work. Gololo and Majozi (2011) developed a comprehensive cooling water system model consisting of the cooling tower model and the cooling water network model. Their work accounts for the interaction between the cooling towers and the cooling water network.

3.2 Cooling water system model

Cooling water systems consist of cooling water network and cooling towers. There is a strong interaction between the two components thus their performances are related. Bernier (1994) showed that the cooling tower coefficient of performance can be improved by increasing the cooling tower inlet temperature while decreasing the cooling tower inlet flowrate. Hence, the comprehensive model developed by Gololo and Majozi (2011) included both cooling water network and cooling towers.

Cooling water network model

The cooling water network model was based on a superstructure in which all possible cooling water reuse were explored. The cooling water reuse was only possible in the case where the limiting heat exchanger inlet temperature is above the cooling water supply temperature. Therefore, if the outlet temperature of the cooling water from one heat exchanger is at least $\Delta T_{\min}$ lower than the process temperature in any heat exchanger, it could be reused to supply that heat exchanger. The cooling water reuse philosophy results in cooling water network with series-parallel combination of heat exchangers. This renders the system more flexible compared to the traditional parallel arrangement.

The model for cooling water network was developed considering the following two practical cases:

Case I. Any cooling tower can supply any cooling water using operation whilst the cooling water using operation can return to any cooling tower.

Case II. This is similar to Case I except that the geographic constrains were taken into account. A particular cooling tower can only supply a particular set of cooling water using operations and these cooling water using operations can only return water to the same supplier.

The optimum cooling water network was synthesized by minimizing the total cooling tower inlet flowrates. The model was solved using GAMS platform.

Mathematical formulation

The mathematical formulation entails the following sets, parameters and constraints:

Sets:

$i = \{ i \mid i$ is a cooling water using operation$\}$

n = { n |n is a cooling tower}

Parameters:

$Q(i)$ Duty of cooling water using operation $i(kW)$

$T_{ctout}(n)$ Cooling water supply temperature from cooling tower $n(^oC)$

$OS^n(n)$ Maximum design capacity of cooling tower $n(kg/s)$

$T_{out}^U(i)$ Limiting outlet temperature of cooling water using operation $i(^oC)$

$T_{in}^U(i)$ Limiting inlet temperature of cooling water using operation $i(^oC)$

$F_{in}^U(i)$ Maximum inlet flowrate of cooling water using operation $i(kg/s)$

$T_{ret}^U(n)$ Limiting inlet temperature of cooling water using operation $n(^oC)$

$B(n)$ Blowdown flowrate for cooling tower $n(kg/s)$

$M(n)$ Makeup flowrate for cooling tower $n(kg/s)$

$E(n)$ Blowdown flowrate for cooling tower $n(kg/s)$

c_p Specific heat capacity of water 4.2 (kJ/kg^oC).

T_{amb} Ambient temperature (^oC)

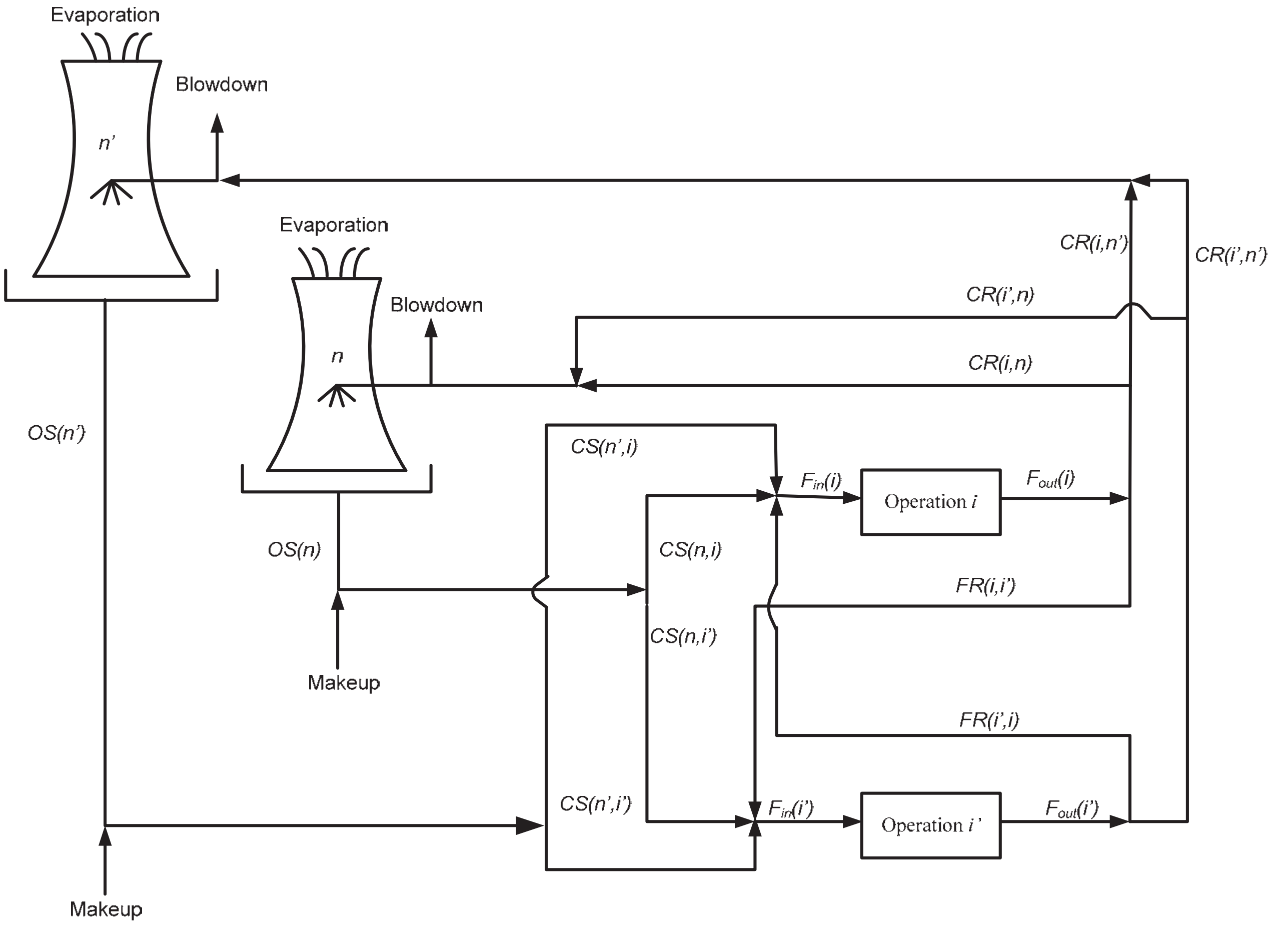

Figure 17: *Superstructure for a cooling system*

Continuous variables:

$OS(n)$	Operating capacity of cooling tower $n(kg / s)$
CW	Overall cooling water supply from all cooling tower (kg / s)
$CS(n,i)$	Cooling water supply from cooling tower *n* to cooling water using operating
$CR(i,n)$	Return cooling water to cooling tower *n* from cooling water using operating
$FR(i',i)$	Reuse cooling water to cooling water using operating *i'* from cooling water using operating $i(kg / s)$
$F_{in}(i)$	Total cooling water into cooling water using operating $i(kg / s)$
$F_{out}(i)$	Total cooling water from cooling water using operating $i(kg / s)$
$T_{in}(i)$	Inlet temperature of cooling water to cooling water using operation $i(^{o}C)$
$T_{out}(i)$	Outlet temperature of cooling water to cooling water using operation $i(^{o}C)$
$T_{st}(n)$	Cooling water supply temperature form cooling tower *n* after adding makeup (^{o}C)
$crt(i,n)$	Linearization variable for relaxation technique
$frt(i',i)$	Linearization variable for relaxation technique
$fnt(i)$	Linearization variable for relaxation technique
$tcs(n,i)$	Linearization variable for relaxation technique

The mathematical optimization formulation was developed from the superstructure given in Figure 17 by considering energy and mass balance equations across each cooling water using operation and at each node. Two cases that were considered are given in the following sections.

3.2.1.1 Case I

In this case there is no dedicated source or sink for any cooling water using operation. The water using operation can be supplied by one or more cooling towers. The maximum cooling water return temperatures to the cooling towers are also specified. This situation arises when packing material inside the cooling tower is sensitive to temperature and any cooling tower can supply any water using operation and the water using operation can return to any cooling tower.

Constraint (3-1) stipulates that the total cooling water is made up of cooling water from all cooling towers:

$$CW = \sum_{n \in N} OS(n) \tag{3-1}$$

Constraints (3-2) and (3-3) ensure that the inlet and outlet cooling water flowrates for any cooling tower are equal:

Mass balance at each node

$$OS(n) = \sum_{i \in I} CS(n,i) - M(n) \qquad \forall n \in N \tag{3-2}$$

$$OS(n) = \sum_{i \in I} CR(i,n) - B(n) - E(n) \qquad \forall n \in N \qquad (3\text{-}3)$$

Constraint (3-4) stipulates that the total water flowrate to water using operation *i* is made up of cooling water from cooling towers and reuse cooling water from other operations.

$$F_{in}(i) = \sum_{n \in N} CS(n,i) + \sum_{i' \in I} FR(i',i) \qquad \forall i \in I \qquad (3\text{-}4)$$

Constraint (3-5) states that the total water flowrate from water using operation *i* is made up of reuse cooling water to other operations and cooling water recycling back to the cooling towers.

$$F_{out}(i) = \sum_{i \in I} CR(i,n) + \sum_{i' \in I} FR(i,i') \qquad \forall i \in I \qquad (3\text{-}5)$$

Constraint (3-6) ensures that water is conserved through each cooling water using operation.

$$F_{in}(i) = F_{out}(i) \qquad \forall i \in I \qquad (3\text{-}6)$$

Constraint (3-7) is the definition of inlet temperature into operation *i.*

$$T_{in}(i) = \frac{\sum_{i' \in I} FR(i,i')T_{out}(i') + \sum_{n \in N} CS(n,i)T_s(n)}{F_{in}(i)} \qquad \forall i \in I \qquad (3\text{-}7)$$

Constraint (3-8) is the definition of circuit inlet temperature from any cooling tower after adding make up. This constraint caters for a change in temperature of cooling water as a result of makeup water addition.

$$T_s(n) = \frac{M(n)T_{amb} + OS(n)T_{ctout}(n)}{\left(CS(n)\right)} \qquad \forall n \in N \qquad (3\text{-}8)$$

Constraint (3-9) is the definition of return temperature to any cooling tower.

$$T_{ret}(n) = \frac{\sum_{i \in I} CR(i,n)T_{out}(i)}{\sum_{i \in I} CR(i,n)} \qquad \forall n \in N \qquad (3\text{-}9)$$

Energy balance across water using operation *i* is given by constraint (3-10).

$$\left(T_{out}(i) - T_{in}(i)\right)F_{in}(i)c_p = Q\ (i) \qquad \forall i \in I \qquad (3\text{-}10)$$

By substituting constraint (3-7) into constraint (3-10), the bilinear term $F_{in}(i)T_{in}(i)$ can be eliminated and constraint (3-7) and constraint (3-10) will be replaced by constraint (3-11).

$$Q(i) + c_p \sum_{n \in N} CS(n,i)T_s(n) + c_p \sum_{i' \in I} FR(i',i)T_{out}(i') = F_{in}(i)c_pT_{out}(i) \qquad \forall i \in I \qquad (3\text{-}11)$$

Design constraints

The equipments within the cooling water system have the maximum allowable flowrates and temperatures. The design constraints ensure that all the equipments are operated within their specified design limits.

Constraints (3-12) and (3-13) ensure that the cooling towers are operated below their maximum throughputs and the maximum allowable temperatures respectively.

$$OS(n) \le OS^u(n) \qquad \forall n \in N \qquad (3\text{-}12)$$

$$T_{ret}(n) \le T_{ret}^u(n) \qquad \forall n \in N \qquad (3\text{-}13)$$

Constraint (3-14) ensures that the flowrate through water using operations does not exceed its maximum design flowrate.

$$F_{in}(i) \le F_{in}^u(i) \qquad \forall i \in I \qquad (3\text{-}14)$$

The formulation for Case I entails constraints (3-1)-(3-6), (3-8), (3-9) and (3-11)-(3-13). The objective function of this model is to minimize the total cooling water as given in constraint (3-1). Constraints (3-8), (3-9) and (3-11) consist of bilinear terms which are nonconvex thus rendering the model NLP. This model is difficult to initialize because the starting point might be infeasible or the solution might be locally optimum. To overcome these difficulties the technique proposed by Sherali and Alameddine (1992) was used to linearize the bilinear terms. This technique uses the upper and the lower bounds to create a convex space for the bilinear terms as shown in the next section.

Relaxation linearization

Let,

$$crt(i,n) = CR(i,n)T_{out}(i) \quad \forall n \in N, i \in I$$

$$frt(i',i) = FR(i',i)T_{out}(i) \quad i \in I, i' \in I$$

$$fnt(i) = F_{in}(i)T_{out}(i) \quad \forall i \in I$$

$$tcs(n,i) \;=\; CS(n,i)T_s(n) \quad \forall n \;\in\; N, i \;\in\; I$$

The lower bound for the flowrates is zero and the lower bound for the temperature is the wet bulb temperature.

$$\begin{aligned} & crt(i,n) \geq OS^u(n)T_{out}(i) + CR(i,n)T_{out}^u(i) \\ & - OS^u(n)T_{out}^u(i) \end{aligned} \qquad \forall n \;\in\; N,\; \forall i \;\in\; I \tag{3-15}$$

$$crt(i,n) \geq CR(i,n)T_{out}^L(i) \qquad \forall n \;\in\; N,\; \forall i \;\in\; I \tag{3-16}$$

$$\begin{aligned} & crt(i,n) \leq OS^u(n)T_{out}(i) + CR(i,n)T_{out}^L(i) \\ & - OS^u(n)T_{out}^u(i) \end{aligned} \qquad \forall n \;\in\; N, \forall i \;\in\; I \tag{3-17}$$

$$crt(i,n) \leq CR(i,n)T_{out}^u(i) \qquad \forall n \;\in\; N, \forall i \;\in\; I \tag{3-18}$$

$$\begin{aligned} & frt(i',i) \geq F_{in}^u(i)T_{out}(i') + FR(i',i)T_{out}^u(i) \\ & - F_{in}^u(i)T_{out}^u(i) \end{aligned} \qquad i \;\in\; I, i' \;\in\; I \tag{3-19}$$

$$frt(i',i) \geq FR(i',i)T_{out}^L(i) \qquad i \;\in\; I, i' \;\in\; I \tag{3-20}$$

$$\begin{aligned} & frt(i',i) \leq F_{in}^u(i)T_{out}(i') + FR(i',i)T_{out}^L(i) \\ & - F_{in}^u(i)T_{out}^L(i) \end{aligned} \qquad i \;\in\; I, i' \;\in\; I \tag{3-21}$$

$$frt(i',i) \leq FR(i',i)T_{out}^u(i) \qquad i \;\in\; I, i' \;\in\; I \tag{3-22}$$

$$fnt(i) \geq F_{in}^u(i)T_{out}(i) + F_{in}(i)T_{out}^u(i) - F_{in}^u(i)T_{out}^u(i) \qquad \forall i \;\in\; I \tag{3-23}$$

$$fnt(i) \geq F_{in}(i)T_{out}^L(i) \qquad \forall i \;\in\; I \tag{3-24}$$

$$fnt(i) \leq F_{in}^u(i)T_{out}(i) + F_{in}(i)T_{out}^L(i) - F_{in}^u(i)T_{out}^L(i) \qquad \forall i \;\in\; I \tag{3-25}$$

$$fnt(i) \leq F_{in}(i)T_{out}^u(i) \qquad \forall i \;\in\; I \tag{3-26}$$

$$\begin{aligned} & tcs(n,i) \geq OS^u(n)T_s(n) + CS(n,i)T_s^u(n) \\ & - OS^u(n)T_s^u(n) \end{aligned} \qquad \forall n \;\in\; N, i \;\in\; I \tag{3-27}$$

$$tcs(n,i) \geq CS(n,i)T_s^L(n) \qquad \forall n \;\in\; N, i \;\in\; I \tag{3-28}$$

$$tcs(n,i) \geq OS^{u}(n)T_{s}(n) + CS(n,i)T_{s}^{L}(n) - OS^{u}(n)T_{s}^{L}(n) \qquad \forall n \in N, i \in I \qquad (3\text{-}29)$$

$$tcs(n,i) \geq CS(n,i)T_{s}^{u}(n) \qquad \forall n \in N, i \in I \qquad (3\text{-}30)$$

Introduction of linearization variables requires constraint (3-9), constraint (3-11) and constraint (3-8) to be modified as shown in constraint (3-31), constraint (3-32) and constraint (3-33), respectively.

$$T_{ret}(n) = \frac{\sum_{i \in I} crt(i,n)}{\sum_{i \in I} CR(i,n)} \qquad \forall n \in N \qquad (3\text{-}31)$$

$$Q(i) + c_{p}\sum_{n \in N} tcs(n,i) + c_{p}\sum_{i' \in I} frt(i',i) = fnt(i)c_{p} \qquad \forall i \in I \qquad (3\text{-}32)$$

$$\sum_{i \in I} tcs(n,i) = M(n)T_{amb} + OS(n)T_{ctout}(n) \qquad \forall n \in N \qquad (3\text{-}33)$$

The relaxed LP model for Case I consists of Constraint (3-1)-(3-6), (3-12)-(3-13) and (3-15)-(3-33). To get the solution for Case I, the relaxed model is first solved by minimizing the total cooling water. The solution of the relaxed model is then used as a starting point for solving the exact model.

3.2.1.2 Case II

In this case the source and sink for any cooling water using operation are the same. This implies that no pre-mixing or post-splitting of cooling water return is allowed. A set of heat exchanger can only be supplied by one cooling tower. Furthermore, the return cooling water from cooling water using operation must supply the source cooling tower. However, reuse of water within the network is still allowed. All the constraints in Case I are still applicable. Few constraints needed to be added to control the source and the sink.

Constraints (3-34) and (3-35) prevent pre-mixing. Constraint (3-34) ensures that the supply flowrate from any cooling tower to operation *i* cannot exceed the maximum flowrate. Constraint (3-35) ensures that cooling water using operation *i* can only be supplied by a maximum of one cooling tower.

$$CS(n,i) \leq F_{in}^{u}(i)ys(n,i) \qquad \forall i \in I \qquad \forall n \in N \qquad (3\text{-}34)$$

$$\sum_{n \in N} ys(n,i) \leq 1 \qquad \forall n \in N \qquad (3\text{-}35)$$

Post-splitting can also be prevented by constraints (3-36) and (3-37). Constraint (3-36) ensures that the return flowrate from operation *i* to any cooling tower cannot exceed the maximum flowrate into that operation. Constraint (3-37) ensures that cooling water using operation *i* can supply a maximum of one cooling tower.

$$CR(i,n) \le F_{in}^{u}(i) yr(i,n) \qquad \forall i \in I \qquad \forall n \in N \tag{3-16}$$

$$\sum_{n \in N} yr(i,n) \le 1 \qquad \forall n \in N \tag{3-37}$$

Constraints (3-38) and (3-39) ensure that the source and the sink cooling water supply is the same for a particular cooling water using operation.

$$yr(i,n) \le ys(n,i) + \left(2 - \sum_{n \in N} ys(n,i) - \sum_{n \in N} yr(i,n) \right) \forall i \in I \ \forall n \in N \tag{3-38}$$

$$yr(i,n) \ge ys(n,i) - \left(2 - \sum_{n \in N} ys(n,i) - \sum_{n \in N} yr(i,n) \right) \forall i \in I \ \forall n \in N \tag{3-39}$$

The formulation for Case II entails constraint (3-1)-(3-6), (3-8), (3-9), (3-11)-(3-13), (3-34)-(3-39). Constraints (3-34)-(3-39) consist of binary variables while constraint (3-8), (3-9) and (3-13) consist of bilinear terms which are nonconvex. This renders the model MINLP. Similar to Case I, the model was linearized using the linearization relaxation procedure by Sherali and Alameddine (1992) as shown in constraints (3-15)-(3-33). The relaxed model entails constraints (3-1)-(3-6), (3-12)-(3-13) and (3-15)-(3-39).

The solution procedure starts by first solving the relaxed model by minimizing the total cooling water. The solution of the relaxed model is then used as a starting point for solving the exact model.

Cooling tower model

Gololo and Majozi (2011) used the cooling tower model developed by Kröger (2004) to study the interaction between the cooling water network and the cooling towers. One cooling tower model was used for all cooling towers in the cooling water system. The only distinction in the cooling towers was the correlations for the cooling tower coefficient of performance $\left(\frac{KaV}{m_w} \right)$.

The governing equations that predict the thermal performance of a cooling tower are given by equations (3-40), (3-41) and (3-42). Equations (3-40) and (3-41) define the mass and energy balance for the control volume, respectively. Equation (3-42) defines the air enthalpy change through the cooling tower fill.

$$\frac{dm_w}{dz} = m_a \frac{dw}{dz} \tag{3-40}$$

$$\frac{dT_w}{dz} = \frac{m_a}{cp_w m_w} \left(\frac{1}{cp_w} \frac{dH_a}{dz} - T_w \frac{dw}{dz} \right) \tag{3-41}$$

$$\frac{dH_a}{dz} = \frac{Ka_{fi} A_{fi}}{m_a} \left(Le_f (H_{as} - H_a) + (1 - Le_f) H_v (w_s - w) \right) \tag{3-42}$$

The model predicted the outlet water temperature, effectiveness, evaporation, makeup and blowdown for each cooling tower. This model was solved using MATLAB.

To synthesize the overall cooling water system, both the cooling tower model and the heat exchanger network model should be solved simultaneously. The algorithm for synthesizing the overall cooling water system is given in the following section.

Solution procedure

The cooling tower model and the cooling water network model were simultaneously solved using the procedure shown in Figure 18. The first step was to optimize the cooling water network model without the cooling towers. The results from the first iteration, which are cooling water return (CWR) temperatures and flowrates, become the input to the cooling tower models. Each cooling tower model then predicts the outlet water temperatures and flowrates. This was done by first assuming the outlet water temperature of a cooling tower. The assumption is done by subtracting 0.5°C from the given cooling tower inlet temperature. The governing mass and heat transfer equations were then solved numerically using forth order Runge_Kutta method starting from the bottom of the cooling tower moving upwards at stepsize Δz. When the maximum height is reached, the temperature at this point will be compared with the CWR temperature. If the two agree within a specified tolerance, the cooling tower model will stop and the outlet temperature will be given as the assumed temperature, else the inlet temperature will be adjusted until the CWR temperature agrees with the calculated temperature. The predicted outlet cooling tower temperatures and flowrates then become the input to the heat exchanger network model. If the outlet temperature of the cooling tower model agrees with the previous inlet temperature to the heat exchanger network model, the algorithm stops which implies that final results have been obtained. Otherwise the iteration continues. The procedure used to connect MATLAB and GAMS was adopted from Ferris (2005).

3.3 Case studies

The application of the proposed technique is demonstrated by considering one example for Cases I and II.

Base case

Cooling water system in Figure 19 shows a set of cooling water networks which are supplied by a set of cooling towers. Each cooling water using operation is supplied by fresh water from the cooling tower and return back to the cooling tower.

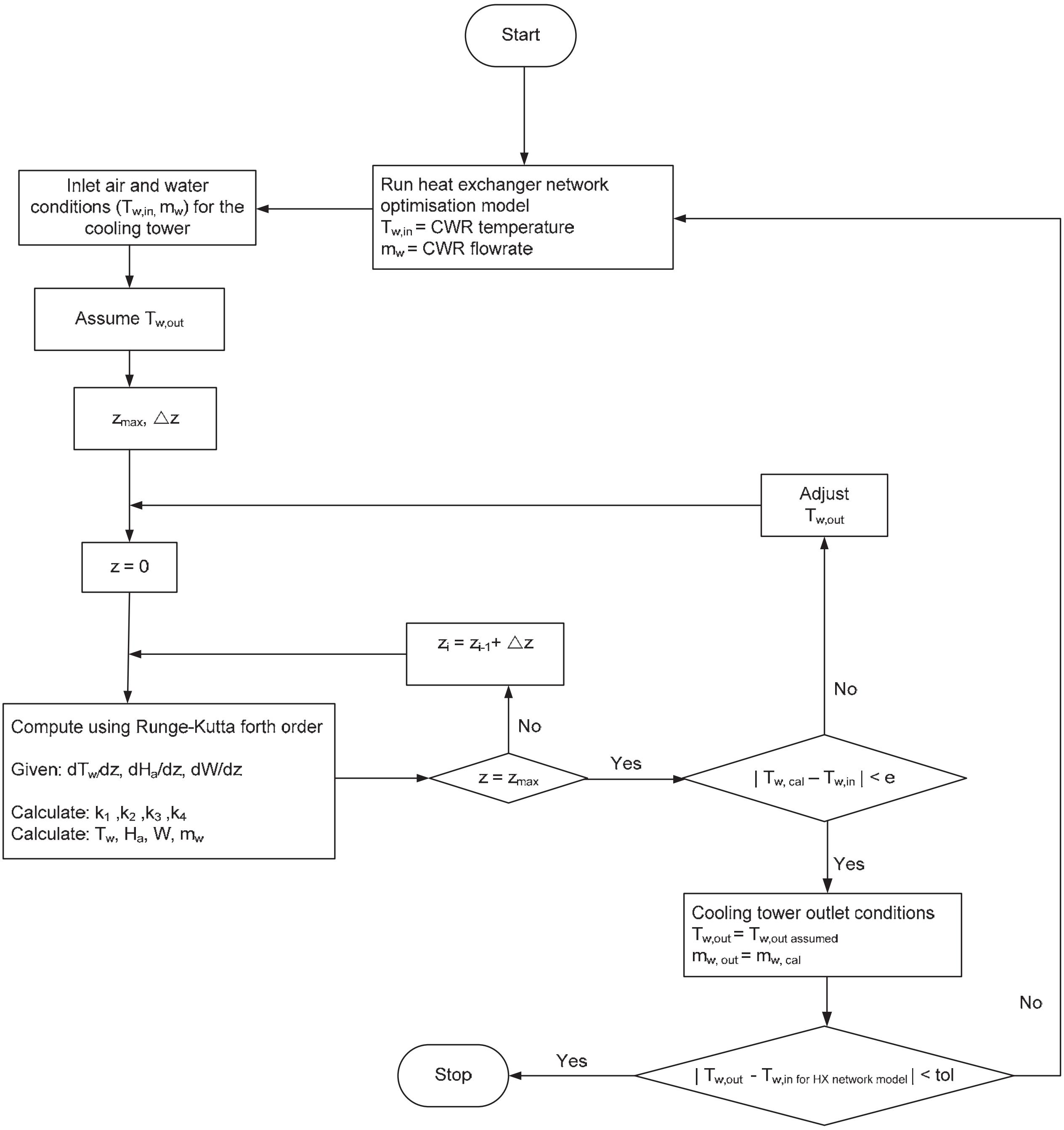
Start
Run heat exchanger network optimisation model
$T_{w,in}$ = CWR temperature
m_w = CWR flowrate
Inlet air and water conditions ($T_{w,in}$, m_w) for the cooling tower
Assume $T_{w,out}$
z_{max}, △z
Adjust $T_{w,out}$
z = 0
$z_i = z_{i-1} + \triangle z$
No
No
Compute using Runge-Kutta forth order
Given: dT_w/dz, dH_a/dz, dW/dz
Calculate: k_1, k_2, k_3, k_4
Calculate: T_w, H_a, W, m_w
$z = z_{max}$
Yes
$| T_{w, cal} - T_{w,in} | < e$
Yes
Cooling tower outlet conditions
$T_{w,out} = T_{w,out\ assumed}$
$m_{w,\ out} = m_{w,\ cal}$
No
Stop
Yes
$| T_{w,out} - T_{w,in\ for\ HX\ network\ model} | < tol$

***Figure 18**: Flowchart for cooling water system model (Gololo and Majozi, 2011)*

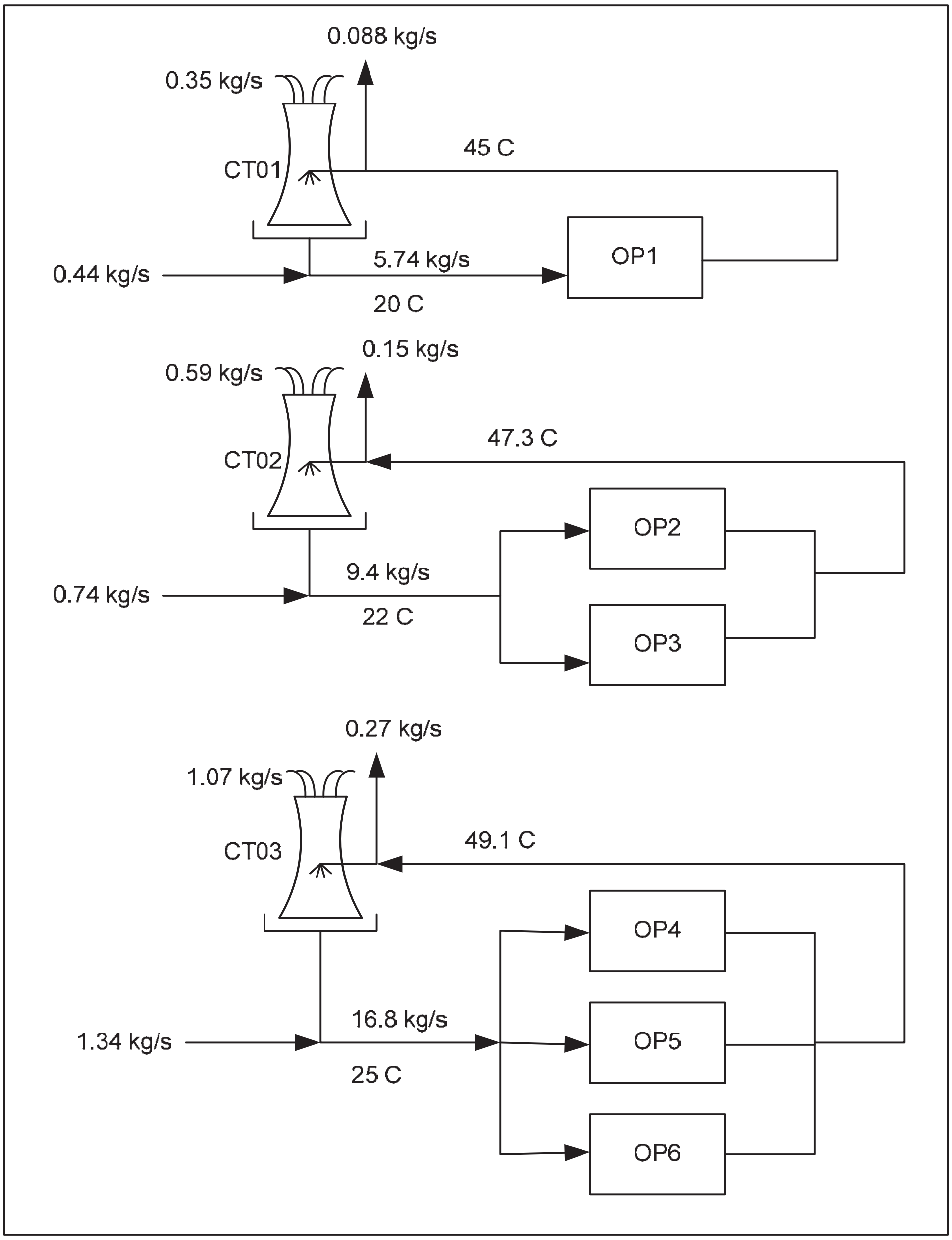

***Figure 19**: Base case (Gololo and Majozi, 2011)*

The heat duties, temperature limits and design information are shown in Tables 1 and 2. T_{ret}^{u} is the maximum allowable temperature for packing inside the cooling towers while OS^{u} is the maximum flowrate of the cooling tower before flooding. T_{in}^{u} and T_{out}^{u} are the thermodynamic temperature limits for the inlet and outlet temperature of the cooling water using operation respectively.

