University of Alberta

Optimization of Steel Microstructure during Laminar Cooling

by

Baher Bineshmarvasti

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Process Control

Department of Chemical and Materials Engineering

Fall 2011
Edmonton, Alberta

Abstract

Optimization techniques, in conjunction with a finite element thermal model, are used in this thesis to optimize the temperature profile (i.e. cooling rate and coiling temperature) of a steel skelp during laminar cooling. Optimization parameters include skelp velocity, laminar cooling bank configuration, as well as side-spray conditions. The optimization techniques include two stochastic optimization methods (Genetic Algorithms and Particle swarm optimization) and one deterministic method (The branch-and-bound). A comparison between optimization methods showed that the branch-and-bound method can achieve global optimum faster than the stochastic techniques. The branch-and-bound method was used to set the coiling temperature, using three different cooling strategies (early, late and constant cooling), to reach the specified coiling temperature (550°C). Also the temperature profile optimizations was done, in order to maximize volume of the steel strip, which cool through a desired zone in Continues cooling transformation diagram, was done using branch-and-bound method.

Acknowledgements

First I would like to thank Dr. Amos Ben-Zvi, my supervisor, for his support, guidance and above all, encouragement throughout my Masters program. I have learned knowledge and its creation under his instruction that will enable me to make more contributions in the future. His careful revision enormously contributed to the production of this thesis.

I wish to express my thanks to Dr. J. Barry Wiskel and Dr. Hani Henein, for their valuable suggestions and insightful discussions on a variety of technical aspects of my work. My thanks also goes to Dr. Laurie Collins and EVRAZ Inc. NA, for their valuable inputs and support with experimental data.

I also owe thanks to the all the research group members, Katherine Johnson, Stephen Kenny, Ata Kamyabi, Nasseh Khodaei, and Hua Deng for their help in the middle of project and in revision of thesis. My special thanks also go to Fariborz Kiasi for his kind support and helpful suggestions.

My deepest gratitude goes to my family, for their endless love, wise education, and unconditional support throughout my life. Thanks for providing the best possible environment for me to grow up and get educated. You have always been a source of strength and encouragement.

Table of contents

List of tables

Table of Figures

Nomenclature

a	Thermal diffusivity
A_r	Surface area subjected to radiation
C_1	Self confidence factor
C_2	Swarm confidence factor
C_p	Specific heat capacity, J/Kg.K
f	Cost function
$h(x)$	Spatially distributed heat transfer coefficient function
h_a	Heat transfer coefficient of forced air convection
h_s	Strip thickness
h_w	Heat transfer coefficient between rolled metal and water
h_{wb}	Unknown coefficient
h_{wt}	Unknown coefficient
H_u	Latent heat of phase transformation
J	Cost value
k	Generation number
K	Temperature-dependent thermal conductivity
L_1	Length of impact zone, m
n	Rank of each individual
n^*	Number of nodes in water cooling section
NT	Nodal temperature
p	Probability of choosing each individual
P_i	Parent number i
r	Random value from interval (0,SUM)
r_1	Random number between 0 and1
r_2	Random number between 0 and1
S	Scaled cost values
$Speed$	Velocity of the strip
SUM	Sum of the entire individual's fitnesses in a generation
t_r	Radiation time

t_w	Water contact time
T	Temperature, K
T_a	Ambient temperature
T_w	Water temperature
x	Each individual in optimization
u	Velocity of the strip
v	Velocity of each individual
V	Volume of the body
w	Inertia factor
W_I	Jet impact diameter, m

Greek Letters

$\boldsymbol{\rho}$	Density of Material, m/s
ε	Emissivity, m^2/s^3
γ_u	Rate of phase transformation
σ	The Stefan-Boltzmann Constant
Δl	Water contact length
ΔT_r	Temperature change due to heat radiation

Subscripts

d	Digit number in each individual
f	Final value
i	Number of individual in a generation
k	Generation number
s	Specified value

Abbreviations

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
BB	Branch and Bound
CCT	Continuous Cooling Transformation
FEA	Finite Element Analysis
FEM	Finite Element Method
gbest	Global best
GA	Genetic Algorithms
GB	Global Best
lbest	Local Best
PB	Particle Best
PSO	Particle Swarm Optimization
ROT	Run-out Table
SI	Swarm Intelligence
TMCP	Thermo Mechanical Control Processing

1

1) Introduction

In the modern hot-rolling mill, a critical objective is to achieve the desired mechanical properties in the finished product. In this regard, microstructure control of metal is a key point in favorable hot working operations (Serajzadeh S., 2007). The microstructure of steel is influenced by the temperature-time profile of steel strip on the run-out table (A roller bed which carries the continuous moving steel strip at a high temperature from the exit of the hot rolling mill to a lower temperature which is then coiled into spools) (Biswas S.K., 1997). As a result, the temperature profile control during the cooling of steel is a significant challenge to achieving the microstructure that is responsible for the desired mechanical properties for the steel product. Control variables on the run-out table include water valves, strip velocity and side-sprays. The ability to control the thermal history of the steel on the run-out table is highly related to these control variables, which are the specific to an individual plant and an particular type of steel.

In order to achieve the desired properties of the steel at the end of the run out table, control systems were developed to maintain the coiling temperature (temperature at the end of the run-out table) within the required temperature range. To achieve this objective, it is essential to develop an accurate model of the thermal history of the steel through the runout table (Chatterje S., 2001). Simplified mathematical models (Biswas S.k., 1997) (Guo R.M., 1997) were used

in former works, however, these mathematical models could not predict the temperature behaviour of steel product accurately. The finite element method (FEM) (its practical application commonly known as finite element analysis, or FEA) is a numerical technique for finding approximate solutions of partial differential equations (PDE), as well as of integral equations. FEA enables one to simulate the temperature profile in a steel strip through the entire cooling process.

Several techniques have been developed to analyze the temperature profile of steel in cooling systems, including statistical methods with simplified heat transfer models, model reference controls, a combination of feed-forward feedback controls (Chatterje S., 2001), and identification of nonlinear run-out table cooling models using real-time rolling data (Guo R.M., 1991). Most of these techniques concentrate only on controlling the coiling temperature. The classical automation concepts compensate for the disturbances as a rule at the end or the beginning of the main cooling zone or in the radiation zone. The classical automation concept has enormous disadvantages in this connection, with lengthy cooling sections in particular (Latzel S., 2001). In order to achieve the desirable properties for steel slabs, temperature control at distances between the first and the end of the run-out table can help to ascertain the steel phases during the cooling process.

The temperature optimization of steel strip during the cooling process is conducted between all run-out table configurations, using a finite element thermal model. The optimization algorithms to find the optimal configuration between all possibilities are called global optimization techniques. Global optimization techniques have been used extensively due to their capability in handling complex engineering problems. Optimization methods in general can be classified into two main categories: deterministic and stochastic (heuristic). Deterministic methods require the optimization problem to have mathematical characteristics that do not exist in most of the simulation-based optimization problems. Stochastic optimization methods, however, are based on random search in the solutions' area

and do not guarantee the optimum result (Younis A., 2010). Binary variables are a subset of integer/discrete variables that are restricted to zero or one values. Binary variables are usually associated with yes or no decisions, for example, to turn a valve on or off.

Discrete problems of large scales are common in practice. Decision between yes and no, or on and off are samples of these discrete problems in industries. These problems need to be solved, but finding the optimum solution needs much time using deterministic methods. Instead, it is reasonable to concentrate on finding a reasonably accurate solution within limits of time and availability of computer memory. This means that heuristic methods are applicable in these cases. Genetic algorithms (GA) as stochastic optimization methods are straightforward to program, and they work directly with complete solutions, not with bounding functions (Chinneck J. W., 2010). Particle swarm optimization (PSO) is a recently introduced global optimization technique that has a straightforward formulation, which makes it easier to implement, and unlike Genetic Algorithms, has no evolution operators such as 'crossover' and 'mutation.' GA and PSO presented in this work optimized the configuration of the run-out table to achieve a desired temperature profile, while the FEA model of strip cooling was treated as a black box.

Branch-and-bound is the basic technique for solving integer and discrete programming. Although the results of stochastic optimization methods were acceptable in this work, it is possible to use the branch-and-bound technique for a cooling system. The optimization method is based on the observation that the enumeration of integer solutions has a tree structure. The time of optimization and the accuracy of the optimum result was the goal to find new optimization methods for this work.

Objectives and Contributions

The goal of optimization of steel cooling system is to achieve desirable mechanical and metallurgical properties for the steel. Those mechanical properties are highly dependent on the phase and microstructure of the steel strip during the cooling process. The phase and microstructure of the steel is determined by its temperature profile and cooling rate. The main goal of this work is, therefore, to develop algorithms that can optimize the control variables of a steel run-out table to achieve the desired temperature profile or thermal history of the steel as it cools along the entire length of the run-out table. The proposed algorithm was applied to a finite element model of the run-out table. The proposed algorithm would be evaluated on the basis of achieving an acceptable degree of output prediction. In addition, the algorithm should decrease computational cost and time when compared to optimization methods, which were previously used in this work. The specific objectives of the work presented in this thesis are:

1. Determine a suitable optimization method to optimize the temperature history on a run-out table using velocity of the strip, bank conditions and side-spray conditions as control variables.
2. Find an optimization method best suited for the configuration of the FEM thermal model.
3. Find optimization methods that are faster than stochastic (heuristic) optimization methods in finding the solution with higher accuracy for this problem.
4. Using the optimization algorithm to find the best configuration to achieve the desirable coiling temperature and cooling rate for steel during laminar cooling.
5. Utilize simulations, based on the designed inputs, to compare the performance of each configuration.

The following are the significant contributions of this work:

1. Optimized the temperature profile of a steel plate during the cooling process using control variables such as bank and side-spray conditions (on/off) and velocity of the strip.
2. Developed an optimization strategy by considering the characteristics of the cooling process and used branch and bound to achieve the best solution with reduced time.
3. Optimized the runout table configuration in order to achieve not only the desirable coiling temperature, but also the desirable temperature profile (including uniform and non-uniform cooling rates) through the whole runout table
4. Collected a database of all results from the simulations to eliminate simulation time during the optimization process.
5. Identified approaches for increasing the velocity of the strip while achieving the desired temperature profile for the steel strip on the run-out table .

Thesis outline

In this thesis, optimization procedures have been developed to control the temperature behaviour of steel during the cooling process. In Chapter 2 the background on issues related to this work is covered. These issues include the importance of the steel cooling model, the previous control techniques for temperature control on the run-out table, and the available optimization algorithms for this problem. In Chapter 3 description of the proposed optimization policy to achieve the desired temperature profile during laminar cooling is brought. In addition, in Chapter 3 an investigation on differences between three optimization methods (Genetic Algorithms, Particle swarm optimization and branch-and-bound) is presented in order to determine the fastest and most accurate solution for the optimization of the run-out table configuration. In Chapter 4 the optimization policy used to obtain the desired cooling rate, coiling

temperature, or both, using GA as an optimization method for the black box model (Finite element thermal model) is explained, and also evaluation of the sensitivity of the optimal solution to different control nodes is presented. In Chapter 5, the ability of the branch-and-bound method (BB) to achieve the optimal configuration to set the temperature profile on a desired non-uniform cooling rate is presented. Also, the BB method is used to manage the temperature of maximum possible volume of the strip in desired zone in continuous cooling transformation (CCT) diagram. Chapter 6 includes the summary of the results of the present work, conclusion of the entire thesis, and recommendations for directions for future work. The MATLAB scripts of optimization methods are presented in Appendix A. Chapters 4 and 5 are reprint of the papers, which were written and prepared for submission during Master of Science program.

This thesis has been presented in a paper format according to the requirements of the Faculty of Graduate Studies and Research (FGSR) at the University of Alberta. In order to connect the materials in different chapters and at the same time to ensure the completeness of individual chapters, there is some overlap between chapters.

References

(Biswas S.k., 1997) Biswas S.k. Chen S.J., Satyanarayana A. *"Optimal temperature tracking for accelerated cooling processes in hot rolling of steel"*, Dynamics and Controls. Vol 7, 1997, pp. 327–340.

(Chatterje S., 2001) Chatterje S. Simonelli G., Chizeck H. *"Parameter identification of a nonlinear coiling temperature model for run out table control at LTV Cleveland works in hot strip mill"*, Case Western Reserve University. 2001.

(Chinneck J. W., 2010) Chinneck J. W. Practical optimization : a Gentle Introduction [Book]. 2010.

(Guo R.M., 1997) Guo R.M. *"Modeling and simulations of run-out table cooling control using feed-forward-feedback and element tracking system"*, IEEE Transactions on Industry Applications. Vol. 33., 1997, p. 304.

(Guo R.M., 1991) Guo R. M. "*Development, Identification and Application of a nonlinear Model for Non-symmetric Runout Table Cooling"*, American Institute of Steel Engineers Annual Conference Record. Pittsburgh, PA. 1991.

(Latzel S., 2001) Latzel S. *"Advanced Automation Concept of Run-out Table Skelp Cooling for Hot Skelp and Plate Mills"*, Transactions on Industry Applications . Vol. 37, 2001, p. 1088.

(Serajzadeh S., 2003) Serajzadeh S. *"Prediction of temerature distribution and phase transformation on the run-out table in the process of hot stri rolling"*, Applied mathematical modelling. 2003, pp. 861-875.

(Wiskel J.B., 2011) Wiskel J.B Deng H.,Jefferies C., Henein H., *"Infrared Thermography of a TMCP Microalloyed steel skelp at the upcoiler and its application in quantifying the laminar jet/skelp interaction"*, ISM. No 1 : Vol. 38, 2011, pp. 35-44.

(Younis A., 2010) Younis A. Dong Z. *"Trends, Features, and tests of common and recently introduced global optimization methods"*, Engineering Optimization. No 2 : Vol. 48, August 2010, pp. 691-718.

2

2) Background

Introduction

This chapter contains the relevant background information pertaining to this thesis. Section 2.1 is a general introduction to steel industry, hot rolling mill and steel cooling configuration including run-out table. Section 2.2 reviews the finite element and mathematical models used to simulate the cooling process. In addition, the model used in this work is explained in Section 2.2. In Section 2.3, methods, which were used previously, to control the temperature on the run-out table, are presented, and the relevant methods for the optimization in this work are discussed.

2.1 Processing steel

The current production of steel around the world is approximately 1200 million tons per year (Hunt P., 2011). As such, it is one the most influential metals in the world today. Steel mills turn molten steel into ingots, blooms, slabs through casting and in turn these cast products are transformed into sheet, by hot and cold rolling. The mechanical properties of steel can be customized during processing by subjecting the steel to different cooling practices and manufacturing

treatments. The following sections describe the steel processing for skelp manufacture through the hot rolling mill.

2.1.1 Hot rolling mill

A rolling mill is a general term used to describe a wide range of manufacturing processes by which a strip of sheet metal is deformed continuously. A typical hot rolling operation will include a reheat furnace, a roughing mill, a finishing mill and a run-out table.

In a hot rolling mill, the steel slabs are reheated in a heating furnace to a temperature near 1200℃, rolled sequentially by the roughing and finishing mills, and turned into strip anywhere from 1 to 20 mm in thickness. After being rolled, the strip goes through the run-out table and is cooled to about 500℃ and coiled. The hot rolling mill is illustrated in Figure 2-1.

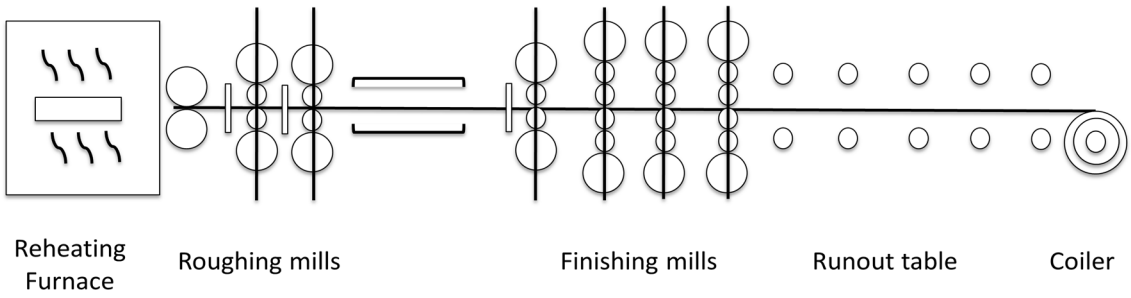

Figure 2-1. A typical of hot rolling mill

2.1.2 Steel cooling configuration

In the run-out table (ROT), the steel strip is cooled by water from the top and the bottom. Temperature control on the ROT is an essential tool used to ensure the desired mechanical properties of a steel strip are obtained. There are three main types of cooling systems used for the controlled cooling process of steel strip on a

ROT: water sprays, laminar flow streams and water curtain, as shown in Figure 2-2 (Xu F., 2006).

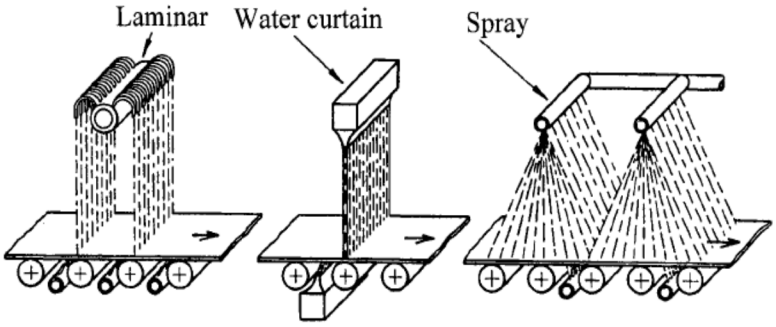

Figure 2-2. Controlled cooling systems (Xu F., 2006)

In a water-spray system, the water as coolant impinges from a row of sprays onto the surface of the strip. The water spray is often used to cool down the bottom surface of the strip.

Laminar flow streams are used since they can penetrate the vapour film on the cooled surface and increase the coolant residence time. Each laminar flow stream consists of a number of U-pipes. This system creates low pressures in laminar flow streams and, therefore, has a higher cooling efficiency than the water spray system (Xu F., 2006). In the water curtain system, the strip is cooled by a planar jet, which spans the entire width of the strip (Xu F., 2006). This cooling system is not addressed in this study and will not be discussed further.

In this work, only a laminar cooling system will be discussed. The laminar flow cooling system in this work includes six water banks and side-sprays located after each of the water banks. Each water bank includes a set of U-pipe water valves in a row. The side-sprays remove water from the surface of the skelp (using air flow) and will reduce the amount of heat removed by the water from the skelp. In the ROT of interest in this work, the velocity of the skelp can be varied, the water bank condition is considered to be either on or off, as are the side-sprays (on or

off). A schematic of the laminar cooling system used in this work is shown in Figure 2-3 and includes the relative location of the cooling banks and side-sprays used. The system shown in Figure 2-3 has three cooling zones: 1) The radiation zone before the first water bank; 2) the water-cooling section represented by six water banks in Figure 2-3; and 3) the radiation zone located between the last water bank and the coiler. The heat transfer in the water-cooling section is the main method by which the temperature of the skelp is controlled.

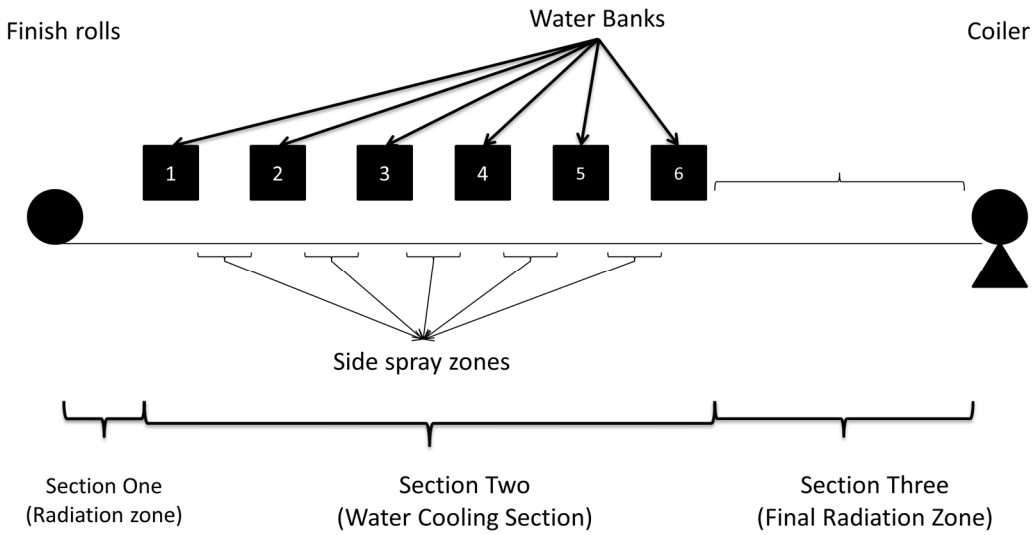

Figure 2-3. General lay-out of a run-out table

2.2 Modelling the heat transfer of steel cooling

A key component in optimizing the temperature profile of the steel is the use of an accurate thermal model of the steel transiting the ROT. It, therefore, becomes necessary to understand, predict, and successfully simulate the controlled cooling process of the skelp. Numerous investigations have been performed to provide insight into the fundamentals of the complex cooling process.

2.2.1 Classical heat transfer models for the run-out table

In this section, the models developed and described in the literature to simulate the cooling process are reviewed. 3D models (i.e. models which considered all physical dimensions of the strip including length, width and depth of the strip) (Xu F., 2006), which simulate the heat transfer in complete detail, are not presented, as those models require lots of time to run and are not practical for control or optimization of the run-out table. In Figure 2-4, the shape of steel strip is illustrated. The physical dimensions of the strip are length, thickness and width. In addition, time can be considered as the fourth dimension of the heat transfer model of the strip.

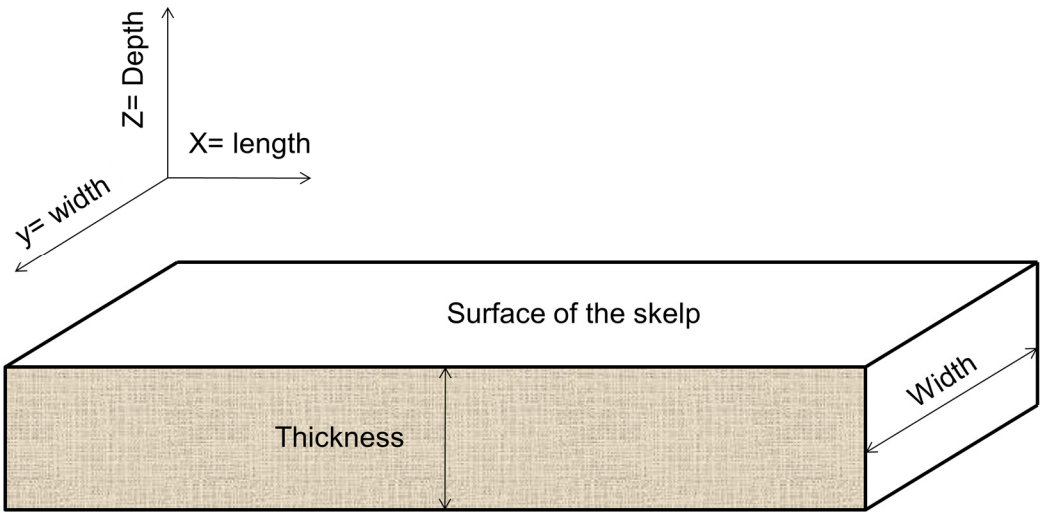

Figure 2-4. Dimensions of a typical steel strip on the run-out table

In the run-out table of a hot strip mill, the strip is cooled by water from the top and bottom. The prime modes of heat transfer on the ROT are convective heat transfer with water and radiative heat transfer. Conduction along the thickness of the strip and conduction to the support rolls also affect the temperature of the strip in the run-out table. The latent heat and thermal conductivity of steel vary with the temperature of the strip during the cooling process (Kumar R., 1996).

Thermo-physical models are those in which the changes in phase of the steel strip are considered in the heat transfer of steel, such as the models of Kumar, Latzel and Serajzadeh (Kumar R., 1996) (Latzel S., 2001) (Serajzadeh S., 2003).

Hinrichsen (Hinrichsen E.N., 1976) modelled the cooling process by a first-order equation for heat transfer with a time constant. Sonehara (Sonehara M., 1987) used a lumped model with constant surface heat transfer coefficient to model the heat transfer of the run-out table. Yaniro (Yaniro K., 1991) used a linear one-dimensional heat conduction equation considering the heat transfer in the thickness direction in the strip to model the strip temperature during the cooling process. However, none of these models considered the variation in heat transfer along the length of the run-out table (Uetz G., 1991) (Yanangi K., 1993).

In 1997, Biswas expanded the heat transfer model in his work to two-dimensions (i.e. in the length and the depth of the strip or in the moving and the thickness directions of the strip); however, he did not consider the effects of conductive heat transfer by contact between strip and support rolls and radiation in his model (Biswas S.k., 1997). Guo (Guo R.M., 1997) used an adaptive model (i.e. the model parameters are tuned online during cooling process), and considered convective, conductive, and radiative heat transfer along the entire cooling process. The parameters in the mathematical heat transfer equations were tuned in order to track the coiling temperature correctly (i.e. adaptive model). However, Guo did not consider the variation of latent heat of phase transformation or dependence of thermal conductivity with temperature. The heat transfer model in Samaras' work (Samaras N.S., 2001) was the same as Guo's, with differences in mathematical equations for radiative and convective heat transfers. In addition, Samaras did not use an online system to tune the heat transfer coefficients of the model. In addition, Samaras did not consider the heat losses by convection and conduction to the work rolls.

The heat transfer models of previous researchers were validated with industrial data from their specific plants. Heat transfer models were mainly used to predict the temperature of the steel skelp (specifically the coiling temperature) at the end of the cooling process, to control the properties of steel product. However, the mechanical properties are dependent on the constituents formed during the cooling process, which is a function of both cooling rate and coiling temperature. These constituents can be determined using CCT diagram, which is explained in the former section. Therefore, there is a need for a model that accurately predicts the temperature-time or temperature-length profile and the skelp temperature at the end of cooling process. There is also a need for a model that is validated with industrial data from the target cooling plant. In Appendix A, the mathematical equations and characteristics of models developed by other ROT researchers are represented.

Heat transfer models were used to predict the temperature of the steel skelp (specifically the coiling temperature) at the end of cooling process to control the properties of steel product; however, the mechanical properties are dependent on the constituents formed during the cooling process, which is related to cooling rate and coiling temperature together. Therefore, there is a need to have a model that can accurately predict the temperature of the skelp not only at the end of cooling process, but throughout the entire length of ROT.

2.2.2 FEM thermal model

The finite element thermal model of the run-out table developed by Wiskel et al. (Wiskel J.B., 2011) was used in this work. The dynamics of the system are represented by the temporal and spatial variation of temperature.

A key component of the model is the heat transfer boundary conditions that exist at the water/skelp interface. The model was developed based on the physical dimensions of a run-out table including the position of all headers and side-sprays, as well as the velocity of the strip on the run-out table.

The cooling system consists of a series of headers (individual nozzles) in the header extending across the skelp width. The cooling conditions directly under a laminar jet impacting a flat plate are complicated and include direct single-phase convection cooling, nucleate transition boiling, and film boiling.

Based on the geometry of the system, the following assumptions were made in the model:

1) The heat flow in the longitudinal direction of the skelp was not considered (i.e. only a transverse slice of the steel skelp was modelled).
2) Symmetry between the top and bottom faces of the skelp was assumed.
3) The heat transfer due to water flow from each of the nozzle jets was considered identical and independent of other nozzles.
4) Radiation heat transfer was assumed for conditions of no water on the surface (i.e. side-sprays)

A schematic of a portion of a laminar cooling system is shown in Figure 2-5. Included in this figure are two headers and two side-sprays on the run-out table. Also included are the heat transfer of regions associated with both impact and the residual water regions, defined by either direct impact cooling or water boiling, respectively. The transverse slice of the steel slab used in the model is shown with a dashed box in Figure 2-5.

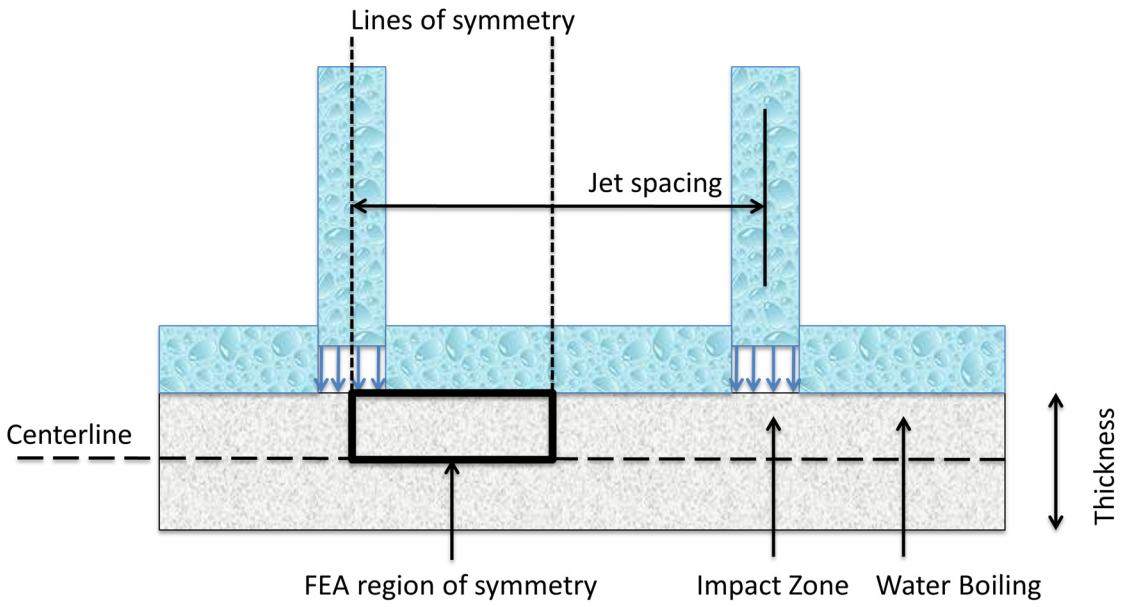

Figure 2-5. Schematic of laminar cooling / skelp interaction transverse to the arrow of the velocity of the strip (Wiskel J.B., 2011)

A top view of the skelp can show the exact position of the simulated region in the model (Figure 2-6). The region has dimensions of W_1, and L_1, where the value of former is equal to d_c =10 mm and the L_1 is two times of d_c.

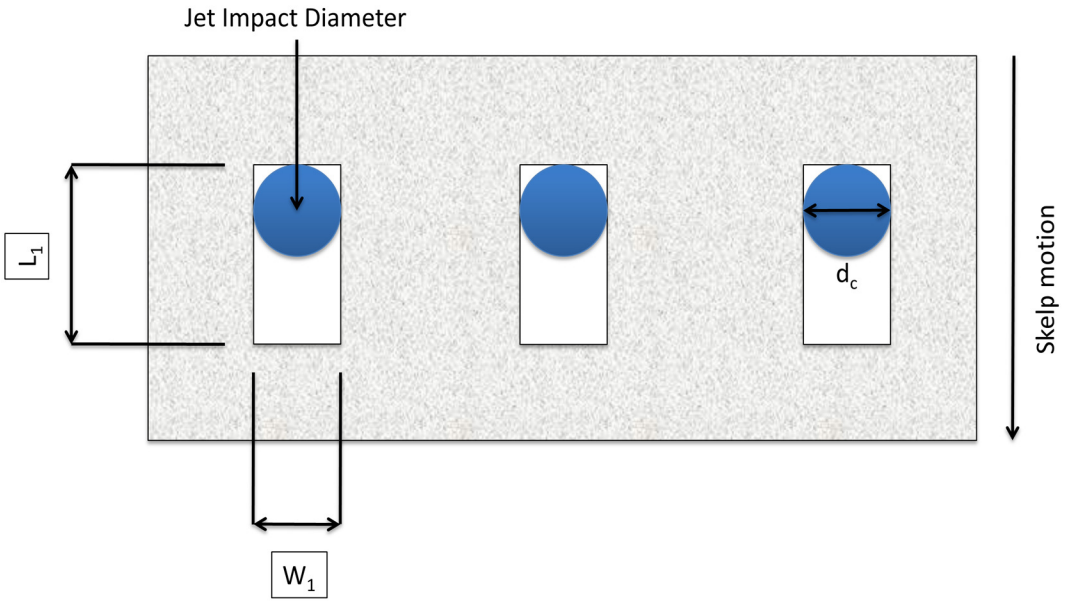

Figure 2-6. Schematic of laminar cooling / skelp interaction transverse to skelp motion (Wiskel J.B., 2011)

The transverse two-dimensional finite element mesh used to model the steel skelp is shown in Figure 2-7. Included in the figure is the impact zone and water boiling heat transfer zone. In addition, the dotted lines indicate planes of symmetry.

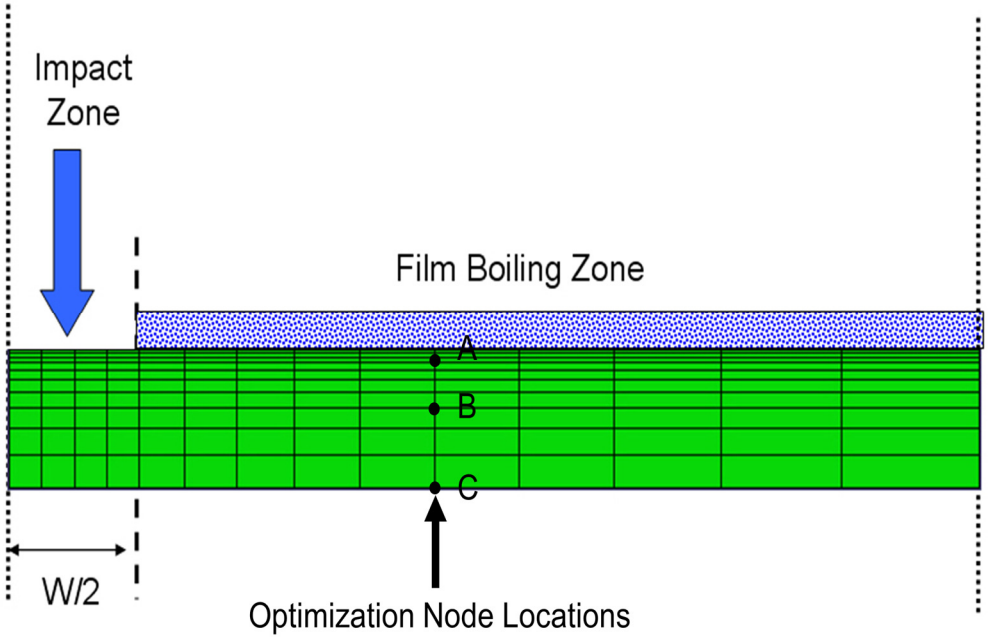

Figure 2-7. Schematic of FEM mesh and control Node location (Wiskel J.B., 2011)

Node B is considered as representative of the strip nodes for most of optimizations, because it is physically in the middle of the all nodes in the simulated region. Nodes near the surface are not good choices for the control node because of the fluctuation in their temperature profile through the length of the ROT.

The thermo-physical properties of steel for the FEM model include temperature-dependent thermal conductivity (K) and heat capacity (specific heat) (C_p). Thermal conductivity, K, is the property of a material's ability to conduct heat. Specific heat capacity is the measurable physical quantity that characterizes the amount of heat required to change a substance's temperature by a given amount.

The variation of thermo-physical properties with temperature, obtained from the literature, are presented in Table 2-1. The values of properties between these values of temperature were linearly interpolated.

Table 2-1. Thermo physical properties of X-70 (Wiskel J.B., 2011)

T(°C)	25	100	200	300	400	500	600	700	800
K(W/m^2°C)	59.5	57.8	53.2	49.4	45.6	41.0	37.0	33.1	28.5
C_p(j/Kg)	481	500	528	553	595	662	754	867	867

As it is presented in Figure 2-7, the two dimensional mesh of the steel skelp, includes 459 nodes. These nodes include nodes on the intersections of lines and in the middle of two intersections. Using FEM thermal model for each input (i.e. staring temperature and ROT configuration), the model gives a temperature-time profile for each node through the entire length of the ROT. In Figure 2-8, the final temperature for each node at the end of cooling process is presented with colors. The hottest node is the node at the bottom and end right of the box, and the coolest node is at the surface and end left of the simulated zone.

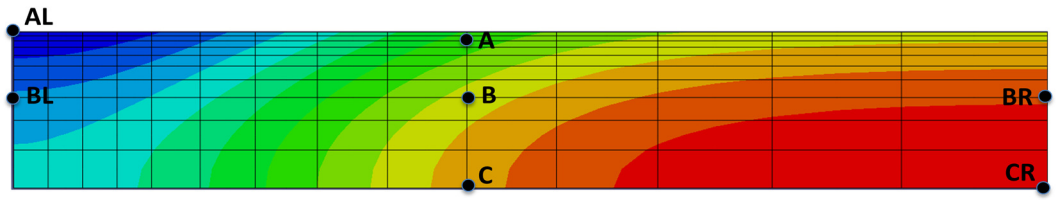

Figure 2-8, Final temperatures of all nodes in the simulated box presented in colors

The temperature-time profiles for nodes, which are shown in Figure 2-8 through the entire length of the ROT are presented in Figures 2-9, 2-10 and 2-11. The ROT configuration for these results is considered as all banks “on”, all side-sprays “off”, and the strip velocity equal to 4 m/s. These figures show the differences in temperature history through the thickness and along the length of the strip.

Figure 2-9 presents the temperature time profile for Nodes A, B and C, which are on a vertical line. Temperature profiles of these nodes represent the difference in temperature behavior for different nodes through the thickness of the skelp. As it can be seen in the figure Node A has lower temperature in comparison with temperature of Nodes B and C, because it is closer to the surface of the skelp that is in contact with the cooling water.

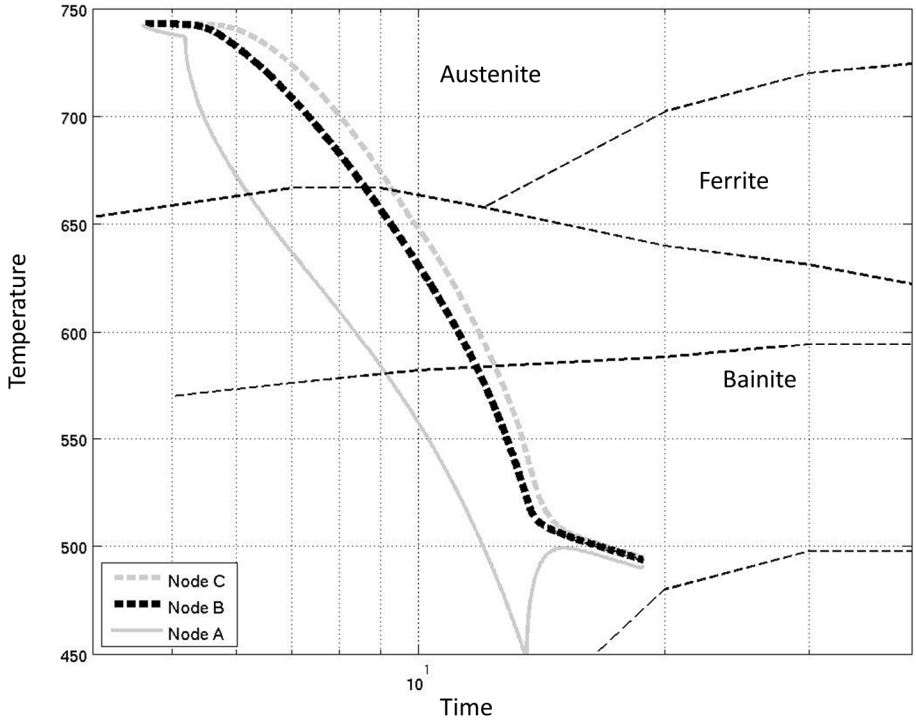

Figure 2-9, Temperature profile of Nodes A, B and C through the entire length of the ROT

In Figure 2-10, the temperature profile of Nodes B, BL and BR are showed in a CCT diagram. The temperature history of these nodes are similar together, because they have the same distance from the surface of the skelp; however the nodes on the left side of the simulated area are cooler than nodes on the right side of the simulated area.

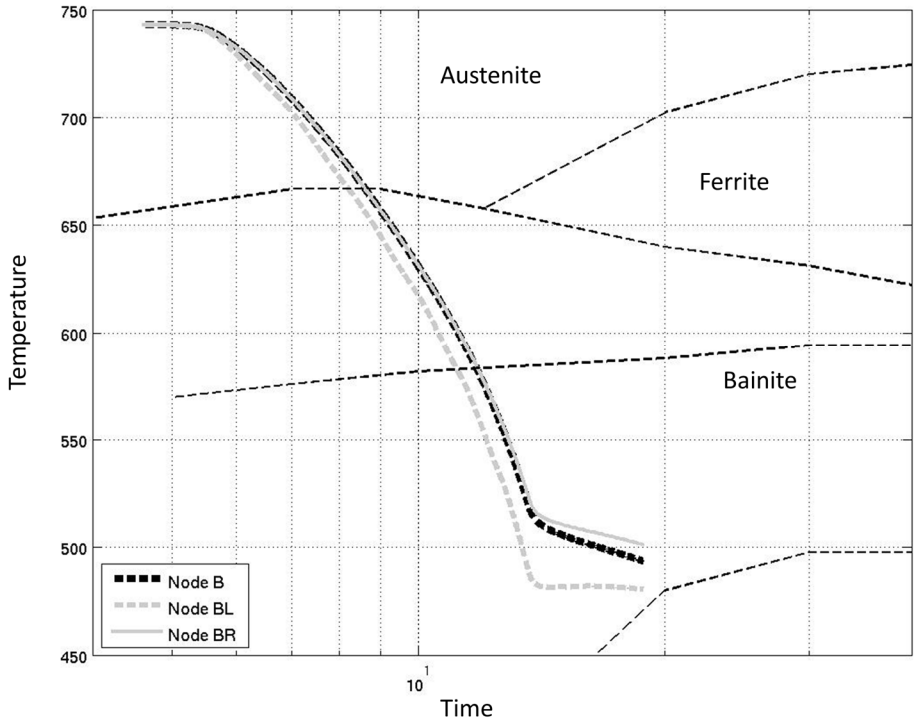

Figure 2-10, Temperature profile of Nodes B, BL and BR through the entire length of the ROT

In Figure 2-11, the temperature profiles of Nodes B, AL and CR are compared. Node AL is the coolest node in simulated box and Node CR is the hottest node through the entire cooling process on the ROT. Node AL is at the surface of the skelp, so it has temperature fluctuations during cooling process. The reason for these fluctuations is the effect of impact zone under each water jet on the temperature of the surface nodes.

