DEVELOPMENT OF MANUFACTURING EXECUTION SYSTEM OF STEEL MAKING PROCESS USING SIMULATION

MR. KIATKAJOHN WORAPRADYA

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF ENGINEERING
(INTEGRATED PRODUCT DESIGN AND MANUFACTURING ENGINEERING)
SCHOOL OF ENERGY, ENVIRONMENT AND MATERIALS
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2013

Development of Manufacturing Execution System of
Steel Making Process using Simulation

Mr. Kiatkajohn Worapradya M.Eng. (Electrical Engineering)

A Dissertation Submitted in Partial Fulfillment of the Requirements for
the Degree of Doctor of Engineering (Integrated Product Design and Manufacturing)
School of Energy, Environment and Materials
King Mongkut's University of Technology Thonburi
2013

Dissertation Committee

…… Chairman of Dissertation Committee
(Assoc. Prof. Peerayuth Charnsethikul, Ph.D.)

…… Member and Dissertation Advisor
(Assoc. Prof. Purit Thanakijkasem, Ph.D.)

…… Member
(Asst. Prof. Panya Srichandr, Ph.D.)

…… Member
(Asst. Prof. Siriporn Rojananan, Ph.D.)

…….... Member
(Julathep Kajornchaiyakul, Ph.D.)

Dissertation Title	Development of Manufacturing Execution System of Steel Making Process using Simulation
Dissertation Credits	36
Candidate	Mr. Kiatkajohn Worapradya
Dissertation Advisor	Assoc. Prof. Dr. Purit Thanakijkasem
Program	Doctor of Engineering
Field of Study	Program of Integrated Product Design and Manufacturing
Department	Materials Technology
Faculty	School of Energy, Environment and Materials
Academic Year	2013

Abstract

This dissertation develops a simulation-based optimization framework in Manufacturing Execution System (MES) of a steelmaking and continuous casting (SCC) plant. This framework incorporates 2 substantial functions; production planning and executing tasks. In the planning phase, 2 tiers for the steel scheduling are studied. The cast planning for continuous caster (CC) is firstly dealt and then the overall scheduling for SCC plant is studied.

For the cast planning, an optimization model reducing the production cost and tailored for a multi-strand CC is proposed. This model can effectively group, sequence, and allocate jobs. This optimization model is later demonstrated in 2 cases under uncertainty environment; 1) uncertain processing time and 2) uncertain disruptions. In the first case, a robust proactive-reactive scheduling is adopted. In this scheduling, the stability of the production due to the rescheduling is taken into account and the best worst case is used to be a performance criterion. An uncertainty model is created and validated. The simulation results show that the robust scheduling/rescheduling can be achieved under uncertain processing time via the optimization model with stability. In the case of uncertain disruption, the uncertain machine breakdown is characterized and modeled via an actual failure distribution. A worst case performance scheduling is also adopted. The result shows that the best worst case schedule also retains an acceptable performance when the worst disruption occurred. However, the best worst case schedule still provides much slack. In other words, it provides a rather conservative solution.

For 2nd tier of SCC production scheduling, a novel methodology that simultaneously integrates cast planning and steelmaking scheduling is proposed. A mix-integer optimization model and a hierarchical genetic algorithm (HGA) tailored particularly for searching an optimal schedule in this problem is presented. The optimization model is constructed by taking into account actual technological and economical restrictions, which add extra difficulty to the scheduling problem. The performance of the proposed

methodology is compared with a 2-phase traditional scheduling. Both qualitative and quantitative performance measures are investigated. This optimization model is also utilized in an unpredictable machine breakdown environment. Besides this optimization model, a new robustness based on probability distribution curve of a system performance is proposed. In this work, an artificial neural network (ANN) through Monte Carlo Simulation (MCS) is adopted for uncertainty assessment. With a challenge of designing an accurate ANN model due to the complexity from the discrete and nonlinear properties of the system performance, the model is achieved by applying *k*-mean clustering that can simplify the model complexity. The experimental result shows that the proposed methodology is successful to design a robust schedule that provides a lower production cost under an acceptable breakdown probability and also consumes less computational time compared with the traditional approach.

In the schedule execution phase, the optimal plan is used in the quality management or production optimization modules of MES. In this work, a machine setup of a spray cooling system in CC process is raised as a case study. To optimize the cooling pattern, both a solidification model and an objective function, which satisfies the metallurgical criteria and the resource consumption, are developed. Case studies on quality and productivity improvements are studied for an optimal solution. The results show that the effective maintain of the strand temperature with low water consumption can be achieved under an increased casting speed.

This dissertation investigates the framework that collaborates between the production planning and executing via simulations. Several simulation models that are designed in this work and case studies under a real situation show the potentials of MES with the framework. The system can streamline the SCC process from designing an effective schedule under uncertain disruption to execution of the process.

Keywords : Manufacturing Execution System / Optimization / Production Scheduling / Steel making / Uncertainty

หัวข้อวิทยานิพนธ์	การพัฒนาระบบสนับสนุนการผลิตสำหรับกระบวนการผลิตเหล็กด้วยการจำลองสถานการณ์
หน่วยกิต	36
ผู้เขียน	นายเกียรติขจร วรปรัชญา
อาจารย์ที่ปรึกษา	รศ.ดร.ภูริต ธนะกิจเกษม
หลักสูตร	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	การออกแบบและผลิตแบบบูรณาการ
สายวิชา	เทคโนโลยีวัสดุ
คณะ	พลังงานสิ่งแวดล้อมและวัสดุ
ปีการศึกษา	2556

บทคัดย่อ

ดุษฎีนิพนธ์นี้ ได้พัฒนารูปแบบการหาความเหมาะสมที่สุดด้วยการจำลองสถานการณ์ สำหรับระบบสนับสนุนการผลิตในอุตสาหกรรมผลิตเหล็กและการหล่อเหล็กแบบต่อเนื่อง รูปแบบ ดังกล่าวได้ประสาน 2 ฟังก์ชันที่สำคัญ ได้แก่ การวางแผนการผลิตและการดำเนินงานตามแผน ในการวางแผนการผลิตเหล็กจะประกอบด้วย 2 ขั้นตอนหลัก ดุษฎีนิพนธ์นี้เริ่มจากการวางแผนการหล่อเหล็กแบบต่อเนื่อง และต่อมาวางแผนการผลิตเหล็กรวมทั้งโรงงาน

สำหรับการวางแผนการหล่อเหล็ก ได้เสนอแบบจำลองการหาความเหมาะสมที่สุดที่ลดต้นทุนการผลิตและถูกสร้างให้สอดรับกับกระบวนการหล่อเหล็กแบบต่อเนื่อง การจัดตารางการผลิตแบบเชิงกำหนดได้พิสูจน์สมรรถนะของแบบจำลองว่าสามารถจัดกลุ่ม จัดลำดับ และจัดสรรงานได้อย่างมีประสิทธิภาพ แบบจำลองดังกล่าวได้พิสูจน์สมรรถนะกับกรณีสภาวะที่มีความไม่แน่นอน 2 กรณี ได้แก่ เวลาในการผลิตมีความไม่แน่นอน และมีการขัดจังหวะการผลิตที่ไม่แน่นอน ในกรณีแรก ใช้วิธีการจัดตารางการผลิตแบบคงทนชนิดโปรแอคทีฟ-รีแอคทีฟ โดยได้คำนึงถึงเสถียรภาพการผลิตเมื่อต้องจัดตารางการผลิตใหม่ด้วย และได้ประเมินสมรรถนะของ ตารางการผลิตโดยคำนึงถึงสมรรถนะแย่ที่สุด แบบจำลองความไม่แน่นอนที่ใช้ในการจำลองสถานการณ์ได้ถูกสร้างและยืนยันความถูกต้อง ผลการจำลองสถานการณ์ชี้ว่าสามารถหาตารางการผลิตที่มีประสิทธิผลภายใต้ความไม่แน่นอนได้ด้วยแบบจำลองความเหมาะสมที่สุดที่คำนึงถึงเสถียรภาพการผลิตที่ได้ออกแบบขึ้น ส่วนในกรณีมีการขัดจังหวะที่ไม่แน่นอน ได้พัฒนา แบบจำลองความน่าจะเป็นของการเสียหายของเครื่องจักรด้วยข้อมูลจริงขึ้น และได้ใช้ การจัดตารางการผลิตที่คำนึงถึงสมรรถนะแย่ที่สุด ผลการจำลองสถานการณ์ชี้ว่าตารางการผลิตที่คำนึงถึงสมรรถนะแย่สุด มีสมรรถนะที่รับได้เมื่อเกิดกรณีเครื่องจักรเสียหาย อย่างไรก็ตามตารางการผลิตดังกล่าวเผื่อเวลาล่าช้า ไว้มาก หรืออีกนัยหนึ่งเป็นการหาคำตอบที่ค่อนข้างมีความอนุรักษ์มาก

สำหรับการจัดตารางการผลิตรวมทั้งโรงงาน ได้เสนอวิธีการใหม่ที่รวมการจัดตารางการหล่อและการผลิตเหล็กในขั้นตอนเดียว ซึ่งได้เสนอแบบจำลองการหาความเหมาะสมที่สุด ที่ใช้ร่วมกับขั้น ตอนวิธีทางพันธุกรรมแบบลำดับชั้นที่ออกแบบอย่างเฉพาะสำหรับปัญหานี้ แบบจำลอง ดังกล่าวถูกสร้างโดยคำนึงถึงข้อจำกัดทั้งด้านเทคโนโลยีการผลิตและทางเศรษฐกิจ ซึ่งจะเพิ่มความยากให้แก่ปัญหาการจัดตารางการผลิต สมรรถนะของวิธีการใหม่นี้ได้ ถูกเปรียบเทียบกับกรณีการจัดตารางการผลิตแบบเก่าที่แยกการจัดตารางเป็น 2 เฟส โดยการวัดเชิงคุณภาพและปริมาณ แบบจำลองการหาความเหมาะสมที่สุดนี้ ยังได้นำไปใช้ในสภาวะที่เครื่องจักรเกิดความเสียหาย อย่างคาดการไม่ได้ นอกจากแบบจำลองดังกล่าวดุษฎีนิพนธ์นี้ได้เสนอการจัดตารางการผลิตที่ใช้ตัวชี้วัดความคงทน ที่อิงกับเส้นโค้งการกระจายความน่าจะเป็นของสมรรถนะของระบบ และใช้โครงข่ายประสาทเทียมในการสร้างแบบจำลองสำหรับประเมินความไม่แน่นอนนี้ ด้วยความท้าทายในการสร้างแบบจำลองให้แม่นยำสำหรับปัญหาที่มีความไม่ต่อเนื่องและไม่เชิงเส้นสูง พบว่าการสร้างแบบจำลองประสบความสำเร็จจากการใช้เทคนิคการแบ่งกลุ่มแบบ เค -มีน ในการลดความซับซ้อนของระบบลง วิธีดังกล่าวประสบความสำเร็จเป็นอย่างดีในการออกแบบตารางการผลิตแบบคงทน ที่สามารถลดต้นทุนการผลิตภายใต้เงื่อนไขที่ยอมให้มีกรณีเครื่องจักรเสียหายได้ด้วยความน่าจะเป็นที่ต่ำ และยังใช้เวลาในการคำนวนน้อยกว่าวิธีดั้งเดิมด้วย

ในส่วนของการดำเนินงานตามแผนการผลิต ตารางการผลิตที่เหมาะสม จะถูกนำมาปรับตั้งเครื่องจักรโดยโมดูลการบริหารคุณภาพหรือโมดูลควบคุมการผลิตในระบบสนับสนุนการผลิต ในดุษฎีนิพนธ์นี้ได้ใช้การปรับตั้งอัตราการหล่อเย็นแบบฉีดน้ำในโรงหล่อเป็นกรณีศึกษา การหาค่าเหมาะสมทำได้โดยอาศัยแบบจำลองการแข็งตัวของโลหะที่พัฒนาขึ้นด้วยระเบียบวิธีผลต่างสืบเนื่อง และฟังก์ชันจุดประสงค์ที่กำหนดขึ้นโดยอิงเกณฑ์ทางโลหะวิทยาและการใช้ทรัพยากร ในงานนี้ได้ยกกรณีศึกษาการปรับปรุงคุณภาพและผลิตผล และได้หาคำตอบด้วยขั้นตอนวิธีทางพันธุกรรม ผลการจำลองสถานการณ์แสดงให้เห็นว่าสามารถควบคุมอุณหภูมิของผลิตภัณฑ์ได้อย่างมีประสิทธิภาพ ขณะที่สามารถใช้ความเร็วในการหล่อที่เพิ่มขึ้นได้

โดยสรุป ดุษฎีนิพนธ์นี้ศึกษารูปแบบระบบ ที่ประสานงานร่วมกันระหว่างการวางแผนการผลิตและการดำเนินงานตามแผนผ่านการจำลองสถานการณ์ แบบจำลองที่ได้ออกแบบขึ้นและกรณีศึกษาต่างๆ ภายใต้สถานการณ์จริงในงานชิ้นนี้ ได้แสดงให้เห็นถึงศักยภาพของระบบสนับสนุนการผลิตที่ใช้รูปแบบการประสานงานดังกล่าว ว่าสามารถปรับปรุงประสิทธิภาพของกระบวนการผลิตเหล็ก ตั้งแต่ขั้นตอนการจัดตารางการผลิตที่มีประสิทธิผลจนถึงการดำเนินงานตามแผน

คำสำคัญ : การจัดตารางการผลิต / การผลิตเหล็ก / การหาความเหมาะสมที่สุด / ความไม่แน่นอน / ระบบสนับสนุนการผลิต /

ACKNOWLEDGMENTS

I am specially indebted to my advisor, Assoc. Prof. Dr. Purit Thanakijkasem, for his great effort and useful comments on all works during my study in D.Eng. program at KMUTT. I would like to thank all committee members: Assoc. Prof. Dr. Peerayuth Charnsethikul, Asst. Prof. Dr. Panya Srichandr, Asst. Prof. Dr. Siriporn Rojananan, and Dr. Julathep Kajornchaiyakul to participate in the committee and provide invaluable suggestions and comments.

I would like to thank the Higher Education Research Promotion and National Research University Project of Thailand, Office of the Higher Education Commission for partial support in many projects. I wish to thank my friend and Bookdose Co., Ltd.'s studio for place and some facilities.

Finally, I wish to thank my family and parents for their encouragement and belief in my ability to finish what I had started.

CONTENTS

LIST OF TABLES

LIST OF TABLES (Ext.)

LIST OF FIGURES

Figures **Page**

LIST OF FIGURES (Ext.)

Figures **Page**

LIST OF FIGURES (Ext.)

Figures **Page**

CHAPTER 1 MOTIVATION AND FRONT-END STUDY

Iron and steel sector is one of the most important industries due to being the upstream for industrial chain. More than two decades it has been a powerful symbol of an increasingly global symbol market economy (Lee et al., 1996). Current trend in the market has made medium and compact steel factories turn to mass customized production. In this mass customized production, an improvement on quality and productivity is the key to success. Consequently, a migration from conventional factory-floor control strategies to effective collaborative manufacturing management and control system is necessary (Leitão et. al., 2005). Manufacturing Execution System (MES) plays an important role to bridge the data gap between Enterprise Resource Planning (ERP) system and the factory-floor control level (Siemens, 2006).

In the mid-1990s, the concept of MES is developed in the USA to support above exigencies. A non-profit organization with the name of MESA (Manufacturing Execution System Association) firstly started standardizing MES and later ISA- S95 standard (of a committee consisting of about 200 users and manufacturers) has been approved. However, there are a few researches that study on MES in steel industry after 1990. Furthermore, due to complexity and special requirements of the steel production, the steel industry needs MES that tailored for them.

To develop MES for steel industry, this dissertation starts from surveying the possible opportunities to be a commercial product. The opportunities which satisfy the problems and are suited to the author's capabilities will be selected and guided to a statement of the hypotheses objectives in this work.

1.1 Front-End Study

In order to achieve the desired product, Front-End study is adopted. The front-end process began from an opportunity scanning, which studies on the industrial structure, market trend, and research trend. The potential opportunities from the study are assessed for developing objectives and product design specifications. A full report of the front-end study appears in Worapradya (2006) and is summarized here.

1.1.1 Opportunity Scanning

Although MES is standardized in the mid-1990s, it is not developed to be a product before the mid-2000s. Therefore the opportunity scanning is on a well-known area such as the integrated control system or automation areas, which cover functions of MES.

1.1.1.1 *Industry Study*

- **Current competitive situation:** A supply chain of the automation and integrated control system industry (including MES) is shown in Figure 1.1. It involves with 3 industries; 1) Industry of software tools 2) Industry of automation system and instrument 3) System integrators (SI) on control system and operations management.

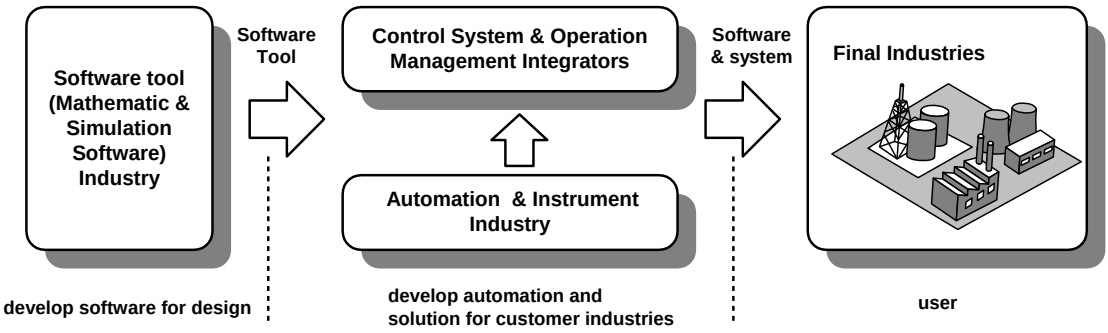


Figure 1.1 Supply chain and relevant industries

Focusing on the players in the segment of SI that develops MES supported steel industry, it can be classified as follows:

- *Enterprise software developers:* This group is the companies that expert in operations management solution software, e.g., Wonderware or SAP.
- *Factory automation developers*: This group comes from the companies that expert in automation system and control devices. MES software from these developers is full of the complex process models that are compatible with their automation system or devices. Examples of these groups are SIEMENS and Yogokawa.
- *Small system integrators*: This group is the small companies that own no software and automation systems while produce the customized MES from integrating above software products.

Currently, most of MES in steel industry comes from the first and second groups. The market leaders are SIEMENS and Wonderware. Competitive behavior of these developers consists of 2 phases:

- *New product installation phase*: In this phase, the price competitive is not essential but satisfying needs and improving competency for customers are a key to success.
- *Modernization phase*: A new system is not a customer need on this period of time but they just need to upgrade some obsolete parts and plan to have

the new system on next period. In this situation, the pricing below the competition is a key to success for this phase.

- **Competitive strategy:** Examples of the competitive strategies of each developer are analyzed and summarized in Table 1.1.

Table 1.1 Competitive strategy in MES industry

Organization	Competitive strategies
SIEMENS	– SIEMENS positions their product and company to be expert in factory automation system. MES of this developer provides more engineering and special functions or models for optimizing the production in many industries, e.g., paper, steel, and auto part industries. – Focusing on the customers in areas of large factories and selling the overall solution system including MES and the compatible automation system. – Providing the outsourcing service and the system maintenance service for factories.
Wonderware	– Wonderware clearly positions their product and company to be expert in the operations management and human-machine interface (HMI). – Focusing on the customers in areas of small or medium factories and selling MES to replace the existing system.
Other small system Integrators	– Selling by customer lifetime value strategy and personal contact. – Focusing on niche market, e.g., building automation and amusement parks.

1.1.1.2 *Market Study*

Market survey of the control system integrator organizations is appeared in Boyes (2002). The market volume and trends are summarized below.

- **Volume of MES and control system integrators**: The Control System Integrators Association (CSIA) predicted worldwide gross revenues for professional control system integrator companies to be US$12 billion per year (in 2001), based on a total of 2200 companies that meet the CSIA definitions and qualifications. The market for control system integration (including MES) is

growing at a rate of 9% to 10% per year. Control Systems Integrators are experiencing rapid growth in Latin America and Asia, as well as in North America. The market volume at 2001 is shown in Figure 1.2.

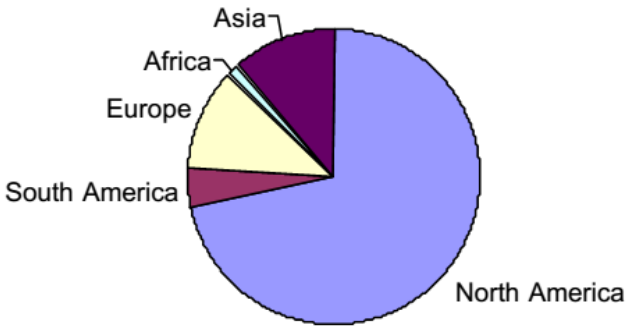


Figure 1.2 Market volume in 2001

- **Current market trend:** Control system integrators have increasingly shifted from machine control and plant floor process control to Enterprise Integrators (MES level). It becomes increasingly important to tie the plant floor systems to the business systems of the enterprise. Other significant trends in the global automation market and system integration are as follows:

  - Consolidation: The major automation and control vendors and the major Architectural Engineering (AE) firms have been undergoing severe consolidation since the early 1990s.
  - Competition from manufacturers and software vendors: The success and growth of the market for control system integration to a large worldwide value urge software vendors to establish their own integration divisions.
  - Providing proprietary value-add: One way that the control system integrators are combating an encroachment by the automation and control vendors is by providing proprietary value-added features and services. At the same time, it will be differentiating themselves from other system integrators. For example, several high end MES are now providing proprietary templates and other proprietary tools and methodologies to their clients.
  - Outsourcing of projects by manufacturers, especially MRO projects: Because of the downsizing of plant engineering and maintenance engineering personnel at industrial enterprises, the role of control system integrators has increased dramatically. The industrial enterprises outsource these MRO (Maintenance, Repair and Operations) projects to a control system integrator, who acts as the plant or maintenance engineering department.
  - Increasing importance of information technology (IT) in the decision making process for system integration projects: There is a convergence

between the process engineering function and the enterprise IT function. Previously, integration projects rarely left the factory floor, they now often include MES integration into the enterprise business systems. Thus, it is becoming more common for IT to be in at least partial charge of integration projects, rather than plant engineering.

- **Future trend:** Several trends are developing for the future of the control system integrator market that are worth watching:
  - Outsourcing of projects by manufacturers, especially MRO projects.
  - Continuing growth in MES and enterprise integration.
  - Supply chain consolidation.
  - Proprietary tools and value added by system integrators.
  - Increased marketing by automation manufacturers through system integrators.
  - Competition with system integrators by HMI and software/hardware manufacturers.
  - Increasing importance of IT in the decision making process for system integration projects.
  - Open systems.

1.1.1.3 *Research Trend*

Overview of research trends in the control system integration, automation, and enterprise integration are reviewed in Murray et al. (2003), SanZ and Årzén (2003), and Qiu and Zhou (2004). This section concludes the significant research trends of the control system integration that involves with MES as follows:

- *Controls in distributed and network environments*: A distributed manufacturing system, which distributes controls to many units and separates the responsibility to each individual, is expected to be a new paradigm. Another well-known name is the Holonic or Agile Manufacturing System.
- *Applying expert systems and Artificial Intelligent (AI) techniques*: The expert and complex decision systems will play important role in the automation and operations management systems. It is not only using the traditional logic but AI is also applied in the large-scale system.
- *Automatic synthesis with integrated validation and verification*: The concept of virtual manufacturing used for the effective designing and redesigning tools. This concept developed models for simulations directly collaborated with hardware, including uses for validation and verification tasks.
- *Graphic User Interface (GUI) development*: The concept of applying graphical techniques and animations in real time will be adopted for developing the automation and operations management systems.

1.1.2 Opportunity Assessment

The opportunity assessment or opportunity recognition is a process which assists and selects competitive opportunities from innovation ideas and academic challenge issues. This process uses the potential ideas or opportunuties from the front-end study to make a decision for selecting the optimal opportunities. The potential opportunities fitted for steel industry are chosen and summarized as follows:

Issues from industrial study:

- ***Development of core MES functions to satisfy special industry:*** In order to make competitive advantages in the new product installation phase, the core functions of MES, i.e., production planning and optimization should be improved an efficiency and tailored for steel industry.

Issues from market study:

- ***Development of MES by providing proprietary value-add***: Special value-added features should be provided for steel industry.
- ***Development of MES to the open system*****:** It focused on the development of software and hardware interface programs for MES to support the control system hardware while producers are going to develop the open system.

Issues from the research trend:

- ***Development of MES based on AI techniques*****:** Several researches clearly demonstrated that MES can be efficiently developed in the logic, control strategies, and decision making system by using AI.
- ***Development of MES based on simulations*****:** Several literatures coincidently mention that simulations are the high performance tools for reducing cost and time periods of the system design and the development.
- ***Development of UI of MES*****:** A friendly user interface is a strategy which the local developers make competitive advantages.

The potential opportunities above have to be analyzed and developed because the advantages on the most of ideas are not clear enough to visualize the product concept. An experimenting technique, which is the combination of the advantages on each idea, is introduced by Luecke and Katz (2003). With this technique, it firstly seeks out the difference of ideas and clusters them. In this case, the preliminary ideas are clustered into 2 types; the functions/systems and the techniques. Then, the matrix is created as shown in Table 1.1 and is filled by matching the advantages of each idea. The problem solving is to improve the operations management and to reduce in cost throughout life cycle. The result is shown on Table 1.2.

Table 1.2 Matrix of idea combination

		Functions and Systems			
		Develop core functions	Develop UI	Develop value-add features	Develop to opened system
Techniques	Use AI	*Match*	*Mismatch*	*Match*	*Mismatch*
	Use Simulation	*Match*	*Match*	*Match*	*Mismatch*

The new ideas, which are easily illustrated on the product concept, are concluded the following:

- **Case I: *Development of MES based on AI*:** Using AI to develop MES in steel making industry. Core functions of MES such as a decision making system, planning/scheduling and production optimization will be improved to be an expert or smart systems by using AI techniques (e.g., holonic technique, neural network, Genetic algorithms and so on).

- **Case II: *Development of MES based on simulations*:** Using simulations to support the production management for users. The simulation may be utilized for 2 purposes; 1) user utilizes simulations to assist the planning/scheduling task 2) user uses simulations to assist the interface with the production optimization and supervisory control.

To translate the new idea to be the optimal opportunities, Kim and Mauborgne (2000) introduced *the buyer utility map*, which mentioned the customer attractive of innovation concepts or new products. To use this map, the guideline questions to assess the opportunities are given as follows:

1. Which parts can we do the most obvious utilization to customers? And do our innovation ideas support doing that or not?
2. Those utilities are more or less than the ones proposed on products or services of competitors.
3. What is the most important utility for customers?
4. How do we adapt the ideas on our products to bring the most utilization on the most important section to customers?

The table is created to answer above questions. It consists of 6 parts of the utilization for customer (on the left column) and the buyer experience cycle (on the top row). The

highlighted zones are the most obvious utilization parts for customers and depend on the expected product. It is the answer on the 1st and 3rd questions. Then, the detail of the utilization is specified. The results are shown in Tables 1.3 and 1.4 for Case I and II, respectively.