***Table 1**: Cooling towers design information*

Cooling towers	$T_{ret}^{u}(^{o}C)$	$OS^{u}(kg/s)$	$\frac{Ka_{fi}}{m_w} = \frac{a_d}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{d_{da}} ATD^{b_{db}}$
CT01	50	9.6	$\frac{Ka_{fi}}{m_w} = \frac{2.49}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{-0.67} 0.609^{-0.062}$
CT02	50	16	$\frac{Ka_{fi}}{m_w} = \frac{1.664}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{-0.62} 0.914^{-0.27}$
CT03	55	20	$\frac{Ka_{fi}}{m_w} = \frac{2.49}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{-0.67} 0.914^{-0.062}$
Height = 2.438, Area = 5.943, CC = 5			

***Table 2**: Limiting cooling water data*

Operations	T_{in}^{u} **(C)**	T_{out}^{u} **(C)**	F_{in} **(kg/s)**	**Q(i)(kW)**
OP01	30	45	9.52	600
OP02	40	60	3.57	300
OP03	25	50	7.62	800
OP04	45	60	7.14	600
OP05	40	55	4.76	300
OP06	30	45	11.1	700

Case I

In this case each cooling tower can supply any cooling water using operation. The return streams from any cooling water using operation can go to any cooling tower. The return temperature to any cooling tower is however specified.

Figure 20 shows the cooling water system after applying the methodology described above. By exploiting the opportunity for cooling water reuse, the overall circulating water decreased by 22% and one cooling tower was eliminated. The cooling tower inlet temperatures are at their maximum values.

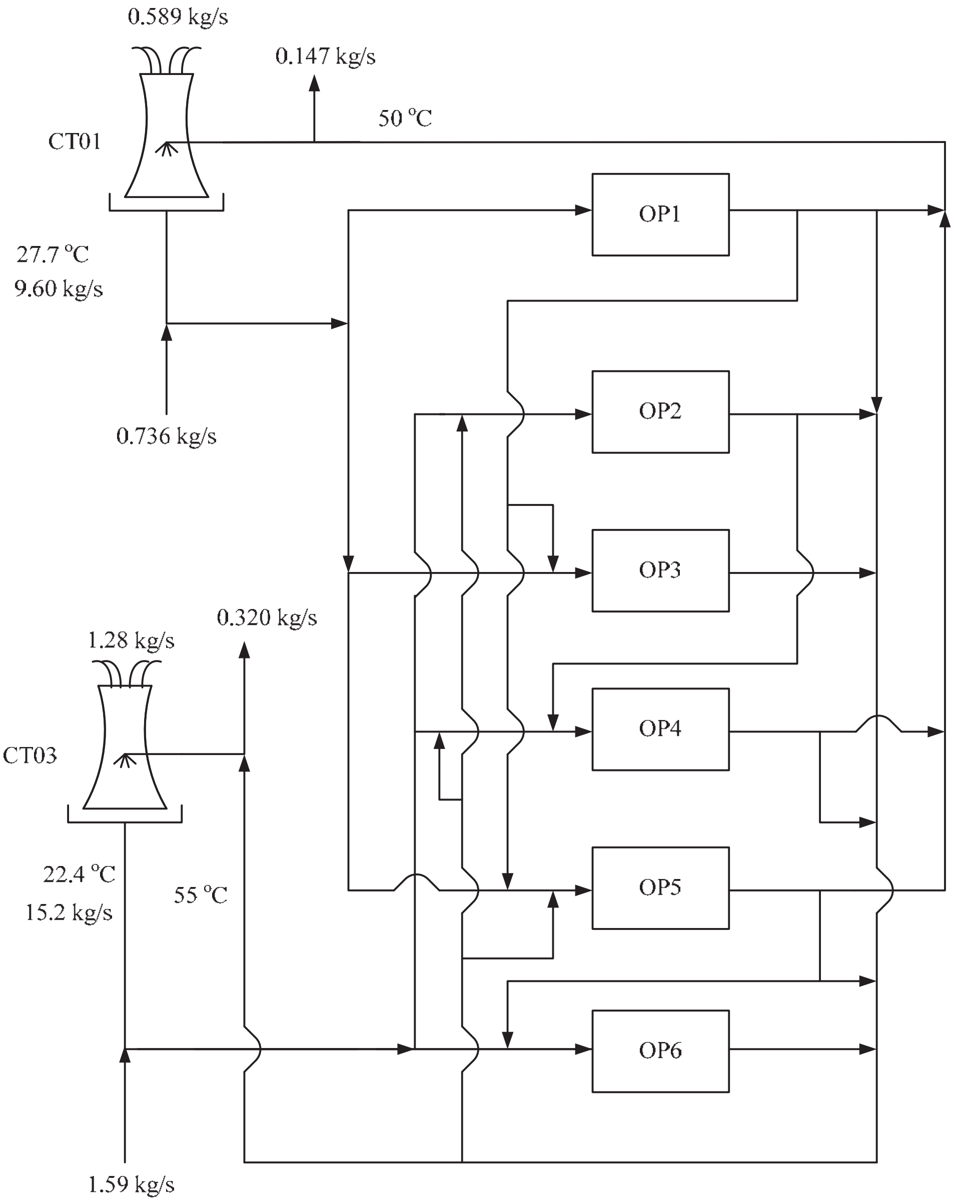

***Figure 20**: Final design of the cooling water system*

These results show the opportunity to increase the heat duties, through expansions, without investing on a new cooling tower. The only additional investment required is on piping for reuse streams.

Case II

In this case a cooling tower can only supply a dedicated set of heat exchangers. This implies that each operation can only be supplied by one cooling tower. The return streams from any cooling water using operation can only go to its supplier cooling tower. The return temperatures to the cooling towers are also specified. Figure 21 shows the heat exchanger network after applying the methodology described above.

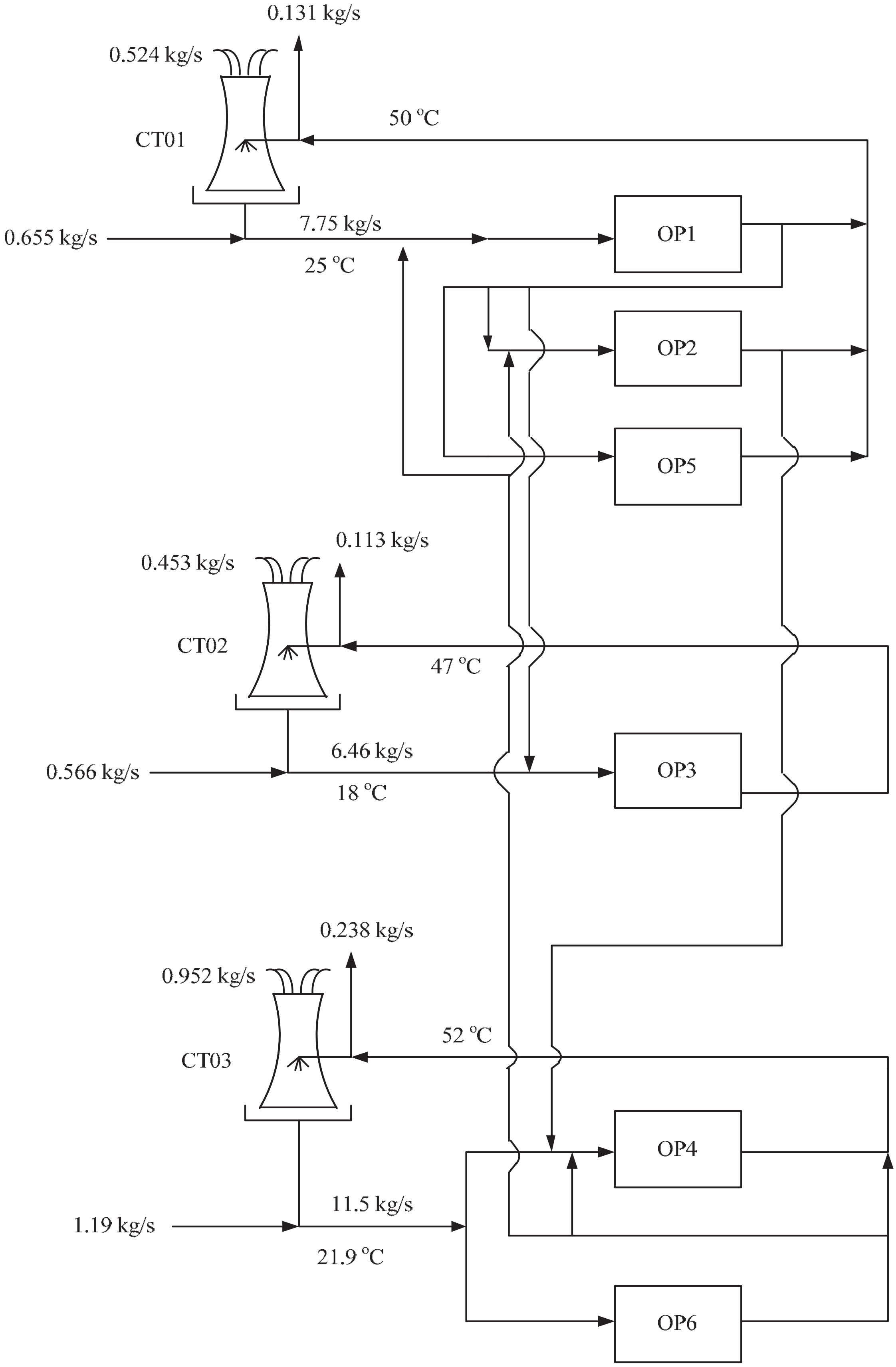

***Figure 21**: Final design of the cooling water system*

By allowing for the cooling water reuse, the overall circulating water decreased by 20%. This will decrease the pumping power requirement for the circulating pump thus reducing the pumping cost. The cooling towers spare capacity is also increased giving opportunities for increased heat load without investing in a new cooling tower. To satisfy the required heat duties with the reduced flowrate, the return temperature to the cooling towers is increased to the maximum value.

The overall effectiveness for multiple cooling towers

Table 3-3 shows an increase in the overall effectiveness of the cooling towers when applying the proposed design methodology. The increase in the overall cooling towers effectiveness is attributed to a decrease in the overall circulating water and an increase in return water temperature.

***Table 3**: Effectiveness for base Case, Case I and Case II*

	Effectiveness	Circulating water (kg/s)
Base case	0.90	31.94
Case I	0.91	24.80
Case II	0.94	25.69

Single source approach

Kim and Smith (2001) proposed a technique for optimization of cooling water system with a single cooling water source. This technique does not consider interaction of between various cooling circuits within the same operation. Using the case study shown above, each circuit was independently optimized and the results are shown in Figure 22. Each circuit was optimized independently thus restricting cooling water reuse from one circuit to the other.

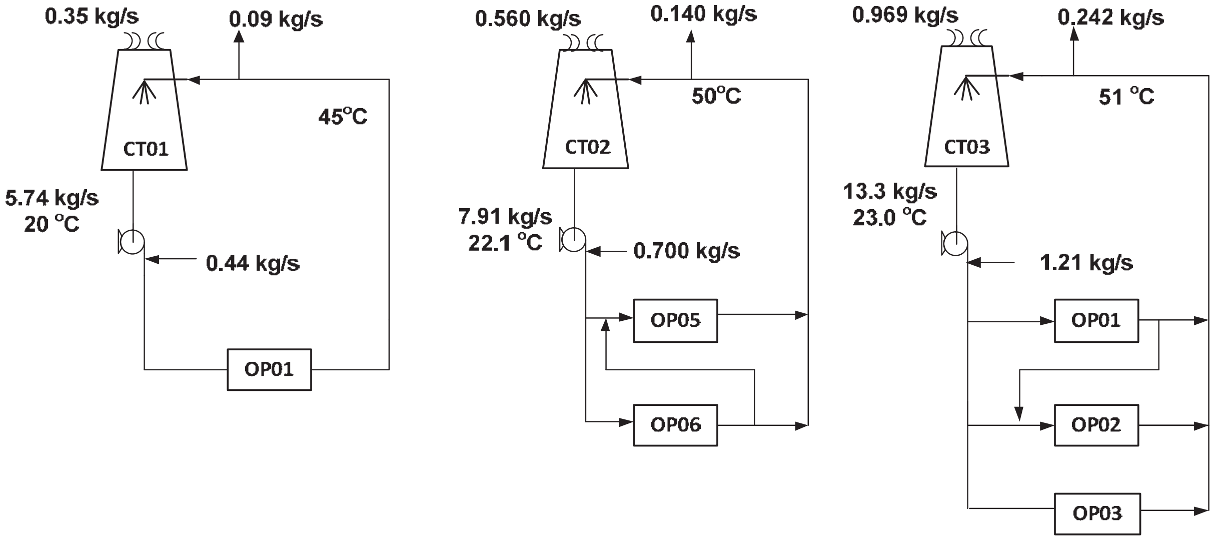

***Figure 22**: Debottlenecked cooling water system with no interaction between circuits*

The total circulating water flowrate is 27 kg/s. This is equivalent to only 15% decrease in circulating water flowrate. The overall effectiveness is 92.7%. The comparison between individual optimization technique and the integrated technique suggests that it is better to use an integrated approach to optimize the cooling water system.

3.4 Conclusions

The mathematical technique for optimization and synthesis of cooling water systems with multiple cooling towers has been presented in this chapter to demonstrate the benefit of a comprehensive approach to cooling water system design. The detailed mathematical framework consisting of the cooling towers and the cooling water network models was developed considering two practical cases. Case I involves a cooling water system with no dedicated cooling water sources and sinks. This implies that a set of heat exchangers can be supplied by any cooling tower and return the cooling water to any cooling tower. Case II involves a cooling water system with dedicated cooling water sources and sinks. This implies that a set of heat exchangers can only be supplied by one cooling tower. No pre-mixing or post-splitting of cooling water return is allowed. However, reuse of water within the network is still allowed. The formulations for Case I yield nonlinear programming (NLP) structure whilst Case II yield mixed integer nonlinear programming (MINLP) structure.

The proposed technique debottlenecked the cooling towers by decreasing the total circulating water flowrate. This implies that a given set of cooling towers can manage an increased heat load. From the case study, 22% and 20% decrease in circulating water flowrate was realized for Case I and Case II respectively. A decrease in the overall circulation water has an added benefit of decreasing the overall power consumption of the circulating pumps.

An improvement of up to 4% overall cooling towers effectiveness was realized by applying the proposed technique in a case study. This improvement was due to a decrease in circulating water flowrate with an increase in return cooling water temperature. When the return cooling water temperature is high, the driving forces between air and water are improved thus more heat is removed from the cooling water.

This technique is more holistic because it caters for the effect of cooling tower performance on cooling water network. The results obtained using this technique are more practical, since all components of the cooling water system are included in the analysis. The comparison between Gololo and Majozi (2011), and Kim and Smith (2001) technique shows that better results could be achieved by considering the interaction of all cooling water networks.

Although the mathematical technique by Gololo and Majozi (2011) is holistic, they are few setbacks. The topology of the debottlenecked cooling water network is more complex due to the additional reuse streams. The network also consists of series-parallel combination of heat exchanger thus prone to higher network pressure drop. Therefore, it is imperative to consider cooling water network pressure drop when synthesizing and optimizing cooling water systems. Another setback is the solution procedure. The global optimality cannot be guaranteed because two platforms were used to model different components of the cooling water systems. The authors used MATLAB to solve the cooling tower model and GAMS to optimize the cooling water network. This approach does not simultaneously optimize the whole cooling water system. Better results could be obtained by using one platform approach.

4 MODEL DEVELOPMENT

4.1 Introduction

The cooling water system consists of cooling towers and heat exchanger networks thus, the mathematical model for synthesis and optimization of the cooling water system entails the heat exchanger network model and the cooling tower model. The heat exchanger model is based on a superstructure in which all possible cooling water reuse opportunities are explored. The interaction between the heat exchanger network and the cooling towers is investigated using the cooling tower model derived by Kröger (2004). The following sections detail the development of the cooling tower model and the heat exchanger network model.

4.2 Cooling Tower Model

The cooling tower model is derived by considering a control volume as shown Figure 23.

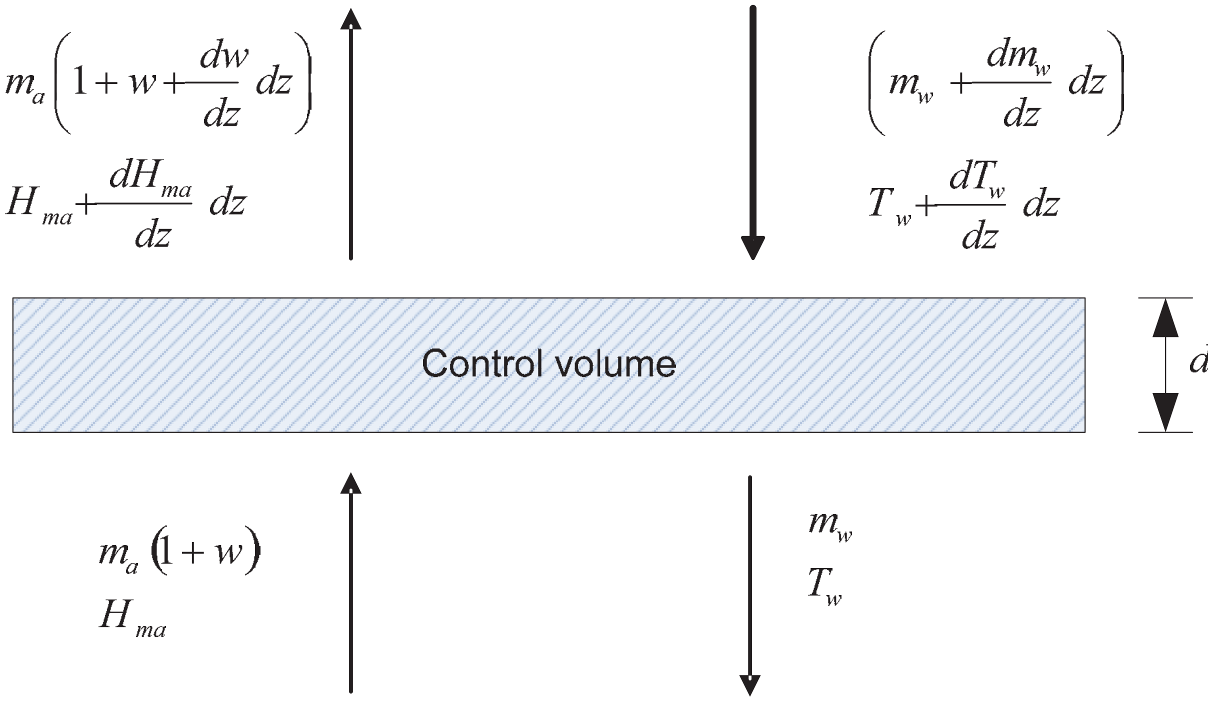
$m_a\left(1+w+\frac{dw}{dz}dz\right)$
$H_{ma}+\frac{dH_{ma}}{dz}dz$
$\left(m_w+\frac{dm_w}{dz}dz\right)$
$T_w+\frac{dT_w}{dz}dz$
Control volume
dz
$m_a(1+w)$
H_{ma}
m_w
T_w

***Figure 23**: Control volume*

The following assumptions are made:

- Interface water temperature is the same as the bulk temperature
- Air and water properties are the same at any horizontal cross section
- Heat and mass transfer area are identical

The governing equations that predict the thermal performance of a cooling tower are given by equations (4-2), (4-4) and (4-14).

The mass and energy balance equations for the control volume are given in equations (4-1) and (4-3) respectively.

$$m_a(1+w)+\left(m_w+\frac{dm_w}{dz}dz\right)=m_a\left[1+\left(w+\frac{dw}{dz}dz\right)\right]+m_w \tag{4-1}$$

The above equation can be further simplified as shown in equation (4-2).

$$\frac{dm_w}{dz}=m_a\frac{dw}{dz} \tag{4-2}$$

$$m_aH_a+\left(m_w+\frac{dm_w}{dz}dz\right)c_{pw}\left(T_w+\frac{dT_w}{dz}dz\right)=m_a\left(H_a+\frac{dH_a}{dz}dz\right)+m_wc_{pw}T_w \tag{4-3}$$

Ignoring the second order terms, equation (4-3) can be further simplified as shown in equation (4-4).

$$\frac{dT_w}{dz}=\frac{m_a}{cp_wm_w}\left(\frac{1}{cp_w}\frac{dH_a}{dz}-T_w\frac{dw}{dz}\right) \tag{4-4}$$

The enthalpy transfer between the air and water interface is given in equation (4-5).

$$dQ=dQ_m+dQ_c \tag{4-5}$$

where, $dQ_m=H_VK(w_s-w)dA$ (Mass transfer enthalpy) (4-6)

and $dQ_c=h(T_w-T_a)dA$ (Convective heat transfer) (4-7)

therefore, $dQ=H_VK(w_s-w)dA+h(T_w-T_a)dA$ (4-8)

The enthalpies of unsaturated air and saturated air are given in equations (4-9) and (4-10) respectively.

$$H_a=c_{pa}T_a+w(H_{fgwo}+c_{pv}T_a) \tag{4-9}$$

$$H_{as}=c_{pa}T_w+w_s(H_{fgwo}+c_{pv}T_w)=c_{pa}T_w+w_sH_V \tag{4-10}$$

The difference of the above equations is shown in equation (4-11).

$$H_{as}-H_a=(c_{pa}+wc_{pv})(T_w-T_a)+(w_s-w)H_v \tag{4-11}$$

Or

$$T_w-T_a=\left[(H_{as}-H_a)-(w_s-w)H_v\right]\Big/c_{pma} \tag{4-12}$$

where $c_{pma}=c_{pa}+wc_{pv}$

By substituting the expression ($T_w - T_a$) given in equation (4-12) into equation (4-8), the enthalpy transfer between the air and water interface can be expressed as follows:

$$dQ = K\left[\frac{h}{Kc_{pma}}(H_{as} - H_a) + \left(1 + \frac{h}{Kc_{pma}}\right)H_v(w_s - w)\right]dA \tag{4-13}$$

The enthalpy change must be equal to the enthalpy change of air stream thus, the enthalpy of air across the length of the cooling tower can be expressed as follows:

$$\frac{dH_a}{dz} = \frac{1}{m_a}\frac{dQ}{dz} = \frac{Ka_{fi}A_{fi}}{m_a}\left(Le_f(H_{as} - H_a) + (1 - Le_f)H_v(w_s - w)\right) \tag{4-14}$$

where $dA = a_{fi}A_{fr}dz$ and $\frac{h}{Kc_{pma}} = Le$

a_{fi} is the wetted area divided by the corresponding volume of the fill and A_{fr} is a frontal area. $\frac{h}{Kc_{pma}} = Le$ is a Lewis factor.

The Lewis factor is calculated from the expression given in equation (2-17) (repeated below) which was used by Klopper and Kröger (2005) to study the influence of the Lewis factor on the performance prediction of wet cooling tower.

$$Le_f = 0.866^{0.667}\left(\frac{w_s + 0.622}{w + 0.622} - 1\right) / \ln\left(\frac{w_s + 0.622}{w + 0.622}\right)$$

Coefficient of performance

Kröger (2004) suggested a correlation for counter flow fills as shown equation (4-15).

$$\frac{Ka_{fi}}{m_w} = \frac{a_d}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{d_{da}} ATD^{b_{db}} \tag{4-15}$$

where a_d, d_{da}, ATD and b_{db} are system parameters.

Makeup and blowdown

Equations (3-16) and (3-17) are used to calculate the makeup and blowdown flowrates.

$$M = E\frac{CC}{CC - 1} \text{ where CC is the cycle of concentration} \tag{4-16}$$

$$M = E + B \tag{4-17}$$

The overall effectiveness of the cooling towers

The effectiveness is defined as the ratio of actual heat transferred to the maximum theoretical amount of heat that can be transferred shown in constraint (4-18) (Jaber and Webb, 1987). In a circuit consisting of multiple cooling towers, the overall effectiveness for the cooling towers is evaluated using the expression in constraint (4-19). This expression is derived from the thermodynamic definition of the effectiveness for one cooling tower shown in Constraint (4-18).

$$\varepsilon = \frac{Q_{act}}{Q_{\max}} = \frac{m_w c_{pw}\left(T_{w,in} - T_{w,out}\right)}{m_{cap}\left(H_{as,out} - H_{a,in}\right)} \tag{4-18}$$

where m_w is the cooling tower water flowrate, c_{pw} is the specific heat capacity of water, $\left(T_{w,in} - T_{w,out}\right)$ is the cooling tower range, m_{cap} is the water capacity rate, $H_{as,out}$ is the enthalpy of saturated air at inlet water temperature and $H_{a,in}$ is the enthalpy of air at inlet air temperature.

$$\varepsilon = \frac{\sum_{n=1}^{N} Q_{act}^{n}}{\sum_{n=1}^{N} Q^{n}{}_{\max}} \tag{4-19}$$

where superscript *n* represents cooling tower *n* in circuit consisting of *N* number of cooling towers. The term $\sum_{n=1}^{N} Q_{act}^{n}$ represents the total amount of actual heat transferred and $\sum_{n=1}^{N} Q^{n}{}_{\max}$ represents the total maximum theoretical heat that can be transferred.

4.3 Heat Exchanger Network Model

A two-step approach is employed to synthesize and optimize the cooling water system with multiple cooling towers considering pressure drop. The first step involves targeting of the minimum circulating water flowrate and in the second step the CPA is incorporated to synthesize the cooling water network with minimum pressure drop.

Using the superstructure in Figure 24, the mathematical formulation is developed considering two cases. The cooling water network model is adapted from the paper of Gololo and Majozi (2011).

Case I The first case involves a cooling water system with no dedicated cooling water sources and sinks. This implies that a set of heat exchangers can be supplied by any cooling tower and return the cooling water to any cooling tower.

Case II The second case involves a cooling water system with dedicated cooling water sources and sinks. This implies that a set of heat exchangers can only be supplied by one cooling tower. No pre-mixing or post-splitting of cooling water return is allowed. However, reuse of water within the network is still allowed.