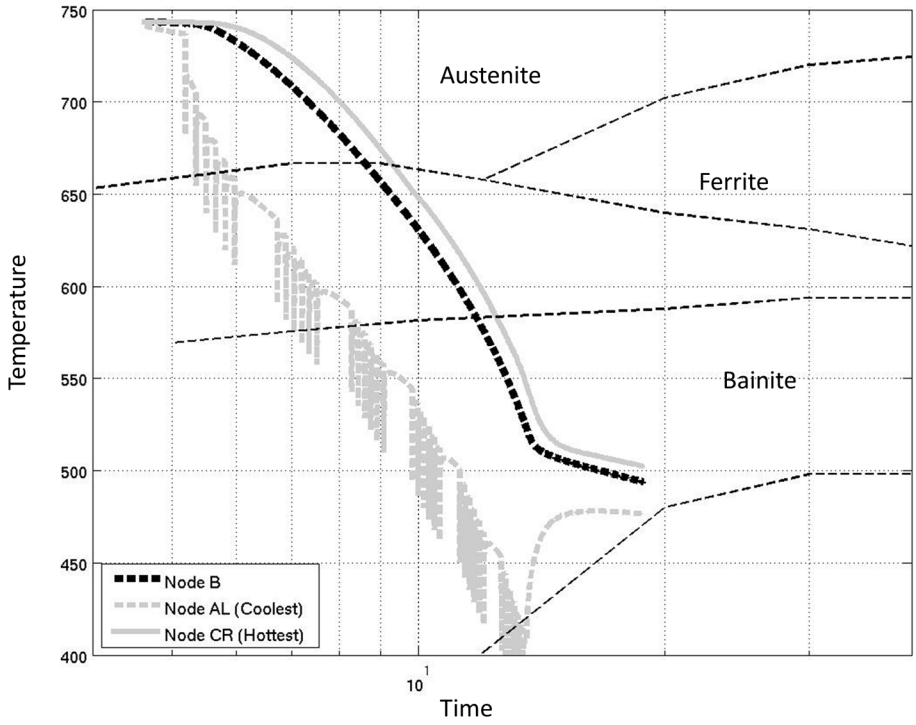

Figure 2-11, Temperature profile of Nodes B, AL and CR through the entire length of the ROT

It can be seen in figures that the temperatures of nodes from left to right, and from surface to center of the simulated box are increasing. Also, the temperature profiles of all nodes in the middle of simulated area are lower than the temperature profile of the hottest node, and higher than the temperature profile of the coolest node. Using temperature of different nodes it is possible to estimate the volume of the skelp which is in a temperature zone during cooling process. It is important to set temperature the maximum possible volume of the skelp in a determined zone during laminar cooling to achieve the desired properties for steel at the end of the ROT. The properties of steel are dependent to all these temperatures (i.e. phases) during and at the end of the cooling process.

In order to show the differences in the ROT configurations, the temperature profiles of Nodes A, B and C are compared for another ROT configuration. The

configuration includes the first three banks “on”, the last three banks “off”, the first three side-sprays “on”, the last three side-sprays “off”, and the velocity of the strip equal to 4 m/s. The results show the differences, also the effect of side-sprays after each banks can be seen in the temperature profile of Node A, which is near the surface of the skelp.

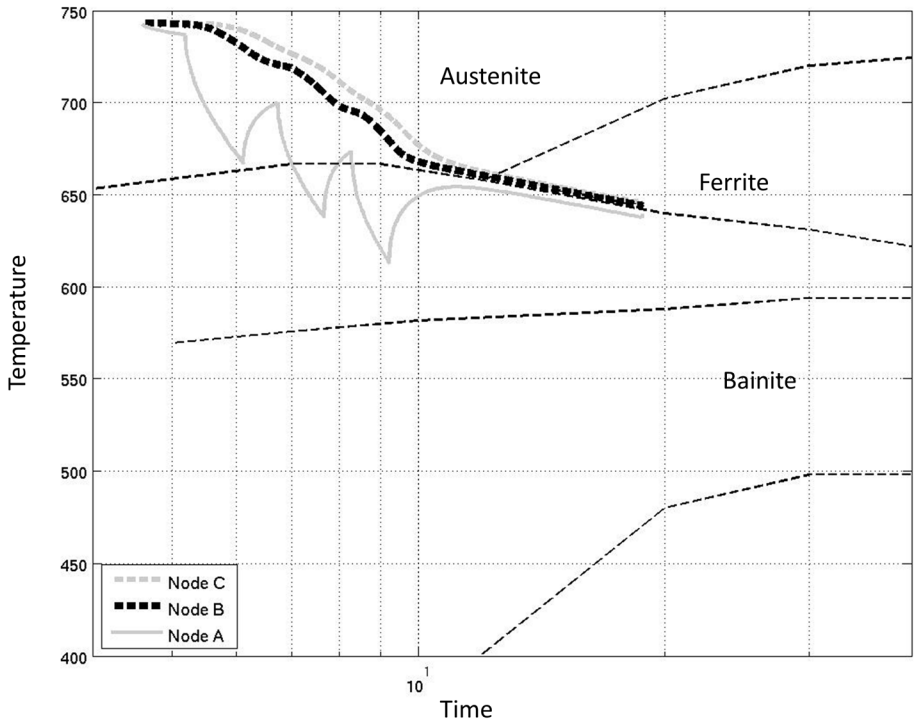

Figure 2-12, Temperature profile of Nodes B, AL and CR through the entire length of the ROT- The configuration of the runout table is the first three banks and their side-sprays are “on”, the last three banks and their side-sprays are “off” and the velocity of the strip is 4 m/s

2.3 Controlling the cooling process

The cooling process strongly influences the microstructure in the steel product being cooled. Consequently, the finished product’s metallurgical and mechanical

properties dictate the desired temperature profile required during the cooling process. This target temperature profile is successfully satisfied through a proper control that is capable of achieving the target cooling rate and coiling temperature on the run-out table. Development of such an optimization strategy is the primary goal of this work.

2.3.1 Classical control methods for the run-out table

In order to have a dominant and better way to set the steel temperature on the desired temperature profile during laminar cooling, it is needed to review previous works on modelling of the heat transfer on the ROT and on controlling the temperature of the steel strip in cooling section. Knowing those control methods enable us to develop a technique, which is better and faster, also, achieve more than what previous methods got. Previous control methods include online control (i.e. feedback, feed-forward loops), automation concepts, and optimization techniques. This section reviews previous control attempts, automation concepts and optimization techniques, which are used on the ROT, to achieve desired coiling temperature or temperature profile.

Historically, temperature control at the run-out table was done by presetting the water valves at certain fixed positions based on heuristics (Biswas S.k., 1997). The first investigations on the run-out table cooling control started in the 1960s (Guo R.M., 1997). Simple heat transfer models for run-out table were some of the first developing stages in cooling control (Auman P.M., 1967).

Most control systems used in hot strip mills are based on feedback loops, with variations in the strip gauge, the finishing temperature and the velocity of the strip; the feedback control system alone, however, cannot provide the control accuracy without an auxiliary feed-forward loop. Using feed-forward control and model reference adaptive techniques, the control methods improve control accuracy for steel products (Hinrichsen E.N., 1976, Moffat et al. R.W., 1985).

In classical automation concepts for ROT, there were two sets of pyrometers used for the control of the laminar hot strip cooling section – one at the beginning and one at the end of the run-out table – to report the finishing and coiling temperature for feedback and feed-forward loops usage (Latzel S., 2001). The effect of disturbance on each strip point differs; therefore, the automation can only compensate for the disturbances at the end or the beginning of the main cooling zone or in the radiation section (Latzel S., 2001).

Latzel (Latzel S., 2001) proposed an advanced automation concept, based on operational experience with the classical concept, for the automation of cooling sections. A thermo-physical model was used for setup calculation and control during production. A process observer calculates the actual (real) temperature profile, taking all measurements, including finishing temperature, water flow, strip speed, etc. into account. The difference between the reference and this observed temperature is individually controlled by the temperature closed-loop control using the water flow of the cooling headers. A monitor closed-loop control compensates remaining temperature deviations between reference and measured temperature at the coiler pyrometer location. Parallel to the strip production, deviations between the observed and measured coiling temperature are adapted. Property coefficients of the thermo-physical model were consequently tuned using adaptation coefficients. This system has enormous advantages, as the influence of process disturbances (e.g. by power speed-up and speed-down), is effective simultaneously at each point within the cooling section.

Guo (Guo R.M., 1997) introduced a control program that features three control loops: feed-forward temperature control, intra-coil feedback loop, and coil-to-coil adaptive loop. The control system is used in conjunction with an offline mathematical heat transfer model. A mathematical model was developed by Guo from combining academic and industrial research findings (Guo R.M., 1991). The tracking system with speed estimator and a flow modification factor made Guo's

system particularly strong. The control system could activate or deactivate each header to control the coiling temperature on the run-out table.

Xie (Xie H. B., 2006) controlled the coiling temperature by a feedback feed-forward and self-adaption loop. The process model was used to calculate the necessary water flow rate. The mathematical heat transfer model included two parts for radiation and water cooling. In addition, the model is used for supporting the feed-forward and feedback control systems, self-adaption, and optimization of model. The proposed control strategy and parameter regression method for the control system could control coiling temperature with high precision, in the range of ±15 °C. In addition, Xie introduced two layer-automation levels: process control level and basic automation level (Xie H.B., 2007). The process control level had the task of process optimization based on the cooling model, and the basic automation level determined the real-time regulation to keep the derived set point standing in the permissible range. The process control level includes feedback and feed-forward control, element tracking, self adaption control, model evaluation and dynamic setup. Using these tools in this layer, an online adaptive calculation method for improving accuracy of strip coiling temperature control on ROT was developed. The control accuracy of the coiling temperature in Xie's most recent work has improved up to 95.6% within ±15 °C.

Samaras (Samaras N.S., 2001) developed a dynamic programming control algorithm (i.e. optimization technique) based on Bellman's principle of optimality to control the coiling temperature on the run-out table. The manipulated variables included 30 on/off conditions for the headers during the cooling process. The final performance in terms of coiling temperature error minimization was confirmed via simulation analysis.

The main objective in most of the preceding works was to control only the coiling temperature at a predetermined value (Xie H. B., 2006), (Peng L., 2008),(Biswas S.k., 1997). There were a few works, however, which were multi-objective and

considered not only the coiling temperature control but also a middle target temperature or cooling rate (temperature profile) control during the cooling process (Xie H.B., 2007, Latzel S., 2001). In other word, the importance of temperature-time profile of the steel strip on the microstructure of the product steel from the start to the end point of the run-out table was not considered in the previous researches. In addition, the manipulated variables included water jet flow (Biswas S.k., 1997) and header on/off conditions (Guo R.M., 1997, Samaras N.S., 2001); however, manipulating the velocity of the strip on run-out table can definitely empower the control policy to set the temperature profile according to a desired cooling strategy.

2.4 Temperature profile and steel microstructure

The temperature profile or cooling rate of steel has a dominant effect on the final grain size and microstructure of the rolled material. For example, the higher cooling rate, leads to a greater under-cooling, a higher nucleation rate of the new phase and a microstructure of a finer grain size (Serajzadeh S., 2003). If the cooling rate of the austenite exceeds a critical value, the transformation can change from a diffusional control process to a displacive mechanism, forming the martensite constituent during rapid cooling (Serajzadeh S., 2003).

The continuous cooling transformation (CCT) diagrams are used to represent which types of phase changes will occur if a material as it is cooled at different rates. Using the CCT diagram, it is possible to determine the phase of steel according to the steel cooling rate. In Figure 2-13, three different cooling rates are shown in the CCT diagram. As it is shown in previous section, different ROT configurations lead to different cooling rates for the each node in the skelp. Each cooling rate leads to determined phases formed during the cooling process. Controlling phases and cooling rates of steel enable us to achieve the desired mechanical properties of product steel. The cooling rate equal to 11.51°C/s is

chosen to achieve the point between three phases during the cooling process; however, it is hardly possible to reach this cooling rate accurately in a steel plant. The point between three phases, which is called triple point in this research, is a critical point, because of different phases around it. A little difference in temperature around this point leads to different phase for steel and different properties for the final product. In addition, achieving the accurate 11.5°C/s as the cooling rate is not completely possible with optimization tools in this process.

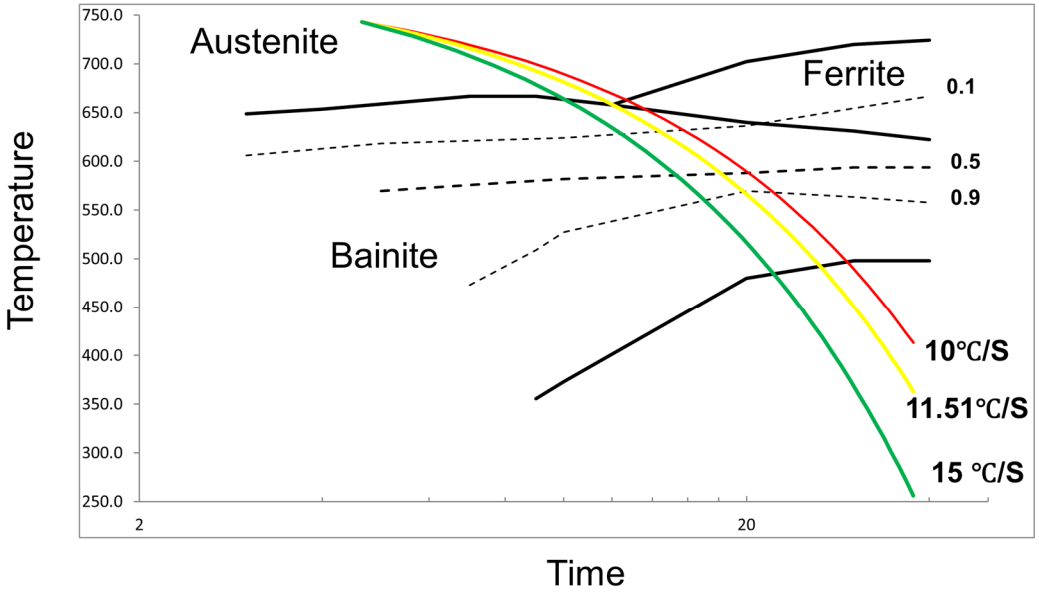

Figure 2-13. A sample CCT diagram for a micro-alloyed steel

The cooling strategy along the ROT is determined according to the steel grade, desired coiling temperature, and initial strip temperature. Also, the process should run at the fastest possible velocity to have a high productivity. Typical cooling strategies include early, uniform, and late cooling. In Figure 2-14 these different cooling strategies are illustrated on a CCT diagram. Late cooling will produce a microstructure with a ferritic microstructure; on the other hand, early cooling will result in a microstructure primarily composed of bainite.

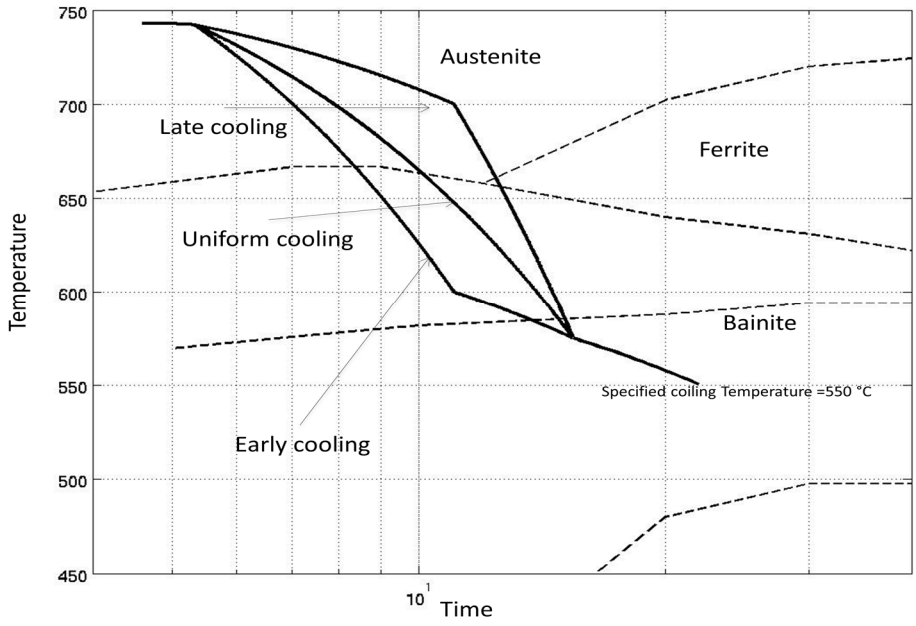

Figure 2-14. Sketch of cooling strategies: early, late, and uniform cooling

Previous researches on the ROT were concentrated on modeling of the cooling process to simulate the coiling temperature exactly. Also, some researchers tried to model the whole cooling process, using a 3D model (i.e. considering heat transfer in all directions: thickness, width, and length of the ROT). However, the former researchers did not consider the fact that the temperatures (i.e. phases) of the steel strip in the time between the start point and the end point of the ROT influence the final properties of product steel. The latter group modeled the ROT with all details, but the models are not suitable to use for control or optimization concepts. Because of the time which is needed for each simulation, those models are not proper for optimization loops that need a number of simulations for each run. The model which is used in this work simulates the temperature of the steel skelp through the entire length of the ROT, in addition the model can be used in the optimization of this work. In Chapter 3, the optimization policy and the optimization methods used to determine the configuration of the run-out table are explained.

References

(Auman P.M., 1967) Auman P.M. Griffits D.K., Hill D.R., *"Hot strip mill run-out table temperature control"*, Iron an steel Engineer. Iron an steel Engineer, 1967, pp. 174-179.

(Biswas S.k., 1997) Biswas S.k. Chen S.J., Satyanarayana A. *"Optimal temperature tracking for accelerated cooling processes in hot rolling of steel"*, Dynamics and Controls. Vol. 7, 1997, pp. 327–340.

(Chatterje S., 2001) Chatterje S. Simonelli G., Chizeck H. *"Parameter identification of a nonlinear coiling temperature model for run-out table control at LTV Cleveland works 84-In hot strip mill"*, Case Western Reserve University. 2001.

(Chinneck J. W., 2010) Chinneck J. W. *"Practical optimization : a Gentle Introduction"*, 2010.

(Darken L.S., 1953) Darken L.S. Gurry R.W., *"Physical chemistry of metals, New York"* : McGraw-Hil, 1953.

(Guo R.M., 1991) Guo R. M. *"Development, Identification and Application of a nonlinear Model for Non-symmetric Runout Table Cooling"*, American Institute of Steel Engineers Annual Conference Record. Pittsburgh, PA. 1991.

(Guo R.M., 1997) Guo R.M. *"Modelling and simulations of run-out table cooling control using feed-forward-feedback and element tracking system"*, IEEE Transactions on Industry Applications. Vol. 33, 1997, p. 304.

(Hinrichsen E.N., 1976)Hinrichsen E.N. *"Hot strip mill run-out table cooling- A system view of control, operation and equipment"*, AISE yearly Proceedings. 1976, pp. 403-408.

(Hunt P., 2011) Hunt Phil World Steel Production Report [Report]. ISSB Ltd, 2011.

(Kumar R., 1996) Kumar R. Lahiri A.K., *"Modelling of the cooling process on the runout table of a hot strip mill"*, IEEE, 1996.

(Latzel S., 2001) Latzel S. *"Advanced Automation Concept of Run-out Table Skelp Cooling for Hot Skelp and Plate Mills"*, IEEE Transactions on Industry Applications . Vol. 37, 2001, p. 1088.

(Moffat et al. R.W., 1985) Moffat et al. R.W. *"Computer control of hot strip coiling temperature with variable flow laminar spray"*, Iron an steel engineer. 1985, pp. 21-28.

(Samaras N.S., 2001) Samaras N.S. Simaan M.A. *"Novel Control Structure for Runout Table Coiling Temperature Control"*, AISE steel technology. No 6 : Vol. 87, June 2001. pp. 55-59.

(Serajzadeh S., 2003) Serajzadeh S. *"Prediction of temerature distribution and phase transformation on the run-out table in the process of hot stri rolling",* Applied mathematical modelling. 2003, pp. 861-875.

(Sonehara M., 1987) Sonehara M. Yamane T., Yuasa Y., *"New temperature control system of hot strip mill run-out table"*, NKK technical reort. Vol. 129, 1987. pp. 9-14.

(Uetz G., 1991) Uetz G. Woelk G., Bischops T., *"Influencing the formation of the steel structure by suitable temperature control in the run-out table sections of hot strip mills"*, Steel research. No 5 : Vol. 62, 1991, pp. 216-222.

(Wiskel J.B., 2011) Wiskel J.B Deng H.,Jefferies C., Henein H., *"Infrared Thermography of a TMCP Microalloyed steel skelp at the upcoiler and its application in quantifying the laminar jet/skelp interaction"*, ISM. No 1 : Vol. 38, 2011, pp. 35-44.

(Xie H. B., 2006) Xie H. B. Liu X. , Wang G. , Zhang Z., *"Optimization and Model of Laminar Cooling Control Thickness measurement System for Hot Strip Mills"*, JOURNAL OF lRON AND STEEL RESEARCH, INTERNMIONAL. No 13(1) : Vol. 9, 2006. pp. 18-22.

(Xie H.B., 2007) Xie H.B. Jiang Z.Y., Liu X.H., Wang G.D., Zhou T.G., Tieu A.K., "On-line optimization of coiling Temperature control on Run-out table for hot strip mills", Key Engineering Materials. Vols. 340-341, 2007. pp. 701-706.

(Xu F., 2006) Xu F. *"Finite element simulation of water cooling process of steel strip on tun out table"*, PhD thesis in Univeristy of British columbia, 2006.

(Yanangi K., 1993) Yanangi K. *"Prediction of strip temperature for hot strip: a combination of physical modelling, control roblems and practical adaption"*, Trans on Automatic control. No 7 : Vol. 38,1993. pp. 1060-1065.

(Yaniro K., 1991) Yaniro K. Yamasaki M., Furukawa M., et al., *"Development of coiling temperature control system on hot strip mill"*, Technical report. Vol. 24, 1991.

3

3) Optimized configuration and optimization methods

Introduction

This chapter contains an explanation of the optimization policy used in this work, as well as a explanation of the optimization algorithms for Genetic Algorithms, Particle Swarm Optimization, and Branch-and-Bound. The last section of this chapter includes a comparison between optimization algorithms with respect to convergence time and accuracy of results.

3.1 Optimized configuration

The optimized configuration indicates the configuration of the run-out table during the cooling process to set the temperature profile on the desired one. The temperature-time profile during the cooling process (i.e. simulation of the cooling section of a hot rolling mill) was optimized by an FEM thermal model, in conjunction with an optimization algorithm.

3.2 Optimization algorithms

In this section, the algorithms of the global optimization methods, which were used in this work, are explained. Three methods are used in this work: two are population-based evolutionary optimization methods, while the third is branch-and-bound. The stochastic optimization methods (i.e. GA and PSO) are random search methods and cannot guarantee to achieve the global optimum. Branch-and-bound, on the other hand, as a deterministic optimization method, is used to decrease the number of simulations to achieve a global optimum.

3.2.1 Genetic Algorithms

A genetic algorithm (GA) is an optimization method inspired by Darwin's theory on natural selection and Mendel's work on gene inheritance. According to Darwin's theory, the initial population will generate the next generation by mating the best pairs with respect to their fitness values, while genetic mutation will cause random genes to search in different areas of the cost function (Avilla V.H., 1994). In a GA, each possible configuration is expressed as a string of bits, which are thought of as the genes that make up an "organism." The goal of a GA is to use the principles of evolution to produce an organism (or equivalently, a configuration string) that is optimal.

The earliest examples of what might currently be called genetic algorithms introduced in the late 1950s, programmed on computers by evolutionary biologists. "*Evolution strategy"* was developed, in 1965, by Ingo Rechenberg. In this method, there was no population or crossover; one parent was mutated to produce one child, and the better of the two individuals was kept and became the parent for the next mutation (Haupt R., 1998). Later versions of evolution strategy introduced the concept of a population. Evolution strategies are still employed today by scientists and engineers, especially in Germany (Marczyk A., 2004). The next step in the field came in 1966, by introducing the "*evolutionary*

programming" by Fogel, and Owens in America. In this method, candidate solutions were represented as finite-state machines; same as Rechenberg's evolution strategy, their algorithm worked by arbitrarily mutating one of those simulated machines and keeping the better of the two (Mitchell M., 1996) (Goldberg D.E., 1989).

In 1962, John Holland's work on adaptive systems was the foundation for later developments; also he was the first person, who proposed crossover and other recombination operators. However, the influential work in the field of genetic algorithms was in 1975, with the publication of "*Adaptation in Natural and Artificial Systems"*. It was the first work, which systematically presents the idea of adaptive systems using selection, mutation and crossover, simulating processes of biological evolution, as a problem-solving strategy (Haupt R., 1998) (Mitchell M., 1996).

By the 1980s, genetic algorithms were being applied to a wide range of subjects, from abstract mathematical problems to engineering problems such as pipeline flow control and structural optimization (Goldberg D.E., 1989). Those applications were mainly theoretical. However, with the development of the Internet and, the exponential growth of computing power, genetic algorithms migrated to the commercial sector (Haupt R., 1998).

Today, evolutionary computation and genetic algorithms are "solving problems of everyday interest" (Haupt R., 1998) in areas of study like stock market prediction and portfolio planning, microchip design, aerospace engineering, biochemistry and molecular biology, and scheduling at airports and assembly lines (Marczyk A., 2004).

GA has been used in the modelling of the cooling of steel skelp to estimate model parameters (Peng L., 2008); however, in this work, a GA is used to find an optimal strategy for control variables (laminar cooling system configuration) to

obtain the specified temperature profile (cooling rate) and coiling temperature. The FEM thermal model used in this work, has no mathematical equation to use in quantitative optimization techniques; therefore, a stochastic optimization method like GA is a useful tools for this type of problem. In other word, there is not an exact way to calculate the best solution for this type of problem, but it is possible to estimate the solution by a heuristic optimization method. The other advantage of GA is that it works on a population of the solutions in comparison with other heuristic methods which has a single solution in their iterations. Finally, GA applies the rules of evolution to each individual. GA finds best individuals and combines them together to find better result and evolve good solutions.

The population types can be vectors or strings. In this work, chromosomes are bit-string; therefore, the method is called binary genetic algorithm. The fitness is defined according to the fitness function f (or cost function). The best individuals, according to individual's fitness $f(x_i^k)$ (for individual number i at generation number k), will build the "mating pool". The "fit" of each individual organism is computed by assessing the score produced by that individual (Goldberg D.E., 1989). The score used in this work is also the optimization objective function. A lower score is better than a higher score. The individuals are ranked from most fit to least fit. The fit of an individual affects the probability that some or all of the 15-bit string representing an individual will be used in subsequent simulations. This process of selection by fit is designed to mimic the process of natural evolution.

The first way by which individuals pass on their configuration to the next generation of simulations is by acting as a "parent." The parents for the next generation of individuals are chosen randomly, but with a probability that is proportional to the inverse of a score generated by the individual. For example, an individual with a score of 10 is twice as likely as an individual with a score of 20 to be chosen as a "parent." The bit string representing the "children" is computed

using a crossover algorithm (Mitchell M., 1996). In addition to the use of parents, the top-ranked individuals are chosen as "elites," which are automatically included in the next generation. Finally, the remaining individuals in the next generation are produced from the "mutation" of individuals in previous generations. These mutations are generated by randomly selecting bits in each individual to be altered. The selection process is applied using a uniform distribution, which means that each bit has an equal chance of being altered in the next generation. The GA algorithm is terminated when no improvement in the objective function is observed for ten generations (Holland J.H., 1975).

3.2.1.1 Genetic Algorithm Parameters

Genetic Algorithms as an evolutionary method have some parameters, which are indicated according to the process characteristics. These parameters include methods of scaling fitness scores, selecting parents, mating parents, selecting mutated individuals, and mutating between individuals.

3.2.1.1.1 Fitness scaling

The fitness scaling converts the raw scores of fitness function values to scaled scores. The following are some techniques for fitness scaling:

1. Rank: Based on the rank of raw fitness scores. For example, the best individual's fitness gets first place, the second individual with respect to fitness value gets second place, and so on.
2. Proportional: Assigns a value to each individual, proportional to its raw fitness score.
3. Top: Assigns a constant value called "quantity" to the fittest individuals and "0" value to the others.
4. Shift linear: Values for scaled fitness are equal to a constant multiplied by the average score.

Fitness scaling converts the raw cost values that are returned by the cost function to values in a range that is appropriate for the selection function. The selection function uses the scaled cost values to determine the parents of the following generation. The selection function assigns a higher probability of selection to individuals with higher scaled values.

The variety of the scaled values influences the performance of the genetic algorithm. If the scaled scores vary too widely, the individuals with the highest scaled values reselect too rapidly, take over the population gene pool too quickly, and prevent the genetic algorithm from penetrating other areas of the solution space. Otherwise, if the scaled values differ only a little, all individuals have similar chances of reproduction and the search will progress quite slowly.

The Rank fitness scaling scales the cost values placed on the rank of each individual instead of its raw cost. The rank of an individual is its position in the sorted values (in Figure 3-1): the rank of the most fit individual is one, the next-most-fit individual is two, and so on. The rank scaling function determines scaled values so that:

- The scaled value of an individual with rank *n* is determined by an equation which is inversely proportional to n.
- The sum of the scaled values over the entire generation equals the number of parents needed to produce the next generation

Rank fitness scaling eliminates the effect of the wide range of raw scores. Figure 3-1 presents a sample of the scaled values of the raw scores using rank scaling for a typical generation of size 20.

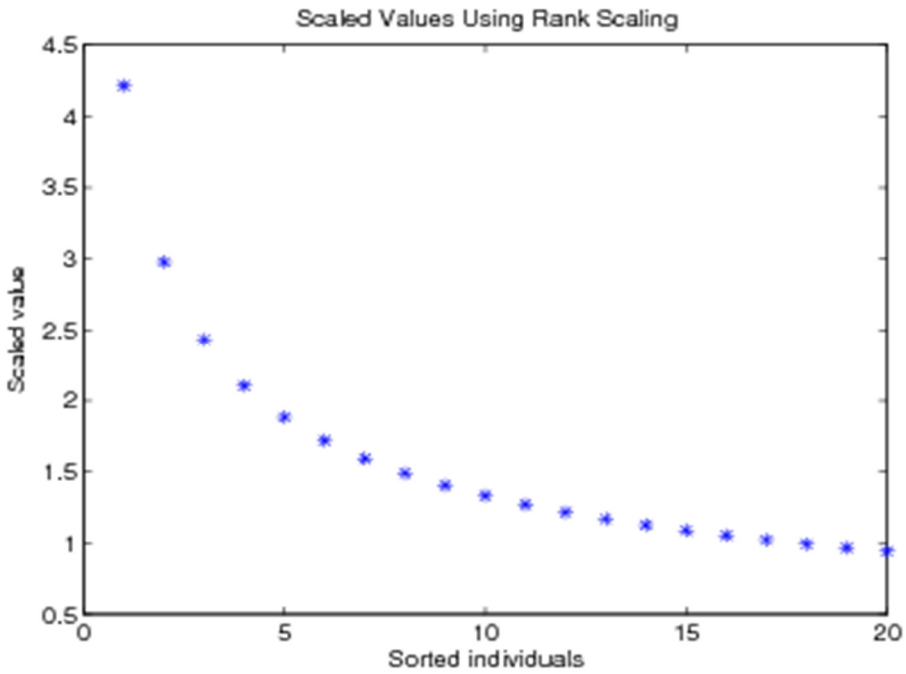

Figure 3-1. Scaled values for sorted individuals in one generation

Because the algorithm minimizes the cost function, lower raw fitness scores have higher scaled values. In addition, because rank scaling determines values that count on an individual's rank, not raw fitness scores, the scaled values shown would be the same for any generation of size 20.

3.2.1.1.2 Selection methods

Parents mate to make children for the next generation. The selection method indicates how parents will choose to participate in the mating pool. There are five common methods for selection:

1. Stochastic uniform: This selection method chooses parents from a line where each individual has part of the line equal to its scaled value. It moves along the line with equal steps and selects the individuals from the section of line it lands on.

2. Remainder: This incorporates the roulette method, in which each individual has a probability equal to the integer part of its scaled value. For example, if the scaled value of an individual is equal to 2.3, the probability of choosing it is equal to 2. The final probability is equal to:

$$Probability\ (p_i) = \frac{f(x_i^k)}{\sum_{i-1}^{20} f(x_i^k)} \qquad (3\text{-}1)$$

3. Uniform: This method uses a random search between all parents, which is not a reliable method for selection. It only works for a wide search in all areas of the problem.

4. Roulette: The basics of this method are according to the roulette wheel, whereby each individual has part of wheel equal to its scaled value. Individuals with higher scaled values have a higher chance of being chosen as parents. Roulette is the most common method for selection due to its randomness. With fitness-proportionate selection, there is a possibility that some weaker solutions may survive the selection process; this is advantageous, as although a solution might be weak, it may comprise some genes that could prove useful following the recombination process.

5. Tournament: The basis of this method is to select the best individual among groups of randomly selected individuals. The number of tournaments is equal to the number of parents, which are needed for the next generation (Goldberg D.E., 1989).

After fitness scaling, the selection method indicates how parents are chosen to participate in the mating pool. The selection method in this work is roulette.

Parents are selected according to their cost values. The better the chromosomes are, the more possibilities to be selected they have. Consider a roulette wheel where all the chromosomes in one generation are placed: each has its place sized accordingly to its scaled value, as on the following diagram. A “toss” is then

thrown and a chromosome is selected. The chromosome with a bigger scaled score will be selected more times.

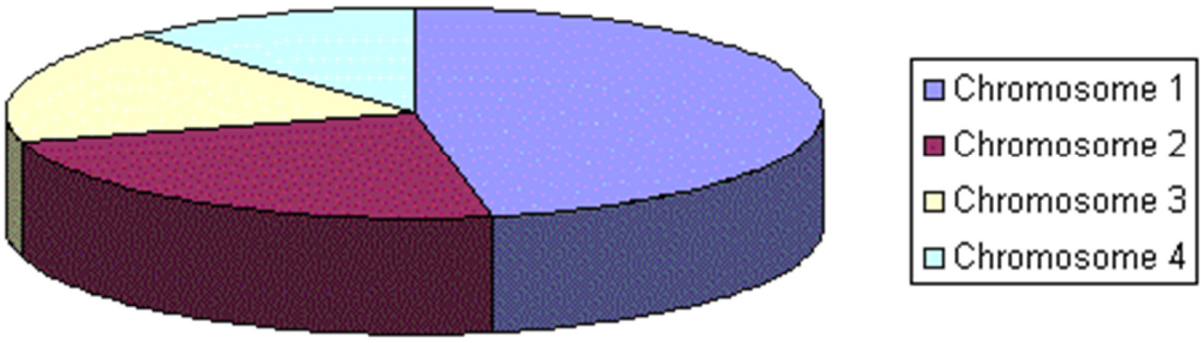

Figure 3-2. Roulette wheel proportional to chromosomal possibilities

This process can be presented by the following algorithm:

1. [Sum] Calculate the sum of all individual fitnesses in a generation - SUM *S*. The sum of scaled scores in this work is 33.96 for each generation, derived by using the "Rank" method for fitness scaling. Step 1 is performed only once for each generation.
2. [Select] Generate a random value from interval *(0, SUM)* - *r*.
3. [Loop] Go through the generation and sum fitnesses from *0* - sum *S*. When the sum *S* is greater than *r*, stop and return the individual where you are.

3.2.1.1.3 Reproduction

Reproduction is the second phase of the GA, after the analysis of the initial population. This phase includes mating (crossover) and mutation. The crossover fraction indicates the percentage of individuals in each generation that were generated by crossover (mating) from parents of the previous generation. The value for the crossover fraction in this work is 0.75, which means that 75% of the second generation's individuals (other than elites) are made by mating parents from the previous generation.

3.2.1.1.4 Crossover

Crossover is a representation of mating in the GA, and operates on the mating pool. It is the combination of two individuals (parents) to make a new child. However, it can be more than two parents for this purpose. There are different options for crossover during the genetic algorithm process.

1. Scattered: The genes are selected from one of two parents randomly. For example, consider that p_1 and p_2 are parents. Then we have:

$$P_1 = [a\ b\ c\ d\ e\ f\ g\ h] \quad (3\text{-}2)$$

$$P_2 = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8] \quad (3\text{-}3)$$

By using scattered crossover:

$$Child = [a\ b\ 3\ 4\ e\ 6\ 7\ 8] \quad (3\text{-}4)$$

2. Single point: One entry location is chosen randomly. The entries before that location are chosen from one parent, and after that location from the other parent. For example, the child from a single point crossover of p_1 and p_2, which single point location is randomly number 4 is:

$$Child = [a\ b\ c\ 4\ 5\ 6\ 7\ 8] \quad (3\text{-}5)$$

3. Two point: Like the single point method the points are chosen randomly. The child is made by a combination of parents. Similarly to the previous example, the child from two-point crossover of p_1 and p_2 is:

$$Child = [a\ b\ c\ 4\ 5\ 6\ g\ h] \quad (3\text{-}6)$$

4. Intermediate: The crossover process for each entry is:

$$Child = parent1 + rand * (parent2 - parent1) \quad (3\text{-}7)$$

Parent 1 is better fitted in comparison with parent 2. In binary GA, the numbers should be converted to 1 and 0.

5. Heuristic: Similar to the previous method, but only Ratio is multiplied by the difference of parents:

$$Child = parent1 + rand * Ratio * (parent2 - parent1) \quad (3\text{-}8)$$

By selecting the Ratio, it is possible to specify how close the child to the better parent. A lower Ratio leads to children closer to parent 1.

3.2.1.1.5 Mutation

A genetic algorithm makes random individuals by mutation in previous generations to search in a broader space. Mutation enables the GA to search in new parts of the solution's area, and is a convenient way to make sure that the final result is not a local minimum. Mutation changes the previous genes in the parent by small random changes. There are two main methods for mutation: Gaussian and Uniform.

1. Gaussian: This method adds a Gaussian value with mean equal to 0 to each gene in parent.
2. Uniform: In this method, the algorithm first selects a number of genes to perform the mutation on them. The selected genes are then replaced by a random number. Each entry has the probability rate of being mutated, which is almost about 0.01.

The mutation rate can be time-variant during the process. In fact, if the fitness function is constant along the process, there is no need to have a high mutation rate. If the fitness function is time-varying like online systems, it is better have a higher mutation rate to explore all areas of possible solutions to make sure that the method is not getting stuck in a local maxima or minima.

3.2.1.2 Stopping Criteria

Stopping criteria indicate the conditions to stop the optimization process. There are different ways to finish the job in a GA:

1. Generation numbers: It is possible to limit the GA by the number of generations. For example, the process will cease when 100 generations made.
2. Time limit: The search finishes after a defined time. For example, one day is the time limit to finish the process.
3. Fitness limit: When the fitness function is higher or lower than fitness limits the GA will stop.
4. Stall generations: If there is no improvement in best fitness function after a certain number of generations the process will stop.
5. Stall time: If there is no improvement in best fitness value after determined time the process will stop.

3.2.1.3 Algorithm of GA

A complete GA can be represented by the following steps:

1. Encode the input values.
2. Define the parameters, including population size, crossover and mutation types and probability and stopping criteria.
3. Define the initial population.
4. Find the fitness function for each individual.
5. Select the desired individuals for the mating pool.
6. Reproduce (mate and mutate) to make the next generation, according crossover fraction and mutation rate.
7. Repeat step 4 again until termination.
8. Determine the results.

3.2.2 Particle Swarm Optimization (PSO)

The first antecedents of Particle swarm optimization can be traced from work of Reeves (Reeves W. T., 1983), who introduced particle systems to model objects that are dynamic and cannot be represented by surfaces or polygons, like fire, water, and smoke. Later in 1987, Reynolds used a particle system to simulate the group behavior of a flock of birds. Also, social psychology research, in particular the dynamic theory of social impact (Nowak A., 1990), was another source of inspiration in the development of the particle swarm optimization algorithm (Kennedy J., 2006). Finally, PSO, as an evolutionary optimization method, was developed by James Kennedy and Russell Eberhart in 1995 (Hassan R., 2006). PSO was derived from the analysis of social behaviour of bird flocks. It is used primarily for optimization of nonlinear functions (Hassan R., 2006). Later, a discrete binary version for PSO was developed by Kennedy (Kennedy J., 1997). In this work, the discrete binary version of PSO was used as optimization algorithm.

3.2.2.1 Particle, Swarm, local and global best

In concept, the PSO method is similar to genetic algorithms, but it does not have crossover and mutation. The terms “individuals” and “population” in a GA are called “particle” and “swarm” in PSO. Unlike in a GA, there is no natural evolution in PSO to evolve the population to the next generation. In PSO, each particle’s position is adjusted in order to pursue that particle’s and swarm’s best previous positions.

Each particle aims to reach the global minimum using the swarm’s experiences. The best result for each particle is called “lbest” (local best), while the best result in all particles (swarm) is called “gbest” (global best). The speed of each particle is modified to reach both of these items. The “gbest” guarantees the search for the global minimum, while the “lbest” guarantees the search on local minimum

neighbourhood. It is similar to the social behaviour of humans, who use their own experiences in addition to experiences from society, to reach their goals. At each step, the speed of the particles toward the global best is determined by the differences between particle's positions plus "lbest" and "gbest."

In past years, PSO has shown that it is faster than other evolutionary methods. Another advantage of PSO is that it has fewer parameters (four), of which two are random and the other two are dependent on the problem. A genetic algorithm, on the other hand, has too many parameters like crossover fraction, mutation rate, and selection method.

A discrete binary PSO developed by Kennedy et al. in 1997 (Hassan R., 2006) represents problems that contain binary inputs. In this problem, the position of the particle should be assigned as "0" or "1." After finding new positions for each particle, the value should round to "0" or "1" for use in the process.

Each particle's position is defined as x_i^k, where i is the particle number and k is the swarm iteration number. Each particle of the swarm is a d-dimensional vector as follows:

$$x_i^k = \left[x_{i1}^k, x_{i2}^k, ..., x_{id}^k\right] \tag{3-9}$$

An n-dimensional population is defined as:

$$Pop^k = \left[x_1^k, x_2^k, x_3^k, ..., x_n^k\right] \tag{3-10}$$

The most common value for n is two multiplied by the d value.

For adjusting the particle's position, the particle's velocity is defined as:

$$v_i^k = \left[v_{i1}^k, v_{i2}^k, ..., v_{id}^k\right] \tag{3-11}$$

The velocity modifies the particle's position and affects each entry in the particle.

The particle best is the best value for each particle from previous experiments:

$$PB_i^k = \left[pb_{i1}^k, pb_{i2}^k, ..., pb_{id}^k\right] \tag{3-12}$$

The best result during the evaluation of all individuals in population is the global best, which can be expressed as:

$$GB_i^k = \left[gb_1^k, gb_2^k, ..., gb_d^k\right] \tag{3-13}$$

As in a GA, PSO has termination criteria that complete the optimization process. Criteria such stall generation, stall time, generation limit and time limit are similar to those found in a GA.

3.2.2.2 Initialization

The initial population is found randomly. The probability of values "0" and "1" for one particle is 0.5:

$$\begin{aligned} &\text{If: } U(0,1) > 0.5 \text{ Then: } x_i^d = 1 \\ &\text{Else: } x_i^d = 0 \end{aligned} \tag{3-14}$$

There is a limit for the speed of particles; the common setting is: $V_{\min} = -V_{\max}$. The range of speed is considered as $V_i^k = \left[V_{\min}, V_{\max}\right] = \left[-4, 4\right]$.

The velocity of particle i in d^{th} dimension is:

$$v_{id}^0 = V_{\min} + (V_{\max} - V_{\min}) * rand() \tag{3-15}$$

The fitness function for particle i in generation k is shown as: $f(x_i^k)$

As in the GA, the fitness function represents the result's proximity to the goal while considering the constraints.