Table 1.3 Buyer utility map for Case I

	Purchase	Delivery	Use	Supplements	Maintenance	Disposal
Productivity			*Improve performance of production management and control system*			
Simplicity			*Easy to operation*			
Convenience			*1) Assist to solve complex problems and optimizations* *2) Reduce time of decision making (expert system case)*			
Risk						
Fun and image						
Environmental friendliness						

Table 1.4 Buyer utility map for Case II

	Purchase	Delivery	Use	Supplements	Maintenance	Disposal
Productivity		*Reduce commissioning time*	*Improve performance of production management*		*Reduce maintenance period*	
Simplicity						
Convenience		*Assist fine tuning and system configuration*	*Use simulation as a tool for optimizing operation management*		*Assist testing and modification*	
Risk			*Training operators*		*Training technicians*	
Fun and image						
Environmental friendliness						

By considering the highlighted zone, it is clear that Case I is emphasized on the performance of the ‘use’ part of MES while Case II emphasizes throughout the life cycle. Therefore, Case II can be more useful for customers and better satisfy the problems than Case I (this statement answers the 2nd question, that it can satisfy the problems). In addition, the result also mentions the answer for the 4th question, which is adapting the rough ideas to the new product concept which can make the most obvious utilization to customers. Therefore, development of MES for steel making process via simulations is the optimal opportunity, which leads to a statement of the objectives in the next section.

1.2 Objectives

This dissertation presents the development of the manufacturing execution system (MES) for a steel making process. By an effective collaboration of MES and simulation, the production scheduling can reduce the production costs and increase productivity while the executions can improve the product quality.

CHAPTER 2 THEORY AND LITERATURE REVIEW

2.1 Overview of Steel Making Process

The steel making-continuous casting (SCC) production normally consists of 3 stages: (1) melting, (2) refining, and (3) continuous casting as shown in Figure 2.1. In the melting stage, C, S, Si, and other impurity contents of molten iron are reduced to a desirable level by burning with oxygen in an electric arc furnace (EAF). The output from this stage is the molten steel with the main alloy elements. The molten steel is transported by a crane to a refining furnace (RF) for refining. The operations at this stage further refine the chemicals and eliminate impurities in the molten steel and/or add the required alloy ingredients.

After refining, the molten steel is delivered to the continuous caster (CC) with a refractory line steel pot, called a ladle, and is poured into a tundish for casting. In the casting stage, the molten steel flows down from a hole at the bottom of the tundish into the crystallizer of a continuous caster. The molten steel continuously solidifies into a strand at the bottom of the caster. The strand is straightened and directed toward shear machines for cutting into steel slabs.

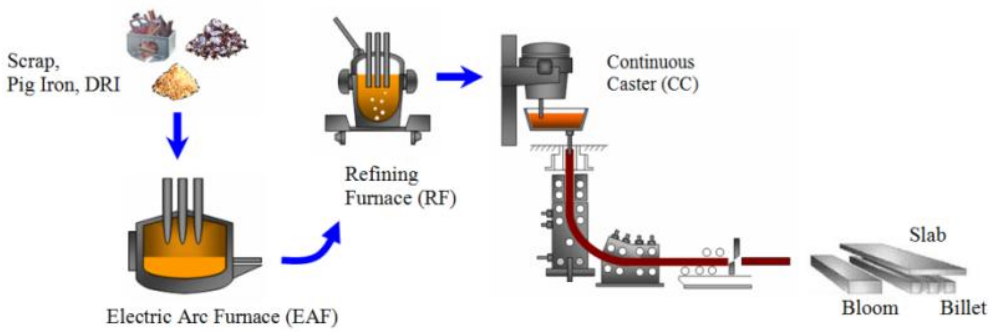


Figure 2.1 The steel production

In some technologies for steel production, the slab will be sent to the rolling mill shop (for producing a steel coil). For instance, a compact strip production (CSP), the production process links the melt shop, caster, rolling mill areas, and allows for steel to be melted, cast, and rolled in a continuous manner. There are no inventory buffer between the melt shop, caster, and rolling mill shop. Thus, this is a very interesting issue in operations management since the CC process is the bottle-neck of this steel production. The effective scheduling of this process is thus critical to productivity improvement of the entire production systems (Tang et al., 2002).

2.2 Manufacturing Execution System

Computer systems have been used to manage manufacturing operations for more than three decades. In modern manufacturing, an information system that can provide capability for integrating real-time production data available on shop-floor with the business planning and management functions at an enterprise level is required. A manufacturing execution system (MES) is a system that bridges the gap between the planning system and the control system using on-line information to manage manufacturing resources: people, equipment and inventory (Michael, 2000). Both theoretical and practical manufacturing execution systems are described in this section.

2.2.1 MES in ISA SP-95 Standard

To clarify the role of MES, the ISA SP-95 control hierarchy model (Enterprise-Control System Integration specification; an international standard agreed upon by a consortium of manufacturers, system suppliers, and opinion leaders) is illustrated in Figure 2.2(a). In the figure, ISA SP-95 contains 3 main levels; enterprise resource planning level (ERP), manufacturing execution system level (MES), and the factory floor control level. ERP (Level 4) defines business planning and logistics, plant production scheduling, operational management, etc. while the factory control level (Levels 1 and 2) includes batch, continuous, and discrete control. MES bridges ERP and factory control by making an optimized production schedule, which is carried out by the factory control level. The core functions of MES in ISA SP-95 can be summarized as follows:

1. *Resources management*: To manage and monitor resources such as machines, tools etc.,
2. *Operation and Detail scheduling*: To provide sequencing based on priorities, attributes, characteristics, and production rules,
3. *Dispatch production unit*: To manage flow of production units in terms of jobs, orders, batches, lots, and work orders,
4. *Document control*: To control records/forms, including work instructions, part programs, and so on,
5. *Data collection/acquisition*: To provide interface links and collect the production data from shop floor (including the data analysis for quality management),
6. *Labor management*: To provide the status of personnel in and up-to-the-minute time frame,
7. *Quality management*: To provide a real time analysis of measurements collected from the production to assure a proper product quality control,
8. *Process management:* To monitor production and provide a decision support,
9. *Product tracking*: To track and record the status of product and/or production,
10. *Performance analysis*: To provide up-to-the-minute report on resource utilization and availability, product unit cycle time, conformance to schedule, and performance to standards, and

11. *Maintenance management*: To track and direct the activities to maintain equipment and tools to insure their availability for manufacturing.

MES is not only widely used to support the production line but also extended to the product design and development cycle. According to the activity models in Feng (2000), the role of MES is to electronically link the information from CAD/CAM and process planning to the machine execution.

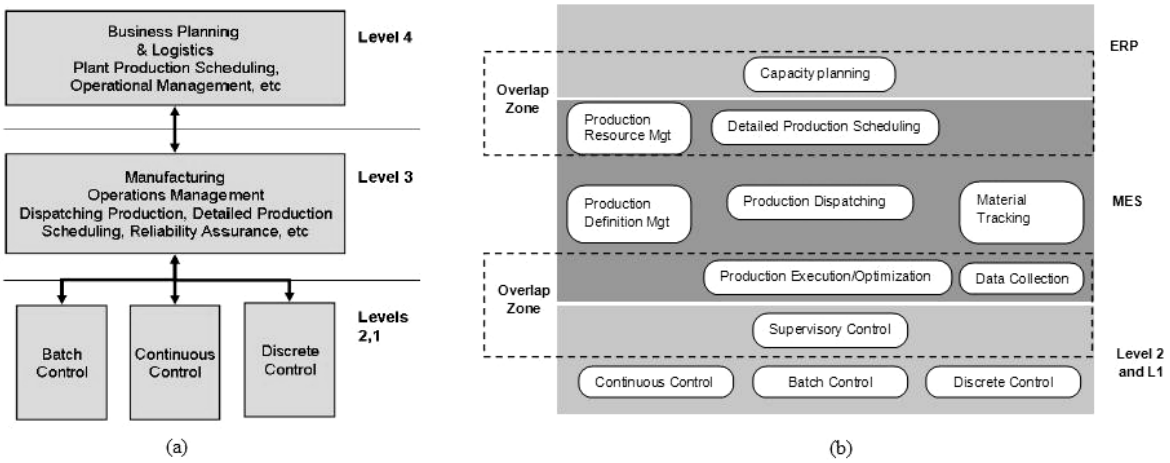


Figure 2.2 (a) ISA SP-95 control hierarchy (Siemens, 2006) and (b) Overlapped function n zone

2.2.2 MES in Steel Industry

Although the standardizations were achieved, however, the implementation of MES often is depended on the production types. Currently, two most important types of the practical MES are found; MES for process industry and assembly industry (Kletti, 2007). In the first type, the production optimization and control system will form a great part of MES. In the second type, MES is mostly an information management system. Therefore, MES has often the overlapped functions with other manufacturing systems, especially, between ERP and automation layers as shown in Figure 2.2(b).

Steel industry MES frequently belongs to the first type, which emphasizes the production optimization and the control strategies supporting the complexity of the steel production. In addition, since the steel enterprises are characterized as a multi-location enterprises environment, some functions of MES may be distributed on the sub-plants (Liu et al., 2007). An example is illustrated in Figure 2.3, which is the steel automation system of Voest-Alpine Industrieanlagenbau GmbH & Co (VAI) at Saudi Iron and Steel (Bayer et al., 2000). VAI automation system consists of 4 levels. Level 1 is the process control systems. Level 2 provides the supervisory control and the production optimization based on the metallurgical and physical models. Level 3 covers MES that mainly includes the production planning and control and Level 4 is Management

Information System (MIS). As seen in the figure, some functions of this MES overlap with Level 2, especially, in the production optimization.

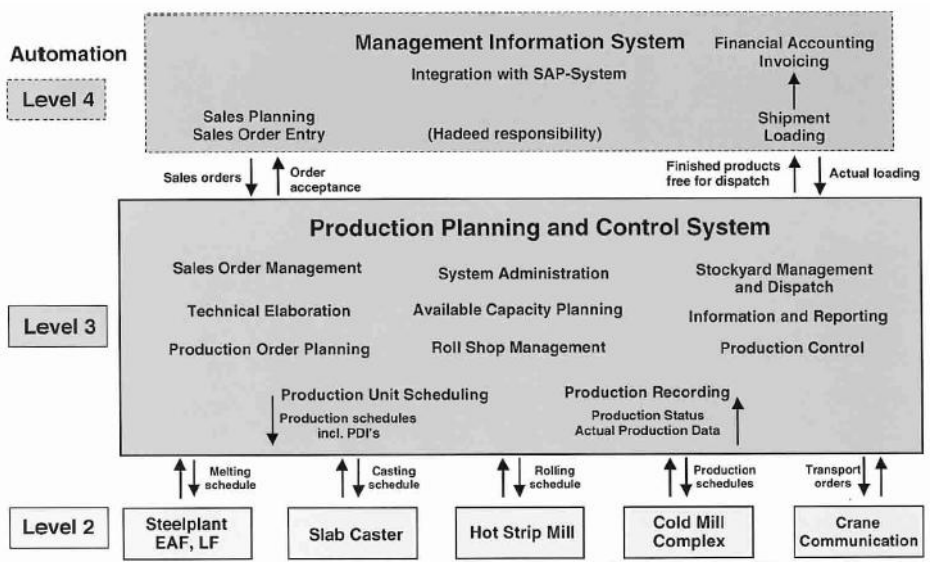


Figure 2.3 An VAI automation in a steelmaking factory (Bayer et al., 2000)

2.3 Production Scheduling

Scheduling is defined as the determination of relative position of jobs with respect to a processing machine, including the assignment of definite times at which processing occurs (Nawaz et al., 1983). In production scheduling, it aims to maximize the efficiency of the operation and reduce costs. The basic problem of scheduling is assigning *n* jobs into *m* machines. Cheng and Sin (1990) suggest 3 practical issues concerned with scheduling jobs on a set of machines: 1) What machine should be allocated to which job? 2) How to sequence the jobs in order to obtain the best schedule and meet the constraints? 3) How can a schedule be rationalized? The more complex scheduling problem will be formulated into several standard decision models such as flow shop scheduling, flexible flow shop job, and shop scheduling. Since the steel production is characterized as the flexible flow shop production, the overview of flow shop scheduling and the state of art of solving techniques are described below.

2.3.1 Flexible Flow Shop Scheduling Problem

A flexible flow shop scheduling (FFS) described in Sethanan (2001) is a generalization of the flow shop and the parallel processor environments. A flexible flow shop is alternatively called a multi-processor flow shop. In the most general form of a flexible flow shop environment, there are multiple stages (*S* stages), each stage consists of parallel processors ($m(s)$, where $s = 1,2,...,S$) as shown in Figure 2.4. The processors in each stage may be identical, uniform, or unrelated. Machines are uniform if the time to process a job on any machine is a constant ratio of its processing time on other machines. In other words, uniform machines are identical processors that do not have equal speeds. Unrelated machines are machines for which the time to process a job on any machine has no particular relationship of its processing time on any other machine (Cheng and Sin, 1990). In a FFS environment, each job is processed first at stage 1, then at stage 2, and so on. Normally, a job requires only one machine at each stage and any machine can process any job.

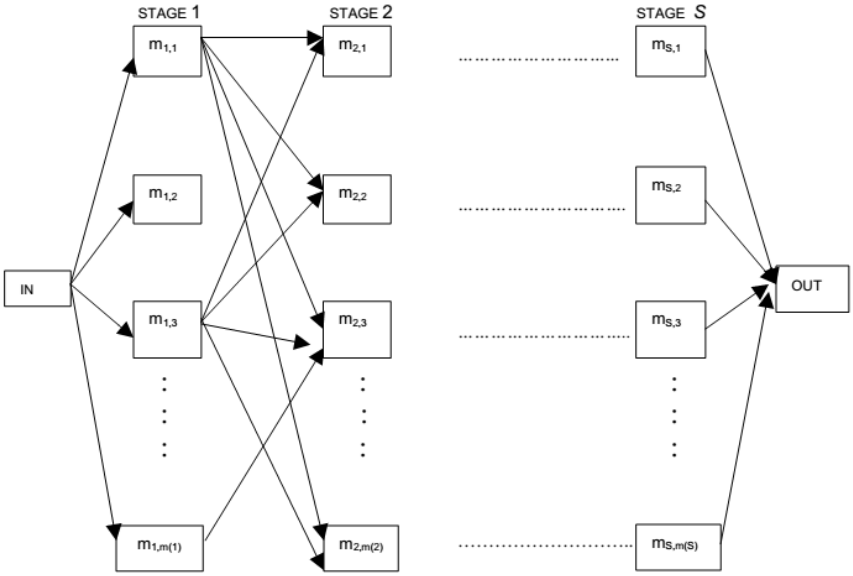


Figure 2.4 A Schematic of a flexible flow shop production (Sethanan, 2001)

2.3.2 Scheduling Techniques in Flexible Flow Shop

Several techniques for the flexible flow shop are described in Aufenanger et al. (2009). It is summarized here:

2.3.2.1 *Optimization Techniques*

Branch-and-bound approach approach is a technique that is suggested for the flexible flow shop environments by many researchers (Wang, 2005). The branch-and-bound approach (the details is given in Brucker (2007)) avoids enumerating all possibilities by calculating upper and lower bounds in order to prune possible solutions which cannot be the optimum as early as possible. Branch-and-bound techniques are methods for implicit enumeration. Practically, it is faster than completely enumerating the possible solutions. Nevertheless, their worst-case run-time is not better.

2.3.2.2 *Heuristics*

A basic heuristic for solving flexible flow shop scheduling problems can be derived from the branch-and-bound approach (Blazewicz et al. (2007): If not all generated branches of a decision tree are considered as starting points for a further search for better solutions, the width of the tree and hence the complexity of the search will be significantly reduced. However, the solution quality strongly depends on the right selection criteria for branches to explore. An entire class of heuristics for solving the flexible flow shop scheduling problems is based on a different idea. For example, Sriskandarajah and Sethi (1989) divide a flexible flow shop problem in two kinds of sub

problems: 1) A number of regular flow shop problem are built. Each of the machines on each stage of the flexible flow shop is then assigned to one of these regular flow shops. 2) Algorithms which are designed for solving regular flow shop scheduling problems can be applied. More sophisticated heuristics are needed when considering flexible flow shops with uniform machines. Some papers such as Kyparisis and Koulamas (2006) suggest building partial schedules for each production stage by using techniques which are designed for parallel machines problems and then combining these to a complete schedule for the original flexible flow shop problem.

2.3.2.3 *Artificial Intelligence Techniques*

To improve the solution quality, the heuristics can be enhanced by utilizing artificial intelligence techniques. The basic idea is to imitate the human process of decision making, which is often based on experience. This can be achieved by machine learning methods which create knowledge bases. A knowledge base contains decisions and the observed outcome together with descriptions of the situations while the decisions were made in. Scheduling methods can exploit such a knowledge base by comparing the current situation with the situations stored in the knowledge base and determine which decision is the best. Techniques following this approach are usually fast in decision making but they need extra time prior to the production process to build a knowledge base.

2.3.3 Production Scheduling in Steel Plant

The production scheduling is different from the assembly manufacturing because of the special requirement and technological constraints of the steel plant. A guideline of the practical production in the steel plant is described below.

2.3.3.1 *Steps of Production Scheduling*

A steel plant, which is focused in this problem, is a multi-production line characterized as FFS as shown in Figure 2.5. The steel production scheduling problem is basically evaluated by four steps as shown in Figure 2.6. Cast sequencing is firstly arranged. Sub-schedules and rough schedule are then established. Finally, the whole schedule is formed by taking into account the availability of machines at all stages. A brief description for each of these steps summarized by Tang et al. (2000) is given below.

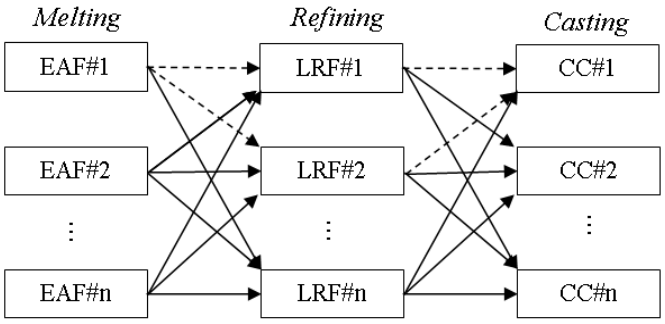


Figure 2.5 Schema of multi-production line

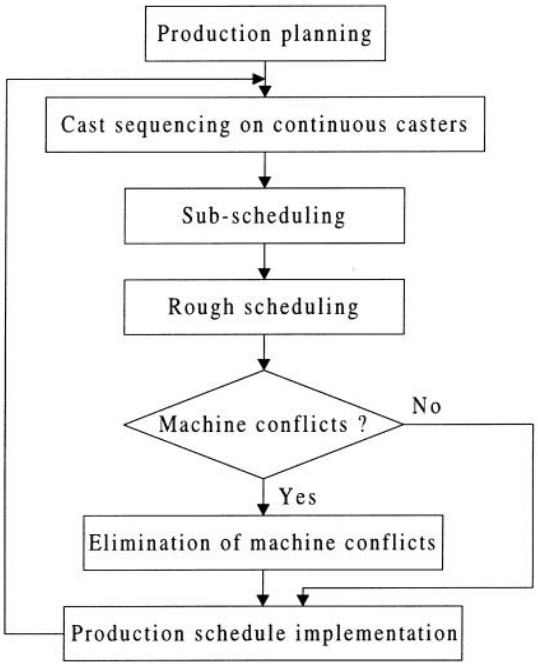


Figure 2.6 Steps of SCC scheduling (Tang et al., 2000)

1. *Cast sequencing*: Cast sequences on the casters and charge sequence in each cast are determined in this step based on their delivery times. These can be considered as single machine sequencing problems without resource constraints.
2. *Establishment of sub-schedules*: After a cast has been established, a cast job timetable called sub-schedule is formed for each cast in this step according to time progress of the operations including steelmaking, refining and continuous casting in each charge.
3. *Establishment of a rough schedule*: In this step, the sub-schedules with relative times are combined to produce a rough job schedule (superposition of sub-schedules) with physical time.
4. *Elimination of machine conflicts*: Because only the machine operation conditions in one cast are taken into account, machine conflicts often exist in the rough schedule. Only when machine conflicts are completely eliminated, a

feasible and practical schedule can be established. This step is to produce an optimal schedule in which all machine conflicts are eliminated.

2.3.3.2 *Summarized Requirements and Constraints in Steel Plant*

In steel production scheduling, two terms are introduced for the planning problem as follows:

- ***Charge*** (or a job of SCC scheduling) is a basic unit of steel making production.
- ***Cast*** is a set of charges casted continuously on the same continuous caster and having a similar chemical composition.

The process flow of Cast in the parallel production line is illustrated in Figure 2.7. The horizontal axis is for time and each line stands for machine. The optimization model in scheduling problem will be created under the following assumptions and constraints based on practical requirements.

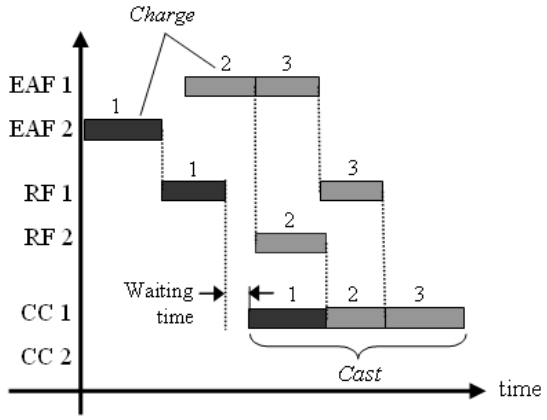


Figure 2.7 Process flow of a *Cast*

Assumptions in SCC Problem

- All charges must go through melting, refining, and then continuous casting, subsequently. At each stage, a charge (job) can be processed on any machine.
- Only three kinds of machines, i.e., (1) EAF, (2) RF, and (3) CC, are considered.
- Each machine cannot process more than one charge at the same time.

Constraints

- The number of charges in a cast sequence is defined by the life time of the nozzle at the bottom of a tundish.
- The machine setup time of each cast is required in practice to change equipment.
- Each charge must be completely processed in operation before preceded to the next sub-sequential operation.
- All charges processed on the same machine must be handled separately and subsequently.

Special Requirements

The main objective chosen for the SCC scheduling model is mainly to minimize the production cost through ensuring production continuity and JIT delivery.

- % copper between the two first charges should be minimal.

In the casting stage, several losses must be considered:

- Loss from steel grade difference in adjacent charges: a slab mixed with another steel grade will be assigned to the second grade product.
- Loss from width and thickness change: the material between the changes of charges will be disposed of.
- Loss from consignment date; e.g., inventory and compensation to customers and shipping.
- Loss from cast break: the cast break causes the remaining of molten steel to be re-heated or drained out.
- Loss from waiting time: the drop of the molten steel temperature from the waiting time affects the quality of steel.

2.4 Deterministic Optimization using GA

Since the gradient-based optimization is not practical to find an optimal solution for the scheduling problem with the binary or integer programming model, GA is adopted in this flexible flow shop problem (Bierwirth and Mattfeld, 1999; Reeves, 1995). In a real-life process, GA can provide a good solution on a wide range of problems and can be easily modified with respect to the objectives and constraints (Bürvenich, 1999). With GA, the procedure of the proposed methodology is given in Fig. 9.

Many optimization problems in real life do not satisfy convexity conditions and usually have large combinatorial explosions with discontinuous space. With these difficulties, Genetic Algorithm (GA) has been receiving increased attentions. GA is introduced by Holland (1975) in the mid-seventies. It is a search heuristic that is routinely used to generate useful solutions to optimization and search problems. GA is inspired by the mechanism of natural selection that stronger individuals are likely the winner in a competing environment. Good introductions of GA can be found in Man et al. (1999).

2.4.1 GA Operations

The basic idea of GA is to start from a set of solutions called “population” is randomly chosen and each solution is coded in terms of “chromosome”. The population is evaluated the fitness value, which is used to reflect the degree of “goodness” of chromosome. The fitness value would be highly related with the objective value of the problem. The potential solution space of a specific problem allows the population to evolve from generation to generation by generic operators i.e., selection, crossover, and mutation until the stopping criteria are satisfied. The GA process is illustrated in Figure 2.8. Each GA operation is explained as follows:

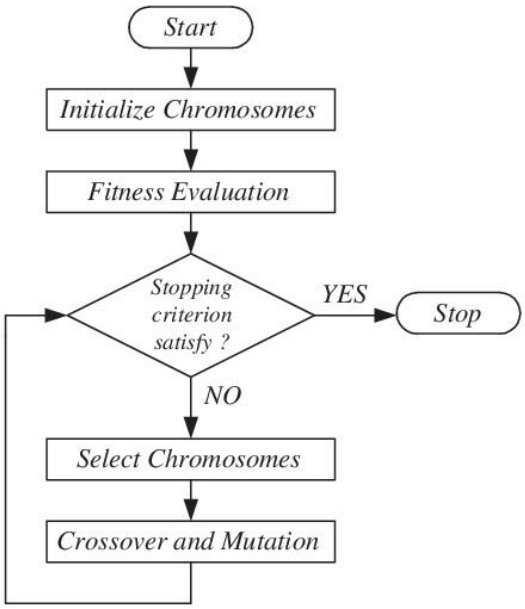


Figure 2.8 GA process

- ***Initialization:*** Initially, several possible solutions (population) are randomly generated. The population size depends on each problem. Typically, it contains around hundreds to thousands of possible solutions. To avoid the local optima, the solutions may be seeded in areas where optimal solutions are likely to be found.

- ***Coding*:** A population of individuals must be appropriately formulated for GA operations. Each individual solution is coded into a finite length string (or vector) of variables in terms of some alphabet, generally the binary alphabet (0 and 1) as shown in Figure 2.9. In the figure, 3 variables; x_1, x_2, and x_3, are represented by the binary coding. There are other forms of coding e.g., grey, integer, and real-value coding. To continue the genetic analogy, these individuals are likened to chromosomes while the variables are analogous to genes. Thus a chromosome (solution) is composed of several genes (variables).

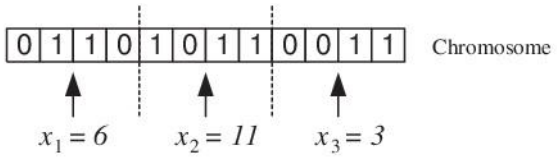


Figure 2.9 Binary coding

- ***Selection:*** During the stage of successive generation, a new generation will be procreated from a proportion of the existing population (parents). Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are more likely to be selected. Practically, "Roulette Wheel Selection", which is one of most common techniques, is adopted. The procedure of this method is as follows (Man et al., 1999):

  - Sumarize the fitness of all population members; named as total fitness (F_{sum}).
  - Generate a random number (n) between 0 and total fitness F_{sum}.
  - Return the first population member whose fitness, added to the fitness of the preceding population member, is greater than or equal to n.

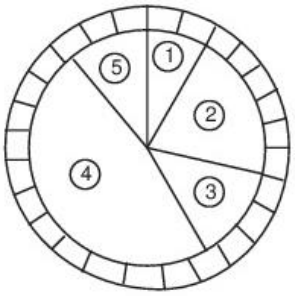


Figure 2.10 Roulette wheel selection for 5 chromosomes

For example, the circumference of the Roulette wheel is F_{sum} for all five chromosomes is illustrated in Figure 2.10. Chromosome 4 is the fittest chromosome and occupies the largest interval while chromosome 1 is the least fit that represented by a smaller interval within the Roulette wheel. To select a chromosome, a random number is generated in the interval $[0, F_{sum}]$ and the individual whose segment spans the random number is selected.

- ***Crossover:*** In this step, a pair of parent chromosomes is randomly selected and a new offspring will be created from interchanging the genes from the parents at a crossover point. The simplest way is the single crossover as shown in Figure 2.11. It is noted that the new offspring has some parts of solution from the parents. There are other ways to make crossover, e.g., multi-point crossover as shown in Figure 2.12. The crossover is occurred with the crossover probability at p_c that typically is around 0.6-1.0 (Man et al., 1999).