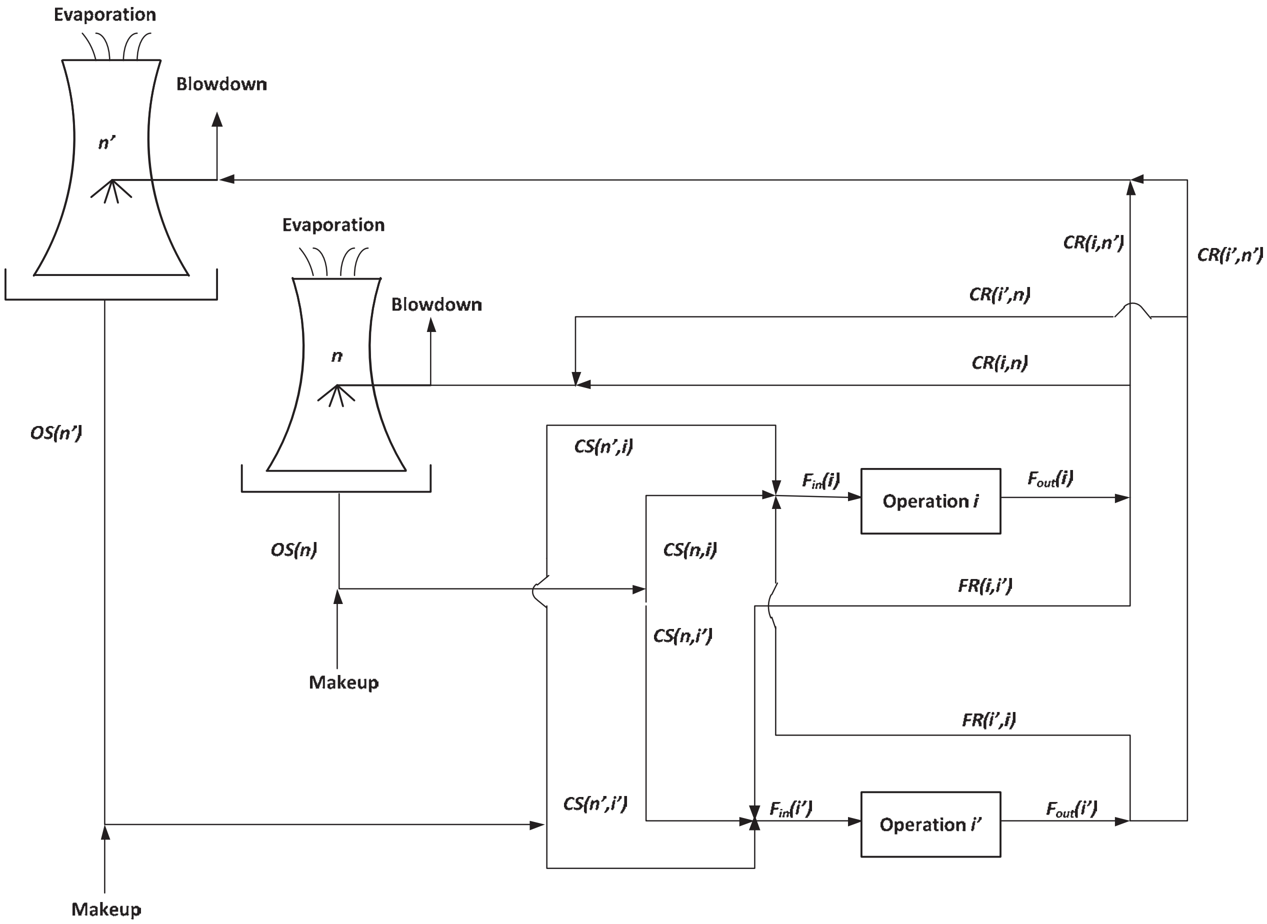

***Figure 24**: Superstructure for a cooling water system (Gololo and Majozi, 2011).*

Mathematical formulation

The mathematical formulation is the extension of the work of Majozi and Moodley (2008) and, Gololo and Majozi (2011). Additional constraints included in the model include the cooling tower model and pressure drop constraint, makeup, blowdown, and evaporation. The formulation entailed the following sets, parameters, continuous variables and constraints:

4.3.1.1 Sets

i = { *i* |*i* is a cooling water using operation}
n = { *n* |*n* is a cooling tower}

Parameters

$Q(i)$	Duty of cooling water using operation $i(kW)$
$T_{ctout}(n)$	Cooling water supply temperature from cooling tower $n(^oC)$
$OS^n(n)$	Maximum design capacity of cooling tower $n(kg/s)$
$T_{out}^U(i)$	Limiting outlet temperature of cooling water using operation $i(^oC)$
$T_{in}^U(i)$	Limiting inlet temperature of cooling water using operation $i(^oC)$
$F_{in}^U(i)$	Maximum inlet flowrate of cooling water using operation $i(kg/s)$
$T_{ret}^U(n)$	Limiting inlet temperature of cooling water using operation $n(^oC)$

$B(n)$	Blowdown flowrate for cooling tower $n(kg/s)$
$M(n)$	Makeup flowrate for cooling tower $n(kg/s)$
$E(n)$	Blowdown flowrate for cooling tower $n(kg/s)$
c_p	Specific heat capacity of water 4.2 (kJ/kg^oC).
T_{amb}	Ambient temperature (^oC)
T	Target (kg/s)
d_i, d_o	Heat exchanger tube diameter (m)
A	Heat exchanger area (m^2)
ρ	Density (kg/m^3)
μ	Viscosity (Ns/m^2)
N_t	Number of tubes in the heat exchanger
n_{tp}	Heat exchanger tube passes

Continuous variables

$OS(n)$	Operating capacity of cooling tower $n(kg/s)$
CW	Overall cooling water supply from all cooling tower (kg/s)
$CS(n,i)$	Cooling water supply from cooling tower *n* to cooling water using operation $i(kg/s)$
$CR(i,n)$	Return cooling water to cooling tower *n* from cooling water using operation $i(kg/s)$
$FR(i',i)$	Reuse cooling water to cooling water using operating *i'* from cooling water using operation $i(kg/s)$
$F_{in}(i)$	Total cooling water into cooling water using operation $i(kg/s)$
$F_{out}(i)$	Total cooling water from cooling water using operation $i(kg/s)$
f_j	Flowrate through the heat exchanger (kg/s)
F_p	Flowrate through piping (kg/s)
$T_{in}(i)$	Inlet temperature of cooling water to cooling water using operation $i(^oC)$
$T_{out}(i)$	Outlet temperature of cooling water to cooling water using operation $i(^oC)$
$T_s(n)$	Cooling water supply temperature from cooling tower *n* after adding makeup (^oC)
ΔP	Pressure drop (kPa)
S	Slack variable (kg/s)
$fret_{tot}(i)$	Fraction of the outlet stream being returned to cooling tower from operation *i*.
$fret(i,n)$	Fraction of the total return stream from operation *i*.to cooling tower *n*.
$freu(i,j)$	Fraction of the total reuse stream from operation *i*.to cooling tower *n*.
$crt(i,n)$	Linearization variable for relaxation technique
$frt(i',i)$	Linearization variable for relaxation technique
$fnt(i)$	Linearization variable for relaxation technique
$tcs(n,i)$	Linearization variable for relaxation technique

Binary variables

$x(n,i)$ Binary variable indicating existence of stream from cooling tower *n* to cooling water using operating i

$y(i',i)$ Binary variable indicating existence of stream from cooling water using operation *i'* to cooling water using operation i

$z(i,n)$ Binary variable indicating existence of stream from cooling water using operation i to cooling tower *n*

The mathematical formulation is developed from the superstructure given in Figure 3-2 by considering energy and mass balance equations across each cooling water using operation and at each node. Two cases that are considered are given in the following sections.

4.3.1.1 Case I

In this case there is no dedicated source or sink for any cooling water using operation. The water using operation can be supplied by one or more cooling towers. The maximum cooling water return temperatures to the cooling towers are also specified. This situation arises when packing material inside the cooling tower is sensitive to temperature and any cooling tower can supply any water using operation and the water using operation can return to any cooling tower.

The model is developed by considering mass balance, energy balance and pressure drop constraints.

Mass balance constraints

Constraint (4-20) sets the bound for circulating cooling water flowrate in cooling tower *n*. *S* is a slack variable used to relax the target.

$$OS(n) = T(n) + S(n) \tag{4-20}$$

Constraint (4-21) stipulates that the total cooling water is the sum of all cooling water from cooling tower *n.*

$$CW = \sum_{n \in N} OS(n) \tag{4-21}$$

Constraints (4-22) and (4-23) ensure that the inlet and outlet cooling water flowrates of cooling tower *n* are equal.

$$OS(n) = \sum_{i \in I} CS(n,i) - M(n) \qquad \forall n \in N \tag{4-22}$$

$$OS(n) = \sum_{i \in I} CR(i,n) - B(n) - E(n) \qquad \forall n \in N \tag{4-23}$$

The total water flowrate to water using operation *i* is the sum of all reuse cooling water from operation *i'* and the sum of cooling water flowrates from cooling tower *n* as given in constraint (4-24).

$$F_{in}(i) = \sum_{n \in N} CS(n,i) + \sum_{i' \in I} FR(i',i) \qquad \forall i \in I \qquad (4\text{-}24)$$

The total water flowrate from water using operation *i* is the sum of all reuse cooling water to operation *i'* and the sum of cooling water flowrates to cooling tower *n* as given in constraint (4-25).

$$F_{out}(i) = \sum_{i \in I} CR(i,n) + \sum_{i' \in I} FR(i,i') \qquad \forall i \in I \qquad (4\text{-}25)$$

Constraint (4-26) ensures that water is conserved through each cooling water using operation.

$$F_{in}(i) = F_{out}(i) \qquad \forall i \in I \qquad (4\text{-}26)$$

<u>*Energy balance constraints*</u>

Constraint (4-27) is the definition of inlet temperature into operation *i.*

$$F_{in}(i)T_{in}(i) = \sum_{i' \in I} FR(i',i)T_{out}(i') + \sum_{n \in N} CS(n,i)T_s\ (n) \qquad \forall i \in I \qquad (4\text{-}27)$$

Constraint (4-28) is the definition of circuit inlet temperature from cooling tower *n* after adding makeup.

$$T_s(n)\big(OS(n) + M(n)\big) = M(n)T_{amb} + OS(n)T_{ctout}(n) \qquad \forall n \in N \qquad (4\text{-}28)$$

Constraint (4-29) is the definition of return temperature to cooling tower *n*

$$T_{ret}(n)\sum_{i \in I} CR(i,n) = \sum_{i \in I} CR(i,n)T_{out}(i) \qquad \forall n \in N \qquad (4\text{-}29)$$

Energy balance across water using operation *i* is given by constraint (4-30).

$$\big(T_{out}(i) - T_{in}(i)\big)F_{in}(i)c_p = Q\ (i) \qquad \forall i \in I \qquad (4\text{-}30)$$

By substituting constraint (4-27) into constraint (4-30), the bilinear term $F_{in}(i)T_{in}(i)$ can be eliminated and constraint (4-27) and constraint (4-30) can be replaced by constraint (4-31).

$$\begin{aligned} & Q(i) + c_p \sum_{n \in N} CS(n,i)T(n) + c_p \sum_{i' \in I} FR(i',i)T_{out}(i') \\ & = F_{in}(i)c_p T_{out}(i) \end{aligned} \qquad \forall i \in I \qquad (4\text{-}31)$$

Design constraints

The cooling tower design constraints are given in constraints (4-32) and (4-33). Constraint (4-32) sets the upper limit of the flowrate for cooling tower (*n*). Constraint (4-33) sets the upper limit of the return water temperature for cooling tower (*n*).

$$OS(n) \leq OS^{u}(n) \qquad \forall n \in N \qquad (4\text{-}32)$$

$$T_{ret}(n) \leq T_{ret}^{u}(n) \qquad \forall n \in N \qquad (4\text{-}33)$$

Constraint (4-34) sets the upper limit of the inlet flowrate for operation *i*.

$$F_{in}(i) \leq F_{in}^{u}(i) \qquad \forall i \in I \qquad (4\text{-}34)$$

Constraints (4-35) and (4-36) assign a binary variable and set the bounds for any stream from cooling source *n* to operation *i*.

$$CS(n,i) \leq F_{in}^{u}(i)x(n,i) \qquad \forall i \in I \qquad \forall n \in N \qquad (4\text{-}35)$$

$$CS(n,i) \leq F_{in}^{L}(i)x(n,i) \qquad \forall i \in I \qquad \forall n \in N \qquad (4\text{-}36)$$

Constraints (4-37) and (4-38) assign a binary variable and set the bounds for any reuse stream from operation *i’* to operation *i*.

$$FR(i',i) \leq F_{in}^{u}(i)y(i',i) \qquad \forall i \in I \qquad (4\text{-}37)$$

$$FR(i',i) \leq F_{in}^{L}(i)y(i',i) \qquad \forall i \in I \qquad (4\text{-}38)$$

Constraints (4-39) and (4-40) assign a binary variable and set bounds for any stream from operation *i* to cooling source *n*.

$$CR(i,n) \leq F_{in}^{u}(i)z(i,n) \qquad \forall i \in I \qquad \forall n \in N \qquad (4\text{-}39)$$

$$CR(i,n) \leq F_{in}^{L}(i)z(i,n) \qquad \forall i \in I \qquad \forall n \in N \qquad (4\text{-}40)$$

Constraint (4-41) is used to limit the number of inlet streams to any water using operation.

$$\sum_{n} x_{n,i} + \sum_{i'} y_{i',i} = K \qquad \forall i \in I \qquad \forall n \in N \qquad (4\text{-}41)$$

where *K* is a natural integer value.

The targeting model consists of constraints (4-21)-(4-30) and (4-32)-(4-34). This formulation has bilinear terms thus rendering the model an NLP. The model is solved by minimizing circulating cooling water flowrate (*CW*) given in constraint (4-21).

Pressure drop constraints

The cooling water network model by Gololo and Majozi (2011) is improved by incorporating the modified heat exchangers and pipes pressure drop correlations of Nie and Zhu (1999) shown in constraints (4-42) and (4-45) respectively. In this paper the correlation of Nie and Zhu (1999) is expressed in terms of mass flowrate.

$$\Delta P(i) = N_{t1} F_{in}(i)^{1.8} + N_{t2} F_{in}(i)^{2} \qquad (4\text{-}42)$$

$$\text{where } N_{t1} = \frac{1.115567 \mu^{0.2} n_{tp}{}^{2.8} A}{\pi^{2.8} \rho N_t^{2.8} d_o d_i^{4.8}} \qquad (4\text{-}43)$$

$$N_{t2} = \frac{20 n_{tp}{}^{3} \rho}{\pi^{2} N_t^{2} d_i^{4}} \qquad (4\text{-}44)$$

The line pressure drop is calculated from constraint (4-45).

$$\Delta P = N_p \frac{1}{F_p^{0.36}} \qquad (4\text{-}45)$$

where F_p is the flowrate of any stream ($CS(n,i), F(i',i), CR(i,n)$)

$$\text{where } N_p = \frac{188.318 \rho^{0.536} \mu^{0.2} L}{\pi^{1.8}} \qquad (4\text{-}46)$$

The CPA from the paper of Kim and Smith (2003) is adapted to select the cooling water network with minimum pressure drop. The authors used the superstructure shown in Figure 25(a). The superstructure is based on a single source cooling water network. By modifying the superstructure for a single source cooling water systems, a multiple sources superstructure is shown in Figure 25(b). The CPA used by Kim and Smith (2003) is based on finding a path from source to sink with maximum pressure drop. The maximum pressure drop path is then minimized during optimization to obtain the network with minimum pressure drop. Constraint (4-47) is used to identify the maximum pressure drop path between the source and the sink.

$$P_m - P_n \geq \Delta P_{mn} \qquad (4\text{-}47)$$

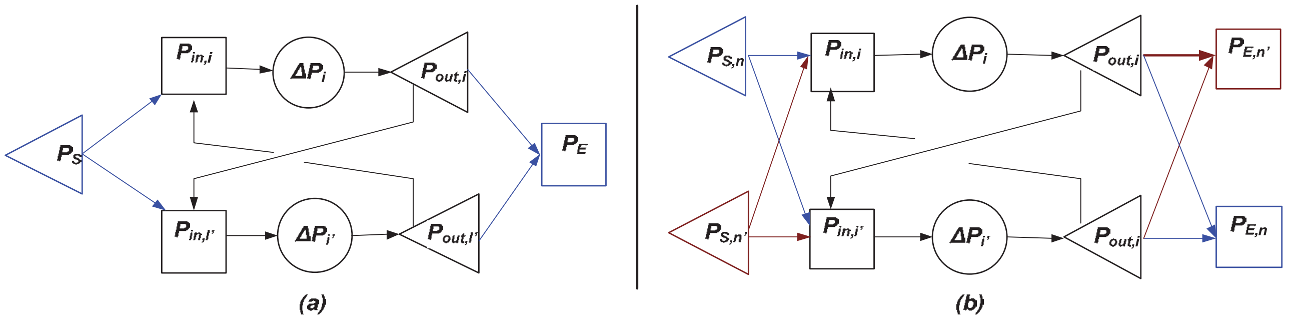

***Figure 25**: Cooling water system superstructure; (a) Single source (b) Multiple sources*

To cater for multiple sources and sinks, the superstructure in Figure 25(b) is modified by using single imaginary source and sink as shown in Figure 26. Constraint (4-48) is then used to define the pressure of source node *n* from the imaginary source node.

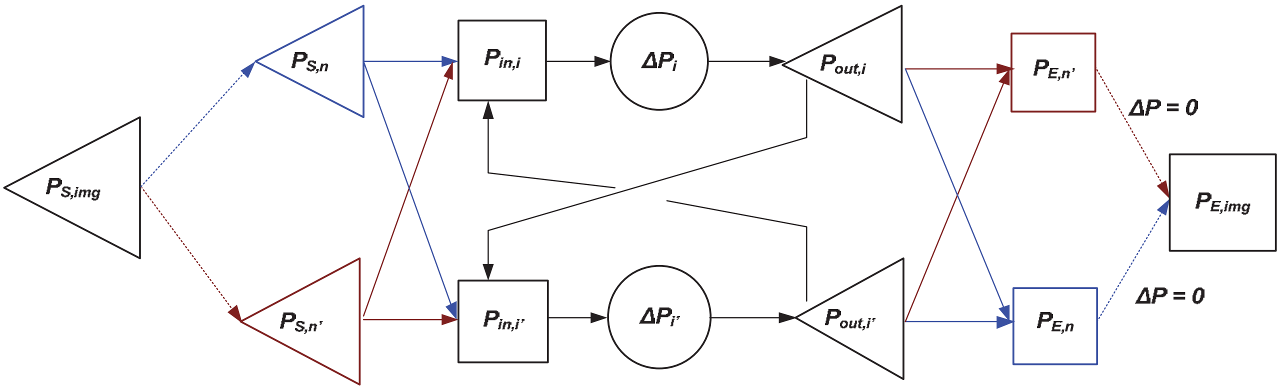

Figure 26: Multiple sources cooling water system superstructure

Constraints (4-49) to (4-52) represent the CPA adapted from Kim and Smith (2003). Constraint (4-53) defines the pressure at the imaginary sink node. From this equation the imaginary sink node will assume a value from all sink nodes with minimum pressure thus identifying a path with maximum pressure drop. The pressure drop of this critical path is then minimized to synthesize a cooling water network with minimum pressure drop.

$$P_{S,img} - P_{S,n} = \Delta P \quad (4\text{-}48)$$

$$P_{S,n} - P_{in,i} + LV(1 - x(n,i)) \geq \Delta P_{n,i} \quad (4\text{-}49)$$

$$P_{out,i'} - P_{in,i} + LV(1 - y(i',i)) \geq \Delta P_{i',i} \quad (4\text{-}50)$$

$$P_{in,i} - P_{out,i} = \Delta P_i \quad (4\text{-}51)$$

$$P_{out,i} - P_{E,n} + LV(1 - z(i,n)) \geq \Delta P_{i,n} \quad (4\text{-}52)$$

where, x, y and z are a binary variables indicating the existence of a stream from any source n/ operation i' to operation i /sink n. LV is a large value.

$$P_{E,n} - P_{E,img} \geq \Delta P_{n,img} \quad (4\text{-}53)$$

The network topology with minimum pressure drop is then synthesized by minimizing the objective function shown in constraint (3-54). The expression in constraint (3-54) also minimizes the slack variable which is used to relax the targeted circulating water flowrate (*CW*). Other parameters in constraint (3-54) are used to make the expression dimensionally consistent.

$$OB = (Cost_{power})\left(CW/_{1000}\right)(\Delta P)(1hr) + (Cost_{water})(CW)(3600\,s) \quad (4\text{-}54)$$

where:

$$\left(Cost_{power}\right): \frac{c.u.}{kWh}$$

$$\left(CW\!\big/_{1000}\right): \frac{m^3}{s}$$

$$\left(\Delta P\right): kPa$$

$$\left(Cost_{water}\right): \frac{c.u.}{l}$$

$$\left(CW\right): \frac{kg}{s}$$

The debottlenecking model with pressure drop consists of constraints (4-20)-(4-30) and (4-32)-(4-54). This formulation consists of binary terms and bilinear terms thus rendering the model a MINLP. The objective function for this model is given in constraint (4-54). The detailed solution procedure outlined in Section 4.4.2.1 is used to address this problem.

4.3.1.2 Case II

In this case there are dedicated source and sink for any cooling water using operation. This implies that no pre-mixing or post-splitting of cooling water return is allowed. A set of heat exchanger can only be supplied by one cooling tower. The mass and energy balance constraints in Case I are included in Case II. To control sources and sinks in each cooling water using operation, constraints (4-55)-(4-60) are incorporated. Constraints (3-55) and (3-56) prevent pre-mixing. Constraint (4-56) ensures that cooling water using operation *i* can only be supplied by a maximum of one cooling tower *n*.

$$CS(n,i) \leq F_{in}^{u}(i)x(n,i) \qquad \forall i \in I \qquad \forall n \in N \qquad (4\text{-}55)$$

$$\sum_{n \in N} x(n,i) \leq 1 \qquad \forall n \in N \qquad (4\text{-}56)$$

Post-splitting is prevented by constraints (4-57) and (4-58). Constraint (4-58) ensures that cooling water using operation *i* can supply a maximum of one cooling tower *n*.

$$CR(i,n) \leq F_{in}^{u}(i)z(i,n) \qquad \forall i \in I \qquad \forall n \in N \qquad (4\text{-}57)$$

$$\sum_{n \in N} z(i,n) \leq 1 \qquad \forall i \in I \quad \forall n \in N \qquad (4\text{-}58)$$

Constraints (4-59) and (4-60) ensure that the source and the sink cooling water supply is the same for a particular cooling water using operation.

$$x(i,n) \leq z(n,i) + \left(2 - \sum_{n \in N} x(n,i) - \sum_{n \in N} z(i,n) \right) \qquad (4\text{-}59)$$

$$z(i,n) \geq x(n,i) - \left(2 - \sum_{n \in N} x(n,i) - \sum_{n \in N} z(i,n) \right) \qquad (4\text{-}60)$$

The targeting model consists of constraints (4-21)-(4-30), (4-32)-(4-34) and constraints (4-55)-(4-60). The formulation has bilinear terms and binary variables thus rendering the model MINLP. The model is solved by minimizing circulating water flowrate (*CW*) given in constraint (4-21).

The debottlenecking model with pressure drop consists of constraints (4-20)-(4-30) and (4-32)-(4-60). The formulation also consists of binary terms and bilinear terms thus rendering the model MINLP. The objective function for this model is given in constraint (4-54).

The MINLP models exhibit multiple local optimum solutions and they are also generally difficult to solve because the starting point might yield suboptimal or infeasible results. Thus, it is important to obtain a good starting point before solving the exact MINLP problem. Similar to Case I, the model was linearized using the linearization relaxation procedure by Quesada and Grossmann (1995). The solution procedure outlined in Section 4.4.2.1 is used to address this problem.

Solution Procedure

4.3.1.3 Cooling water network

The solution procedure involves linearization of bilinear terms and using resultant MILP model as a starting point for the exact MINLP model. The bilinear terms in the cooling water network model which appear in constraints (4-29)-(4-31) are linearized using the Reformulation-Linearization technique by Sherali and Alameddine (1992) as shown in the paper of Gololo and Majozi (2011). However, a piecewise linearization technique is used to linearize the pressure drop correlations in constraints (4-42) and (4-45).

Piecewise linearization

The functions shown in constraints (4-42) and (4-45) are first plotted within the operating range of the heat exchanger and the piecewise linearization is then used to approximate the nonlinear function with a linear function (Kim and Smith, 2003). Figure 27 shows an illustration of piecewise linearization. The operating range the heat exchanger is divided into two regions which are defined by separate straight line equation. The operating range of the heat exchangers where divided into two boundaries and the linear expressions for the two regions of each heat exchanger pressure drop correlation were derived.

The binary variable is assign to each equation using maximum and minimum mass flowrate values as shown in constraints (4-61)-(4-62).

$$f_j \geq f_j^{\min} b_j \qquad j = 1,2 \qquad (4\text{-}61)$$

$$f_j \leq f_j^{\max} b_j \qquad j = 1,2 \qquad (4\text{-}62)$$

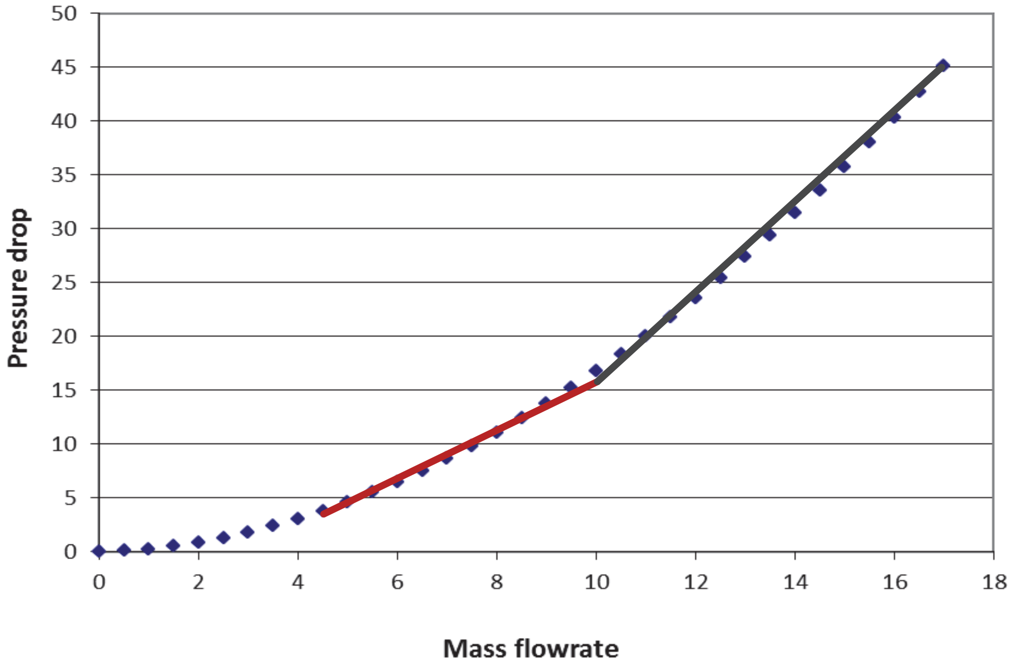

***Figure 27**: Illustration of piecewise linearization technique (Kim and Smith, 2003)*

Constraint (4-63) is used to ensure that only one region is active.

$$\sum_{j} b_j = 1 \qquad j = 1,2 \qquad (4\text{-}63)$$

The pressure drop which is a function of f_j is then calculated from constraint (3-64)

$$\Delta P_j = m\left(f_j\right) + c \qquad j = 1,2 \qquad (4\text{-}64)$$

where *m* is the gradient of the straight line and *c* is the y-intercept.

<u>*Reformulation linearization technique*</u>

The technique starts by assigning a variable to all bilinear terms as shown below.

Let,

$$crt(i,n) = CR(i,n)T_{out}(i) \quad \forall n \in N, i \in I$$

$$frt(i',i) = FR(i',i)T_{out}(i) \quad i \in I, i' \in I$$

$$fnt(i) = F_{in}(i)T_{out}(i) \quad \forall i \in I$$

$$tcs(n,i) = CS(n,i)T_s(n) \quad \forall n \in N, i \in I$$

The upper and the lower bounds for variables in each bilinear term were defined as follows: The lower bound for the flowrates is zero and the upper bound was given a value. The lower bound for the temperatures is the wet bulb temperature and the upper bound was also assigned a value. The bilinear term $CR(i,n)T_{out}(i)$ can now be defined by constraints (4-65) to (4-68).

$$\begin{aligned} crt(i,n) \geq\ & OS^{u}(n)T_{out}(i) + CR(i,n)T_{out}^{u}(i) \\ & - OS^{u}(n)T_{out}^{u}(i) \end{aligned} \qquad \forall n \in N, \qquad \forall i \in I \qquad (4\text{-}65)$$

$$crt(i,n) \geq CR(i,n)T_{out}^{L}(i) \qquad \forall n \in N, \qquad \forall i \in I \qquad (4\text{-}66)$$

$$\begin{aligned} crt(i,n) \leq\ & OS^{u}(n)T_{out}(i) + CR(i,n)T_{out}^{L}(i) \\ & - OS^{u}(n)T_{out}^{u}(i) \end{aligned} \qquad \forall n \in N, \forall i \in I \qquad (4\text{-}67)$$

$$crt(i,n) \leq CR(i,n)T_{out}^{u}(i) \qquad \forall n \in N, \forall i \in I \qquad (4\text{-}68)$$

Similarly, the bilinear term $FR(i',i)T_{out}(i)$ is defined by constraints (4-69) to (4-72), the bilinear term $F_{in}(i)T_{out}(i)$ by constraints (4-73) to (4-76) and the bilinear term $CS(n,i)T_{s}(n)$ by constraints (4-77) to (4-80).

$$\begin{aligned} frt(i',i) \geq\ & F_{in}^{u}(i)T_{out}(i') + FR(i',i)T_{out}^{u}(i) \\ & - F_{in}^{u}(i)T_{out}^{u}(i) \end{aligned} \qquad i \in I, i' \in I \qquad (4\text{-}69)$$

$$frt(i',i) \geq FR(i',i)T_{out}^{L}(i) \qquad i \in I, i' \in I \qquad (4\text{-}70)$$

$$\begin{aligned} frt(i',i) \leq\ & F_{in}^{u}(i)T_{out}(i') + FR(i',i)T_{out}^{L}(i) \\ & - F_{in}^{u}(i)T_{out}^{L}(i) \end{aligned} \qquad i \in I, i' \in I \qquad (4\text{-}71)$$

$$frt(i',i) \leq FR(i',i)T_{out}^{u}(i) \qquad i \in I, i' \in I \qquad (4\text{-}72)$$

$$fnt(i) \geq F_{in}^{u}(i)T_{out}(i) + F_{in}(i)T_{out}^{u}(i) - F_{in}^{u}(i)T_{out}^{u}(i) \qquad \forall i \in I \qquad (4\text{-}73)$$

$$fnt(i) \geq F_{in}(i)T_{out}^{L}(i) \qquad \forall i \in I \qquad (4\text{-}74)$$

$$fnt(i) \leq F_{in}^{u}(i)T_{out}(i) + F_{in}(i)T_{out}^{L}(i) - F_{in}^{u}(i)T_{out}^{L}(i) \qquad \forall i \in I \qquad (4\text{-}75)$$

$$fnt(i) \leq F_{in}(i)T_{out}^{u}(i) \qquad \forall i \in I \qquad (4\text{-}76)$$

$$\begin{aligned} tcs(n,i) \geq\ & OS^{u}(n)T_{s}(n) + CS(n,i)T_{s}^{u}(n) \\ & - OS^{u}(n)T_{s}^{u}(n) \end{aligned} \qquad \forall n \in N, i \in I \qquad (4\text{-}77)$$

$$tcs(n,i) \geq CS(n,i)T_{s}^{L}(n) \qquad \forall n \in N, i \in I \qquad (4\text{-}78)$$

$$\begin{aligned} &tcs(n,i) \geq OS^{u}(n)T_{s}(n) + CS(n,i)T_{s}^{L}(n) \\ &- OS^{u}(n)T_{s}^{L}(n) \end{aligned} \qquad \forall n \in N \;\; i \in I \;, \tag{4-79}$$

$$tcs(n,i) \geq CS(n,i)T_{s}^{u}(n) \qquad \forall n \in N \;\; i \in I \;, \tag{4-80}$$

Introduction of linearization variables require constraint (4-28), constraint (4-29) and constraint (4-31) to be modified as shown in constraint (4-81), constraint (4-82) and constraint (4-83), respectively.

$$\sum_{i \in I} tcs(n,i) = M(n)T_{amb} + OS(n)T_{ctout}(n) \qquad \forall n \in N \tag{4-81}$$

$$T_{ret}(n) = \frac{\sum_{i \in I} crt(i,n)}{\sum_{i \in I} CR(i,n)} \qquad \forall n \in N \tag{4-82}$$

$$Q(i) + c_{p} \sum_{n \in N} tcs(n,i) + c_{p} \sum_{i' \in I} frt(i',i) = fnt(i)c_{p} \quad \forall i \in I \tag{4-83}$$

4.3.1.4 Cooling tower model

The cooling tower model predicts the outlet cooling water conditions given the inlet conditions. This is done by first assuming the outlet water temperature of a cooling tower. The assumption is done by subtracting 0.5 ^{o}C from the given cooling tower inlet temperature. The three governing mass and heat transfer equations, i.e. equations (4-2), (4-4) and (4-14) are then solved numerically using forth order Runge_Kutta method starting from the bottom of the cooling tower moving upwards at stepsize Δz. When the maximum height is reached, the temperature at this point will be compared with the inlet water temperature. If the two agree within a specified tolerance, the cooling tower model will stop and the outlet temperature will be given as the assumed temperature, else the inlet temperature will be adjusted until the inlet water temperature agrees with the calculated temperature. The solution algorithm is shown in Figure 28.