3.2.2.2.1. Generating a new population (updating particles' speed and position)

For discrete binary PSO, two functions are required to modify each particle to be 0 or 1. Piece-wise function keeps the particle's speed between $V_{\min}$ and $V_{\max}$.
The piece-wise function is illustrated as:

$$h(v_{id}^{k}) = \begin{cases} V_{\max}, \text{ if } v_{id}^{k} > V_{\max} \\ v_{id}^{k}, \text{ if } |v_{id}^{k}| > V_{\max} \\ V_{\min}, \text{ if } v_{id}^{k} < V_{\min} \end{cases} \quad (3\text{-}16)$$

After piece-wise function, the sigmoid function is used to modify the velocities between 0 and 1:

$$Sigmoid(v_{id}^{k}) = \frac{1}{1+e^{-v_{id}^{k}}} \quad (3\text{-}17)$$

The particle's speed is:

$$\Delta v_{id}^{k-1} = c_1 r_1 (pb_{id}^{k-1} - x_{id}^{k-1}) + c_2 r_2 (gb_{d}^{k-1} - x_{id}^{k-1}) \quad (3\text{-}18)$$

The c_1 and c_2 are parameters, which are related to swarm's characteristics. The r_1 and r_2 are random numbers between 0 and 1. The updated velocity is:

$$v_{id}^{k} = h(v_{id}^{k-1} + \Delta v_{id}^{k}) \quad (3\text{-}19)$$

The velocity of particle *i* in generation number *k* is:

$$\Delta v_{id}^{k-1} = w v_{id}^{k} + c_1 r_1 (pb_{id}^{k-1} - x_{id}^{k-1}) + c_2 r_2 (gb_{d}^{k-1} - x_{id}^{k-1}) \quad (3\text{-}20)$$

Where *w* is the inertia factor, which is between 0.4 and 1.4; c_1 is the self-

confidence factor, which is between 1.5 and 2; and c_2 is swarm confidence factor, which is between 2 to 2.5.

Finally, position is updated according to particle's velocity:

$$x_i^{k+1} = x_i^k + v_i^{k+1}\Delta t \tag{3-21}$$

The d^{th} dimension of particle i is updated as:

$$\begin{gathered} \text{If } U(0,1) < sigmoid(v_{id}^k) \text{ then } x_i^d = 1 \\ \text{Else } x_i^d = 0 \end{gathered} \tag{3-22}$$

Update particle best as follow:

$$\begin{gathered} \text{If } f_i^k(x_i^k, i = 1,2,...,n) < f_i^{pbest}(PB_i^{k-1}, i = 1,2,..,n) \\ \text{Then} \\ f_i^{pbest}(x_i^k, i = 1,2,...,n) = f_i^k(PB_i^{k-1}, i = 1,2,..,n) \\ \text{Else} \\ f_i^{pbest}(x_i^k, i = 1,2,...,n) = f_i^{pbest}(PB_i^{k-1}, i = 1,2,..,n) \end{gathered} \tag{3-23}$$

Update global best:

$$\begin{gathered} f^{gbest}(GB^k) = \min\{ f_i^{lbest}(PB_i^{k-1}, i = 1,2,...,n)\} \\ \text{if } f^{gbest}(GB^k) < f^{gbest}(GB^{k-1}) \\ \text{Then} \\ f^{gbest}(GB^k) = f^k(GB^k) \\ \text{Else} \\ f^{gbest}(GB^k) = f^k(GB^{k-1}) \end{gathered} \tag{3-24}$$

3.2.2.3 Stopping criteria

Like GA, the stopping criteria can be defined as stall generation, simulation limit, time limit, etc. If the stopping criterion for the optimization is defined as "stall generation = 10", it means that if there were no improvements in GB after 10 generations, the process would be stopped.

3.2.2.4 PSO algorithm

A PSO algorithm can be represented as:

1. Encode the data.
2. Set the parameters for PSO like generation size and speed coefficients.
3. Define the initial population.
4. Evaluate the fitness function for each individual.
5. Set each particle as the "lbest" and the fittest particle as the "gbest" for the initial population.
6. Determine the speed for each particle according to the distance of the particle's position from "lbest" and "gbest" and the coefficients of the velocity equation.
7. Generate the next population using the speed and position of the previous particles.
8. Evaluate the fitness function for each individual.
9. If the fitness of each particle is better than "lbest," set that one as "lbest" for that particle.
10. If the best fitness in population is better than "gbest," set it as "gbest."
11. The process will stop if the best cost value reaches the stopping criteria. If the cost values are not acceptable according to the stopping criteria the optimization will continue from step 6.

3.2.3 Branch-and-bound

Branch and bound is a systematic method for solving various optimization problems, especially discrete and combinatorial optimization. Eastman (1958) probably developed the first branch-and-bound algorithm, however Land and Doig (1960) proposed branch-and-bound as a generic algorithm for discrete programming in 1960 (Land A. H., 1960).

Unlike evolutionary optimizations, the branch-and-bound (BB) as a deterministic optimization method is a search method that can definitively find the best solution for the problem. In this technique, the feasible set is relaxed and subsequently split into parts (branching) over which lower (and often also upper) bounds of the objective function values can be determined (bounding). BB has been successfully applied to certain problems such as minimizing a concave function over a convex set (Horst R., 1993).

In Figure 3-3 the variation of all control actions are shown. Figure 3-3 is an overall view of all possibilities of run-out table configurations, which looks like a tree with the origin (or root node) on the left, and the branches (or leaf nodes) on the right. The branches represent the definite enumerated solutions, so there are 3^6 possibilities for all configurations of banks and side-sprays for each velocity. For example, the node at the upper right represents the solution in which all banks and side-sprays are "off." It is equal to "0000000000000" in evolutionary optimizations. The other leaf nodes, on the right, can be thought of as representing sets of possible configurations. Intermediate nodes (e.g. the second node to the right of the root node) represent subsets of all of the possible solutions. For any two directly connected nodes in the structure, the parent node is the one closer to the root, and the child is the one closer to the leaves (Chinneck J. W., 2010).

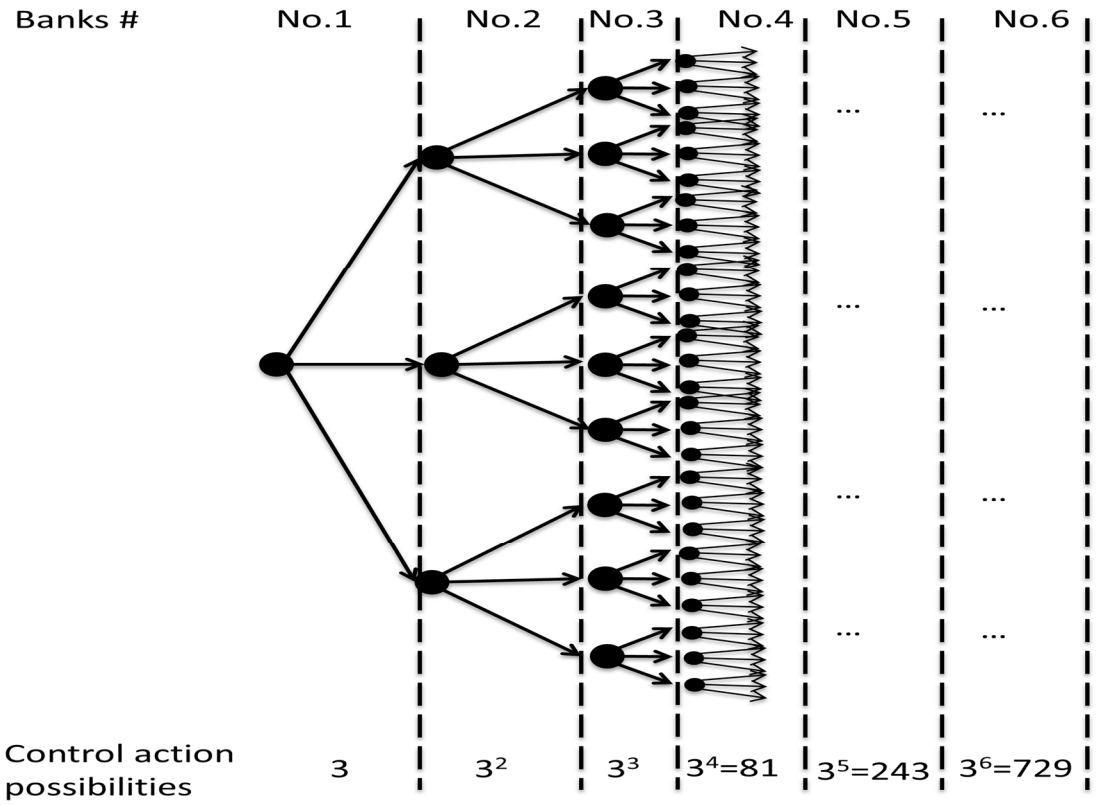

Figure 3-3. The scheme of all possible routes

As illustrated in Figure 3-3, the steel cooling optimization is a tree that shows the possibilities for control actions from the first bank to the end of the sixth bank. The tree grows from one bank to the next. Each control action in one bank has three control actions in the next bank; therefore, there are 3^6 possibilities for the entire cooling section for each velocity.

The main goal in branch-and-bound is to avoid growing the whole tree as much as possible, because the whole tree is just too large for whole optimization problem. Instead, branch-and-bound grows the tree in steps and grows only the most capable nodes at any step. It determines which node is the most talented by estimating a bound on the best value of the cost function that can be attained by growing that node to next steps. The name of the method comes from the branching that happens when a node is chosen for further growth and the next

stage after that node is created. The bounding appears when the bound on the best value achieved by growing a node is estimated.

Another significant aspect of the method is pruning, in which the node or any its descendants that will never be optimal are cut off and permanently discarded. Pruning is one of the most critical facets of branch-and-bound, since it is exactly what prevents the search tree from rising too much (Chinneck J. W., 2010).

Figure 3-4 is an example of accepted and rejected control actions. By rejecting a control action at each bank, the descendants from that control action will be cancelled, thereby reducing the number of simulations. (For example, by cancelling one control action in the first bank, 1/3 of all possibilities are eliminated and only 2/3 remain).

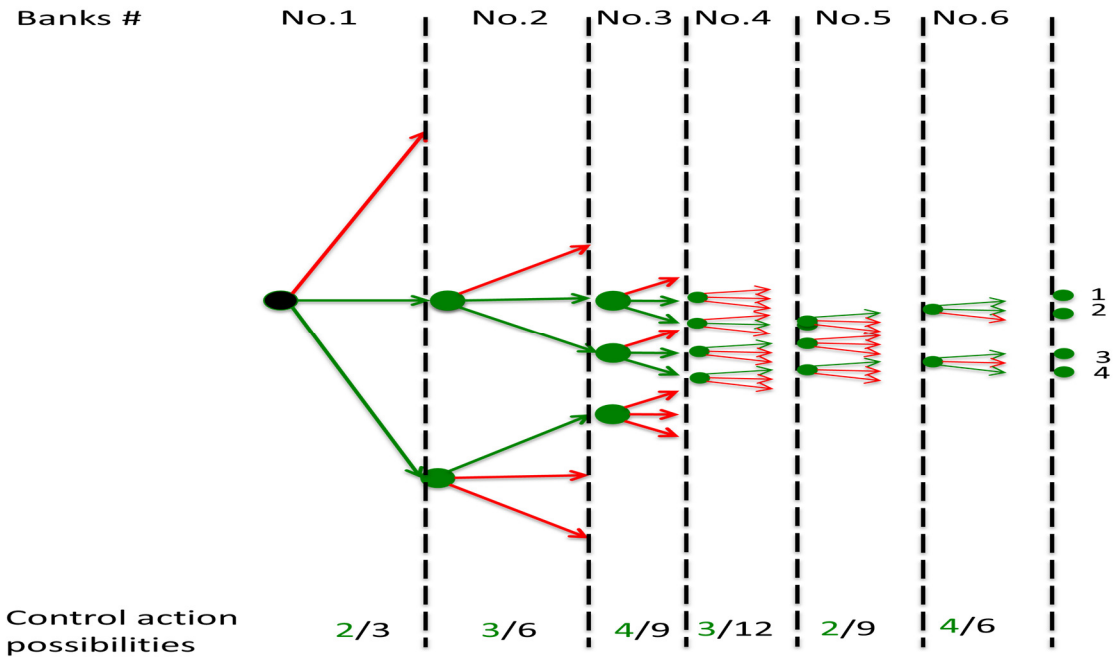

Figure 3-4. Acceptable routes during the search process for a constant velocity

One advantage of BB is that during the iteration process, one can usually delete certain subsets of total possibilities, since the optimal solution cannot be attained

there. One typical disadvantage is that, as a rule, the accuracy of an approximate solution can only be measured by the difference between the upper and lower limits of the current bounds. Hence a "good" feasible point found early in the iteration may be deleted as "good" only much later after many further refinements (Horst R., 1993).

Branch-and-bound has several distinct characteristics:

- The problem can be divided into stages: by breaking the problem down into smaller parts, it is possible to find an optimum way to solve the smallest part. In doing so, it is possible to find the optimum solution for the entire problem.
- Each step has a number of states: it will indicate the possibilities for each small part.
- The decision at one step updates the state at this step to a new state at the next step.
- The optimal decisions for future steps are independent of decisions made in previous steps (the principle of optimality).

3.2.3.1 Branch-and-bound terminology

Node: Any partial of a complete solution. For example, 1-2 -2-3-?-?, in which the first variable has value of 1, the second variable has a value of 2, the third has a value of 2 and the fourth one has a value of 3. The values of the last two variables are not yet set.

Leaf (leaf node): A complete configuration in which all of the variable nodes (Values) are known.

Bud (intermediate node): A partial solution, either acceptable or unacceptable.

Bounding function: The method to estimate the best value of the cost function obtainable by growing an intermediate node further. Only bud nodes have associated bounding function values. Leaf nodes have cost function values, which are actual values and not estimates.

Growing a node: the process of generating child nodes for an intermediate node. One child node is created for each possible value for the next variable. For instance, if the next variable is the conditions of banks and side-sprays, there will be one child associated with the value "on-on," one child node associated with the value "on-off," and on child node associated with the value "off-off."

Incumbent: the most complete, acceptable solution found so far. There may not be an incumbent when the solution process begins. In that case, the first complete, acceptable solution found during the growing process becomes the first incumbent.

3.2.3.2 Bounds

In order to reduce the number of evaluations in each optimization, bounds are defined for objective functions. Upper and lower bounds during each stage authenticate or reject control actions to continue to the next stage. The limitations are different for cooling rate and coiling temperature optimizations.

In order to have minimum error between simulated cooling rate and target cooling rate during the cooling process, temperature bounds were applied on the temperature profiles. These constraints controlled the temperature variation from the desired temperature profile. It means that the temperature profile cannot go out of the determined space at any point. The temperature profile and its confidence interval are shown in Figure 3-5.

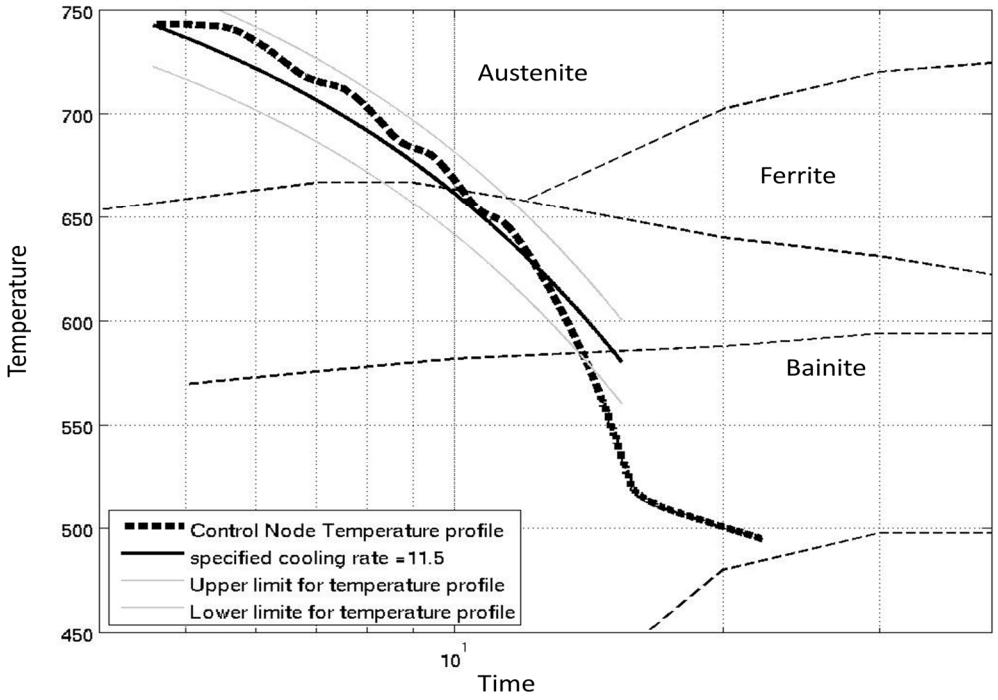

Figure 3-5. Temperature profile of control Node in comparison with specified cooling rate

3.2.3.3 Stopping criteria

The stopping criteria are dependent to objectives of optimizations. If the cost value of an individual is in the acceptable range according to the stopping criterion for that optimization, the process will be stopped. For some objectives, there is no stopping criterion and the optimization process continues to search the entire solution space (e.g. cooling rate optimizations).

3.2.3.4 Algorithm of Branch and bound

According to banks and side-sprays configuration the problem divided to six stages (Figure 3-3). In each stage, the cost of temperature profile in the area of a related bank is evaluated. Top-down BB algorithm for the steel cooling process is:

1. Encode the inputs from configuration conditions.
2. Define the lower and upper bounds of the problem in order to reject improper control actions.
3. The process starts with the first bank and evaluates the cost of each control action.
4. The control actions, which are acceptable according to objective for that optimization, have permission to continue in the second bank. For example, if two of three control actions are allowed to continue, the number of simulations for the second bank will be six instead of nine evaluations (Figure 3-5).
5. Step 2 continues for the next banks, and the feasible answers are permitted to continue to the next bank.
6. At the end of sixth bank, the survived solutions are ranked according to their fitness values for the first velocity, which is the maximum possible velocity.
7. The best fitness value will be the minimum requirement for the next velocity costs (i.e. fitness values should be lower than the best cost of previous velocity to survive). If there is no fitness value, which is better than the best one from the previous velocity, the search for that velocity will be stopped. The computation continues for the next higher velocity.
8. The recursive computation (steps 2 to 5) is repeated for the next velocity. The best fitness value of next velocity will substitute to the previous best solution.

References

(Avilla V.H., 1994) Avilla V.H. *"Modeling of the thermal evolution of steel strip cooled in the hot rolling run out table"*, PhD thesis, university of British Columbia. 1994.

(Chinneck J. W., 2010) Chinneck J. W. *"Practical optimization : a Gentle Introduction"*. 2010.

(Goldberg D.E., 1989) Goldberg D.E. *"Genetic Algorithms in Search, Optimization and Machine Learning "*, Boston, MA. : Kluwer Academic Publishers, 1989.

(Hassan R., 2006) Hassan R. Cohanim B., de Weck O., *"A Comparison of particle swarm optimization and genetic Algorithms"*, - Colorado Springs : Vanderplaats Research and Development, Inc., 2006.

(Holland J.H., 1975) Holland J.H. *"Adaption in Natural and Artificial Systems"*, Ann Arbor, MI, 1975.

(Horst R., 1996) Horst R. Tuy H., *"Global Optimization: Deterministic Approaches"*, Springer, 1996.

(Land A. H., 1960) Land A. H. Doig A. G. *"An automatic method of solving discrete programming problems"*, Econometrica, Vol. 28(3), July 1960. pp. 497-520.

(Mitchell M., 1996) Mitchell M. *"An Introduction to Genetic Algorithms"*, Cambridge, MA : MIT Press, 1996.

(Peng L., 2008) Peng L. Li Q., Zhou Z. *"Cooling hot rolling steel strip using combined tactics"*, Journal of University of Science and Technology Beijing. No 3 : Vol. 15, June 2008. p. 362.

(Yaniro K., 1991) Yaniro K. Yamasaki M., Furukawa M., et al., *"Development of coiling temperature control system on hot strip mill"*, Technical report. Vol. 24, 1991.

(Younis A., 2010) Younis A. Dong Z. *"Trends, Features, and tests of common and recently introduced global optimization methods"*, Engineering Optimization. No 2 : Vol. 48, August 2010. pp. 691-718.

4

4) Skelp temperature profile optimization during laminar cooling using Genetic Algorithm[1]

Introduction

A common processing technique for manufacturing steel strip and skelp with acceptable properties and a relatively low cost is by Thermo Mechanical Control Processing (TMCP). A schematic of the TMCP process is shown in Figure 4-1. Included in this figure are a reheat furnace, a hot rolling system and the laminar cooling system for reducing the hot rolled steel temperature from the austenite temperature to a low temperature. The mechanical properties of coiled steel are highly dependent on the phase(s) formed during the cooling process.

[1] This Chapter is reprint of: Binesh B. Wiskel J.B., Ben-Zvi A. Henein H., *"Skelp temperature profile control during laminar cooling using Genetic Algorithms"*, 2011.

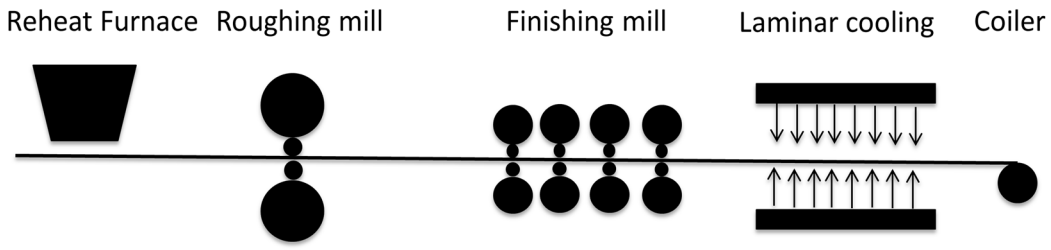

Figure 4-1. Schematic of TMCP process

The different phases of steel that can form during laminar cooling are depicted on the schematic Continuous Cooling Transformation (CCT) curve shown in Figure 4-2. A fast cooling rate and low coiling temperature will result in a microstructure primarily composed of bainite while a significantly slower cooling rate (and higher coiling temperature) will produce a microstructure with a dominant ferrite-pearlite microstructure. In most industrial systems the microstructure of the steel strip is primarily controlled by specifying a coiling temperature. This target coiling temperature is typically achieved via the manipulation of the strip velocity and laminar system configuration. However, as shown in Figure 4-2, the final microstructure of the strip is dependent on the time-temperature profile of the steel as it cools. Therefore, the ability to control both the cooling rate and the coiling temperature (hence forth referred to as cooling profile) is deemed important in controlling the final microstructure and ultimately the mechanical properties of the steel.

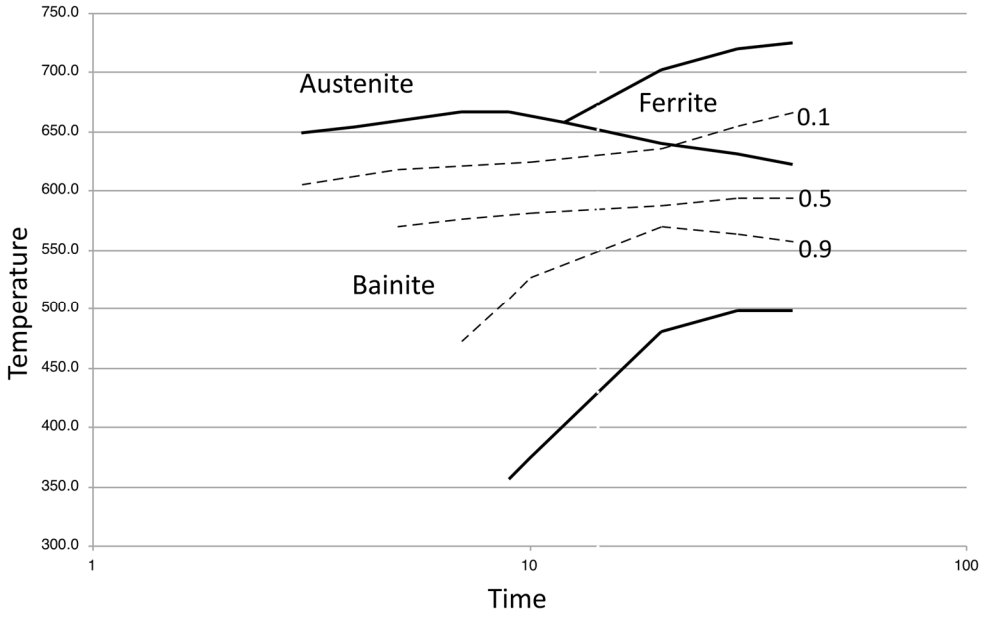

Figure 4-2. Sample CCT diagram for a micro-alloyed steel

In this work a methodology for obtaining the optimum desired cooling profile of a steel strip is presented. The method relies on a Finite Element (FEM) thermal model to predict the cooling profile of the steel strip as it travels though the laminar cooling system. In conjunction with the thermal model a Genetic Algorithm-based method is used to generate the optimal configuration of the laminar cooling system necessary to achieve the desired cooling profile.

4.1 Background

In this section, a review of the laminar cooling system and of control strategies necessary for the implementation of the proposed methodology will be undertaken. In addition, the finite element model used to provide temperature data to the optimization system will be discussed.

4.1.1 Laminar Cooling

A schematic of the laminar cooling system simulated (or modelled) in this work is shown in Figure 4-3 and includes the approximate location of the cooling banks and side-sprays used. The system shown in Figure 4-3 has three cooling zones: 1) The radiation zone before the first bank; 2) the water-cooling section; and 3) the radiation zone at the end between the last bank and the coiler. The heat transfer in the water-cooling section (Section Two) is the main method by which the temperature of the skelp is controlled. There are three main types of water-cooling systems that are used in the steel processing industry; these include water sprays, water curtain, and laminar flow (Xu F., 2006). In this paper, only the latter will be discussed. The laminar flow cooling system used in this work includes six water banks and side-sprays located after each of the water banks. The side-sprays remove water from the surface of the skelp and will reduce the amount of heat removed from the skelp. Manipulated values in the model are the speed of the skelp, conditions of both the water banks (either on or off) and the side-sprays (either on or off).

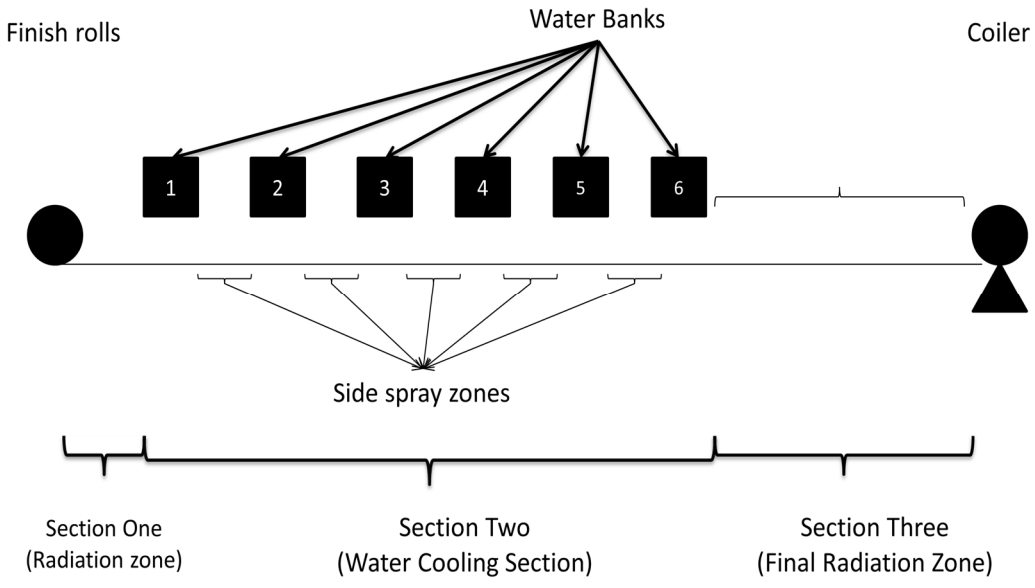

Figure 4-3. Schematic of a laminar cooling system

4.1.2 Laminar cooling optimization

Before 1960, little research had been done on control of the steel skelp temperature in run-out tables (Samaras N.S., 2001). Typically, the steel coiling temperature was controlled by selecting the velocity of the skelp and water bank configuration using heuristics (i.e., operator experience). Xie et al. (Xie H.B., 2006) used a reference feedback and feed-forward system to determine the coiling temperature in a laminar cooling system. The strategy proposed by Xie et al., relied on the development of a process model by parameter regression using existing data. The process model was then used to select set-points for the cooling water flow rate. The strategy and parameter regression method could control the coiling temperature with a high degree of precision (±15°C).

A control scheme based on operational experience for the automation of cooling system was proposed by Latzel (Latzel S., 2001). Latzel's method was based on modelling the skelp of steel as a geometrically distributed system. To estimate the temperature profile in the cooling section, a state observer was used. Samaras et al. (Samaras N.S., 2001), proposed a method for controlling the coiling temperature by manipulating the flow rate (on/off) of the water headers. In their work, Samaras et al. assumed equal distance between headers, identical temperature reduction in each header and negligible heat transfer due to radiation. The optimal set of activated headers was obtained using dynamic programming.

While considerable attention has been paid to control of the skelp temperature at a particular point along the laminar cooling system (e.g., at the coiler), little attention has been paid to controlling the temperature profile of the steel along the entire length of the run-out table. Control of the entire cooling profile is advantageous because the cooling profile dictates the steel microstructure (as shown in the CCT curve plotted in Figure 4-2). The steel microstructure, in turn, determines the final mechanical properties of the steel. Biswas et al (Biswas S.J.,

1997) proposed a control scheme for tracking of a desired temperature-time profile. This approach used a set of differential equations to approximate the heat transfer in the cooling system. The Gamkrelidze minimum principle was applied to the differential equations in order to compute the optimal water jet velocity (i.e. heat transfer) required for temperature tracking. However, this approach did not incorporate the use of side-sprays, variations in skelp velocity or the ability to turn the waters banks on and off.

4.1.3 Finite Element Thermal Model (FEM)

In this work a Finite Element (FEM) model of a laminar cooling system is used. This model was developed and validated by Wiskel et al. (Wiskel J.B., 2011). Though a full description of the model is provided elsewhere (Wiskel J.B., 2011), a brief overview of the key components of the model is presented below. A schematic of a portion of the laminar cooling system (transverse to the direction of the skelp velocity) is shown in Fig. 4-3 and includes regions of residual water retained on the skelp where water boiling occurs. Also included in Fig. 4-4 is the region encompassed by the two-dimensional (2D) finite element thermal model used in this optimization study (dashed box). Assumptions made in developing a representative 2D model include; negligible longitudinal heat flow (i.e. a transverse slice of the skelp is representative of the heat flow conditions), heat flow is symmetrical about the centerline of the skelp), the cooling associated with each jet/nozzle was identical and independent of other nozzles and radiation heat transfer was assumed in regions of the run-out table where the skelp was not exposed to water.

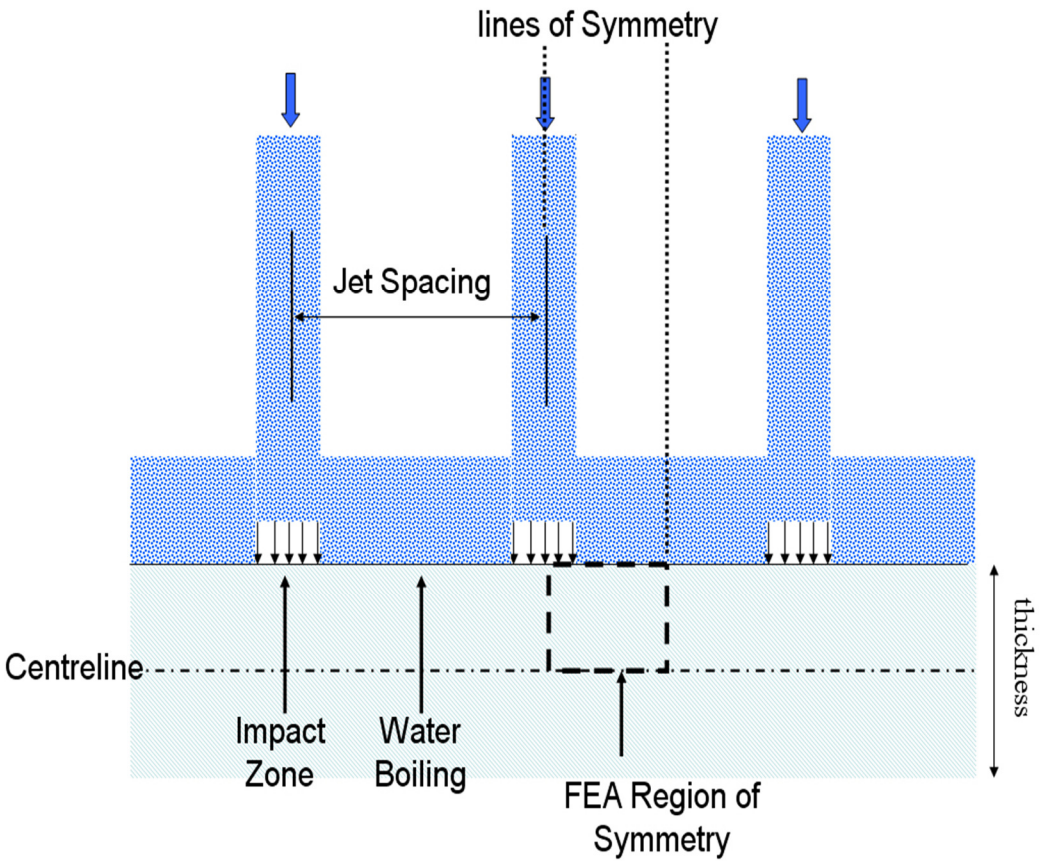

Figure 4-4. Schematic of a portion of a laminar cooling system transverse to skelp motion

Figure 4-5 is a schematic of the finite element mesh used for the 2D model and includes the location of the direct water impact cooling zone, the water boiling regime and the location of the nodes used for temperature control. The heat transfer coefficient for the impact cooling zone was calibrated from infrared imaging taken of an industrial run-out table and was determined to have a value of 10.5 MW m^{-2}. The water boiling regime heat flow was calculated using a standard water boiling curve adapted to the specific laminar cooling system studied. At the relatively high skelp temperatures (as encountered in this work), film boiling heat transfer dominates in this regime. A temperature dependent heat transfer coefficient on the order of 2000 W/m^2K was used to represent the magnitude of film boiling heat transfer.

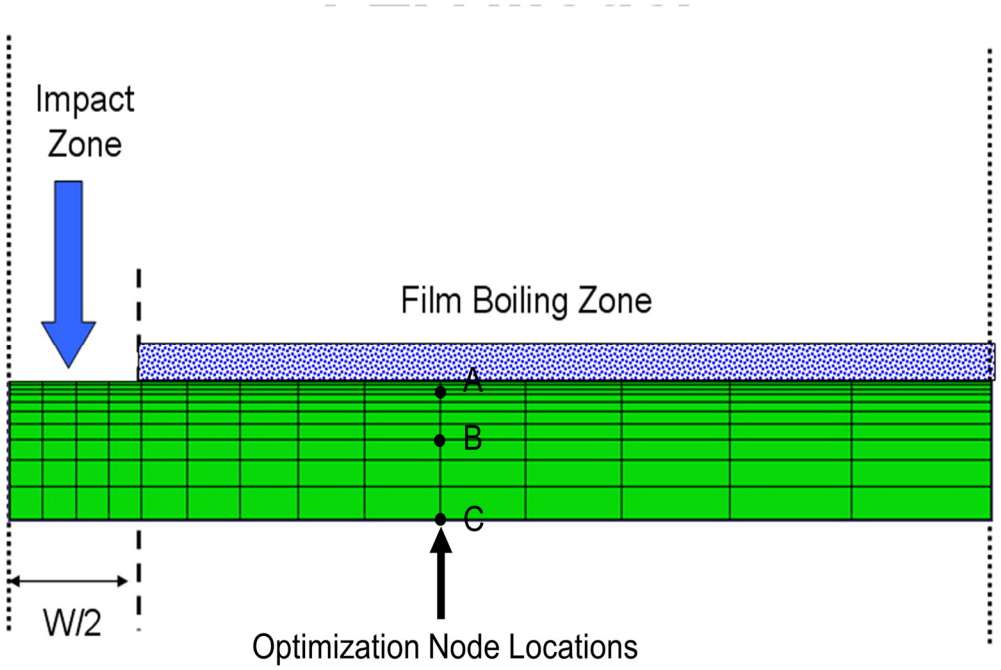

Figure 4-5. Finite element mesh showing impact zone and water boiling boundary conditions occuring at each header. Dotted vertical lines represent heat flux symmetry boundaries

For optimization purposes, the following temperature locations were used: Node A located near (at about 1/8 the depth of the skelp) the water cooled surface, Node

B at ¼ the depth below the surface and Node C is at the skelp centre line. For the optimization procedure proposed in this work, Node B was used to represent the temperature of the steel. It should be noted however, that Node B is an arbitrary choice and that the proposed procedure could be applied to at any location (or using an aggregate temperature representation). However, temperature optimization locations near the direct impact water cooling zone should be used with caution as these nodes would experience wide fluctuations in temperature (as the skelp passes under each laminar cooling header) thus reducing the control system stability. As a case in point, Nodes A and C were used to determine the sensitivity of the results to the choice of measurement point and will be discussed later.

4.1.4 Optimization Algorithms

In order to compute an optimal configuration for the system, an optimization procedure is required. In this paper a Generic Algorithm (GA) method (Goldberg, 1989) was used to determine the optimal laminar cooling configuration for achieving both the desired temperature profile and coiling temperature. GA is a family of popular global optimization procedures, which can be applied to systems where derivative information is not always available. Also, GA algorithms are suitable for obtaining nearly optimal solutions when the set of possible configurations is very large. For the laminar cooling system the set of possible configurations includes the condition of the headers and side-sprays (*on* or *off*), as well as the speed of the skelp. An *on* is represented by a 1 and *off* by a 0. For the cooling system considered in this work the number of water banks and side-sprays were chosen to be six. Each bank and side-spray can be set to either *off* (a value of zero) or *on* (a value of one). The skelp velocity can be digitized using P binary digits by dividing the range of allowable speeds into 2^P equally spaced values. In this work, the value of P was chosen to be *three (000 to 111)*, which provides eight discrete speed values (see Table 4-1). This value was chosen because the speed range used in this work (3 to 4.75 m/s) was sufficiently small

so that results could be obtained to within 0.25 m/s. The values of 3 and 4.75 m/s were arbitrarily chosen as the minimum and maximum allowable strip speeds, respectively, for a hypothetical process. This range of speeds could be increased or decreased by adding additional bits to encode the skelp velocity. In addition, the increments of skelp speed, shown in Table 4-1, could also be made finer or coarser. The total number of possible configurations for this work is therefore $2^6 \times 2^6 \times 2^3 = 32{,}768$ configurations. Simulating one configuration requires about five minutes of CPU time. To simulate all possible configurations would therefore require 114 days of CPU time. Even if a sufficient number of CPUs were available, this approach is inefficient because if the FEM model was modified in any way the whole set of simulations would have to be repeated. GA provides a computationally efficient approach for searching the configuration space while having to simulate only a small number of the total possible configurations.

Table 4-1. Bit Codes Used for Strip Speed

Three Bit Code	Strip Speed (m/s)
000	3.00
001	3.25
010	3.50
011	3.75
100	4.00
101	4.25
110	4.50
111	4.75

4.1.4.1 Genetic Algorithm

Genetic Algorithms (GA) are a family of numerical optimization methods (Goldberg, 1989). In GA, each possible configuration is expressed as a string of bits which are thought of as the genes *(0's or 1's)* that make up an "organism" (i.e. the configuration of sprays and speed for the run-out table in our case). The goal of GA is to use the principles of evolution to produce an organism (or, equivalently a configuration string) that yields an optimal result (i.e. skelp speed for a desired temperature profile and skelp temperature). To this end, parents' organisms "mate" to make offspring organisms with similar characteristics to themselves. In addition to mating and reproduction (which implies the crossover of genes) organisms may also experience mutation. GA has previously been used in the modelling of the cooling of steel skelp to estimate model parameters (Peng L., 2008); however, in this work GA is used to find an optimal laminar cooling system configuration to obtain the specified temperature profile (e.g., a constant cooling rate) and coiling temperature.

The optimization starts with a population of 20 organisms, chosen from a uniform distribution, where each individual is a 15-bit string (for example. 1010…01) that represents a specific laminar cooling configuration. Table 4-2 shows a sample initial population used for the algorithm. More precisely, each organism is defined so that the first six bits corresponds to the bank conditions, with one and zero corresponding to the bank being "*on*" or "*off*", respectively. The next six bits correspond to the side-spray condition, with one and zero corresponding to the side-spray being "*on*" or "*off*", respectively. The last three bits encode the speed of the skelp. Table 4-1 lists the three-bit coded speed values as well as their corresponding strip speed values.

Table 4-2. Sample initial population representing the run-out table configuration

Organism No.	Cooling Banks configuration	Side-sprays configuration	Strip velocity (see Table 4-1)
1	1 1 0 0 1 1	0 0 0 0 0 1	0 1 0
2	0 1 1 0 0 0	0 0 0 0 0 0	0 1 1
3	1 1 1 1 0 1	1 0 0 0 0 1	1 0 0
4	1 1 1 1 0 0	0 1 0 1 0 0	0 1 1
5	0 1 0 1 1 1	0 0 0 1 1 1	0 1 1
6	1 1 1 1 1 1	0 1 1 1 0 1	1 0 0
7	1 1 1 1 1 0	0 0 0 0 1 0	1 1 0
8	0 1 0 1 0 0	0 0 0 0 0 0	1 1 1
9	1 1 1 1 1 1	0 1 0 1 0 1	1 1 1
10	1 1 1 1 0 0	1 1 0 1 0 0	0 0 1
11	0 1 1 0 1 0	0 1 0 0 0 0	0 1 0
12	1 0 1 0 1 1	0 0 1 0 0 1	1 1 0
13	0 0 1 1 0 1	0 0 0 0 0 1	1 0 1
14	1 1 0 1 1 1	0 0 0 0 0 1	0 0 0
15	1 0 1 1 1 0	0 0 1 0 0 0	1 1 0
16	1 1 0 0 1 1	0 0 0 0 0 1	1 1 1
17	1 1 0 1 1 1	1 1 0 1 1 1	1 1 0
18	1 1 0 1 0 0	0 1 0 1 0 0	1 1 0
19	1 1 1 1 0 1	1 0 1 1 0 1	1 0 0
20	0 0 0 1 1 1	0 0 0 0 0 1	1 1 0

The outcome of each individual organism is computed by assessing the cost variable produced by that individual (Goldberg, 1989). The cost variable used in this work is also the optimization objective function, J, given by:

$$J = A\left|T_f - T_f^s\right| - B(Speed) + C\sum_{i=1}^{i=n^*}\left|NT(i) - NT(i)^s\right| \quad (4\text{-}1)$$

Table 4-3. Cost function parameters

Parameter	Definition/values
A	Coefficient used to penalize deviation from the desired coiling temperature.
B	Coefficient used to penalize slow speed values.
C	Coefficient used to penalize deviation from the temperature profile (e.g., cooling rate).
Speed	Velocity of the skelp
T_f^d	Specified coiling temperature.
T_f	Coiling temperature.
NT(i)	Temperature at node *i.*
NT(i)^s	Specified temperature at node *i.*
n^*	The number of nodes in the water cooling section.

where the equation parameters are defined in Table 4-3. The first term in Equation (4-1) is used to penalize the deviation (or difference) between the coiling temperature that is obtained (T_f) and the desired coiling temperature (T^s_f). The second term in Equation (4-1) is used to penalize slow operation of the run-out skelp because slow skelp velocity implies loss of production. The final term in Equation (1) is used to penalize deviation between the desired temperature at ith node (NT(i)^s) and the temperature at the ith node (NT(i)) that is obtained under a proposed cooling system configuration.