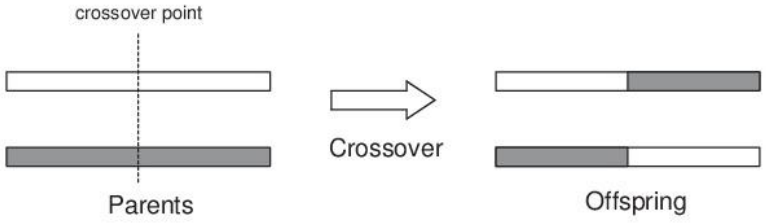


Figure 2.11 Single crossover

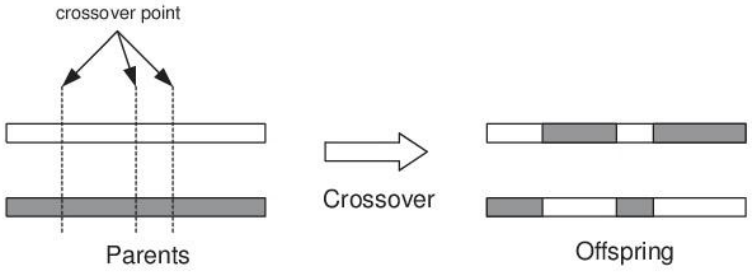


Figure 2.12 Multi-point crossover

- ***Mutation:*** In some problems, performing only the crossover is insufficient to provide the variety of the offspring in every generation. Therefore, mutation should be performed. This is to prevent falling all solutions in population into a local optimum. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1 as shown in Figure 2.13.To avoiding infeasible solution, the mutation rate is suggested to be small value such as 0-0.1 (Man et al., 1999).

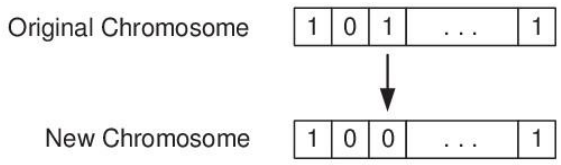


Figure 2.13 Bit mutation

2.4.2 Fitness Function

Because the new generation of population has to be selected by containing a trend of an optimal solution, a fitness function will be related with an objective function of the problem. Typically, the fitness function is defined as a inversely proportional to the objective function.

2.5 Robust Design

To successfully design high quality products, conventional design approaches are no longer sufficient to meet design target under uncertain environments. Thus, concepts of robust optimal design methods play an important role and have been widely developed in different engineering fields. This section describes a general robust design and a robust optimization using GA and meta-model technique. The design procedure is given in this section.

2.5.1 Traditional Robust Design

Robust design was originally proposed by Taguchi (1992). The original idea is to improve the product quality via minimizing the effect of the variation without eliminating the variation causes. In term of the robust optimization, this method is the consideration of both the optimality and the robustness of the objective function. In other words, the robust optimization is obtained by minimizing expected performance and minimizing performance variance, simultaneously (Du et al., 2000). The mathematical model of robust design optimization is addressed by reforming a conventional optimization problem into a robust design formulation. The conventional problem is as follows:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && g_j(\mathbf{x}) \le 0 \qquad j = 1,2,\ldots,N \\ & && \mathbf{x}^L \le \mathbf{x} \le \mathbf{x}^U \end{aligned} \tag{2.1}$$

where $\mathbf{x}$, $\mathbf{x}^L$, and $\mathbf{x}^U$ are vectors of the design variables, lower bounds, and upper bounds, respectively. $f(\mathbf{x})$ stands for the system performance and $g(\mathbf{x})$ is a constraint function vector. From Eq. (2.1), the robust design is stated as a bi-objective design problem that is expressed as follows:

$$\begin{aligned} &\text{minimize} && \mu_f, \sigma_f^2 \\ &\text{subject to} && g_j(\mathbf{x}) \le 0 \qquad j = 1,2,\ldots,N \\ & && \mathbf{x}^L \le \mathbf{x} \le \mathbf{x}^U \end{aligned} \tag{2.2}$$

where μ_f and σ_f are the mean and standard deviation of the system performance $f(x)$.

Frequently, the bi-objective function in Eq. (2.2) is transformed into a single objective function (z), which compromises between the mean (μ_f) and variance (σ_f) of the system performance as follows:

$$z = w_1\mu_f + w_2\sigma_f^2 \tag{2.3}$$

where w_1 and w_2 are the mean and variance weights, respectively.

2.5.2 metaModeling

The uncertainty assessment and analysis are necessary to find the mean and variance for above objective functions. Two approaches are mostly referred; Monte-Carlo simulation (MCS) strategies and metamodel approach. MCS is a method that characterizes uncertainty from simulating circumstances with a considerable amount of data (that describe the system dynamics) while the metamodel approach uses an abstraction model (a model of model that aims to reduce model complexity) for predicting the statistical information.

However, MCS approach requires expensive computational time. Thus, the metamodel for uncertainty assessment seems to be a better choice. There are several metamodeling techniques. However, this section briefs the classical technique as polynomial regression and Artificial Neural Network (ANN) examined in our study.

2.5.2.1 *Polynomial Regression (PR)*

The metamodel or response surface methodology based on polynomial regression has been widely applied in engineering area because its easiness in implementation and interpretation. The detail is deeply described in Jin et al. (2003) and is summarized here. A second-order polynomial model can be expressed as:

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j \geq i}^{k} \beta_{ij} x_i x_j \tag{2.4}$$

Where $\hat{y}$ is the fitted function, k is the number of parameters, β_0, β_i, and β_{ij} are constants, and x_i is the ith model parameters.

To create a reasonable metamodel by polynomial regression, the sample size should be at least two or three times of the number of model coefficients (Jin et al., 2003). For example, a problem with n input variables, there are $(n+1)(n+2)/2$ coefficients for a quadratic polynomial model. However, it is noted that with the number of input variables increasing, higher order polynomial models could easily become unaffordable.

2.5.2.2 *Artificial Neural Network (ANN)*

Artificial Neural Network (ANN) is an effective tool to construct a metamodel. ANN is used for mapping between design factors and the system performance in the robust manufacturing environment. ANN is modeled to simultaneously estimate both the mean and variance of the system performance. It can be briefed as following.

The ANN architecture in Figure 2.14 consists of basic components inspired by biological nervous systems. There are mainly an input layer, hidden layers, and an output layer. Each layer has its neurons (or nodes) and connection weights. Each of inputs will be multiplied by a connection weight. The products are summarized and fed

through a transfer function to generate a result and then output. The number of neurons in the input and output layers is depended on the number of the input and output parameters, respectively.

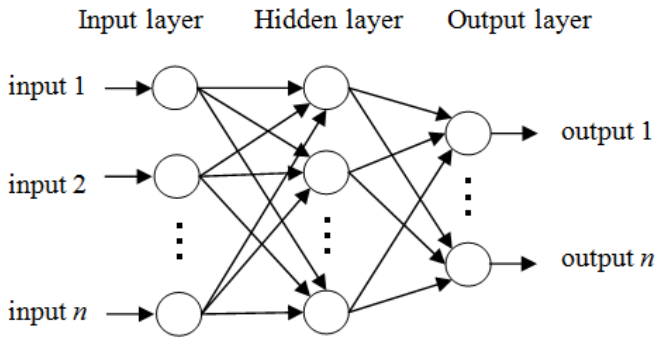


Figure 2.14 Multilayer feed-forward backpropagation ANN

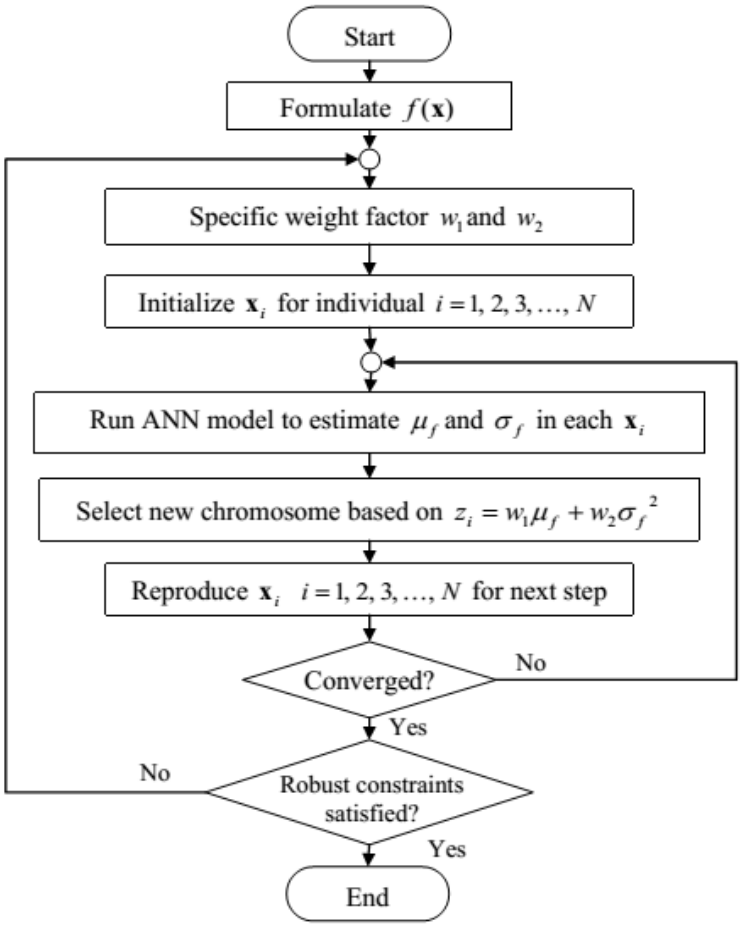


Figure 2.15 Procedure of robust design optimization based on GA-ANN

2.5.3 Robust Optimization with metaModeling

The design procedure of the methodology based on GA and ANN is illustrated in Figure 2.15. From this procedure, the system performance $f(\mathbf{x})$ is firstly defined and then the initial weights w_1 and w_2 are specified. The next step is a GA loop. The chromosomes are initialized and then the ANN model, which is properly predefined, is run for estimating μ_f and σ_f in each chromosome. The objective function z is evaluated and the GA operations (such as the selection, crossover, and mutation) are performed in this loop. The optimal solution will be finally tested and the mean and variance weights must be adjusted until the robustness criteria are satisfied.

2.6 Literature Review

2.6.1 Review of Development of MES using Simulation

Simulations are successfully applied to improve the substantial functions of MES in many industries such as the production planning, job dispatching, and even executing or control system. For example, Douglas (1998) presented a concept of evaluating a shop floor schedule in a semiconductor fab by a software package of the off-line simulation based on data from MES. Hwa et al. (1999) developed a discrete event simulator for the MES using the object-oriented technique. Sunkara and Rao (2003) used the discrete event simulation to design and analyze the real time dispatching (RTD) of a semiconductor fab. They used a model-based simulation, which was developed based on the information from MES (i.e., the route definition, equipment states). This simulation was used to generate the output for driving RTD decisions. Recently, Mönch, et al. (2003) presented a framework of the control system performance assessment based on the simulation. The difference between shop floor models (by scheduler/dispatcher which was supported by MES) and the control system was tested via the simulation and the implementation showing the effective of this approach. This framework was inherited and improved for fabrication processes in Ponsignon and Mönch (2014) as shown in Figure 2.16. In the paper, a reduced discrete-event simulation model was adopted for the production planning under uncertainty that takes the plan execution into account. The result confirms the necessity to evaluate planning algorithms via the approach while confronted with demand and system uncertainty.

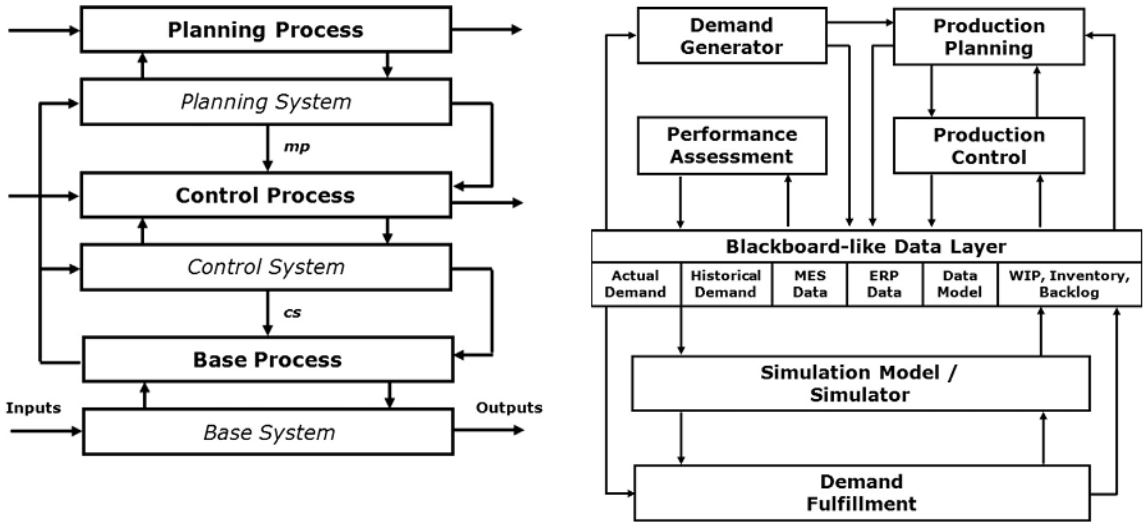


Figure 2.16 Architecture of the proposed simulation-based framework in Ponsignon and Mönch (2014)

In the bottom level of ISA SP 95, which overlaps with MES (such as continuous/batch and supervisory control), was also developed using the simulation tools. Some literatures integrated the shop floor control with the plant simulator for virtual manufacturing purpose. In addition, the simulation for the system design and

development purpose in the decision support system was examined. Recently, Adolfsson et al. (2000), and Olofsgård et al. (2002) presented the role of virtual manufacturing using the graphical simulation aided the design and development of the machine system include the control system, namely the VIR-ENG project. Carrasco and Paljakka (2004) introduced the process simulation assisted the automation system. The state of art of the simulator for the process control and the simulator for testing were described.

About steel industry, Nagasaka (1999) presented an implementation of using the simulation for the casting process. Three types of models, which are geometry model for a product, mathematical model for physical phenomena and activity, model for human operations, were used to construct a virtual manufacturing environment for casting and heat treatment processes. Recently, Azadeh and Maghsoudi (2010) integrated a computer simulation with design of experiment, and Tabu search to improve the steelmaking production optimization. This integration could improve several benefits for the plant. Recently, Melouk et al. (2013) employed a simulation optimization approach to develop a decision support system for steel manufacturing. Their simulation optimization approach as in Figure 2.17 is designed for a slab yard inventory management. The result showed the potential to adopt in other complex system of the steel plant.

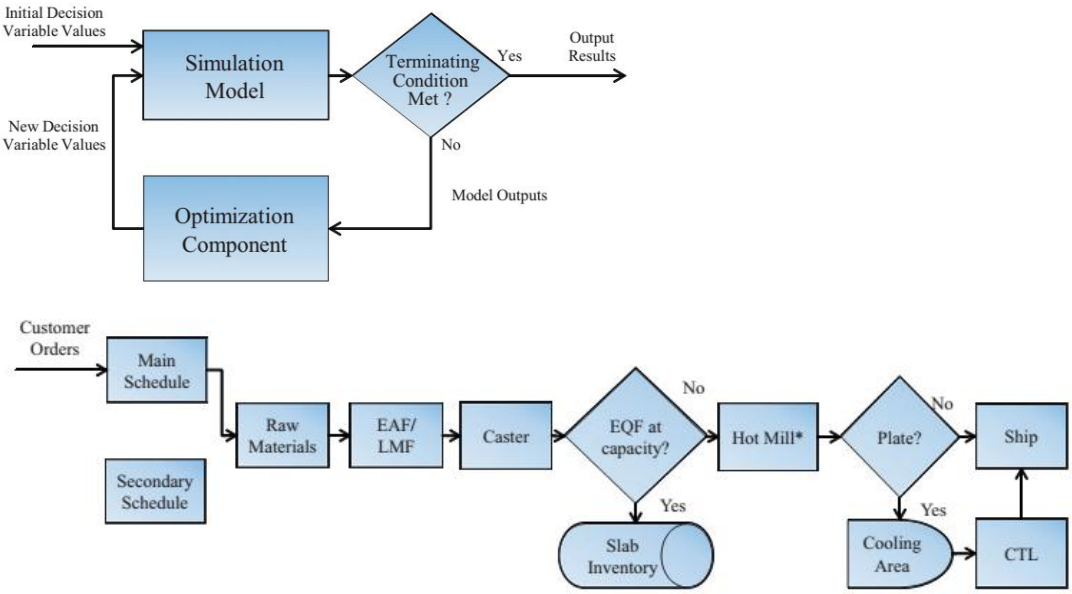


Figure 2.17 Simulation optimization structure in slab yard inventory (Melouk et al., 2013)

2.6.2 Review of Production Scheduling and Its Executing in Steel production

Research on the scheduling of SCC plant mostly concerned on 5 main areas (Lee et al., 1996); 1) Utilization of manufacturing unit, 2) Allocation of production among parallel manufacturing, 3) Sequencing and grouping of heats, 4) Coordination of schedules

between production stage, and 5) Rescheduling. First four issues are often classified to a predictive scheduling that deals with a schedule evaluation without taking disturbances (unexpected events) into account while the final issue is classified to a reactive scheduling that reacts to the disturbances and deals with a schedule revision after the schedule is executed.

Considering the development of the scheduling in first four issues, a review with various steel production scheduling was comprehensively addressed in Tang et al. (2001) and in Dutta and Fourer (2001). A SCC scheduling process was earlier introduced by Numao and Morishita (1991) and then was inherited in some later literatures. Their heuristic scheduling was developed at 2 tiers; (1) sequencing the cast (sub-scheduling); and (2) scheduling of steelmaking process and timing of the jobs (rough scheduling), including elimination of machine conflicts (optimal scheduling). Most researches concentrate on either tier while the rest is given. For instance, Chang et al. (2000), Xue et al. (2004), Jian et al. (2004), and Gravel et al. (2002) determined the cast sequence on continuous casters (the 1^{st} step) by a binary or integer programming formulation and solve it by heuristic and artificial intelligent methods. Lee et al. (2004) designed a continuous slab caster schedule via a special class of graphs called interval graphs while Cowling et al. (2004) adopted a multi-agent system for evaluating a dynamic caster schedule.

For instance in the 2^{nd} step, most researches perform the SCC scheduling by getting the cast sequence from a higher level planning. The linear programming and heuristic methods are mostly adopted. Tang et al. (2000) proposed a SCC scheduling, which considers an elimination of machine conflicts problem, via non-linear programming model. In another work, Tang et al. (2002) developed an integer programming model and obtained the optimal solution by combining Lagrangian relaxation and heuristic. Pacciarelli et al. (2004) completed the SCC schedule by using a generalization of the disjunctive graph of Roy and Sussman and Beam search. Bellabdaoui et al. (2005) focused on SCC scheduling inspired by an industrial application from Arcelor Group to eliminate machine conflicts via linear programming and then solve via a heuristic algorithm. In another later work, Bellabdaoui et al. (2006) formulated the same problem into a mixed-integer linear programming, and used a commercial software package. Recently, Atighehchian et al. (2009) designed the SCC scheduling by using a combination of ant colony optimization and nonlinear optimization and compared the efficiency between standard genetic algorithm and a heuristic method.

Most above works of the steel scheduling are deterministic. Since the uncertainty (particularly, on the uncertain processing time) and disturbances (e.g., rush orders, machine breakdown, etc.), which are considered to an internal disruption (Pfeiffer et al., 2007), do exist in real steel production, therefore the robust scheduling against these disruptions is considered to be a very challenging issue. Previously, the reactive scheduling is used to be a scheduling framework against the uncertainty (Honkomp et al., 1997 and Aytug et al., 2005). Recently, a robust predictive-reactive scheduling is

proposed as the framework for the rescheduling against the disturbances in Vieira et al. (2000b), Vieira et al. (2003), and Aytug et al. (2005). This framework generates an initial schedule and then updates the schedule in response to disturbances. The disturbance effect is measured as the schedule robustness or stability in this framework. For example, Cowling et al. (2002) applied the stability measure with the dynamic scheduling for the single machine rescheduling case while Rangsaritratsamee et al. (2004) and Pfeiffer et al. (2007) considered the stability based dynamic scheduling in the job shop production. For the flexible flow shop, Yan-hai et al. (2005) proposed the rescheduling based on stability in case of rush order disruption.

Since the robust measure for the proactive scheduling is based on the statistical information (e.g., mean and variance), the uncertainty assessment are necessary. In literatures, 2 main simulation-based techniques are mostly referred; MCS strategies and meta-model approach. MCS is used for the assessment of the statistical feasibility robustness found in Sandgren and Cameron (2002), Jin et al. (2003), and Martin and Simpson (2006). For the meta-model approach, although several papers apply the response surface methodology (RSM), this technique is not fitted for large-scale robust optimization (Beyer and Sendhoff, 2006). An artificial neural network (ANN) can be applied to predict the expectancy measures of robustness instead of MCS and RSM as in Mezgar et al. (1997). In Mezgar et al. (1997), ANN model was used for mapping between design factors and the system performance in the robust manufacturing environment. Zobel and Keeling (2008) created a meta-model by ANN aimed to predict the probability distributions for decision making under uncertainty. In work of Jung and Yum (2011), ANN was used to map the relationship between design, noise, and signal parameters for Taguchi-based technique. Recently, Choi and Wang (2012) introduced a decomposition-based ANN for the robust flexible flow shop scheduling by applying scheduling policy and strategy based on each partitioned process.

For the plan execution, the review is focused on the continuous casting process, which is used as a case study in this dissertation. Most of works deal with an optimization of quality of the steel slab via simulations. Since 1960's, a heat transfer simulation during solidification plays an important role in analyzing and adapting the parameters of the secondary cooling (to avoid cracking) in the continuous casting of both slab and billet in Mizikar (1967), Brimacombe et al. (1980), and recent ones such as Shen et al. (2002) and Hardin et al. (2003). Recently, optimizations based on the model for the spray cooling with different methods are presented. For instance, Filipic and Laitinen (2005) and Cho et al. (2008) applied a stochastic optimization and a Broydon-Fletchear-Goldfarb-Shanno (BFGS) method to search the optimal spray pattern, respectively. Due to nonlinearity of the solidification model and the objective function, Genetic Algorithm (GA) and Artificial Neural Network (ANN) are used to search the optimal solution (Santos et al. 2003, 2005). However, there are a few works, which defined the objective function by the realistic requirements and the metallurgical criteria (Santos et al. 2003, 2005; Spuy et al., 1999). Thus, the optimization based on the real requirements is still a challenging issue and must be continuously developed.

CHAPTER 3 SIMULATION-BASED MES ON PRODUCTION SCHEDULING IN COTINUOUS CASTING PROCESS

The scheduling, as mentioned previously chapter, mainly consists of 2 tiers; 1) Sequencing of the cast (also called sub-scheduling) and 2) Timing of the jobs (also called rough scheduling). Due to the first tier, this dissertation starts from studying on the sub scheduling while the overall scheduling will be discussed in the later chapter.

3.1 Simulation-Based MES for Steel Plant

In this chapter, a framework of the simulation-based MES is introduced. Since MES bridges ERP and factory control by making the optimized plans carried out by the factory control level, MES is expected to have a capability of optimization (Kletti, 2007). Different simulations play a role on the optimization for the main functions, which involve with the productivity and quality management. As shown in Figure 3.1, the discrete event simulation is used for optimizing the production scheduling while the solidification simulation is used for optimizing its execution (e.g., a production yield and quality management).

- ***Planning phase*:** This phase deals with the tasks of the planning. The production planning and control module will be modified. To achieve the effective scheduling system against the unexpected events (disturbances) and uncertainty, the discrete event simulation (DES) of the steel production is an important tool for evaluating/optimizing the production schedule. Normally, the user of this phase is a planner and engineer. The steel production model is created with the real data from factory and is properly validated. The graphic user interface (GUI) may be used in this phase. Practically, a middleware is used to communicate between MES and the simulation unit, including other systems, which may have different operating systems such as UNIX, VMS, AIX, and Windows (Qiu et al., 2004). An optimal schedule will be sent to execute in the control layer.

- ***Plan execution phase*:** This phase deals with the plan execution in the overlapped zone of MES and the control layer. The modules of yield optimization and quality management are modified by integrating with the continuous simulation. Normally, the user of this phase is the automation and maintenance engineers. The continuous model is the steel solidification in mold and cooling zones. Model's output is the temperature in each position of the caster. This model is a tool for optimizing the production yield and tuning the supervisory control system. When the optimal caster schedule is sent from the scheduling module, those model parameters such as the slab dimension and steel grade will be set from the schedule and then the casting

speed will be calculated by this model. Similarly, the spray cooling rate in the control system will be also optimized by the model.