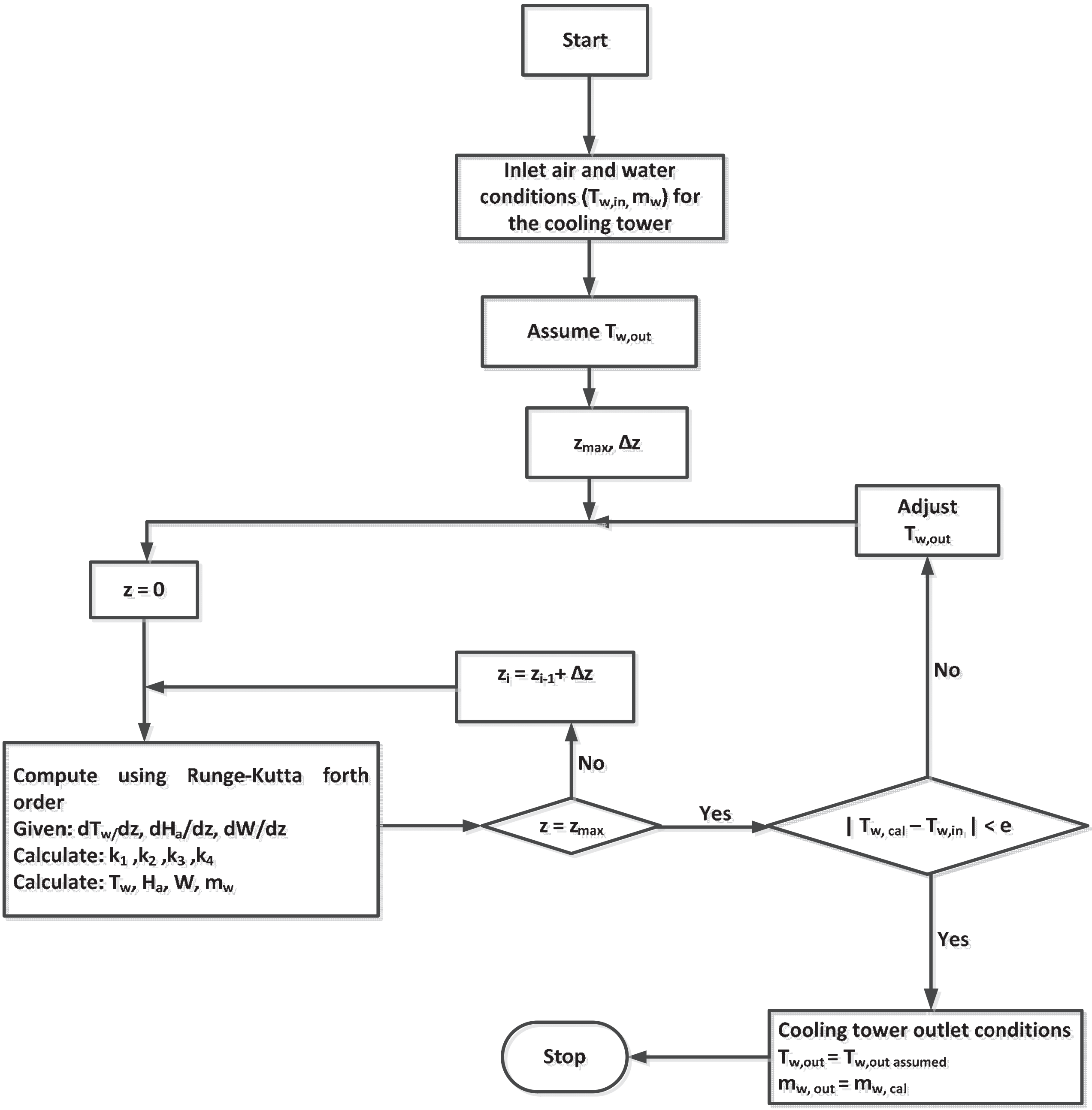
Start
Inlet air and water conditions ($T_{w,in}$, m_w) for the cooling tower
Assume $T_{w,out}$
z_{max}, Δz
Adjust $T_{w,out}$
z = 0
$z_i = z_{i-1} + \Delta z$
No
No
Compute using Runge-Kutta forth order
Given: dT_w/dz, dH_a/dz, dW/dz
Calculate: k_1, k_2, k_3, k_4
Calculate: T_w, H_a, W, m_w
$z = z_{max}$
Yes
$| T_{w, cal} - T_{w,in} | < e$
Yes
Stop
Cooling tower outlet conditions
$T_{w,out} = T_{w,out\ assumed}$
$m_{w, out} = m_{w, cal}$

Figure 28: *Solution procedure for cooling tower model*

4.3.1.5 Overall cooling water system

Optimizing of the overall cooling water system requires the simultaneous solving of both cooling water network and the cooling tower model. The procedure starts by targeting the overall circulating cooling water flowrate with no inclusion of pressure drop constraints. The results from the targeting model are the flowrate and inlet temperature for each cooling tower. The cooling tower model is then used to calculate the outlet temperature, flowrate and evaporation for each cooling tower. These conditions are then used as the inputs to the cooling water network model with pressure drop constraints. Using the target the model with pressure drop constraints is solved by minimizing the overall pressure drop. There is an iterative process between the cooling water network model and the cooling tower model as shown in Figure 29. If the difference between the outlet temperature of the cooling tower model and the previous inlet temperature to the cooling water network is less than 1^{o}C, the model will stop and the final results will be obtained else the iterative process continue. The cooling tower model was solved using MATLAB while the cooling water network was solved using GAMS platform. The procedure used to link MATLAB and GAMS was presented by Ferris (2005).

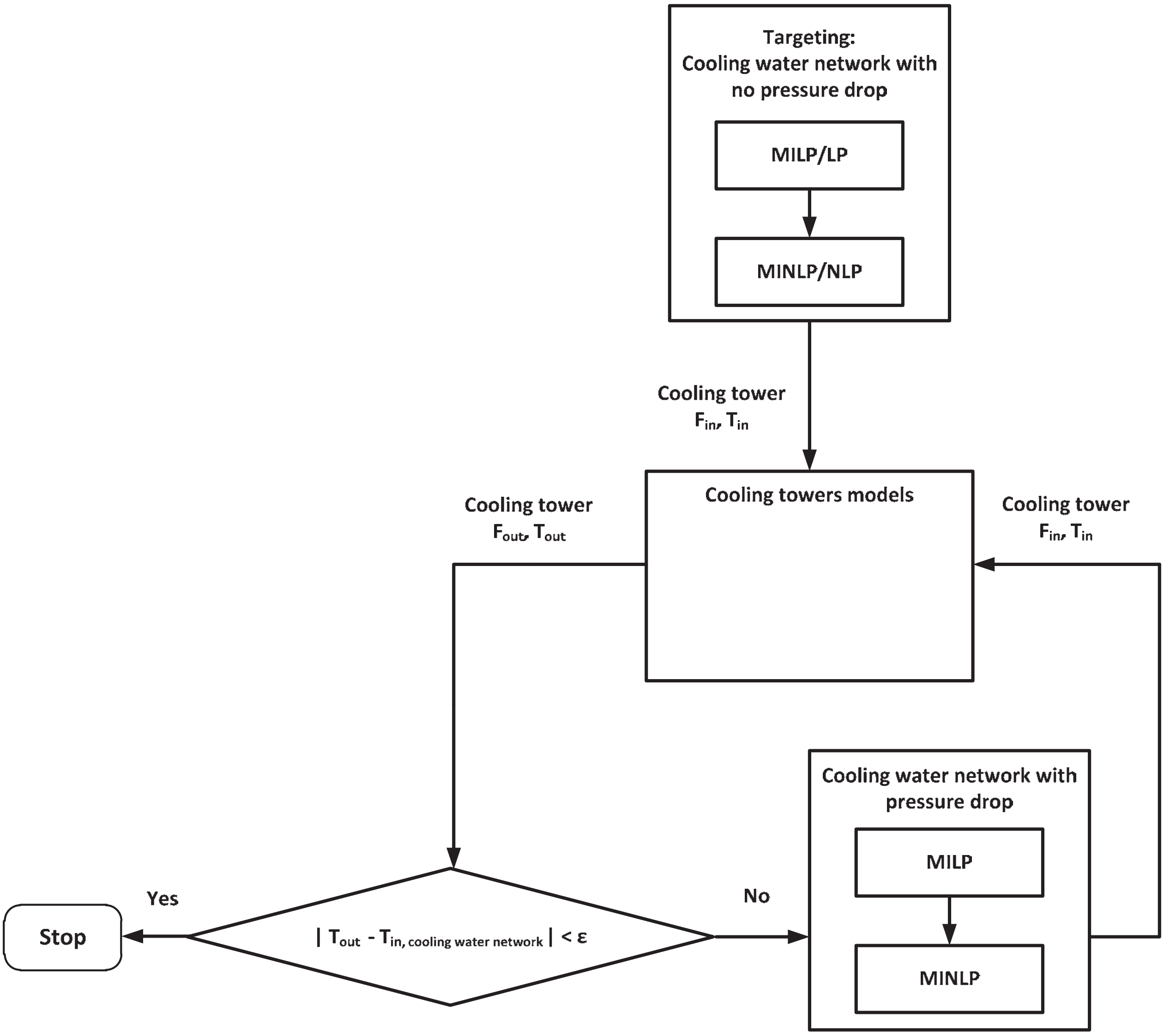

***Figure 29**: Solution procedure for cooling water system model*

4.4 Debottlenecking the cooling water systems with no considering pressure drop

Gololo and Majozi (2011) improved the method developed by Majozi and Moodley (2008) by adding the detailed cooling tower model to study the interaction of the cooling water network and the cooling towers. The authors used two different platforms to model the cooling towers and the cooling water network. The cooling tower model was built in MATLAB whilst the cooling water network model was built in GAMS. An iterative procedure was used to link the two platforms thus allowing an interaction between the two components. Although this approach seems holistic, global optimality cannot be guaranteed because two platforms were used to model the overall different components of the cooling water systems.

This section presents an integrated platform for synthesis and optimization of cooling water systems. A mathematical model which include the cooling water network and the detailed cooling tower model presented in above is built in gPROMS software. The model allows for cooling water reuse within the network. The outlet cooling towers water conditions is simultaneously calculated for each cooling water return conditions. The overall effectiveness of the cooling towers is evaluated to study the impact of cooling water network on cooling towers. In this work pressure drop constraints were excluded. The model presented in section 4.3 is slightly modified by introducing the following constraints:

The individual reuse and return streams from any cooling water using operation are defined explicitly using constraints (4-84) and (4-85).

$$CR(i,n) = F_{out}(i)\, fret_{tot}(i)\, fret(i,n) \qquad \forall i \in I\ \forall n \in N \qquad (4\text{-}84)$$

$$FR(i,j) = F_{out}(i)\left(1 - fret_{tot}(i)\right) freu(i,j) \qquad \forall i \in I \qquad (4\text{-}85)$$

where $fret_{tot}(i)$ is the fractional amount of return cooling water from operation *i* and $fret(i,n)$ is the fractional amount return cooling water from operation *i* to cooling tower *n* as shown in Figure 30 . $freu(i,j)$ is the fractional amount of reuse cooling water from operation *i* to operation *j*.

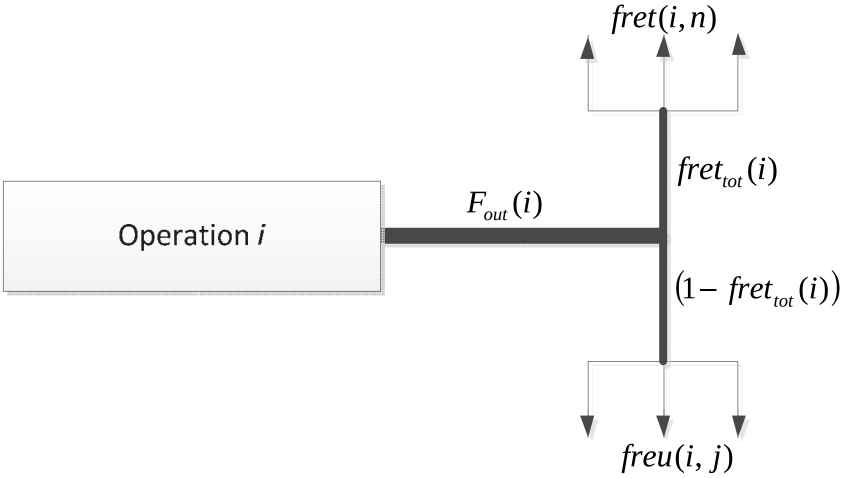

***Figure 30**: Depiction of variables used in the modified model*

Constraint (4-86) ensures that the sum of all fractions for return streams from operation *i* to any cooling tower *n* add up to 1. Similarly, constraint (4-87) ensures that the sum of all fractions for the reuse streams from operation *i* to any operation *j* add up to 1.

$$\sum_{n=1}^{N} fret(i,n) = 1 \qquad \forall i \in I \qquad (4\text{-}86)$$

$$\sum_{j=1}^{J} freu(i,j) = 1 \qquad \forall i \in I \qquad (4\text{-}87)$$

The fractions for the reuse streams are defined using the transformation shown in constraints (4-88) and (4-89).

$$freu(i,1) = \sin^2(z(i,1) \qquad \forall i \in I \qquad (4\text{-}88)$$

$$freu(i, j \Rightarrow 2 : N) = \sin^2(z(i, j)\prod\cos^2(z,1 : j-1) \qquad \forall i \in I \qquad (4\text{-}89)$$

where z is between 0 and 2π.

The inlet and the outlet cooling water flowrate to cooling tower n is defined by constraints (4-90) and (4-91) respectively.

$$OS_{in}(n) = \sum_{n \in N} CS(n,i) - M(n) \qquad \forall n \in N \qquad (4\text{-}90)$$

$$OS_{out}(n) = \sum_{i \in I} CR(i,n) - B(n) \qquad \forall n \in N \qquad (4\text{-}91)$$

Constraint (4-92) defines the mass balance around the cooling tower n.

$$OS_{out}(n) = OS_{in}(n) - E(n) \quad \forall n \in N \qquad (4\text{-}92)$$

Case I consists of constraints (4-21), (4-24)-(4-34) and (4-84)-(4-92). The mathematical formulation consists of bilinear terms thus rendering the model NLP. The objective function minimizes total circulating cooling water flowrate (*CW*) given in constraint (4-21).

Case II consists of constraints (4-21), (4-24)-(4-36), (4-39), (4-40), (4-56), (4-58)-(4-60) and (4-84)-(4-92). The mathematical formulation has bilinear terms and binary variables thus rendering the model MINLP. The objective function minimizes total circulating cooling water flowrate (*CW*).

4.5 Conclusions

The mathematical formulation for synthesis and optimization cooling water system with multiple cooling towers has been presented. The formulation takes into account the cooling tower performance, the equipment and the cooling water network pressure drop. Two practical cases were considered when developing the heat exchanger network model and the formulations yield mixed integer nonlinear programming (MINLP). In the event where pressure drop was ignored the formulation in Case I yield an NLP model whilst Case II yield a MINLP.

5 CASE STUDIES

5.1 Introduction

In this section the application of the proposed technique is demonstrated by considering a case study taken from the paper by Majozi and Moodley (2008). Two practical cases which were illustrated in Chapter 3 are considered.

5.2 Base case

The cooling water system in Figure 31 shows a set of heat exchanger networks which are supplied by a set of cooling towers. Each cooling water using operation is supplied by fresh water from the cooling tower and return back to the cooling tower. The implication of these arrangements results in higher return cooling water flow rate and low return cooling water temperature thus reducing cooling tower efficiency (Bernier, 1994).

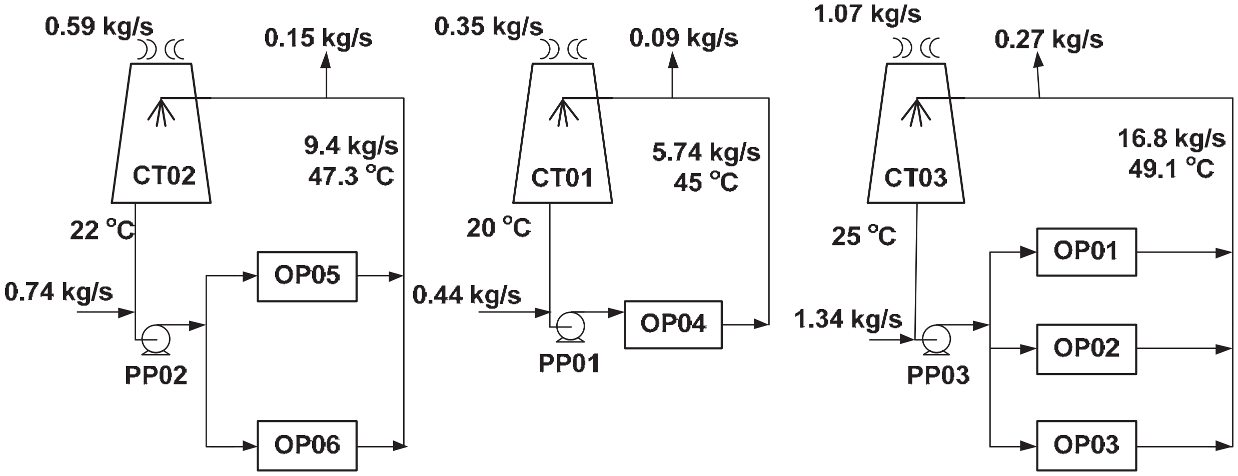

Figure 31: *Base case*

The heat duties, temperature limits and design information are shown in Table 4 and Table 5. T_{ret}^{u} is the maximum allowable temperature for packing inside the cooling towers while OS^{u} is the maximum flowrate of the cooling tower before flooding. T_{in}^{u} and T_{out}^{u} are the thermodynamic temperature limits for the inlet and outlet temperature of the cooling water using operation respectively. The total circulating water flowrate is 31.94 kg/s and the overall cooling towers effectiveness is 90%.

***Table 4**: Cooling towers design information*

Cooling towers	$T_{ret}^{u} (^{o}C)$	$OS^{u} (kg/s)$	$\frac{Ka_{fi}}{m_w} = \frac{a_d}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{d_{da}} ATD^{b_{db}}$
CT01	50	9.6	$\frac{Ka_{fi}}{m_w} = \frac{2.49}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{-0.67} 0.609^{-0.062}$
CT02	50	16	$\frac{Ka_{fi}}{m_w} = \frac{1.664}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{-0.62} 0.914^{-0.27}$
CT03	55	20	$\frac{Ka_{fi}}{m_w} = \frac{2.49}{A_{fr}}\left(\frac{m_w}{m_a}\right)^{-0.67} 0.914^{-0.062}$
Height = 2.438, Area = 5.943, CC = 5			

***Table 5**: Limiting cooling water data*

Operations	$T_{in}{}^{u}$ (C)	$T_{out}{}^{u}$ (C)	F_{in} (kg/s)	Q(i)(kW)
OP01	30	45	9.52	600
OP02	40	60	3.57	300
OP03	25	50	7.62	800
OP04	45	60	7.14	600
OP05	40	55	4.76	300
OP06	30	45	11.1	700

5.3 Debottlenecking the cooling water systems considering pressure drop

Case I

The optimization of the cooling water network was performed in the GAMS platform using DICOPT solver. CPLEX solver was used for MILP subproblems and MINOS5 was used to solve NLP subproblems. The model consists of 434 constraints, 276 continuous variables and 72 discrete variables. Figure 32 shows synthesized cooling water system after the application of the proposed technique. The total circulating cooling water decreased by 26% due to the exploration of cooling water reuse opportunities. Furthermore, the cooling water system can be operated with two cooling towers instead of three. This shows a potential for capital cost savings. The proposed methodology does not only debottleneck the cooling water system but also generate the network topology with the least pressure drop. The pressure drop between sources and sinks $\Delta P_{S2,E2}$ and $\Delta P_{S3,E3}$ is 38 kPa and 37 kPa respectively.

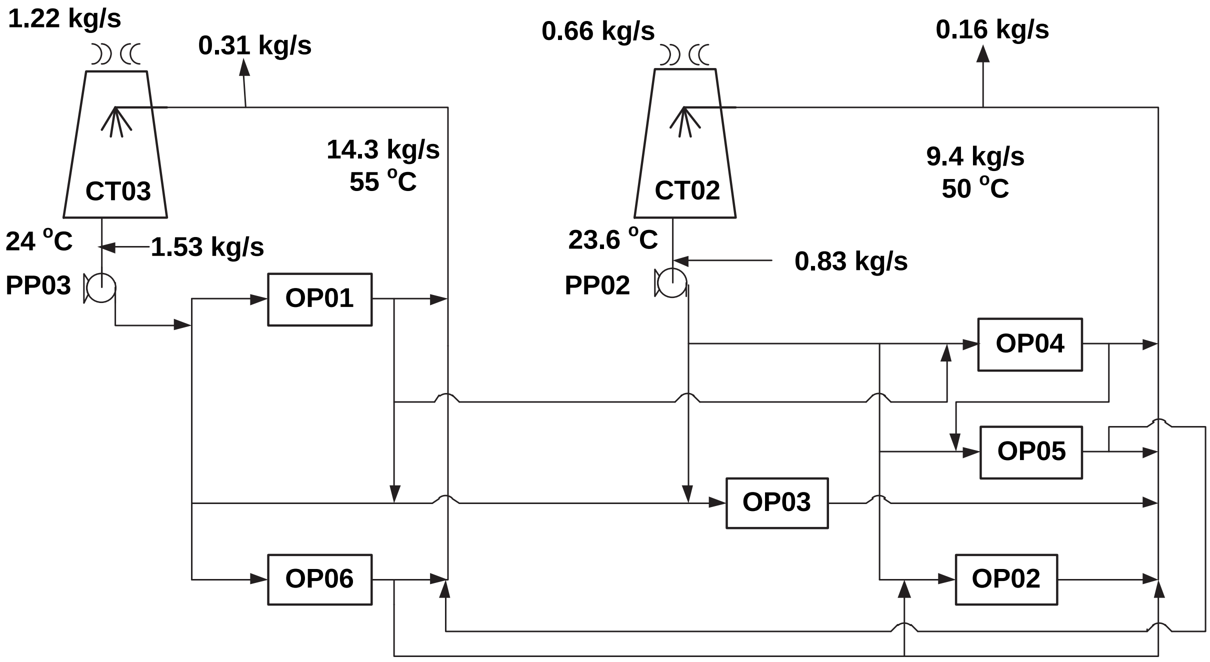

Figure 32: Debottlenecked cooling water system with the minimum pressure drop

The use of cooling tower model presents an important opportunity for evaluating the interaction between the cooling water network and the cooling towers. In this case the overall increase in cooling tower return temperature which was associated with a decrease in the overall circulating water flowrate resulted in a 3% improvement in effectiveness.

Case II

In this case there is a dedicated source and sink for any cooling water using operation. The DICOPT solver was used in the GAMS platform to solve the MINLP model. The MILP and NLP subproblems were solved using CPLEX and MINOS5 solvers respectively. The model consists of 494 constraints, 288 continuous variables and 84 discrete variables.

The total circulating cooling water flowrate decreased by 26% due to the exploration of reuse opportunities and the overall effectiveness improved by 5%. Figure 33 shows the debottlenecked cooling water system with the least pressure drop. The cooling water from OP 01 is reused in OP 06 and OP 02 as shown in Figure 4-7. OP 05 uses fresh water and reuse cooling water from OP 06. This suggests that by allowing interaction between various cooling water networks, the cooling water system could be debottlenecked. The pressure drop between sources and sinks $\Delta P_{S1,E1}$, $\Delta P_{S2,E2}$ and $\Delta P_{S3,E3}$ is 24 kPa, 38 kPa and 49 kPa respectively. The proposed technique also offers the opportunity for a designer to size the required pump capacity for each cooling tower.

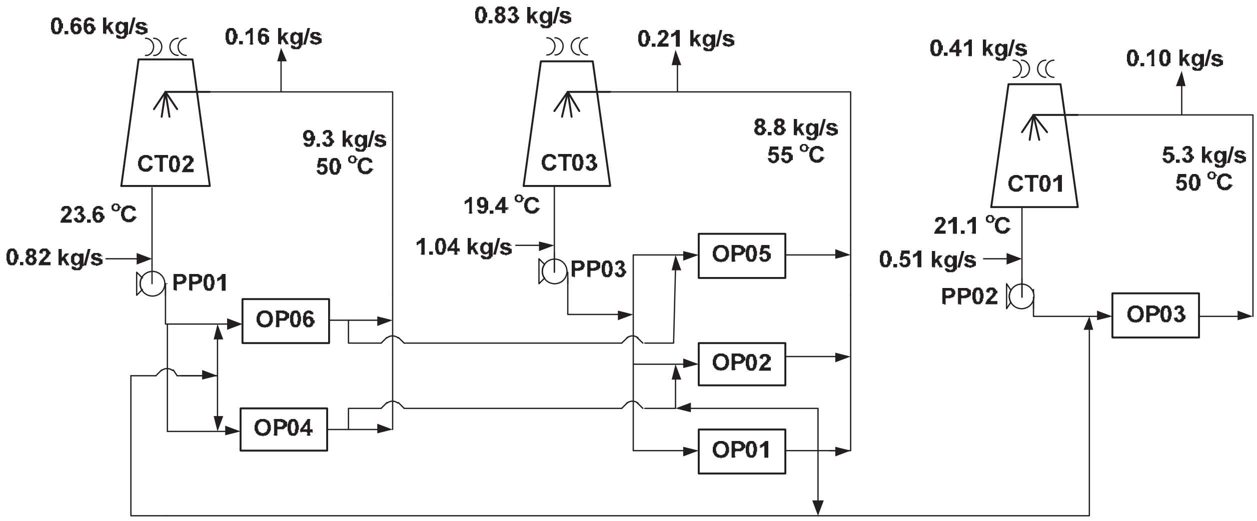

Figure 33: Debottlenecked cooling water system with the minimum pressure drop

The overall effectiveness for multiple cooling towers

The overall effectiveness was calculated based on the procedure outlined in Chapter 3. Table 6 shows an increase in the overall effectiveness of the cooling towers when applying the proposed design methodology. An increase in the overall cooling towers effectiveness is attributed to a decrease in the overall circulating water and an increase in return water temperature.

Table 6: Effectiveness for base Case, Case I and Case II

	Effectiveness	Circulating water flowrate(kg/s)
Base case	0.90	31.94
Case I	0.93	23.7
Case II	0.95	23.4

5.4 Integrated approach

This section presents the results in which one platform was used to debottleneck the cooling water systems. An integrated model which includes the cooling water network and the cooling tower was developed using gPROMS platform.

The proposed model is applied to a base case study shown above. In this case the model is solved without considering pressure drop. The NLP models were solved using SRQPD solver and the MINLP models were solved using OAERAP solver. The flowrate in Case I was reduced to 22.7 kg/s. This equates to 29% decrease in circulating water flowrate. In Case II the flowrate is reduced by 27%. The cooling water system network for Case I and II are shown in Figure 34 and Figure 35 respectively.

The overall effectiveness for Case I is 95.3%. This implies 5.3% improvement compared to the base case. Case II has 5% improvement in effectiveness.