Table 4-4. Cost values, rankings and Scaled scores for the Sample Organisms configured in Table 4-2

Organism No.	Cost variable (*J*)	Ranking *(n)*	Scaled score *(S)*
1	70.8	19	1.02
2	2.2	1	4.47
3	58.1	18	1.05
4	25.9	11	1.34
5	16.6	6	1.82
6	40.8	15	1.15
7	30.5	13	1.24
8	5.7	2	3.16
9	27.4	12	1.29
10	21.0	7	1.69
11	57.1	17	1.08
12	21.7	9	1.49
13	5.9	3	2.58
14	23.4	10	1.41
15	33.7	14	1.19

A lower cost variable (i.e. lower *J* value) identifies a better fit organism. The organisms are then ranked from most fit to least fit (see Table 4-4). Note that to

determine the *J* value for each organism, the thermal simulation of the run-out table must be executed under the conditions given by that organism. The fit of an organism affects the probability that it will be used in subsequent simulations. This process of selection by fit is designed to mimic the process of natural evolution. Thus, a scaled score is assigned to each organism in Table 4-4 using Equation (4-2):

$$S = (20^{1/2})/(n^{1/2}) \qquad (4\text{-}2)$$

Where n is the ranking of the organism. Note that the sum of the scaled scores will always be 33.97. The next set of 20 organisms is ready to be generated. The next 20 organisms will be composed of three sets of organisms: 1) some of the most fit organisms, and 2) children of and 3) mutations from the previous 20 organisms. The relative number of organisms that are selected from each of these three sets is determined by trial and error for each given problem. For the run-out table, the most efficient distribution was to select the four most fit organisms, to generate twelve children and four mutations. To provide the reader insight into the mechanics of the genetic algorithm, a short example is provided as follows.

For the most fit, the elites, organism 2, 8, 13 and 17 will be used in the next population and are now labeled in Table 4-5 as organisms 21 to 24, respectively. The 12 children are generated as follows. Of the 20 organisms used in the first population (Table 4-2), 24 organisms need to be selected to be parents of 12 children for the second population. Consider a linear scale of value equal to the sum of Scale scores, namely 33.97. If the Scaled score of each organism from Table 4-4 is mapped on to this linear scale (Table 4-6), all organisms will be represented but the ones with the higher Scaling score will occupy a longer portion of the linear scale. Now two random numbers are generated between 1 and 33.97. These values will be mapped on to the linear scale and the corresponding organisms are the selected parents of a child. Assume that the two random numbers generated are 4.15 and 9.12. From Table 4-6, the selected

parents are organisms number 2 and 13 (ranks 1 and 3) and are listed in Table 4-7. A random vector 15 x 1 is generated composed of binary values (1's and 0's), also shown in Table 4-7. A *1* would result in the attribute of the first parent to be chosen for the child and a *0* the attribute of the second parent for the child. This is demonstrated in Table 4-7 for one child. This child becomes organism 25 shown in Table 4-5. This process is continued until all 12 children are populated, i.e. organisms 25 to 36 of the second population. This process is termed the cross over algorithm (Mitchell, 1996).

The generation of mutants will now be described. Of the 20 organisms in the first population, four are selected randomly for mutation. The process will be illustrated using organism 4. A random vector (15 x 1) is generated with values from 0 to 1 as shown in Table 4-8. The threshold value of 0.2 was chosen for this problem. Thus, if the value in the random vector is equal to or greater than 0.2, then the original value for that configuration of the organism is kept. If the value of the random vector is less than 0.2 then the value of the configuration in the original organism is reversed. Following the example shown in Table 4-8, the configuration values of the first and last side-spray and the second value of the strip velocity would be reversed. This will yield organism 37 in Table 4-5. Thus, the remaining mutations are generated.

The thermal model is run for each of the configurations shown in Table 4-5. The results of the cost function, rankings, and scale scores are listed in Table 4-9. In addition to minimizing the objective function in Equation 1, the optimizer must also meet process constraints. First, all nodes must have a minimum temperature of 450^{o}C to ensure that the skelp is not too difficult to coil. Also, the velocity of the skelp was assumed to between 3 and 5 m/s. The GA algorithm is terminated when no improvement in the objective function is observed for ten generations (Holland, 1975).

Table 4-5. List of Organisms for the second population

Organism No	Cooling Banks configuration	Side-sprays configuration	Velocity of the strip
Most Fit Organisms from Previous Population			
21	0 1 1 0 0 0	0 0 0 0 0 0	0 1 1
22	0 1 0 1 0 0	0 0 0 0 0 0	1 1 1
23	0 0 1 1 0 1	0 0 0 0 0 1	1 0 1
24	1 1 0 0 1 1	0 0 0 0 0 1	1 1 1
Children of previous Population			
25	0 1 1 1 0 1	0 0 0 0 0 1	1 0 1
26	0 0 1 0 1 1	0 0 0 0 0 1	1 0 1
27	0 1 0 1 1 0	0 0 0 0 0 0	1 1 1
28	0 1 1 0 1 0	0 0 0 0 0 0	0 1 1
29	0 1 1 0 0 0	0 0 0 0 0 0	1 0 1
30	1 1 1 1 0 1	1 0 1 0 0 1	1 1 0
31	1 1 1 0 1 1	0 1 0 0 0 1	1 1 1
32	1 1 1 1 1 0	0 1 0 1 1 0	1 1 0
33	0 1 1 1 0 0	0 1 0 0 0 0	0 1 1
34	1 1 0 0 0 1	0 0 0 0 0 0	0 1 1
35	0 1 0 0 0 0	0 0 0 0 0 0	1 1 1
36	1 1 1 1 1 1	0 0 0 1 1 1	1 1 1
Mutations of previous Population			
37	1 1 0 0 1 1	1 0 0 0 0 0	1 0 1

38	1 1 0 1 1 1	0 1 0 1 0 0	1 1 0
39	1 1 0 1 0 1	0 1 0 1 0 1	0 0 0
40	1 1 0 1 1 1	1 1 0 1 1 1	1 0 0

Table 4-6. Cumulative linear scale of Scores

Organism No.	Rank of individual	Scaled scores	Cumulative values of Scores, S
2	1	4.47	4.47
8	2	3.16	7.63
13	3	2.58	10.22
16	4	2.24	12.45
19	5	2.00	14.45
5	6	1.83	16.28
10	7	1.69	17.97
17	8	1.58	19.55
12	9	1.49	21.04
14	10	1.41	22.45
4	11	1.35	23.80
9	12	1.29	25.09
7	13	1.24	26.33
15	14	1.20	27.53
6	15	1.15	28.68
18	16	1.12	29.80
11	17	1.08	30.89
3	18	1.05	31.94
1	19	1.03	32.97
20	20	1.00	33.97

Table 4-7. Sample generation of a child

Organism No.	Cooling Banks configuration						Side-sprays configuration						Strip velocity (see Table 1)		
2	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1
13	0	0	1	1	0	1	0	0	0	0	0	1	1	0	1
random vector	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1
Child	0	1	1	1	0	1	0	0	0	0	0	1	1	0	1

Table 4-8. : Sample mutation of Organism 4

Organism No.	Cooling Banks configuration						Side-sprays configuration						Strip Velocity		
4	1	1	0	0	1	1	0	0	0	0	0	1	1	1	1
random vector	0.78	0.85	0.46	0.55	0.86	0.75	0.18	0.57	0.55	0.20	0.53	0.13	0.68	0.15	0.33
37	1	1	0	0	1	1	1	0	0	0	0	0	1	0	1

Table 4-9. Results of second population

Organism No	Cost value	Rank of individual	Scaled scores
Elites			
21	2.2	1	4.47
22	5.7	4	2.23
23	5.9	5	2
24	7.6	8	1.58
Children			
25	27.2	13	1.24
26	30.2	15	1.15
27	5.2	2	3.16
28	105.9	20	1
29	6.8	7	1.69
30	34.2	16	1.11
31	26.8	12	1.29
32	6	6	1.82
33	14.8	10	1.41
34	41	18	1.05
35	45.5	19	1.02
36	17.3	11	1.34
Mutations			
37	39	17	1.08
38	13.6	9	1.49
39	5.4	3	2.58
40	27.6	14	1.19

4.2 Optimization Results

The goal of this work is to determine the temperature profile of a steel strip in a laminar cooling system for three specific objective functions. In the two cases, the goal of the optimization procedure is to obtain a specific coiling temperature (regardless of the temperature profile) while maximizing skelp velocity. The coiling temperatures used were 500°C and 600°C. In cases three and four, a constant cooling rate was specified. The cooling rates used were 10°C/s, 11.5°C/s and 15°C/s. In the final set of cases, control of both the cooling rate and coiling

temperature together is achieved. This was done for different combinations of final coiling temperature (i.e., 500°C and 600°C) and cooling rate (i.e., 10°C/s, 11.5°C/s and 15°C/s).

The optimization results are summarized in Table 4-10. All optimization was done using Node B as the control node (see Figure 4-4 for Node B location) as the control point. Note, however, that the proposed method does not depend on the choice of control point and that any control point (or weighted average of control points) can be used. The optimal cooling system configuration including the optimal number of banks, side-sprays and skelp velocities are also listed in Table 4-10. The percent error listed in Table 4-10 is computed using the relation. The errors are the differences between the simulated temperatures and desirable ones (They do not include the industrial temperatures).

$$Error = 100 \times \frac{T_f - T_f^{\,s}}{T_f^{\,s}} \tag{4-3}$$

for Cases 1-2, and the relation

$$Error = 100 \times \frac{\sum_{i=1}^{i=n^*} \left| NT(i) - NT(i)^s \right|}{\sum_{i=1}^{i=n^*} \left| NT(i)^s \right|} \tag{4-4}$$

for Cases 3-5. For cases 6-12 the reported percent error is the maximum deviation between the desired and obtained temperature along the skelp. Mathematically this is expressed as

$$Error = 100 \times \max_{i \in \{1,2,\ldots,n^*\}} \frac{\left| NT(i) - NT(i)^s \right|}{\left| NT(i)^s \right|} \tag{4-5}$$

Table 4-10. Optimization Description and Results

Case No	*Target Coiling Temp (°C)*	*Target Cooling Rate (°C/s)*	*Control Node*	*Optimal Banks Configuration*	*Optimal Side-sprays Configuration*	*Optimal Velocity m/s*	*Simulated Coiling Temp (°C)*	*Error %*
1	*500*	*n.a.*	*B*	*111111*	*000001*	*4.25*	*504.6*	*0.9*
2	*600*	*n.a.*	*B*	*110101*	*000001*	*4.75*	*600.2*	*0.03*
3	*n.a.*	*10*	*B*	*110101*	*110101*	*4.5*	*636.4*	*1.2*
4	*n.a.*	*11.5*	*B*	*111001*	*111001*	*4.75*	*633.4*	*1.5*
5	*n.a.*	*15*	*B*	*100111*	*000111*	*4.75*	*601.8*	*0.9*
6	*500*	*10*	*B*	*111111*	*111001*	*3.5*	*494.8*	*15.2*
7	*600*	*10*	*B*	*100111*	*100101*	*4*	*599*	*4.3*
8	*500*	*11.5*	*B*	*011111*	*000001*	*3.5*	*506.7*	*12.6*
9	*600*	*11.5*	*B*	*110100*	*110000*	*4*	*599.6*	*4.9*
10	*500*	*15*	*B*	*111110*	*100010*	*3.25*	*504.1*	*10.1*
11	*600*	*15*	*B*	*100111*	*000111*	*4.75*	*601.8*	*1.8*
12	*500*	*10*	*B*	*101111*	*101111*	*3.5*	*581.6*	*3.9*

CR = constant cooling rate, CT = coiling temperature

1="ON", 0="OFF"

4.3 Discussion

4.3.1 Coiling Temperature Control

In the first optimization case the coiling temperature was specified to be either 500°C (for Case 1) or 600°C (for Case 2) with the objective being to maximize skelp velocity and minimize deviation from the desired coiling temperature. The coefficients of the general cost function (Equation 1) in this part are A=1, B=1, and C=0. The first term in the cost function penalizes the difference between the achieved coiling temperature and the specified coiling temperature. The second term in the cost function has penalizes the choice of a low skelp velocity. The optimal configuration for the 500°C conditions was 6 banks "on", all side-sprayside-sprays (except for the last one) "off" and a velocity equal to 4.75m/s (Case 1 in Table 4-10). The final predicted temperature is 504.6^{o}C.

While node B was used for optimization, the temperature profiles of Nodes A and C were also computed under the optimal configuration for achieving a coiling temperature of 500 °C and maximum skelp velocity. In Figure 4-6 the temperature profile of node B is compared with nodes A and C. As seen in Figure 6, the steel at all locations in the skelp will form bainite for this optimization condition (i.e. coiling temperature of 500 °C).

The optimal configuration to achieve a 600°C coiling temperature was with Banks 1, 2, 4, and 6 on, Side-spray 6 on, and velocity of 4.5 m/s (Case 2 in Table 10). The deviation from the specified coiling temperature was 0.2^{o}C.

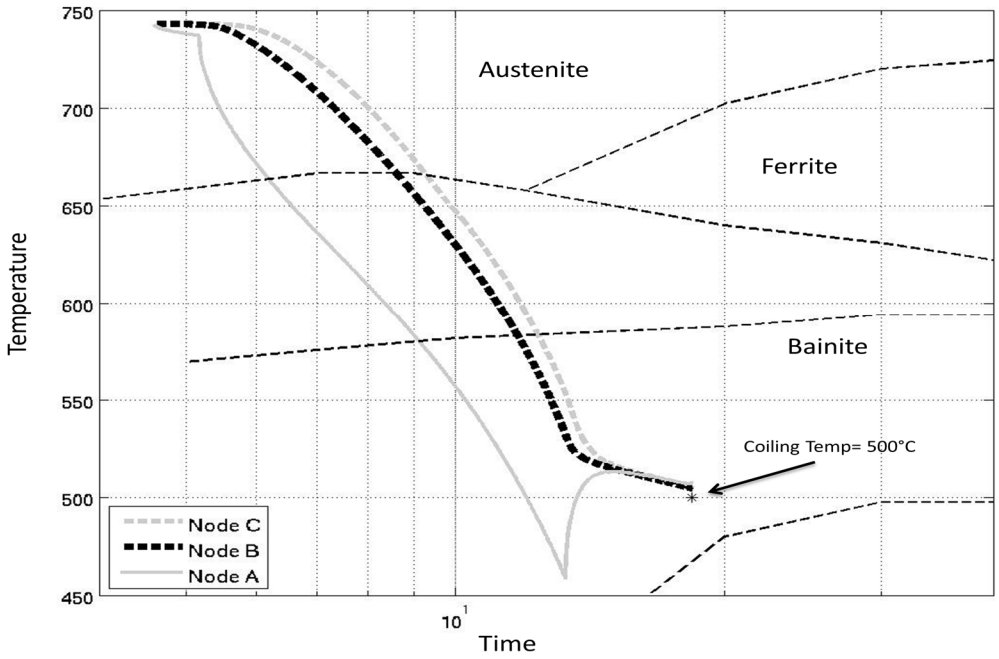

Figure 4-6. Temperature Profile for Nodes A, B and C in CCT diagram Case #1 (Objective: CT=500°C for Node B)

The temperature profiles for nodes A, B and C are shown in Figure 4-7 superimposed on a sample CCT diagram. Nodes near the surface (e.g., Node A) exhibit large variations in temperature due to their proximity to the surface where the heat flux is rapidly altered by the impact and subsequent removal of water. Nodes that are more distant from the surface (e.g., Notes B and C) show a dampened thermal response due to the lower and the more constant heat flux inside the skelp. This effect is due to the geometry of the skelp and is therefore observed in all simulations regardless of the cooling system configuration.

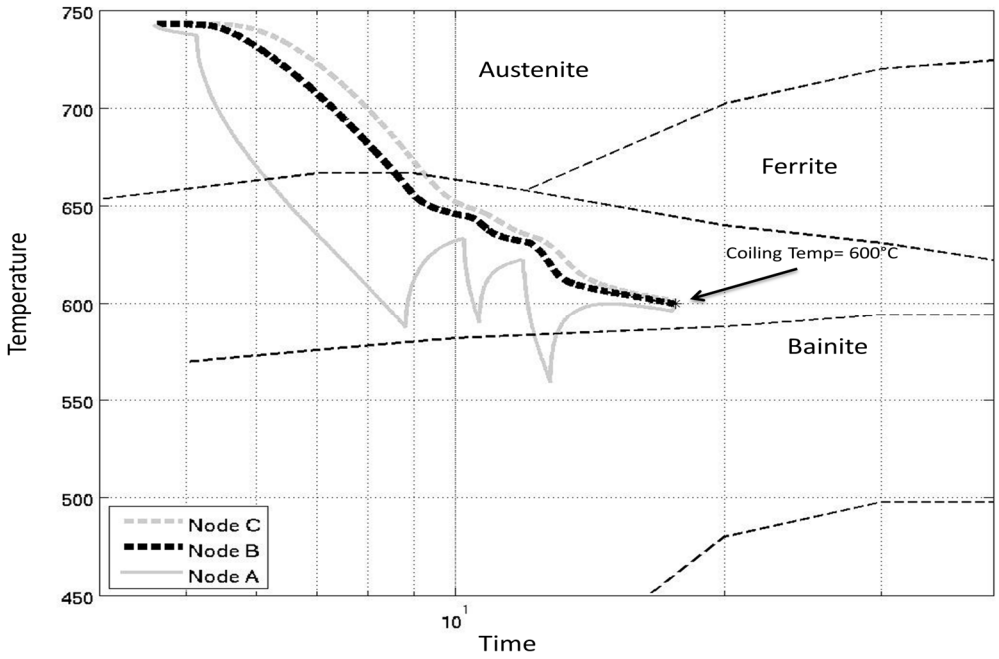

Figure 4-7. Temperature Profile for Nodes A, B and C in CCT diagram Case #2 (Objective: CT=600°C for Node B)

4.3.2 Constant Cooling Rate Control

While controlling the coiling temperature is important for mill operation, it is the *path* of the steel through the CCT diagram that ultimately governs the product microstructure and hence properties. In order to control the path of the steel through the CCT diagram, the cooling rate as it passes through the system must be controlled. While any target cooling rate can be specified using the proposed methodology, for the sake of simplicity a constant cooling rate will be used in this work. Furthermore, in this section, the value of the coiling temperature was not used in the optimization (the case where the coiling temperature and cooling rate are simultaneously controlled is addressed in the next section).

In order to obtain the desired objective function for this section, the coefficients of general cost function (described in Equation 1) are A=B=0 and C=1. The third term (with the C coefficient) in the cost function (i.e., Equation (1)) indicates the

penalty for the difference between achieved and specified temperature profile during water-cooling section.

In this part, three different cooling rates were specified; 10 °C/s, 11.5 °C/s and 15°C/s. These rates were chosen as each will result in a different microstructure. For example, the 10 °C/s will form a microstructure that is primarily ferritic while the 15°C/s rate will form a microstructure that is primarily bainitic. The optimal configuration for each cooling rate is listed in Table 4-10 (Cases 3 to 5). The optimized temperature profiles for cooling rates of 10 °C/s, 11.5 °C/s and 15°C/s are shown in Figures 4-8 to 4-10, respectively. The solid line in Figures 4-8 to 4-10 are used to indicate the specified (i.e., target) cooling rate. The difference between the specified and achieved cooling rate is measured over the entire length of the cooling system. The average error for Cases 3 to 5 were 0.87%, 0.47% and 0.84% respectively, which indicates that the average of differences in achieved and specified temperatures for all points were approximately 4.4°C, 2.2°C and 4.2°C.

Note that while these results indicate that the cooling rate can be well controlled the coiling temperature for Cases 3 to 5 were 636.4^{o}C, 633.4^{o}C and 601.4^{o}C, respectively which may be considered too high depending on operating conditions and product requirements. As a result, cooling rate control alone cannot be used to effectively be used to meet all of the mill operating requirements. Rather, cooling rate and coiling temperature must be controlled simultaneously.

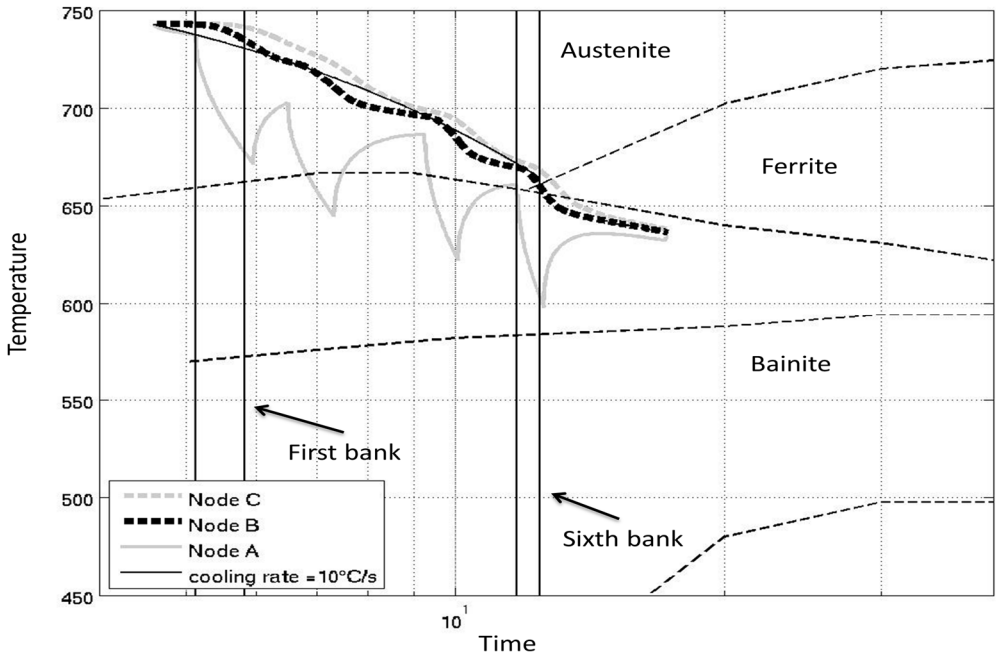

Figure 4-8. Temperature Profile for Nodes A, B and C in CCT diagram Case #3 (Objective: Constant CR=10°C/s for Node B)

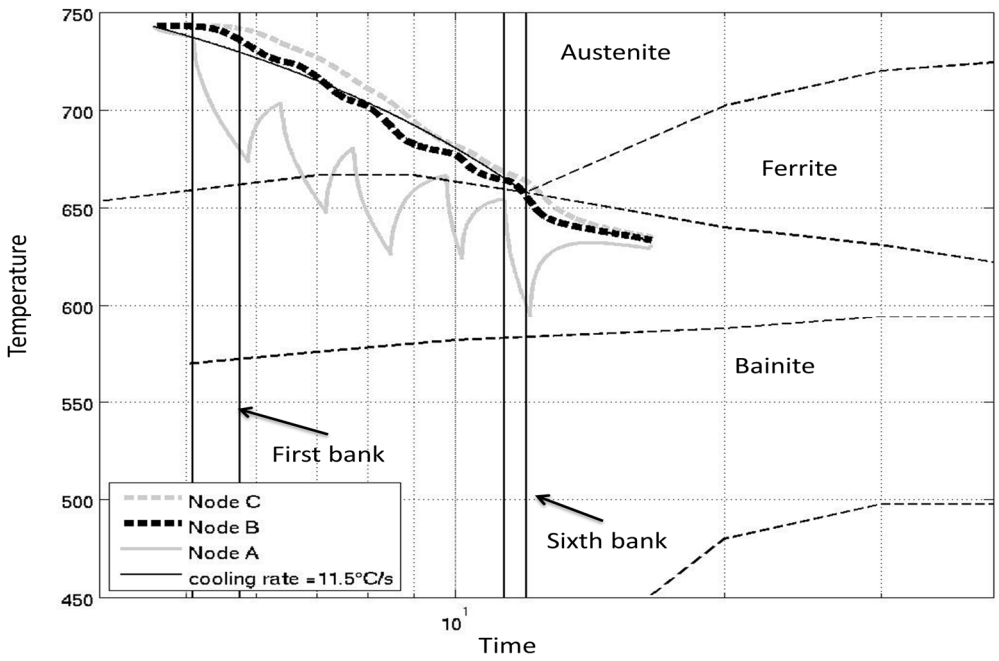

Figure 4-9. Temperature Profile for Nodes A, B and C in CCT diagram Case #4 (Objective: Constant CR=11.5°C/s for Node B)

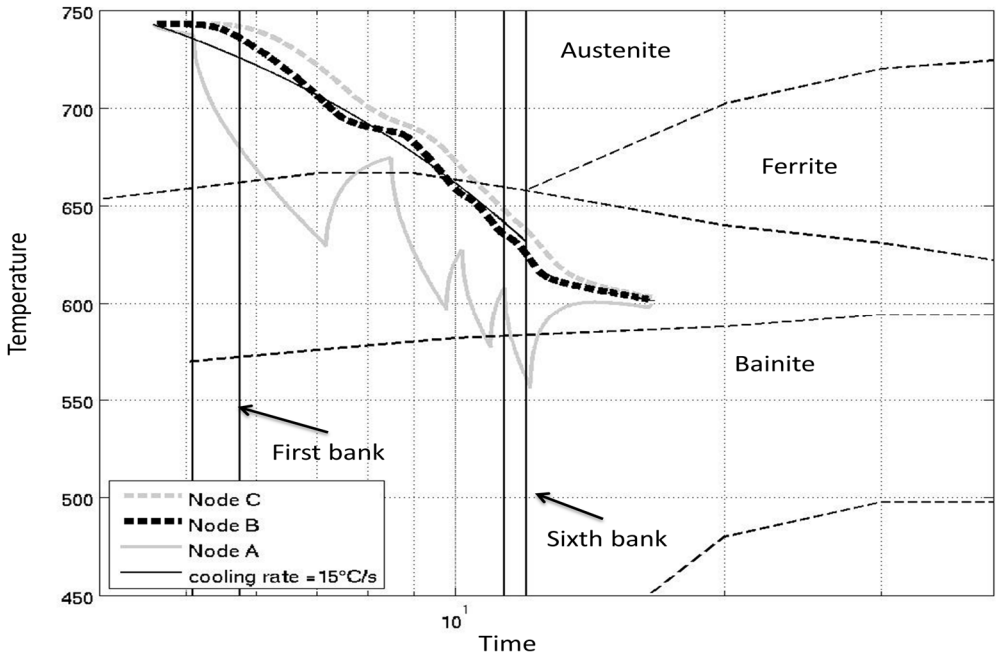

Figure 4-10. Temperature Profile for Nodes A, B and C in CCT diagram Case #5 (Objective: Constant CR=15°C/s for Node B)

4.3.3 Constant Cooling Rate & Coiling Temperature Control

As discussed in the previous section, in order to obtain the desired microstructure while meeting all of the mill's processing requirements, it may be necessary for both the constant cooling rate and final coiling temperature to be controlled. In this section the optimization objective is to control both the constant cooling rate (at 10°C/s, 11.5°C/s or 15°C/s) and the final temperature (at 500°C or 600°C) for node B.

The desire to meet both a cooling rate objective and a coiling temperature objective naturally leads to optimization trade-offs. For example, at a constant cooling rate of 10°C/s, the final coiling temperature was 640°C, which is much higher than both of the desired coiling temperatures (i.e., 500°C or 600°C). As a result, an objective function which weights both the error in the coiling

temperature and deviation from the desired cooling rate must be specified. In this work an objective function, which weighs both, equally was chosen. As a result, the coefficients in the general cost function (shown in Equation 1) were as follows; A=100, B=0, C=1. The reason for the large value of A is that there are approximately a hundred points in the temperature profile of the water-cooling section in Figure 4-3, and, as a result, the coefficient used to penalize deviation from the desired coiling temperature (i.e., A) should be 100 times larger than the coefficient used to penalize deviations from the desired temperature profile (i.e., C). These values may be altered depending on the degree of emphasis one wishes to place on the water cooling section and the coiling temperature.

The optimal configuration to achieve each of the specified cooling rates and coiling temperatures are listed in Table 4-10 (Cases 6 to 11). The maximum deviation (as a percent) from the desired cooling profile for each case is listed in Table 4-10. Also listed in Table 4-10 are the target and obtained coiling temperature for each case. The temperature profiles of nodes A, B and C versus specified cooling rate and final temperature for a coiling temperature of 500^{o}C and constant cooling rates of 10°C/s, 11.5°C/s and 15°C/s are shown in Figures 4-11, 4-12 and 4-13, respectively. Similarly, the temperature profiles for a coiling temperature of 600^{o}C and constant cooling rates at 10°C/s, 11.5°C/s and 15°C/s are shown in Figures 4-14, 4-15 and 4-16, respectively.

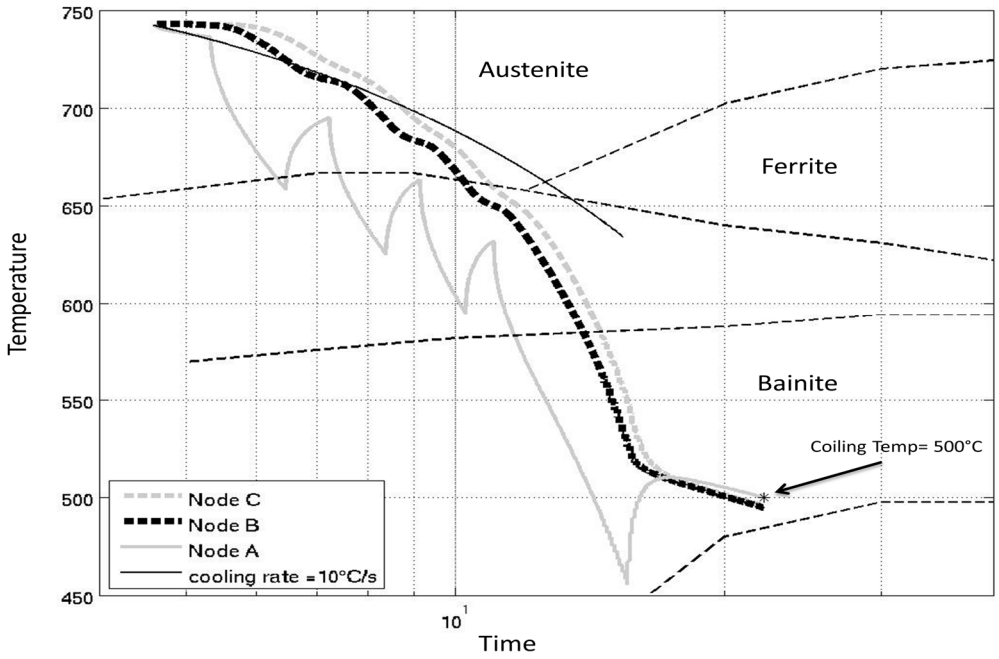

Figure 4-11. Temperature Profile for Nodes A, B and C in CCT diagram Case #6 (Objective: Constant CR=10°C/s & CT=500°C for Node B)

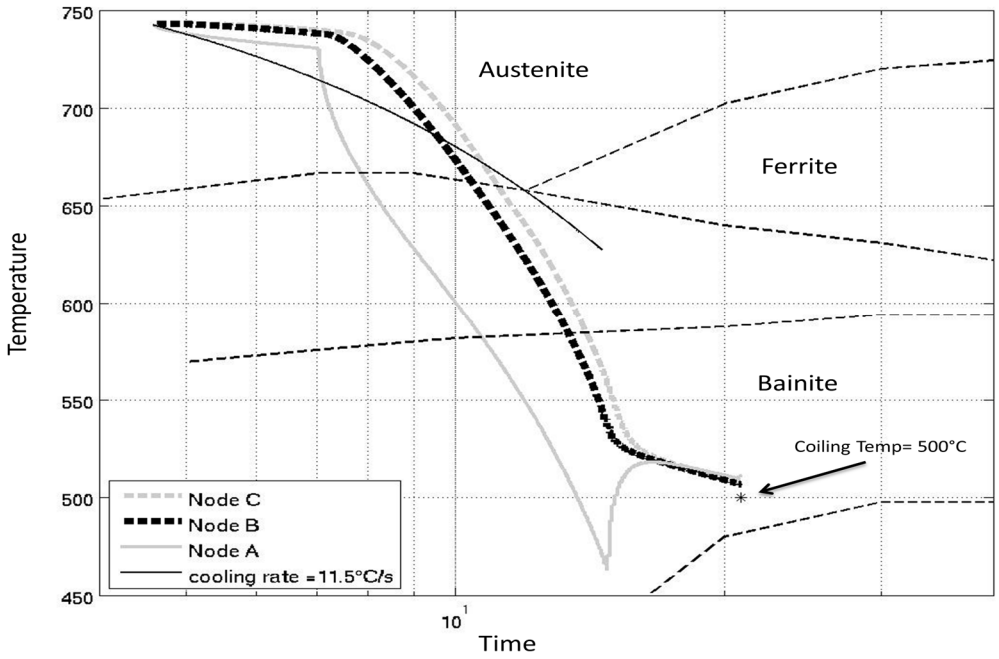

Figure 4-12. . Temperature Profile for Nodes A, B and C in CCT diagram Case #7 (Objective: Constant CR=11.5°C/s and CT=500°C for Node B)

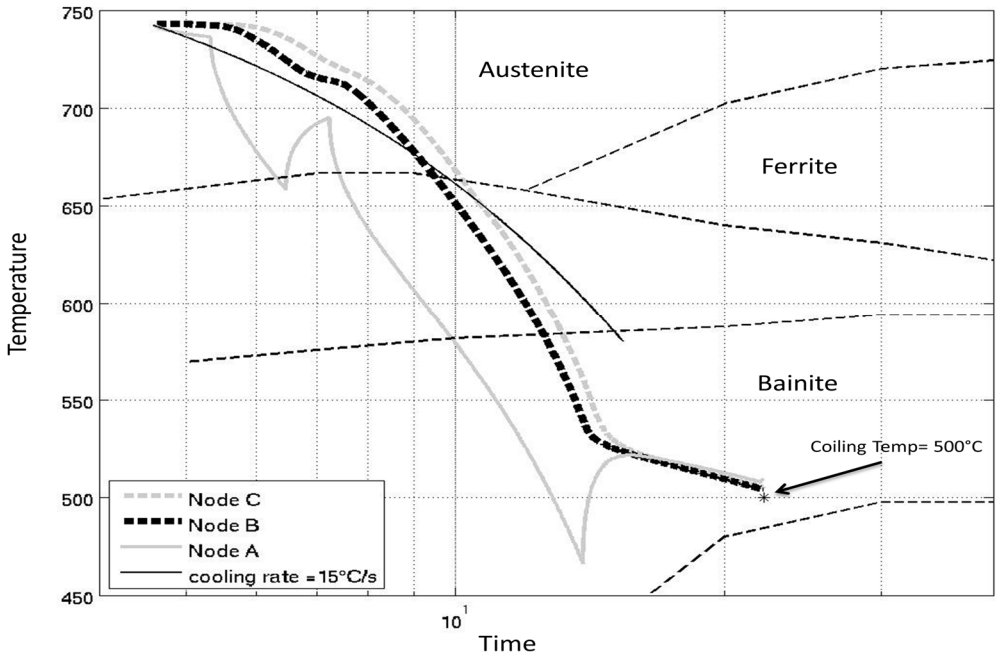

Figure 4-13. Temperature Profile for Nodes A, B and C in CCT diagram Case #8 (Objective: Constant CR=15°C/s and CT=500°C for Node B)

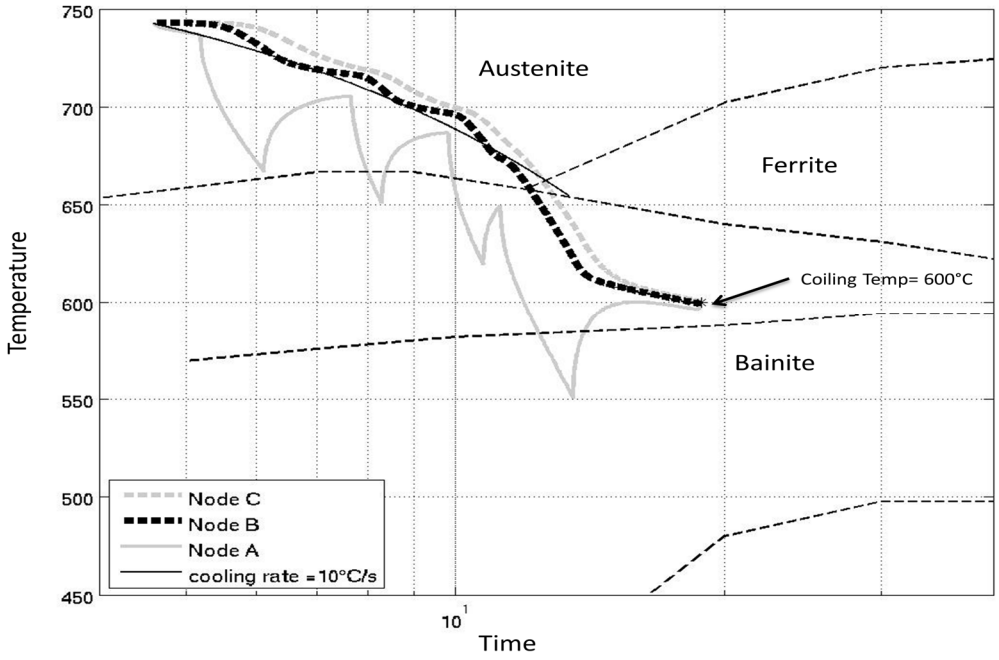

Figure 4-14. Temperature Profile for Nodes A, B and C in CCT diagram Case #9 (Objective: Constant CR=10°C/s and CT=600°C for Node B)

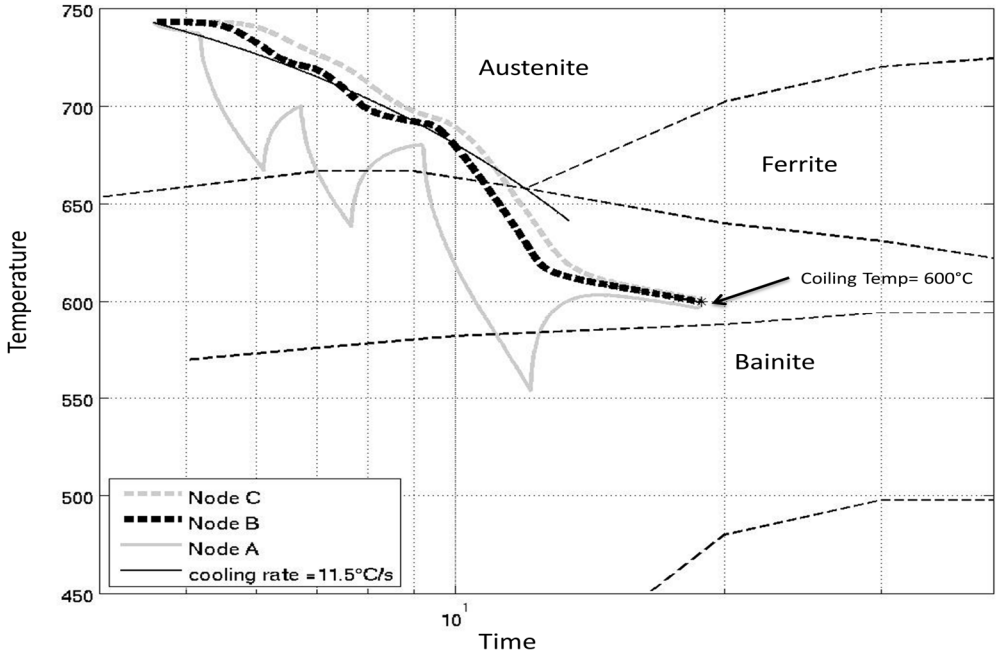

Figure 4-15. Temperature Profile for Nodes A, B and C in CCT diagram Case #10 (Objective: Constant CR=11.5°C/s and CT=600°C for Node B)

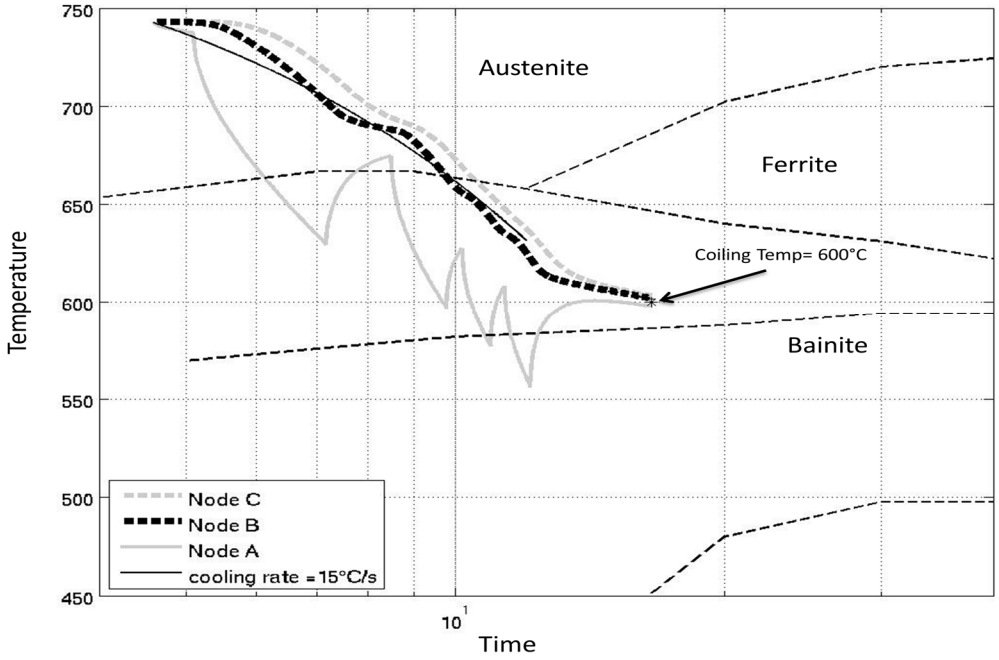

Figure 4-16. Temperature Profile for Nodes A, B and C in CCT diagram Case #11 (Objective: Constant CR=15°C/s and CT=600°C for Node B)

As can be seen from Figures 4-11 to 4-13, it is not possible to simultaneously obtain the target coiling temperature and the target cooling rate. This is because once the constant cooling rate is assigned for the entire length of the cooling system; the coiling temperature cannot be independently manipulated. Future work should focus on specifying a cooling rate for only a portion of the cooling system, leaving enough banks for control of the coiling temperature (e.g., a non-uniform cooling regime). However, even under the current scheme, it is possible to "trade off" error in the cooling rate against error in the coiling temperature. This can be achieved by varying the value of A relative to C in the objective function. To illustrate this idea, the temperature profile of a run with a specified constant cooling rate of 10^{o}C, a target coiling temperature of 500^{o}C (same as Case 6 in Figure 4-11) but with objective function parameters A = 10 and C=1 is shown in Figure 4-17. Notice that the cooling rate in Figure 4-17 is much closer to the target value of 10^{o}C/s than the results of Run 6 shown in Figure 4-11. This is because much less weight is given to the final coiling temperature (i.e., A=10 as opposed to A=100 in Run 6).