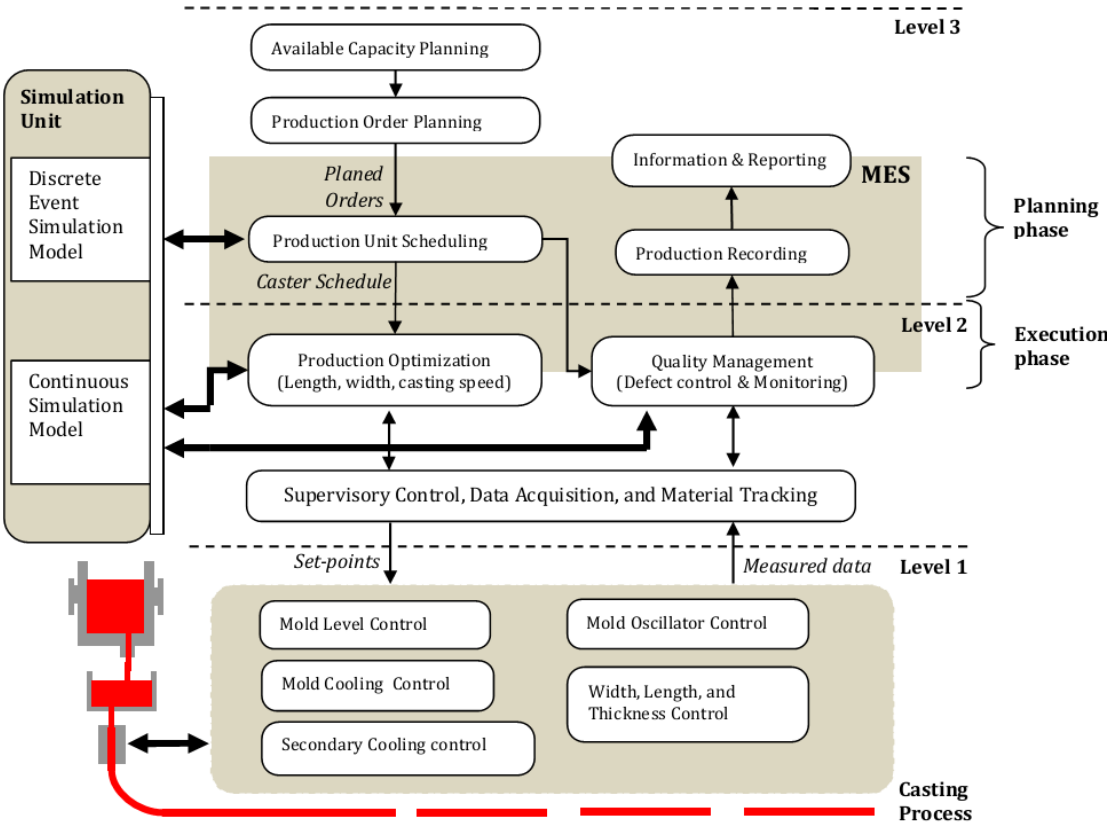


Figure 3.1 Proposed simulation-based MES

3.2 Deterministic Scheduling for CC Process

The CC scheduling problem is often formulated as a non-convex integer programming model that is a discrete problem. It is complex and may be inappropriate to solve by traditional approaches as gradient-based method (Mahdavi and Hanna, 2004). Most researches solved the problem by heuristic and meta-heuristic methods (Yagmahan and Yenisey, 2008). Comparative studies among the heuristic methods or the traditional method, including a combination of them are often applied in order to develop an effective scheduling system. Recently, Atighehchian et al. (2009), which combined ant colony optimization with nonlinear optimization and compared the efficiency with genetic algorithm and heuristic method. Similarly, this section studies the comparison of the CC scheduling by GA and gradient-based optimization. Due to this non-convex objective function, local optimum may be reached. Thus, verification by the performance comparison with other optimization methods is necessary. Since the uncertainty of processing time is ignored (constant steel weight in ladle), a deterministic situation is considered and with some techniques as linear relaxation, gradient-based method can be used to compare with genetic algorithm

3.2.1 Efficiency Criterion for CC Process

Typically, an efficiency criterion for a production scheduling problem is often measured by makespan and tardiness (Pfeiffer et al., 2007 and Ouelhadj , 2003). However, the efficiency criterion for CC process mostly takes the job/group sequencing of the production order, and allocation of production and utilization of manufacturing unit into account. Therefore, this work defines the efficiency criterion, which is inspired from Xue et al. (2004) and Tang et al. (2002)'s objective functions, and the special requirement of caster in Chapter 2, in terms of the production cost as shown on Eq. (3.1), (3.2), (3.3) and (3.4).

$$E = \sum_{m=1}^{M} \sum_{k \in P_m} \mathbf{X}_k{}^{\mathrm{T}} \mathbf{Q} \mathbf{X}_k \tag{3.1}$$

$$\mathbf{Q} = \begin{bmatrix} q_{11} & \cdots & q_{1N} \\ \vdots & \ddots & \vdots \\ q_{N1} & \cdots & q_{NN} \end{bmatrix}, \quad \mathbf{X}_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{Nk} \end{bmatrix},$$

$$q_{ij} = c_{ij}^{g} + c_{ij}^{w} + c_{ij}^{th} + c_{ij}^{d} + c_{ij}^{q}$$

Subjected to:

$$\sum_{j=1}^{P} x_{ij} = 1, \quad i = 1, \cdots, N \tag{3.2}$$

$$2 \le \sum_{i=1}^{N} x_{ij} \le L, \quad j = 1, \cdots, P \tag{3.3}$$

$$x_{ij} \in \{0,1\}, i = 1,\cdots,N \quad j = 1,\cdots,P \tag{3.4}$$

where

M the total number of machines

P the total number of casts

P_m the set of casts which are processed on machine m

N the total number of charges to be arranged

L the maximum number of charges for a cast

$\mathbf{X}_k$ decision vector of the k^{th} cast, each element is the binary decision variable x_{ij} where equals 1 if charge j belongs cast i and is processed on machine m

q_{ij} production cost matrix between charge i and, j which contains the losses as follows:

c_{ij}^{g} loss of steel grade difference

c_{ij}^{th} loss of thickness change

c_{ij}^{wd} loss of width change

c_{ij}^{cd} loss of consignment date

c_{ij}^{cb} loss of cast break and waiting time.

All of the loss functions can be shown as follows:

$$c_{ij}^{g} = \begin{cases} 0 & \text{steel grade of charge } i \text{ and } j \text{ being the same} \\ f_1 & \text{charge } i \text{ and } j \text{ belonging to the same steel grade serial} \\ f_{\max} & \text{charge } i \text{ and } j \text{ belonging to the difference steel grade serial} \end{cases}$$

$$c_{ij}^{wd} = \begin{cases} 0 & d_i = d_j \\ f_2 * \left| d_i - d_j \right| & \left| d_i - d_j \right| \geq \Delta D_{\max} \end{cases}$$

$$c_{ij}^{th} = \begin{cases} 0 & h_i = h_j \\ f_3 * \left| h_i - h_j \right| & \left| h_i - h_j \right| \geq 0 \end{cases}$$

$$c_{ij}^{cd} = \begin{cases} f_4 * (c_i - c_j) & c_i - c_j \geq 0 \\ f_5 * (c_j - c_i) & c_j - c_i < 0 \end{cases}$$

$$c_{ij}^{cb} = \begin{cases} f_{6} * (t_i - a_i) & t_i - a_i \geq TW_{\max} \\ f_{\max} & t_i - a_i < 0 \end{cases}$$

$f_1, \cdots, f_6$ penalty factors

$f_{\max}$ maximum breakdown cost

d_i ordered slab width of charge i

h_i ordered slab thickness of charge i

$\Delta D_{\max}$ maximum width

$TW_{\max}$ maximum waiting time

c_i consignment date of charge i

t_i start time of charge i

T_i processing time of charge i

a_i arrived time of caster of charge

3.2.2 Formulation to Standard Form

The efficiency criterion in Eq. (3.1)-(3.4) is formulated into a standard form, which is an equivalent equation. The decision variable is transformed to be a vector $\mathbf{z}$ that has the vector size equal $N \times P$ (from $\mathbf{X}_1$ to $\mathbf{X}_P$) where $\mathbf{X}_i \in P_m$. The standard equation can be shown as follows:

$$f(\mathbf{z}) = \mathbf{z}\mathbf{Q}_t\mathbf{z}, \tag{3.5}$$

$$\mathbf{Q}_t = \begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} X_1 \\ \vdots \\ X_P \end{bmatrix}$$

- Equality constraint in form of $[Aeq][x] = [Beq]$

$$\sum_{j=1}^{P} x_{ij} = 1, \quad i = 1, \cdots, N \tag{3.6}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{P} x_{ij} = N \tag{3.7}$$

- Inequality constraint in form of $[A][x] \le [B]$

$$\sum_{i=1}^{N} x_{ij} \le L, \quad j = 1, \cdots, P \tag{3.8}$$

- Nonlinear constraint in form of $c(x) \le 0$

$$8 - \sum_{i=1}^{N} \sum_{j=1}^{P} x_{ij} \le 0 \tag{3.9}$$

- Boundary constraint

$$x_{ij} \in \{0, 1\}, i = 1, \cdots, N, \ j = 1, \cdots, P \tag{3.10}$$

To simulate the parallel production, the processing time is considered. Normally, the melting and refining processes are often defined with the same processing time at each machine (Bellabdaoui et al., 2006). Based on a real steel factory, the melting and refining time are defined at 50 minutes. Differently, the continuous casting process is

defined with dynamic processing time. The casting time ($T_i^{(m)}$) is calculated from slab dimension, casting speed (v_i), and steel weight (W_{ladle}) in the ladle as expressed in Eq. (3.11). Although the steel weight has uncertainty, it is ignored in this section.

$$T_i^{(m)} = \frac{W_{ladle}}{v_i w_i h_i \rho_{steel}} \tag{3.11}$$

where m is caster number, ρ_{steel} is steel specific weight, w_i and h_i are slab width and thickness of charge i, respectively. The cast speed depends on steel grade and mould dimension shown in Table 3.1.

Table 3.1 Relation of casting speed and mold dimension

Steel Type	Slab width (mm)	Casting speed with varying thickness (m/min)	
		60 mm	50 mm
Low Carbon	<1350	5.2	5.5
	1350-1450	5.0	5.2
	>1450	4.8	5.0
Medium Carbon	<1350	4.8	5.0
	1350-1450	4.6	4.8
	>1450	4.4	4.6

3.2.3 Optimal Scheduling using GA

3.2.3.1 *GA Process*

With GA's concept, the feasible solution is developed to be optimal by simulating the natural selection and survival of the fittest with selective breeding. To optimize using the standard GA, 3 processes; chromosome coding, crossover, and mutation, must be defined to fit with the problem. Firstly, the feasible solutions must be coded into the binary chromosomes in terms of $(\mathbf{X}_1, \cdots, \mathbf{X}_P)$, where $\mathbf{X}_i$ is the decision vector in Eq. (3.1). By this coding, the charge grouping/sequencing is pointed by the amount and position of elements of $\mathbf{X}_i$ that is set to 1 and half of all casts are allocated to one of the caster. For the crossover, to avoid the constraint violation, the crossover will be done in the cast level; between the elements of $\mathbf{X}_i$ and $\mathbf{X}_j$. For mutation, when the cast $\mathbf{X}_i$ and one of its elements are selected by random, the element will be assigned 0 or 1 and Eq. (3.2) is used to assign in the other casts.

3.2.3.2 *GA Experiment*

For the experiments, the optimal schedule of 12 charge orders as shown in Table 3.2 will be searched to explore the optimal number of iterations and populations.

Table 3.2 Production orders for CC process

Charge No	Consignment date	Grade Serial	Steel grade	Slab width (mm)	Slab thickness (mm)	Weight (kg)
1	16/12/07	D020	1008MnDK2	1272	60	14300
2	16/12/07	D020	1008MnDK2	1258	60	14300
3	16/12/07	D020	1008MnDK2	1252	50	14300
4	16/12/07	D011	1008MnDK-M1	1230	60	16000
5	16/12/07	D011	1008MnDK-M1	1230	60	16000
6	16/12/07	D013	1008MnDK-M3	1272	60	24500
7	16/12/07	D013	1008MnDK-M3	1552	50	14300
8	16/12/07	D013	1008MnDK-M3	1245	50	16000
9	16/12/07	D021	1008MnDK2	1240	60	16000
10	16/12/07	G001	1007DKGXA-00	1272	60	14300
11	16/12/07	G001	1007DKGXA-00	1255	60	19500
12	16/12/07	G001	1007DKGXA-00	1275	50	24500

Experiment I: The size of populations 20, 30, 40, and 50 will be set for searching the optimal schedules. In each number of populations (20-50), the iterations will be increased by starting from 30 to 50 with the increasing rate 10 iterations per step. To avoid the local solution, 10 replications of optimization will be done in each case and the frequency and average of the achieved minimum solution will be observed. In this experiment, modified GA toolbox on MATLAB is applied and its parameters, which are defined by a Trial and Error method, are shown in Table 4.3.

In order to reflect the real cost, this work defines the parameters of the developed objective function by calculating from the real factory production summarized in Table 3.3 and described as follows:

- The penalty coefficient due to the grade difference (f_1 **)** is approximately defined by the loss cost \$20/ton of the approximate mixed steel 30 ton in each heat.
- The penalty coefficients due to the width (f_2) and thickness change (f_3) are approximately defined based on the disposed steel around 4 meters and recycle cost \$52/ton.
- The penalty coefficient due to consignment date (f_4 and f_5) are approximately defined by the shipping compensation with \$300/ton.
- The penalty coefficient due to cast break (f_6) is approximately defined by reheat rate 30 kwh per 30 tons.
- $f_{\max}$ is assigned by the steel loss of a ladle (180 ton) with the cost rate \$80/ton.

Table 3.3 Objective function and GA parameters

Objective function		Genetic Algorithm	
Parameter	Value	Parameter	Value
$f_1, f_2,$	600, 7.36,	Representation	Binary
$f_3, f_4,$	1612, 300,	Population size	{20, 30, 40, 50}
f_5, f_6	300, 6.7	Max. iteration	{30, 40, 50}
N, P, L	12, 2, 4	Crossover	Single point (rate 0.8)
$f_{\max}$	14400	Selection	Stochastic uniform
$\Delta W_{\max}$	4 mm	Mutation	Random (rate 0.01)

3.2.4 Optimal Scheduling using Gradient-based method

3.2.4.1 *Linear relaxation*

Since the objective function is an integer quadratic programming model (a discrete problem), it is not easy to solve by the gradient-based method, especially, with an optimization toolbox on MATLAB. Thus, constraint relaxations are often used to modify the constraints. For example, Shaojian et al. (2006) used linear relaxation to solve a quadratic programming. In Luh and Hoitomt (1993), and Tang et al. (2002)'s works, Lagrangian relaxation was used to solve an integer programming for steel making process. Similarly, the liner relaxation is applied to modify the constraint in this paper. The integer constraint in Eq. (3.4) is relaxed as below:

$$0 \le x_{ij} \le 1 \text{ for all } i, j \tag{3.12}$$

3.2.4.2 *Gradient-based method experiments*

Two experiments for comparison with GA and extension of GA's result are performed. They can be described as follows:

Experiment II: To compare with GA, the initial populations of GA's experiment (Experiment I), which results the optimal schedule, will be used as the initial solutions for the gradient-based method. Then the optimal solution will be compared with GA's optimal solution.

Experiment III: To extend the GA results, the optimal schedule from GA will be used as the initial solution for gradient-based method.

In this section, the experiments can be achieved via the gradient-based optimization toolbox (automatically default at Quasi-Newton) on MATLAB.

3.2.5 Experimental results

Experiment I: The results indicate the minimum objective value at around $5579.2 and they are plotted in Figure 3.2, which shows the achievable frequency of optimal schedule in each population and iteration, while the average objective values in each case, are shown in Figure 3.3. Distinctly, it is not only the number of the achieved optimums is increased by the number of populations and iterations, but the average objective values are also decreased. Considering Figure 3.3, the average objective value in case of 30 populations is not significantly different with the case of 50 populations while the case considering the amount of iterations in case of 40 iterations also points the same trend.

Numerically, the case of 30 populations and 40 iterations approximately spends the computational time around 20 minutes while the case of 50 iterations approximately spends the time more than 1.5 times of that (30 minutes) with a 1.66 GHz Intel (R) Core 2 CPU RAM 1 GHz computer. Therefore, it can claim that in the case of 30 populations and 40 iterations is an optimal one by the computational time consumption. In Figure 3.4, the best objective values in each generation in the case of 30 populations and 40 iterations are used as the example of GA's behavior.

It is noted that although an enlargement of the populations and iterations provides the good solutions (lowest average objective value), it spends more computational time. In the same time, the smaller populations and iterations (such as 30 populations and 40 iterations) can provide acceptable results (low average objective value) with similar computational time. Therefore the case of smaller populations and iterations is suitable to use as an initial solution for the next experiment.

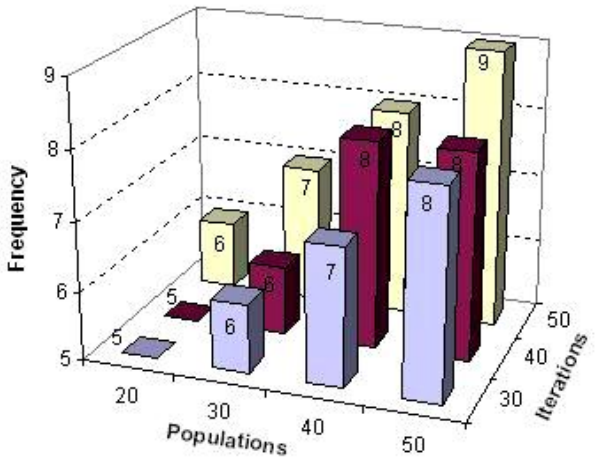


Figure 3.2 Frequency of achievement in Experiment I

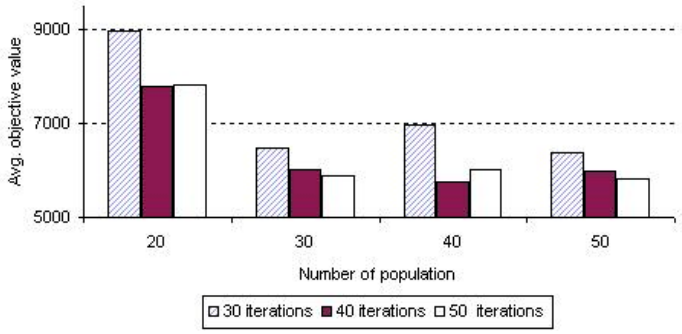


Figure 3.3 Average objective values in Experiment I

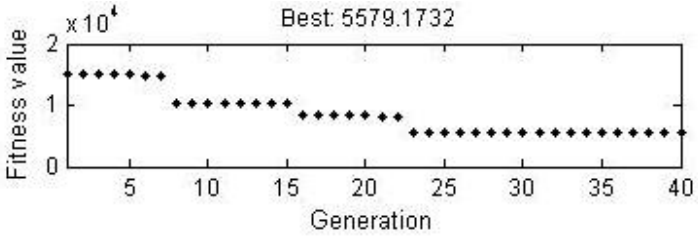


Figure 3.4 Best objective values in each generation in the case of 30 populations and 40 iterations

Experiment II: To explore with GA, GA's initial populations in the case of 30 populations and 40 iterations, which results minimum objective value ($5579.2), are selected to use as an initial for the gradient-based method. The result is shown in Figure 3.5. In the figure, the minimum objective value is around $5503.1. Although results from both of GA and gradient-based method are not identical, they provide the same structure of the schedule because of the effect of linear relaxation, which x_{ij} is not rounded off. From the results, the convergence to optimum in the gradient-based optimization is depended on the initial solutions (not on replications). Its performance will be measured by the frequency of the achievement, which is directly counted by Figure 3.5. It shows the similar frequency of the achievement as Experiment I.

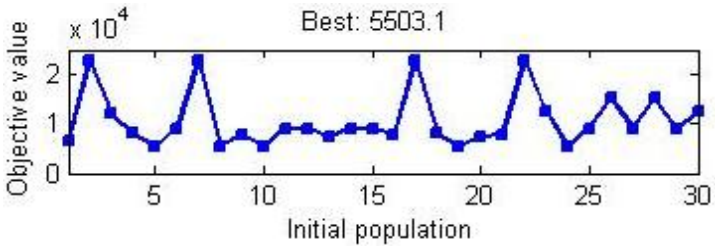


Figure 3.5 Optimal objective values by gradient-based method with the initial from GA in case of 30 populations and 40 iterations

For the computational time of the gradient-based method, it approximately spends more than 10 minutes depending on the initial solution. Regarding this computational time, it

can be used to improve the GA's result with the small generation and iterations to reach on the optimal solution.

Experiment III: To extend the GA's results, the low frequency of the achievement as seen in the case of 20 and 30 populations is improved. The final solutions of the GA's results are used as the initial of the gradient-based method. Figure 3.6 and 3.7 indicate that the frequency of the achievement is significantly increased. Although it is clear that the combination can improve the result with the increased frequency, it can provide the worst average objective value than the old one in some cases due to the local optimality as shown in Figure 3.7 (in the case of 30 populations and 40 iterations).

Figure 3.8 and Table 3.4 represent an optimal schedule, which are appropriately arranged with grouping the steel grades into the same family in each cast, and show that the idle and waiting time are acceptable. Although there is some idle time, it is not the cast break and is normally included to a turnaround time.

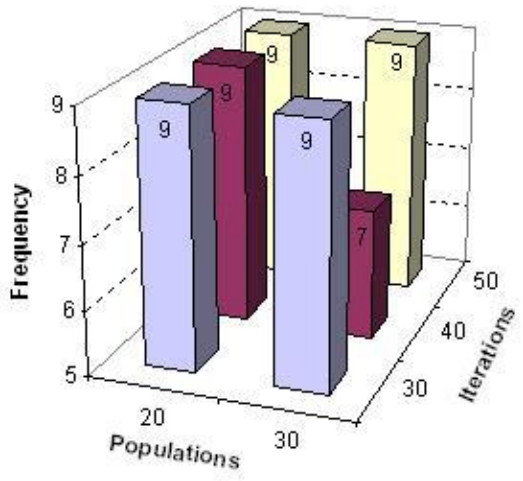


Figure 3.6 Frequency of achievement in Case III

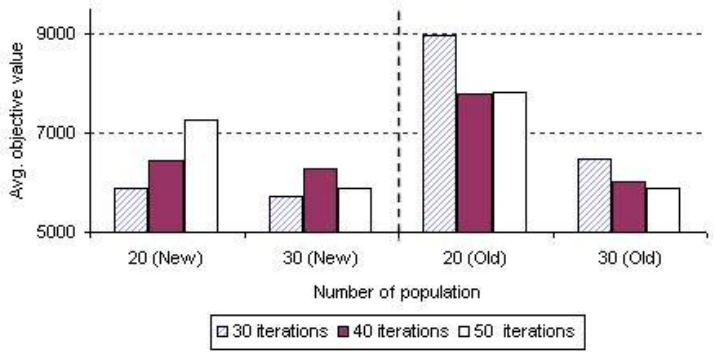


Figure 3.7 Comparison of average objective values in Case III

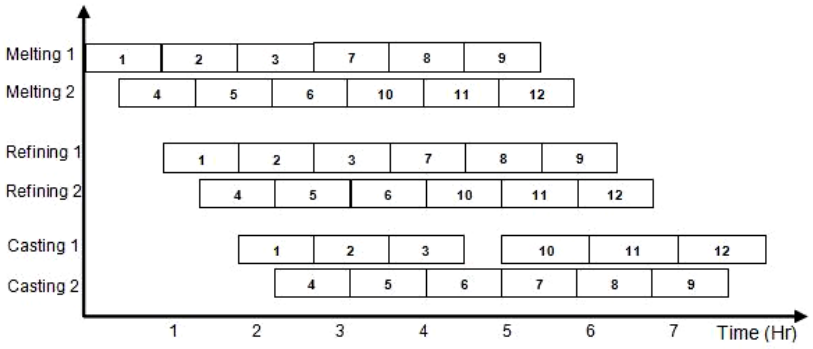


Figure 3.8 Gantt chart of the optimal schedule

Table 3.4 The optimal schedule

Cast No	Charge No	Consignment date	Grade Serial	Steel grade	Slab width (mm)	Slab thickness (mm)	Weight (kg)	Location of Casters	
								1	2
1	1	16/12/07	D020	1008MnDK2	1272	60	14300	×	
	2	16/12/07	D020	1008MnDK2	1258	60	14300	×	
	3	16/12/07	D020	1008MnDK2	1252	50	14300	×	
2	10	16/12/07	G001	1007DKGXA-00	1272	60	14300	×	
	11	16/12/07	G001	1007DKGXA-00	1255	60	19500	×	
	12	16/12/07	G001	1007DKGXA-00	1275	50	24500	×	
3	4	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
	5	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
	6	16/12/07	D013	1008MnDK-M3	1272	60	24500		×
4	7	16/12/07	D013	1008MnDK-M3	1552	50	14300		×
	8	16/12/07	D013	1008MnDK-M3	1245	50	16000		×
	9	16/12/07	D021	1008MnDK2	1240	60	16000		×

3.3 A Robust Scheduling under Uncertain Processing Time in CC Process

This section proposes a robust predictive-reactive scheduling, which is robust to both disturbances and uncertainty for the multi-strand continuous slab casting. The revised schedule will be optimized through a process model, which is constructed with a real uncertain factory process time, and MCS is used in uncertainty assessment.

3.3.1 A Robust Predictive-Reactive Scheduling based on Simulation

To construct the structure of an effective scheduling/rescheduling system, the environment and scheduling/rescheduling strategy have to be considered. Vieira et al., (2003) classify the production environment into the static environment (off-line) and the dynamic environment (on-line). The static environment has a finite set of jobs while the dynamic environment has an infinite set of jobs. Due to inappropriateness of the technological constraints to utilize the on-line scheduling in the steel production (Ouelhadj, 2003), the steel production is therefore defined as the static environment in this chapter. Practically, the steel production is impossible to update schedule too frequently. However, updating the existing schedule when disruptions occur is necessary.

Above characteristics are close to be a *predictive-reactive scheduling*, which is suggested to be the rescheduling framework against the disruptions in many literatures such as Cowling and Johanssan (2002), Ouelhadj (2003), Vieira et al., (2003), Cowling et al. (2004), Aytug et al. (2005), and Pfeiffer et al. (2007). In the predictive-reactive scheduling, 2 phases; predictive and reaction, are performed. First, a predictive schedule is generated. The second step updates the schedule in reaction to the disruptions. At this step, robust schedule in sense of minimizing the effects of disruption called a *robust predictive-reactive scheduling* may be created such as Ouelhadj (2003), and Cowling et al. (2004). Updating timing is normally classified into 3 alternatives; periodic, event-driven, and hybrid (Vieira et al., 2003). In this work, the event-driven is investigated. The detail is discussed in the rescheduling strategy section.

Considering a practical steel production environment, besides an unexpected event such as the machine breakdown and rush order, the uncertain processing time is another internal disruption. Simulation is one of the widely used operations research tools, which currently is brought to manage the complex production rescheduling problem under uncertainty as appeared in Li et al. (2000) and Pfeiffer et al. (2007).

This work constructs a robust predictive-reactive scheduling as shown in Figure 3.9. It can be explained in the following: the system will make a decision to revise the schedule when the disruptions occur in the production line. The decision making is performed through an established strategy, which depends on the category of the information sensed from the shop floor. The optimal revised schedule will be achieved

by the optimization based on the discrete event simulation. During the rescheduling, the order from an order pool, which contains both the remaining orders and new orders, may be utilized. The revised schedule will be sent to execute at the shop floor in real time. Clearly, the real time information coming from the shop floor is essential for activating the revision in this structure. The real time information may be acquired from various ways such as the process control system as appeared in Li et al. (2000), MES or enterprise resource planning (ERP) that most factories have nowadays this system installed as suggested in Cowling and Johanssan (2002) and Wang et al. (2006).

To achieve the robust predictive-reactive scheduling, 2 performance indicators (i.e., efficiency and stability) are proposed in further detail later in this section.

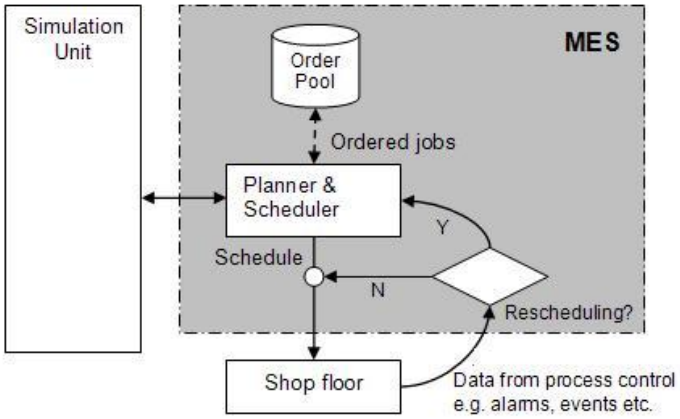


Figure 3.9 Predictive-reactive scheduling system based on simulation

3.3.1.1 *Stability criterion for continuous casting*

When a revised schedule of an area is executed, the changed job sequence in the revised schedule has an effect on the upstream production through downstream production such as machine re-setup, rescheduling in other areas. This effect would degrade the performance of the production system. The robust predictive-reactive scheduling was used to minimize the effect of the disruptions on the performance measures by some researchers, e.g., Vieira et al. (2003) and Ouelhadj (2003), Cowling and Johanssan (2002), Rangsaritratsamee et al. (2004), and Pfeiffer et al. (2007). Mostly, the effect is measured in terms of the robustness or stability, which is expressed by the deviation of start and/or completion of production. In order to reduce the effect on the upstream steel production, this paper assumes that there is actually enough time for rescheduling in the downstream production (in many steel manufactures, the caster is the final unit of steel-making production). In other words, the study on starting time deviation is used to measure the schedule robustness in Eq. (3.13).

$$ST = \sum_{m=1}^{M} \sum_{i \in N_m} \alpha \left| t_{i,m} - t'_{i,m} \right| \tag{3.13}$$

where $t_{i,m}$ and $t'_{i,m}$ are the start casting process time of charge i on machine m of the old and revised schedules, respectively. α is the weighting factor and N_m is the total charges, which are cast on machine m.