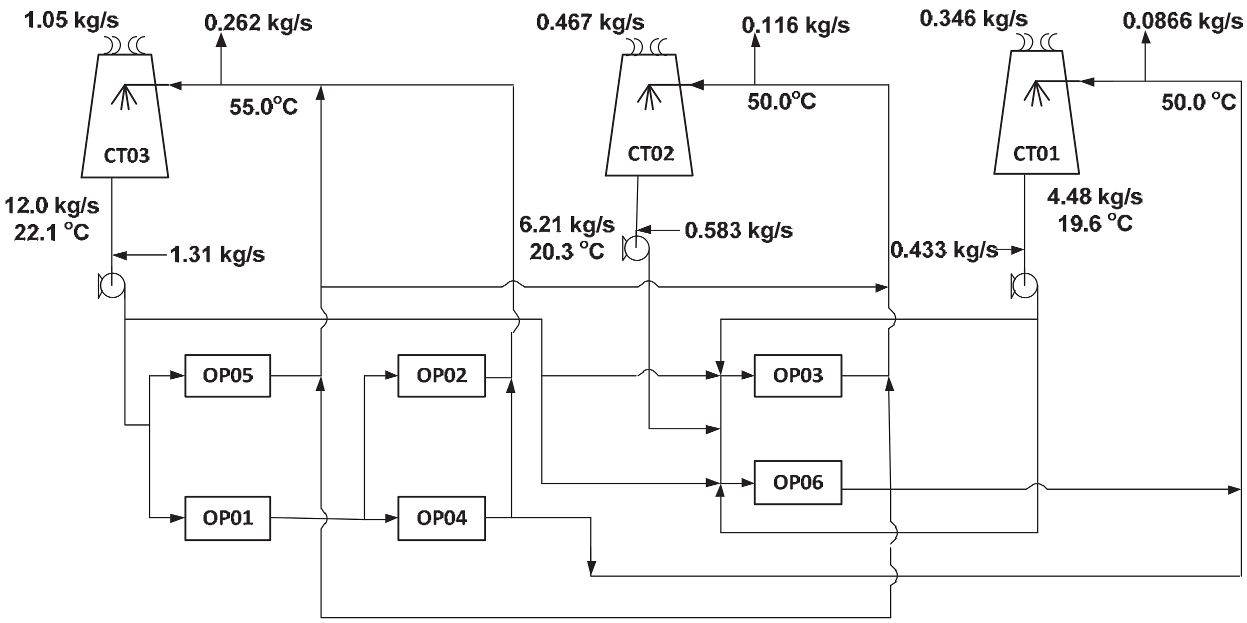

Figure 34: Cooling water system for Case I

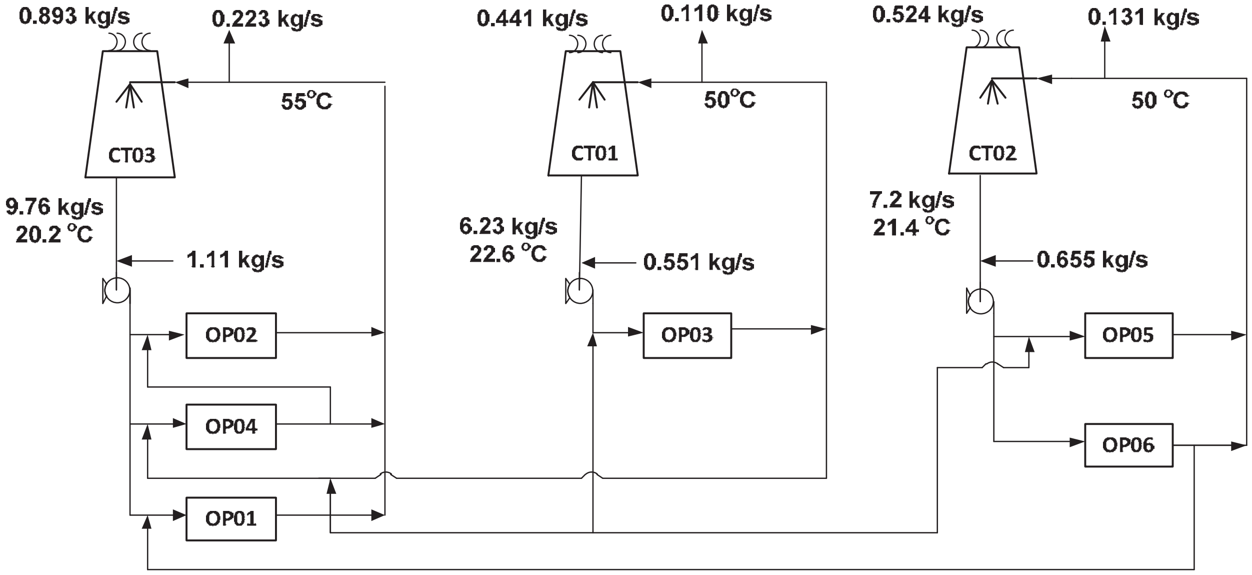

***Figure 35**: Cooling water* system *for Case II*

Gololo and Majozi (2011) technique achieved a maximum of 22% reduction in circulating water flowrate. In this work a maximum of 29% reduction in circulating water flowrate was achieved. The main difference between this work and the work by Gololo and Majozi (2011) is the solution procedure. Gololo and Majozi (2011) used two separate platforms to build a cooling water network and the cooling towers. However, in this work a single platform was used to build both cooling water network model and the cooling tower model. This suggests that it is better to build an integrated optimization model in one platform.

5.5 Conclusions

Case studies to demonstrate the proposed technique were presented. The results obtained using this technique show that by exploring the opportunities for cooling water reuse the cooling tower can be debottlenecked. This implies that a given set of cooling towers can manage an increased heat load. The proposed technique can also improve the overall effectiveness for the cooling towers.

6 CONCLUSIONS

The mathematical technique for optimization and synthesis of cooling water systems with multiple cooling towers has been presented. This technique is more holistic because it caters for the effect of cooling tower performance on heat exchanger network. The cooling tower thermal performance is predicted using the mathematical model. The results obtained using this technique are more practical, since all components of the cooling water system are included in the analysis.

6.1 Debottlenecking the cooling water systems considering pressure drop

The mathematical model for synthesis and optimization of cooling water systems with multiple cooling sources which takes into account the pressure drop is presented. The proposed technique is based on the CPA and the superstructural approach. The detailed mathematical model consisting of the cooling towers and the cooling water networks model was developed considering two practical cases. Case I involves a cooling water system with no dedicated cooling water sources and sinks. This implies that a set of heat exchangers can be supplied by any cooling tower and return the cooling water to any cooling tower. Case II involves a cooling water system with dedicated cooling water sources and sinks. This implies that a set of heat exchangers can only be supplied by one cooling tower. No pre-mixing or post-splitting of cooling water return is allowed. However, reuse of water within the network is still allowed. The formulations for both cases yield mixed integer nonlinear programming (MINLP) structure. Piecewise linearization and reformulation-relaxation technique were used to get a good starting point for solving the exact MINLP model.

The case studies showed a 26% decrease in circulating water flowrate due to the exploitation of reuse opportunities. The return cooling tower temperatures were increased with a decrease in circulating water flowrate. This resulted in 3% and 5% improvement in the overall effectiveness in Case I and II respectively. The synthesized cooling water networks have a maximum pressure drop of 38 kPa and 49 kPa in Case I and II respectively. The proposed technique offer the opportunity to debottleneck cooling water systems with multiple cooling towers while maintaining minimum pressure drop and maximizing the cooling tower effectiveness.

6.2 Integrated approach

An integrated platform for synthesis and optimization of cooling water systems with multiple cooling sources is also presented. Two practical scenarios as described above are considered. The mathematical formulations for Case I and II yield NLP and MINLP structures respectively.

The model was built in gPROMS software and the results showed 29% and 27% decrease in circulating water flowrate for Case I and II respectively. This is 7% better than the results where MATLAB and GAMS were used. This suggests that it is better build an integrated model in a single platform.

6.3 Overall effectiveness for multiple cooling towers

A maximum of 5% improvement in overall effectiveness was realized. This was mainly due to a decrease in circulating water flowrate with an increase in return water temperature.

7 RECOMMENDATIONS

Although the proposed methodology explores the interaction between the cooling towers and the cooling water networks, and takes into account the network pressure drop, there are still more features that need to be considered and included in the model as summarized below.

7.1 Environment

Cooling water systems are one of the major consumers of water and generators of effluent in industry. The systems consist of cooling towers which use the mechanism of evaporative cooling to remove heat from the cooling water. Evaporation of water from the cooling towers results in an increase in the concentration of the dissolved solids in the circulating cooling water. Thus, blowdown mechanism is employed to maintain the concentration of the dissolved solids at an allowable level. Makeup water is then added to replace evaporation and blowdown losses. The flowrates for the blowdown and makeup depend on the rate of evaporation loss and the cycles of concentration. Therefore, it is recommended to develop a systematic procedure that will minimize the makeup and blowdown.

7.2 Costs

The proposed technique does not take into account the total capital and operating costs of the cooling water systems. Hence, it is recommended that the economic study be included in the proposed technique.

7.3 Integrated approach vs two platform approach

In a two platform approach MATLAB was used to solve the cooling tower model and GAMS was used to optimize the cooling water network. This approach does not optimize the whole cooling water system. However, optimization in an integrated approach simultaneously considers the cooling tower model and the cooling water network model. Hence, an integrated approach gives better results than the two platform approach. It is therefore recommended that cooling water system model should be built using one platform.

8 REFERENCES

ASANTE NDK and ZHU XX (1996) An automated approach for heat exchanger network retrofits featuring minimal topology modifications. Computer and Chemical Engineering **20** 7-12

ATAEI A, GHARAIE M, PARAND R and PANJESHAHI MH (2010) Application of ozone treatment and pinch technology in cooling water system design for water and energy conservation. International Journal of Energy Research **34** 494-506

ATAEI A, PANJESHAHI MH, GHARAIE M and TAHOUNI N (2009b) New method for designing an optimum distributed cooling system for effluent thermal treatment. International Journal of Environmental Research **3(2)** 155-166

ATAEI A, PANJESHAHI MH, PARAND R and TAHOUNI N (2009a) Application of an optimum design of cooling water system by regeneration concept and pinch technology for water and energy conservation. Journal of Applied Science **9(10)** 1847-1858

BAGAJEWICZ MJ, PHAM R and MANOUSIOUTHAKIS V (1998) On the state space approach to mass/heat exchanger network design Chemical Engineering Science 53(14) 2595-2621

BERNIER MA (1994) Cooling tower performance: Theory and experiments. ASHRAE Transactions Research **100** 114-121

CASTRO MM, SONG TW and PINTO JM (2000) Minimization of operational costs in cooling water systems. Trans IChemE **78(Part A)** 192-201

CHEN C and HUNG P (2007) Synthesis of flexible heat exchange networks and mass exchange networks. Computer and Chemical Engineering **31** 1619-1632

COETZEE WAS (2007) *Steam system network synthesis using process integration*. Masters Report, University of Pretoria, Pretoria, South Africa

CORTINOVIS GF, PAIVA JL, SONG TW and PINTO JM (2009b) A systemic approach for optimal cooling tower operation. Energy Conversion and Management **50** 2200-2209

CORTINOVIS GF, RIBIERO MT, PAIVA JL and PINTO JM (2009a) Integrated analysis of cooling water systems: Modeling and experimental validation. Applied Thermal Engineering **29** 3124-3131

COULSON JM and RICHARDSON JF (1996) *Chemical engineering: Fluid flow, heat transfer and mass transfer*. 5th Ed., The Bath Press, Bath University, UK

DONG H, LIN C, and CHANG C (2008) Simultaneous optimization approach for integrated water-allocation and heat-exchanger network. Chemical Engineering Science 63 3664-3678

DOYLE S and SMITH R (1997) Targeting water reuse with multiple contaminants. Institution of Chemical Engineers **75** 181-189

EL-HALWAGI M and MANOUSIOUTHAKIS V (1989) Synthesis of mass-exchange networks. AIChE Journal **35(8)** 1233-1244

EL-HALWAGI MM and MANOUSIOUTHAKIS V (1990a) Automatic synthesis of mass-exchange and regeneration networks. Chemical Engineering Science **45(9)** 2813-2831

EL-HALWAGI MM and MANOUSIOUTHAKIS V (1990b) Simultaneous synthesis of mass-exchange networks with single-component targets. AICHE Journal **36(8)** 1209-1219

FENG X SHEN R and WANG B (2005) Recirculation cooling-water network with an intermediate cooling-water main. Energy and Fuel **19** 1723-1728

FERRIS MC (2005) *MATLAB and GAMS: Interfacing optimization and visualization software*. Computer Sciences Department, University of Wisconsin-Madison, Madison, Wisconsin, USA

FISENKO SP BRIN AA and PETRUCHIK AI (2004) Evaporative cooling of water in a mechanical draft cooling tower. International Journal of Heat and Mass Transfer **47** 165-177

FLOWER JR and LINNHOFF BA (1980) Thermodynamic-Combinatorial Approach to the Design of Optimum Heat Exchanger Networks. AIChE Journal **26(1)** 1-9

FRAUSTO-HERNÁNDEZ S, RICO-RAMÍREZ V, JIMÉNEZ-GUTIÉRREZ A and HERNÁNDEZ-CASTRO S (2003) MINLP synthesis of heat exchanger networks considering pressure drop effects. Computer and Chemical Engineering **27** 1143-1152

GHARAGHEIZI F, HAYATI R and FATEMI S (2007) Experimental study on the performance of mechanical cooling tower with two types of film packing. Energy Conversion and Management **48** 277-280

GOLOLO KV and MAJOZI T (2011) On synthesis and optimization of cooling water systems with multiple cooling towers. Industrial Engineering and Chemistry Research **50(7)** 3775-3787

HOHMANN EC (1971) *Optimum networks for heat exchange*. PhD Thesis, University of Southern California, Los Angeles, California, USA

HUANG S and CHANG C (2012) An improved mathematical programming model for generating optimal heat-exchanger network (HEN) design. Industrial and Engineering Chemistry Research **51** 3508-3515

HUNG SW and KIM JK (2012) Optimization of water systems with the consideration of pressure and pumping. Industrial and Engineering Chemistry Research **51** 848-859

ISAFIADE AJ and FRASER DM (2008) Interval-based MINLP superstructure synthesis of heat exchange networks. Chemical Engineering Research and Design **86(A3)** 245-257

JABER H and WEBB RL (1987) Design of cooling by the effectiveness-NTU method. Journal of Heat Transfer **111** 837-843

KERN D (1950) *Process Heat Transfer* McGraw Hill Book Company

KHAN J-U-R and ZUBAIR SM (2001) An improved design and rating analyses of counter flow wet cooling towers. Journal of Heat Transfer **123(4)** 770-778

KHAN J-U-R, QUERESHI BA and ZUBAIR SM (2004) A comprehensive design and performance evaluation study of counter flow wet cooling towers. International Journal of Refrigeration **27** 914-923

KHAN J-U-R, YAQUB M and ZUBAIR SM (2002) Performance characteristics of counter flow wet cooling towers. Energy Conversion and Management **44** 2073-2091

KIM JK and SMITH R (2001) Cooling water system design. Chemical Engineering Science **56** 3641-3658

KIM JK and SMITH R (2003) Automated retrofit design of cooling waters systems. AiCHE Journal **49(7)** 1712-1730

KIM JK and SMITH R (2004) Cooling system design for water and waste minimization Industrial and Engineering Chemistry Research **43** 608-613

KIM JK, SAVULESCU L and SMITH R (2001) Design of cooling systems for effluent temperature reduction. Chemical Engineering Science **56** 1811-1830

KLOPPER JC and KRÖGER DG (2005a) A critical investigation into the heat and mass transfer analysis of counter flow wet-cooling towers. International Journal of Heat and Mass Transfer **48** 765-777

KLOPPER JC and KRÖGER DG (2005b) The Lewis factor and its influence on the performance prediction of wet-cooling towers. International Journal of thermal Science **44** 879-884

KRÖGER DG (2004) *Air-cooled heat exchangers and cooling towers: mass transfer and evaporative cooling*. Penn Well Corporation, USA

KUO W and SMITH R (1997) Effluent treatment system design. Chemical Engineering Science **52** 4273-4290

KUO W and SMITH R (1998a) Designing for the interactions between water-use and effluent treatment. Transactions of the Institute of Chemical Engineers **76(A)** 287-301

KUO W and SMITH R (1998b) Design of water-using systems involving regeneration. Transactions of the Institute of Chemical Engineers **76(B)** 94-114

LEE JY, CHEN CL, WEN TL, NG DKS, FOO DCY and WANG TC (2013) Synthesis and design of chilled water networks using mathematical optimization. Applied Thermal Engineering **58** 638-649

LEMOUARI M, BOUMAZA M and MUJTABA IM (2007) Thermal performance investigation of a wet cooling tower. Applied Thermal Engineering **27** 902-909

LINNHOFF B and FLOWER JR (1978) Synthesis of Heat Exchanger Networks. AIChE Journal **24(4)** 633-642

LINNHOFF B and HINDMARSH E (1983) The pinch design method for heat exchanger networks. Chemical Engineering Science **38(5)** 745-763

LINNHOFF B, DAVID RM and WARDLE I (1979) Understanding Heat Exchanger Networks. Computer and Chemical Engineering **3** 295-302

LINNHOFF B, TOWNSEND DW, BOLAND D, HEWITT GF, THOMAS BEA, GUY AR and MARSLAND RH (1982) *User guide on process integration for efficient use of energy*. 1st Ed, IChemE

MAJOZI T and MOODLEY A (2008) Simultaneous targeting and design for cooling water systems with multiple cooling water supplies. Computer and Chemical Engineering **32** 540-551

MAJOZI T and NYATHI N (2007) On cooling-water systems design for South African industry: two recent development. South African Journal of Science **103** 239-244

MCCORMICK G (1976) Computability of global solutions to factorable nonconvex programs-part I- convex underestimating problems. Mathematical Programming **10** 146-175

MILOSAVLJEVIC N and HEIKKILA P (2001) A comprehensive approach to cooling tower design. Applied Thermal Engineering **21** 899-915

NAPHON P (2005) Study on the heat transfer characteristics of evaporative cooling towers. International Communication in Heat and Mass Transfer **32** 1066-1074

NIE X and ZHU XX (1999) Heat exchanger retrofit considering pressure drop and heat transfer enhancement. AICHE J **45** 1999 1239

PANJESHAHI MH and ATAEI A (2008) Application of an environmentally optimum cooling water system design in water and energy conservation. International Journal of Environmental Science and Technology **5(2)** 251-262

PANJESHAHI MH, ATAEI A, GHARAIE M and PARAND R (2009) Optimum design of cooling water systems for energy and water conservation. Chemical Engineering Research and Design **87** 200-209

PANJESHAHI MH, GHARAIE M and ATAEI A (2010) Debottlenecking procedure of effluent thermal treatment system energy. doi:101016/jenergy201007046

PAPAEFTHIMIOU VD, ROGDAKIS ED, KORONAKI IP and ZANNIS TC (2012) Thermodynamic study of the effects of ambient air conditions on the thermal performance characteristics of a closed wet cooling tower. Applied Thermal Engineering **33(34)** 199-207

PICÓN-NÚÑEZ M, MORALES-FUENTES A and VÁZQUEZ-RAMÍREZ EE (2007) Effect of network arrangement on the heat transfer area of cooling networks Applied Thermal Engineering **27** 2650-2656

POLLEY GT, PANJEHSHAHI MH and JEGEDE FO (1990) Pressure drop considerations in the retrofit of heat exchanger networks. Transactions of the Institute of Chemical Engineers **68 (Part A)** 211-220

PONCE-ORTEGA JM, SERNA-GONZÁLEZ M and JIMÉNEZ-GUTIÉRREZ A (2007) MINLP synthesis of optimal cooling networks. Chemical Engineering Science **62** 5728-5735

PONCE-ORTEGA JM, SERNA-GONZÁLEZ M and JIMÉNEZ-GUTIÉRREZ A (2010a) Optimization model for re-circulating cooling water systems. Computer and Chemical Engineering **34** 177-195

PONCE-ORTEGA JM, SERNA-GONZÁLEZ M and JIMÉNEZ-GUTIÉRREZ A (2010b) Synthesis of heat exchanger network with optimal placement of multiple utilities. Industrial Engineering and Chemistry Research **49** 2849-2856

PRICE T (2010) *Steam system optimization using process integration: Focus on boiler efficiency and pressure drop*. Masters Report, University of Pretoria, Pretoria, South Africa

QI X, LIU Z and LI D (2007) Performance characteristics of a shower cooling tower. Energy Conversion and Management **48** 193-203

QUESADA I and GROSSMANN I (1995) Global optimisation of bilinear process networks with multi component flows. Computers and Chemical Engineering **19(21)** 1219-1242

QURESHI BA and ZUBAIR SM (2006) A complete model of wet cooling towers with fouling in fills. Applied Thermal Engineering **26** 1982-1989

REZAEI E, SHAFIEI S and ABDOLLANHNEZHAD A (2010) Reducing water consumption of an industrial plant cooling unit using hybrid cooling tower. Energy Conversion and Management **51** 311-319

RUBIO-CASTRO E, SERNA-GONZÁLEZ M and PONCE-ORTEGA JM (2010) Optimal design of effluent-cooling systems using a mathematical programming model. Applied Thermal Engineering **30** 2116-2126

RUBIO-CASTRO E, SERNA-GONZÁLEZ M and PONCE-ORTEGA JM (2012) Synthesis of cooling water systems with multiple cooling towers. Applied Thermal Engineering **50** 957-974

SAVELSKI M and BAGAJEWICZ M (2000) On the optimality conditions of water utilization systems in process plants with single contaminants. Chemical Engineering Science **55(21)** 5035-5048

SAVULESCU L, KIM JK and SMITH R (2005) Studies on simultaneous energy and water minimization-Part I: Systems with no water re-use. Chemical Engineering Science **60** 3279-3290

SHENOY AU and SHENOY UV (2013) Targeting and design of CWNs (cooling water networks). Energy **55** 1033-1043

SHENOY U V, SINHA A and BONDYOPADHYAY S (1998) Multiple utilities targeting for heat exchanger networks. Chemical Engineering Research and Design **76 (Part A)** 259-272

SHERALI H and ALAMEDDINE A (1992) A new reformulation linearisation technique for bilinear programming problems. Journal of Global Optimisation **2** 379-410

TARIGHALESLAMI AH, HESAS RH and OMODKHAH MR (2010) An approach for water cost of cooling water systems in oil refinery. Chemical Engineering Transactions **21** 103-108

TELES J, CASTRO P and NOVAIS A (2008) LP-based strategies for the optimal design of industrial water networks with multiple contaminants. Chemical Engineering Science **63** 376-394

WANG Y and SMITH R (1994a) Wastewater minimization. Chemical Engineering Science **49(7)** 981-1006

WANG Y and SMITH R (1994b) Design of distributed effluent treatment systems. Chemical Engineering Science **49(18)** 3127-3145

WICAKSONO DS and KARIMI IA (2008) Piecewise MILP under- and over estimators for global optimization of bilinear programs. AICHE Journal **54(4)** 991-1008

YEE TF and GROSSMANN IE (1990) Simultaneous optimization model for heat integration-II Heat exchanger network synthesis. Computers and Chemical Engineering **14(10)** 1165-1184

YEE TF, GROSSMANN IE and KRAVANJA A (1990a) Simultaneous optimization model for heat integration-I Area and energy targeting and modeling of multi-stream exchangers. Computers and Chemical Engineering **14(10)** 1151-1164

YEE TF, GROSSMANN IE and KRAVANJA A (1990b) Simultaneous optimization model for heat integration-III Process and heat exchanger network optimization. Computers and Chemical Engineering **14(10)** 1185-1200

YINGJIAN L, XINKUI Y, QI Q and JIEZHI L (2011) The study on the evaporation cooling efficiency and effectiveness of cooling tower of film type. Energy Conversion and management **52** 53-59

APPENDIX A: LINEARIZATION

Piecewise linearization

The nonlinear heat exchangers and pipes pressure drop correlations of Nie and Zhu (1999) are shown in constraint (A 1) and (A 2) respectively.

$$\Delta P(i) = N_{t1} F_{in}(i)^{1.8} + N_{t2} F_{in}(i)^{2} \qquad \text{(A 1)}$$

$$\Delta P = N_p \frac{1}{F_p^{0.36}} \qquad \text{(A 2)}$$

where N_{t1}, N_{t2} and N_p are parameters

Piecewise linearization involves dividing the operating range of the heat exchangers into boundaries which are then linearized separately to find the linear expression for each boundary. The operating range of the heat exchangers where divided into two boundaries and Table A1 shows the linear expressions for the two regions of each heat exchanger pressure drop correlation.

***Table A1**: Linearized equations for the heat exchangers pressure drop correlation*

Operation	**Boundary (2.5 kg/s - 6 kg/s)**	**R-squared values**	**Boundary (6.0001 kg/s - 11 kg/s)**	**R-squared values**
OP01	1.49 f_1 - 2.71	0.98	2.78 f_2 - 10.8	0.99
OP02	2.52 f_1 - 4.60	0.98	4.72 f_2 - 18.4	0.99
OP03	1.01 f_1 - 1.84	0.98	1.88 f_2 - 7.32	0.99
OP04	3.65 f_1 - 6.66	0.98	6.83 f_2 - 26.7	0.99
OP05	2.85 f_1 - 5.20	0.98	5.33 f_2 - 20.8	0.99
OP06	1.79 f_1 - 3.26	0.98	3.34 f_2 - 13.0	0.99

Similarly, the line pressure drop correlation was divided into three boundaries and the linearized expressions are shown in Table A2.

***Table A2**: Linearized equations for the piping pressure drop correlation*

Boundary	**Linear equation**	**R-squared values**
0.2 kg/s - 1 kg/s	-10896 $f_{p,1}$ + 22312	0.88
1.0001 kg/s - 4 kg/s	-1517 $f_{p,2}$ + 13009	0.93
4.0001 kg/s - 16 kg/s	-223 $f_{p,3}$ + 7762	0.94

APPENDIX B: MATLAB AND GAMS CODE

Cooling tower model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
error = 2;
gg=0;
maxstepp = 100;
while error > 1
    gg = gg + 1;
if gg == 1
gams_output = 'std';
[CT,CM,s] = gams('start');

else
   if (model_status == 8)||(model_status == 1)||(model_status == 2)
       gams_output = 'std';
       [CT,CM,s] = gams('optimum1','E','B','M','Tctout','OSa');
       optimum1 = 1;
   else
gams_output = 'std';
[CT,CM,s] = gams('optimum','E','B','M','Tctout','watertaget');
optimum1 = 0;
   end

end

 NN = 3;

%model status
model_status = s(1);
solve_status = s(2);

mamax(1)=9.6;
mamax(2)=16;
mamax(3)=20;
mwin(1) = CM(1,1);
mwin(2) = CM(2,1);
mwin(3) = CM(3,1);
target(gg) = mwin(1)+mwin(2)+mwin(3);
twctin(1) = CT(1,1);
twctin(2) = CT(2,1);
twctin(3) = CT(3,1);
eff = zeros(3,1);
E = zeros(3,1);
B= zeros(3,1);
M = zeros(3,1);
Tctout=zeros(3,1);
OSa= zeros(3,1);
watertaget=zeros(3,1);
```

```
for m = 1:NN
err = 2;
j = 1;
maxstep = 10000;
while err > 1
mw = mwin(m);% water flowrate
mamax(1)=9.6;
mamax(2)=16;
mamax(3)=20;
ma = mamax(m);%air flowrate
twin = twctin(m)+273;
if mw <= 0.2*ma && mw > 0
        twout = 295;
        mwin(m) = 0.21*ma;
        Evap_loss = 0.05*mwin(m);

    break
end
if mw <= 0
        twout = 295;
        mwin(m) = 0;
        Evap_loss = 0;

    break
end
twexp = twin;
ta = 290.15;
twb = 288.95;
z =0;
a = z;
b = 2.438;
Af = 5.943;
V = 14.49;
N = 1000;
h = (b-a)/N; % cooling tower height
p = 101325;%total pressure
ifwo = 2.5016e+006;%vapor formation at 0C)
tw = twin-j/100;

x = zeros(1,N);
y = zeros(1,N);
y1 = zeros(1,N);
y2 = zeros(1,N);
y3 = zeros(1,N);
y4 = zeros(1,N);
y5 = zeros(1,N);
y6 = zeros(1,N);

cpw = 0;
cpa = 0;
cpv = 0;
```

```
hfv = 0;
pvwb = 0;
w= 0;
cpma= 0;
pvs= 0;
ws= 0;
Hv = 0;
Hma = 0;
Hmas= 0;
Ka= 0;
Le = 0;
k11=0;
k12=0;
k13=0;
k14=0;
k21=0;
k22=0;
k23=0;
k24=0;
k31=0;
k32=0;
k33=0;
k34=0;
k41=0;
k42=0;
k43=0;
k44=0;

cpa = (1.045356e3)-(3.161783e-1)*(ta/2)+(7.083814e-4)*...
        ((ta/2)^2)-(2.705209e-7)*((ta/2)^3);

cpv = (1.3605e3)+(2.31334)*(ta/2)-(2.46784e-10)*((ta/2)^5)...
        -(5.91332e-13)*((ta/2)^6);

hfv =  (3.4831814e6)-(5.8627703e3)*ta+(12.139568)*(ta^2)...
    -(1.40290431e-2)*(ta^3);

pvwb = 10^(10.79586*(1-273.16/twb)+5.02808*log10(273.16/twb)+...
      (1.50474*10^-4)*(1-10^(-8.29692*(twb/273.16-1)))+...
      (4.2873*10^-4)*(10^(4.76955*(1-273.16/twb))-1)+2.7861);
w = ((2501.6-2.3263*(twb-273.16))/(2501.6 + 1.8577*(ta-273.16)-...
        4.184*(twb-273.16)))*(0.62509*pvwb/(p-1.005*pvwb))-...
        ((1.00416*(ta-twb))/(2501.6 + 1.8577*...
        (ta-273.16)-4.184*(twb-273.16)));
cpma = cpa + w*cpv;
pvs = 10^(10.79586*(1-273.16/tw)+5.02808*log10(273.16/tw)+...
      (1.50474*10^-4)*(1-10^(-8.29692*(tw/273.16-1)))+...
      (4.2873*10^-4)*(10^(4.76955*(1-273.16/tw))-1)+2.7861);
ws = 0.622*pvs/(p-pvs);
Hv =ifwo+ ((1.3605e3)+(2.31334)*(ta/2)-(2.46784e-10)*...
    ((ta/2)^5)-(5.91332e-13)*((ta/2)^6))*ta;
```