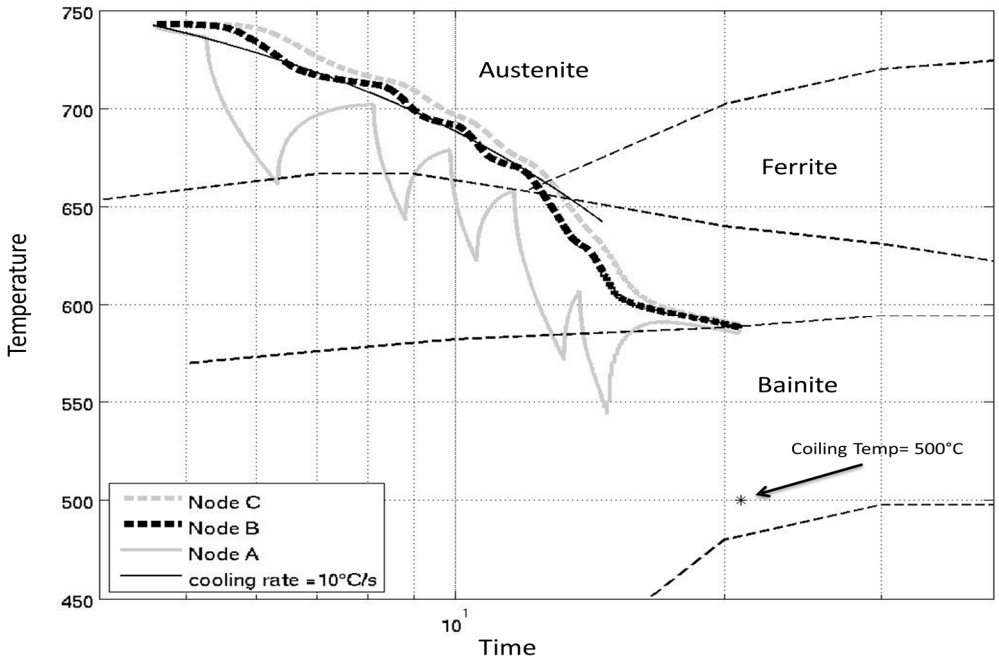

Figure 4-17. Temperature Profile for Nodes A, B and C in CCT diagram Case #12 (Objective: Constant CR=10°C/s and CT=500°C for Node B)

4.4 Sensitivity Analysis

The final temperature profile in the skelp is a function of the configuration of the cooling system. In this work, a configuration that is optimal for a specific node (Node B) was computed. However, this configuration may be suboptimal for other nodes. More generally, it is important to carefully examine the effect of node selection on the optimal cooling system configuration and temperature profiles within the skelp.

To assess the sensitivity of the system configuration temperature profiles to the choice of control node, two additional nodes were selected as a target for optimization. Due to the symmetry of the system, these two additional nodes were placed at the same horizontal position as Node B, but at different distances from

the skelp surface. These nodes, labeled Node A, and Node C in Figure 4-5, correspond to a near-surface node and a centerline node, respectively. The optimization objective chosen for this analysis was a constant cooling rate of 11.5°C, and a coiling temperature of 600°C. The objective function weights were A=100, B=0, and C=1.

The optimal configurations for the cooling-system using Nodes B, C, and A are listed in Table 4-11 as Cases 9, 13, and 14, respectively. Also listed in Table 4-11 is the optimal velocity, simulated coiling temperature, and the maximum deviation between the desired and simulated temperature profile for each run.

Table 4-11. Sensitivity analysis

Case No	*CT (°C)*	*CR (°C/s)*	*Control Node*	*Optimal Banks Configuration*	*Optimal Side-sprays Configuration*	*Optimal Velocity m/s*	*Simulated Coiling Temp (°C)*	*Error %*
9	*600*	*11.5*	*B*	*110100*	*110000*	*4*	*599.6*	*4.9*
13	*600*	*11.5*	*C*	*111011*	*111001*	*4.5*	*600.3*	*1.7*
14	*600*	*11.5*	*A*	*011011*	*011011*	*3.5*	*631.6*	*6.7*

As shown in Figures 4-15 and 4-18, the temperature profiles of Nodes B and C are offset with Node C being hotter than Node B. However, in both cases the system can be configured so that the control node does follow the desired temperature profile. It is important to note, however, that although the temperature profiles obtained for Nodes B and C are similar, the cooling system configuration used to obtain these profiles are different.

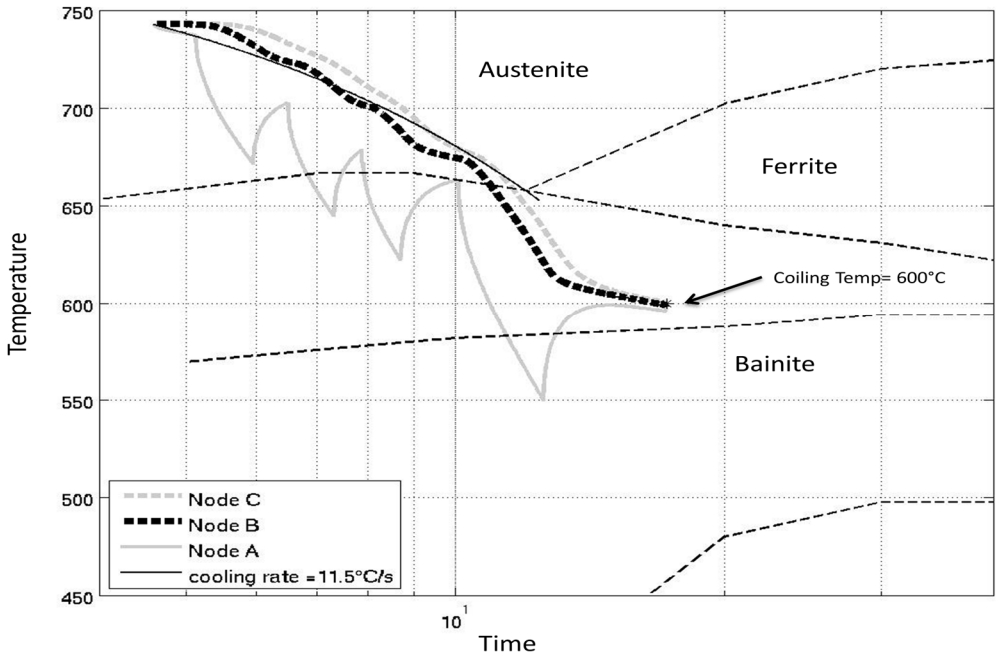

Figure 4-18. Temperature Profile for Nodes A, B and C in CCT diagram Case #13 (Objective: Constant CR=11.5°C/s and CT=600°C for Node C)

In the case of Node A the effect of the direct water-cooling is very noticeable. As a result, the temperature profile for Node A is much more jagged than that for Nodes B and C. This effect is due to the proximity of Node A to the skelp surface. More generally, it is not possible to obtain a smooth cooling profile for Node A. Rather, the optimal temperature profile for Node A can, at best, be centered about the desired cooling profile.

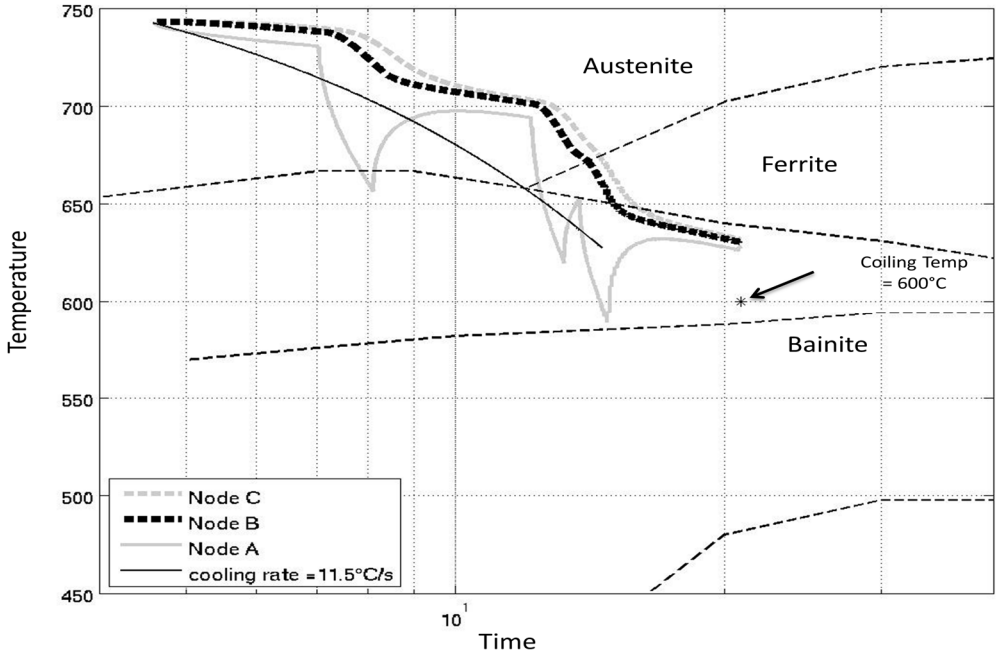

Figure 4-19. Temperature Profile for Nodes A, B and C in CCT diagram Case #14 (Objective: Constant CR=11.5°C/s and CT=600°C for Node A)

The choice of control node is important for obtaining the desired microstructure in the skelp. For example, nodes that are further from the cooling surface are more representative of the bulk temperature profile in the skelp. However, near-surface nodes provide a measure of the maximum temperature variation during the cooling process.

4.5 Summary and Conclusions

The mechanical properties of steel strip are defined by the steel microstructure. This microstructure is a function of the temperature profile in the strip during the cooling process. In this work, a genetic-algorithm based approach for obtaining a desired temperature profile in the strip is presented. It is shown that genetic algorithms are a computationally viable tool for obtaining optimal configurations for the laminar cooling system. Three general cases were considered: a fixed

coiling temperature, a constant cooling rate, and both a fixed coiling temperature and constant cooling rate. The optimization algorithm was used to, in simulation, provide configurations that could closely match the desired coiling temperature and cooling rates for the first two cases. However, when both coiling temperature and the cooling rate were specified the physical layout of the system implied that both could not be satisfied simultaneously.

Sensitivity analysis showed that the choice of control node greatly affects the optimal cooling system configuration. The temperature of near-surface nodes is much more difficult to control than near-centerline nodes because of the greater heat flux near the surface.

The methodology given by the genetic algorithm is very flexible. Firstly, there is the issue of how best to "trade-off" deviation from the desired temperature profile against deviation from the desired coiling temperature. It means that GA has flexibility to be tuned in order to bias to one of its optimization goals during optimization process. Secondly, the connection between temperature profile and microstructure can be more closely examined. Finally, the procedure proposed in this work could be used to evaluate potential cooling systems with respect to their ability to generate a desired cooling profile (and hence a desired microstructure).

References

(Biswas S.k., 1997) Biswas S.k. Chen S.J., Satyanarayana A. *"Optimal temperature tracking for accelerated cooling processes in hot rolling of steel"*, Dynamics and Controls. Vol. 7, 1997, pp. 327–340.

(Chen S.B., 1990) Chen, S. B. "*Modelling and Analysis of Controlled Cooling for Hot Moving Metal Plates"* Monitoring and Control for Manufacturing Processes, Vol. 44, 1990.

(Goldberg D.E., 1989) Goldberg D.E. *"Genetic Algorithms in Search, Optimization and Machine Learning "*, Boston, MA. : Kluwer Academic Publishers, 1989.

(Guo R.M., 1997) Guo R.M. *"Modelling and simulations of run-out table cooling control using feed-forward-feedback and element tracking system"*, IEEE Transactions on Industry Applications. Vol. 33, 1997, p. 304.

(Holland J.H., 1975) Holland J.H. *"Adaption in Natural and Artificial Systems"*, Ann Arbor, University of Michigan Press, MI, 1975.

(Latzel S., 2001) Latzel S. *"Advanced Automation Concept of Run-out Table Skelp Cooling for Hot Skelp and Plate Mills"*, IEEE Transactions on Industry Applications . Vol. 37, 2001, p. 1088.

(Mitchell M., 1996) Mitchell M. *"An Introduction to Genetic Algorithms"*, Cambridge, MA : MIT Press, 1996.

(Peng L., 2008) Peng L. Li Q., Zhou Z. *"Cooling hot rolling steel strip using combined tactics"*, Journal of University of Science and Technology Beijing. No 3 : Vol. 15, June 2008. p. 362.

(Samaras N.S., 2001) Samaras N.S. Simaan M.A. *"Novel Control Structure for Runout Table Coiling Temperature Control"*, AISE steel technology. No 6 : Vol. 87, June 2001. pp. 55-59.

(Wiskel J.B., 2011) Wiskel J.B., Deng H., Jefferies C., Henein H., *"Infrared Thermography of a TMCP Microalloyed steel skelp at the upcoiler and its application in quantifying the laminar jet/skelp interaction"*, ISM. Vol. 38(1), 2011, pp. 35-44.

(Xie H. B., 2006) Xie H. B. Liu X. , Wang G. , Zhang Z., *"Optimization and Model of Laminar Cooling Control Thickness measurement System for Hot Strip Mills"*, JOURNAL OF lRON AND STEEL RESEARCH, INTERNMIONAL. No 13(1) : Vol. 9, 2006. pp. 18-22.

(Xu F., 2006) Xu F. *"Finite element simulation of water cooling process of steel strip on tun out table"*, PhD thesis in Univeristy of British columbia, 2006.

5

5) Skelp temperature profile optimization during laminar cooling using Particle Swarm Optimization

Introduction

This chapter reviews the results of the PSO algorithm. The PSO is an evolutionary optimization technique, which is suitable for this type of problem. This method is used in the same manner like GA algorithms in previous Chapter. Optimizations to achieve desired coiling temperature, cooling rate and both of them are done using PSO method. These optimal cooling system configurations are validated by simulation the thermal model.

5.1 Optimization Algorithms

In order to compute an optimal configuration for the system, an optimization procedure is required. In this Chapter Particle Swarm Optimization (PSO) method (Kennedy J., 1995) was used to determine the optimal laminar cooling configuration for achieving both the desired temperature profile and coiling temperature. PSO and GA are evolutionary optimization procedures, which can be applied to systems where derivative information is not always available. PSO

(1995) is a newer version of evolutionary algorithms than GA, which its first concepts introduced in 1950's. PSO is inspired from the swarm intelligence (SI) theories, which its concept is employed to work on artificial intelligence (AI). In this section a binary PSO is implemented to find optimal configuration for the laminar cooling system to achieve the temperature goals. The set of possible configurations, range of skelp velocity and the way for digitization of the skelp velocity, using 3 binary digits, are the same as the previous chapter (GA optimization). PSO provides a computationally efficient approach for searching the configuration space while having to simulate only a small number of the total possible configurations.

5.1.1 Particle Swarm Optimization

In concept, the PSO method is similar to genetic algorithms, but it does not have crossover and mutation. The terms "individuals" and "population" in a GA are called "particle" and "swarm" in PSO. Unlike in a GA, there is no natural evolution in PSO to evolve the population to the next generation. In PSO, each particle's position is adjusted in order to pursue that particle's and swarm's best previous positions.

Each particle aims to reach the global minimum using the swarm's experiences. The best result for each particle is called "lbest" (local best), while the best result in all particles (swarm) is called "gbest" (global best). The speed of each particle is modified to reach both of these items. The "gbest" guarantees the search for the global minimum, while the "lbest" guarantees the search on local minimum neighborhood. It is similar to the social behaviour of humans, who use their own experiences in addition to experiences from society, to reach their goals. At each step, the speed of the particles toward the global best is determined by the differences between particle's positions plus "lbest" and "gbest."

A discrete binary PSO developed by Kennedy et al. in 1997 (Hassan R., 2006) represents problems that contain binary inputs. Complete introduction to PSO was brought in Chapter 3. As it was introduced in Chapter 3, each particle's position is defined as x_i^k, where i is the particle number and k is the swarm iteration (i.e. generation) number. In this work, each particle of the swarm is a 15-dimensional vector as follows (d=15):

$$x_i^k = \left[x_{i1}^k, x_{i2}^k, ..., x_{id}^k\right] \tag{3-9}$$

An 20-dimensional population is defined as (n=20):

$$Pop^k = \left[x_1^k, x_2^k, x_3^k, ..., x_n^k\right] \tag{3-10}$$

For adjusting the particle's position, the particle's velocity is defined as:

$$v_i^k = \left[v_{i1}^k, v_{i2}^k, ..., v_{id}^k\right] \tag{3-11}$$

The first population is generated randomly. Each particle is its PB (particle best) and the best individual with regard to its cost value is GB (global best) for first generation. For discrete binary PSO, two functions are required to modify each particle to be 0 or 1. Piece-wise function (equation 3-16) and keeps the particle's speed between $V_{\min}$ and $V_{\max}$, and sigmoid function (equation 3-17) modify the velocities to 0 and 1.

The velocity of digit d of particle *i* in generation number *k* is:

$$\Delta v_{id}^{k-1} = w v_{id}^k + c_1 r_1 (pb_{id}^{k-1} - x_{id}^{k-1}) + c_2 r_2 (gb_d^{k-1} - x_{id}^{k-1}) \tag{3-20}$$

Where *w* is the inertia factor, which is between 0.4 and 1.4; c_1 is the self-confidence factor, which is between 1.5 and 2; and c_2 is swarm confidence factor, which is between 2 to 2.5. The original PSO algorithm uses the values of 1, 2 and 2 for *w*, c_1 and c_2 respectively. The tuning of the PSO algorithm weight factors is

a topic of parameters optimization but is outside the scope of this work (Hassan R., 2006). For all the problems investigated in this work, the weight factors use the values of 0.5, 1.5 and 1.5 for *w*, c_1 and c_2 respectively.

Finally, position is updated according to particle's velocity:

$$x_i^{k+1} = x_i^k + v_i^{k+1}\Delta t \tag{3-21}$$

The d^{th} dimension of particle i is updated as:

$$\text{If } U(0,1) < sigmoid(v_{id}^k) \text{ then } x_i^d = 1 \qquad \text{Else } x_i^d = 0 \tag{3-22}$$

The stopping criterion for this optimization is "stall generation = 10", which means if there were no improvements in GB after 10 generations, the process would be stopped. The PSO continues for next generation by updating its particle bests and global best for each generation. PSO needs only three parameters (i.e. c_1, c_2 and w) to be defined, which is fewer than GA with lots of parameters; this is the most important advantage of PSO in comparison with GA. In next section, the results of optimization using PSO are presented and in the Chapter 7, some results of PSO and GA are compared.

5.2 Optimization Results

Like Chapter 4, the goal of this section is to determine the temperature profile of a steel strip in a laminar cooling system for three specific objective functions. In the first two cases, the goal of the optimization procedure is to obtain a specific coiling temperature (regardless of the temperature profile) while maximizing skelp velocity. The coiling temperatures used were 500°C and 600°C. In cases three and four, a constant cooling rate was specified. The cooling rates used were

10°C/s and 11.5°C/s. In the final set of cases, control of both the cooling rate and coiling temperature together is achieved. This was done for the combination of final coiling temperature equal to 600°C and cooling rate equal to 10°C/s.

The optimization results are summarized in Table 5-1. All optimization was done using Node B as the control node (see Figure 4-4 for Node B location) as the control point. Note, however, that the proposed method does not depend on the choice of control point and that any control point (or weighted average of control points) can be used. The optimal cooling system configuration including the optimal number of banks, side-sprays and skelp velocities are also listed in Table 5-1. The percent error listed in Table 5-1 is computed using the relation. The errors are the differences between the simulated temperatures and desirable ones (They do not include the industrial temperatures).

$$Error = 100 \times \frac{T_f - T_f{}^s}{T_f{}^s} \tag{5-1}$$

for Cases 1-2, and the relation

$$Error = 100 \times \frac{\sum_{i=1}^{i=n^*} \left| NT(i) - NT(i)^s \right|}{\sum_{i=1}^{i=n^*} \left| NT(i)^s \right|} \tag{5-2}$$

for Cases 3-4. For case 5 the reported percent error is the maximum deviation between the desired and obtained temperature along the skelp. Mathematically this is expressed as

$$Error = 100 \times \max_{i \in \{1,2,\ldots,n^*\}} \frac{\left| NT(i) - NT(i)^s \right|}{\left| NT(i)^s \right|} \tag{5-3}$$

Table 5-1. Optimization Description and Results

Case No	*Target Coiling Temp (°C)*	*Target Cooling Rate (°C/s)*	*Control Node*	*Optimal Banks Configuration*	*Optimal Side-sprays Configuration*	*Optimal Velocity m/s*	*Simulated Coiling Temp (°C)*	*Error %*
1	*500*	*n.a.*	*B*	*111110*	*010000*	*3.75*	*497.6*	*1*
2	*600*	*n.a.*	*B*	*011111*	*011011*	*4.75*	*600.3*	*0.02*
3	*n.a.*	*10*	*B*	*101011*	*101010*	*4.75*	*645.4*	*1.8*
4	*n.a.*	*11.5*	*B*	*101100*	*101100*	*4.75*	*650.2*	*4.1*
5	*600*	*10*	*B*	*100111*	*110101*	*4.00*	*600.1*	*4.4*

CR = constant cooling rate, CT = coiling temperature

1="ON", 0="OFF"

5.3 Discussion

5.3.1 Coiling Temperature Control

In the first optimization case the coiling temperature was specified to be either 500°C (for Case 1) or 600°C (for Case 2) with the objective being to maximize skelp velocity and minimize deviation from the desired coiling temperature. The coefficients of the general cost function (Equation 1) in this part are A=1, B=1, and C=0. The optimal configuration for the 500°C conditions was 5 banks "on", one side-spray "on" and a velocity equal to 3.75m/s (Case 1 in Table 5-1). The final predicted temperature is 497.6°C. In Figure 5-1 the temperature profile of node B is compared with nodes A and C. As seen in Figure 5-1, the steel at all locations in the skelp will form bainite for this optimization condition (i.e. coiling temperature of 500 °C).

The optimal configuration to achieve a 600°C coiling temperature was with 5 Banks "on" (except first bank), 4 Side-sprays "on", and velocity of 4.75 m/s (Case 2 in Table 5-1). The deviation from the specified coiling temperature was 0.2°C.

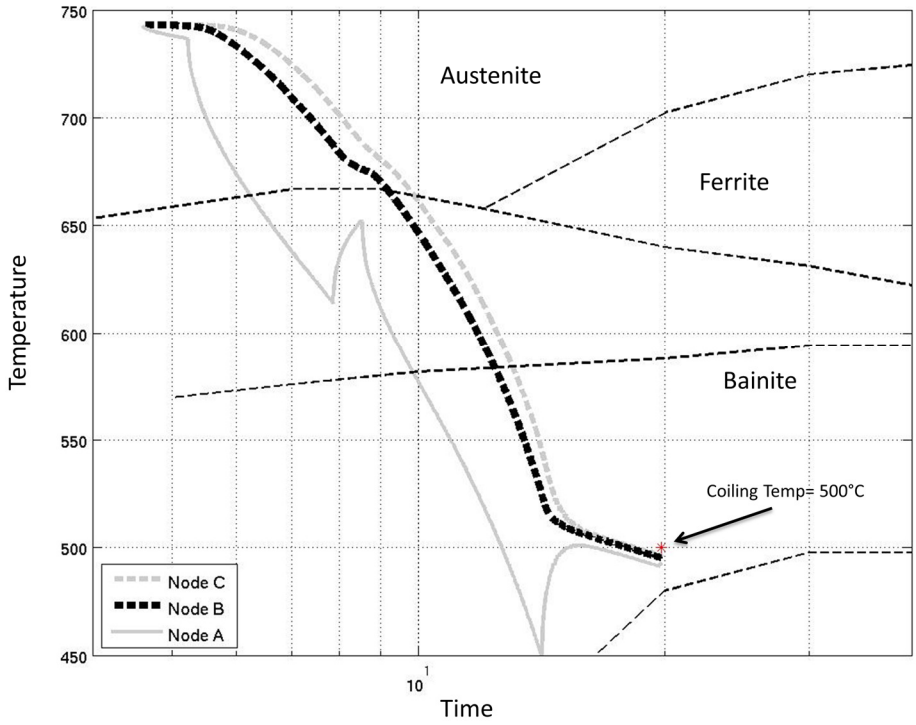

Figure 5-1. Temperature Profile for Node A, B and C in CCT diagram (Objective: Coiling Temperature=500°C for Node B)

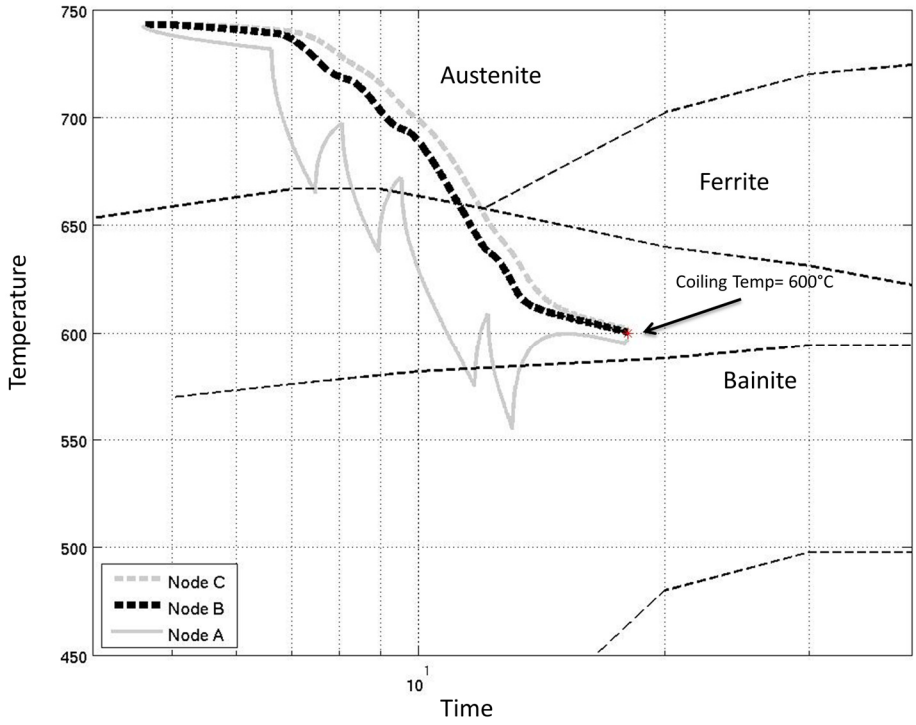

Figure 5-2. Temperature Profile for Node A, B and C in CCT diagram (Objective: Coiling Temperature=600°C for Node B)

5.3.2 Constant Cooling Rate Control

While controlling the coiling temperature is important for mill operation, it is the *path* of the steel through the CCT diagram that ultimately governs the product microstructure and hence properties. In order to control the path of the steel through the CCT diagram, the cooling rate as it passes through the system must be controlled. While any target cooling rate can be specified using the proposed methodology, for the sake of simplicity a constant cooling rate will be used in this work. Furthermore, in this section, the value of the coiling temperature was not used in the optimization (the case where the coiling temperature and cooling rate are simultaneously controlled is addressed in the next section).

In order to obtain the desired objective function for this section, the coefficients of general cost function (described in Equation 4-1) are A=B=0 and C=1. The third

term (with the C coefficient) in the cost function (i.e., Equation (4-1)) indicates the penalty for the difference between achieved and specified temperature profile during water-cooling section.

In this part, three different cooling rates were specified; 10 °C/s and 11.5 °C/s. These rates were chosen as each will result in a different microstructure. The optimal configuration for each cooling rate is listed in Table 5-1 (Cases 3 and 4). The optimized temperature profiles for cooling rates of 10 °C/s and 11.5 °C/s are shown in Figures 5-3 and 5-4, respectively. The solid line in Figures 5-3 and 5-4 are used to indicate the specified (i.e., target) cooling rate. The difference between the specified and achieved cooling rate is measured over the entire length of the cooling system. The average error for Cases 3 and 4 were 1.8% and 4.1% respectively.

Note that while these results indicate that the cooling rate can be well controlled the coiling temperature for Cases 3 and 4 were 645.4^{o}C and 650.2^{o}C, respectively which may be considered too high depending on operating conditions and product requirements. As a result, cooling rate control alone cannot be used to effectively be used to meet all of the mill operating requirements. Rather, cooling rate and coiling temperature must be controlled simultaneously.

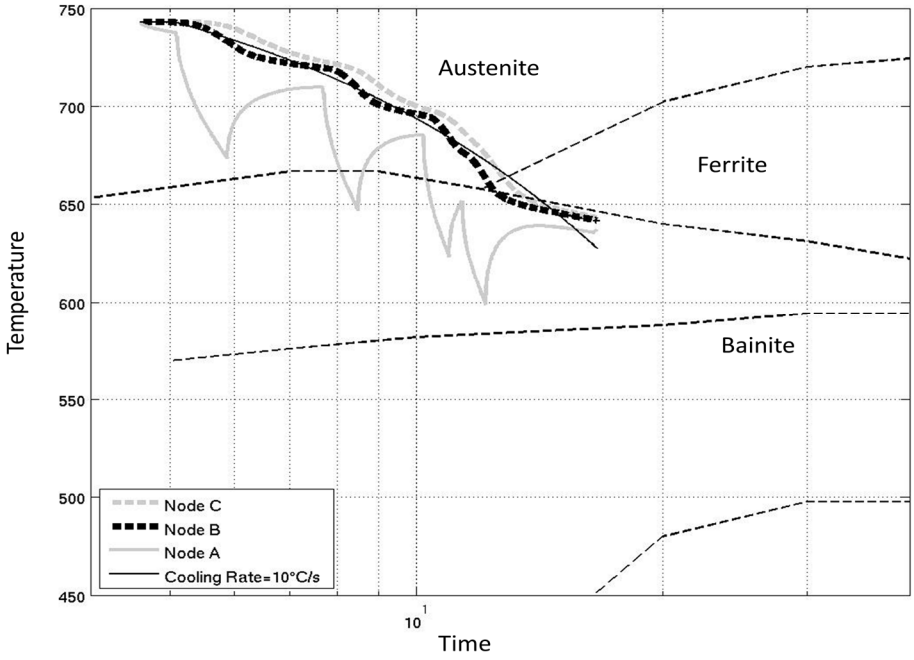

Figure 5-3. Temperature Profile for Node A, B and C in CCT diagram (Objective: Constant Cooling Rate=10°C/s for Node B)

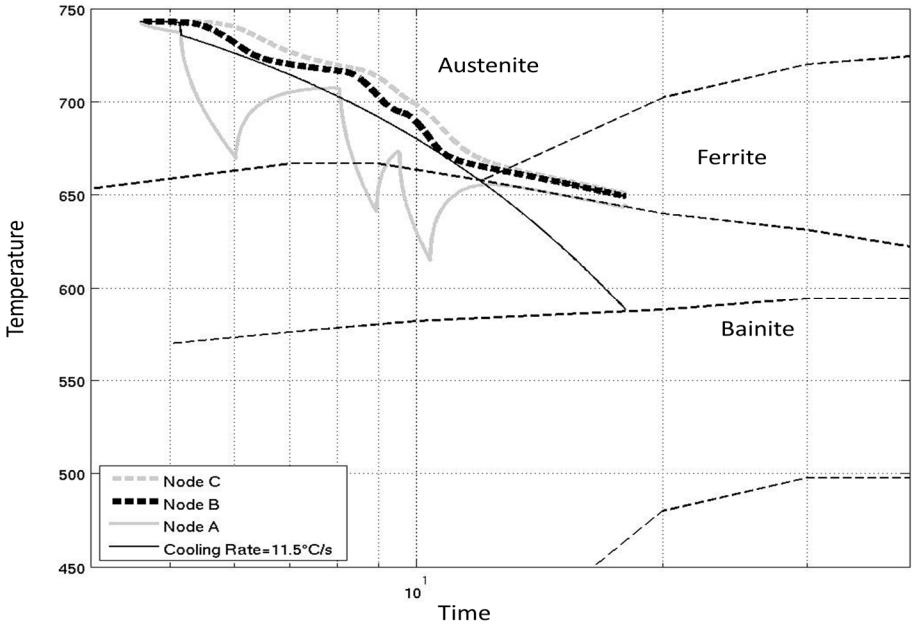

Figure 5-4. Temperature Profile for Node A, B and C in CCT diagram (Objective: Constant Cooling Rate=11.5°C/s for Node B)

5.3.3 Constant Cooling Rate & Coiling Temperature Control

As discussed in the previous section, in order to obtain the desired microstructure while meeting all of the mill's processing requirements, it may be necessary for both the constant cooling rate and final coiling temperature to be controlled. In this section the optimization objective is to control both the constant cooling rate (at 10°C/s) and the final temperature (at 600°C) for node B.

The coefficients in the general cost function (shown in Equation 4-1) were as follows; A=100, B=0, C=1. The optimal configuration to achieve the specified cooling rates and coiling temperature is listed in Table 5-1 (Cases 5). The maximum deviation (as a percent) from the desired cooling profile for each case is listed in Table 5-1. Also listed in Table 5-1 is the target and obtained coiling temperature for this case. The temperature profiles of nodes A, B and C versus specified cooling rate and final temperature for a coiling temperature of 600°C and constant cooling rates of 10°C/s are shown in Figures 5-4.

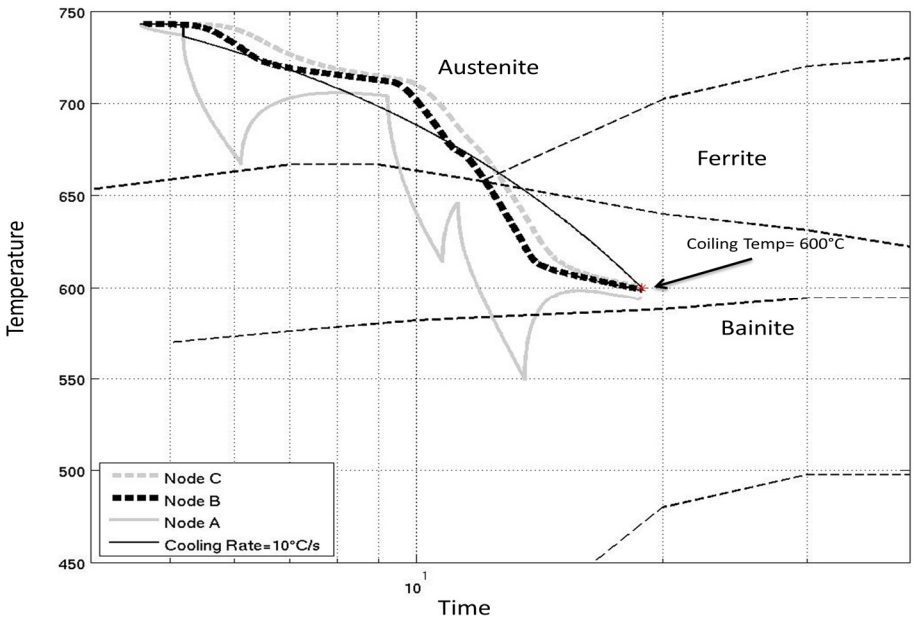

Figure 5-5. Temperature Profile for Node A, B and C in CCT diagram (Objective: Constant Cooling Rate=11.5°C/s for Node B)

5.4 Summary and Conclusions

In this Chapter, the results of PSO were presented. PSO is a newer version of evolutionary optimization in comparison with GA and its results are acceptable for this problem; however, PSO has not a distinctive advantage on GA with respect to errors of the best solutions or time of the optimization process for this problem. The reason is that both of these methods are stochastic optimization methods and are based on random search in the solution space. In next Chapter, a deterministic method is presented and developed for this problem.

References

Kennedy J. Eberhart R. *"Particle Swarm Optimization"*, Proceedings of IEEE International Conference on Neural Networks. Vol. IV, 1995, pp. 1942-1948.

(Hassan R., 2006) Hassan R. Cohanim B., de Weck O., *"A Comparison of particle swarm optimization and genetic Algorithms"*, - Colorado Springs : Vanderplaats Research and Development, Inc., 2006.

6

6) Microstructure optimization during laminar cooling of steel[2]

Introduction

Steel is by far the most widely used metallic material in the world. With 1199 million tons produced in 2009 and 706 million tons in the first six months of 2010 (Hunt P., 2011), the production continues to increase along with global economic growth. In a steel hot-rolling mill, cooling on the run-out table is one of the last and most important sections. The metallurgical characteristics of hot-rolled steel depend significantly on the thermal processes that steel is subjected to. The controlled cooling on the run-out table of a hot-strip mill is accomplished by supplying water to the top and bottom surfaces of the strip, using water jets issuing from circular nozzles. The system is often called laminar cooling because of the streamlined appearance of the jets (Avilla V.H., 1994).

[2] This Chapter is reprint of: Binesh B. Wiskel J.B., Ben-Zvi A. Henein H., *"Microstructure optimization during laminar cooling of steel "*, 2011.

In this work, the main objective was to achieve the desired temperature profile during laminar cooling of steel to control the microstructure of steel during the cooling process, which guarantees the desired mechanical and metallurgical properties of the steel product. A control policy is anticipated from each optimization to obtain a desired coiling temperature at the end of cooling section, or a desired cooling rate (temperature profile) during laminar cooling. In addition, the optimization algorithm was expected to find the optimum configuration between all acceptable solutions with the highest possible velocity. The proposed cooling strategies are simulated, using a thermal finite element model (FEM) of the laminar cooling system. Branch-and-bound (BB) as an optimization algorithm, is used to generate the optimal configuration of the laminar cooling system.

Finding the optimal solution for attaining the desired properties of product steel requires an optimization process that explores all possible combinations of manipulate variables. Global optimization techniques have been used extensively due to their capability in handling complex engineering problems. Optimization methods in general can be classified into two main categories: deterministic and stochastic (or heuristic) optimizations. Deterministic methods require the optimization problem to have definite mathematical behaviours (linear and nonlinear) that do not exist in most of the simulation-based optimization problems. However, stochastic optimization methods are based on a random search and do not guarantee to obtain the global optimum result (Younis A., 2010). The branch-and-bound technique (BB) is mostly used in medium-size problems because it is exponential, and hence requires excessive time and resources (including computational power) for larger-scale problems. By decreasing the configuration possibilities, branch-and-bound is optimal as a deterministic method to configure the run-out table and achieve desired cooling strategies, as well as to control the temperature of maximum volume of the strip within a desired area of the CCT diagram.

6.1 Background

In this section, a review of the laminar cooling system and of control strategies necessary for the implementation of the proposed methodology will be undertaken. In addition, the finite element model used to provide temperature data to the optimization system will be discussed.

6.1.1 Laminar Cooling

A schematic of the laminar cooling system used in this work is shown in Figure 6-1 and includes the approximate location of the cooling banks and side-sprays used. The system shown in Figure 6-2 has three cooling zones: 1) the radiation zone before the first bank; 2) the water-cooling section; and 3) the radiation zone at the end between the last bank and the coiler. The heat transfer in the water-cooling section is the primary method by which the temperature of the skelp is controlled. The water-cooling section includes six water banks and six side-sprays located after each of the water banks. Side-sprays remove water from the surface of the skelp and help reduce the amount of heat removed from the skelp. Manipulated variables in the model are the skelp velocity, the water banks' conditions (on or off), and the conditions of the side-sprays (on or off). The aim of this work is to determine the banks' conditions, the side-sprays' conditions, and the skelp velocity in order to obtain specific values for skelp temperature (i.e. cooling rate and coiling temperatures) during transit on the run-out table (Binesh B., 2011).

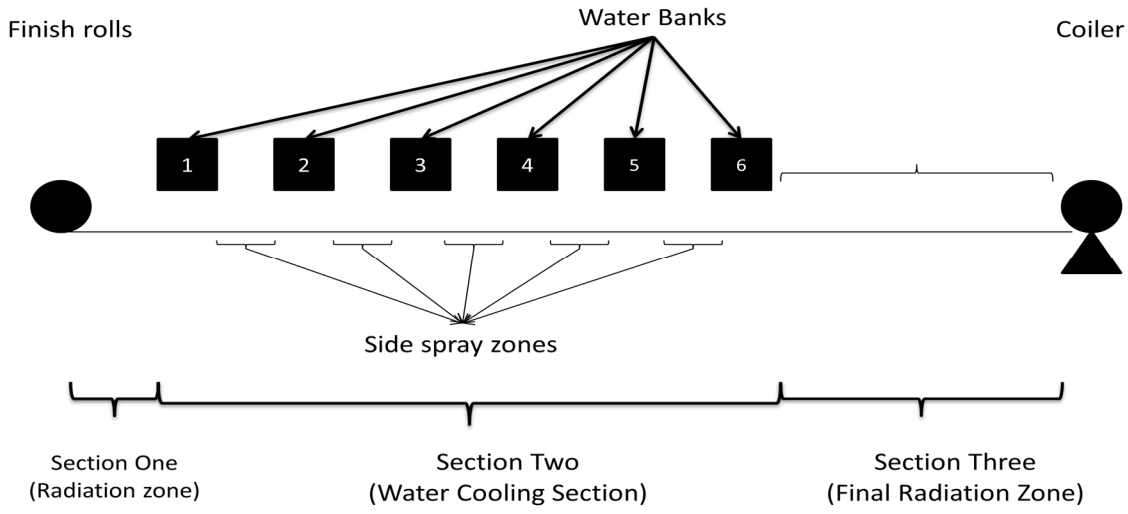

Figure 6-1 Schematic of the laminar cooling system (Binesh B., 2011)

6.1.2 Laminar cooling optimization

In order to control the simulated temperature on the run-out table, it is necessary that the simulated plant (FEM model) is consistent with real plant data. Tuning the model parameters was the goal of some previous works (Xie H. B., 2006) (Peng L., 2008) (Chatterje S., 2001), (Guo R.M., 1991). In this work, an FEM model of the laminar cooling system, which was developed and validated by Wiskel et al. (Wiskel J.B., 2011), was used as the model for the cooling process.

Research on the control of steel skelp temperature started in the 1960s (Guo R.M., 1997). The main objective in a number of preceding works was to manipulate the cooling variables to result in a specific coiling temperature, due to its importance in the final steel microstructure (Xie H. B., 2006) (Peng L., 2008)(Biswas S.k., 1997). However, some works were multi-objective and considered not only the coiling temperature but also a middle target temperature or cooling rate (temperature profile) during the cooling process (Xie H.B., 2007), (Latzel S., 2001). In previous researches, offline optimizations were done in order to obtain a control policy (i.e. laminar configuration) that would achieve the desired coiling temperature (Samaras N.S., 2001)(Biswas S.k., 1997). In this work, an offline

control strategy is proposed for the simulated cooling process. The objective of these optimizations was to control the coiling temperature with respect to specified coiling temperature while maximizing the strip velocity and controlling the temperature profile with respect to uniform and non-uniform cooling rates.

6.1.3 Finite element thermal model

There are about 500 nodes in the strip, according to the mesh definition in the FEM model. In Figure 6-2, all nodes are shown in the strip. Node B (at about ¼ the depth) is the representative of all nodes for the first optimization (i.e. the coiling temperature with different cooling strategies). However, temperatures for all nodes are controlled in the second section of this work in order to achieve temperature for the maximum strip volume in the desired zone.