Finally, the performance criteria for the robust predictive-reactive scheduling can be defined by combining Eq. (3.1) and (3.13) into (3.14). By using the stability factor (β), the compromise between the efficiency and stability can be performed.

$$z = \beta E + (1-\beta)ST \tag{3.14}$$

3.3.1.2 *Rescheduling strategy*

The rescheduling strategy is to control the decision based on the events or disruptions for handling the remaining jobs such as decision to do nothing or repair a schedule, including to appropriately define some parameters in the system (if rescheduling is decided). The rescheduling strategy is often discussed in 2 issues; a policy and a rescheduling method. In the predictive-reactive scheduling, the policy deals with an updating period. Three alternatives of the policies; periodic, event-driven, and hybrid, are classified in Vieira et al. (2003). The periodic policy means that an existent schedule is revised at the regular intervals (rescheduling point) while the event-driven policy revises the schedule based on the real time events. For the hybrid policy, it is a combination of two basic methods. The rescheduling frequency significantly affects to the system performance (Vieira et al., 2000b), especially, leading to be unstable (shown in the experiment in Pfeiffer et al., 2007). The event-driven policy, which is mostly used in many rescheduling researches, including this work, is agreed that it is better than the periodic policy (Ouelhadj, 2003).

In the predictive-reactive rescheduling based on the event-driven for the steel production, the real time events or disruptions come from a variety of reasons (Lee et al., 1996). For example,

- A manufacturing unit goes down.
- Excessive defects occur during an operation.
- A new, high-priority order is introduced.
- An order is canceled.

However, most papers mainly focus on 2 disruptions; rush order case (Guo and Li, 2007), and machine failure case, which consists of 2 main cases; shutting down for maintenance (Guo and Nonaka, 1999) and extending the processing time (Akturk and Gorgul, 1999), including the rescheduling by considering a machine available in the work of Li and Shaw (1998). In this section, both cases are taken into account.

For the rescheduling method, it deals with the schedule modification. Vieira et al. (2003) define the rescheduling to 2 basic methods; a partial rescheduling and complete

rescheduling. The partial scheduling revises the schedule only on some operations affected by the disruptions. The target is to preserve the initial schedule as much as possible. Right-shift rescheduling is a common partial rescheduling. It postpones (by the expected delay time) each remaining operations, which are affected by the disruptions (especially the machine breakdown), to the right on Gantt chart. There are other types of the partial rescheduling such as match-up rescheduling and heuristics. For instances, Akturk and Gorgul (1999) rescheduled by match-up point determination procedure for a flow shop in case of machine breakdown while a developed heuristic rescheduling method based on Gantt chart by taking the delay time effect into account is proposed in Guo and Nonaka (1999).

For complete rescheduling, it means that all remaining jobs from the initial schedule are revised. These basic methods will be used to establish the rescheduling strategy that fits for the steel production constraint later on.

3.3.2 Uncertainty and Modeling

Since the stability is defined according to the upstream line, the parallel production shown in Figure 3.10, which contains the melting and refining processes, is simulated. The real data from a factory, which is a single line, is used to complete the parallel line model. In this model, the uncertainty of the processing time of each process is considered as follows.

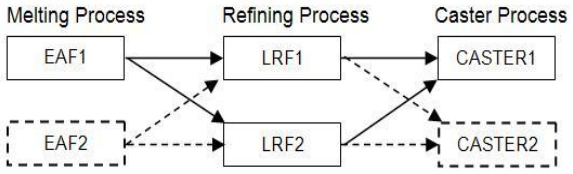


Figure 3.10 An actual production line from a factory (solid line)

3.3.2.1 *Uncertainty analysis*

- *Melting time*: Typically, the variation of the processing time depended on the operator's experience and skill such as how to manually control electric input and chemical elements, and slag forming. By the actual production data of a steel factory, 1000 samples of the processing time are represented by a probability density function (PDF) shown in Table 3.5.
- *Refining time*: Normally, refining processes are deterministically defined with the same processing time as the melting. Similarly, the uncertainty is similar with the melting case, which is depended on experience and skill. By the same number of data, the processing time are expressed as shown in Table 3.5.

- *Casting time*: Since the processing time of the continuous casting process depends on various factors such as slab dimension, casting speed and steel weight in ladle, it provides a different processing time at each charge as Eq. (4.11). The uncertainty occurs on the steel weight, which is different in each ladle depending on the amount of the added element during the refining process. By the same number of data, the steel weight is expressed as shown in Table 3.5.

Table 3.5 Expression of the process time

Parameters	Range	Expression
Melting process time (min)	40.5-79.5	42.5 + Gamma (2.03, 10.6)
Refining process time (min)	50.5-99.5	50.5 + 49*Beta (1.4, 1.01)
Steel weight per a ladle (ton)	170-206	Norm (188, 6.18)

3.3.2.2 *Validation*

To validate the process model, a confidence interval approach, which is described in Sargent (1998) and Buranathiti et al. (2006), is applied. According to this approach, if all corresponding experiment results are found in confidence intervals, the model will be not considered to be invalid at the design space tested with that confidence probability. In other words, there is not enough statistical evidence to reject the null hypothesis (accepted H_0 with type I error).

To achieve this approach, MCS using the processing time in Tables 3.1 is applied. The simulation model is created on MATLAB/Simulink and 100,000 replications are run. The completion time is recorded and calculated confidence intervals are shown in Table 3.6 and Figure 3.11. The effect of the number of samples of MCS on mean and standard deviation is illustrated in Figure 3.12. The results point out that the actual data are in the 95% confidence intervals from the simulation results. Thus designed model is good enough to represent the actual production.

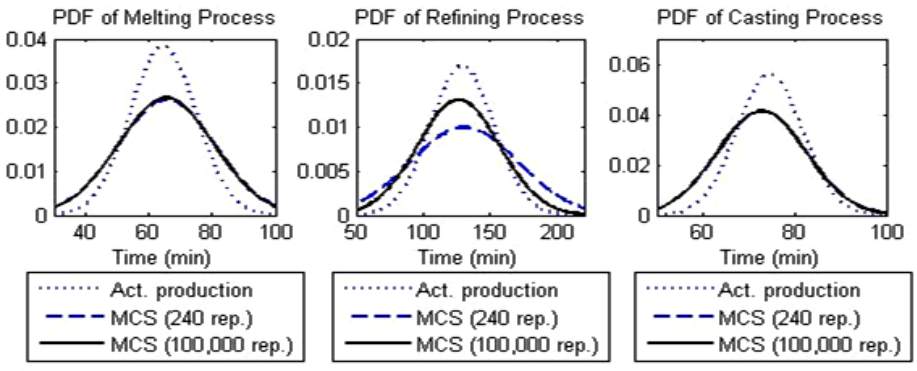


Figure 3.11 Comparison among normal (Norm.) PDF, the simulation results (Sim.), and actual data (Act.)

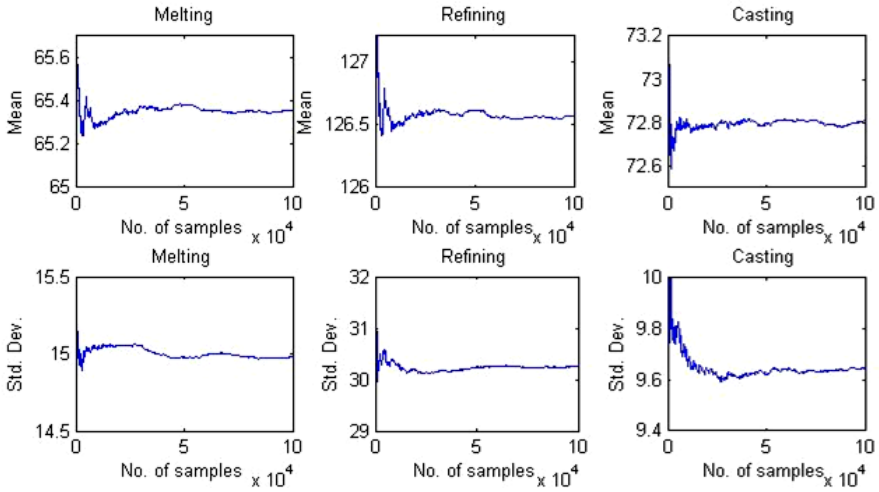


Figure 3.12 The effect of the number of samples Monte Carlo Simulation

Table 3.6 Monte Carlo simulation results

Data source	No. of Data	Distribution	Description		Confidence interval (95%)
			$\bar{x}$	σ_x	
Melting process					
Actual production	240	Normal	64.42	10.40	[44.03, 84.80]
Simulation	$1x10^5$	Normal	65.35	14.98	[36.00, 94.72]
Refining process					
Actual production	240	Normal	128.84	23.59	[82.55, 175.10]
Simulation	$1x10^5$	Normal	126.57	30.58	[67.26, 185.88]
Casting process					
Actual production	240	Normal	74.53	7.14	[60.51, 88.54]
Simulation	$1x10^5$	Normal	72.80	9.64	[53.90, 91.70]

3.3.3 Minimax Optimization

3.3.3.1 *Minimax Formulation*

To deal with the uncertainty, the robust discrete optimization is applied in this work. In general, it can be formulated as follows: Let X be the set of all solutions, S be the set of all possible scenarios, and $z(x,s)$ is an objective function of solution $x \in X$ in scenario $s \in S$. The problem is to find the best solution on the worst-case scenario. In other words, it is the same as minimizing (overall solutions) the maximum (overall scenarios) cost:

$$\min_{x \in X} \max_{s \in S} z(x,s) \tag{3.15}$$

3.3.3.2 *Genetic Coding for Minimax Optimization*

By the minimax problem, two loops of GA; for searching the optimal schedule structure (outer loop) and the worst scenario (inner loop), are performed as shown in Figure 3.13. The result of the inner loop will be used as the fitness value of the outer loop.

For chromosome coding for the rescheduling problem in this work, the feasible solutions and scenarios are coded into the binary chromosomes in a form of $x = (\mathbf{X}_1, \cdots, \mathbf{X}_P)$ and $s = (W_1, \cdots, W_M)$, where $\mathbf{X}_i$ is the decision vector in Eq. (3.1) and W_i is the steel weight for casting on each machine. By this coding, the charge grouping/sequencing is determined by the amount and position of elements of $\mathbf{X}_i$ that is set to 1. Other parameters of GA, which are defined by a Trial and Error method, are shown in Table 3.7.

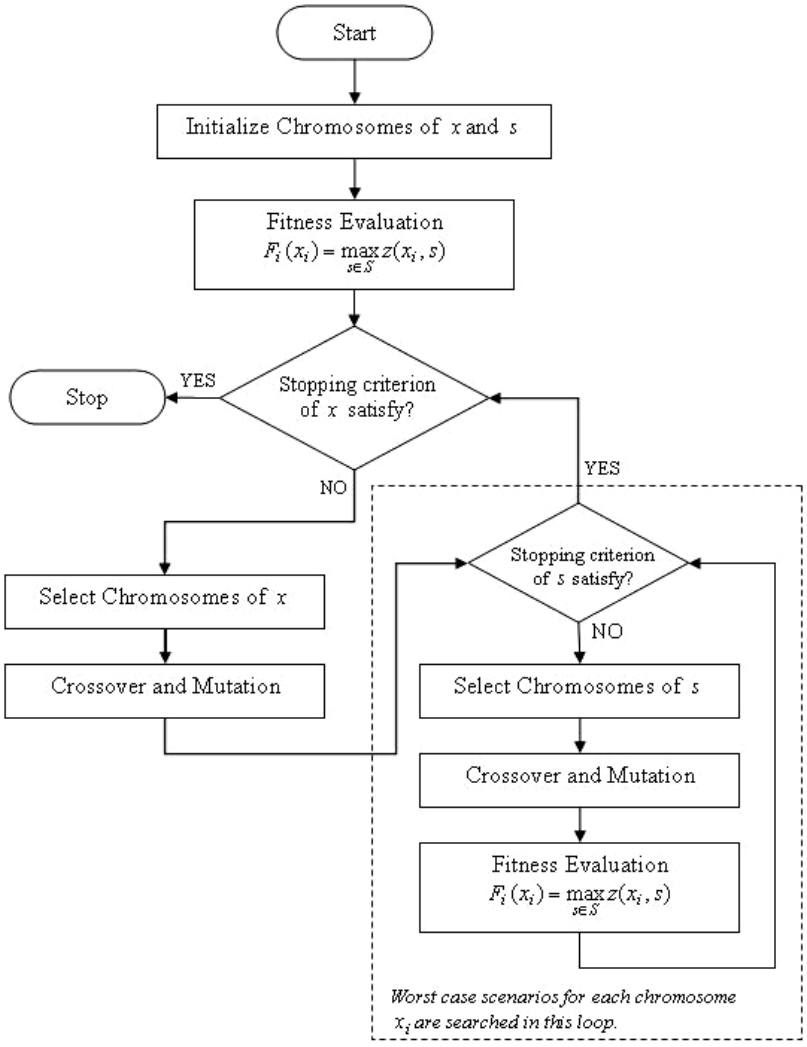


Figure 3.13 GA process for minimax optimization

Table 3.7 GA parameters

Genetic Algorithms	
Parameters	Value
Representation	Binary
Population size	30, 50 (x and s)
Max. iteration	40
Crossover	Single point (rate 0.8)
Selection	Stochastic uniform
Mutation	Random (rate 0.01)

3.3.4 Robust scheduling and rescheduling

In order to demonstrate the performance of the developed scheduling system, 3 experiments shown below are raised under disturbance and uncertainty environment.

1. *Case I*: An establishment of an optimal scheduling/rescheduling strategy.
2. *Case II*: The robust rescheduling in case of machine failure.
3. *Case III*: The robust rescheduling in case of rush order.

In the first experiment, the scheduling and rescheduling strategy will be appropriately established and it is utilized for the remaining experiments. The second experiment finds the best worst case schedule when caster failure occurs and its performance is pointed out. The suitable stability factors in each delay time are also presented. In the third experiment, the best worst case schedule for rush order is found and its performance is investigated.

3.3.4.1 *Case I: Establishment of rescheduling strategy*

As discussed in the previous subsection, the strategy contains the policy and schedule repair method. To define the repair method, the behavior of the objective function through varying the stability factor (β) is firstly studied. During varying β from 0 to 1, the optimization without disruptions and uncertainty is done. Each β is replicated 5 times in order to facilitate statistical analysis. The average value of the objective parts such as the total queue time (q_T), total cast break time (cb_T), efficiency (*eff*), and stability (*stb*) will be observed.

The results are shown in Table 3.8 and the normalized trend is shown in Figure 3.14. It is noted that in addition to the stability-oriented cases (low β) providing a bad efficiency (high efficiency value), the cast break is also bad while the stability and queue time trend are decreased. Regarding the parameter trends, the scheduling/rescheduling strategy can be defined as follows:

- *S1-Normal scheduling*: Because it is a pre-scheduling, the stability will be not significant while the efficiency is considered. Therefore, the objective function in this case is set for the stability factor to be 1 ($\beta = 1$).
- *S2-Swap schedule repair*: It is a partial rescheduling for the rush order case. In this situation, some products (charges or heats) must be replaced by the rush orders. Since rescheduling has to be done through the production line, the efficiency should be therefore emphasized by choosing the high stability factor. Thus, $\beta = [0.6, 1.0]$ is used for this case. The exact stability factor will be shown in *Case III.*
- *S3-Shift schedule repair*: It is another partial rescheduling for the machine failure case. This repair called right-shifts only postpones each remaining heats by the delay time. It is applied in case of the remaining molten steel after the caster breakdown/failure is enough (practically, > 60% of the ladle capacity). It is not necessary to re-sequencing because the molten steel ladle can be reheated at the refining furnace without adding steel and returning to be casted again with the initial schedule.
- *S4-Complete rescheduling*: It is the rescheduling in the case of machine failure and there is little remaining molten steel in the current ladle at a caster (is not enough to regard as a new charge or heat). This remaining molten steel will be mixed/added to a successor charge. Therefore re-sequencing all remaining heats is necessary. Because unchanging the schedule at the upstream line is needed, thus a low stability factor, which provides the low effect to upstream line, has to be chosen. This case, $\beta = [0.2, 0.4]$, which is close to the compromising point between the total cast break and queue, is appropriated. The exact stability factor will be shown in *Case II.*

Table 3.8 Mean value of each objective parts varying β

β	Objective Parts				
	Queue (min)	Cast break (min)	Efficiency ($)	Stability ($)	z ($)
1	203.4	15.4	15255	2886.4	15255
0.8	179.4	22.9	14750	2184.9	11346
0.6	121.3	28.7	15834	2098.9	10340
0.4	108.3	28.7	17458	2332.3	8652
0.2	84.5	28.7	19555	2174.5	5650
0	50.1	56.6	30676	1514.9	1515

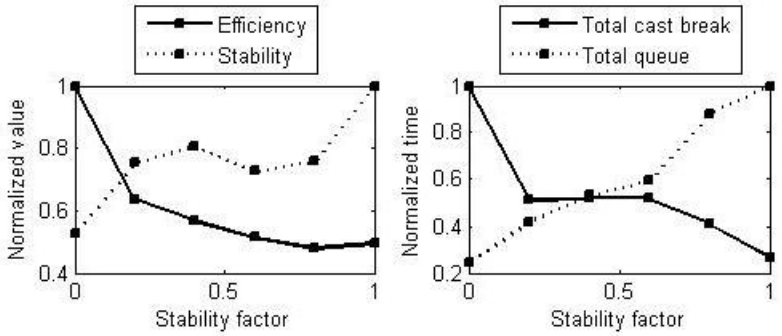


Figure 3.14 Relation of objective parts by varying β

3.3.4.2 *Case II: The Robust Rescheduling in Case of Machine Failure*

Since this case study is the rescheduling in an unavailable resource (machines) situation, the stability and uncertainty effect will be demonstrated. Regarding the defined strategy, objective function, and optimization parameters in Table 3.8, the following conditions are considered in this rescheduling case:

- The initial schedule shown in Table 3.9 is used before a disruption occurs. The steel production is simulated under the stochastic processing time.
- The machine failure disruption is assumed to occur on Caster 1 during casting the 2nd charge (while the 5th change is on Caster 2).
- The remaining molten steel in ladle is assumed to be less (< 60% of the ladle capacity). Therefore strategy "S4" is chosen and all remaining charges (i.e., 3rd and 6th-12th charges) will be rescheduled.
- To study the impact of the delay time from the machine breakdown (d_{br}) on each stability factor (β = 0.2, 0.4, and 0.6), the delay time of Caster 1 is assigned to 20, 40, 60 minutes. Then the exact stability factors, which are suitable for each delay time, will be defined.
- The robust performance of the revised schedule will be demonstrated by comparing the revised schedules without taking stability (β = 0.1) and uncertainty into account.

Table 3.9 Original schedule (from a factory)

Cast No.	Charge No	Consign-ment date	Grade Serial	Steel grade	Width (mm)	Thickness (mm)	Weight (kg)	Caster	
								1	2
1	1	16/12/07	D020	1008MnDK2	1272	60	14300	×	
	2	16/12/07	D020	1008MnDK2	1258	60	14300	×	
	3	16/12/07	D020	1008MnDK2	1252	50	14300	×	
2	4	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
	5	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
	6	16/12/07	D013	1008MnDK-M3	1272	60	24500		×
3	7	16/12/07	D013	1008MnDK-M3	1552	50	14300	×	
	8	16/12/07	D013	1008MnDK-M3	1245	50	16000	×	
	9	16/12/07	D021	1008MnDK2	1240	60	16000	×	
4	10	16/12/07	G001	1007DKGXA-00	1272	60	14300		×
	11	16/12/07	G001	1007DKGXA-00	1255	60	19500		×
	12	16/12/07	G001	1007DKGXA-00	1275	50	24500		×

The optimization results are shown in Table 3.10. In the table, the means of objective parts (i.e., q_T, cb_T, *eff*, and *stb*) from MCS with 5000 replications are observed in each β and delay time. Considering the effect of the delay time on those parts, it is noted that increasing the delay time in every β results to increase of q_T, *eff*, and *stb* while only cb_T is decreased. It makes sense that q_T is increasingly proportioned with the increment of the delay time due to the postponement of the starting time and it cannot be reduced by the optimization. Consequently, cb_T is unavoidably minimized instead of q_T. Due to the large penalty factor of cb_T, therefore *eff* is also reduced. Similarly, the postponement of the starting time makes the *stb* tend to be increased.

For the effect of increasing β on the objective parts, the results in Table 3.10 mention in the same trend of *Case I* as shown in Figure 3.15, 3.16, and 3.17. By the figures, a suitable β in each d_{br} is considered. β = 0.4 is the best choice for all d_{br} because it can provide lower *eff* and *stb* simultaneously while provide admitted values of q_T and cb_T (although they are not the lowest values, besides in the case of d_{br} =20).

Table 3.10 Mean and worst case value of the objective parameters

β	d_{br} (min)	Mean value of each objective part				Worst case value of *z*
		q_T (min)	cb_T (min)	*eff* ($)	*stb* ($)	
0.2	20	85.8	71.4	27347	4220.0	9203.1
	40	90.1	57.3	26727	4236.6	9444.3
	60	112.2	42.5	25579	4224.3	9283.4
0.4	20	67.4	46.2	26551	4144.5	14468
	40	91.4	55.2	26589	4206.4	13952
	60	114.7	42.5	25236	4225.3	14142
0.6	20	81.2	72.3	27460	4221.9	18530
	40	97.7	52.3	26418	4249.6	18552
	60	124.8	43.1	25090	4243.9	18634
1.0	20	94.5	52.4	25604	4478.3	27772

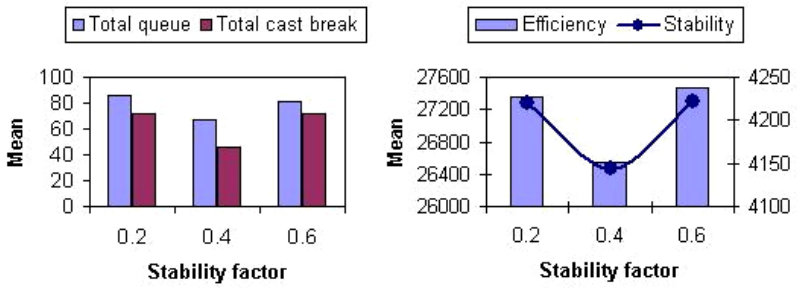


Figure 3.15 The relation between β and the objective parts at d_{br} = 20 min

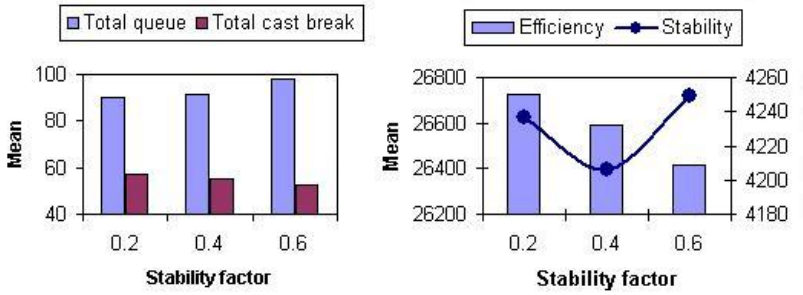


Figure 3.16 The relation between β and the objective parts at d_{br} = 40 min

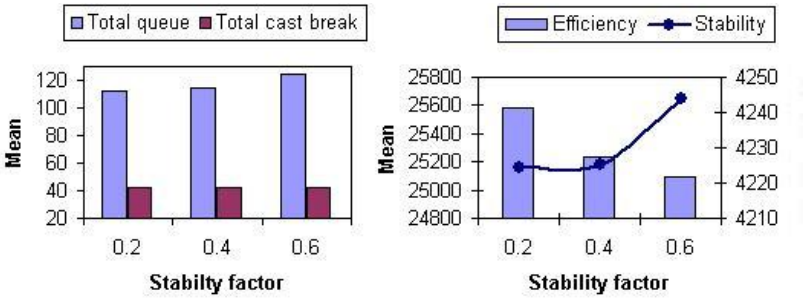


Figure 3.17 The relation between β and the objective parts at d_{br} = 60 min

Considering the schedule performance, the scheduling robustness is demonstrated in two senses; the robustness against the uncertainty and the robustness in terms of stability. The best worst case result in the case of β = 0.4 with d_{br} = 20 (in Table 4.9) is used as a case study for the performance analysis. Its optimization behavior is shown on Figure 3.18.

To prove the robustness against the uncertainty, it is compared with the best deterministic schedule, which is optimized without taking the uncertainty into account (without the minimax approach). The comparison of MCS results (with 200 replications from 5,000 replications) is illustrated in Figure 3.19(a) and (b). Figure 3.19(a) reveals that almost all objective values from MCS (>95%) are lower than the best worst case objective value ($z = 14468$) while most objective values from MCS in Figure 3.19(b) are higher than the best worst case objective value. It means that the best worst case schedule is more robust to the uncertain processing time than the best deterministic schedule.

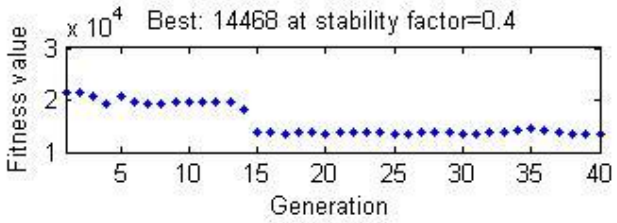


Figure 3.18 GA searching results in case of β = 0.4 at d_{br} = 0.2

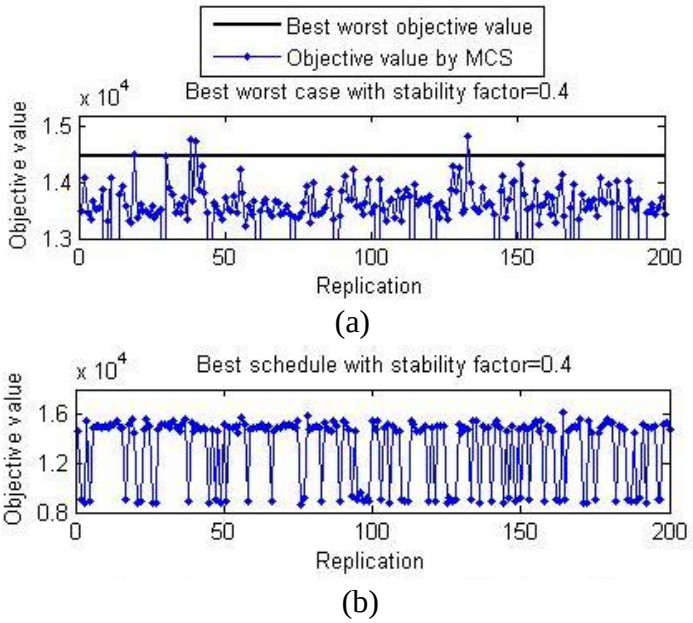


Figure 3.19 MCS results with β =0.4 at d_{br} = 20 min; (a) the best worst case schedule optimized by minimax approach and (b) the best deterministic schedule optimized without uncertainty

To guarantee the robust performance of the best worst case schedule, all high objective values (which are evaluated from bad scenarios) of both schedules; the best deterministic schedule and the best worst case schedule, are considered in terms of PDF as shown in Figure 3.20. By the confidence interval approach, it is proved that the best worst case objective value is out of the confidence interval of the PDF from the best worst case schedule.