```
Hma = cpa*ta+w*(ifwo+cpv*ta);
Hmas = ((1.045356e3)-(3.161783e-1)*(ta/2)+(7.083814e-4)*...
    ((ta/2)^2)-(2.705209e-7)*((ta/2)^3))*ta+ws*Hv;
 twout =tw;
 win = w;
 Hmain = Hma;
 Hmasin =Hmas;
 wsin=ws;

ss(1) = (mw/Af)*2.49*((mw/ma)^-0.67)*(0.609^-0.062);
ss(2) = (mw/Af)*1.664*((mw/ma)^-0.62)*(0.914^-0.27);
ss(3) = (mw/Af)*2.49*((mw/ma)^-0.67)*(0.914^-0.062);
Ka = ss(m);
if tw < twb
        twout = 295;
       Evap_loss = 0;
       mwin(m) = 0;
    break
end
for i = 1:N

    ifwo = 2.5016e+006;%vapor formation at 0C)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
    cpw = (8.15599e3)-(2.80627*10)*tw/2+(5.11283e-2)*((tw/2)^2)...
        -(2.17582e-13)*((tw/2)^6);

    pvs = 10^(10.79586*(1-273.16/tw)+5.02808*log10(273.16/tw)+...
      (1.50474*10^-4)*(1-10^(-8.29692*(tw/273.16-1)))+...
      (4.2873*10^-4)*(10^(4.76955*(1-273.16/tw))-1)+2.7861);

    ws = 0.622*pvs/(p-pvs);
    if ws < 0
       break
    end

    cpa = (1.045356e3)-(3.161783e-1)*(ta/2)+(7.083814e-4)*...
        ((ta/2)^2)-(2.705209e-7)*((ta/2)^3);

    cpv = (1.3605e3)+(2.31334)*(ta/2)-(2.46784e-10)*((ta/2)^5)...
        -(5.91332e-13)*((ta/2)^6);

    cpma = cpa + w*cpv;

    Hv =ifwo+ ((1.3605e3)+(2.31334)*(tw/2)-(2.46784e-10)*...
    ((tw/2)^5)-(5.91332e-13)*((tw/2)^6))*tw;

    Hmas = ((1.045356e3)-(3.161783e-1)*(tw/2)+(7.083814e-4)*...
    ((tw/2)^2)-(2.705209e-7)*((tw/2)^3))*tw+ws*Hv;

    Le = (0.866^0.667)*((ws+0.622)/(w+0.622)-1)/log((ws+0.622)/(w+0.622));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
   k11 = h*((Ka*Af/(cpw*mw))*(-(ws-w)*cpw*tw+(Hmas-Hma)+(Le-1)*(Hmas-Hma-(ws-
w)*Hv)));
   k12 = h * (Ka*Af*(ws-w));
   k13 = h * (Ka*Af*(ws-w)/ma);
   k14 = h * ((Ka*Af/ma)*(Le*(Hmas-Hma)+(1-Le)*Hv*(ws-w)));
%calculating k2 needs w + k1/2,
   tw = tw + k11/2;
   mw = mw + k12/2;
   w  = w + k13/2;
   Hma = Hma + k14/2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
   cpw = (8.15599e3)-(2.80627*10)*tw/2+(5.11283e-2)*((tw/2)^2)...
      -(2.17582e-13)*((tw/2)^6);

   pvs = 10^(10.79586*(1-273.16/tw)+5.02808*log10(273.16/tw)+...
     (1.50474*10^-4)*(1-10^(-8.29692*(tw/273.16-1)))+...
     (4.2873*10^-4)*(10^(4.76955*(1-273.16/tw))-1)+2.7861);

   ws = 0.622*pvs/(p-pvs);

   cpa = (1.045356e3)-(3.161783e-1)*(ta/2)+(7.083814e-4)*...
      ((ta/2)^2)-(2.705209e-7)*((ta/2)^3);

   cpv = (1.3605e3)+(2.31334)*(ta/2)-(2.46784e-10)*((ta/2)^5)...
      -(5.91332e-13)*((ta/2)^6);

   cpma = cpa + w*cpv;

   Hv =ifwo+ ((1.3605e3)+(2.31334)*(tw/2)-(2.46784e-10)*...
   ((tw/2)^5)-(5.91332e-13)*((tw/2)^6))*tw;
   Hmas = ((1.045356e3)-(3.161783e-1)*(tw/2)+(7.083814e-4)*...
   ((tw/2)^2)-(2.705209e-7)*((tw/2)^3))*tw+ws*Hv;

   Le = (0.866^0.667)*((ws+0.622)/(w+0.622)-1)/log((ws+0.622)/(w+0.622));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
   k21 = h*((Ka*Af/(cpw*mw))*(-(ws-w)*cpw*tw+(Hmas-Hma)+(Le-1)*(Hmas-Hma-(ws-
w)*Hv)));
   k22 = h * (Ka*Af*(ws-w));
   k23 = h * (Ka*Af*(ws-w)/ma);
   k24 = h * ((Ka*Af/ma)*(Le*(Hmas-Hma)+(1-Le)*Hv*(ws-w)));

   tw = tw - k11/2;
   mw = mw - k12/2;
   w  = w  - k13/2;
   Hma = Hma - k14/2;
%calculating k3 needs w + k2/2,
   tw = tw + k21/2;
   mw = mw + k22/2;
   w  = w  + k23/2;
   Hma = Hma + k24/2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
   cpw = (8.15599e3)-(2.80627*10)*tw/2+(5.11283e-2)*((tw/2)^2)...
      -(2.17582e-13)*((tw/2)^6);

   pvs = 10^(10.79586*(1-273.16/tw)+5.02808*log10(273.16/tw)+...
     (1.50474*10^-4)*(1-10^(-8.29692*(tw/273.16-1)))+...
     (4.2873*10^-4)*(10^(4.76955*(1-273.16/tw))-1)+2.7861);

   ws = 0.622*pvs/(p-pvs);

   cpa = (1.045356e3)-(3.161783e-1)*(ta/2)+(7.083814e-4)*...
      ((ta/2)^2)-(2.705209e-7)*((ta/2)^3);

   cpv = (1.3605e3)+(2.31334)*(ta/2)-(2.46784e-10)*((ta/2)^5)...
      -(5.91332e-13)*((ta/2)^6);

   cpma = cpa + w*cpv;

   Hv =ifwo+ ((1.3605e3)+(2.31334)*(tw/2)-(2.46784e-10)*...
   ((tw/2)^5)-(5.91332e-13)*((tw/2)^6))*tw;
   Hmas = ((1.045356e3)-(3.161783e-1)*(tw/2)+(7.083814e-4)*...
   ((tw/2)^2)-(2.705209e-7)*((tw/2)^3))*tw+ws*Hv;

   Le = (0.866^0.667)*((ws+0.622)/(w+0.622)-1)/log((ws+0.622)/(w+0.622));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%    k31 = h*((Ka*Af/(cpw*mw))*(-(ws-w)*cpw*tw+(Hmas-
Hma)+(Le-1)*(Hmas-Hma-(ws-w)*Hv)));
   k32 = h * (Ka*Af*(ws-w));
   k33 = h * (Ka*Af*(ws-w)/ma);
   k34 = h * ((Ka*Af/ma)*(Le*(Hmas-Hma)+(1-Le)*Hv*(ws-w)));

   tw = tw - k21/2;
   mw = mw - k22/2;
   w  = w  - k23/2;
   Hma = Hma - k24/2;
%calculating k4 needs k2 + w,
   tw = tw + k31;
   mw = mw + k32;
   w  = w  + k33;
   Hma = Hma + k34;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
   cpw = (8.15599e3)-(2.80627*10)*tw/2+(5.11283e-2)*((tw/2)^2)...
      -(2.17582e-13)*((tw/2)^6);
   pvs = 10^(10.79586*(1-273.16/tw)+5.02808*log10(273.16/tw)+...
     (1.50474*10^-4)*(1-10^(-8.29692*(tw/273.16-1)))+...
     (4.2873*10^-4)*(10^(4.76955*(1-273.16/tw))-1)+2.7861);

   ws = 0.622*pvs/(p-pvs);

   cpa = (1.045356e3)-(3.161783e-1)*(ta/2)+(7.083814e-4)*...
      ((ta/2)^2)-(2.705209e-7)*((ta/2)^3);

   cpv = (1.3605e3)+(2.31334)*(ta/2)-(2.46784e-10)*((ta/2)^5)...
      -(5.91332e-13)*((ta/2)^6);

   cpma = cpa + w*cpv;

   Hv =ifwo+ ((1.3605e3)+(2.31334)*(tw/2)-(2.46784e-10)*...
   ((tw/2)^5)-(5.91332e-13)*((tw/2)^6))*tw;
   Hmas = ((1.045356e3)-(3.161783e-1)*(tw/2)+(7.083814e-4)*...
   ((tw/2)^2)-(2.705209e-7)*((tw/2)^3))*tw+ws*Hv;

   Le = (0.866^0.667)*((ws+0.622)/(w+0.622)-1)/log((ws+0.622)/(w+0.622));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%    k41 = h*((Ka*Af/(cpw*mw))*(-(ws-w)*cpw*tw+(Hmas-
Hma)+(Le-1)*(Hmas-Hma-(ws-w)*Hv)));
   k42 = h * (Ka*Af*(ws-w));
   k43 = h * (Ka*Af*(ws-w)/ma);
   k44 = h * ((Ka*Af/ma)*(Le*(Hmas-Hma)+(1-Le)*Hv*(ws-w)));

   tw = tw - k31;
   mw = mw - k32;
   w  = w  - k33;
   Hma = Hma - k34;

   tw = tw + (k11 + 2*k21 + 2*k31 + k41)/6;
   mw = mw + (k12 + 2*k22 + 2*k32 + k42)/6;
   w = w   + (k13 + 2*k23 + 2*k33 + k43)/6;
   Hma = Hma + (k14 + 2*k24 + 2*k34+ k44)/6;

   ta = tw - ((Hmas-Hma)-(ws-w)*Hv)/cpma;

   z = a + i*h;

   %storage for ploting data
   x(i) = z; %distance
   y(i)=tw;  %water temperature
   y1(i)=ta; %air temperature
   %storage for effectiveness calculations
   y2(i) = Hmas;
   y3(i) = Hma;
   y4(i) = ws;
    y5(i) = w;
    y6(i) = mw;

end

if imag(tw)== 0 && i == N
err = abs(tw -twexp);
Evap_loss = ma*(w -win);
end

  j = j + 1;
  if j >= maxstep
      err = 0.01;
      fprintf(1,'maximum limit reached for cooling tower %d \n',m);
      display(twin)
  end
end
CC=5;
%eff(m) = (twin-twout)/(twin-twb)*100;

if mw > 0
Cemin = (mw*cpw/((Hmas-Hmasin)/(twin-twout)));
eff(m) = mw*cpw*(twin-twout)/(Cemin*(Hmas-Hmain));
E(m) = real(Evap_loss);
B(m) = E(m)/(CC - 1);
M(m) = E(m)*CC/(CC-1);
Tctout(m) = twout - 273;
OSa(m) = mwin(m);
watertaget(m) = target(1);
%averall effectiveness
Hact(m)= mw*cpw*(twin-twout);
Hmax(m) = (Cemin*(Hmas-Hmain));
else
eff(m) = 0;
E(m) = 0;
B(m) = E(m)/(CC - 1);
M(m) = E(m)*CC/(CC-1);
Tctout(m) = twout - 273;
OSa(m) = mwin(m);
watertaget(m) = target(1);
%averall effectiveness
Hact(m)= 0;
Hmax(m) = 0;
end
end

if (gg>=2)
    error1 = abs(XX1-Tctout(1));
    error2 = abs(XX2-Tctout(2));
    error3 = abs(XX3-Tctout(3));
    Overall_error = max([error1 error2 error3])
```

```
end
if (gg >= 2) && (model_status == 8||model_status == 1||model_status==2)&& (optimum1 == 1)
   if (error1 < 1) && (error2 < 1) && (error3 < 1)
      error = 0.01;
   else
      error = 10;
   end

else
   error =10;
end

XX1 = Tctout(1);
XX2 = Tctout(2);
XX3 = Tctout(3);



  if gg >= maxstepp
     error = 0.01;
     fprintf(1,'maximum iteration limit reached \n');

  end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%

gg
averall_eff = sum(Hact)/sum(Hmax);

display('Gams model status')
display(model_status)

display('inlet temperatures')
display(twctin)

display('outlet temperatures')
display(Tctout)

display('outlet flowrates')
display(OSa)

display('Evaporation')
display(E)

display('Blowdown')
display(B)

display('Make up')
display(M)
display('effectiveness')
display(eff)

display('averall effectiveness')
display(averall_eff)
end
%%%%%%%%%%%%%%%%%%%%%%%%%END%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Heat exchanger network model

Targeting

Start

```
Sets
i cooling water using operation /hx1*hx6/
n cooling towers                /ct1*ct3/
alias(i,j);

Parameters

Q(i) heat loads /hx1 600
          hx2 300
          hx3 800
          hx4 600
          hx5 300
          hx6 700/

Toutmax(i) limiting outlet temperature /hx1 45
                        hx2 60
                        hx3 50
                        hx4 60
                        hx5 55
                        hx6 45/
Tinmax(i)  limiting inlet temperature /hx1 30
                       hx2 40
                       hx3 25
                       hx4 45
                       hx5 40
                       hx6 30/

Finmax(i) maximum flowrate through cooling water using operation /hx1 9.52
                                         hx2 3.57
                                         hx3 7.62
                                         hx4 7.14
                                         hx5 4.76
                                         hx6 11.1/
OSmax(n) maximum cooling tower capacity /ct1 5.74
                         ct2 9.4
                         ct3 16.8/

Tctout(n) cooling tower supply temperature  /ct1 20
                           ct2 22
```

```
                              ct3 25/
Tretmax(n) maximum return temperature   /ct1 50
                           ct2 50
                           ct3 55/

Tsmax(n) maximum circuit supply temperature /ct1 50
                            ct2 50
                            ct3 55/
Tmin minimum temperature is equal to wet bulb temperature /15/

Cp                                   /4.18/
Tamb                                 /25/
CC                                    /5/
E(n) evaporation           /ct1 0
                          ct2 0
                         ct3 0/
B(n)                              /ct1 0
                                   ct2 0
                                  ct3 0 /
M(n)                            /ct1 0
                                 ct2 0
                                 ct3 0/;
Variables
CW overall cooling water supply
OS(n) operating cooling tower flowrate
OSin(n)
CS(n,i) fresh cooling water supply from n cooling tower to i operation
CR(i,n) return cooling water from i operation to n cooling tower
CTin cooling tower inlet flowrate
CTout cooling tower outlet flowrate
FR(j,i) cooling water reuse from j to i
Fin(i) inlet cooling water using operation flowrate
Fout(i) outlet cooling water using operation flowrate
Tin(i) inlet cooling water using operation temperature
Tout(i) outlet cooling water using operation temperature
Tret(n) return temperature to the cooling tower n
Ts(n) temperature after adding makeup
Mt total make up
Bt total blow down
Et total evaporation
crt(i,n) linearization variable CR*Tout
frt(j,i) linearization variable FR*Tout
fnt(i)   linearization variable F*Tout
tos(n) linearization variable Ts*OS
tcs(n,i) linearization variable Ts*CS;

Positive variable
OS
OSin
CS
CR
```

```
FR
Fin
Fout
Tin
Tout
Tret
Ts
Bt
Mt
Et
crt
frt
fnt
tos
tcs;

Binary variables
Lin(n,i)  connection node from cooling source n to operation i
L(j,i)    connection node from operation j to operation i
Lout(i,n) connection node from operation i to cooling source n
sumLin(i)
sumLout(i);

Tin.lo(i) = 15;
Tout.lo(i) = 15;
Tret.lo(n) = 15;
Ts.lo(n) = 15;

Equations

overall_cooling_water
cooling_towerMB1(n)
cooling_towerMB2(n)
circuit_supply_temperature(n)
operation_inlet_flowrate(i)
operation_recycle(i)
total_blowdown
total_makeup
total_evaporation
operationMB(i)
cooling_tower_design(n)
operation_design(i)
return_temperature_coolingtower(n)
return_temp_limit_operation(i)
operationEB(i)
operation_inlet_tempmax(i)

linearization1(i,n)
linearization2(i,n)
linearization3(i,n)
linearization4(i,n)
```

```
linearization5(j,i)
linearization6(j,i)
linearization7(j,i)
linearization8(j,i)
linearization9(i)
linearization10(i)
linearization11(i)
linearization12(i)
linearization13(n)
linearization14(n)
linearization15(n)
linearization16(n)
linearization17(n,i)
linearization18(n,i)
linearization19(n,i)
linearization20(n,i)

performance_index(n)
overall_PI
nopremix(i)
nopostmix(i)
samesource1(i,n)
samesource2(i,n)
same1(i)
same2(i)
source_linemax(n,i)
source_linemin(n,i)
reuse_linemax(j,i)
reuse_linemin(j,i)
sink_linemax(i,n)
sink_linemin(i,n)


loverall_cooling_water
lcooling_towerMB1(n)
lcooling_towerMB2(n)
lcircuit_supply_temperature(n)
loperation_inlet_flowrate(i)
loperation_recycle(i)
ltotal_makeup
ltotal_evaporation
loperationMB(i)
lcooling_tower_design(n)
loperation_design(i)
lreturn_temp_coolingtower(n)
lreturn_temp_limit_operation(i)
loperationEB(i)
loperation_inlet_temp(i)
loperation_inlet_tempmax(i)
loperationEB1(i)
loperationEB(i)
```

```
loperation_inlet_temp(i)
lCWR_temp(n)
lCWR_flow(n)
lperformance_index(n)
loverall_PI;


overall_cooling_water.. CW =e= sum(n,OS(n));
cooling_towerMB1(n).. OS(n)+M(n) =e= sum(i,CS(n,i));
cooling_towerMB2(n).. OS(n) =e= sum(i,CR(i,n))-B(n)-E(n);
circuit_supply_temperature(n).. sum(i,tcs(n,i)) =e=  Tamb*M(n) +  Tctout(n)*OS(n);
operation_inlet_flowrate(i).. Fin(i) =e= sum(n,CS(n,i))+ sum(j,FR(j,i)$(ord(i) ne ord(j)));
operation_recycle(i).. Fout(i) =e= sum(j,FR(i,j)$(ord(i) ne ord(j))) + sum(n,CR(i,n));
total_blowdown.. Bt =e= sum(n,B(n));
total_makeup..Mt =e= sum(n,M(n));
total_evaporation.. Et =e= sum(n,E(n));
operationMB(i)..Fin(i) =e= Fout(i);
cooling_tower_design(n).. OS(n) =l= OSmax(n);
operation_design(i).. Fin(i)=l= Finmax(i);
return_temperature_coolingtower(n).. Tretmax(n)*sum(i,CR(i,n))=g= sum(i,crt(i,n));
return_temp_limit_operation(i).. Tout(i) =l= Toutmax(i);
operationEB(i).. cp*fnt(i)  =e= Q(i) + cp*sum(j,frt(j,i)$(ord(i) ne ord(j)))+ cp*sum(n,tcs(n,i));
operation_inlet_tempmax(i).. Tin(i) =l= Tinmax(i);
linearization1(i,n).. crt(i,n) =g= OSmax(n)*Tout(i) + CR(i,n)*Toutmax(i)-OSmax(n)*Toutmax(i);
linearization2(i,n).. crt(i,n) =g= CR(i,n)*Tmin;
linearization3(i,n).. crt(i,n) =l= OSmax(n)*Tout(i)+CR(i,n)*Tmin-OSmax(n)*Tmin;
linearization4(i,n).. crt(i,n) =l= CR(i,n)*Toutmax(i);
linearization5(j,i)$(ord(i) ne ord(j)).. frt(j,i) =g= Finmax(i)*Tout(j) + FR(j,i)*Toutmax(i)-
Finmax(i)*Toutmax(j);
linearization6(j,i)$(ord(i) ne ord(j)).. frt(j,i) =g= FR(j,i)*Tmin;
linearization7(j,i)$(ord(i) ne ord(j)).. frt(j,i) =l= Finmax(i)*Tout(j)+FR(j,i)*Tmin-Finmax(i)*Tmin;
linearization8(j,i)$(ord(i) ne ord(j)).. frt(j,i) =l= FR(j,i)*Toutmax(j);
linearization9(i).. fnt(i) =g= Finmax(i)*Tout(i) + Fin(i)*Toutmax(i)-Finmax(i)*Toutmax(i);
linearization10(i).. fnt(i) =g= Fin(i)*Tmin;
linearization11(i).. fnt(i) =l= Finmax(i)*Tout(i)+Fin(i)*Tmin-Finmax(i)*Tmin;
linearization12(i).. fnt(i) =l= Fin(i)*Toutmax(i);
linearization13(n).. tos(n) =g= OSmax(n)*Ts(n) + OS(n)*Tsmax(n)-OSmax(n)*Tsmax(n);
linearization14(n).. tos(n) =g= OS(n)*Tmin;
linearization15(n).. tos(n) =l= OSmax(n)*Ts(n)+OS(n)*Tmin+OSmax(n)*Tmin;
linearization16(n).. tos(n) =l= OS(n)*Tsmax(n);
linearization17(n,i).. tcs(n,i) =g= OSmax(n)*Ts(n) + CS(n,i)*Tsmax(n)-OSmax(n)*Tsmax(n);
linearization18(n,i).. tcs(n,i) =g= CS(n,i)*Tmin;
linearization19(n,i).. tcs(n,i) =l= OSmax(n)*Ts(n)+CS(n,i)*Tmin-OSmax(n)*Tmin;
linearization20(n,i).. tcs(n,i) =l= CS(n,i)*Tsmax(n);

source_linemax(n,i)..CS(n,i) =l= Finmax(i)*Lin(n,i);
source_linemin(n,i)..CS(n,i) =g= 0*Lin(n,i);
reuse_linemax(j,i)$(ord(i) ne ord(j))..FR(j,i) =l= Finmax(i)*L(j,i);
reuse_linemin(j,i)$(ord(i) ne ord(j))..FR(j,i) =g= 0*L(j,i);
sink_linemax(i,n)..CR(i,n) =l= Finmax(i)*Lout(i,n);
sink_linemin(i,n)..CR(i,n) =g= 0*Lout(i,n);
```

```
nopremix(i).. sum(n,Lin(n,i)) =l= 1;
nopostmix(i).. sum(n,Lout(i,n)) =l= 1;
same1(i).. sumLin(i) =e= sum(n,Lin(n,i));
same2(i).. sumLout(i) =e= sum(n,Lout(i,n));
samesource1(i,n).. Lout(i,n) =l= Lin(n,i) + (2 - sumLin(i)- sumLout(i));
samesource2(i,n).. Lout(i,n) =g= Lin(n,i) - (2 - sumLin(i)- sumLout(i));


Model linear
/overall_cooling_water,cooling_towerMB1,cooling_towerMB2,circuit_supply_temperature,operati
on_inlet_flowrate,operation_recycle,total_blowdown,total_makeup,total_evaporation,operationMB,
cooling_tower_design,operation_design,return_temperature_coolingtower,return_temp_limit_opera
tion,operationEB,operation_inlet_tempmax,linearization1,linearization2,linearization3,linearization
4,linearization5,linearization6,linearization7,linearization8,linearization9,linearization10,linearizati
on11,linearization12,linearization17,linearization18,linearization19,linearization20,
source_linemax
source_linemin
reuse_linemax
reuse_linemin
sink_linemax
sink_linemin
nopremix
nopostmix
samesource1
samesource2
same1
same2 /;


loverall_cooling_water.. CW =e= sum(n,OS(n));
lcooling_towerMB1(n).. OS(n) =e= sum(i,CS(n,i))-M(n);
lcooling_towerMB2(n).. OS(n) =e= sum(i,CR(i,n))-B(n)-E(n);
lcircuit_supply_temperature(n).. Ts(n)*(OS(n)+M(n)) =e=  Tamb*M(n) +  Tctout(n)*OS(n);
loperation_inlet_flowrate(i).. Fin(i) =e= sum(n,CS(n,i))+ sum(j,FR(j,i)$(ord(i) ne ord(j)));
loperation_recycle(i).. Fout(i) =e= sum(j,FR(i,j)$(ord(i) ne ord(j))) + sum(n,CR(i,n));
ltotal_makeup..Mt =e= sum(n,M(n));
ltotal_evaporation.. Et =e=sum(n,E(n));
loperationMB(i)..Fin(i) =e= Fout(i);
lcooling_tower_design(n).. OS(n) =l= OSmax(n);
loperation_design(i).. Fin(i)=l= Finmax(i);
lreturn_temp_coolingtower(n).. Tretmax(n)*(sum(i,CR(i,n)))=g= sum(i,CR(i,n)*Tout(i));
lreturn_temp_limit_operation(i).. Tout(i) =l= Toutmax(i);
Fin(i)*cp*Tout(i);
loperationEB(i).. Q(i)  =e= Fin(i)*Tout(i)*cp - cp*Fin(i)*Tin(i);
loperation_inlet_temp(i).. Fin(i)*Tin(i) =e=  sum(j,Tout(j)*FR(j,i)$(ord(i) ne ord(j)))+
sum(n,Ts(n)*CS(n,i));
loperation_inlet_tempmax(i).. Tin(i) =l= Tinmax(i);
lCWR_temp(n).. Tret(n)*(sum(i,CR(i,n))) =e= sum(i,CR(i,n)*Tout(i));
lCWR_flow(n).. OSin(n) =e= OS(n);
```

```
Model exact/
loverall_cooling_water,lcooling_towerMB1,lcooling_towerMB2,lcircuit_supply_temperature,loper
ation_inlet_flowrate,loperation_recycle,ltotal_makeup,ltotal_evaporation,loperationMB,lcooling_to
wer_design,loperation_design,lreturn_temp_coolingtower,lreturn_temp_limit_operation,lCWR_tem
p,lCWR_flow
loperationEB
loperation_inlet_temp
loperation_inlet_tempmax
source_linemax
source_linemin
reuse_linemax
reuse_linemin
sink_linemax
sink_linemin
nopremix
nopostmix
samesource1
samesource2
same1
same2/;

option iterlim = 1000000;
option reslim = 1000000;
Solve linear using mip min CW;
option nlp = minos5;
Solve exact using minlp min CW;
set stat /modelstat,solvestat,d/;
parameter returnStat(stat);
returnStat('modelstat') = exact.modelstat;
returnStat('solvestat') = exact.solvestat;
returnStat('d') = 0;

$libinclude matout Tret.l n
$libinclude matout OSin.l n
$libinclude matout returnStat stat
***************END***************
```