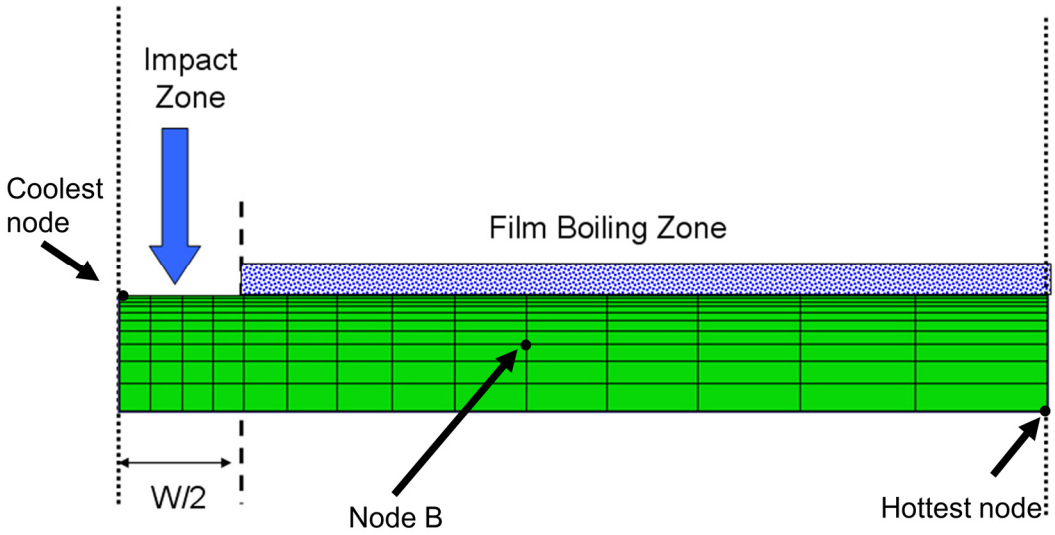

Figure 6-2. Schematic of FEM mesh and control Node location (Binesh B., 2011)

6.1.4 Global optimization methods

Three optimization methods were used to optimize the problem in this paper (Binesh B., 2011), including GA, PSO and BB. GA and PSO are stochastic optimization methods based on the random generation of inputs. GA belongs to a larger class of evolutionary algorithms in which the search method is based on natural evolution. In addition, Heuristic methods such as GA and PSO can be applied to solve stochastic "black box" problems. As long as the evaluation of the objective function is not an expensive process, these heuristic methods can be used as a good method for searching between possibilities. The GA method also has disadvantages, namely 1) the performance is not competitive for some problems, 2) representation of the weights is difficult, and 3) the genetic operators have to be carefully chosen and developed. PSO is a notable algorithm for several reasons: its formulation is remarkably straightforward to implement, apply and extend, meaning there are few parameters to tune for various applications; in addition, it is an advanced source of emergent changes, which are the essence of swarm intelligence (Younis A., 2010).

In 1960, A.H. Land and A.G. Doig developed the branch-and-bound (BB) method for linear programming. BB is a general algorithm for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization. It consists of a systematic enumeration of all candidate solutions, where large subsets of unsuccessful candidates are discarded by using the upper- and lower-estimate bounds of the quantity being optimized. The use of bounds for the function to be optimized, combined with the value of the current best solution, enables the algorithm to search various parts of the solution space (Clausen J., 1999).

For the cooling system considered in this work, the number of water banks and side-sprays was established at six. Each bank and side-spray can be set to either off (a value of 0) or on (a value of 1). The skelp velocity is digitized using P

binary digits by dividing the range of allowable speeds into 2^P equally spaced values. In this work, the value of P was established at 3, which provides eight discrete speed values. This value was chosen because the speed range used in this work (3 to 5m/s) was sufficiently small so that results could be obtained to within 0.25 m/s.

6.1.5 Cooling strategy

Three cooling strategies can be used for the overall scheme of temperature profile during laminar cooling: early cooling, late cooling and uniform cooling. Figure 6-3 is the sketch of cooling strategies in a CCT diagram. There is no control on the first and third sections of ROT (see Figure 6-3), which means that the water cooling section is the only part to control the temperature profile of the strip; therefore, the final temperature of water cooling section determines the coiling temperature on the ROT. A specified cooling rate leads to a definite coiling temperature for each velocity (i.e. no control on the radiation zone in Figure 6-1); therefore, the cooling rate control embraces the coiling temperature control as well. Using different cooling strategies, different phases can be achieved during the cooling process for the same coiling temperature. In Figure 6-3, the phases for the steel strip use three cooling strategies, resulting in a coiling temperature of 550°C, are shown. For example, by using early cooling, the steel goes from austenite to bainite directly. However, by using a late-cooling strategy, the steel goes to a Ferritic phase on its way from austenite to bainite.

It is important to note that there is little control on the first and third sections because there are no banks and side-sprays in those areas. However, the velocity of the strip can change the time of heat transfer in each section, which affects the temperature profile in that section.

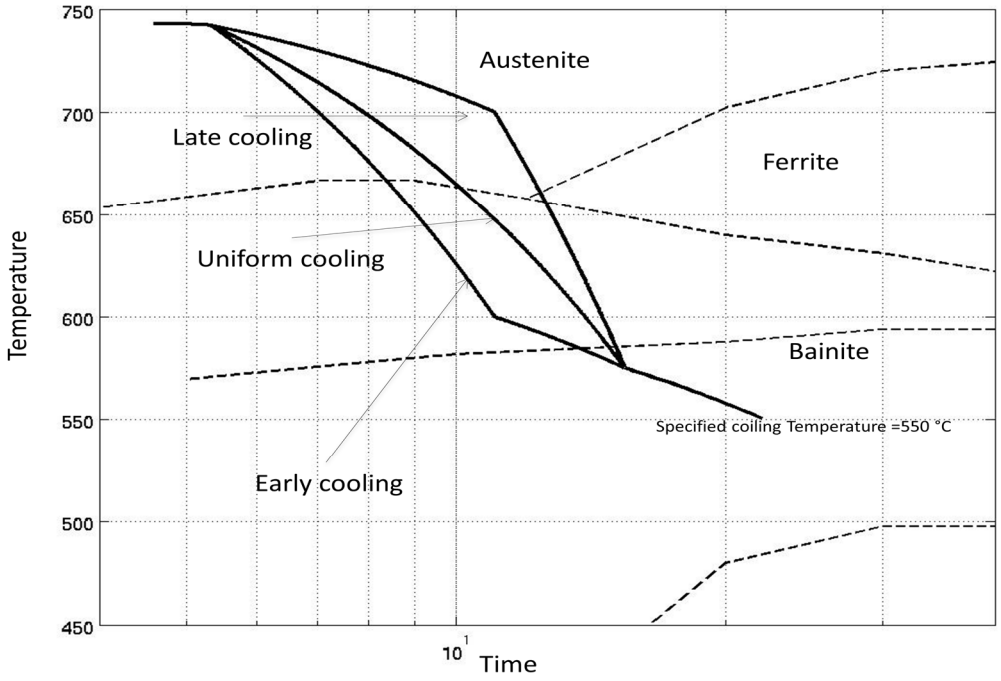

Figure 6-3. Sketch of cooling strategies: early cooling, late cooling and uniform cooling

6.1.6 Optimization Methods

In order to compute an optimal configuration for the cooling system, three optimization methods were compared, with respect to error and total number of simulations for the optimization algorithms. Two of these were population-based evolutionary optimization methods, while the third was branch-and-bound. The comparisons verified that the BB method is faster than heuristic methods, and, moreover, that it can achieve the global optimum between all possibilities. Therefore, in this work, BB is used to optimize the configuration of run-out table for the desired objectives.

6.1.6.1 Branch-and-bound

Unlike evolutionary optimizations, branch-and-bound as a deterministic optimization method is a search method that can definitively find the global best solution for the problem. In this technique, the feasible set is relaxed and subsequently split into parts (branching) over which lower (and often also upper) bounds of the objective function values can be determined (bounding). BB can be applied to various types of problems with a tree structure in possibilities of manipulated variables.

As described above, the control values include six water-bank and side-spray conditions, as well as the strip velocity. Each bank and its associated side-spray are considered together in control actions. If one bank is "off", then there is no need to turn on its side-spray, because there is no water on the surface of the skelp to remove. This means that all possibilities where side-spray is "on" and its bank is "off" will be rejected from the control actions. Therefore, there are three possibilities for each bank and corresponding side-spray: "off-off", "on-off", and "on-on" (The first term being the bank condition and the second the side-spray condition).

Branch-and-bound optimizes the configuration of the run-out table for each velocity and then compares results for different velocities to obtain the best solution. Considering banks and side-sprays together, the total control actions are 3^6=729 for each velocity. Considering 2^3 possibilities for velocities, the total number of possible configurations for this work is therefore $3^6 \times 2^3 = 5832$ configurations. Each optimization starts from highest velocity (4.75 m/s) and finishes with lowest velocity (3 m/s); therefore, configurations with higher velocities are chosen as optimal solutions and are considered as the limits for the next velocities. This trend, to search from the highest velocity to the lowest, is advantageous, because the configurations with lower velocities can only beat the configurations with higher velocities if they have a better cost-value compared to

higher-velocity competitors. However, for some objectives there is no need to do all this simulation until the last velocity. For example, the stopping criterion for coiling temperature optimization is any error (i.e. difference between simulated and desired coiling temperature) lower than 5°C. This means the process stops when the configuration with the highest possible velocity with an error lower than 5°C is found.

In Figure 6-4, the variations of all control actions for one velocity are illustrated. The structure of all possibilities for one velocity in Figure 6-4 resembles a tree, with the root (the start of the cooling process) on the left, and the leaves (the run-out table configurations) on the right. The leaf nodes represent the actual enumerated run-out table configurations, so there are 3^6 of possibilities for each velocity. For any two directly connected nodes in the structure, the parent node is the one closer to the root, and the child node is the one closer to the leaves (Chinneck J. W., 2010). Each step represents the possibilities for each bank and associate side-spray. So, nodes in one step represent a partial run-out table configuration from the first bank and associate side-spray to the bank and associate side-spray of that step. For example, a node in step 4 shows a run-out table configuration determined for the first three banks and their associate side-sprays, but the last three banks and side-sprays conditions are not indicated.

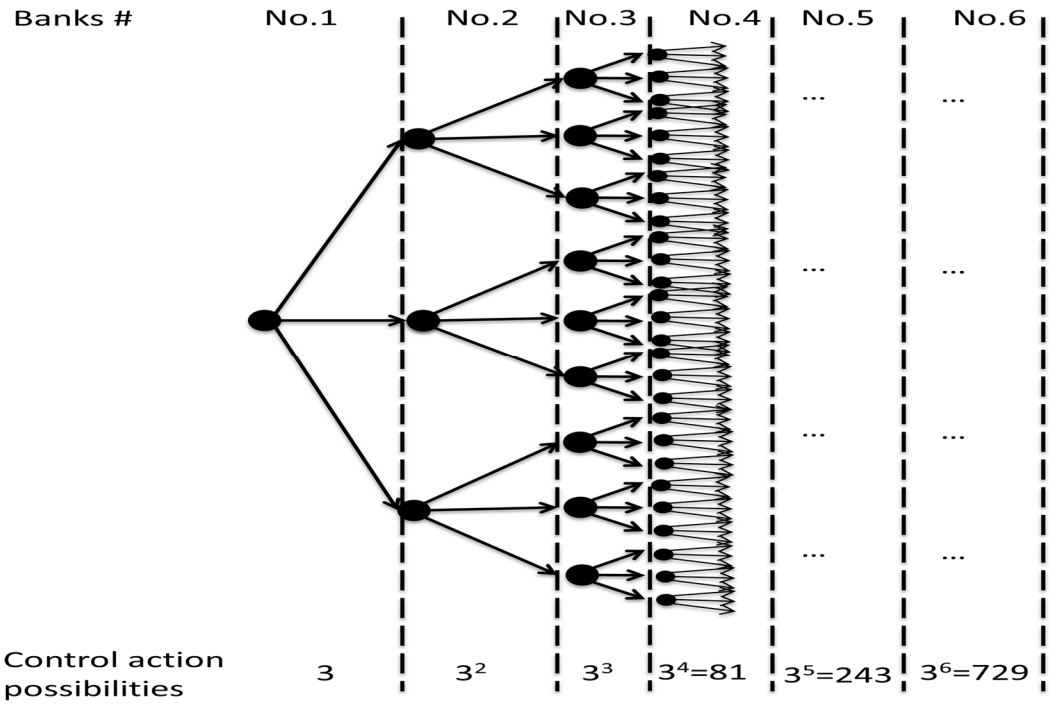

Figure 6-4. The scheme of all possible routes

The main idea in branch-and-bound is to avoid growing the tree as much as possible, because the entire tree would be too big for the specific optimization problem. Instead, branch-and-bound grows the tree in stages, and grows only the most promising nodes at any stage. Other nodes, which will never be optimal, are cut off and permanently discarded.

In Figure 6-5, an example of accepted and rejected control actions are shown. By rejecting a control action at each bank, the descendants from that control action are cancelled and the number of simulations are reduced (e.g. by cancelation of one control action in the first bank, 1/3 of all possibilities were cancelled and only 2/3 remained). At the end, only 45 simulations (3+6+9+12+9+6=45) were done to evaluate all possibilities for one velocity; however, the entire possibilities are 729 configurations for one velocity. In addition, only four complete run-out table configurations could pass the limits.

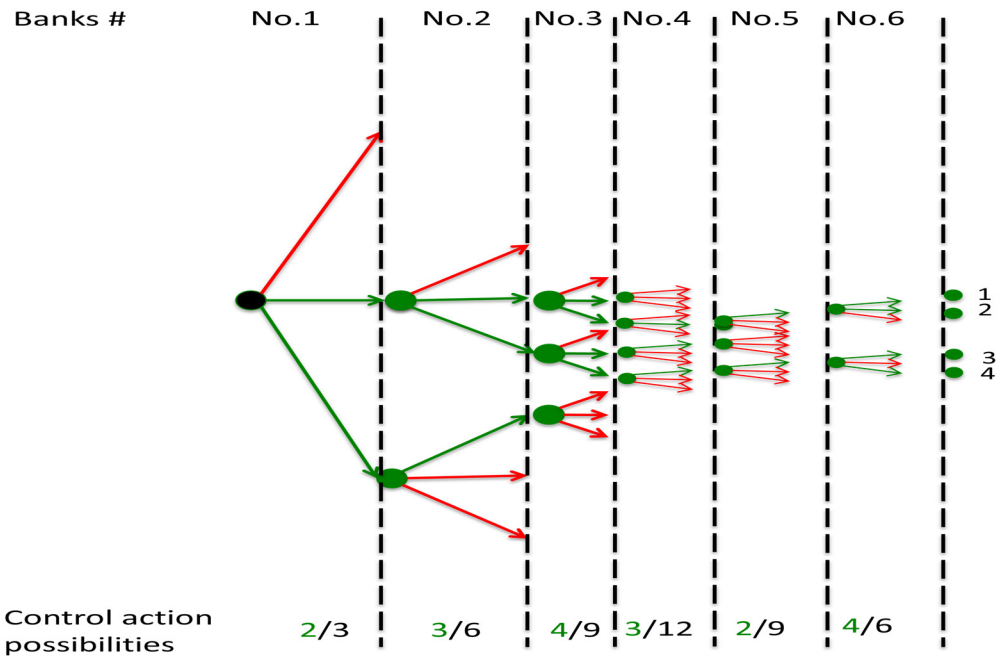

Figure 6-5. The acceptable routes during the search process for a constant velocity

An advantage of BB is that during the iteration process, one can usually delete certain subsets of total possibilities, since one knows that the optimal solution cannot be attained there. A typical disadvantage is that, as a rule, the accuracy of an approximate solution can only be measured by the difference in the upper and lower limits of the current bounds. Hence a reasonably "good" point found early may be deleted only much later, after many further refinements (Horst R., 1993).

6.1.6.1.1 Bounds

In order to reduce the number of evaluations in each optimization, bounds are defined for the objective functions. Upper and lower bounds during each stage authenticate or reject control actions to continue to the next stage. The limitations are different for cooling rate and coiling temperature optimizations.

6.1.6.1.2 Bounds for cooling rate optimization

The first bounds for cooling rate optimization are the upper and lower temperature limits. In order to have a minimum error between simulated and targeted cooling rates during the cooling process, temperature bounds are applied on the temperature profiles. These constraints control the temperature variation from the desired temperature profile. The limit for all optimizations is ±10°C. This means if the temperature profile of a configuration goes out of the determined space (between upper and lower temperature limits) at any point, the configuration and its descendants will be removed from the possibilities of the optimization. A sample of an acceptable temperature profile and its confidence interval are shown in Figure 6-6.

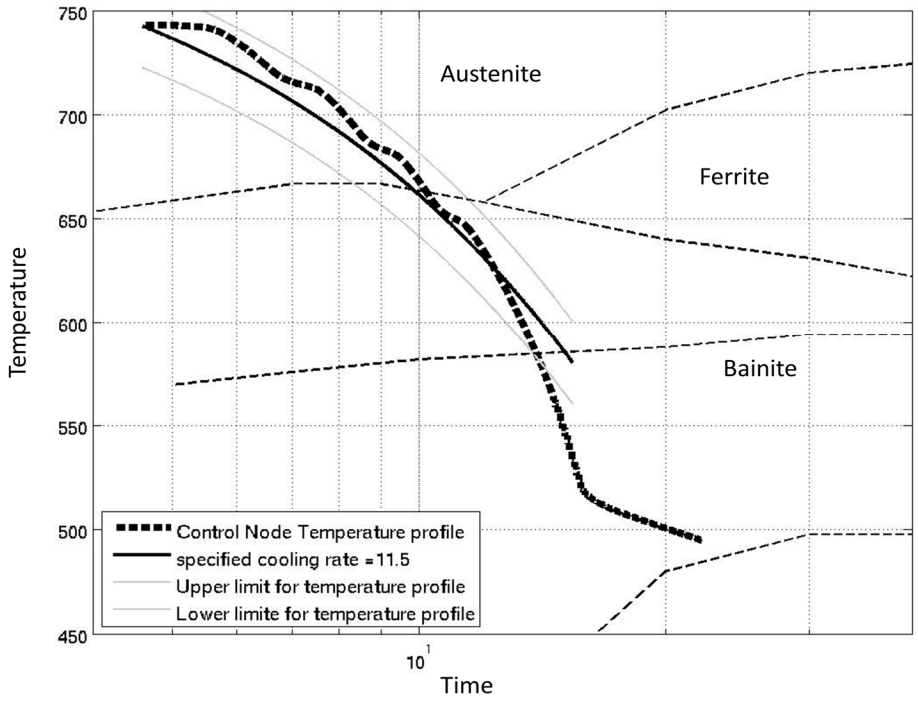

Figure 6-6. Temperature profile of control Node compared with specific cooling rate

The second bound in cooling rate optimization is the limit for the cost values. The cost value of a configuration with respect to a desired cooling rate is the sum of all cost values for all six sections (i.e. six banks and associate side-sprays). So, if the cost value of a configuration after three or four banks is higher than the limit for cost value, it will be rejected from the possibilities, and no branches beyond that node will be considered. At the start of the optimization process (possibilities for first velocity), there is no limit for cost value, but after the first velocity (i.e. 4.75 m/s), the limit becomes the minimum cost value between all configurations of the first velocity (i.e. optimal configuration). Thus the first bound ensures that the optimization algorithm gives acceptable results, while the second bound makes the optimization algorithm more efficient.

6.1.6.1.3 Bounds for coiling temperature control

The coiling temperature control has fewer constraints compared to the cooling rate, because there are no characteristics in the middle of the cooling process that are strongly related to the final temperature. In order to reject results that obviously cannot meet the specified coiling temperature, a confidence interval for a uniform cooling rate that ends at a specified coiling temperature is defined. The confidence interval rejects the control actions that are far from the specified cooling rate.

6.1.6.1.4 Stopping criteria

If the coiling temperature control is the objective of optimization, the stopping criterion is meant to achieve cost function values in the range of ±5°C. However, the cooling rate control has no stopping criteria, and the search will continue until the last velocity to investigate feasible and optimal possibilities.

6.2 Optimization Results

The goals of this work are: 1) controlling the coiling temperature using different cooling strategies (either uniform or non-uniform) during laminar cooling, and 2) maximizing the volume of the strips that have a temperature profile within a desired area in the CCT diagram. The optimization methods of GA, PSO and BB were compared, with respect to the accuracy of the final solution as well as time spent on the optimization search. BB, as the best optimization method, was used to control the coiling temperature at 550°C using different cooling strategies, including early cooling, uniform cooling and late cooling.

The goal of optimization is to minimize the cost function. Therefore, before proceeding with a comparison of optimization methods, the cost functions used for each objective are presented. The general fitness (objective) function for the optimization is:

$$J = A\left|T_f - T_f^S\right| - B(Speed) + C\sum_{i=1}^{i=n^*}\left|NT(i) - NT(i)^S\right| \quad (1)$$

Where the equation parameters are defined in Table 6-1.

Table 6-1. Cost function parameters

Parameter	Definition/values
A	Coefficient used to penalize deviation from the desired coiling temperature.
B	Coefficient used to penalize slow speed values.
C	Coefficient used to penalize deviation from the temperature profile (e.g., cooling rate).
Speed	Velocity of the skelp

T_f^d	Specified coiling temperature.
T_f	Coiling temperature.
NT(i)	Temperature at node *i.*
NT(i)^d	Specified temperature at node *i.*
n^*	The number of nodes in the water-cooling section.

The optimization algorithm minimizes the objective function in Equation 1. In addition, it must also fulfill process constraints: first, all nodes must have a minimum temperature of 450°C during the cooling process to ensure that the skelp is not too difficult to coil. Second, the velocity of the skelp is assumed to be between 3 and 5 m/s.

6.2.1 Coiling temperature optimization using three cooling strategies

The aim of this section is to control the coiling temperature using three different cooling strategies. As discussed above, three major cooling strategies are considered in this work: early cooling, late cooling and uniform cooling. A uniform cooling rate guarantees that the steel cooling process is completely smooth on the run-out table. The specified coiling temperature (550°C) and different cooling strategies to obtain this coiling temperature are presented in Figure 6-7. As shown above (Figure 6-2), the CCT diagram for each cooling strategy shows certain steel phases that indicate the product steel's properties.

Using different cooling strategies (see Figure 6-7) shows the ability of the optimization method to achieve temperature profiles (i.e. temperature routes) in a wide range (from starting temperature to coiling temperature) with the same coiling temperature (550°C). In addition, the cooling strategies are simple cases of non-uniform cooling rates, an ongoing research topic in the materials industry.

In this paper, the early and late cooling strategies are defined according to the temperature after the third bank (middle temperature). Therefore, the optimization's cost function was the sum of errors for two temperatures: 1) coiling temperature with respect to 550°C, and 2) temperature at the end of the third bank, with respect to 600°C for early cooling and 700°C for late cooling. The middle temperature for uniform cooling was 647°C. The values for middle and coiling temperatures and the temperature profile for each cooling strategies are illustrated in Figure 6-7.

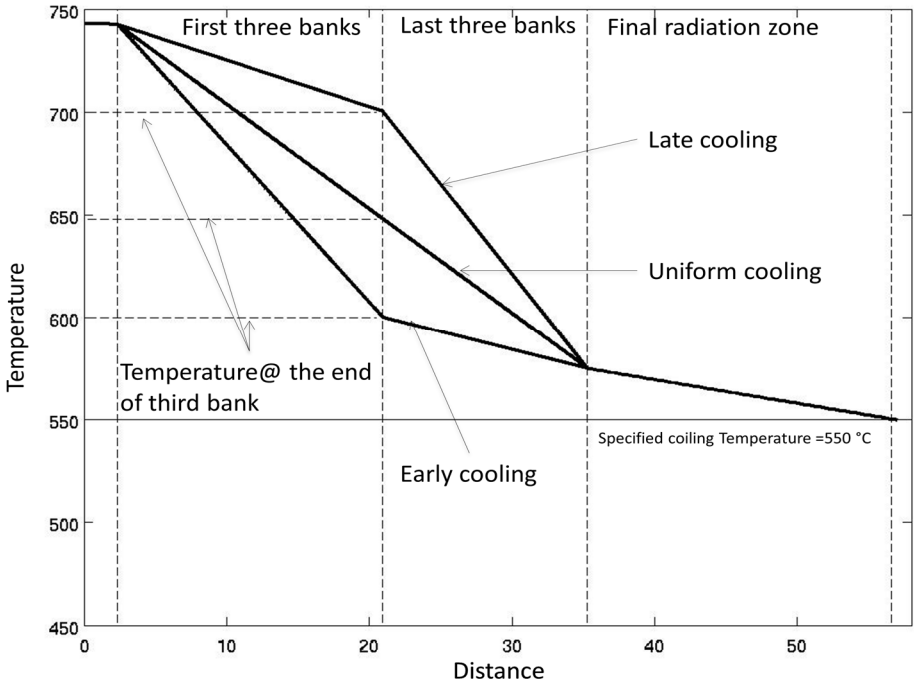

Figure 6-7. Cooling rate strategies and their specified middle and coiling temperatures

BB demonstrated good performance in cooling rate control for both speed and accuracy of results. Therefore, in this part BB was used to control the cooling rates (early, late and uniform cooling), which led to the specified coiling temperature (550°C). The cost value in this section is defined as the sum of middle and coiling temperature errors. In addition, by using a confidence interval

of ±20°C around the specified cooling rate, the temperature profile was restricted close to the desirable cooling rate. The configuration of the run-out table, coiling and middle temperature errors and number of simulations for each optimization by branch-and-bound are presented in Table 6-2.

Table 6-2. Optimization results for early, uniform and late cooling strategies by BB

Cooling rate strategy	Bank configurations	Side-sprays configurations	Velocity of the strip	Middle and Coiling temperature Error (℃)
Early cooling	*111100*	*001100*	*3.5*	*26.7*
Late cooling	*010111*	*010001*	*3*	*2.7*
Uniform cooling	*111110*	*001100*	*4.25*	*0.4*

The optimal bank and side-spray configurations indicate the desirable cooling strategy for each optimization. For example, the optimal configuration for early cooling sees the first three banks "on", which means the cooling is emphasized in the first half of the water-cooling section. However, the optimal solution for late cooling sees the last three banks "on", which emphasizes the cooling in the second half of the water-cooling section. The temperature profile of the optimal configuration, the specified cooling rate and coiling temperature, and the confidence interval for the desired cooling rate for each optimization are all illustrated in Figures 6-8 to 6-10.

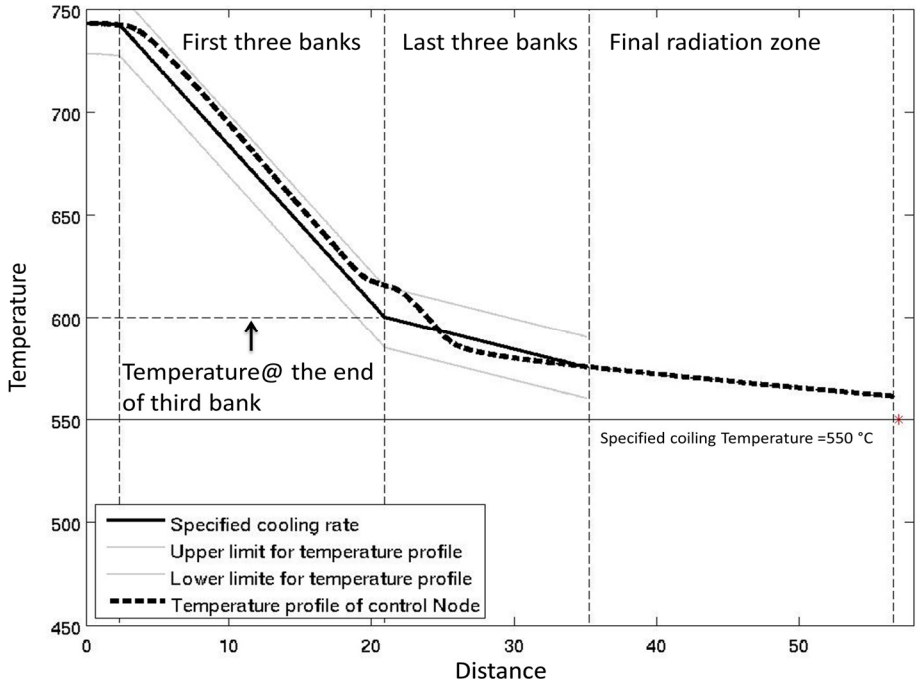

Figure 6-8. Temperature profile of control Node for early cooling strategy

(Objective: coiling temperature = 550°C for control Node)

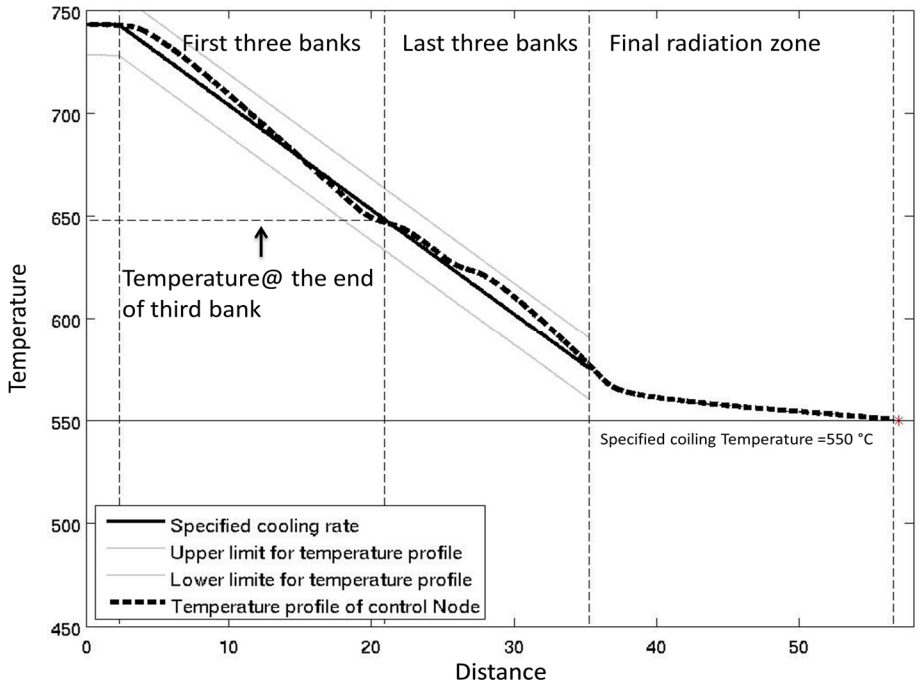

Figure 6-9. Temperature profile of control Node for uniform cooling strategy

(Objective: coiling temperature = 550°C for control Node)

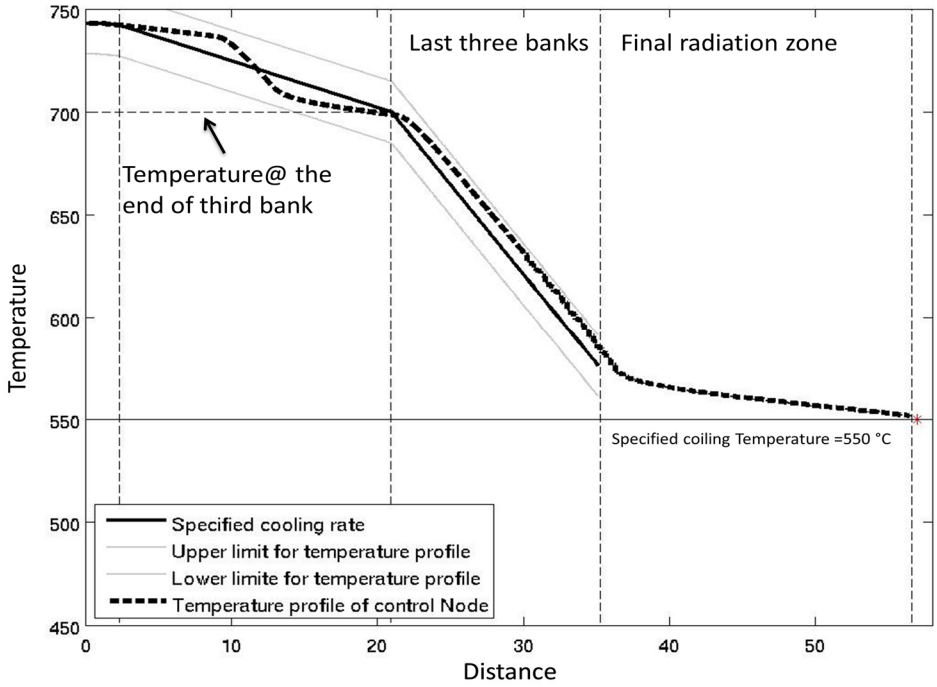

Figure 6-10. Temperature profile of control Node for late cooling strategy

(Objective: coiling temperature = 550°C for control Node)

The first constraint in each optimization was that the temperature profile should be within the confidence interval (within 15°C of specified cooling rate). The best result was selected between the remaining configurations, which were within the limited area. The final result had the best cost value (i.e. the sum of errors for coiling and middle temperatures) between all acceptable results.

The break in cooling rate during the water-cooling section enabled the temperature profile to reach the desirable temperature in the middle of the cooling process. Nevertheless, the tracking of these cooling rates was harder in comparison to uniform cooling, because there were a low number of possibilities that could satisfy that constraint.

In the first case (early cooling), the optimal solution could not achieve a low error for both middle and coiling temperature (sum of error= 27°C). In spite of that, the cooling rate was in the restricted area for cooling rate. The BB method found the best possible solution for this case; therefore it needs to have more control actions to control the temperature profile, and it is not related to the optimization method. In the late and uniform cooling cases, the optimal solution's middle and the coiling temperatures' errors were 2.7 and 0.4, respectively (which are acceptable), and the temperature profiles could satisfy the limits completely.

6.2.2 Run-out table optimization to set the temperature of maximum volume of the strip in a specified area in CCT diagram

The main goal of temperature control on ROT is to achieve the desired properties for the steel product, which pertains to the temperature profile (the phases in CCT diagram). Temperature control for the entire strip is the objective of this section. The optimum configuration controls the temperature profiles of the entire volume of the strip to be in the specified area, which is illustrated in Figure 6-11. The specified zone is in the middle of three phases (i.e. Austenite, Ferrite and Bainite). Controlling the temperature in this area, the optimal solution can guarantee that the steel phase doesn't change directly from Austenite to Bainite. In this section the temperatures of all nodes in the strip are controlled during the cooling process, while in the previous optimizations the temperature of only a single node (Node B) was controlled.

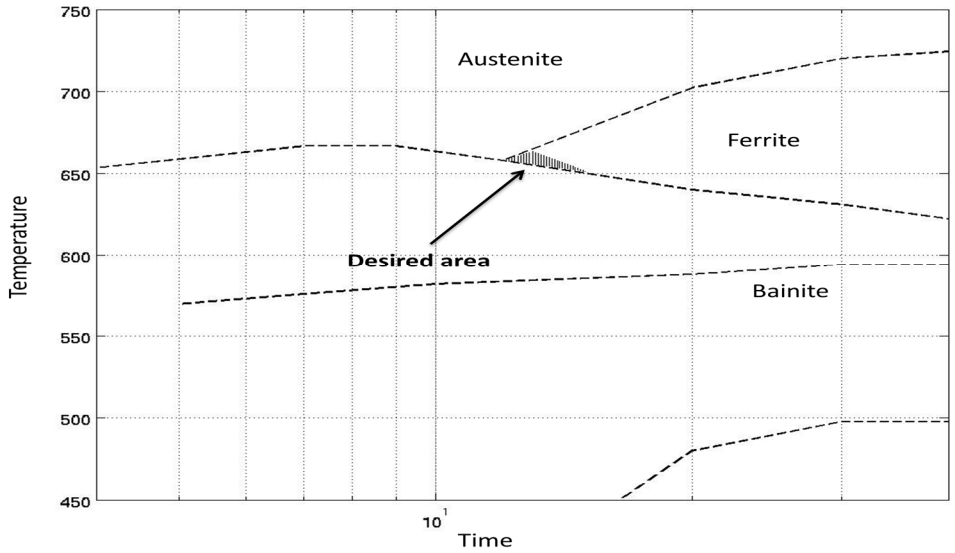

Figure 6-11. Specified zone in CCT diagram

The cost function evaluates the deviation of temperature profile from the specified zone. The evaluation is based on the percent volume of the strip whose temperature is in the specified zone. Finding which nodes have the suitable temperature profile indicates the volume percent of the strip whose temperature is in the desired condition. Another constraint is that the coiling temperature must be between 500°C and 600°C for every node. Finally, each node's temperature must be higher than 450°C at all times during the cooling process.

Maximum volumes of the strip, whose temperature profiles are in the specified zone in CCT diagram, for each velocity, are presented in figure 6-12. By increasing the velocity of the strip, it is harder to control the temperature profile to meet the desired area and the coiling temperature constraints (i.e. between 500°C and 600°C). At higher velocities, the time required to decrease the temperature after the specified area is not enough to reach the desired coiling temperature (i.e. between 500°C and 600°C). However, higher velocities can speed up the production process, which can lead to higher productivity and other benefits for industry.

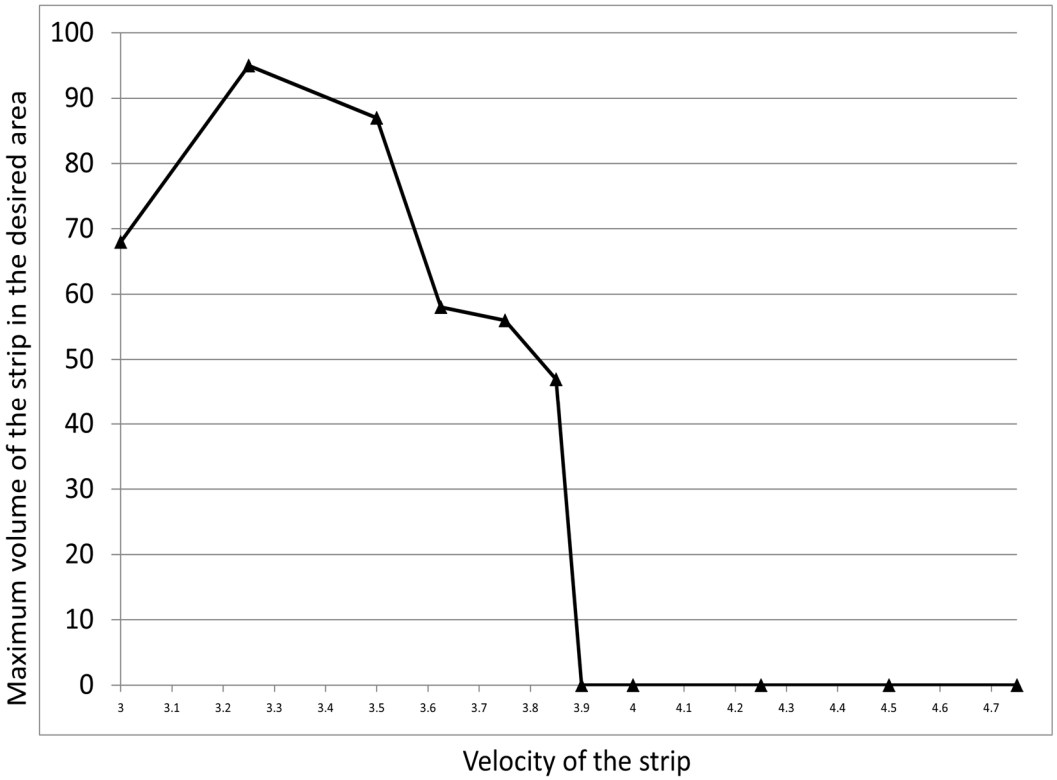

Figure 6-12. Maximum percent volume of the strip whose temperature is in the desired zone for each velocity

The optimization results are summarized in Table 6-3. All optimizations were done using all nodes on the strip (see Figure 6-2 for Nodes locations) as control points. The optimal cooling system configuration, including the optimal number of banks, side-sprays and skelp velocities, are also listed in Table 6-3.

Table 6-3. Optimization Description and Results

Run No	*Velocity (m/s)*	*Optimal Banks Configuration*	*Optimal Side-sprays Configuration*	*Maximum Volume (%)*
1	*3*	*100101*	*100001*	*68*
2	*3.25*	*110010*	*110000*	*95*
3	*3.5*	*011010*	*011000*	*87*
4	*3.625*	*100010*	*000000*	*58*
5	*3.75*	*100010*	*000000*	*56*
6	*3.85*	*001110*	*001000*	*47*
7	*3.9↑*	*N.A.*	*N.A.*	*0*

1="ON", 0="OFF"

If the velocity is set to 3.25 m/s, the temperature of 95% of the entire volume of the strip can be controlled to be in the desired zone and in the specified coiling temperature range. On the other hand, it is not possible to keep any nodes in the specified zone for velocities higher than 3.9 m/s. These factors can have an important influence on the decision between higher production volume or higher quality.

In velocities more than 3.9 m/s, the specified zone is in the place of banks 5 and 6 (the last parts of water cooling section in Figure 6-1); therefore, there is no water to decrease the temperature of the strip after the specified zone, and the strip goes to the radiation zone (Figure 6-1). This lack of coolant prevents the strip from reaching the specified coiling temperature.

In next Figure, the temperature profiles of the hottest node, coolest node, and node B of the optimal configuration for run 2 (velocity = 3.25 m/s) are shown in CCT diagram. Temperatures of all nodes are between those hottest and coolest nodes (Figure 6-2) during the water-cooling process. So, the temperature profiles show that most of the strip volume has the temperature profile in the specified zone.

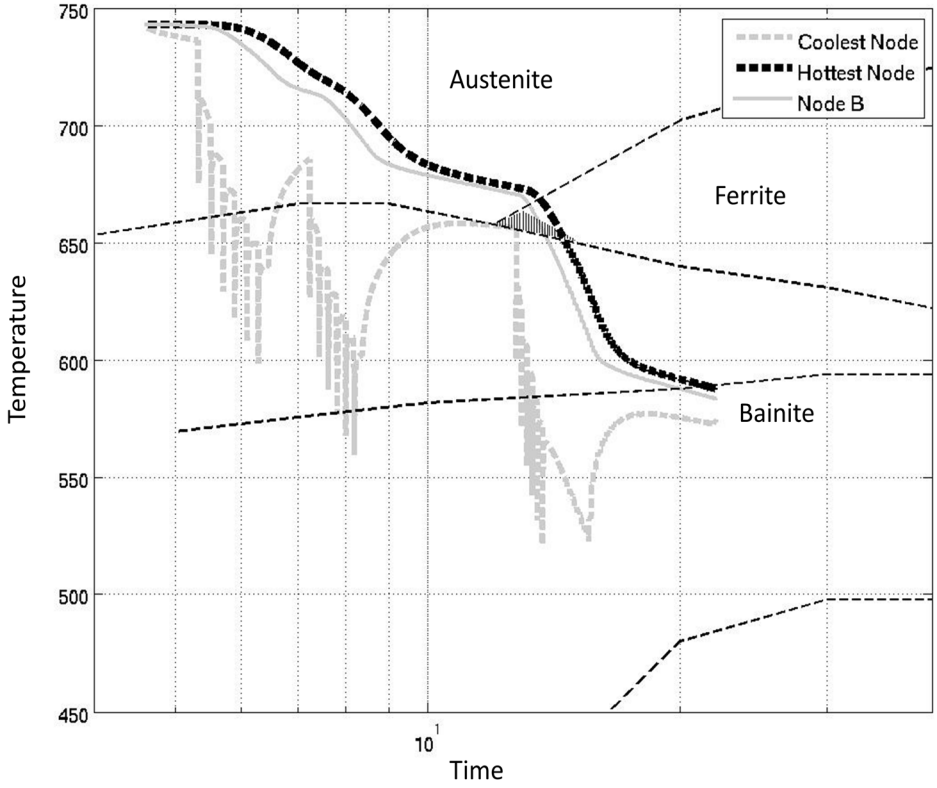

Figure 6-13. Temperature profiles of hottest node, coolest node and node B (Objective: maximum volume in desired zone for velocity equal to 3.25 m/s)

The optimal temperature profiles of the hottest node, coolest node and node B of the optimal solution for velocity equal to 3.75 m/s are shown in Figure 6-14. Also, the volume of the strip whose temperature profile is in desired area is shown in Figure 6-15. The black points are nodes, which have the desired temperature in the specified zone. Conversely, the white points could not meet the specified zone in Figure 6-11. The green part of the strip is the part controlled by BB in the specified zone, which comprises 56% of the total volume of the strip.