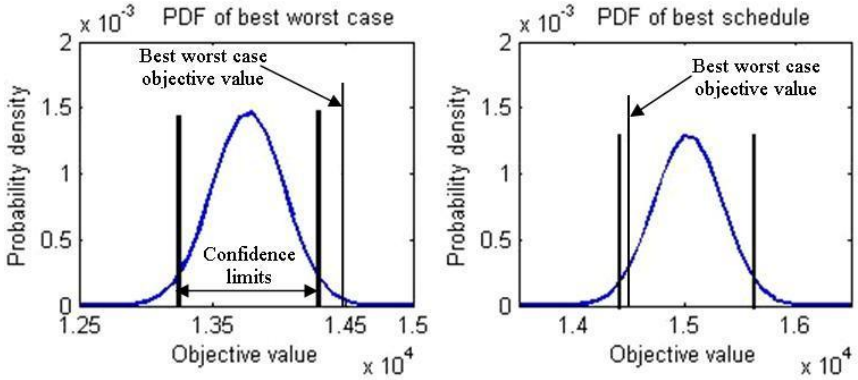


Figure 3.20 Comparison of PDF between the best worst case schedule and the best deterministic schedule in case of β = 0.4 at $\boldsymbol{d_{br}}$= 0.2

To demonstrate the robustness in terms of stability, the best worst case schedule in the case of =0.4 at d_{br} = 20 is compared with the best worst case schedule in case of β = 1

at d_{br} = 20 (which is evaluated without taking the stability into account) as shown in Table 3.10. It is noted that the mean stability value of the last schedule is higher than the first schedule while mean efficiency is lower than the first schedule. It mentions that the revised schedule with β =0.4 is more similar to the initial schedule than the revised schedule with β = 1, according to the schedule structures in Table 3.11 and 3.12.

Table 3.11 Revised schedule in case of β = 0.4 (machine failure case)

Cast No.	Charge No	Consign-ment date	Grade Serial	Steel grade	Width (mm)	Thickness (mm)	Weight (kg)	Caster	
								1	2
	1	16/12/07	D020	1008MnDK2	1272	60	14300	×	
1	2	16/12/07	D020	1008MnDK2	1258	60	14300	×	
	4	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
2	5	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
	3	16/12/07	D020	1008MnDK2	1252	50	14300	×	
3	10	16/12/07	G001	1007DKGXA-00	1272	60	14300	×	
	11	16/12/07	G001	1007DKGXA-00	1255	60	19500	×	
	12	16/12/07	G001	1007DKGXA-00	1275	50	24500	×	
	6	16/12/07	D013	1008MnDK-M3	1272	60	24500		×
4	7	16/12/07	D013	1008MnDK-M3	1552	50	14300		×
	8	16/12/07	D013	1008MnDK-M3	1245	50	16000		×
	9	16/12/07	D021	1008MnDK2	1240	60	16000		×

Table 3.12 Revised schedule in case of β = 1.0 (machine failure case)

Cast No.	Charge No	Consign-ment date	Grade Serial	Steel grade	Width (mm)	Thickness (mm)	Weight (kg)	Caster	
								1	2
	1	16/12/07	D020	1008MnDK2	1272	60	14300	×	
1	2	16/12/07	D020	1008MnDK2	1258	60	14300	×	
	4	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
2	5	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
	3	16/12/07	D020	1008MnDK2	1252	50	14300	×	
3	6	16/12/07	D013	1008MnDK-M3	1272	60	24500	×	
	7	16/12/07	D013	1008MnDK-M3	1552	50	14300	×	
	8	16/12/07	D013	1008MnDK-M3	1245	50	16000	×	
	9	16/12/07	D021	1008MnDK2	1240	60	16000	×	
4	10	16/12/07	G001	1007DKGXA-00	1272	60	14300		×
	11	16/12/07	G001	1007DKGXA-00	1255	60	19500		×
	12	16/12/07	G001	1007DKGXA-00	1275	50	24500		×

3.3.4.3 *Case III: Rescheduling with Uncertainty in Case of Rush Order*

Under the same circumstance of the machine failure case, a planner receives a rush order disruption for a charge as shown in Table 3.13. The following conditions are considered in this rescheduling case:

- The rush order disruption is assumed to occur during casting the 2nd charge. Thus, the 3rd and 6th to 12th charges have to be re-arranged
- The strategy "S2" is chosen. In this case, a replaced charge and optimal revised schedule have to be searched.

- To define the exact β that is suitable for the rush order case, β is varied between 0.6-1.0 and other parameters are observed.

Table 3.13 Rush order

Consignment date	Grade Serial	Steel grade	Width (mm)	Thickn. (mm)	Weight (kg)
14/12/07	D011	1008MnDK-M1	1230	60	16000

The results in each β are used to run MCS with 5,000 replications and the mean value of parameters are shown in Table 3.14. The optimal revised schedule is achieved by replacing the 7th charge. The results indicate that the case of β = 0.8 is the compromised solution because in addition to the low stability value, it provides the shortest total queue time while it still keeps the admitted cast break time.

To compare the robust performance, the result in the case of β = 0.8 is used to compare with the initial schedule, which is swap the 7th charge by the rush order. MCS results shown in Figure 3.21 indicate that the new schedule (with β = 0.8) is a better solution because of the higher robustness (the modified initial schedule provides the objective value equal \$26750 at the worst scenarios). The structure of the revised schedule in this case is shown in Table 3.15.

Table 3.14 Optimized results in machine failure and rush order case

β	Mean value of objective parameters				Worst case value of z
	q_T (min)	cb_T (min)	*eff* (\$)	*stb* (\$)	
0.6	148.1	69.8	30451	3653.4	25673
0.8	138.6	70.3	29012	3612.4	26297
1.0	150.9	82.9	27865	4110.6	28096

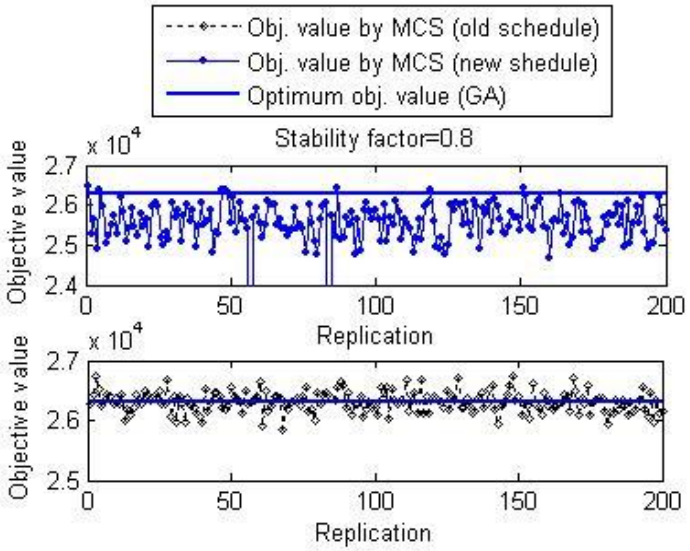


Figure 3.21 Comparison of MCS in case of the revised schedule with β =0.8 and the modified initial schedule

Table 3.15 Revised schedule in case of rush order

Cast No.	Charge No	Consign-ment date	Grade Serial	Steel grade	Width (mm)	Thickness (mm)	Weight (kg)	Caster 1	Caster 2
	1	16/12/07	D020	1008MnDK2	1272	60	14300	×	
1	2	16/12/07	D020	1008MnDK2	1258	60	14300	×	
	4	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
2	5	16/12/07	D011	1008MnDK-M1	1230	60	16000		×
	3	16/12/07	D020	1008MnDK2	1252	50	14300	×	
3	6	16/12/07	D013	1008MnDK-M3	1272	60z	24500	×	
	8	16/12/07	D013	1008MnDK-M3	1245	50	16000	×	
	9	16/12/07	D021	1008MnDK2	1240	60	16000	×	
	7	**14/12/07**	**D011**	**1008MnDK-M1**	**1230**	**60**	**16000**		**×**
4	10	16/12/07	G001	1007DKGXA-00	1272	60	14300		×
	11	16/12/07	G001	1007DKGXA-00	1255	60	19500		×
	12	16/12/07	G001	1007DKGXA-00	1275	50	24500		×

3.4 Robust Scheduling under Uncertain Disruption in CC Process

Several disruptions (e.g., machine failure, quality problem, etc.) in a steel production often occur in practice and lead to a delay in the production. A robust predictive scheduling, which takes the disruption effect into account, is a more suitable choice for facing the disruption. In this section, a worst case performance scheduling via Minimax optimization for CC process is presented.

3.4.1 Uncertain Daily Disruptions

Many disruptions that occur during steel production mainly can be focused on 4 groups (Lee et al., 1996); 1) Machine breakdown/failure. 2) A new, high-priority order (rush order) is introduced. 3) Excessive defects occur during an operation. 4) An order is canceled. In case of the manufacturing breakdown and coming or canceling order, changing the current schedule cannot be avoided. Differently, some disruptions such as the failure in some machine parts or the defect problem only delay the production. A complete rescheduling is not necessary. Right-shifting, which postpones each remaining operations affected by the disruptions as shown in Figure 3.22 should be adopted. For instance, an inefficient cooling from the damage or clog in some spray nozzles in the caster will press the operator to decrease the casting speed. The disruptions in the relevant processes such as a scrap jam in the melting can result decrease of casting speed. This kind of disruptions, which daily occur and much delay the operation, is considered in this paper and called “uncertain daily disruption”.

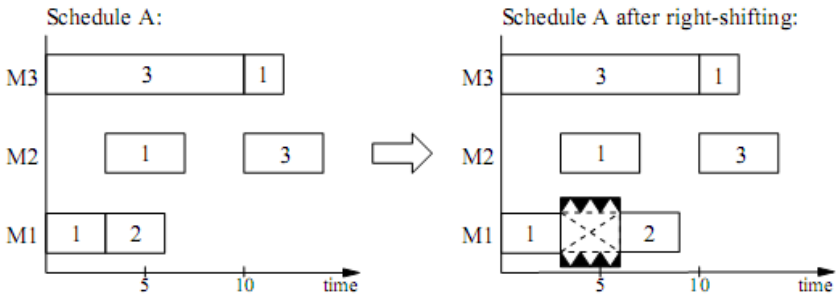


Figure 3.22 Right-shift rescheduling (Jensen, 2001)

To characterize the uncertain daily disruption, the failure distribution time and failure duration have to be identified. In steel factories, these disruptions are recorded in the enterprise resource planning (ERP) or the maintenance management system (MMS) for managing the machine reliability/availability. In this work, actual data from a factory for a period of 10 months that approximately contain 40 samples in each process are investigated. They are utilized to model the disruption in terms of a probability density function (PDF) in each process by Arena simulation software (version 10). Most results are fitted into the beta distribution as expressed in Table 3.16. Since the raw data is investigated per a product lot containing 12-14 heats, the failure distribution time is shown in the heat number unit.

Table 3.16 Disruption distribution in each process

Breakdown Parameter	Range	Expression
Melting		
Failure heat no.	1-12	1 + 10*Beta (0.75, 0.67)
Duration (min)	5-33	5 + 28*Beta (0.75, 1.05)
Refining		
Failure heat no.	1-12	1 + 10*Beta (0.70, 0.75)
Duration (min)	5-36	5 + 28*Beta (0.73, 1.05)
Casting		
Failure heat no.	1-12	1 + 10*Beta (0.96, 0.65)
Duration (min)	4.5-33.5	4.5 + Logn (15.10, 13.30)

3.4.2 Scheduling based on Minimax Formulation

3.4.2.1 *Minimax Formulation*

To deal with the uncertainty, the robust discrete optimization is applied in this work. In general, it can be formulated in Eq. (3.15) ($\min_{x\in X}\max_{s\in S} z(x, s)$) as follows: Let X be the set of all solutions, S be the set of all possible scenarios and $z(x,s)$ be an objective function of solution $x \in X$ in scenario $s \in S$. The problem is to find the best solution on the worst-case scenario. In other words, it is the same as minimizing (overall solutions) the maximum (overall scenarios) cost. In this section, X is the set of the feasible schedules while S is the set of the future disruption events. By the characteristics in Table 3.16, the disruption set is defined as $S = \{S_m, S_h, S_d\}$, where S_m is the set of the failure machine that is defined as $S_m = \{\text{EAF\#1}, \cdots, \text{CC\#2}\}$. S_h is the set of the failure heat number in each process that is defined as $S_h = \{1, \cdots, 12\}$. S_d is the set of the failure duration, which is defined as $S_d = [0, 40]$.

3.4.2.2 *Genetic Coding for Minimax Optimization*

To code the chromosome for minimax problem, the feasible solutions and scenarios are coded into the binary chromosomes in a form of $x = (X_1, \cdots, X_P, t_i, \cdots, t_N)$ and $s = (s_m, s_h, s_d)$, where X_i is the decision vector in Eq. (2) and t_i is the starting time of charge i. s_m, s_h, s_d are the machine number, heat number, and failure duration, respectively. Other parameters of GA, which are defined by a Trial and Error method, are shown in Table 3.17.

Table 3.17 Optimization parameters

Genetic Algorithms	
Parameters	Value
Representation	Binary
Population size	x =40 and s =50
Max. iteration	50
Crossover	Two point (rate 0.8)
Selection	Stochastic uniform
Mutation	Random (rate 0.01)

3.4.3 Experiment and Result

In this section, the best worst case schedule of a production lot that contains 12 charges as shown in Table 3.2 is implemented. We would like to optimally arrange the orders in to 4 casts (grade groups). In the experiment, it is supposed that the production is attacked by a single disruption event (a scenario per a disruption event) and can happen in all processes (melting, refining, and casting) and all machines.

The optimization result in Figure 3.23 shows that the objective value is continuously decreased and the best worst case solution is $z = 13{,}519$. The best worst case schedule allocates the charges into each cast and machine as shown in Table 3.18. The starting time in each charge can be shown in Gantt-chart in Figure 3.24.

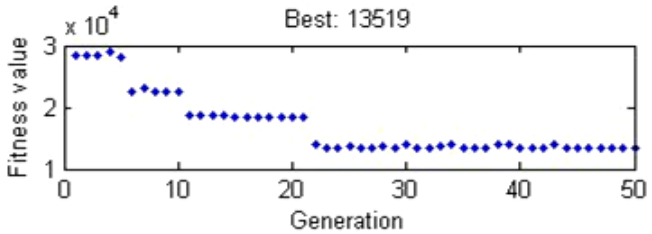


Figure 3.23 Objective value in each generation

Table 3.18 Best worst case schedule and worst disruption

	Result	Machine
Schedule	Cast#1 ={2,3} Cast#2 ={10, 11, 12} Cast#3 ={1, 4, 5} Cast#4 ={6, 7, 8, 9} Starting time = Fig. 5	CC#1 CC#1 CC#2 CC#2
Disruption	Charge no.=1 Start at min. 15th Duration =39 mins	CC#2

The solution provides the robust performance via MCS and the result is compared with the best deterministic schedule as shown in Figure 3.25. It reveals that although the best deterministic schedule provides a lower mean objective value, the best worst case schedule can provide a significantly better objective value when the disruption attacks (the bad scenario).

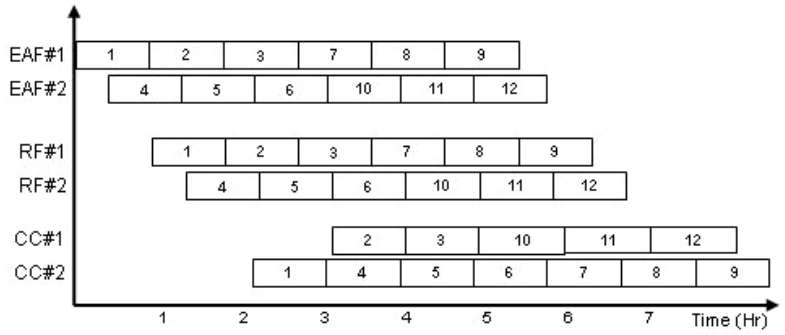


Figure 3.24 Gantt-chart of the best worst case schedule

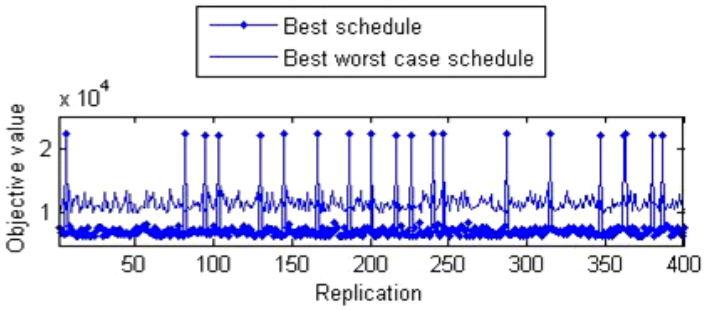


Figure 3.25 Comparison between the best worst case and best deterministic schedules

Table 3.19 Statistical values from MCS result with 10,000 replications

	Best worst case schedule	Best schedule
μ	12112	8550.1
σ	878	4435.8

When MCS is run with 10,000 replications, it still confirms the same trend as shown in Table 3.19. The best deterministic schedule provides huge standard deviation of the MCS result. In addition, the optimization results in Table 3.18 also points to the worst disruption for this best worst case schedule. It attacks on CC#2 around charge no. 1. Since this disruption produces the biggest impact in the production, it is mentioned to be the critical zone.

CHAPTER 4 SIMULATION-BASED MES ON PRODUCTION SCHEDULING IN STEELMAKING-CONTINUOUS CASTING PLANT

4.1 Deterministic Scheduling for SCC plant

This section formulates a SCC optimization model that more closely represents real-world situations and a hierarchical genetic algorithm (HGA) tailored particularly for searching an optimal SCC schedule. The developed model is achieved by integrating 2 main planning phases of the traditional scheduling, i.e., (1) planning cast sequence, and (2) scheduling of a steel-making and timing of all jobs.

4.1.1 Proposed optimization model

The performance criteria for the SCC scheduling need to also satisfy the steel technological constraints and special requirements described in Section 3.2. The proposed model extends the efficiency model of CC process in Section 4.1 into a mix integer programming model to satisfy the purpose of combining the processes of cast sequencing and job timing. The objective is created in terms of the production costs as follows:

$$\begin{aligned} f(\mathbf{X},\mathbf{Y},st) \;=\; & \frac{1}{2}\left(\sum_{m=1}^{M_C}\sum_{k\in P_m}\mathbf{x}^{(k)\mathrm{T}}\mathbf{Q}_C\mathbf{x}^{(k)}+\sum_{m=1}^{M_F}\mathbf{Y}^{(m)\mathrm{T}}\mathbf{Q}_F\mathbf{Y}^{(m)}\right) \\ & +\sum_{k=1}^{P}\sum_{i,\,j\in\Omega_k} C_B*(st_j^{(m)}-st_i^{(m)}-T_i^{(m)}) \\ & +\sum_{i=1}^{N}\sum_{m,n\in\Pi_i} C_W*(st_i^{(m)}-st_i^{(n)}-T_i^{(n)}) \\ & +\sum_{i=1}^{N}\sum_{m\in\Pi_C} C_L*\max(0,\, st_i^{(m)}+T_i^{(m)}-d_i) \\ & -\sum_{i=1}^{N}\sum_{m\in\Pi_C} C_E*\min(0,\, st_i^{(m)}+T_i^{(m)}-d_i) \end{aligned} \tag{4.1}$$

subjected to:

- For binary variables (Assignment)

$$\sum_{k=1}^{P} x_i^{(k)} = 1 \tag{4.2}$$

$$\sum_{m=1}^{M_E} y_i^{(m)} = 1 \quad \text{and} \quad \sum_{m=(M_E+1)}^{M_F} y_i^{(m)} = 1 \tag{4.3}$$

$$2 \le \sum_{i=1}^{N} x_i^{(k)} \le L \tag{4.4}$$

$$x_i^{(k)},\, y_i^{(m)} \in \{0,1\} \tag{4.5}$$

- For continuous variables (Time relations)

  Melting and Refining ($m \in \Pi_F$)

$$st_i^{(m)} \geq st_i^{(n)} + T_i^{(n)} \quad , \; m,n \in \Pi_i \tag{4.6}$$

$$st_j^{(m)} \geq st_i^{(m)} + T_i^{(m)} \quad , \; m \in \Pi_i \tag{4.7}$$

  Continuous Casting ($m \in \Pi_C$)

$$st_j^{(m)} \geq st_i^{(m)} + T_i^{(m)} + \delta^{(m)} \quad , \; i \in \Omega_k, j \in \Omega_{k+1} \tag{4.8}$$

$$st_j^{(m)} = st_i^{(m)} + T_i^{(m)} \quad , \; i,j \in \Omega_k \tag{4.9}$$

  where

$$\mathbf{X}^{(k)} = \begin{bmatrix} x_1^{(k)} \\ \vdots \\ x_N^{(k)} \end{bmatrix}, \quad \mathbf{Y}^{(k)} = \begin{bmatrix} y_1^{(m)} \\ \vdots \\ y_N^{(m)} \end{bmatrix} \tag{4.10}$$

$$\mathbf{Q}_C = \begin{bmatrix} q_{1,1} & \cdots & q_{1,N} \\ \vdots & \ddots & \vdots \\ q_{N,1} & \cdots & q_{N,N} \end{bmatrix}, \; \mathbf{Q}_F = \begin{bmatrix} c_{1,1}^G & \cdots & c_{1,N}^G \\ \vdots & \ddots & \vdots \\ c_{N,1}^G & \cdots & c_{N,N}^G \end{bmatrix}, \quad q_{i,j} = c_{i,j}^G + c_{i,j}^{WD} + c_{i,j}^{TH} \tag{4.11}$$

$$c_{i,j}^G = \begin{cases} 0 & \text{charge } i \text{ and } j \text{ are the same steel grade serial} \\ f_1 & \text{charge } i \text{ and } j \text{ belong to the same steel grade serial} \\ f_2 & \text{charge } i \text{ and } j \text{ are the different steel grade serial} \end{cases} \tag{4.12}$$

$$c_{i,j}^{WD} = f_3 * (\Delta wd_{i,j})^2, \quad \Delta wd_{i,j} = wd_i - wd_j \tag{4.13}$$

$$c_{i,j}^{TH} = f_4 * (\Delta h_{i,j})^2, \quad \Delta h_{i,j} = h_i - h_j \tag{4.14}$$

$$i,j \in \Omega \text{ and } j \text{ follows } i \tag{4.15}$$

$$m,n \in \Pi \text{ and } m \text{ follows } n \tag{4.16}$$

$$k = 1, \cdots, P \tag{4.17}$$

Decision variables

- $\mathbf{X}^{(k)}$ is a binary decision vector to allocate the charge into cast k on CC. Charge i belongs to cast k if and only if $x_i^{(k)}$ is equal to 1.
- $\mathbf{Y}^{(m)}$ is a binary decision vector to allocate the charge into EAF and RF m ($m \in \Pi_F$). Charge i is operated on machine m if and only if $y_i^{(m)}$ is equal to 1.
- $st_i^{(m)}$ is the starting time of the charge i on machine m. $st_i^{(m)}$ defines the sequence of the enabled members (equal to 1) of $\mathbf{X}^{(k)}$ and $\mathbf{Y}^{(m)}$. $st_i^{(m)}$ that the highest value will be the first charge of machine m or cast k.

Other notations

- $\mathbf{Q}_C$ is a cost matrix for CC and $\mathbf{Q}_F$ is a cost matrix for both EAF and RF. $q_{i,j}$ is the production cost between charge i and j due to losses as follows: $c_{i,j}^G$, $c_{i,j}^{WD}$, and $c_{i,j}^{TH}$ are losses of steel grade difference, loss of width change, and loss of thickness change, respectively.

- Π is a set of all machines, $\Pi = \{1, 2, \ldots, M\}$, where M is the total number of machines.
- Π_F is a set of steel making furnaces (EAF and RF), $\Pi_F = \{1, 2, \ldots, M_F\}$, where M_F is the total number of EAF and RF. $\Pi_F \subset \Pi$. M_E is total number of EAF and M_R is total number of RF, where $M_E + M_R = M_F$. Π_C is the set of CC, $\Pi_C = \{1, 2, \ldots, M_C\}$, where M_C is total number of casters. $M_C + M_F = M$, $\Pi_C \subset \Pi$, and $\Pi_C \cap \Pi_F = \varnothing$. Π_i is the set of machines used for charge i, $\Pi_i \subset \Pi$.
- P is the total number of casts and P_m is a set of casts that are processed on caster m. For example, $P_1 = \{1, 3\}$ means the cast no. 1 and 3 are operated on the caster no. 1.
- Ω is a set of all charges, $\Omega = \{1,2,..., N\}$, where N the total number of charges to be arranged. Ω_k is a set of charges in cast k
- L is the maximum number of charges in a cast.
- $f_1, \cdots, f_4$ are penalty factors.
- wd_i is the ordered slab width of charge i. h_i is the ordered slab thickness of charge i and d_i is the consignment date or time of charge i
- $T_i^{(m)}$ is processing time of charge i on machine m. $\delta^{(m)}$ is the set up time of caster m
- C_W, C_B, C_L and C_E are the penalty factors of waiting time (W), cast break time (B), lateness (L), and earliness (E), respectively.

The last 4 terms in Eq. (4.1) represent the cast break loss, waiting time, lateness, and earliness penalties, respectively. While the real decision variable ($st_i^{(m)}$) is used for timing the charges on the machines, it also implies the charge sequence. The constraint in Eq. (4.2) ensures that every charge must be arranged to a cast while the constraint in Eq. (4.3) ensures that every charge must be arranged once to a machine in each process. Eq. (4.4) ensures that the number of charges in each cast must not be less than 2 and cannot exceed the endurance capability of each tundish. Eq. (4.6) ensures that the two consecutive operations of each charge are executed subsequently. Eq. (4.7) ensures that two contiguous charges are not processed simultaneously on the same machine. Eq. (4.8) ensures that enough setup time is required between casts while the constraint in Eq. (4.9) ensures the continuity of casting.

4.1.2 Designing hierarchical genetic Algorithm (HGA)

Since the gradient-based optimization methods are not practical to find an optimal schedule for the proposed mix-integer programming model, GA is applied in this work. Because this problem focuses on a particular type of applications, this section proposes a concept of hierarchical GA, which provides suitable GA operations to avoid infeasible solutions.

4.1.2.1 *Hierarchical Chromosome Coding*

A hierarchical chromosome coding for GA is introduced in Tang et al. (1998). This concept is treated as DNA in a biological chromosome. DNA consists of two types of genes, i.e., (1) structure genes, and (2) regulatory genes. The structure genes contain the genetic information while the regulatory genes control coding of the structural genes.

Similarly, a mathematic hierarchical chromosome can be classified into two different types, i.e., (1) parametric genes, and (2) control genes. The parametric genes, which contain parameter values, are analogous to the structure genes. The activation of the parametric genes is governed by the value of the control genes, which are analogous to the regulatory genes. In Figure 4.1, 3 level gene structures are illustrated. The parametric genes are governed by the 1st level control genes, which are governed by the 2nd level control genes, respectively. The value 0 or 1 of the control genes is used to define the activation of the parametric genes.