Optimization with relaxed target

Optimum

```
Sets
i cooling water using operation /hx1*hx6/
n cooling towers                /1*3/
alias(i,j);

Parameters

Q(i) heat loads /hx1 600
                 hx2 300
                 hx3 800
                 hx4 600
                 hx5 300
                 hx6 700/

Toutmax(i) limiting outlet temperature /hx1 45
                                        hx2 60
                                        hx3 50
                                        hx4 60
                                        hx5 55
                                        hx6 45/
Tinmax(i)  limiting inlet temperature /hx1 30
                                       hx2 40
                                       hx3 25
                                       hx4 45
                                       hx5 40
                                       hx6 30/

Finmax(i) maximum flowrate through cooling water using operation /hx1 9.52
                                                                  hx2 3.57
                                                                  hx3 7.62
                                                                  hx4 7.14
                                                                  hx5 4.76
                                                                  hx6 11.1/
OSmax(n) maximum cooling tower capacity /1 5.74
                                         2 9.4
                                         3 16.8/
Tretmax(n) maximum return temperature   /1 50
                                        2 50
                                        3 55/
Tsmax(n) maximum circuit supply temperature /1 50
                                         2 50
                                         3 55/
Tmin minimum temperature is equal to wet bulb temperature /15/
cp /4.18/
Tamb /25/
CC /5/
largeV /10000/
Psp source pressure/1000/
```

```
Plimgin(n)/1 0
2 0
3 0/
Plimgout(n)/1 0
2 0
3 0/
Pimgin/1000/;
parameter E(n);
parameter B(n);
parameter M(n);
parameter Tctout(n);
parameter watertaget(n);
$if exist matdata.gms $include matdata.gms

Variables
CW overall cooling water supply
OS(n) operating cooling tower flowrate
CS(n,i) fresh cooling water supply from n cooling tower to i operation
CR(i,n) return cooling water from i operation to n cooling tower
CTin cooling tower inlet flowrate
CTout cooling tower outlet flowrate

FR(j,i) cooling water reuse from j to i
Fin(i) inlet cooling water using operation flowrate
Fout(i) outlet cooling water using operation flowrate
Tin(i) inlet cooling water using operation temperature
Tout(i) outlet cooling water using operation temperature
Tret(n) return temperature to the cooling tower n
Ts(n) temperature after adding makeup
Mt total make up
Bt total blow down
Et total evaporation
crt(i,n) linearization variable CR*Tout
frt(j,i) linearization variable FR*Tout
fnt(i)   linearization variable F*Tout
fntin(i) linearization variable F*Tin
tos(n) linearization variable Ts*OS
tcs(n,i) linearization variable Ts*CS
OSin(n)

Plin(n,i) line pressure drop from source to heat exchanger mixer
Plru(j,i)  line pressure drop from heat exchanger i to heat exchanger j
Plout(i,n) line pressure drop from heat exchanger splitter to sink
PHE(i)   heat exchanger pressure drop
PHEin(i) heat exchanger mixer pressure
PHEout(i) heat exchanger splitter
Pcwin(n) Source pressure
Pcwout(n) Sink pressure
Lpipin(n,i) inlet pipe length
Lpipru(j,i) pipe length for reuse
Lpipout(i,n) oulet pipe length
```

```
deltaP(n) pressure drop for each cooling tower pump
Pmin total pressure drop
Pimgout imaginary sink pressure
sumLin(i)
sumLout(i)
slak(n);

Positive variable
OS
CS
CR
FR
Fin
Fout
Tin
Tout
Tret
Ts
Bt
Mt
Et
crt
frt
fnt
tos
tcs
OSin
Ashel
pt

Plin
Plru
Plout
PHE
PHEin
PHEout
Pcwin
Pcwout
Lpipin
Lpipru
Lpipin
us
deltaP
slak;

negative variables
Ns3;
Tin.lo(i) = 15;
Tout.lo(i) = 15;
Tret.lo(n) = 15;
Ts.lo(n) = 15;
```

```
CW.lo = 10;
Lpipin.lo(n,i) = 2;
Lpipru.lo(j,i) = 2;
Lpipout.lo(i,n) = 2;
PHEin.up(i) = Psp;
PHEout.up(i) = Psp;
Pcwout.up(n) = Psp;

Binary variables
Lin(n,i)  connection node from cooling source n to operation i
L(j,i)    connection node from operation j to operation i
Lout(i,n) connection node from operation i to cooling source n
sumLin(i)
sumLout(i);

positive variables
CS1(n,i) piecewise linearization fresh cooling water supply from n cooling tower to i operation
CS2(n,i)
CS3(n,i)
CR1(i,n) piecewise linearization return cooling water from i operation to n cooling tower
CR2(i,n)
CR3(i,n)
FR1(j,i) piecewise linearization cooling water reuse from j to i
FR2(j,i)
FR3(j,i)
Fin1(i) piecewise linearization inlet cooling water using operation flowrate
Fin2(i);
binary variables
blin1(n,i)   piecewise linearization  connection node from cooling source n to operation i
blin2(n,i)
blin3(n,i)
bl1(j,i)    piecewise linearization  connection node from operation j to operation i
bl2(j,i)
bl3(j,i)
blout1(i,n) piecewise linearization   connection node from operation i to cooling source n
blout2(i,n)
blout3(i,n)
bfin1(i)    piecewise linearization   choosing the appropriate linear equation to use
bfin2(i);

Equations
target
ltarget
overall_cooling_water
cooling_towerMB1(n)
cooling_towerMB2(n)
circuit_supply_temperature(n)
operation_inlet_flowrate(i)
operation_recycle(i)
total_blowdown
total_makeup
```

```
total_evaporation
operationMB(i)
cooling_tower_design(n)
operation_design(i)
return_temp_coolingtower(n)
return_temp_limit_operation(i)
operationEB(i)
operation_inlet_temp(i)
operation_inlet_tempmax(i)
nopremix(i)
nopostmix(i)
samesource1(i,n)
samesource2(i,n)
same1(i)
same2(i)
source_linemax(n,i)
source_linemin(n,i)
reuse_linemax(j,i)
reuse_linemin(j,i)
sink_linemax(i,n)
sink_linemin(i,n)
logiconstPin(n,i)
logiconstPru(j,i)
logiconstPout(i,n)
pressure_source(n,i)
pressure_reuse(j,i)
pressure_sink(i,n)
maxlimitCS1(n,i)
minlimitCS1(n,i)
maxlimitCS2(n,i)
minlimitCS2(n,i)
maxlimitCS3(n,i)
minlimitCS3(n,i)
maxlimitCR1(i,n)
minlimitCR1(i,n)
maxlimitCR2(i,n)
minlimitCR2(i,n)
maxlimitCR3(i,n)
minlimitCR3(i,n)
maxlimitFR1(j,i)
minlimitFR1(j,i)
maxlimitFR2(j,i)
minlimitFR2(j,i)
maxlimitFR3(j,i)
minlimitFR3(j,i)
maxlimitFin1(i)
minlimitFin1(i)
maxlimitFin2(i)
minlimitFin2(i)
sumbnaryFin(i)
sumbnaryCS(n,i)
```

```
sumbnaryCR(i,n)
sumbnaryFR(j,i)
sumactulCS(n,i)
sumactulCR(i,n)
sumactulFR(j,i)
sumactulFin(i)
pressure_HE1
pressure_HE2
pressure_HE3
pressure_HE4
pressure_HE5
pressure_HE6
pressure_imgin(n)
pressure_imgout(n)
pressure_maxsource(n,i)
pressure_maxreuse(j,i)
pressure_maxsink(i,n)
pressuredrop_heatexchanger(i)
pressuredrop_per_coolingtower(n)
overall_pressuredrop
linearization1(i,n)
linearization2(i,n)
linearization3(i,n)
linearization4(i,n)
linearization5(j,i)
linearization6(j,i)
linearization7(j,i)
linearization8(j,i)
linearization9(i)
linearization10(i)
linearization11(i)
linearization12(i)
linearization13(n)
linearization14(n)
linearization15(n)
linearization16(n)
linearization17(n,i)
linearization18(n,i)
linearization19(n,i)
linearization20(n,i)
linearization21(i)
linearization22(i)
linearization23(i)
linearization24(i)
lsource_linemax(n,i)
lsource_linemin(n,i)
lreuse_linemax(j,i)
lreuse_linemin(j,i)
lsink_linemax(i,n)
lsink_linemin(i,n)
llogiconstPin(n,i)
```

```
llogiconstPru(j,i)
llogiconstPout(i,n)
lpressure_source(n,i)
lpressure_reuse(j,i)
lpressure_sink(i,n)
lpressure_HE1
lpressure_HE2
lpressure_HE3
lpressure_HE4
lpressure_HE5
lpressure_HE6
lnopremix(i)
lnopostmix(i)
lsamesource1(i,n)
lsamesource2(i,n)
lsame1(i)
lsame2(i)
lpressure_imgin(n)
lpressure_imgout(n)
lpressure_maxsource(n,i)
lpressure_maxreuse(j,i)
lpressure_maxsink(i,n)
lpressuredrop_heatexchanger(i)
lpressuredrop_per_coolingtower(n)
loverall_pressuredrop
loverall_cooling_water
lcooling_towerMB1(n)
lcooling_towerMB2(n)
lcircuit_supply_temperature(n)
loperation_inlet_flowrate(i)
loperation_recycle(i)
ltotal_blowdown
ltotal_makeup
ltotal_evaporation
loperationMB(i)
lcooling_tower_design(n)
loperation_design(i)
lreturn_temp_coolingtower(n)
lreturn_temp_limit_operation(i)
loperationEB(i)
loperation_inlet_temp(i)
loperation_inlet_tempmax(i)
loperationEB1(i)
lCWR_temp(n)
lCWR_flow(n)
totalpresdrop(n)
maxstreams(i);

maxstreams(i)..sum(n,Lin(n,i)) + sum(j,L(j,i)) =l= 4;
target.. CW =l= watertaget('1')+sum(n,slak(n));
overall_cooling_water.. CW =e= sum(n,OS(n));
```

```
cooling_towerMB1(n).. OS(n)+M(n) =e= sum(i,CS(n,i));
cooling_towerMB2(n).. OS(n) =e= sum(i,CR(i,n))-B(n)-E(n);
circuit_supply_temperature(n).. sum(i,tcs(n,i)) =e=  Tamb*M(n)+ Tctout(n)*OS(n);
operation_inlet_flowrate(i).. Fin(i) =e= sum(n,CS(n,i))+ sum(j,FR(j,i)$(ord(i) ne ord(j)));
operation_recycle(i).. Fout(i) =e= sum(j,FR(i,j)$(ord(i) ne ord(j))) + sum(n,CR(i,n));
total_blowdown.. Bt =e= sum(n,B(n));
total_makeup..Mt =e= sum(n,M(n));
total_evaporation.. Et =e= sum(n,E(n));
operationMB(i)..Fin(i) =e= Fout(i);
cooling_tower_design(n).. OS(n) =l= OSmax(n);
operation_design(i).. Fin(i)=l= Finmax(i);
return_temp_coolingtower(n).. Tretmax(n)*sum(i,CR(i,n))=g= sum(i,crt(i,n));
return_temp_limit_operation(i).. Tout(i) =l= Toutmax(i);
operation_inlet_tempmax(i).. Tin(i) =l= Tinmax(i);
operationEB(i).. cp*fnt(i)  =e= Q(i) + cp*sum(j,frt(j,i)$(ord(i) ne ord(j)))+ cp*sum(n,tcs(n,i));
linearization1(i,n).. crt(i,n) =g= OSmax(n)*Tout(i) + CR(i,n)*Toutmax(i)-OSmax(n)*Toutmax(i);
linearization2(i,n).. crt(i,n) =g= CR(i,n)*Tmin;
linearization3(i,n).. crt(i,n) =l= OSmax(n)*Tout(i)+CR(i,n)*Tmin-OSmax(n)*Tmin;
linearization4(i,n).. crt(i,n) =l= CR(i,n)*Toutmax(i);
linearization5(j,i)$(ord(i) ne ord(j)).. frt(j,i) =g= Finmax(i)*Tout(j) + FR(j,i)*Toutmax(i)-
Finmax(i)*Toutmax(j);
linearization6(j,i)$(ord(i) ne ord(j)).. frt(j,i) =g= FR(j,i)*Tmin;
linearization7(j,i)$(ord(i) ne ord(j)).. frt(j,i) =l= Finmax(i)*Tout(j)+FR(j,i)*Tmin-Finmax(i)*Tmin;
linearization8(j,i)$(ord(i) ne ord(j)).. frt(j,i) =l= FR(j,i)*Toutmax(j);
linearization9(i).. fnt(i) =g= Finmax(i)*Tout(i) + Fin(i)*Toutmax(i)-Finmax(i)*Toutmax(i);
linearization10(i).. fnt(i) =g= Fin(i)*Tmin;
linearization11(i).. fnt(i) =l= Finmax(i)*Tout(i)+Fin(i)*Tmin-Finmax(i)*Tmin;
linearization12(i).. fnt(i) =l= Fin(i)*Toutmax(i);
linearization13(n).. tos(n) =g= OSmax(n)*Ts(n) + OS(n)*Tsmax(n)-OSmax(n)*Tsmax(n);
linearization14(n).. tos(n) =g= OS(n)*Tmin;
linearization15(n).. tos(n) =l= OSmax(n)*Ts(n)+OS(n)*Tmin+OSmax(n)*Tmin;
linearization16(n).. tos(n) =l= OS(n)*Tsmax(n);
linearization17(n,i).. tcs(n,i) =g= OSmax(n)*Ts(n) + CS(n,i)*Tsmax(n)-OSmax(n)*Tsmax(n);
linearization18(n,i).. tcs(n,i) =g= CS(n,i)*Tmin;
linearization19(n,i).. tcs(n,i) =l= OSmax(n)*Ts(n)+CS(n,i)*Tmin-OSmax(n)*Tmin;
linearization20(n,i).. tcs(n,i) =l= CS(n,i)*Tsmax(n);
linearization21(i).. fntin(i) =g= Finmax(i)*Tin(i) + Fin(i)*Tinmax(i)-Finmax(i)*Tinmax(i);
linearization22(i).. fntin(i) =g= Fin(i)*Tmin;
linearization23(i).. fntin(i) =l= Finmax(i)*Tin(i)+Fin(i)*Tmin-Finmax(i)*Tmin;
linearization24(i).. fntin(i) =l= Fin(i)*Tinmax(i);
source_linemax(n,i)..CS(n,i) =l= Finmax(i)*Lin(n,i);
source_linemin(n,i)..CS(n,i) =g= 0.278*Lin(n,i);
reuse_linemax(j,i)$(ord(i) ne ord(j))..FR(j,i) =l= Finmax(i)*L(j,i);
reuse_linemin(j,i)$(ord(i) ne ord(j))..FR(j,i) =g= 0.278*L(j,i);
sink_linemax(i,n)..CR(i,n) =l= Finmax(i)*Lout(i,n);
sink_linemin(i,n)..CR(i,n) =g= 0.278*Lout(i,n);

logiconstPin(n,i)..Plin(n,i) =l= largeV*Lin(n,i);
logiconstPru(j,i)..Plru(j,i)$(ord(i) ne ord(j)) =l= largeV*L(j,i);
logiconstPout(i,n)..Plout(i,n) =l= largeV*Lout(i,n);
```

```
pressure_source(n,i)..Plin(n,i)*1000=e= -10896*CS1(n,i)+22312*blin1(n,i)-
1517*CS2(n,i)+13009*blin2(n,i)-223*CS3(n,i)+7762*blin3(n,i);
pressure_reuse(j,i)..Plru(j,i)*1000=e= -10896*FR1(j,i)+22312*bl1(j,i)-
1517*FR2(j,i)+13009*bl2(j,i)-223*FR3(j,i)+7762*bl3(j,i);
pressure_sink(i,n)..Plout(i,n)*1000=e= -10896*CR1(i,n)+2232*blout1(i,n)-
1517*CR2(i,n)+13009*blout2(i,n)-223*CR3(i,n)+7762*blout3(i,n);
maxlimitCS1(n,i)..CS1(n,i)=l=1*blin1(n,i);
minlimitCS1(n,i)..CS1(n,i)=g=0.278*blin1(n,i);
maxlimitCS2(n,i)..CS2(n,i)=l=4*blin2(n,i);
minlimitCS2(n,i)..CS2(n,i)=g=1.0001*blin2(n,i);
maxlimitCS3(n,i)..CS3(n,i)=l=16*blin3(n,i);
minlimitCS3(n,i)..CS3(n,i)=g=4.0001*blin3(n,i);
sumbnaryCS(n,i)..Lin(n,i) =e= blin1(n,i)+blin2(n,i)+blin3(n,i);
maxlimitCR1(i,n)..CR1(i,n)=l=1*blout1(i,n);
minlimitCR1(i,n)..CR1(i,n)=g=0.278*blout1(i,n);
maxlimitCR2(i,n)..CR2(i,n)=l=4*blout2(i,n);
minlimitCR2(i,n)..CR2(i,n)=g=1.0001*blout2(i,n);
maxlimitCR3(i,n)..CR3(i,n)=l=16*blout3(i,n);
minlimitCR3(i,n)..CR3(i,n)=g=4.0001*blout3(i,n);
sumbnaryCR(i,n).. Lout(i,n) =e= blout1(i,n)+blout2(i,n)+blout3(i,n);
maxlimitFR1(j,i)..FR1(j,i)=l=1*bl1(j,i);
minlimitFR1(j,i)..FR1(j,i)=g=0.278*bl1(j,i);
maxlimitFR2(j,i)..FR2(j,i)=l=4*bl2(j,i);
minlimitFR2(j,i)..FR2(j,i)=g=1.0001*bl2(j,i);
maxlimitFR3(j,i)..FR3(j,i)=l=16*bl3(j,i);
minlimitFR3(j,i)..FR3(j,i)=g=4.0001*bl3(j,i);
sumbnaryFR(j,i).. L(j,i) =e= bl1(j,i)+bl2(j,i)+bl3(j,i);
maxlimitFin1(i)..Fin1(i)=l=6*bfin1(i);
minlimitFin1(i)..Fin1(i)=g=2.5*bfin1(i);
maxlimitFin2(i)..Fin2(i)=l=11.5*bfin2(i);
minlimitFin2(i)..Fin2(i)=g=6.0001*bfin2(i);
sumbnaryFin(i).. 1 =e= bfin1(i)+bfin2(i);
sumactulCS(n,i)..CS(n,i) =e= CS1(n,i)+CS2(n,i)+CS3(n,i);
sumactulCR(i,n)..CR(i,n) =e= CR1(i,n)+CR2(i,n)+CR3(i,n);
sumactulFR(j,i)..FR(j,i) =e= FR1(j,i)+FR2(j,i)+FR3(j,i);
sumactulFin(i)..Fin(i) =e= Fin1(i)+Fin2(i);
pressure_HE1.. PHE('hx1')*1000=e= 1.4912*Fin1('hx1')-2.7126*bfin1('hx1') + 2.7812*Fin2('hx1')-
10.825*bfin2('hx1');
pressure_HE2.. PHE('hx2')*1000=e= 2.5256*Fin1('hx2')-4.6006*bfin1('hx2') + 4.7185*Fin2('hx2')-
18.393*bfin2('hx2');
pressure_HE3.. PHE('hx3')*1000=e= 1.0103*Fin1('hx3')-1.836*bfin1('hx3') + 1.8819*Fin2('hx3')-
7.317*bfin2('hx3');
pressure_HE4.. PHE('hx4')*1000=e= 3.6517*Fin1('hx4')-6.6586*bfin1('hx4') + 6.8309*Fin2('hx4')-
26.655*bfin2('hx4');
pressure_HE5.. PHE('hx5')*1000=e= 2.854*Fin1('hx5')-5.2008*bfin1('hx5') + 5.3345*Fin2('hx5')-
20.802*bfin2('hx5');
pressure_HE6.. PHE('hx6')*1000=e= 1.7914*Fin1('hx6')-3.2603*bfin1('hx6') + 3.3431*Fin2('hx6')-
13.019*bfin2('hx6');
nopremix(i).. sum(n,Lin(n,i)) =l= 1;
nopostmix(i).. sum(n,Lout(i,n)) =l= 1;
same1(i).. sumLin(i) =e= sum(n,Lin(n,i));
```

```
same2(i).. sumLout(i) =e= sum(n,Lout(i,n));
samesource1(i,n).. Lout(i,n) =l= Lin(n,i) + (2 - sumLin(i)- sumLout(i));
samesource2(i,n).. Lout(i,n) =g= Lin(n,i) - (2 - sumLin(i)- sumLout(i));
pressure_imgin(n)..Plimgin(n)=e= Pimgin - Pcwin(n);
pressure_imgout(n)..Plimgout(n) =l=Pcwout(n) - Pimgout;
pressure_maxsource(n,i)..Plin(n,i) =l= Pcwin(n)-PHEin(i)+largeV*(1-Lin(n,i));
pressure_maxreuse(j,i)$(ord(i) ne ord(j))..Plru(j,i) =l= PHEout(j)-PHEin(i)+largeV*(1-L(j,i));
pressure_maxsink(i,n)..Plout(i,n) =l= PHEout(i)-Pcwout(n)+largeV*(1-Lout(i,n));
pressuredrop_heatexchanger(i)..PHE(i) =e= PHEin(i)-PHEout(i);
pressuredrop_per_coolingtower(n)..deltaP(n) =e= Pcwin(n)-Pcwout(n);
overall_pressuredrop..Pmin =e= 0.52*watertaget('1')*(Pimgin-Pimgout)/1000+3.5*CW*3.6;
Model linear /
maxstreams
target
overall_cooling_water
cooling_towerMB1
cooling_towerMB2
circuit_supply_temperature
operation_inlet_flowrate
operation_recycle
total_blowdown
total_makeup
total_evaporation
operationMB
cooling_tower_design
operation_design
return_temp_coolingtower
return_temp_limit_operation,
operation_inlet_tempmax
operationEB
linearization1
linearization2
linearization3
linearization4
linearization5
linearization6
linearization7
linearization8
linearization9
linearization10
linearization11
linearization12
linearization17
linearization18
linearization19
linearization20
linearization21
linearization22
linearization23
linearization24
source_linemax
```

```
source_linemin
reuse_linemax
reuse_linemin
sink_linemax
sink_linemin

logiconstPin
logiconstPru
logiconstPout
pressure_source
pressure_reuse
pressure_sink
*piesewise linearization
maxlimitCS1
minlimitCS1
maxlimitCS2
minlimitCS2
maxlimitCS3
minlimitCS3
maxlimitCR1
minlimitCR1
maxlimitCR2
minlimitCR2
maxlimitCR3
minlimitCR3
maxlimitFR1
minlimitFR1
maxlimitFR2
minlimitFR2
maxlimitFR3
minlimitFR3
maxlimitFin1
minlimitFin1
maxlimitFin2
minlimitFin2
sumbnaryFin
sumbnaryCS
sumbnaryCR
sumbnaryFR
sumactulCS
sumactulCR
sumactulFR
sumactulFin
pressure_HE1
pressure_HE2
pressure_HE3
pressure_HE4
pressure_HE5
pressure_HE6
nopremix
nopostmix
```

```
samesource1
samesource2
same1
same2
pressure_imgin
pressure_imgout
pressure_maxsource
pressure_maxreuse
pressure_maxsink
pressuredrop_heatexchanger
pressuredrop_per_coolingtower
overall_pressuredrop
/;

ltarget.. CW =l= watertaget('1')+sum(n,slak(n));
loverall_cooling_water.. CW =e= sum(n,OS(n));
lcooling_towerMB1(n).. OS(n) =e= sum(i,CS(n,i))-M(n);
lcooling_towerMB2(n).. OS(n) =e= sum(i,CR(i,n))-B(n)-E(n);
lcircuit_supply_temperature(n).. Ts(n)*(M(n)+OS(n)) =e=  Tamb*M(n) +  Tctout(n)*OS(n);
loperation_inlet_flowrate(i).. Fin(i) =e= sum(n,CS(n,i))+ sum(j,FR(j,i)$(ord(i) ne ord(j)));
loperation_recycle(i).. Fout(i) =e= sum(j,FR(i,j)$(ord(i) ne ord(j))) + sum(n,CR(i,n));
ltotal_blowdown.. Bt =e= sum(n,B(n));
ltotal_makeup..Mt =e= sum(n,M(n));
ltotal_evaporation.. Et =e=sum(n,E(n));
loperationMB(i)..Fin(i) =e= Fout(i);
lcooling_tower_design(n)..OS(n) =l= OSmax(n);
loperation_design(i)..Fin(i)=l= Finmax(i);
lreturn_temp_coolingtower(n)..Tretmax(n)*(sum(i,CR(i,n)))=g= sum(i,CR(i,n)*Tout(i));
lreturn_temp_limit_operation(i)..Tout(i) =l= Toutmax(i);
Fin(i)*cp*Tout(i);
loperationEB(i).. Q(i)  =e= Fin(i)*Tout(i)*cp - cp*Fin(i)*Tin(i);
loperation_inlet_temp(i).. Fin(i)*Tin(i) =e=  sum(j,Tout(j)*FR(j,i)$(ord(i) ne ord(j)))+
sum(n,Ts(n)*CS(n,i));
loperation_inlet_tempmax(i).. Tin(i) =l= Tinmax(i);
lCWR_temp(n).. Tret(n)*(sum(i,CR(i,n))) =e= sum(i,CR(i,n)*Tout(i));
lCWR_flow(n).. OSin(n) =e= OS(n);
lnopremix(i).. sum(n,Lin(n,i)) =l= 1;
lnopostmix(i).. sum(n,Lout(i,n)) =l= 1;
lsame1(i).. sumLin(i) =e= sum(n,Lin(n,i));
lsame2(i).. sumLout(i) =e= sum(n,Lout(i,n));
lsamesource1(i,n).. Lout(i,n) =l= Lin(n,i) + (2 - sumLin(i)- sumLout(i));
lsamesource2(i,n).. Lout(i,n) =g= Lin(n,i) - (2 - sumLin(i)- sumLout(i));
lsource_linemax(n,i)..CS(n,i) =l= Finmax(i)*Lin(n,i);
lsource_linemin(n,i)..CS(n,i) =g= 0.278*Lin(n,i);
lreuse_linemax(j,i)$(ord(i) ne ord(j))..FR(j,i) =l= Finmax(i)*L(j,i);
lreuse_linemin(j,i)$(ord(i) ne ord(j))..FR(j,i) =g= 0.278*L(j,i);
lsink_linemax(i,n)..CR(i,n) =l= Finmax(i)*Lout(i,n);
lsink_linemin(i,n)..CR(i,n) =g= 0.278*Lout(i,n);
llogiconstPin(n,i)..Plin(n,i) =l= largeV*Lin(n,i);
llogiconstPru(j,i)$(ord(i) ne ord(j))..Plru(j,i) =l= largeV*L(j,i);
llogiconstPout(i,n)..Plout(i,n) =l= largeV*Lout(i,n);
*correlations*
```

```
lpressure_source(n,i)..Plin(n,i)*1000 =e= 1.31*CS(n,i)*CS(n,i)*CS(n,i)*CS(n,i)-
51.5*CS(n,i)*CS(n,i)*CS(n,i)
                              +725.6*CS(n,i)*CS(n,i)-4518.8*CS(n,i)+16593*Lin(n,i);
lpressure_reuse(j,i)$(ord(i) ne ord(j))..Plru(j,i)*1000 =e=  1.31*FR(j,i)*FR(j,i)*FR(j,i)*FR(j,i)-
51.5*FR(j,i)*FR(j,i)*FR(j,i)
                                          +725.6*FR(j,i)*FR(j,i)-4518.8*FR(j,i)+16593*L(j,i);
lpressure_sink(i,n)..Plout(i,n)*1000 =e=  1.31*CR(i,n)*CR(i,n)*CR(i,n)*CR(i,n)-
51.5*CR(i,n)*CR(i,n)*CR(i,n)
                              +725.6*CR(i,n)*CR(i,n)-4518.8*CR(i,n)+16593*Lout(i,n);
lpressure_HE1..PHE('hx1')*1000 =e= 0.1431*Fin('hx1')*Fin('hx1')+0.2372*Fin('hx1');
lpressure_HE2..PHE('hx2')*1000 =e= 0.2432*Fin('hx2')*Fin('hx2')+0.395*Fin('hx2');
lpressure_HE3..PHE('hx3')*1000 =e= 0.0967*Fin('hx3')*Fin('hx3')+0.1627*Fin('hx3');
lpressure_HE4..PHE('hx4')*1000 =e= 0.3526*Fin('hx4')*Fin('hx4')+0.5642*Fin('hx4');
lpressure_HE5..PHE('hx5')*1000 =e= 0.2751*Fin('hx5')*Fin('hx5')+0.4446*Fin('hx5');
lpressure_HE6..PHE('hx6')*1000 =e= 0.1809*Fin('hx6')*Fin('hx6')+0.2083*Fin('hx6');
lpressure_imgin(n)..Plimgin(n)=e= Pimgin - Pcwin(n);
lpressure_imgout(n)..Plimgout(n) =l=Pcwout(n) - Pimgout;
lpressure_maxsource(n,i)..Plin(n,i) =l= Pcwin(n)-PHEin(i)+largeV*(1-Lin(n,i));
lpressure_maxreuse(j,i)$(ord(i) ne ord(j))..Plru(j,i) =l= PHEout(j)-PHEin(i)+largeV*(1-L(j,i));
lpressure_maxsink(i,n)..Plout(i,n) =l= PHEout(i)-Pcwout(n)+largeV*(1-Lout(i,n));
lpressuredrop_heatexchanger(i)..PHE(i) =e= PHEin(i)-PHEout(i);
lpressuredrop_per_coolingtower(n)..deltaP(n) =e= Pcwin(n)-Pcwout(n);
loverall_pressuredrop..Pmin =e= 0.52*CW*(Pimgin-Pimgout)/1000+3.5*CW*3.6;
Model exact/
maxstreams
ltarget
loverall_cooling_water
lcooling_towerMB1
lcooling_towerMB2
lcircuit_supply_temperature
loperation_inlet_flowrate
loperation_recycle
ltotal_blowdown
ltotal_makeup
ltotal_evaporation
loperationMB
lcooling_tower_design
loperation_design
lreturn_temp_coolingtower
lreturn_temp_limit_operation
*loperationEB1
loperationEB
loperation_inlet_temp
loperation_inlet_tempmax
lCWR_temp
lCWR_flow
lnopremix
lnopostmix
lsamesource1
lsamesource2
lsame1
```

```
lsame2
lsource_linemax
lsource_linemin
lreuse_linemax
lreuse_linemin
lsink_linemax
lsink_linemin
llogiconstPin
llogiconstPru
llogiconstPout
lpressure_source
lpressure_reuse
lpressure_sink
lpressure_HE1
lpressure_HE2
lpressure_HE3
lpressure_HE4
lpressure_HE5
lpressure_HE6
lpressure_imgin
lpressure_imgout
lpressure_maxsource
lpressure_maxreuse
lpressure_maxsink
lpressuredrop_heatexchanger
lpressuredrop_per_coolingtower
loverall_pressuredrop
/;

option iterlim = 1000000;
option reslim = 1000000;
option optcr = 1;
option nlp = minos5;

$if exist matdata.gms $include matdata.gms

Solve linear using mip min Pmin;
$if exist matdata.gms $include matdata.gms
exact.optfile = 1;
Solve exact using MINLP min Pmin;

set stat /modelstat,solvestat,d/;
parameter returnStat(stat);
returnStat('modelstat') = exact.modelstat;
returnStat('solvestat') = exact.solvestat;
returnStat('d') = 0;

$libinclude matout Tret.l n
$libinclude matout OSin.l n
$libinclude matout returnStat stat
***************END***************
```