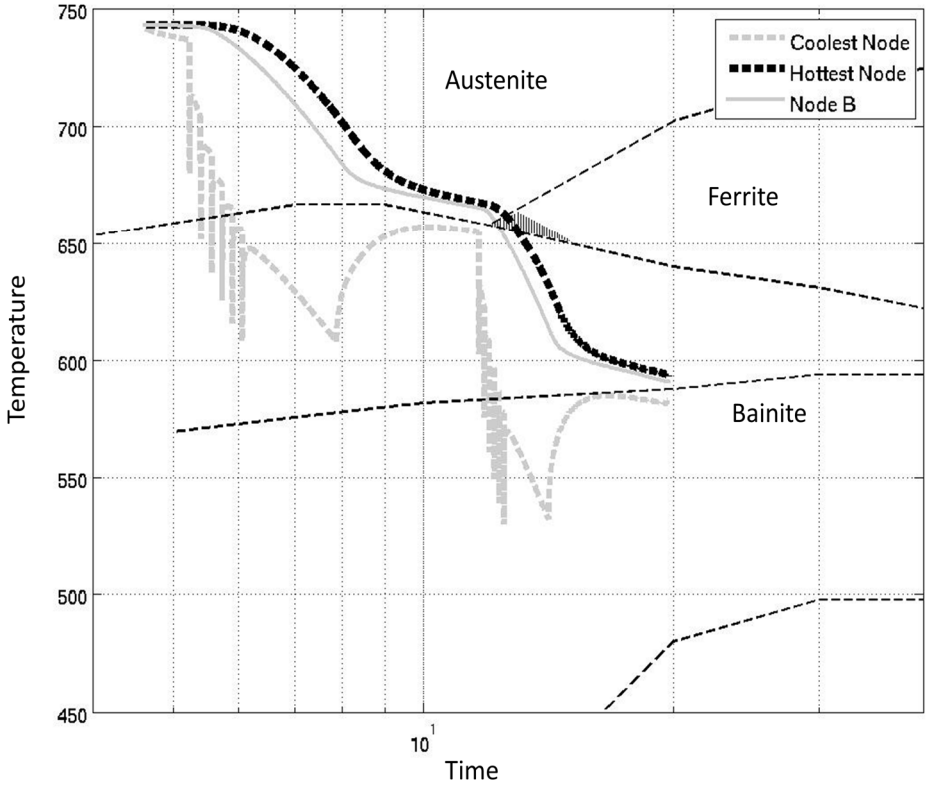

Figure 6-14. Temperature profiles of hottest node, coolest node and node B (Objective: maximum volume in desired zone for velocity equal to 3.75 m/s

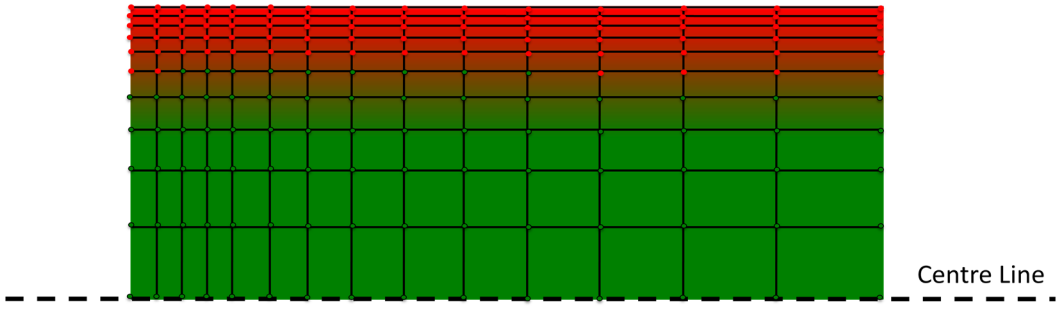

Figure 6-15. The volume of the strip in the desired condition for velocity equal to 3.75 m/s

6.3 Discussion

There was an FEM thermal model in this work to evaluate the cost of control actions on temperature profile during laminar cooling. Therefore, there was no mathematical equation to model the cooling section in order to use deterministic optimization methods, and the problem was considered a "black box" optimization. Therefore, the black-box optimization methods like GA and PSO can be applied to this problem.

There were some characteristics in the cooling process that could reduce the number of possibilities in the cooling process (e.g. considering each bank and its side-spray together in control actions). In addition, the problem has two attributes of overlapping sub-problems and optimal substructure, which means it is possible to determine the optimal solution for each bank independent of the other banks' configuration. Therefore, it is possible to use BB to control the temperature profile on the run-out table.

Different cooling strategies were used in order to show the ability of BB to control the cooling rate on the run-out table. The results had acceptable errors and were within the confidence interval of the desired cooling rate. The early cooling rate means the steel phase is changed from austenite to bainite directly. However, late cooling guarantees the ferrite phase between austenite and bainite during laminar cooling. In Figure 6-7, the phases of late, early and uniform cooling are determined in a CCT diagram.

The temperature control for all nodes of the strip shows the ability of the optimizer to control the temperature of the entire strip within a defined area. In higher velocities there is not enough time to decrease the temperature in the water-cooling section in order to reach the coiling temperature constraints. Therefore, adding more banks at the end of the water-cooling section makes it

possible to obtain a higher volume of the strip in the desired conditions for higher velocities.

To alleviate large deviations in cooling rate along the length of the laminar cooling system, it may be more practical to use a control strategy that employs a multitude of cooling rates. However, the effect of a non-uniform cooling rate on transformation and microstructure is unknown – further metallurgical work delineating the relationship between non-uniform cooling and microstructure is needed.

6.4 Summary and Conclusions

Stochastic optimization techniques such as GA and PSO were efficient in finding the optimal solution with reasonable accuracy (i.e. errors lower than 10% for each optimization) and within an acceptable time frame (i.e. number of simulations) for coiling temperature and cooling rate as is presented in Table 6-4. However, they could not guarantee obtaining the global minimum solution.

The branch-and-bound method was used to achieve the global minimum in this problem (Table 6-4). In addition to obtaining the global minimum, BB was able to get the final solution faster than PSO and GA. The comparison between the accuracy of the optimal solution, convergence time and total number of simulations (Figure 6-6) for three methods underscored the ability of the branch-and-bound.

Three different cooling strategies were used to reach the desirable coiling temperature of 550ºC. Dynamic programming showed good accuracy in control of both the coiling temperature and cooling rate combined. Therefore, it can safely be used in non-uniform cooling rate control.

Finally, BB led to the optimal configuration of ROT to achieve the maximum volume of the strip in the desired zone (Figure 6-11). The comparison between the maximum volume for different velocities shows that in higher velocities it is harder to control the temperature profile. However, in the lower velocities it is possible to control nearly all nodes of the strip. Ultimately, it is a trade-off between faster production and higher strip volume in the desired condition.

References

(Avilla V.H., 1994) Avilla V.H. *"Modeling of the thermal evolution of steel strip cooled in the hot rolling run out table"*, PhD thesis, university of British Columbia. 1994.

(Binesh B., 2011) Binesh B., Wiskel J.B., Ben-Zvi A. Henein H., *"Skelp temperature profile control during laminar cooling using Genetic Algorithms"*, 2011.

(Biswas S.k., 1997) Biswas S.k. Chen S.J., Satyanarayana A. *"Optimal temperature tracking for accelerated cooling processes in hot rolling of steel"*, Dynamics and Controls. Vol. 7, 1997, pp. 327–340.

(Chatterje S., 2001) Chatterje S. Simonelli G., Chizeck H. *"Parameter identification of a nonlinear coiling temperature model for run-out table control at LTV Cleveland works 84-In hot strip mill"*, Case Western Reserve University. 2001.

(Fatih M., 2003) Fatih M. Liang Y. *"A binary particle swarm optimization algorithm for lot sizing"*, Journal of economic and social research. No 2 : Vol. 5, 2003, pp. 1-20.

(Guo R.M., 1997)Guo R.M. *"Modeling and simulations of run-out table cooling control using feed-forward-feedback and element tracking system"*, IEEE Transactions on Industry Applications. Vol. 33, 1997, p. 304.

(Guo R.M., 1991) Guo R. M. *"Development, Identification and Application of a nonlinear Model for Non-symmetric Runout Table Cooling"*, American Institute of Steel Engineers Annual Conference Record. Pittsburgh, PA. 1991.

(Hassan R., 2006) Hassan R. Cohanim B., de Weck O., *"A Comparison of particle swarm optimization and genetic Algorithms"*, - Colorado Springs : Vanderplaats Research and Development, Inc., 2006.

(Holland J.H., 1975) Holland J.H. *"Adaption in Natural and Artificial Systems"*, Ann Arbor, MI, 1975.

(Hunt P., 2011) Hunt Phil World Steel Production Report [Report]. ISSB Ltd, 2011.

(Kennedy J., 1995) Kennedy J. Eberhart R. *"Particle Swarm Optimization"*, Proceedings of IEEE International Conference on Neural Networks. Vol. IV, 1995, pp. 1942-1948.

(Kennedy J., 1997) Kennedy J. Eberhart R.C. *"A discrete binary version of the particle swarm algorithm"*, IEEE International Conference on Systems, Man, and Cybernetics. 1997.

(Latzel S., 2001) Latzel S. *"Advanced Automation Concept of Run-out Table Skelp Cooling for Hot Skelp and Plate Mills"*, IEEE Transactions on Industry Applications . Vol. 37, 2001, p. 1088.

(Passino, 2005) Passino Kevin M. *"Biomimicry for Optimization, Control, and Automation"*, London, UK : Springer-Verlag, 2005.

(Peng L., 2008) Peng L. Li Q., Zhou Z. *"Cooling hot rolling steel strip using combined tactics"*, Journal of University of Science and Technology Beijing. No 3 : Vol. 15, June 2008. p. 362.

(Samaras N.S., 2001) Samaras N.S. Simaan M.A. *"Novel Control Structure for Runout Table Coiling Temperature Control"*, AISE steel technology. No 6 : Vol. 87, June 2001. pp. 55-59.

(Shi Y., 1998) Shi Y. Eberhart R.C. *"A modified particle swarm optimizer"*, Proceedings of IEEE International Conference on Evolutionary Computation. 1998, pp. 69-73.

(Wiskel J.B., 2011) Wiskel J.B., Deng H.,Jefferies C., Henein H., *"Infrared Thermography of a TMCP Microalloyed steel skelp at the upcoiler and its application in quantifying the laminar jet/skelp interaction"*, ISM. Vol. 38(1), 2011, pp. 35-44.

(Xie H. B., 2006) Xie H. B. Liu X. , Wang G. , Zhang Z., *"Optimization and Model of Laminar Cooling Control Thickness measurement System for Hot Strip Mills"*, JOURNAL OF lRON AND STEEL RESEARCH, INTERNMIONAL. No 13(1) : Vol. 9, 2006. pp. 18-22.

(Xie H.B., 2007) Xie H.B. Jiang Z.Y., Liu X.H., Wang G.D., Zhou T.G., Tieu A.K., "On-line optimization of coiling Temperature control on Run-out table for hot strip mills", Key Engineering Materials. Vols. 340-341, 2007. pp. 701-706.

(Yaniro K., 1991) Yaniro K. Yamasaki M., Furukawa M., et al., *"Development of coiling temperature control system on hot strip mill"*, Technical report. Vol. 24, 1991.

(Younis A., 2010) Younis A. Dong Z. *"Trends, Features, and tests of common and recently introduced global optimization methods"*, Engineering Optimization. No 2 : Vol. 48, August 2010, pp. 691-718.

7

7) Summary, Conclusions and Future work

This chapter contains a review of the all objectives and works presented in this dissertation, as well as a list of ideas for future research and work to be developed.

7.1 Comparison of optimization methods

The objective of optimization is to minimize the cost function. Therefore, before proceeding with the comparison of optimization methods, the cost functions used for each objective are presented. The general fitness (objective) function for the optimization is:

$$J = A\left|T_f - T_f^s\right| - B(Speed) + C\sum_{i=1}^{i=n^*}\left|NT(i) - NT(i)^s\right| \quad (7\text{-}1)$$

Where the equation parameters are defined in Table 7-1.

Table 7-1. . Cost function parameters

Parameter	Definition/values
A	Coefficient used to penalize deviation from the desired coiling temperature.
B	Coefficient used to penalize slow speed values.
C	Coefficient used to penalize deviation from the temperature profile (e.g. cooling rate).
Speed	Velocity of the skelp.
T_f^d	Specified coiling temperature.
T_f	Coiling temperature.
NT(i)	Temperature at node *i.*
NT(i)^d	Specified temperature at node *i.*
n^*	The number of nodes in the water-cooling section.

In addition to minimizing the objective function in Equation 6-1, the optimizer must also meet process constraints. First, all nodes must have a minimum temperature of 450°C to ensure that the skelp is not too difficult to coil. Also, the velocity of the skelp was assumed to be between 3 and 5 m/s.

The important characteristics of stochastic optimization methods are convergence time, accuracy of final results and total time of simulations. The convergence time shows the number of simulations needed to achieve a satisfactory result in each optimization. Further trials are done in order to search for better fitness values to achieve the global minimum in stochastic optimization methods. The accuracy of the results is the error of optimal solutions with respect to desired values. In this work, GA, PSO and BB were applied to control coiling temperature and cooling rate during laminar cooling. These three methods were used to optimize the run-out table configuration, first to manage the coiling temperature at 500°C, and second to set the cooling rate at 11.5°C/s.

Table 7-2 tabulates bank and side-spray configurations, strip velocity, optimization errors (as defined above, as a percentage), convergence time (number of simulations), and the total number of simulations at the end of the optimization process for each objective using all optimization methods (GA, PSO and BB).

The optimization results are summarized in Table 7-2. All optimization was done using Node B as the control node (see Figure 2-8 for Node B location). The optimal cooling system configuration, including the optimal number of banks, side-sprays and skelp velocities, is also shown in Table 7-2. The errors are the differences between the simulated temperatures and desirable ones (They do not include the industrial temperatures). The percent error listed in Table 7-2 is computed using this relation for coiling temperature optimization:

$$Error = T_f - T_f{}^s \tag{7-2}$$

And this relation for cooling rate optimization:

$$Error = \sum_{i=1}^{i=n^*} \left| NT(i) - NT(i)^S \right| \tag{7-3}$$

Table 7-2. Comparison between results of all optimizations

Methods	*Optimal solution characteristics*	*Coiling temp (=500℃)*	*Cooling rate (=11.5℃/s)*
Genetic Algorithms	*Bank configurations*	*110010*	*111001*
	Side-spray configurations	*000000*	*111001*
	Velocity of the strip	*4.5*	*4.75*
	Error (℃)	*6.3*	*3391*
	Convergence time	*220*	*224*
	# of simulations	*520*	*540*
Particle Swarm Optimization	*Bank configurations*	*111110*	*101100*
	Side-spray configurations	*010000*	*101100*
	Velocity of the strip	*3.75*	*4.5*
	Error (℃)	*4.9*	*2278*
	Convergence time	*250*	*200*
	# of simulations	*550*	*600*
Branch-and-bound	*Bank configurations*	*110110*	*110000*
	Side-spray configurations	*000000*	*110000*
	Velocity of the strip	*3.75*	*4.5*
	Error (℃)	*3.1*	*1695*
	Convergence time	*141*	*126*
	# of simulations	*141*	*126*

The comparisons between accuracy, convergence time and the total number of simulations for each optimization of the coiling temperature and cooling rate are presented in Figures 7-1 and 7-2.

The branch-and-bound can provide the fastest result with the lowest error in cooling rate and coiling temperature optimizations. The results for PSO and GA show that, in coiling temperature optimization, the PSO got a lower error for the optimal solution, but it is not as fast as the GA. However, in cooling rate optimization, the PSO is faster with a lower error at the same time.

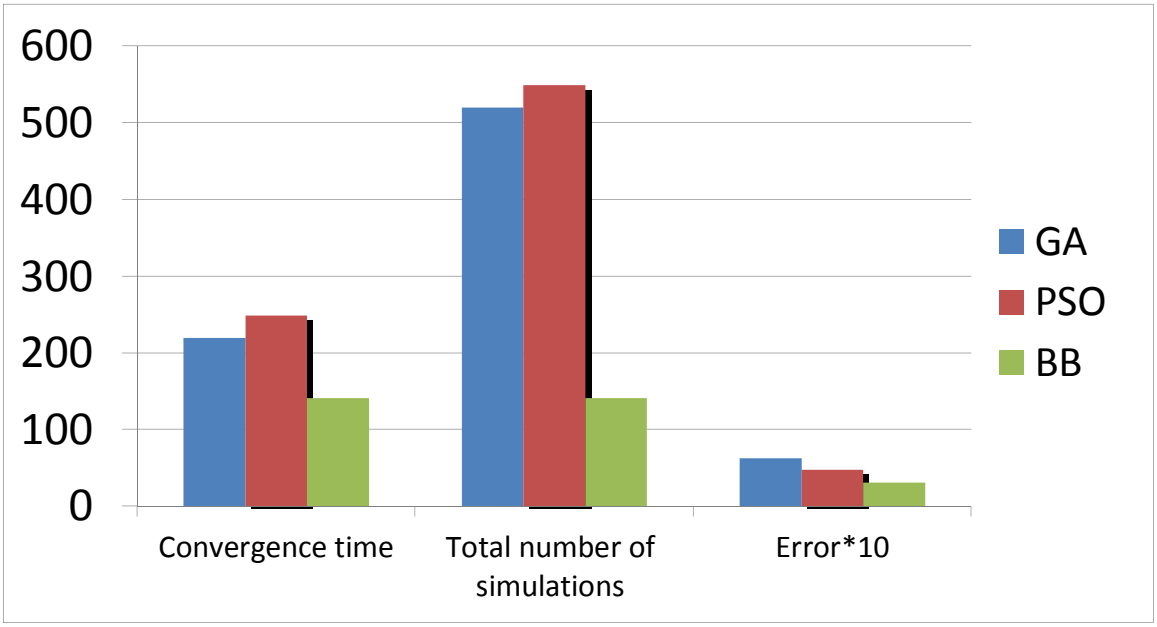

Figure 7-1. Comparison between optimization methods on accuracy, number of simulations and convergence time
(Objective: Coiling temperature = 500°C)

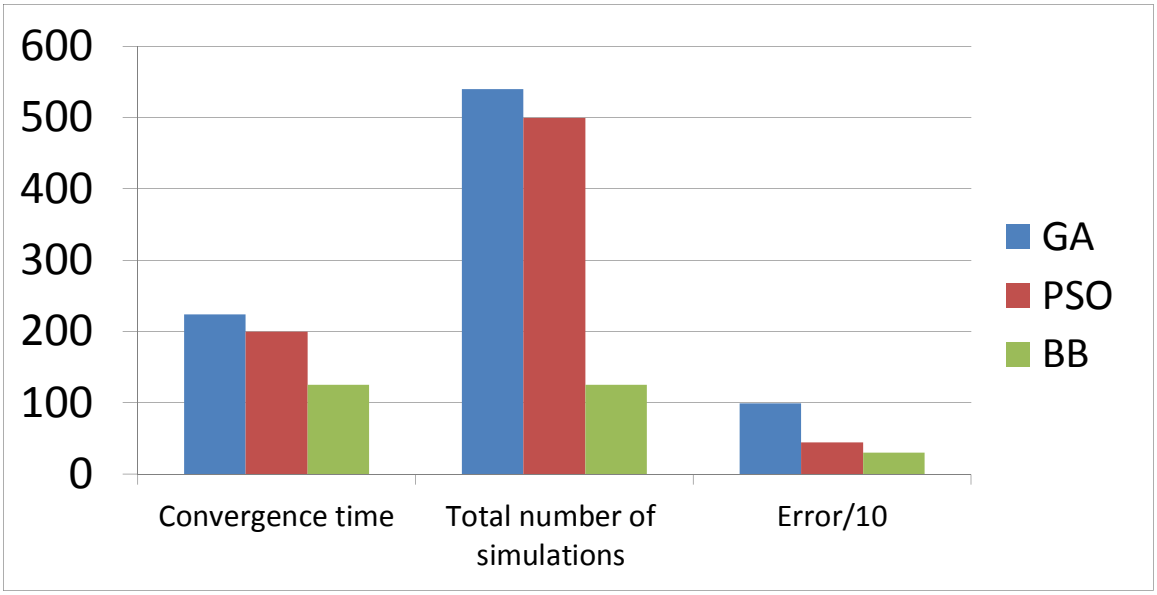

Figure 7-2. Comparison between optimization methods on accuracy, number of simulations and convergence time
(Objective: Cooling rate =11.5°C/s)

7.1.1 Comparison of optimization results on coiling temperature optimization

There are a number of configuration possibilities that lead to a specified coiling temperature of between 500°C and 600°C. The velocity of the strip at the run-out table indicates the time needed to produce the steel strip. Therefore, the cost values are defined accordingly to have the minimum error between simulated coiling temperature, desired coiling temperature, and maximum possible velocity of the strip.

The coefficients of the general cost function in this part are A=1, B=1, and C=0 for GA and PSO. For BB, the velocity was not considered in the cost function because it searches the solution from the highest velocity to the lowest one. The first term in the cost function penalizes the difference between achieved coiling temperature and specified coiling temperature. The second term in the cost function penalizes the choice of a low-skelp velocity. The best solutions and specified coiling temperatures for different optimizations are presented in Figures

7-3, 7-4, and 7-5. The coiling temperature errors for GA, PSO, and BB were 6.3°C, 4.9°C, and 3.1°C respectively (Table 7-2). In addition, the velocity of the strip for optimal solutions of GA, PSO and BB were 4.5 m/s, 3.75 m/s and 3.75 m/s respectively.

The optimal configuration for the 500°C conditions using BB was 4 banks "on," all side-sprays (except for the last one) "off" and a velocity equal to 3.75 m/s (Coiling temperature for BB Table 7-2, Figure 7-5). The predicted final temperature was 496.9°C.

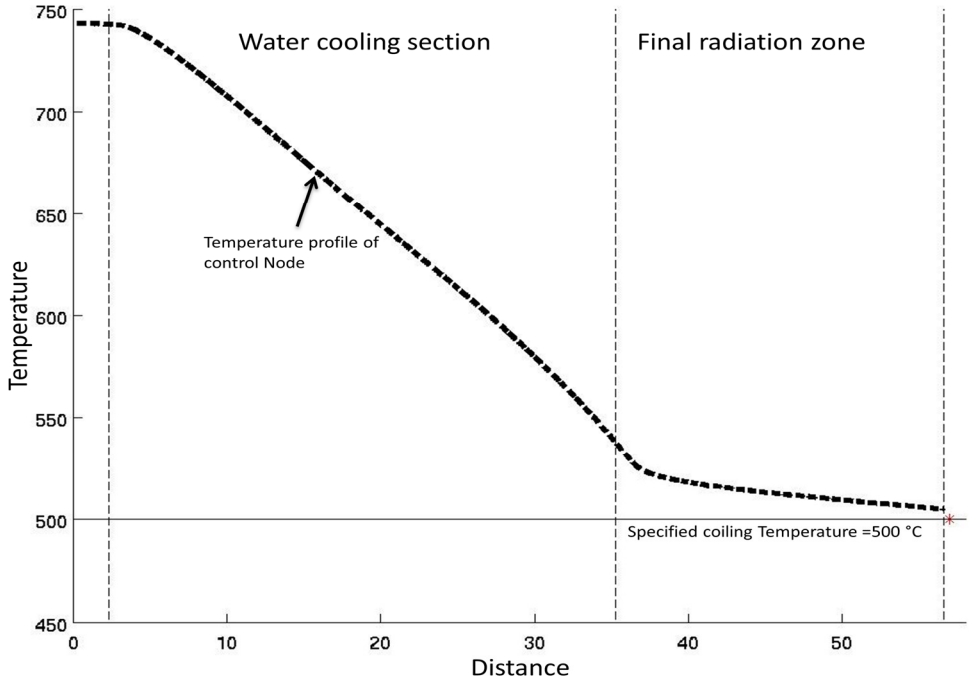

Figure 7-3. Temperature profile of control Node - GA optimization
(Objective: coiling temperature=500°C for control Node)

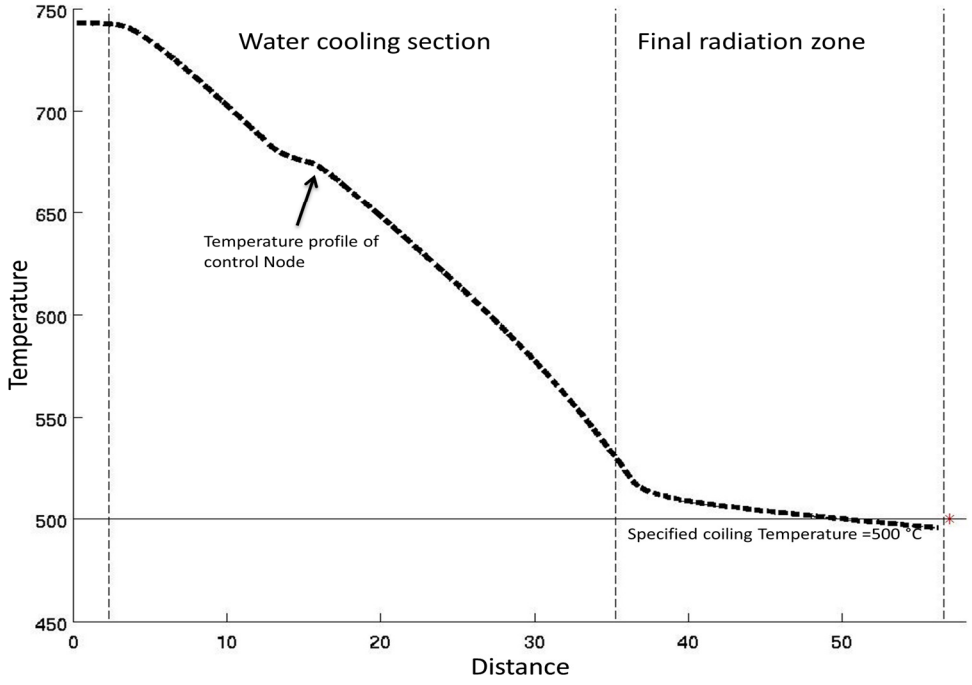

Figure 7-4. Temperature profile of control Node - PSO optimization

(Objective: coiling temperature=500°C for control Node)

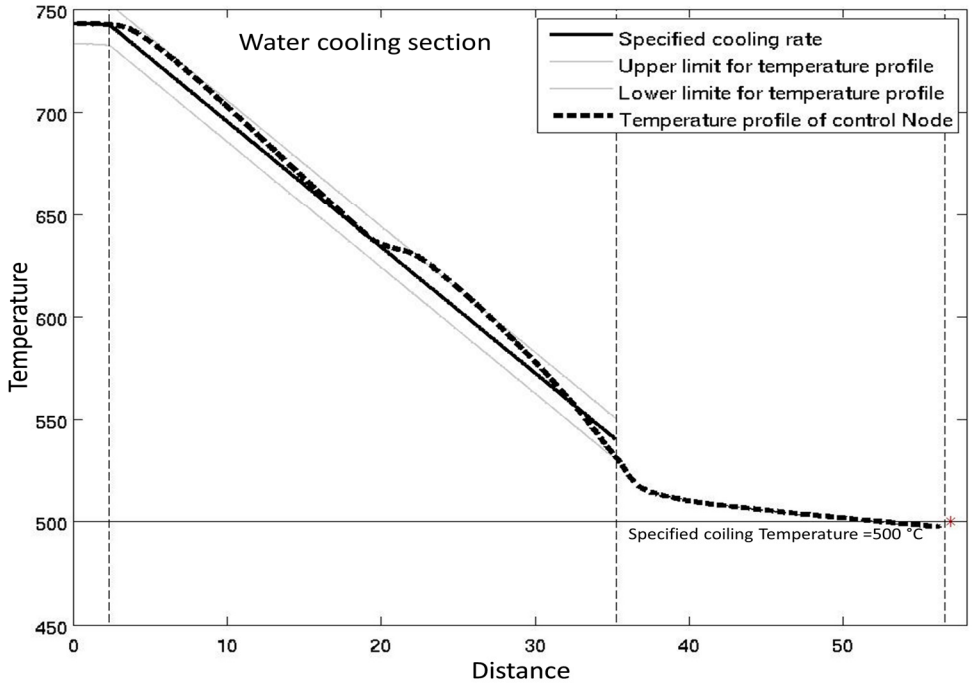

Figure 7-5.Temperature profile of control Node- the branch-and-bound optimization (Objective: coiling temperature=500°C for control Node)

The stopping criterion for the branch-and-bound was an absolute error below 5°C for the coiling temperature, but for GA and PSO, the stopping criteria were the stall generation being equal to 10 generations (i.e. no improvement in best fitness function after 10 generations). Inevitably, branch-and-bound proved the fastest solution (highest skelp velocity) with an acceptable error (lower than 5°C), because the method checked all possibilities, from the highest velocity to the lowest one. In conclusion, a faster velocity in GA does not mean that its result will be better than BB or PSO because the coiling temperature error, which was the stopping criterion in BB, was 5°C.

The temperature profile of the optimal solution by using BB is illustrated in Figure 7-5. The restrictions in BB to reject the improper control actions that do not result in the specified coiling temperature include the confidence interval (±10°C) around the cooling rate, which leads to a coiling temperature equal to

500°C. The control actions, which exceed the limited area around the desired cooling rate (i.e. the confidence interval of ±10°C), were rejected to decrease the search possibilities.

7.1.2 Comparison of optimization results on cooling rate optimization

The objective in this section is to minimize the sum of squared errors for all points on the temperature profile (i.e. cost function).

Using all three optimization methods, the temperature profile was optimized to be as close as possible to the specified cooling rate (11.5°C/s). In this case, the velocity of the strip is not considered in the cost function. The optimal configuration for each method is listed in Table 7-2. The optimized temperature profiles, using GA, PSO, and BB, are shown in Figures 7-6, 7-7, and 7-8 respectively. The grey lines in Figures 7-6, 7-7, and 7-8 are used to indicate the limitation for cooling rate; the solid lines indicate the specified (target) cooling rate. The difference between the specified and achieved cooling rate is measured over the entire length of the water-cooling system. The average error for GA, PSO, and BB were 0.87%, 0.47% and 0.84% respectively, which indicates that the average of differences in achieved and specified temperatures for all points were approximately 992°C, 451°C and 300°C respectively.

The errors that are presented in Table 7-2 for cooling rate optimization in this section are the sum of the errors for all points on the temperature profile, which is calculated from equation 6-1. In order to obtain the desired objective function for stochastic optimization methods in this section, the coefficients of general cost function (described in Equation 6-1) are A=0, B=0 and C=1. The third term (with coefficient C) in the general cost function indicates the penalty for the difference between achieved and specified temperature profile during the water-cooling section only.

The upper and lower limits for the cooling rate were determined to be ±20°C of the desired cooling rate (11.5°C/s). All limits are indicated in Figure 7-6 to 7-8 by gray lines in order to validate results within the limit (i.e. all points of temperature profile have the error under ±20°C). The Branch-and-bound method uses limits to remove the configurations, whose temperature profiles were not within the limits. In other cases, it was used to show the validity of results (i.e. acceptable error with respect to desired temperature profile).

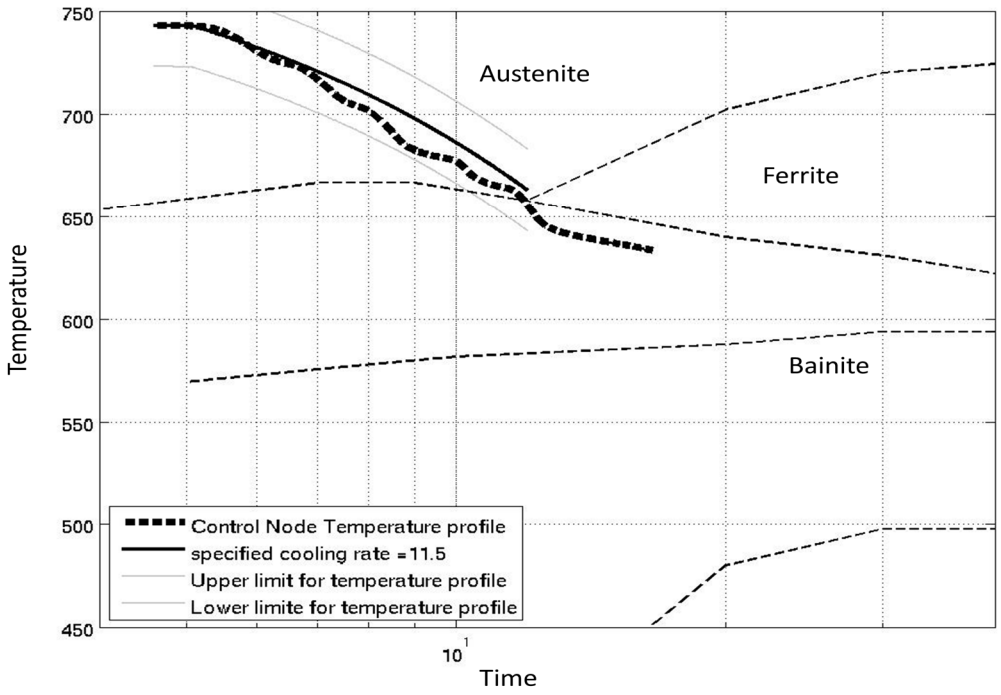

Figure 7-6. Temperature profile of control Node for early cooling strategy- GA optimization (Objective: coiling temperature=550°C for control Node)

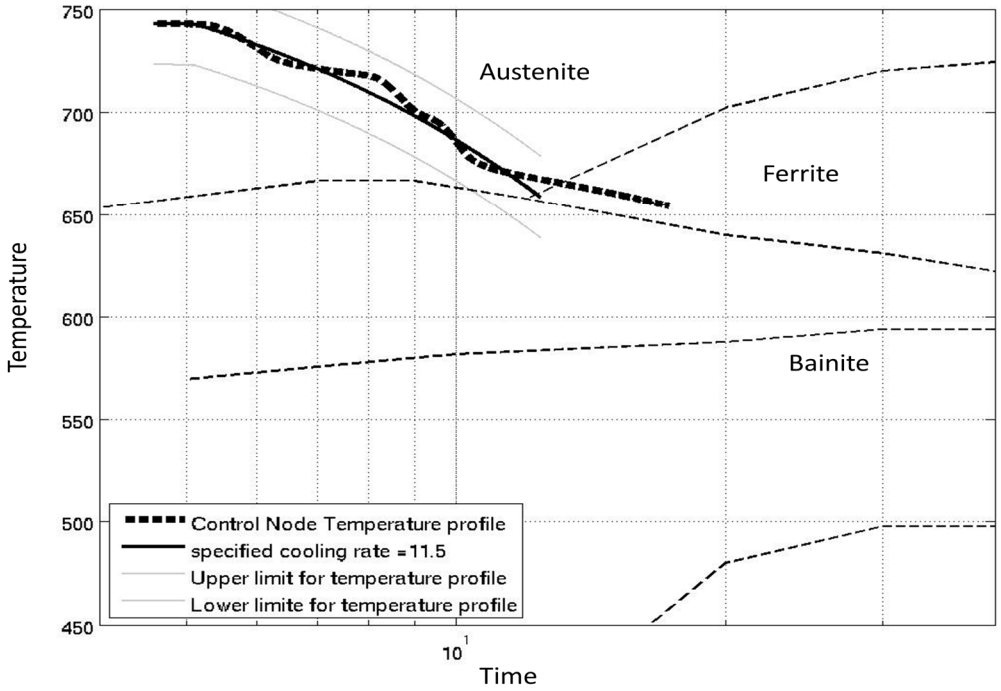

Figure 7-7. Temperature profile of control Node for early cooling strategy – PSO optimization (Objective: coiling temperature=550°C for control Node)

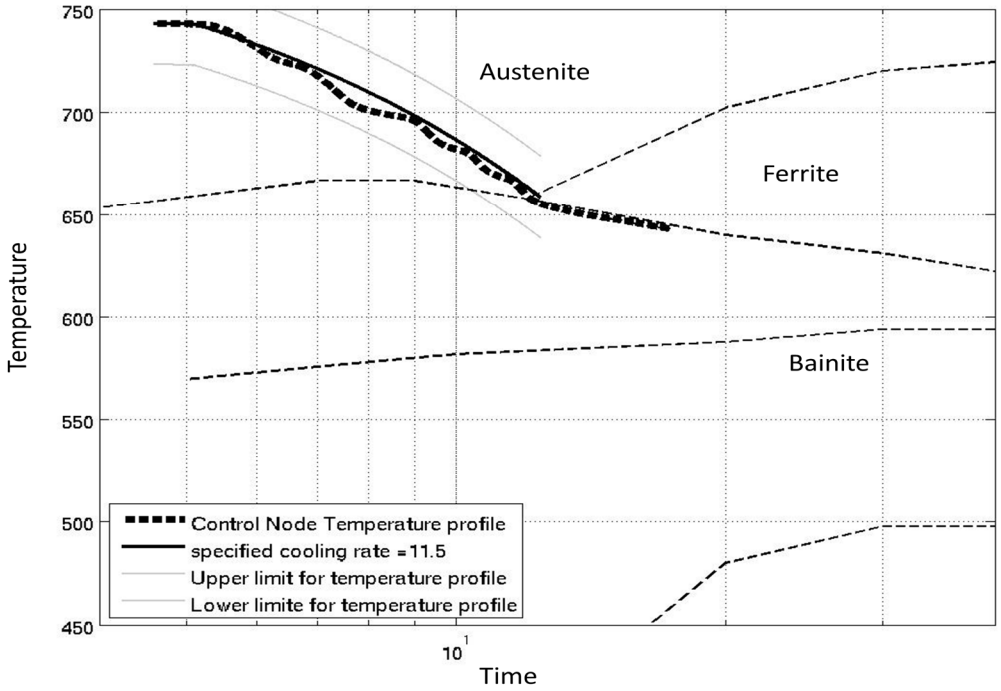

Figure 7-8. Temperature profile of control Node for early cooling strategy- the branch-and-bound optimization (Objective: coiling temperature=550°C for control Node)

As illustrated in Figure 7-6, the confidence interval for temperature variation in BB optimization restricted the temperature profile close to the desired value for the cooling rate. The errors, convergence time and total number of simulations for an optimal solution of cooling rate are presented in Figure 7-2. The optimal solution from BB is the global minimum between all possibilities. Additionally, PSO was faster and had a lower error in comparison with GA in this case. However, the results of all these methods were within the confidence interval of cooling rate.

7.2 Summary and Conclusions

The mechanical properties of a steel strip are defined by its microstructure, which is a function of the temperature profile in the strip during the cooling process. In this work, an optimization approach for obtaining a desired temperature profile in the strip is presented, and the data show that genetic algorithms are a computationally viable method for obtaining optimal configurations for a laminar cooling system. In this case, a fixed coiling temperature, a constant cooling rate, a fixed coiling temperature and a constant cooling rate were all considered. The optimization algorithm was used in simulation to provide configurations that could closely match the desired coiling temperature and cooling rates. However, when both coiling temperature and cooling rate were specified, the physical layout of the system necessitated that both could not be satisfied simultaneously.

Sensitivity analysis showed that the choice of control node greatly affects the optimal cooling system configuration. The temperature of near-surface nodes is much more difficult to control than near-centerline nodes because of the greater heat flux near the surface.

The methodology given by the genetic algorithm is very flexible. First, there is the issue of determining the best "trade-off" deviation from the desired temperature profile versus deviation from the desired coiling temperature. Second, the connection between temperature profile and microstructure needs to be more closely examined. Third, the procedure proposed in Section 4 could be used to evaluate potential cooling systems with respect to their ability to generate a desired cooling profile (and hence a desired microstructure).

The general cost function (equation 3-25) was defined to penalize the coiling temperature errors (first term), maximize the velocity of the strip (second term), and minimize the cooling rate error (third term). This cost function is used for different objectives by changing the coefficients in the general format.

The optimization techniques used in this work include two heuristic methods (Genetic Algorithm and Particle Swarm Optimization) and one deterministic technique (Branch-and-Bound). Stochastic optimization techniques like GA and PSO can lead to an acceptable solution (i.e. errors lower than 2% for coiling temperature), in a reasonable time (with respect to number of simulations), for coiling temperature and cooling rate as presented in Tables 3-7 and 4-10. However, they are not guaranteed to determine the global best in the entire solution, whereas BB can. This global best is the fittest result according to the cost function for the specific objective. Thus a comparison between three optimization methods with respect to coiling temperature and cooling rate shows that the BB method can achieve the best result in the shortest time.

Using Branch-and-Bound, it is also possible to track more complex objectives on the run-out table (ROT). BB was used in order to obtain optimal configuration with a coiling temperature of 550°C using three different cooling strategies (early, late and constant). These strategies also include two different uniform rates for the first and last three banks. In addition, BB was used to optimize the configuration of the ROT to maximize the volume of the strip, which is cooled through a desired zone in the CCT diagram. When using this objective, it is important to ensure that the temperature of the entire strip, not just one node, is in the desired range. This indicates that the mechanical and metallurgical properties of the entire steel strip are in the desired zone during the laminar cooling process.

The comparison between maximum volumes for different velocities shows that in higher velocities it is harder to control the temperature profile in the desired zone. However, in the lower velocities it is possible to control all nodes of the strip.

Ultimately, it is a “trade-off” between choosing faster production and higher strip volume.

7.3 Future work

The field of modelling and controlling the run-out table holds significant potential. In order to better understand the dynamics of the laminar cooling system and improve the economic competiveness of steel production using water banks and side-sprays, the following objectives for future work are recommended:

1. The algorithm proposed for Genetic Algorithms led to satisfactory output prediction; however, the parameters may not be optimal. More time and research will be required to determine the optimal parameters for both GA and PSO.

2. Optimal experimental design could be used to determine the accuracy of the simulated results in an industrial plant.

3. The FEM thermal model could be modified to predict the different phases and constituents of steel during the laminar cooling.

4. A model-based predictive control algorithm could be designed and implemented to take advantage of the model established in this work.

5. The best "trade-off" between desired temperature profile deviation and desired coiling temperature deviation has yet to be determined.

6. The model could be changed in order to allow more banks (e.g. 6 or 7), add 25% more headers per bank, consider a laminar sheet cooling system, or extend the length of the run-out table during the cooling process.

7. By defining a cooling rate for steel (i.e. non-uniform) during the cooling process, it would be possible to guarantee the steel properties during and after the cooling process. The search for a non-uniform cooling rate to obtain the desirable properties for steel is an ongoing one.

8. It is important to note that the results of this thesis are based on simulations only. While the model's predictions are consistent with industry results, the optimization methods should still be tested in an actual factory setting.

9. It will be necessary to experiment with different cooling strategies using different micro-structural and mechanical properties.

A

Mathematical equations to model the heat transfer behaviour of steel skelp during cooling process[3]

In the run out table of a hot strip mill, the strip is cooled by water and air from the top and bottom. The schematic view of ROT is shown in Figure 3. The prime modes of heat transfer on the ROT are convection through the water, and radiation to the air. Conduction along the thickness of the strip and conduction to the support rolls also affect the temperature of the strip in the run out table (Kumar R., 1996). The latent heat and thermal conductivity vary with the temperature of the strip during the cooling process.

Hinrichsen (Hinrichsen E.N., 1976) modeled the cooling process by a first order system with a time constant. Sonehara (Sonehara M., 1987) used a lumped model with constant surface heat transfer coefficient to model the heat transfer of run out table. Yaniro (Yaniro K., 1991) used a linear one dimensional heat conduction equation considering the heat transfer in the thickness direction in the strip to model the strip temperature during the cooling process. All of these models did not consider the variation of thermal conductivity and latent heat with temperature

[3] All variables in this Appendix are explained in Nomenclature.

of the strip during the cooling process. The other models before 1996 considered one dimensional heat transfer model for temperature of the strip and did not take into account the variation of specific heat of the strip heat and thermal conductivity with temperature (Uetz G., 1991) (Yanangi K., 1993).