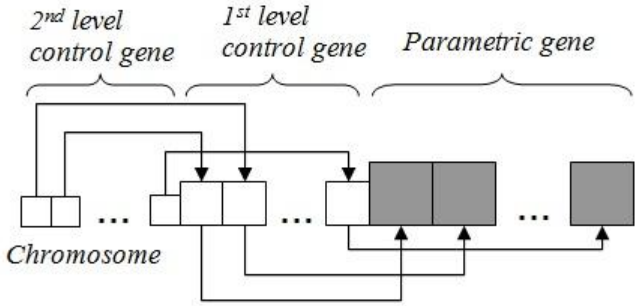


Figure 4.1 General form of the hierarchical chromosome

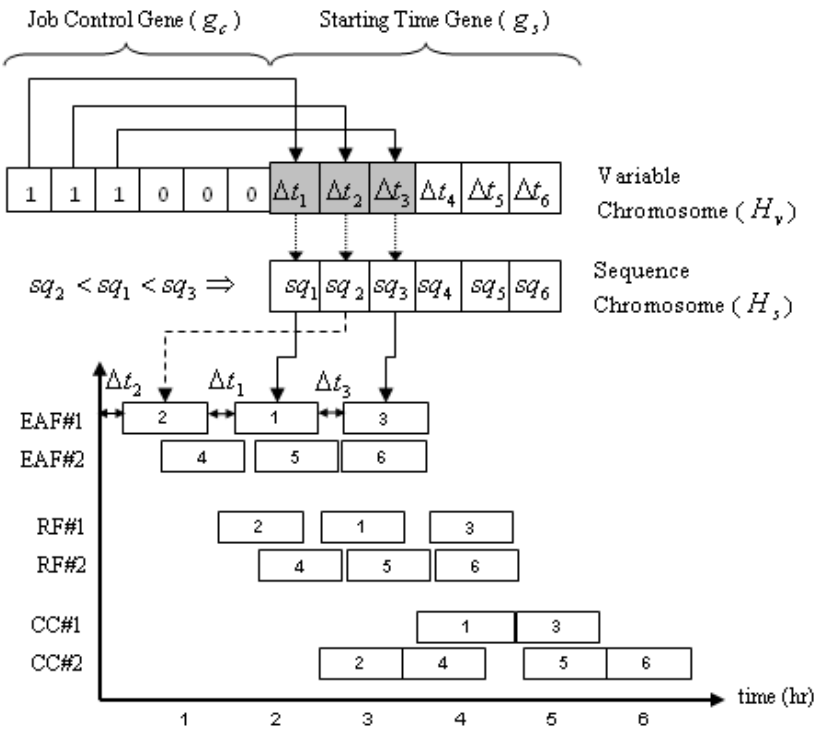


Figure 4.2 The proposed hierarchical chromosome

From the proposed optimization model, the decision variable, $\mathbf{X}^{(k)}$, $\mathbf{Y}^{(m)}$, and $st_i^{(m)}$ are coded into two kinds of chromosomes as follows:

1. Variable chromosome (H_v): The variable chromosome comprises a job control gene and a starting time gene. The job control gene (g_c) contains the binary variable $\mathbf{X}^{(k)}$ or $\mathbf{Y}^{(m)}$ while the starting time gene (g_s) contains the real variable $\Delta t_i^{(m)}$, which is a slack of the starting time according to Eq. (4.7) or $st_j^{(m)} = st_i^{(m)} + T_i^{(m)} + \Delta t_j^{(m)}$, j follows i. The value 1 of the job control gene will activate the associated starting time gene and address to the number of charges on machine m as shown in Figure 4.2.

$$g_s = \{\Delta t_1^{(1)}, \Delta t_2^{(1)}, \cdots, \Delta t_N^{(1)}, \quad \Delta t_1^{(2)}, \cdots, \Delta t_N^{(2)}, \quad \Delta t_1^{(m)}, \cdots \Delta t_N^{(m)}\} \tag{4.18}$$

2. Sequence chromosome (H_s): the sequence chromosome is aimed to arrange the activated starting time gene. It represents a set of integer numbers as follows:

$$H_s = \{h^{(E)}, h^{(R)}, h^{(C)}\} \tag{4.19}$$

$$h^{(E \text{ or } R \text{ or } C)} = \{sq_i \mid sq_i \in [1, N] \text{ and } sq_i \neq sq_j, \ i, j \in \Omega\} \tag{4.20}$$

where $h^{(E)}$, $h^{(R)}$, and $h^{(C)}$ are the subgroups of the sequence chromosome of EAF, RF, and CC, respectively. Each integer number (sq_i) shows the priority of each member ($\Delta t_i^{(m)}$) of the starting time gene. An activated charge with the smallest integer number will be processed at the earliest shown in Figure 4.2.

Figure 4.3 illustrates the chromosomes coding and decoding for the first melting machine (EAF#1). The job control gene is the variable $\mathbf{Y}^{(1)}$, in which the first three charges are enabled. Therefore, the slack $\Delta t_1^{(1)}$, $\Delta t_2^{(2)}$, and $\Delta t_3^{(3)}$ are selected. Then, these activated slacks are sequenced by the sequence chromosome, e.g., $sq_2 < sq_1 < sq_3$. Consequently, charge 2 is executed and is then followed by charge 1 and 3, subsequently. It is noted that the sequence of the activated charges for EAF#1 will be different if the priority in the sequence chromosome is different as shown in Figure 4.3. In the figure, Example 1 illustrates that charge 1 is firstly executed because its priority is the smallest ($sq_1 = 1$) while charge 2 is firstly executed in Example 2.

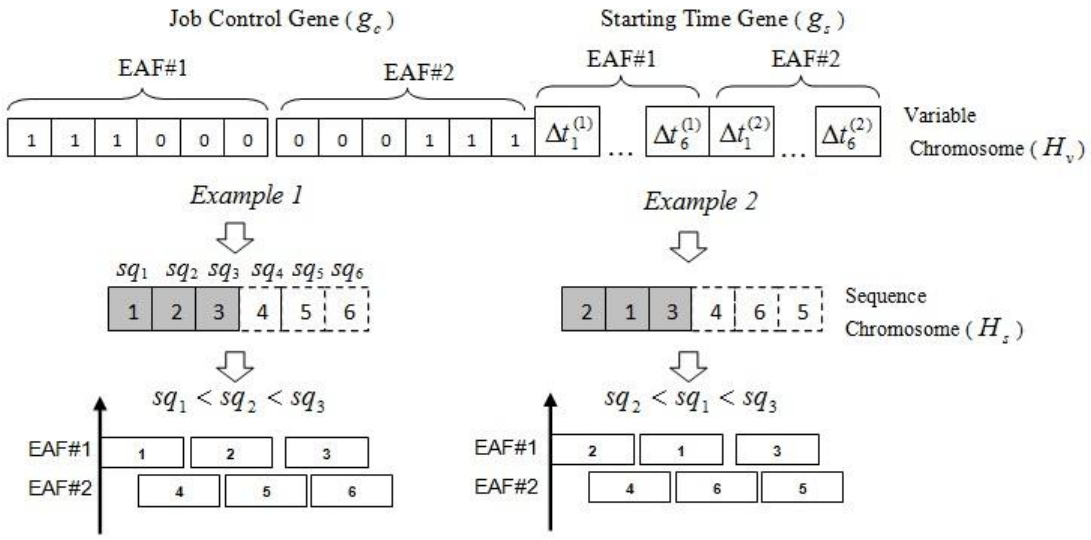


Figure 4.3 Decoding 2 different sequences to be a schedule

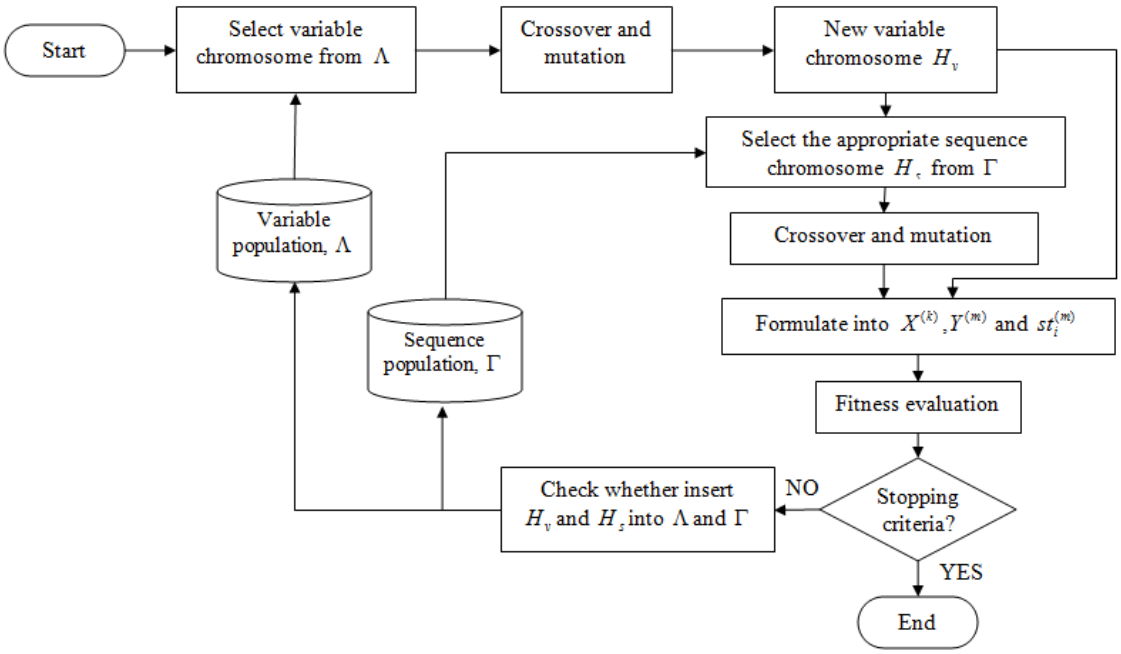


Figure 4.4 Genetic cycle for SCC scheduling

4.1.2.2 *HGA Operations*

Generally, a main operation of GA optimization consists of selection, crossover, and mutation. Since there are two types of genes in the chromosome, specific crossover and mutation method has been designed to suit this particular purpose. A block diagram of the proposed GA operation is illustrated in Figure 4.4.

- ***Variable Chromosome Crossover***: The crossover is performed separately for 2 types of genes. To avoid infeasible solution according to Eq. (4.2), firstly a parent chromosome of the control genes is formed by arranging all job control genes into a matrix. Then, a one-point crossover with a probability rate is performed via swapping the matrix columns shown in Figure 4.5. The offspring A is made from replacing the last two columns behind the crossover point of the parent A with the last two columns of the parent B. For the starting time gene, since it is a vector of real numbers, the standard one or multiple-point crossovers can be directly applied.
- ***Sequence Chromosome Crossover***: The standard one-point crossover is adopted with randomly choosing the fixed crossover points; a and b, as shown in Figure 4.6. An example shows that when the crossover point a is chosen, the offspring A is consequently created from swapping $h^{(R)}$ and $h^{(C)}$ between the parent A and parent B.

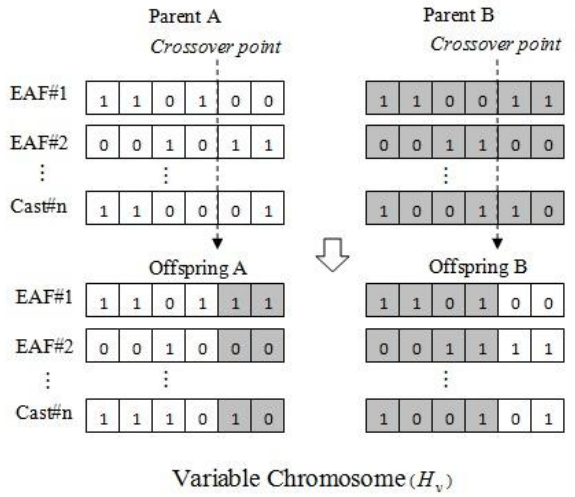


Figure 4.5 Crossover for the variable chromosome

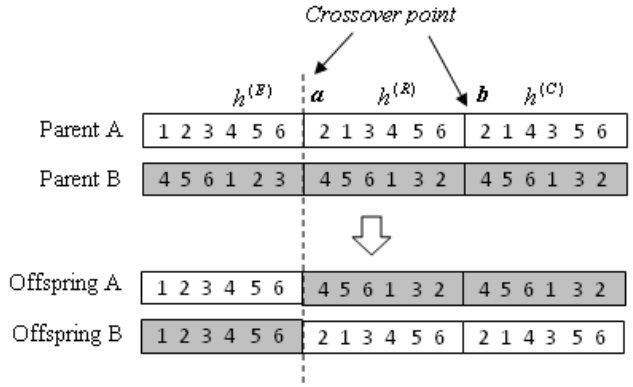


Figure 4.6 One-point crossover for sequence chromosome

1. ***Mutation***: In variable chromosome, a bit mutation is applied to the job control gene while the random mutation shown in Eq. (4.21) is applied to the starting time gene.

$$\Delta t_i^{(m)} = \Delta t_i^{(m)} + \psi(\mu, \sigma) \tag{4.21}$$

where ψ is a random function (maybe normally distributed), μ and σ^2 are mean and variance, respectively.

For the sequence chromosome, a special mutation operator has been desired to find an optimal sequence. A delta shift mutation (Man et al., 1999), which alters each element in the sequence chromosome, is adopted as follows:

$$sq_i^{(m)} = sq_{i+\Delta i}^{(m)} \tag{4.22}$$

where Δi has equal chance to be 1 and -1 with a small probability.

4.1.3 Computational Experiment

To verify the proposed optimization model and test the performance of the methodology, this section presents a case study of scheduling an SCC factory, which consists of 2 EAFs, 2 RFs, and 2 CCs. A daily production lot contains 12 charges shown in Table 4.1 (it is different from Table 3.2 in due time) and the factory wants to optimally arrange these orders into 4 casts (grade groups). The first two casts are processed on caster 1 while the third and the forth casts are processed on caster 2. Therefore, the related set in the model can be defined as follows: $\Pi = \{1,2,3,4,5,6\}$, $\Pi_F = \{1,2,3,4\}$, $\Pi_C = \{1,2\}$, $\Omega = \{1,2,...,12\}$.

Table 4.1 Production orders for SCC plant

Charge No	Due time (min)	Grade serial	Factory steel grade	Width (mm)	Thickness (mm)	Weight (kg)
1	320	SM400A	1008MnDK2	1272	60	14300
2	320	SM400A	1008MnDK2	1258	60	14300
3	320	SM400A	1008MnDK2	1252	50	14300
4	320	SS400	1008MnDK-M1	1230	60	16000
5	320	SS400	1008MnDK-M1	1230	60	16000
6	350	SS400	1008MnDK-M3	1272	60	24500
7	420	SS400	1008MnDK-M3	1552	50	14300
8	450	SS400	1008MnDK-M3	1245	50	16000
9	450	SM400A	1008MnDK2	1240	60	16000
10	480	SPHC	1007DKGXA-00	1272	60	14300
11	480	SPHC	1007DKGXA-00	1255	60	19500
12	500	SPHC	1007DKGXA-00	1275	50	24500

The experiments are implemented on MATLAB and are carried out on a 1.66 GHz Intel(R) Core 2 CPU personal computer with 4 GB memory.

4.1.4 Model Performance

Focusing on the performance of the proposed optimization model, the scheduling is optimized by using the proposed model with a standard GA (a MATLAB tool). The result will be compared with the 2-phase traditional scheduling that cast sequence is firstly defined (based on Xue et al., 2004) and then timing of all jobs is assigned. The model parameters are based on a real factory situation presented in Table 4.2. The penalty factors are approximately defined from the production costs in USD.

The best results (from 10 trials) from the proposed model and the 2-phase traditional approach are presented by a Gantt chart in Figure 4.7 and 4.8, respectively. The performance comparison is shown in Table 4.3. It reveals that the proposed model can provide a better total cost than the traditional approach. It is seen that although the traditional method can provide a somewhat better cast sequence plan in the first phase (considering only the cost from the grade mixing, size changing, and consignment

sequencing at CC), the total cost becomes undesirable in practice when the timing of all processes is assigned in the second phase.

Table 4.2 Optimization model and GA parameters for HGA

Optimization model		GA	
Parameters	Value	Parameters	Value
f_1, f_2, f_3, f_4	600, 14400, 0.07, 1.6	Representation	Binary
C_W, C_B, C_L, C_E	6.7, 14400, 16.2, 5.4	Pop. Size/Max. iter.	600/2000
N	12	Crossover	2 point (rate 0.8)
P	2	Selection	Stochastic unif.
L	4	Mutation	Bit (rate 0.01)

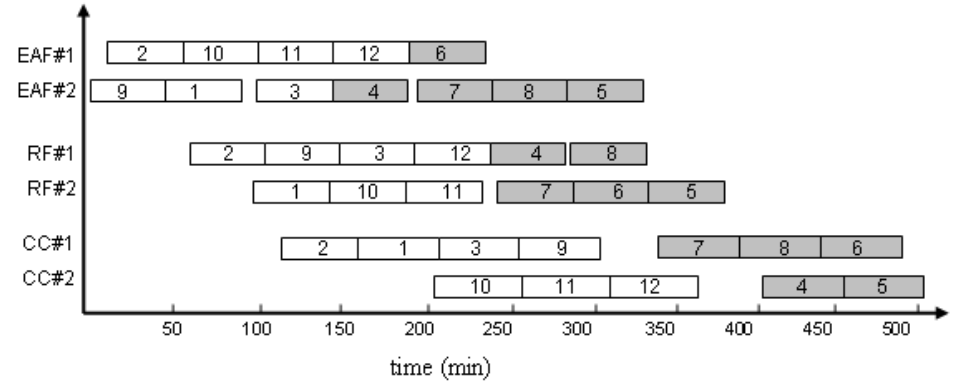


Figure 4.7 Best schedule from the proposed model

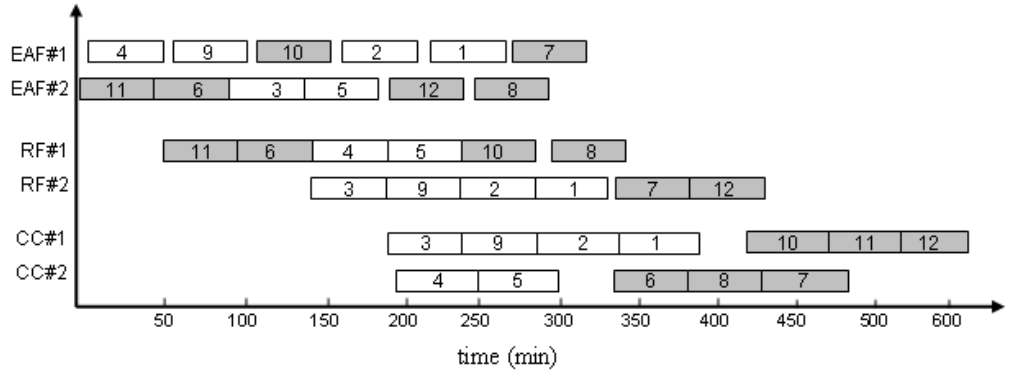


Figure 4.8 Best schedule from the 2-phase traditional approach

Table 4.3 Comparison of results (proposed model vs. 2-phase traditional approach)

Objective factors	Performance (per a production line)	
	Proposed model	2-Phase traditional approach
Optimal objective value ($\times 10^4$ USD)	2.16	2.41
Steel grade mixing (ton)	0	0
Width/ thickness change (ton)	12.8	10.4
Total consignment date delay (min)	218	226
Total cast break (min)	0	0
Total waiting time (min)	118	223
Total completion time (min)	506	619

4.1.5 Optimization Performance

Performance of the proposed methodology, which combines the proposed model and modified HGA, is demonstrated against the standard GA in this section. It is known that the quality of the optimal solution from GA frequently depends on the population size and the same initial solution does not always provide the same result over different searches due to uncertainty in the method. In practice, the factory needs an approach that provides an acceptable result under an appropriate computational time and a good reliability. Thus, the experiment is divided into 3 cases; small (200), medium (400), and large (600) population sizes. In each population size, 5 different test sets (initial set no.1-5) will be executed repeatedly 5 times in each test set (totally 25 samples) for observing the variation. The HGA and GA parameters are presented in Table 4.4.

The optimization results are presented in Table 4.5 to observe overall performance by comparing the best and average results from each population size through 25 trials. Considering the average result, the proposed approach can provide a lower average than the standard GA does for all population sizes although the averages of both approaches are almost identical in the case of small population size. The percentage differences are around 6.6% and 5.7% for large and medium population sizes, respectively, while it is around 0.6% for small population size. Furthermore, the proposed approach provides a smaller standard deviation.

When considering the best results, the proposed approach can provide a better solution than the standard GA at around 3.8% and 8.2% for medium and small population size (400 and 200), respectively. However, the standard GA can provide a better solution around 4.4% for large population size. It is probably due to a better jumping out of local optima of the MATLAB standard GA. It should also be noted that the ‘best’ result is the best one from only 25 samples in each case and there is no guarantee that one would get such the best solution every time. By considering average and best results, it can be

claimed that the proposed approach is more likely to provide a better solution than the standard GA does.

Table 4.4 HGA and GA parameters for methodology demonstration

GA Parameters	Proposed methodology		Standard GA
	H_v	H_s	
Representation	Binary, Real	Integer	Binary
Population size	{200, 400, 600}	{200, 400, 600}	{200, 400, 600}
Max. iteration	2000	2000	2000
Crossover	1 point (rate 0.8)	1 point (rate 0.8)	2 point (rate 0.8)
Selection	Stochastic uniform	Stochastic uniform	Stochastic uniform
Mutation	Random (rate 0.01)	Delta shift (rate 0.01)	Bit (rate 0.01)

Table 4.5 Optimization results

Pop. size	Algorithm	Average			Best result	
		Objective value ($\times 10^4$ USD/Lot)	σ	%Diff*	Objective value ($\times 10^4$ USD/Lot)	%Diff
200	Standard GA	3.35	0.44	-0.6	2.65	-3.8
	Proposed GA	3.33	0.39		2.55	
400	Standard GA	3.14	0.56	-5.7	2.43	-8.2
	Proposed GA	2.96	0.45		2.23	
600	Standard GA	3.05	0.58	-6.6	2.06	4.4
	Proposed GA	2.85	0.35		2.15	

* %Diff = $100 \times (Obj_{pro} - Obj_{std}) / Obj_{std}$

Regarding the qualitative assessment, the objective values of all 25 trials in each population size are sorted and plotted in Figure 4.9. It can be seen that the proposed approach is more likely to provide a lower objective value in the case of bigger population size (400 and 600) while they perform similarly at a low population size (200).

Furthermore, the quantitative performance of the proposed approach is evaluated via a probabilistic assessment. The probability density functions (PDF) of both approaches are fitted from the results of 25 trials in each population size as shown in Figure 4.10(a), (b), and (c). The achievable probability is calculated and compared in Table 4.6. It implies that the proposed approach provides an opportunity around 70% to achieve an acceptable solution while the standard GA provides only 66%, 55%, and 48% for the population size at 200, 400, and 600, respectively. In other words, the proposed approach is more likely to provide a better result.

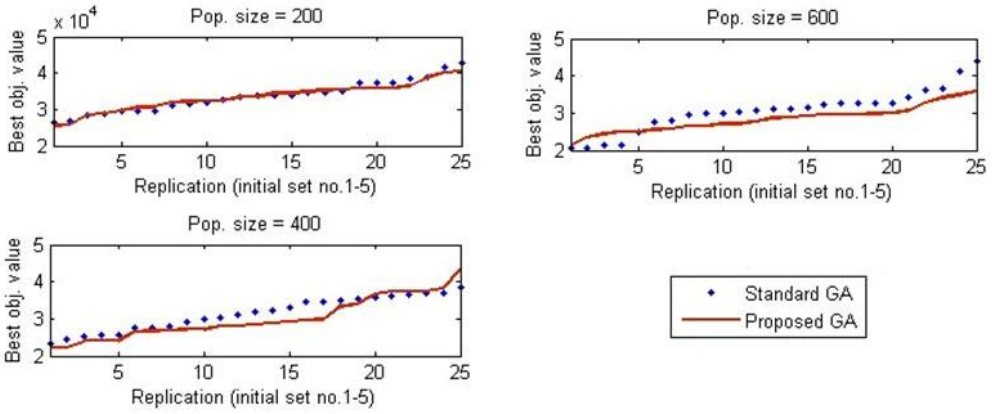
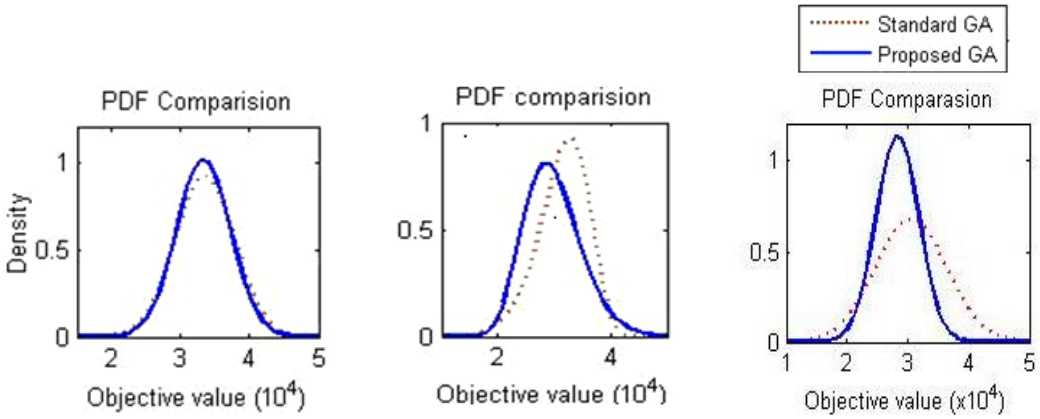


Figure 4.9 Quantitative comparison of all results shown by the population size



(a) Pop. size = 200 (b) Pop. size = 400 (c) Pop. size = 600

Figure 4.10 Plot of average of best objective value in each initial set

Table 4.6 Comparison of probability of the achievement from each method

Pop. size	Proposed GA	Standard GA	$\%Eff = 100 \times (P_{pro}(X) - P_{std}(X)) / P_{std}(X)$
200	$P(X \le 35350) = 0.70$	$P(X \le 35350) = 0.66$	6.06
400	$P(X \le 32510) = 0.70$	$P(X \le 32510) = 0.55$	27.2
600	$P(X \le 30200) = 0.70$	$P(X \le 30200) = 0.48$	45.8

Table 4.7 Average of computational time consumption

Pop. size	Average of computational time (min)	
	Standard GA	Proposed GA
200	86.7	17.6
400	193.3	30.4
600	280.0	42.0

Regarding the computational time, the average time of all cases is shown in Table 4.7. It can be seen that the standard GA needs more computational time than the proposed approach around 5 times. It is expected that a better computational time of the proposed approach is due to the design that better fits the particular scheduling purpose while the standard GA does aim for general purpose.

4.2 Robust Scheduling under Uncertain Disruption for SCC Plant

The proactive scheduling, which utilizes the traditional robust scheduling, often provides an excessively conservative solution. This section proposes a proactive scheduling by using a new robust measure that can satisfy the requirements and practical handling in a steel factory. Due to the discrete and nonlinear characteristics of the problem, this robust measure can be obtained by applying ANN for assessing the uncertainty and it is decomposed the zones or effect of the breakdown attack by using k-mean clustering.

4.2.1 Proactive Scheduling under Uncertain Machine Breakdown

The proposed methodology for coping the uncertain machine breakdown focuses on the proactive approach. This robust schedule will not be rescheduled at the executing time. In a steel production case, the operator can manually decrease the casting speed to keep the continuity of casting during the maintenance time because continuous casting is a must. Therefore, the proposed methodology incorporates a proactive scheduling with Right-shifting strategy (that postpones each remaining operation affected from the breakdown) and the casting speed control at the execution stage as shown in Figure 4.11. The maximum absorption of the disruptions depends on the tundish capacity.