Optimization with fixed target

```
*Optimum1*
Sets
i cooling water using operation /hx1*hx6/
n cooling towers             /1*3/
alias(i,j);

Parameters
Q(i) heat loads /hx1 600
           hx2 300
           hx3 800
           hx4 600
           hx5 300
           hx6 700/

Toutmax(i) limiting outlet temperature /hx1 45
                          hx2 60
                          hx3 50
                          hx4 60
                          hx5 55
                          hx6 45/
Tinmax(i)  limiting inlet temperature /hx1 30
                         hx2 40
                         hx3 25
                         hx4 45
                         hx5 40
                         hx6 30/

Finmax(i) maximum flowrate through cooling water using operation /hx1 9.52
                                                 hx2 3.57
                                                 hx3 7.62
                                                 hx4 7.14
                                                 hx5 4.76
                                                 hx6 11.1/
OSmax(n) maximum cooling tower capacity /1 5.74
                           2 9.4
                           3 16.8/
Tretmax(n) maximum return temperature   /1 50
                          2 50
                          3 55/
Tsmax(n) maximum circuit supply temperature /1 50
                           2 50
                           3 55/
Tmin minimum temperature is equal to wet bulb temperature /15/

cp                                   /4.18/
Tamb                                 /25/
CC                                   /5/

largeV                               /10000/
```

```
Psp source pressure       /1000/
Plimgin(n)                /1 0
                         2 0
                         3 0/
Plimgout(n)                          /1 0
                         2 0
                         3 0/
Pimgin                               /1000/;

parameter E(n);
parameter B(n);
parameter M(n);
parameter Tctout(n);
parameter OSa(n);

$if exist matdata.gms $include matdata.gms

Variables
CW overall cooling water supply
OS(n) operating cooling tower flowrate
CS(n,i) fresh cooling water supply from n cooling tower to i operation
CR(i,n) return cooling water from i operation to n cooling tower
CTin cooling tower inlet flowrate
CTout cooling tower outlet flowrate
FR(j,i) cooling water reuse from j to i
Fin(i) inlet cooling water using operation flowrate
Fout(i) outlet cooling water using operation flowrate
Tin(i) inlet cooling water using operation temperature
Tout(i) outlet cooling water using operation temperature
Tret(n) return temperature to the cooling tower n
Ts(n) temperature after adding makeup

Mt total make up
Bt total blow down
Et total evaporation
crt(i,n) linearization variable CR*Tout
frt(j,i) linearization variable FR*Tout
fnt(i)   linearization variable F*Tout
fntin(i) linearization variable F*Tin
tos(n) linearization variable Ts*OS
tcs(n,i) linearization variable Ts*CS
OSin(n)
Plin(n,i) line pressure drop from source to heat exchanger mixer
Plru(j,i)  line pressure drop from heat exchanger i to heat exchanger j
Plout(i,n) line pressure drop from heat exchanger splitter to sink
PHE(i)   heat exchanger pressure drop
PHEin(i) heat exchanger mixer pressure
PHEout(i) heat exchanger splitter
Pcwin(n) Source pressure
Pcwout(n) Sink pressure
Lpipin(n,i) inlet pipe length
Lpipru(j,i) pipe length for reuse
```

```
Lpipout(i,n) oulet pipe length
deltaP(n) pressure drop for each cooling tower pump
Pmin total pressure drop
Pimgout imaginary sink pressure
sumLin(i)
sumLout(i)
slak(n);

Positive variable
OS
CS
CR
FR
Fin
Fout
Tin
Tout
Tret
Ts
Bt
Mt
Et
crt
frt
fnt
tos
tcs
OSin
Ashel
pt
Plin
Plru
Plout
PHE
PHEin
PHEout
Pcwin
Pcwout
Lpipin
Lpipru
Lpipin
deltaP
slak;

Tin.lo(i) = 15;
Tout.lo(i) = 15;
Tret.lo(n) = 15;
Ts.lo(n) = 15;
CW.lo = 10;
Lpipin.lo(n,i) = 2;
Lpipru.lo(j,i) = 2;
Lpipout.lo(i,n) = 2;
```

```
PHEin.up(i) = Psp;
PHEout.up(i) = Psp;
Pcwout.up(n) = Psp;

Binary variables
Lin(n,i)  connection node from cooling source n to operation i
L(j,i)    connection node from operation j to operation i
Lout(i,n) connection node from operation i to cooling source n
sumLin(i)
sumLout(i);

positive variables
CS1(n,i) piecewise linearization fresh cooling water supply from n cooling tower to i operation
CS2(n,i)
CS3(n,i)
CR1(i,n) piecewise linearization return cooling water from i operation to n cooling tower
CR2(i,n)
CR3(i,n)
FR1(j,i) piecewise linearization cooling water reuse from j to i
FR2(j,i)
FR3(j,i)
Fin1(i) piecewise linearization inlet cooling water using operation flowrate
Fin2(i);
binary variables
blin1(n,i)   piecewise linearization  connection node from cooling source n to operation i
blin2(n,i)
blin3(n,i)
bl1(j,i)    piecewise linearization  connection node from operation j to operation i
bl2(j,i)
bl3(j,i)
blout1(i,n) piecewise linearization   connection node from operation i to cooling source n
blout2(i,n)
blout3(i,n)
bfin1(i)    piecewise linearization   choosing the appropriate linear equation to use
bfin2(i);

Equations
target
ltarget
overall_cooling_water
cooling_towerMB1(n)
cooling_towerMB2(n)
circuit_supply_temperature(n)
operation_inlet_flowrate(i)
operation_recycle(i)
total_blowdown
total_makeup
total_evaporation
operationMB(i)
cooling_tower_design(n)
operation_design(i)
return_temp_coolingtower(n)
```

```
return_temp_limit_operation(i)
operationEB(i)
operation_inlet_temp(i)
operation_inlet_tempmax(i)
nopremix(i)
nopostmix(i)
samesource1(i,n)
samesource2(i,n)
same1(i)
same2(i)
source_linemax(n,i)
source_linemin(n,i)
reuse_linemax(j,i)
reuse_linemin(j,i)
sink_linemax(i,n)
sink_linemin(i,n)
logiconstPin(n,i)
logiconstPru(j,i)
logiconstPout(i,n)
pressure_source(n,i)
pressure_reuse(j,i)
pressure_sink(i,n)
*piesewise linearization
maxlimitCS1(n,i)
minlimitCS1(n,i)
maxlimitCS2(n,i)
minlimitCS2(n,i)
maxlimitCS3(n,i)
minlimitCS3(n,i)
maxlimitCR1(i,n)
minlimitCR1(i,n)
maxlimitCR2(i,n)
minlimitCR2(i,n)
maxlimitCR3(i,n)
minlimitCR3(i,n)
maxlimitFR1(j,i)
minlimitFR1(j,i)
maxlimitFR2(j,i)
minlimitFR2(j,i)
maxlimitFR3(j,i)
minlimitFR3(j,i)
maxlimitFin1(i)
minlimitFin1(i)
maxlimitFin2(i)
minlimitFin2(i)
sumbnaryFin(i)
sumbnaryCS(n,i)
sumbnaryCR(i,n)
sumbnaryFR(j,i)
sumactulCS(n,i)
sumactulCR(i,n)
```

```
sumactulFR(j,i)
sumactulFin(i)
pressure_HE1
pressure_HE2
pressure_HE3
pressure_HE4
pressure_HE5
pressure_HE6
pressure_imgin(n)
pressure_imgout(n)
pressure_maxsource(n,i)
pressure_maxreuse(j,i)
pressure_maxsink(i,n)
pressuredrop_heatexchanger(i)
pressuredrop_per_coolingtower(n)
overall_pressuredrop
linearization1(i,n)
linearization2(i,n)
linearization3(i,n)
linearization4(i,n)
linearization5(j,i)
linearization6(j,i)
linearization7(j,i)
linearization8(j,i)
linearization9(i)
linearization10(i)
linearization11(i)
linearization12(i)
linearization13(n)
linearization14(n)
linearization15(n)
linearization16(n)
linearization17(n,i)
linearization18(n,i)
linearization19(n,i)
linearization20(n,i)
linearization21(i)
linearization22(i)
linearization23(i)
linearization24(i)
lsource_linemax(n,i)
lsource_linemin(n,i)
lreuse_linemax(j,i)
lreuse_linemin(j,i)
lsink_linemax(i,n)
lsink_linemin(i,n)
llogiconstPin(n,i)
llogiconstPru(j,i)
llogiconstPout(i,n)
lpressure_source(n,i)
lpressure_reuse(j,i)
```

```
lpressure_sink(i,n)
lpressure_HE1
lpressure_HE2
lpressure_HE3
lpressure_HE4
lpressure_HE5
lpressure_HE6
lnopremix(i)
lnopostmix(i)
lsamesource1(i,n)
lsamesource2(i,n)
lsame1(i)
lsame2(i)
lpressure_imgin(n)
lpressure_imgout(n)
lpressure_maxsource(n,i)
lpressure_maxreuse(j,i)
lpressure_maxsink(i,n)
lpressuredrop_heatexchanger(i)
lpressuredrop_per_coolingtower(n)
loverall_pressuredrop
loverall_cooling_water
lcooling_towerMB1(n)
lcooling_towerMB2(n)
lcircuit_supply_temperature(n)
loperation_inlet_flowrate(i)
loperation_recycle(i)
ltotal_blowdown
ltotal_makeup
ltotal_evaporation
loperationMB(i)
lcooling_tower_design(n)
loperation_design(i)
lreturn_temp_coolingtower(n)
lreturn_temp_limit_operation(i)
loperationEB(i)
loperation_inlet_temp(i)
loperation_inlet_tempmax(i)
loperationEB1(i)
lCWR_temp(n)
lCWR_flow(n)
totalpresdrop(n)
osaeqt(n)
maxstreams(i);
maxstreams(i)..sum(n,Lin(n,i)) + sum(j,L(j,i)) =l= 4;
osaeqt(n)..OS(n)=e=OSa(n)+slak(n);
*target.. CW =l= watertaget('1');
overall_cooling_water.. CW =e= sum(n,OS(n));
cooling_towerMB1(n).. OS(n)+M(n) =e= sum(i,CS(n,i));
cooling_towerMB2(n).. OS(n) =e= sum(i,CR(i,n))-B(n)-E(n);
circuit_supply_temperature(n).. sum(i,tcs(n,i)) =e=  Tamb*M(n)+ Tctout(n)*OS(n);
```

```
operation_inlet_flowrate(i).. Fin(i) =e= sum(n,CS(n,i))+ sum(j,FR(j,i)$(ord(i) ne ord(j)));
operation_recycle(i).. Fout(i) =e= sum(j,FR(i,j)$(ord(i) ne ord(j))) + sum(n,CR(i,n));

total_blowdown.. Bt =e= sum(n,B(n));
total_makeup..Mt =e= sum(n,M(n));
total_evaporation.. Et =e= sum(n,E(n));
operationMB(i)..Fin(i) =e= Fout(i);
cooling_tower_design(n).. OS(n) =l= OSmax(n);
operation_design(i).. Fin(i)=l= Finmax(i);
return_temp_coolingtower(n).. Tretmax(n)*sum(i,CR(i,n))=g= sum(i,crt(i,n));
return_temp_limit_operation(i).. Tout(i) =l= Toutmax(i);
operation_inlet_tempmax(i).. Tin(i) =l= Tinmax(i);
operationEB(i).. cp*fnt(i)  =e= Q(i) + cp*sum(j,frt(j,i)$(ord(i) ne ord(j)))+ cp*sum(n,tcs(n,i));
linearization1(i,n).. crt(i,n) =g= OSmax(n)*Tout(i) + CR(i,n)*Toutmax(i)-OSmax(n)*Toutmax(i);
linearization2(i,n).. crt(i,n) =g= CR(i,n)*Tmin;
linearization3(i,n).. crt(i,n) =l= OSmax(n)*Tout(i)+CR(i,n)*Tmin-OSmax(n)*Tmin;
linearization4(i,n).. crt(i,n) =l= CR(i,n)*Toutmax(i);
linearization5(j,i)$(ord(i) ne ord(j)).. frt(j,i) =g= Finmax(i)*Tout(j) + FR(j,i)*Toutmax(i)-
Finmax(i)*Toutmax(j);
linearization6(j,i)$(ord(i) ne ord(j)).. frt(j,i) =g= FR(j,i)*Tmin;
linearization7(j,i)$(ord(i) ne ord(j)).. frt(j,i) =l= Finmax(i)*Tout(j)+FR(j,i)*Tmin-Finmax(i)*Tmin;
linearization8(j,i)$(ord(i) ne ord(j)).. frt(j,i) =l= FR(j,i)*Toutmax(j);
linearization9(i).. fnt(i) =g= Finmax(i)*Tout(i) + Fin(i)*Toutmax(i)-Finmax(i)*Toutmax(i);
linearization10(i).. fnt(i) =g= Fin(i)*Tmin;
linearization11(i).. fnt(i) =l= Finmax(i)*Tout(i)+Fin(i)*Tmin-Finmax(i)*Tmin;
linearization12(i).. fnt(i) =l= Fin(i)*Toutmax(i);
linearization13(n).. tos(n) =g= OSmax(n)*Ts(n) + OS(n)*Tsmax(n)-OSmax(n)*Tsmax(n);
linearization14(n).. tos(n) =g= OS(n)*Tmin;
linearization15(n).. tos(n) =l= OSmax(n)*Ts(n)+OS(n)*Tmin+OSmax(n)*Tmin;
linearization16(n).. tos(n) =l= OS(n)*Tsmax(n);
linearization17(n,i).. tcs(n,i) =g= OSmax(n)*Ts(n) + CS(n,i)*Tsmax(n)-OSmax(n)*Tsmax(n);
linearization18(n,i).. tcs(n,i) =g= CS(n,i)*Tmin;
linearization19(n,i).. tcs(n,i) =l= OSmax(n)*Ts(n)+CS(n,i)*Tmin-OSmax(n)*Tmin;
linearization20(n,i).. tcs(n,i) =l= CS(n,i)*Tsmax(n);
linearization21(i).. fntin(i) =g= Finmax(i)*Tin(i) + Fin(i)*Tinmax(i)-Finmax(i)*Tinmax(i);
linearization22(i).. fntin(i) =g= Fin(i)*Tmin;
linearization23(i).. fntin(i) =l= Finmax(i)*Tin(i)+Fin(i)*Tmin-Finmax(i)*Tmin;
linearization24(i).. fntin(i) =l= Fin(i)*Tinmax(i);
source_linemax(n,i)..CS(n,i) =l= Finmax(i)*Lin(n,i);
source_linemin(n,i)..CS(n,i) =g= 0.278*Lin(n,i);
reuse_linemax(j,i)$(ord(i) ne ord(j))..FR(j,i) =l= Finmax(i)*L(j,i);
reuse_linemin(j,i)$(ord(i) ne ord(j))..FR(j,i) =g= 0.278*L(j,i);
sink_linemax(i,n)..CR(i,n) =l= Finmax(i)*Lout(i,n);
sink_linemin(i,n)..CR(i,n) =g= 0.278*Lout(i,n);
logiconstPin(n,i)..Plin(n,i) =l= largeV*Lin(n,i);
logiconstPru(j,i)..Plru(j,i)$(ord(i) ne ord(j)) =l= largeV*L(j,i);
logiconstPout(i,n)..Plout(i,n) =l= largeV*Lout(i,n);
*piecewise linearization*
pressure_source(n,i)..Plin(n,i)*1000=e= -10896*CS1(n,i)+22312*blin1(n,i)-
1517*CS2(n,i)+13009*blin2(n,i)-223*CS3(n,i)+7762*blin3(n,i);
pressure_reuse(j,i)..Plru(j,i)*1000=e= -10896*FR1(j,i)+22312*bl1(j,i)-
1517*FR2(j,i)+13009*bl2(j,i)-223*FR3(j,i)+7762*bl3(j,i);
```

```
pressure_sink(i,n)..Plout(i,n)*1000=e= -10896*CR1(i,n)+2232*blout1(i,n)-
1517*CR2(i,n)+13009*blout2(i,n)-223*CR3(i,n)+7762*blout3(i,n);
maxlimitCS1(n,i)..CS1(n,i)=l=1*blin1(n,i);
minlimitCS1(n,i)..CS1(n,i)=g=0.278*blin1(n,i);
maxlimitCS2(n,i)..CS2(n,i)=l=4*blin2(n,i);
minlimitCS2(n,i)..CS2(n,i)=g=1.0001*blin2(n,i);
maxlimitCS3(n,i)..CS3(n,i)=l=16*blin3(n,i);
minlimitCS3(n,i)..CS3(n,i)=g=4.0001*blin3(n,i);
sumbnaryCS(n,i)..Lin(n,i) =e= blin1(n,i)+blin2(n,i)+blin3(n,i);
maxlimitCR1(i,n)..CR1(i,n)=l=1*blout1(i,n);
minlimitCR1(i,n)..CR1(i,n)=g=0.278*blout1(i,n);
maxlimitCR2(i,n)..CR2(i,n)=l=4*blout2(i,n);
minlimitCR2(i,n)..CR2(i,n)=g=1.0001*blout2(i,n);
maxlimitCR3(i,n)..CR3(i,n)=l=16*blout3(i,n);
minlimitCR3(i,n)..CR3(i,n)=g=4.0001*blout3(i,n);
sumbnaryCR(i,n).. Lout(i,n) =e= blout1(i,n)+blout2(i,n)+blout3(i,n);
maxlimitFR1(j,i)..FR1(j,i)=l=1*bl1(j,i);
minlimitFR1(j,i)..FR1(j,i)=g=0.278*bl1(j,i);
maxlimitFR2(j,i)..FR2(j,i)=l=4*bl2(j,i);
minlimitFR2(j,i)..FR2(j,i)=g=1.0001*bl2(j,i);
maxlimitFR3(j,i)..FR3(j,i)=l=16*bl3(j,i);
minlimitFR3(j,i)..FR3(j,i)=g=4.0001*bl3(j,i);
sumbnaryFR(j,i).. L(j,i) =e= bl1(j,i)+bl2(j,i)+bl3(j,i);
maxlimitFin1(i)..Fin1(i)=l=6*bfin1(i);
minlimitFin1(i)..Fin1(i)=g=2.5*bfin1(i);
maxlimitFin2(i)..Fin2(i)=l=11.5*bfin2(i);
minlimitFin2(i)..Fin2(i)=g=6.0001*bfin2(i);
sumbnaryFin(i).. 1 =e= bfin1(i)+bfin2(i);
sumactulCS(n,i)..CS(n,i) =e= CS1(n,i)+CS2(n,i)+CS3(n,i);
sumactulCR(i,n)..CR(i,n) =e= CR1(i,n)+CR2(i,n)+CR3(i,n);
sumactulFR(j,i)..FR(j,i) =e= FR1(j,i)+FR2(j,i)+FR3(j,i);
sumactulFin(i)..Fin(i) =e= Fin1(i)+Fin2(i);

pressure_HE1.. PHE('hx1')*1000=e= 1.4912*Fin1('hx1')-2.7126*bfin1('hx1') + 2.7812*Fin2('hx1')-
10.825*bfin2('hx1');
pressure_HE2.. PHE('hx2')*1000=e= 2.5256*Fin1('hx2')-4.6006*bfin1('hx2') + 4.7185*Fin2('hx2')-
18.393*bfin2('hx2');
pressure_HE3.. PHE('hx3')*1000=e= 1.0103*Fin1('hx3')-1.836*bfin1('hx3') + 1.8819*Fin2('hx3')-
7.317*bfin2('hx3');
pressure_HE4.. PHE('hx4')*1000=e= 3.6517*Fin1('hx4')-6.6586*bfin1('hx4') + 6.8309*Fin2('hx4')-
26.655*bfin2('hx4');
pressure_HE5.. PHE('hx5')*1000=e= 2.854*Fin1('hx5')-5.2008*bfin1('hx5') + 5.3345*Fin2('hx5')-
20.802*bfin2('hx5');
pressure_HE6.. PHE('hx6')*1000=e= 1.7914*Fin1('hx6')-3.2603*bfin1('hx6') + 3.3431*Fin2('hx6')-
13.019*bfin2('hx6');
nopremix(i).. sum(n,Lin(n,i)) =l= 1;
nopostmix(i).. sum(n,Lout(i,n)) =l= 1;
same1(i).. sumLin(i) =e= sum(n,Lin(n,i));
same2(i).. sumLout(i) =e= sum(n,Lout(i,n));
samesource1(i,n).. Lout(i,n) =l= Lin(n,i) + (2 - sumLin(i)- sumLout(i));
samesource2(i,n).. Lout(i,n) =g= Lin(n,i) - (2 - sumLin(i)- sumLout(i));
pressure_imgin(n)..Plimgin(n)=e= Pimgin - Pcwin(n);
```

```
pressure_imgout(n)..Plimgout(n) =l=Pcwout(n) - Pimgout;
pressure_maxsource(n,i)..Plin(n,i) =l= Pcwin(n)-PHEin(i)+largeV*(1-Lin(n,i));
pressure_maxreuse(j,i)$(ord(i) ne ord(j))..Plru(j,i) =l= PHEout(j)-PHEin(i)+largeV*(1-L(j,i));
pressure_maxsink(i,n)..Plout(i,n) =l= PHEout(i)-Pcwout(n)+largeV*(1-Lout(i,n));
pressuredrop_heatexchanger(i)..PHE(i) =e= PHEin(i)-PHEout(i);
pressuredrop_per_coolingtower(n)..deltaP(n) =e= Pcwin(n)-Pcwout(n);
overall_pressuredrop..Pmin =e= 0.52*sum(n,OSa(n))*(Pimgin-Pimgout)/1000+3.5*CW*3.6;

Model linear /
maxstreams
osaeqt
overall_cooling_water
cooling_towerMB1
cooling_towerMB2
circuit_supply_temperature
operation_inlet_flowrate
operation_recycle
total_blowdown
total_makeup
total_evaporation
operationMB
cooling_tower_design
operation_design
return_temp_coolingtower
return_temp_limit_operation,
operation_inlet_tempmax
operationEB
linearization1
linearization2
linearization3
linearization4
linearization5
linearization6
linearization7
linearization8
linearization9
linearization10
linearization11
linearization12
linearization17
linearization18
linearization19
linearization20
linearization21
linearization22
linearization23
linearization24
source_linemax
source_linemin
reuse_linemax
reuse_linemin
sink_linemax
```

```
sink_linemin
logiconstPin
logiconstPru
logiconstPout
pressure_source
pressure_reuse
pressure_sink
*piesewise linearization
maxlimitCS1
minlimitCS1
maxlimitCS2
minlimitCS2
maxlimitCS3
minlimitCS3
maxlimitCR1
minlimitCR1
maxlimitCR2
minlimitCR2
maxlimitCR3
minlimitCR3
maxlimitFR1
minlimitFR1
maxlimitFR2
minlimitFR2
maxlimitFR3
minlimitFR3
maxlimitFin1
minlimitFin1
maxlimitFin2
minlimitFin2
sumbnaryFin
sumbnaryCS
sumbnaryCR
sumbnaryFR
sumactulCS
sumactulCR
sumactulFR
sumactulFin
pressure_HE1
pressure_HE2
pressure_HE3
pressure_HE4
pressure_HE5
pressure_HE6
nopremix
nopostmix
samesource1
samesource2
same1
same2
*pressure_HE
```

```
pressure_imgin
pressure_imgout
pressure_maxsource
pressure_maxreuse
pressure_maxsink
pressuredrop_heatexchanger
pressuredrop_per_coolingtower
overall_pressuredrop
/;

loverall_cooling_water.. CW =e= sum(n,OS(n));
lcooling_towerMB1(n).. OS(n) =e= sum(i,CS(n,i))-M(n);
lcooling_towerMB2(n).. OS(n) =e= sum(i,CR(i,n))-B(n)-E(n);
lcircuit_supply_temperature(n).. Ts(n)*(M(n)+OS(n)) =e=  Tamb*M(n) +  Tctout(n)*OS(n);
loperation_inlet_flowrate(i).. Fin(i) =e= sum(n,CS(n,i))+ sum(j,FR(j,i)$(ord(i) ne ord(j)));
loperation_recycle(i).. Fout(i) =e= sum(j,FR(i,j)$(ord(i) ne ord(j))) + sum(n,CR(i,n));
ltotal_blowdown.. Bt =e= sum(n,B(n));
ltotal_makeup..Mt =e= sum(n,M(n));
ltotal_evaporation.. Et =e=sum(n,E(n));
loperationMB(i)..Fin(i) =e= Fout(i);
lcooling_tower_design(n)..OS(n) =l= OSmax(n);
loperation_design(i)..Fin(i)=l= Finmax(i);
lreturn_temp_coolingtower(n)..Tretmax(n)*(sum(i,CR(i,n)))=g= sum(i,CR(i,n)*Tout(i));
lreturn_temp_limit_operation(i)..Tout(i) =l= Toutmax(i);
Fin(i)*cp*Tout(i);
loperationEB(i).. Q(i)  =e= Fin(i)*Tout(i)*cp - cp*Fin(i)*Tin(i);
loperation_inlet_temp(i).. Fin(i)*Tin(i) =e=  sum(j,Tout(j)*FR(j,i)$(ord(i) ne ord(j)))+
sum(n,Ts(n)*CS(n,i));
loperation_inlet_tempmax(i).. Tin(i) =l= Tinmax(i);
lCWR_temp(n).. Tret(n)*(sum(i,CR(i,n))) =e= sum(i,CR(i,n)*Tout(i));
lCWR_flow(n).. OSin(n) =e= OS(n);
lnopremix(i).. sum(n,Lin(n,i)) =l= 1;
lnopostmix(i).. sum(n,Lout(i,n)) =l= 1;
lsame1(i).. sumLin(i) =e= sum(n,Lin(n,i));
lsame2(i).. sumLout(i) =e= sum(n,Lout(i,n));
lsamesource1(i,n).. Lout(i,n) =l= Lin(n,i) + (2 - sumLin(i)- sumLout(i));
lsamesource2(i,n).. Lout(i,n) =g= Lin(n,i) - (2 - sumLin(i)- sumLout(i));
lsource_linemax(n,i)..CS(n,i) =l= Finmax(i)*Lin(n,i);
lsource_linemin(n,i)..CS(n,i) =g= 0.278*Lin(n,i);
lreuse_linemax(j,i)$(ord(i) ne ord(j))..FR(j,i) =l= Finmax(i)*L(j,i);
lreuse_linemin(j,i)$(ord(i) ne ord(j))..FR(j,i) =g= 0.278*L(j,i);
lsink_linemax(i,n)..CR(i,n) =l= Finmax(i)*Lout(i,n);
lsink_linemin(i,n)..CR(i,n) =g= 0.278*Lout(i,n);
llogiconstPin(n,i)..Plin(n,i) =l= largeV*Lin(n,i);
llogiconstPru(j,i)$(ord(i) ne ord(j))..Plru(j,i) =l= largeV*L(j,i);
llogiconstPout(i,n)..Plout(i,n) =l= largeV*Lout(i,n);
*correlations*
lpressure_source(n,i)..Plin(n,i)*1000 =e= 1.31*CS(n,i)*CS(n,i)*CS(n,i)*CS(n,i)-
51.5*CS(n,i)*CS(n,i)*CS(n,i)
                                   +725.6*CS(n,i)*CS(n,i)-4518.8*CS(n,i)+16593*Lin(n,i);
lpressure_reuse(j,i)$(ord(i) ne ord(j))..Plru(j,i)*1000 =e=  1.31*FR(j,i)*FR(j,i)*FR(j,i)*FR(j,i)-
51.5*FR(j,i)*FR(j,i)*FR(j,i)
```

```
                                    +725.6*FR(j,i)*FR(j,i)-4518.8*FR(j,i)+16593*L(j,i);
lpressure_sink(i,n)..Plout(i,n)*1000 =e=  1.31*CR(i,n)*CR(i,n)*CR(i,n)*CR(i,n)-
51.5*CR(i,n)*CR(i,n)*CR(i,n)
                               +725.6*CR(i,n)*CR(i,n)-4518.8*CR(i,n)+16593*Lout(i,n);
lpressure_HE1..PHE('hx1')*1000 =e= 0.1431*Fin('hx1')*Fin('hx1')+0.2372*Fin('hx1');
lpressure_HE2..PHE('hx2')*1000 =e= 0.2432*Fin('hx2')*Fin('hx2')+0.395*Fin('hx2');
lpressure_HE3..PHE('hx3')*1000 =e= 0.0967*Fin('hx3')*Fin('hx3')+0.1627*Fin('hx3');
lpressure_HE4..PHE('hx4')*1000 =e= 0.3526*Fin('hx4')*Fin('hx4')+0.5642*Fin('hx4');
lpressure_HE5..PHE('hx5')*1000 =e= 0.2751*Fin('hx5')*Fin('hx5')+0.4446*Fin('hx5');
lpressure_HE6..PHE('hx6')*1000 =e= 0.1809*Fin('hx6')*Fin('hx6')+0.2083*Fin('hx6');
lpressure_imgin(n)..Plimgin(n)=e= Pimgin - Pcwin(n);
lpressure_imgout(n)..Plimgout(n) =l=Pcwout(n) - Pimgout;
lpressure_maxsource(n,i)..Plin(n,i) =l= Pcwin(n)-PHEin(i)+largeV*(1-Lin(n,i));
lpressure_maxreuse(j,i)$(ord(i) ne ord(j))..Plru(j,i) =l= PHEout(j)-PHEin(i)+largeV*(1-L(j,i));
lpressure_maxsink(i,n)..Plout(i,n) =l= PHEout(i)-Pcwout(n)+largeV*(1-Lout(i,n));
lpressuredrop_heatexchanger(i)..PHE(i) =e= PHEin(i)-PHEout(i);
lpressuredrop_per_coolingtower(n)..deltaP(n) =e= Pcwin(n)-Pcwout(n);
loverall_pressuredrop.. Pmin =e= 0.52*CW*(Pimgin-Pimgout)/1000+3.5*CW*3.6;
Model exact/
maxstreams
osaeqt
loverall_cooling_water
lcooling_towerMB1
lcooling_towerMB2
lcircuit_supply_temperature
loperation_inlet_flowrate
loperation_recycle
ltotal_blowdown
ltotal_makeup
ltotal_evaporation
loperationMB
lcooling_tower_design
loperation_design
lreturn_temp_coolingtower
lreturn_temp_limit_operation
loperationEB
loperation_inlet_temp
loperation_inlet_tempmax
lCWR_temp
lCWR_flow
lnopremix
lnopostmix
lsamesource1
lsamesource2
lsame1
lsame2
lsource_linemax
lsource_linemin
lreuse_linemax
lreuse_linemin
lsink_linemax
```

```
lsink_linemin
llogiconstPin
llogiconstPru
llogiconstPout
lpressure_source
lpressure_reuse
lpressure_sink
lpressure_HE1
lpressure_HE2
lpressure_HE3
lpressure_HE4
lpressure_HE5
lpressure_HE6
lpressure_imgin
lpressure_imgout
lpressure_maxsource
lpressure_maxreuse
lpressure_maxsink
lpressuredrop_heatexchanger
lpressuredrop_per_coolingtower
loverall_pressuredrop
/;

option iterlim = 1000000;
option reslim = 1000000;
option nlp = minos5;

$if exist matdata.gms $include matdata.gms

Solve linear using mip min Pmin;
$if exist matdata.gms $include matdata.gms
exact.optfile = 1;
Solve exact using MINLP min Pmin;
set stat /modelstat,solvestat,d/;
parameter returnStat(stat);
returnStat('modelstat') = exact.modelstat;
returnStat('solvestat') = exact.solvestat;
returnStat('d') = 0;

$libinclude matout Tret.l n
$libinclude matout OSin.l n
$libinclude matout returnStat stat
****************END***************
```