Krishna (Kumar R., 1996) modeled the cooling process as a one dimensional model heat conduction equation:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) + H_u \gamma_u \quad \text{(A-1)}$$

Included in his model are the thermal conductivity (k), specific heat (c_p), rate of phase transformation (γ_u) and latent heat of phase transformation (H_u). The variation of thermal conductivity and specific heat with temperature were found vary with temperature as per following equation:

$$k = a_1 - b_1 u \quad \text{(A-2)}$$

$$c_p = 383.7667 + 0.7069\, T \quad \text{(A-3)}$$

Where a_1 and b_1 are coefficients, which depend on the grade of steel. The values of a_1 and b_1 for two typical grades of steel are given:

Carbon steel: a_1=61.8474, b_1=0.0437

Carbon-silicon steel: a_1=53.7294, b_1=0.0329

This model is not thorough since the effects of convection and radiation are ignored. Also, it only considered the conduction heat transfer in one dimension, which is not enough to completely model the cooling process.

Biswas et al. (Biswas S.k., 1997) developed a two dimensional heat transfer model. He ignored the heat transfer by radiation and the conductive heat transfer by contact between the strip and support rolls. Constant velocity of the strip and symmetric cooling at the top and bottom surfaces of the strip were also supposed in their work. By considering a uniform temperature distribution in span direction of z (i.e. vertical to the surface of the strip) the Eulerain heat transfer equation was:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) - \rho c_p \left(u \frac{\partial T}{\partial x}\right) \quad \text{(A-4)}$$

With the following boundary conditions:

$$\begin{aligned} T|_{x=0} &= T_0, \\ \left.\frac{\partial T}{\partial y}\right|_{y=0} &= 0, \\ \left.\frac{\partial T}{\partial x}\right|_{x=L} &= 0 \\ -k\left.\frac{\partial T}{\partial x}\right|_{y=H} &= h(x)(T(x)-T_\infty) \end{aligned} \tag{A-5}$$

The spatially distributed heat transfer coefficient function *h(x)* was determined based on experimental studies as:

$$h(x) = \bar{a}v^{0.6}(x/u)^{1.89}e^{-\bar{b}(x/u)} \tag{A-6}$$

where $\bar{a}$ and $\bar{b}$ are constant, *u* is the velocity of the strip and *v* is the velocity of the water jet. Experimental values for $\bar{a}$ and $\bar{b}$ are $\bar{a} = 17{,}694(\frac{u}{w_c})^{1.89}$ and $\bar{b} = 0{,}352\,\frac{u}{w_c}$, where w_c is the width of the water jet.

The Biswas model was a two-dimensional model; however, it did not consider the latent heat of phase transformation, radiation and variation of thermal conductivity with temperature.

Guo (Guo R.M., 1997) used equation 4 for heat transfer without the last term:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) \tag{A-7}$$

With the boundary conditions on its top and bottom surfaces:

$$\pm\left(k\frac{\partial T}{\partial y}\right) = h(T - T_a) \tag{A-8}$$

Where

$$h = h_a + h_{wt}\frac{T-T_w}{T-T_a}h_{wb}\frac{T-T_w}{T-T_a} + \sigma\varepsilon\frac{T^4-T_a^4}{T-T_a} \tag{A-9}$$

and

Where T_a is the ambient temperature, T_w is the water temperature, T is strip temperature, and E is the emission coefficient.

The radiation and air cooling boundary are only applicable in the dry zones. Heat transfer coefficients h_{wt} and h_{wb} are unknowns, determined in the tuning stages

using the inverse method. The heat transfer of forced air convection, used in Guo's work, was:

$$h_a = 0.010093\, V^{0.926} \tag{A-10}$$

Where V is the velocity of the strip in m/s.

Guo's model was simplified by ignoring the heat transfer in the *x* dimension because it is negligible when compared to that in the y dimension. In addition, the model did not consider variation in the latent heat of phase transformation and thermal conductivity with temperature. The model included did, however, include the radiation, convection and conduction heat transfer during the cooling process.

Samaras (Samaras N.S., 2001) used the same equation as Guo (equation A-7) as the heat transfer model of the cooling section. The only difference is the convection and radiation heat transfer equations. The temperature change due to cooling water in Samaras work was calculated as:

$$\Delta T_d = \frac{2k}{\rho(t)c(T)h}(T - T_w)\left[\frac{\Delta L}{\pi a v}\right]^{0.5} t_w \tag{A-11}$$

Where w is the strip width, Δl is the water contract length, T_w is the water temperature, t_w is the water contact time, a is the thermal diffusivity, v is the strip velocity, and h is the strip thickness.

Temperature change due to heat radiation was calculated as:

$$\Delta T_r = \frac{s\varepsilon A_r}{\rho(t)c(T)V}[(T - 460)^4 - (T_a + 460)^4]^{0.5} t_r \tag{A-12}$$

Where s is the Stefan Boltzman constant, ξ is the emissivity, A_t is the surface area subjected to radiation, T_a is the ambient temperature. $V$is the volume of the body, and t_r is the radiation time.

Samaras did not consider the heat losses by convection and conduction to the work rolls. In addition, the Samaras model is an offline one while the Guo's

model is an online model (which means that the heat transfer coefficients are tuned during the cooling process in order to track the coiling temperature correctly).

Serajzadeh (Serajzadeh S., 2003), used the simplified heat transfer model as:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) \quad \text{(A-13)}$$

The boundary condition for cooling in the air is assumed as:

$$\pm\left(k\frac{\partial T}{\partial n}\right) = h_\infty(T - T_\infty) + \varepsilon\sigma(T^4 - {T_\infty}^4) \quad \text{(A-14)}$$

while for the water descaling zone, it is postulated as:

$$\pm\left(k\frac{\partial T}{\partial n}\right) = h_w(T - T_w) \quad \text{(A-15)}$$

where the dimension *n* refers to both the *y* and *z* dimensions. The terms h_∞ and h_w are heat transfer coefficients between rolled metal and air and water respectively. The Serajzadeh model considered the latent heat of transformation from austenite to bainite and pearlite during the cooling process from the data published by Darken and Gurry (Darken L.S., 1953).

$$\Delta H_{\gamma\to\alpha} = 20789 - 15.623\,T - 0.24\,T^2 \quad \text{(A-16)}$$

$$\Delta H_{\gamma\to p} = 120848 - 52.42\,T - 0.158\,T^2 \quad \text{(A-17)}$$

The variation of specific heat and thermal conductivity with temperature for ferrite, pearlite and austenite are as follows:

$$c_p^\alpha = 14822.82 - 495.64\,T - 0.5523T^2 - 2.0495 * 10^{-4}T^3 \quad \text{(A-18)}$$

$$c_p^p = -1158.44 - 11.31\,T - 0.024\,T^2 - 1.777 * 10^{-5}T^3 \quad \text{(A-19)}$$

$$c_p^\gamma = 474.622 - 1.148\,T \quad \text{(A-20)}$$

$$k = 59.92 - 0.0221\,T - 5.4 * 10^{-5}\,T^2 - 4.3 * 10^{-8}T^3 \qquad \text{(A-21)}$$

Where $c_p^{\alpha}, c_p^{p}\ and\ c_p^{\gamma}$ are the specific heat of ferrite, pearlite and austenite respectively and k is the thermal conductivity.

Chatterjee developed a nonlinear differential equation (Chatterje S., 2001). The dynamics of the system is represented by the time derivative of temperature. The differential equation for the heat transfer can be expressed as:

$$\frac{dT}{dt} = -K\,\frac{f(T).g(V)}{H(h)} \qquad \text{(A-22)}$$

Where

$f(T) = T^a$: A function of Temperature

$g(V) = (\frac{V}{V_b})^m$: A function of speed

$H(h) = h^n$: A function of gauge

Filtered data from 14,000 coils were used to fit parameters for the model. This was done separately for different categories, base upon temperature, thickness, speed and width. The results of this model yielded acceptable result, but they were generated for the specific plant, used in this work.

In conclusion, heat transfer models are used to predict the temperature of the run out table (specifically the coiling temperature) during the cooling process to optimize the mechanical properties of steel. Indeed the mechanical properties are dependent to the constituents formed during the cooling process, which is positively related to cooling rate and temperature of the steel strip. The thermo-physical models are those in which the phase of steel has influence on the temperature of the strip (Kumar R., 1996) (Latzel S., 2001). The other methods are using CCT or Time Temperature Transformation (TTT) diagram to determine the phase of steel during cooling.

Table A-1.Heat transfer model for control or optimization goals

Name of researcher Heat transfer equation Characteristics of the model
Kumar et al. (Kumar R., 1996) **Heat transfer Equation:** $$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) + H_u \gamma_u$$ **Characteristics:** -One-dimensional -For only two grade of steel: Carbon steel and Carbon silicon steel - Conduction and radiation were ignored
Biswas (Biswas S.k., 1997) **Heat transfer Equation:** $$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) - \rho c_p \left(u \frac{\partial T}{\partial x}\right)$$ **Characteristics:** -Two-dimensional - Conduction and radiation were ignored - Constant velocity of the strip - Symmetric cooling at the top and bottom surfaces $-k \frac{\partial T}{\partial x}\Big\|_{y=H} = h(x)(T(x) - T_\infty)$ - Spatially distributed heat transfer coefficient function *h(x)* was determined based on

experimental studies

- Did not consider the latent heat of phase transformation, radiation and variation of thermal conductivity with temperature

Guo (Guo R.M., 1997)

Heat transfer Equation:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)$$

Characteristics:

-The heat transfer in the *x* dimension was ignored

-Boundary conditions on its top and bottom surfaces:

$$\pm\left(k\frac{\partial T}{\partial y}\right) = h(T - T_a)$$

$$h = h_a + h_{wt}\frac{T - T_w}{T - T_a}h_{wb}\frac{T - T_w}{T - T_a} + \sigma\varepsilon\frac{T^4 - T_a^4}{T - T_a}$$

- Did not consider variation in the latent heat of phase transformation and thermal conductivity with temperature

- include the radiation, convection and conduction heat transfer during the cooling process

- The heat transfer coefficients are tuned during the cooling process in order to track the coiling temperature correctly

Samaras (Samaras N.S., 2001)

Heat transfer Equation:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)$$

Characteristics:

-Same as Guo with difference in the convection and radiation heat transfer equations.

$$\Delta T_d = \frac{2k}{\rho(t)c(T)h}(T - T_w)\left[\frac{\Delta L}{\pi a v}\right]^{0.5} t_w$$

$$\Delta T_r = \frac{s\varepsilon A_r}{\rho(t)c(T)V}[(T - 460)^4 - (T_a + 460)^4]^{0.5} t_r$$

-Did not consider the heat losses by convection and conduction to the work rolls

Serajzadeh (Serajzadeh S., 2003)

Heat transfer Equation:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)$$

- cooling in the air

$$\pm\left(k\frac{\partial T}{\partial n}\right) = h_\infty(T - T_\infty) + \varepsilon\sigma(T^4 - {T_\infty}^4)$$

-cooling with water

$$\pm\left(k\frac{\partial T}{\partial n}\right) = h_w(T - T_w)$$

Characteristics:

-The Serajzadeh's model considered the latent heat of transformation from austenite to bainite and pearlite during the cooling process from the data published by Darken and Gurry (Darken L.S., 1953).

(Chatterje S., 2001)

Heat transfer Equation:

$$\frac{dT}{dt} = -K\frac{f(T).g(V)}{H(h)}$$

Characteristics:

-Nonlinear differential equation

-Filtered data from 14,000 coils were used to fit parameters for the model

References

(Biswas S.k., 1997) Biswas S.k. Chen S.J., Satyanarayana A. *"Optimal temperature tracking for accelerated cooling processes in hot rolling of steel"*, Dynamics and Controls. Vol. 7, 1997, pp. 327–340.

(Chatterje S., 2001) Chatterje S. Simonelli G., Chizeck H. *"Parameter identification of a nonlinear coiling temperature model for run-out table control at LTV Cleveland works 84-In hot strip mill"*, Case Western Reserve University. 2001.

(Guo R.M., 1997) Guo R.M. *"Modelling and simulations of run-out table cooling control using feed-forward-feedback and element tracking system"*, IEEE Transactions on Industry Applications. Vol. 33, 1997, p. 304.

(Hinrichsen E.N., 1976)Hinrichsen E.N. *"Hot strip mill run-out table cooling- A system view of control, operation and equipment"*, AISE yearly Proceedings. 1976, pp. 403-408.

(Kumar R., 1996) Kumar R. Lahiri A.K., *"Modelling of the cooling process on the runout table of a hot strip mill"*, IEEE, 1996.

(Samaras N.S., 2001) Samaras N.S. Simaan M.A. *"Novel Control Structure for Runout Table Coiling Temperature Control"*, AISE steel technology. No 6 : Vol. 87, June 2001. pp. 55-59.

(Serajzadeh S., 2003) Serajzadeh S. *"Prediction of temerature distribution and phase transformation on the run-out table in the process of hot stri rolling",* Applied mathematical modelling. 2003, pp. 861-875.

(Sonehara M., 1987) Sonehara M. Yamane T., Yuasa Y., *"New temperature control system of hot strip mill run-out table"*, NKK technical reort. Vol. 129, 1987. pp. 9-14.

(Uetz G., 1991) Uetz G. Woelk G., Bischops T., *"Influencing the formation of the steel structure by suitable temperature control in the run-out table sections of hot strip mills"*, Steel research. No 5 : Vol. 62, 1991, pp. 216-222.

(Yanangi K., 1993) Yanangi K. *"Prediction of strip temperature for hot strip: a combination of physical modelling, control roblems and practical adaption"*, Trans on Automatic control. No 7 : Vol. 38,1993. pp. 1060-1065.

(Yaniro K., 1991) Yaniro K. Yamasaki M., Furukawa M., et al., *"Development of coiling temperature control system on hot strip mill"*, Technical report. Vol. 24, 1991.

B

MATLAB® codes for the Genetic Algortihms optimization method

Genetic Algorithm sample code for first two generations

This section includes MATLAB codes for GA. The GA procedure in this code is like what is explained in Chapter 3 and 4 about GA. This File with the three files in Appendix E can be run on a computer to optimize the ROT configuration with GA method.

```
%%=====================================================================
%% Start with a clear memory

clc
clear all
close all

%% The global variables in all files of GA optimization

global s
pop=20;                     % Number of individuals in one generation
ssinput=15                  % Number of digits in an individual
```

```
%%=======================================================================
%% Random input for first generation. This section generates the first
generation for GA optimization.

M=1
while M==1
p=rand(pop,ssinput);
for i=1:pop
  for j=1:ssinput
    if p(i,j)>0.5
      p(i,j)=1;
    else
      p(i,j)=0;
    end
  end
end
M=0;
for i=1:pop
  for j=1:pop
  if p(i,:)==p(j,:)
    if i==j
    else M=1
    end
  end
  end
end
end
%%=======================================================================
%% Evaluate cost of each entry. The cost values are determined by running
the model. The evaluation function get the results from model and
calculate the cost value.

Loss_old=[];
for i=1:pop
s=0;
  while s==0 || s==2
  p(i,:)=rand(1,15);
  for j=1:15
    if p(i,j)>0.5
      p(i,j)=1;
```

```
    else
      p(i,j)=0;
    end
  end
    input=p(i,:);
  input(i,15)=0;
  cost=baherfuncheckaPSO(input)  % The evaluation function
  p
  end
Loss1=cost;

Loss_old=[Loss_old;Loss1];       % The cost values save In a matrix.
end
p
Loss_old

%%=====================================================================
%% Rank according to cost values. This section gives a rank to each
individual according to its cost value in the first generation.

Loss_old1=Loss_old;
for i=1:20
  [min,indice]=min(Loss_old1);
  rankedcost(i)=Loss_old1(indice);
  rankedinput(i,:)=p(indice,:);
  Loss_old1(indice)=1000000;
  clear min
  clear indice
end
rankedcost
rankedinput

%% Elites. Top 4 individuals in first generation are elites and move
directly to second generation.

Elites=rankedinput(1:4,:)

%% Fitness scaling. In this section, the raw cost values of individuals
are converted to scaled values.

for i=1:20
  scaledcost(i)=(20^.5)*1/(i^.5);
```

```
end
S=0;
for i=1:20
  S=S+scaledcost(i);
end

%%=====================================================================
%% Roulette method. Using Roulette selection method, parents are selected
to generate children for next generation.

roulette=rand(1,24)*S;
for i=1:24
  sum=0;
  j=0;
while roulette(i)>sum
  j=j+1;
  sum=sum+scaledcost(j);
end
parentsnumber(i)=j;
parents(i,:)=rankedinput(j,:);
end
parentsnumber
parents

%% Random Matrice. The random matrix indicates which parent is dominant
in generating each chromosome of one child.

R=rand(12,15);

for i=1:12
  for j=1:15
    if R(i,j)>0.5
      R(i,j)=1;
    else
      R(i,j)=0;
    end
  end
end
R

%%=====================================================================
```

```
%%Crossover. Parents mate together according to crossover method which
was explained before in Chapters 3 and 4.

for i=1:12
  for j=1:15
    if R(i,j)==1
      children(i,j)=parents(2*i-1,j);
    else children(i,j)=parents(2*i,j);
    end
  end
end
M=0;
%%=========================================================================
%% Evaluate cost of children. Again, using evaluation function the cost
values are determined for each individual.
for i=1:12
  M=1
 while M==1
   M=0
  for j=1:pop
  if children(i,:)==p(j,:)
    if i==j
    else M=1
    end
  end
  end
  input=children(i,:);
  input(i,15)=0;
  cost=baherfuncheckaPSO(input)
 if s==0 || s==2
 M=1
 end
 if M==1
 R(i,:)=rand(1,15)
  for j=1:15
    if R(i,j)>0.5
      R(i,j)=1;
    else
      R(i,j)=0;
    end
  end
  R
```

```
  for j=1:15
    if R(i,j)==1
      children(i,j)=parents(2*i-1,j);
    else children(i,j)=parents(2*i,j);
    end
  end

 end
 end
end
%%=====================================================================
%% Mutation. Each generation has 4 mutated individuals.

TM=randi(20,1,4)
for i=1:4
  B=TM(i);
  mutation1(i,:)=rankedinput(B,:);
end
mutation1
T=rand(4,15);

for i=1:4
  for j=1:15
  if T(i,j)<0.2
     if mutation1(i,j)==1
       mutation(i,j)=0;
     else mutation(i,j)=1;
     end
  else mutation(i,j)=mutation1(i,j);
  end
  end
end
for i=1:4
  M=1;
  while M==1
  M=0
    for j=1:pop
  if mutation(i,:)==p(j,:)
    if i==j
    else M=1
    end
  end
```

```
  end

for i=1:4
for j=1:12
if mutation(i,:)==children(j,:)
  if i==j
  else M=1
  end
end
end

input=mutation(i,:);
input(i,15)=0;
cost=baherfuncheckaPSO(input)
s
if s==0 || s==2
M=1
end
if M==1
 T(i,:)=rand(1,15)
  i
  for j=1:15
  if T(i,j)<0.2
   if mutation1(i,j)==1
     mutation(i,j)=0;
    else mutation(i,j)=1;
   end
  else mutation(i,j)=mutation1(i,j);
  end
  end

  input=mutation(i,:);
  input(i,15)=0;
  cost=baherfuncheckaPSO(input)
  s
   if s==0 || s==2
   M=1
   end
  if M==1
    T(i,:)=rand(1,15)
      i
      for j=1:15
```

```
        if T(i,j)<0.2
          if mutation1(i,j)==1
              mutation(i,j)=0;
            else mutation(i,j)=1;
          end
        else mutation(i,j)=mutation1(i,j);
        end
        end
    end
    end
    M
    end
end


%% Display the second generation

p2(1:4,:)=Elites;
p2(5:16,:)=children;
p2(17:20,:)=mutation;
p2

for i=1:pop

input=p(i,:);
input(i,15)=0;
cost=baherfuncheckaPSO(input)
% cost=i*(-5);
Loss1=cost;



Loss_old=[Loss_old;Loss1];
end
Loss_old
```

C

MATLAB® codes for the Particle Swarm Optimization method

In this section, the Particle Swarm Optimization code is presented. Like GA, this file can be used with the other three files in Appendix E.

```
%%=====================================================================
%% Start with a clear memory

clc
clear all

% Defining the parameters and population size of PSO

global best y4 I iter_no GB input1
ssinput=14;
pop=20;
iter_no=0;

%%=====================================================================
% Make the first generation by random digits (0 and 1)
p=rand(pop,ssinput);
for i=1:pop
  for j=1:ssinput
    if p(i,j)>0.5
      p(i,j)=1;
```

```
    else
      p(i,j)=0;
    end
  end
end
%%=========================================================================
V_min=-4;                   % Minimum and Maximum particle velocity.
V_max=4;
r1=.5;                      % Random values between 0 and 1.
r2=.5;
c1=1.5;                     % Self confidence factor.
c2=1.5;                     % Swarm confidence factor.
w=0.5;                      % Inertia factor.
h=1;
%%=========================================================================
V=(V_min*ones(pop,ssinput)+(V_max-V_min)*rand(pop,ssinput));
for i=1:pop
  for j=1:ssinput
    if V(i,j)>V_max
      V(i,j)=V_max;
    elseif V(i,j)<V_min
      p(i,j)=V_min;
    end
  end
end

eps_pso=100;
Loss_old=[];
for i=1:pop
input=p(i,:);
cost=PSOfunction(input)        % Evaluate the cost of each configuration
Loss1=cost;
Loss_old=[Loss_old;Loss1];
end
%%=========================================================================
% Indicating particle best and global best of first generation
PB=p;
[Loss_min_old,indice]=min(Loss_old);
GB=p(indice,:);
iter_no
```

```
%%=========================================================================
% Evaluate the cost, particle best and global best for second generation
to the end of process

while iter_no<N
  iter_no=iter_no+1
  w=1;
  Loss_min_old
  delta_V=[];
  for i=1:pop
    r1=random('unif',0,1);
    r2=random('unif',0,1);
    delta_V1=c1*r1*(PB(i,:)-p(i,:))+c2*r2*(GB-p(i,:));
    delta_V=[delta_V; delta_V1];
  end
  V=w*V+delta_V;
  for i=1:pop
    for j=1:ssinput
      if V(i,j)>V_max
        V(i,j)=V_max;
      elseif V(i,j)<V_min
        V(i,j)=V_min;
      end
    end
  end
  for i=1:pop
    for j=1:ssinput
      sig=1/(1+exp(-V(i,j)));
      U_dist=random('unif',0,1);
      if U_dist<sig
        p(i,j)=1;
      else
        p(i,j)=0;
      end
    end
  end
  Loss_new=[];
  for i=1:pop
    i
    input=p(i,:);
    input(i,15)=0
cost=PSOfunction(input)
```

```
    Loss1=cost;
    Loss_new=[Loss_new;Loss1];
  end

%%=====================================================================
% The PB and GB are updated after evaluating cost values of all
individuals in each generation.
  for i=1:pop
    if Loss_new(i,1)<Loss_old(i,1)
      PB(i,:)=p(i,:);
      Loss_old(i,1)=Loss_new(i,1);
    end
  end
  PB
  Loss_old'
  [Loss_min_new,indice]=min(Loss_new);
  if Loss_min_new<Loss_min_old
    GB=p(indice,:);
    Loss_min_old=Loss_min_new;
  end
end
%%=====================================================================
```

D

MATLAB® codes for the branch-and-bound method

The branch-and-bound as a deterministic method is presented in this Appendix. Like GA and PSO, it needs the files in Appendix E to run the optimization.

```
%%=====================================================================
%% Start with a clear memory
 clc
 close all
 clear all

% indicate the global variables which are same in all files in the
process

 global M

 % The start configurations which indicate the Speed of the strip during
the optimization

 optimuminput1(1,:)=[0 0 0 0 0 0 0 0 0 0 0 0 1 1 1];
 optimuminput1(2,:)=[1 0 0 0 0 0 1 0 0 0 0 0 1 1 1];
 optimuminput1(3,:)=[1 0 0 0 0 0 0 0 0 0 0 0 1 1 1];
 iter_no=1;
 Sop=3;

%%=====================================================================
```

```
% This section evaluates the cost of the zone under the first bank and
its side-spray (in methods chapter 3 and 6- Branch-and-bound).

for ii=2:5
  M=ii;
  for j=1:Sop
  ii
  j
  input=optimuminput1(j,:);
  cost=BBfunction(input);
  optimum(3*j-2)=cost;
  optimuminput(3*j-2,:)=input;


  input=optimuminput1(j,:);
  input(ii)=1;
  input(ii+6)=1;
  cost=BBfunction(input);
  optimum(3*j-1)=cost;
  optimuminput(3*j-1,:)=input;

  input=optimuminput1(j,:);
  input(ii)=1;
  input(ii+6)=0;
  cost=BBfunction(input);
  optimum(3*j)=cost;
  optimuminput(3*j,:)=input;
  end
  clear finaloptimum
  clear finaloptimuminput
  i=1;

%%=========================================================================
% Rank the results according to their cost values
 while min(optimum)<=0
  [min,indice]=min(optimum)
  finaloptimum(i)=optimum(indice);
  finaloptimuminput(i,:)=optimuminput(indice,:);
  optimum(indice)=1000000;
  clear min
       clear indice
       i=i+1;
```

```
  end
       finaloptimum
       finaloptimuminput
i=1;
       SOP=size(finaloptimum);
       Sop=SOP(2);
       clear optimuminput1
 for i=1:Sop
       optimuminput1(i,:)=finaloptimuminput(i,:);
       clear optimuminput
       clear optimum
 end
  optimuminput=[];
  optimum=[];

%%=========================================================================
% Evaluate the survived configurations for sixth bank
for j=1:SOP(2)
  input=optimuminput1(j,:);
  cost=BBfunction(input);
  optimum(2*j-1)=cost;
  optimuminput(2*j-1,:)=input;

  input=optimuminput1(j,:);
  input(6)=1;
  cost=BBfunction(input);
  optimum(2*j)=cost;
  optimuminput(2*j,:)=input;
end
  clear finaloptimum
  clear finaloptimuminput
  i=1;

%%=========================================================================
% Rank the final survived results according to their cost values
if min(optimum)<=0
while min(optimum)<=0
  [min,indice]=min(optimum)
  finaloptimum(i)=optimum(indice);
  finaloptimuminput(i,:)=optimuminput(indice,:);
  optimum(indice)=1000000;
  clear min
```

```
  clear indice
  i=i+1;
end
else
  finaloptimum=[]
  finaloptimuminput=[]
end
```

E

Other files in MATLAB® codes

E-1 Evaluate Function

This function gets the input from the optimization algorithm and submits it to the FEM thermal model. After that the evaluation function get output file from the thermal model and evaluates the cost value for each individual according to the objective of each optimization. The cost values are sent to the optimization method to continue its procedure.

```
% The Function evaluate the cost of temperature profile for each
configuration

function cost=GAfunction(INPUT)

% defining the global variables

global best y4 I
 P=1;
 i=0;
 j4(1,1)=10;

% The input file is loaded to make changes

 load C462367_A37_75mm_Setup.txt;
```

```
%%=============================================================================
% Using the input from GA and change it to the format of input file of
ABAQUS

for iii=1:6
   if INPUT(1,iii)==1
     inputstr(1,iii)='1';
     input(1,iii)=1;
   else input(1,iii)=0;
     inputstr(1,iii)='0';
   end
 end


 for K=1:6
 C462367_A37_75mm_Setup(12*K-9,2)=input(1,K)+1;
 C462367_A37_75mm_Setup(12*K-8,2)=1;
 C462367_A37_75mm_Setup(12*K-7,2)=input(1,K)+1;
 C462367_A37_75mm_Setup(12*K-6,2)=1;
 C462367_A37_75mm_Setup(12*K-5,2)=input(1,K)+1;
 C462367_A37_75mm_Setup(12*K-4,2)=1;
 C462367_A37_75mm_Setup(12*K-3,2)=input(1,K)+1;
 C462367_A37_75mm_Setup(12*K-2,2)=1;
 C462367_A37_75mm_Setup(12*K-1,2)=input(1,K)+1;
 C462367_A37_75mm_Setup(12*K,2)=1;
 C462367_A37_75mm_Setup(12*K+1,2)=input(1,K)+1;
 C462367_A37_75mm_Setup(12*K+2,2)=1;
 end
 jj=1;
 if input(jj)==0
   C462367_A37_75mm_Setup(12*jj-9,2)=0;
   C462367_A37_75mm_Setup(12*jj-8,2)=0;
   C462367_A37_75mm_Setup(12*jj-7,2)=0;
   C462367_A37_75mm_Setup(12*jj-6,2)=0;
   C462367_A37_75mm_Setup(12*jj-5,2)=0;
   C462367_A37_75mm_Setup(12*jj-4,2)=0;
   C462367_A37_75mm_Setup(12*jj-3,2)=0;
   C462367_A37_75mm_Setup(12*jj-2,2)=0;
   C462367_A37_75mm_Setup(12*jj-1,2)=0;
   C462367_A37_75mm_Setup(12*jj,2)=0;
   C462367_A37_75mm_Setup(12*jj+1,2)=0;
```

```
   C462367_A37_75mm_Setup(12*jj+2,2)=0;
 end
 for jj=2:6
   if input(jj)==0 & input(jj-1)==0
   C462367_A37_75mm_Setup(12*jj-9,2)=0;
   C462367_A37_75mm_Setup(12*jj-8,2)=0;
   C462367_A37_75mm_Setup(12*jj-7,2)=0;
   C462367_A37_75mm_Setup(12*jj-6,2)=0;
   C462367_A37_75mm_Setup(12*jj-5,2)=0;
   C462367_A37_75mm_Setup(12*jj-4,2)=0;
   C462367_A37_75mm_Setup(12*jj-3,2)=0;
   C462367_A37_75mm_Setup(12*jj-2,2)=0;
   C462367_A37_75mm_Setup(12*jj-1,2)=0;
   C462367_A37_75mm_Setup(12*jj,2)=0;
   C462367_A37_75mm_Setup(12*jj+1,2)=0;
   C462367_A37_75mm_Setup(12*jj+2,2)=0;
   end
 end
%%=========================================================================
% Velocity of the strip is indicated using the last three digits of each
individual.

 Speed=input(1,13)*1+input(1,14)*.5+input(1,15)*.25+3;


%%=========================================================================
% The converted numbers are inserted to the input of ABAQUS.
fid=fopen('C462367_A37_75mm_Setup.txt','w+');
 A=C462367_A37_75mm_Setup';
 jadid=A;
 A=fprintf(fid,' %4.2f %1.0f\n',A);
 fclose(fid);

%%=========================================================================
% The model will run if the configuration did not run before. The ABAQUS
repeats that run if analysis exited with an error

findingtool
 if s==1
   repeatedoutputfile
 else
! abaqus job=462367_A37_75mm user=462367_A37_75mm_Film.f
```

```
s=unix('grep "THE ANALYSIS HAS BEEN COMPLETED" 462367_A37_75mm.dat')
 k=0;
 while s==1 || s==2
  s=unix('grep "THE ANALYSIS HAS BEEN COMPLETED" 462367_A37_75mm.dat');
  if     unix('grep     "Abaqus/Analysis     exited     with     errors"
462367_A37_75mm.log')==0
    disp('running again')
%%=========================================================================
% The results files are removed before next run.

  ! rm /home/bineshma/*.log;
  ! rm /home/bineshma/*.lck;
  ! rm /home/bineshma/*.mdl;
  ! rm /home/bineshma/*.msg;
  ! rm /home/bineshma/*.odb;
  ! rm /home/bineshma/*.prt;
  ! rm /home/bineshma/*.res;
  ! rm /home/bineshma/*.stt;
  ! rm /home/bineshma/*.cid;
  ! rm /home/bineshma/*.com;
  ! rm /home/bineshma/*.fil;
  ! rm /home/bineshma/*.023;
  ! rm /home/bineshma/area;
  ! rm /home/bineshma/area2;
  ! rm /home/bineshma/*.dat;
  ! rm /home/bineshma/*.sta;
    ! abaqus job=462367_A37_75mm user=462367_A37_75mm_Film.f
  end
  k=k+1;
 end
 end
%%=========================================================================
% The time of the process is read from .sta file
 !sed -i '/1U/ d' 462367_A37_75mm.sta
 !sed -i '/2U/ d' 462367_A37_75mm.sta
 !sed -i '/3U/ d' 462367_A37_75mm.sta
 !sed -i '/4U/ d' 462367_A37_75mm.sta

%%=========================================================================
% In this file all nodes temperature are loaded to MATLAB program
 nodeoutput
```

```
%%=============================================================================
% The time and place of all nodes temperature during the cooling process
is loaded to the MATLAB program.

 !awk '$1="1" {print$8}' 462367_A37_75mm.sta>steptime;
 !awk '/STEP/{getline;next}1' steptime>steptime2;
 load steptime2;
 X=steptime2;
 x=X*Speed;
 clear Matrix
%%=============================================================================
% Indicating the desired cooling rates(uniform, late cooling, early
cooling during the process using the place and time matrices.

l=1;
 while x(l)<=2.36
   l=l+1;
 end
 m=1;
 while x(m)<=20.93
   m=m+1;
 end
 l
 m
 middletemp=y98(m);
 mtemp=700;
 YyCT(1:(l-1))=y98(1:(l-1));
%%=============================================================================
       % For uniform cooling rate:

       CR=(742.1-575)*Speed/(35.25-2.36);
       for bbb=l:size(y98)
   YyCT(bbb)=y98(l)-(X(bbb)-X(l))*CR;
       end

       % For early cooling

       CR1=(742.1-mtemp)*Speed/(20.93-2.36);
       CR2=(mtemp-575)*Speed/(35.25-20.93);
```

```
      for bbb=l:m-1
    YyCT(bbb)=y98(l)-(X(bbb)-X(l))*CR1;
      end

      for bbb=m:size(y98)
    YyCT(bbb)=mtemp-(X(bbb)-X(m))*CR2;
      end
       YCT=YyCT';

      % For late cooling rate

    Yy(1:(l-1))=y98(1:(l-1));
      for bbb=l:m-1
    Yy(bbb)=y98(l)-(X(bbb)-X(l))*CR1;
      End

      for bbb=m:size(y98)
    Yy(bbb)=mtemp-(X(bbb)-X(m))*CR2;
      end
    Y=Yy';
%%=========================================================================

 COSTfunction % This file calculates the cost value for each individual.

%%=========================================================================
% Input (RPT configuration) and the cost function are import to the
result text file
      input1=input;
   input1(16)=cost;
      outputnewfile
  fid=fopen('best1.txt','a+');
      A=fprintf(fid,' %4.0f %4.0f %4.0f %4.0f %4.0f %4.0f %4.0f %4.0f
      %4.0f %4.0f %4.0f %4.0f %4.0f %4.0f %4.0f %4.2f %4.0f\n',input1');
  fclose(fid)

% Remove the results files for next model runs.
  ! rm /home/bineshma/*.dat;
  ! rm /home/bineshma/*.sta;
end
%%=========================================================================
```

E-2 Node output file

This file loads all nodes temperature into the MATLAB workspace to evaluate the cost values of each configuration. This is the sample code for 9 nodes. All nodes are about 459 nodes which need many pages to print, so they are skipped in this Appendix.

```
%%=====================================================================
% Read each node's temperature from .dat file and copy it to area file.
After that remove the extra line to load it in MATLAB workspace
  ! awk '$1=="1" {print $0}' 462367_A37_12mm.dat > area1;
  ! awk '$1<$2 {print $0}' area1 > area101;
  ! awk '$1=="2" {print $0}' 462367_A37_12mm.dat > area2;
  ! awk '$1<$2 {print $0}' area2 > area102;
  ! awk '$1=="3" {print $0}' 462367_A37_12mm.dat > area3;
  ! awk '$1<$2 {print $0}' area3 > area103;
  ! awk '$1=="4" {print $0}' 462367_A37_12mm.dat > area4;
  ! awk '$1<$2 {print $0}' area4 > area104;
  ! awk '$1=="5" {print $0}' 462367_A37_12mm.dat > area5;
  ! awk '$1<$2 {print $0}' area5 > area105;
  ! awk '$1=="6" {print $0}' 462367_A37_12mm.dat > area6;
  ! awk '$1<$2 {print $0}' area6 > area106;
  ! awk '$1=="7" {print $0}' 462367_A37_12mm.dat > area7;
  ! awk '$1<$2 {print $0}' area7 > area107;
  ! awk '$1=="8" {print $0}' 462367_A37_12mm.dat > area8;
  ! awk '$1<$2 {print $0}' area8 > area108;
  ! awk '$1=="9" {print $0}' 462367_A37_12mm.dat > area9;
  ! awk '$1<$2 {print $0}' area9 > area109;

%%=====================================================================
% Load each file in MATLAB and assign the data to each node's matrix
  load area102
  load area103
  load area104
  load area105
  load area106
  load area107
  load area108
  load area109
```

```
y1=area101(:,2);
y2=area102(:,2);
y3=area103(:,2);
y4=area104(:,2);
y5=area105(:,2);
y6=area106(:,2);
y7=area107(:,2);
y8=area108(:,2);
y9=area109(:,2);
%%=============================================================================
```

E-3 Cost function

This is the example Cost function for some objectives, for other objectives the cost function is similar to these cases.

```
%%=====================================================================
% considering nodes with same time distance (0.25) to have equal effect
for all nodes. Also, banks are separated to evaluate the cost for each of
them separately.

mm=1;
m=1;
for mmm=0:.25:8.55
  while x(m)<=mmm-.0001
  m=m+1;
  end
  xx1(mm)=x(m);
  yy351(mm)=y35(m);
  yy981(mm)=y98(m);
  yy51(mm)=y5(m);
  yy61(mm)=y6(m);
  yy531(mm)=y53(m);
  YY1(mm)=Y(m);
  YYCT1(mm)=YCT(m);
  m=m+1;
end
mm=1;
m=1;
for mmm=0:.25:14.74
  while x(m)<=mmm-.0001
  m=m+1;
  end
  xx2(mm)=x(m);
  yy352(mm)=y35(m);
  yy982(mm)=y98(m);
  yy52(mm)=y5(m);
  yy62(mm)=y6(m);
  yy532(mm)=y53(m);
  YY2(mm)=Y(m);
  YYCT2(mm)=YCT(m);
```

```
  mm=mm+1;
end
mm=1;
m=1;
for mmm=0:.25:20.93
  while x(m)<=mmm-.0001
  m=m+1;
  end
  xx3(mm)=x(m);
  yy353(mm)=y35(m);
  yy983(mm)=y98(m);
  yy533(mm)=y5(m);
  yy63(mm)=y6(m);
  yy5333(mm)=y53(m);
  YY3(mm)=Y(m);
  YYCT3(mm)=YCT(m);
  mm=mm+1;
end
  mm=1;
m=1;
for mmm=0:.25:26.53
  while x(m)<=mmm-.0001
  m=m+1;
  end
  xx4(mm)=x(m);
  yy354(mm)=y35(m);
  yy984(mm)=y98(m);
  yy54(mm)=y5(m);
  yy64(mm)=y6(m);
  yy534(mm)=y53(m);
  YY4(mm)=Y(m);
  YYCT4(mm)=YCT(m);
  mm=mm+1;
end
mm=1;
m=1;
for mmm=0:.25:32.13
  while x(m)<=mmm-.0001
  m=m+1;
  end
  xx5(mm)=x(m);
  yy355(mm)=y35(m);
```

```
  yy985(mm)=y98(m);
  yy55(mm)=y5(m);
  yy65(mm)=y6(m);
  yy535(mm)=y53(m);
  YY5(mm)=Y(m);
  YYCT5(mm)=YCT(m);
  mm=mm+1;
end
mm=1;
m=1;

for mmm=0:.25:35.25
  while x(m)<=mmm-.0001

  xx6(mm)=x(m);
  yy356(mm)=y35(m);
  yy986(mm)=y98(m);
  yy56(mm)=y5(m);
  yy66(mm)=y6(m);
  yy536(mm)=y53(m);
  YY6(mm)=Y(m);
  YYCT6(mm)=YCT(m);
  mm=mm+1;
  m=m+1;
  end
end
%%=========================================================================
%Penalty for temperature profile for each bank for node 98

profilediff1=(YY1-yy981)'*(YY1-yy981);
profilediff2=(YY2-yy982)'*(YY2-yy982);
profilediff3=(YY3-yy983)'*(YY3-yy983);
profilediff4=(YY4-yy984)'*(YY4-yy984);
profilediff5=(YY5-yy985)'*(YY5-yy985);
profilediff6=(YY6-yy986)'*(YY6-yy986);
%%=========================================================================
% For cases which the temperature profile is compared with coiling
temperature error the cost is evaluated as follows

D =size(yyy98);
D1=size(yy981);
D2=size(yy982);
```

```
D3=size(yy983);
D4=size(yy984);
D5=size(yy985);
D6=size(yy986);

for i=1:D(2)
  sum=sum+abs(YYY(i)-yy98(i));
end
for i=1:D1(1)
  sum1=sum1+abs(YYY(i)-yy981(i));
end
for i=1:D2(1)
  sum2=sum2+abs(YYY(i)-yy982(i));
end
for i=1:D3(1)
  sum3=sum3+abs(YYY(i)-yy983(i));
end
for i=1:D4(1)
  sum4=sum4+abs(YYY(i)-yy984(i));
end
for i=1:D5(1)
  sum5=sum5+abs(YYY(i)-yy985(i));
end
for i=1:D6(1)
  sum6=sum6+abs(YYY(i)-yy986(i));
end

%%=========================================================================
% Coiling temperature error is evaluated as follows

Tempdiff=abs(((y53(end)-600)))
%%=========================================================================
% Percent volume of strip is evaluated using nodes, which have
temperature in the desired area

for T=(12-4.65):0.1:(13-4.65)
       while X(t)<=T-.0001
       t=t+1;
       end
       T_down=(640-657.8)*(T-(12-4.65))/8+657.8;
       T_up1=(702.2-657.8)*(T-(12-4.65))/8+657.8;
       T_up2=(651.25-663.35)*(T-(13-4.65))/2+663.35;
```

```
       if y4(t)>=T_down & y4(t)<=T_up1
       RES(4)=1;
       end
       if y5(t)>=T_down & y5(t)<=T_up1
       RES(5)=1;
       end
       if y1(t)>=T_down & y1(t)<=T_up1
       RES(1)=1;
       end
 end


for T=(13-4.65):0.1:(15-4.65)
       while X(t)<=T-.0001
       t=t+1;
       end
       T_down=(640-657.8)*(T-(12-4.65))/8+657.8;
       T_up1=(702.2-657.8)*(T-(12-4.65))/8+657.8;
       T_up2=(651.25-663.35)*(T-(13-4.65))/2+663.35;

       if y4(t)>=T_down & y4(t)<=T_up2
       RES(4)=1;
       end
       if y5(t)>=T_down & y5(t)<=T_up2
       RES(5)=1;
       end
       if y1(t)>=T_down & y1(t)<=T_up2
       RES(1)=1;
       end
 end

sum=0;
for i=1:500
sum=sum+RES(i);
end
cost=-sum
t=1;
%%=====================================================================
% coiling temperature constraint for hottest and coolest nodes
if M==6
  if y5(end)>600
```

```
    cost=10
  end
end
if y4(end)<500
  cost=10
end
Size=size(y4);
%%=====================================================================
% Constraint for temperature of all nodes to be higher than 450 during
the cooling process

for i=1:Size

if y4(i)<=450
  cost=10
end
for T=(12-4.65):0.1:(13-4.65)
  while X(t)<=T-.0001
  t=t+1;
  end
  T_down=(640-657.8)*(T-(12-4.65))/8+657.8;
  T_up1=(702.2-657.8)*(T-(12-4.65))/8+657.8;
  T_up2=(651.25-663.35)*(T-(13-4.65))/2+663.35;

  if y5(t)<=T_down
   cost=10;
  end
end
end
%%=====================================================================
```

The End