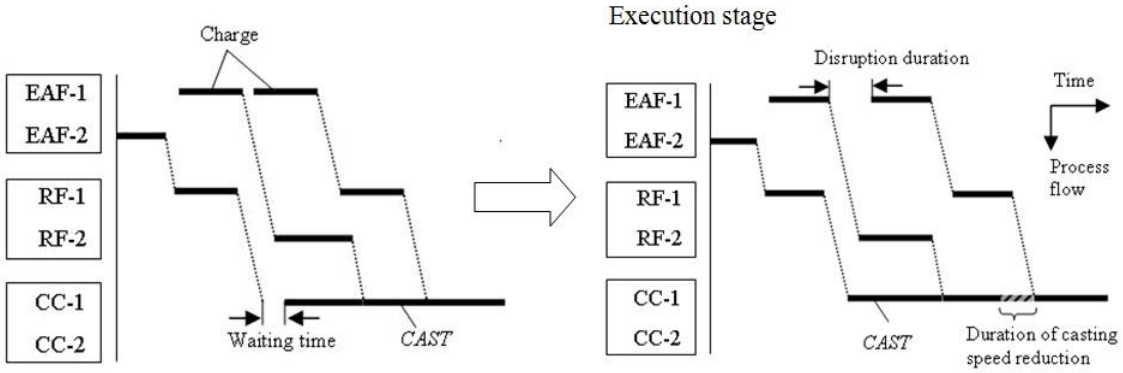


Figure 4.11 Handling disruption with right-shifting and reducing casting speed

4.2.2 Machine breakdown formulation

The disruption from the machine breakdown, which delays the production and does not cause a shutdown, is defined as uncertainty in this paper. For example, an inefficient cooling from some spray nozzles in the caster will force the operator to decrease the casting speed (the production is delayed).

To simulate a machine breakdown, 3 parameters are required (Al-Hinai and ElMekkawy, 2011): (a) the disrupted machine, (b) the breakdown time, and (c) the

breakdown duration. In works of Al-Hinai and ElMekkawy (2011) and Xiong et al. (in press), a disrupted machine is assumed from a machine that put in service more than others. The probability of a machine to fail is approximated by a proportion of the busy time of that machine and the total busy time of all machines. However, the disruption in the steel production should be defined differently. Actual data from a factory points that several breakdowns/failures often happen in early production (see the experiment section). This work defines (a) the disrupted machine (BM) through the following uniform distribution.

$$BM = \text{ceil}[U(\beta_1 M, \beta_2 M)] \tag{4.23}$$

Where $U(\cdot)$ is the uniform distribution function, M is the total machine number, and β_1, β_2 are constant to control the range of the uniform distribution.

For (b) the breakdown time (BT) and (c) the breakdown duration (or delay) (BD), they can be characterized from the real maintenance data in terms of probability density function (PDF).

4.2.3 Distribution-Based Robust Measure

Generally, a typical robust measure, which compromises between mean and variance of the system performance, e.g., min $w_1\mu + w_2\sigma$, is widely used for design under uncertainty (Chen et al., 1999). However, it seems to often provide a more conservative result than desired (a schedule with more slack) in case of high variance (this behavior is confirmed in the experiment section in this work). Several works solved this slack problem via the slack-based methods (Goren and Sabuncuolu, 2008). However, they often fail to schedule a flexible job shop and flexible flow shop production as it is the nature of the problem in this work because of focusing only on the schedule structure while neglect the uncertainty (Xiong et al., 2012).

The steel scheduling problem requires a robust schedule that provides a low mean (than the typical robust approach) while is still flexible enough for permitting some bad scenarios with low frequency. To avoid a slack based method, the robust measure is designed from a probability distribution curve. MCS is adopted to obtain PDF by involving pseudo random numbers. Considering PDF from 3 scenarios in Figure 4.12 that can be represented by the log-normal distribution (e.g., the uncertainty representation in Table 4.14), a suitable robust schedule should be the solid line.

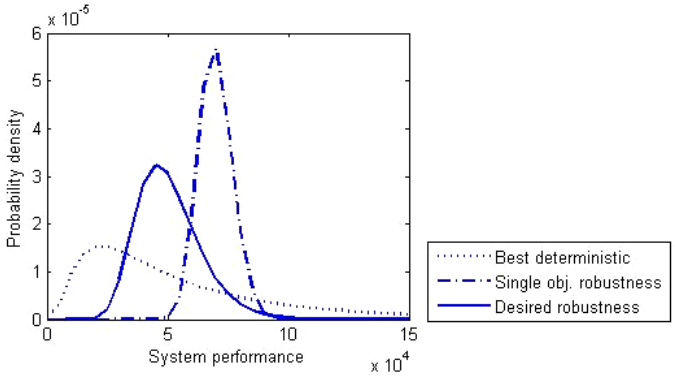


Figure 4.12 Expected PDF from different approach

From this requirement, the robust measure is designed as the real value z_c of the cumulative distribution function (CDF) at $F(z_c) = P(X \le z_c)$, where X is the system performance (random variable). Thus, the problem is formulated as follows:

$$\min\ z_c = \quad F^{-1}(\alpha, \hat{\mu}, \hat{\sigma}) \tag{4.19}$$

where $F^{-1}(\alpha, \hat{\mu}, \hat{\sigma})$ is an inversed CDF and $\alpha \in [0,1]$ is percentile. $\hat{\mu}$ and $\hat{\sigma}$ are parameters of lognormal distribution function, which will be calculated by the following relation:

$$\hat{\mu} = \log(\mu^2 / \sqrt{\sigma^2 + \mu^2}) \tag{4.20}$$

$$\hat{\sigma} = \sqrt{\log(\sigma^2 / \mu^2 + 1)} \tag{4.21}$$

where μ and σ are the mean and standard deviation of the system performance, which are estimated from the ANN response surface model, provided next.

4.2.4 Case Study on Uncertainty Assessment: Peak Function

To investigate the utilization of uncertainty assessment by the ANN model, a well known nonlinear function called "peaks" is raised as a case study. Typically, the peaks function contains 2 design variables; x and y, and has the deterministic optimal solution $(x_{\min}, y_{\min}, z_{\min})$ at around 0.2, -1.6, and -6.3, respectively. In this case study, we suppose that only the decision variable x is perturbed with uncertainty as $x = x + \psi(0, 0.02)$, where $\psi(\cdot)$ is a normal distribution. By the uncertainty assessment via MCS at the minimum point with the number of sample at 10^5, the mean and standard deviation are equal -5.77 and 0.79, respectively. The mean and variance will be improved via the robust design formulated as follows:

$$\begin{aligned} &\text{minimize} \quad z(x,y) = w_1 \mu_f + w_2 \sigma_f^2 \\ &\text{subject to} \quad -3 \le x \le 3 \end{aligned} \tag{4.22}$$

$$-3 \le y \le 3$$

4.2.4.1 *Design of ANN architecture*

The design of ANN architecture is follows:

- *Step 1*: The initial sample set of the relation between design variables and the output variables is made.
- *Step 2*: Constructing and tuning a topology of the ANN model.
- *Step 3*: Retuning the ANN model via a lager initial sample set.

Step 1: Latin Hypercube Design

Creation of the surrogate models needs some samples in order to construct the ANN model to fit the estimation of μ_f and σ_f. These samples could be produced by the design of experiment (DOE) such as *Latin hypercube design* (LHD) (Wei and Yuying, 2008). With LHD, the design space for each variable is uniformly divided into *P* equal levels and each level contains only one point (experiment). In our problem, the LHD creates a sample set of the input variable x, y, and the output variable $\mu_f(x, y)$ and $\sigma_f(x, y)$. Firstly, the set of variable x and y is created by the LHD and then $\mu_f(x, y)$ and $\sigma_f(x, y)$ are achieved by performing Monte Carlo simulation (MCS). In this study, 100 samples (*P*) of LHD and 10^5 replications for each MCS are used.

Step 2: Design of ANN architecture

Typically, there are a number of ANN paradigms. A multi layer feed-forward backpropagation network, which is one of well-known and mostly used paradigms, is used in this study. This subsection constructs 2 ANN models; $\mu_f = h_\mu(x, y)$ and $\sigma_f = h_\sigma(x, y)$, based on the sample sets in Step 1. To optimize the model, 4 main parameters, i.e., 1) the transfer functions, 2) the number of hidden layers, 3) the number of neurons, and 4) training algorithm, must be investigated. The ANN toolbox of MATLAB is utilized in this study.

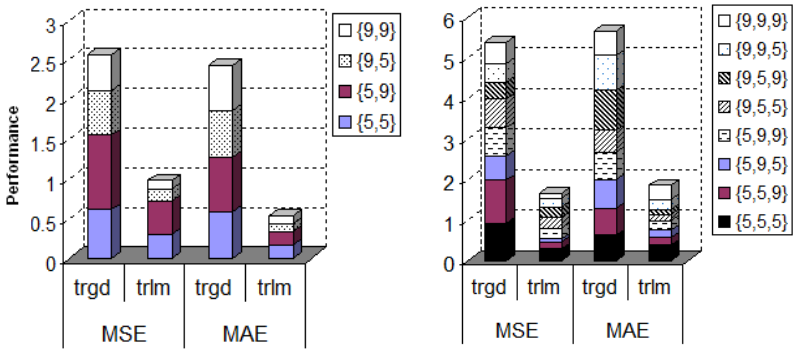
- *Step 2.1*: Only the transfer function is trialed while other parameters are fixed with the standard setting. The basic criterion as the minimum square error (MSE) is chosen to judge the performance of each trial transfer functions during training. In addition, the standard training algorithm (the gradient descent) is adopted. These settings are applied with a basic topology of ANN (one hidden layer and each layer contains 2 neurons). The trial cases are shown in Table 4.8. The results address the ANN structure using the tangent sigmoid transfer function ("tansig") for the hidden layer and the natural transfer function ("purelin") for the output layer is suitable.

Table 4.8 Trial and error of transfer function for ANN

Case	Transfer function		Training perf. (MSE)	
	Hidden layer	Output layer	h_μ	h_σ
1	tansig	tansig	1.54	0.41
	tansig	logsig	1.01	0.36
	tansig	purelin	0.90	0.27
2	logsig	tansig	2.29	0.52
	logsig	logsig	2.37	0.47
	logsig	purelin	1.19	0.40
3	purelin	tansig	1.01	0.29
	purelin	logsig	1.41	0.36
	purelin	purelin	1.83	0.43

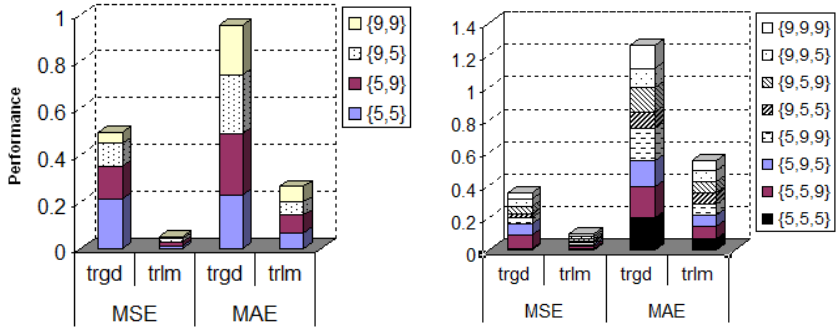
- *Step 2.2*: The number of the hidden layers and neurons, and the training algorithm are trialed, simultaneously. The transfer function in Step 2.1 is inherited into this step. In this stage, the models are separately trained using the MSE criterion and the minimum absolute error (MAE) criterion. In addition, the hidden layers are varied from 2 to 3 layers and the neurons are varied from 5 to 9 nodes in each layer while the training algorithm is tested between the gradient descent (trgd) and Levenberg-Marguardt (trlm) algorithms. The results of training the model h_μ and h_σ are shown in Figure 4.13 and 4.14, respectively. It is noticed that the Levenberg-Marguardt algorithm provides a much better result for all structures and criterions (MSE and MAE). It seems that training with the MAE can provide a smaller structure for both h_μ and h_σ models (2 hidden layers).Other optimal parameters are summarized in Table 4.9.

With the structure in Table 4.9, the developed models are tested against the test sample sets. The results are shown in Figure 4.15. It is noted that the model can properly estimate the mean of the peaks function, which is nonlinear. However, due to being highly nonlinear in the variance, the model h_σ cannot provide a good estimation of σ_f against the test sample sets in cases of both MSE and MAE (even though it can be fit against the initial training sample sets). The models will be improved by re-preparing the initial training sets in the next step.



(a) 2 hidden layers (b) 3 hidden layers

Figure 4.13 Results of varying no. of neurons and layers of the model h_{μ}



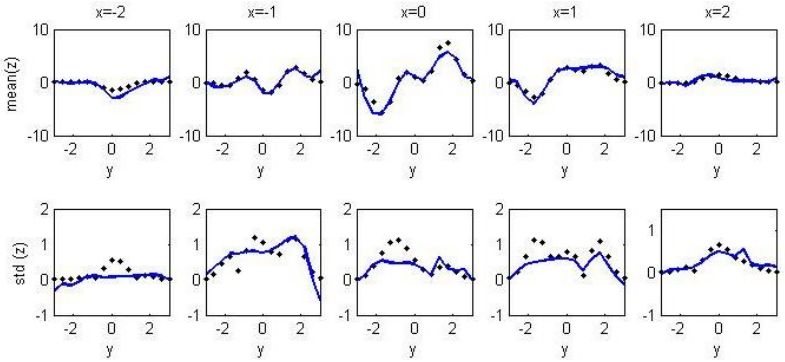
(a) 2 hidden layers (b) 3 hidden layers

Figure 4.14 Results of varying no. of neurons and layers of the model h_{σ}

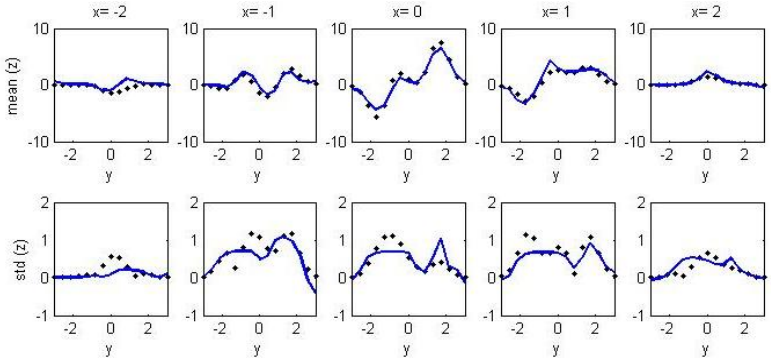
Table 4.9 Summary of trial and error on ANN model

Model	Parameter	Training performance	
		MSE	MAE
h_{μ}	Hidden layer	3	2
	Neuron/layer	{5,9,5}	{9,9}
	Transfer func.	Tansig/Purelin*	Tansig/Purelin
	Training alg.	trlm	trlm
h_{σ}	Hidden layer	3	2
	Node per layer	{9,5,5}	{9,5}
	Transfer func.	Tansig/Purelin	Tansig /Purelin
	Training alg.	trlm	trlm

** Hidden layer/Output layer*



(a) Case of MSE



(b) Case of MAE

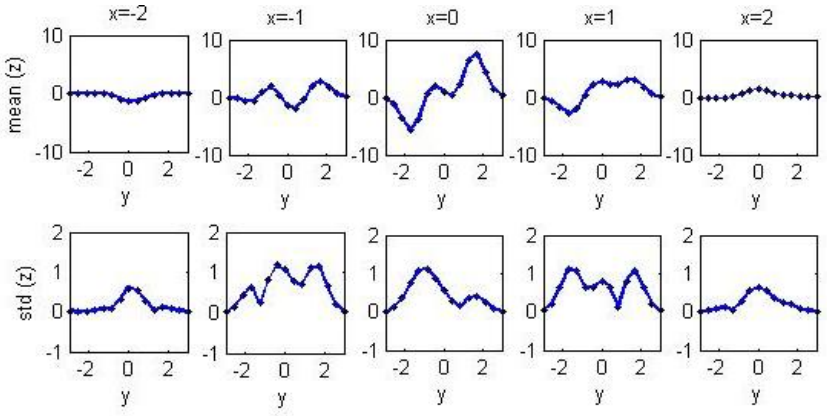
Figure 4.15 Validation of ANN models (h_μ and h_σ)

Step 3: Retuning ANN Architecture

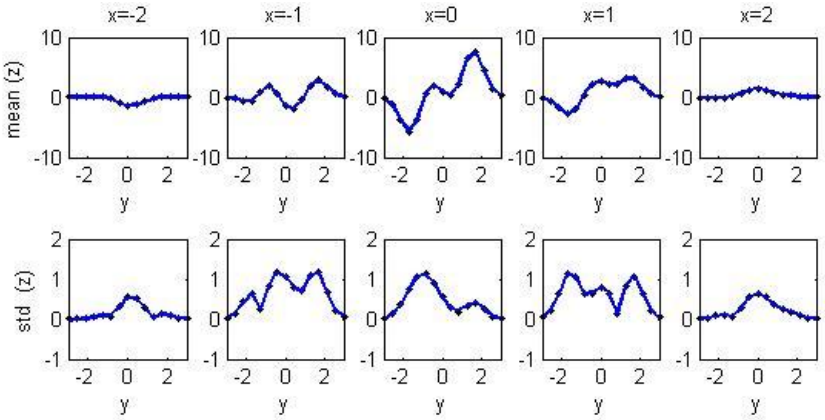
Since the number of the initial samples for the model training and its distribution can affect the accuracy of the model, the step of the Latin hypercube sampling is improved. The number of replication of MCS is increased to 10^6 and then the number of samples is increased to 1000, respectively. The results are shown in Table 4.10. It indicates that the increase of the number of MCS did not significantly affect the performance but the increase of the number of samples can significantly improve the performance as seen in Figure 4.16.

Table 4.10 Results of retuning ANN models

No. of MCS & Sample	Perf. of h_μ		Perf. of h_σ	
	MSE	MAE	MSE	MAE
$\{10^5, 100\}$	0.083	0.098	0.008	0.055
$\{10^6, 100\}$	0.069	0.112	0.008	0.050
$\{10^6, 1000\}$	$1.02*10^{-5}$	$3.43*10^{-4}$	$8.0*10^{-4}$	0.0061



(a) Case of MSE



(b) Case of MAE

Figure 4.16 Validation after retuning ANN models

4.2.4.2 *Demonstration of ANN Model*

In this section, the developed ANN model is utilized to optimize the problem in Eq. (4.22) with the procedure in Figure 2.15. The ANN models based on MSE and MAE criteria will be separately investigated. These developed models will be also demonstrated with the gradient based method. Then, the performance of these meta-model based methods will be compared with the best worse case optimization. The parameters of GA optimization are shown in Table 4.11. The gradient based optimization is performed by MATLAB standard toolbox, called "fmincon", which is a nonlinear programming. The same objective function as Eq. (4.22) will be adopted in this optimization.

Table 4.11 GA optimization parameters of peak function

Parameters	Value	Parameters	Value
Representation	Binary	Crossover	2 points (rate 0.8)
Pop. size	x =300, y =300	Selection	Stochastic uniform
Max. iteration	200	Mutation	Random (rate 0.01)

For the best worst case method, the problem is to find the best solution on the worst-case scenario. It can be generally formulated in Eq. (4.23) as follows: Let X is the set of all solutions and S is the set of all possible scenarios. $z_W(\mathbf{x},\mathbf{s})$ is an objective function of solution $\mathbf{x} \in X$ in scenario $\mathbf{s} \in S$.

$$\min_{\mathbf{x}\in X} \max_{\mathbf{s}\in S} \; z_W(\mathbf{x},\mathbf{s}) \tag{4.23}$$

In this experiment, X is the set of the feasible solution x and y according to Eq. (4.22) while S is the set of the uncertainty by $\psi(0,0.02)$ that is generated by MCS.

The results of the GA and gradient based optimizations under the developed ANN models are shown in Table 4.12. Both methods provide similar solution for all mean and variance weights and there is no different result between MSE and MAE models. Clearly, the optimal robust solution (in case of $w_1 = 1$ and $w_2 = 20$), which consists of an acceptable optimality with a low variance, can be obtained under aiding of the developed ANN models. For result from the best worst case method, it is shown in Table VI and the comparison of the GA optimization based on the ANN models and the best worse case method by MCS is shown in Figure 4.17.

Table 4.12 Results of GA and Gradient based optimizations

Case	w_1 / w_2	$(x_{\min}, y_{\min})$	(μ, σ)	z
MSE	1 : 1	0.22, -1.64	-6.19, 0.47	-5.97
	1 : 10	0.20, -1.74	-6.04, 0.44	-4.07
	1 : 20	0.17, -1.98	-4.81, 0.34	-2.50
MAE	1 : 1	0.22, -1.64	-6.19, 0.47	-5.97
	1 : 10	0.20, -1.76	-6.00, 0.44	-4.05
	1 : 20	0.17, -2.00	-4.71, 0.33	-2.50

Table 4.13 Result of best worst case optimization

$(x_{\min}, y_{\min})$	(μ, σ)	z_W
(0.24, -1.75)	-6.04, 0.44	-2.28

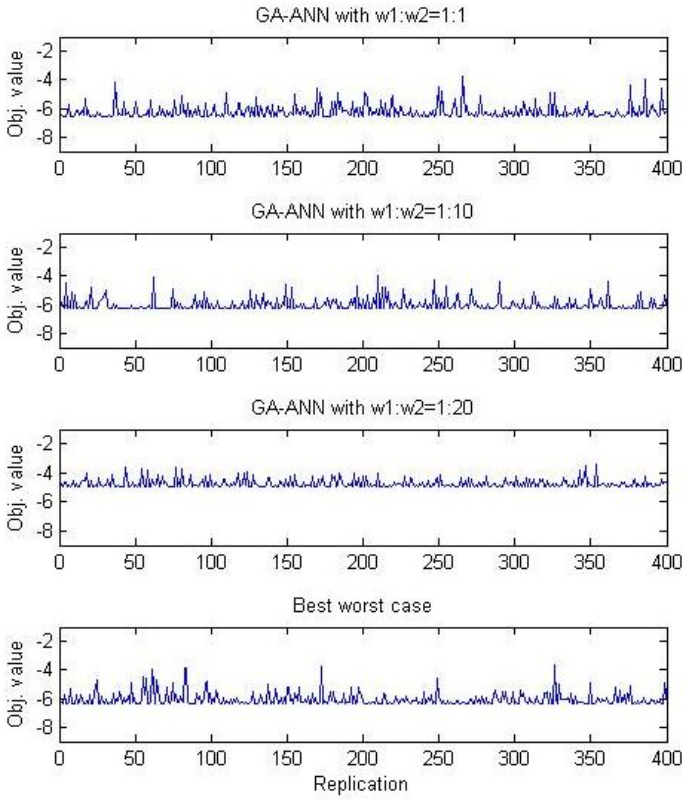


Figure 4.17 MCS of the results of each optimization method

In the optimality viewpoint, Table 4.12 and 4.13 and Figure 4.17 address that the robust design approach (GA-ANN) with low variance weight (w_2 =1 and 10) can provide the lowest objective value and its mean comparing with the best worst case. When a large MCS (10^6 replications) is performed, the maximum objective value of the robust design approach is at -1.143 while the best worst case method provides -2.278. Even though the best worst case can guarantee the lowest maximum objective value at the worst scenario, it seems to be unnecessary. Because the worst scenario rarely occurs (it is noticed from the variance in Table 4.13 that is around 0.44 while the mean can be kept around -6).

Considering the robustness viewpoint, although the best worse case method can provide an acceptable variance (close to the result of the robust design with the low variance weights), the robust design with the high variance weight can provide a much lower variance. The standard deviation can be improved around 20% from the case of lower variance weights. However, the compromise of the mean and variance of the GA-ANN optimization may not be good enough for some serious cases. In other words, $\mu \approx$[-5, -6] with σ > 0.4 should exist.

4.2.5 Uncertainty Assessment with Decomposed ANN

Based on the optimization model in Section 4.1.1, inputs of the ANN model are the design variable $\mathbf{X}$, $\mathbf{Y}$, and st while the outputs are μ and σ of the system performance as illustrated in Figure 4.18. Generally, 2 ANN models; $\mu = g_{\mu}(\mathbf{X}, \mathbf{Y}, st)$ and $\sigma = g_{\sigma}(\mathbf{X}, \mathbf{Y}, st)$, are of interest. Because of the complexity from 3 levels of the uncertain variables (BM, BT, BD), the discrete decision variables and nonlinear behavior, only 2 ANN models fails to estimate μ and σ (see the experiment in the next section). In order to simplify the system, the variable BM is treated by decomposing the ANN model to each machine. Individual ANN models; $\mu^m = g_{\mu}^m(\mathbf{X}, \mathbf{Y}, st)$ and $\sigma^m = g_{\sigma}^m(\mathbf{X}, \mathbf{Y}, st)$, respond to the disruption that attacks to the machine m. However, it produces a lot of the decomposed ANN models in case of a large steel production (as Figure 2.5). To decompose the ANN model, *k*-mean clustering, which is suggested for the flexible flow shop scheduling problem (Choi and Wang, 2012), is adopted.

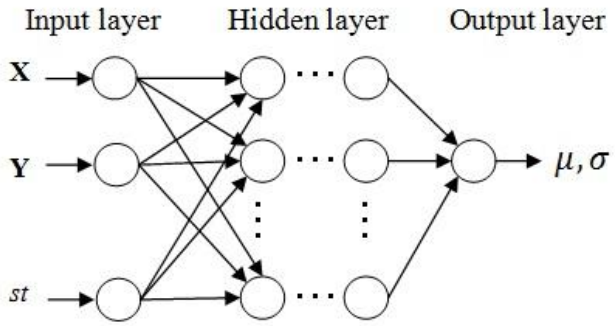


Figure 4.18 Multilayer feed-forward back propagation ANN

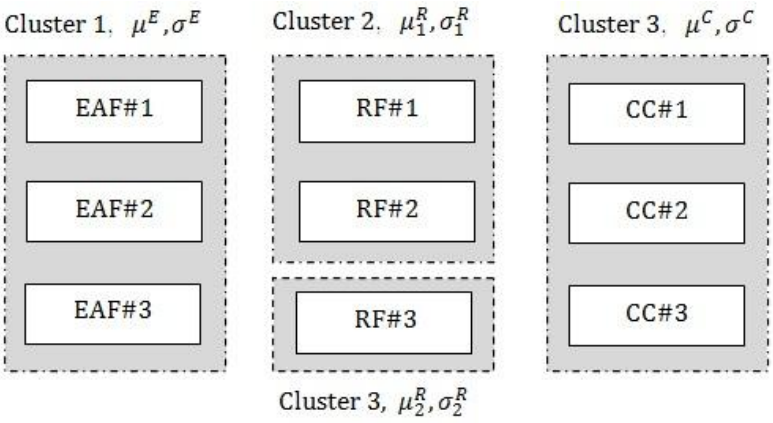


Figure 4.19 Example of clustering for the steel production

The main idea of k-mean clustering is to partition *n* observations (objects) into *k* clusters in which each observation belongs to the cluster with the nearest mean,

serving as a prototype of the cluster. The result will be iteratively improved until no object moves cluster. To apply this method, the machines with the similar effect of the uncertainty will be grouped to utilize the same ANN model. For instance, three parallel lines of a steel production are clustered as Figure 4.19. The ANN models $\mu^E = g_\mu^E(\cdot)$ and $\sigma^E = g_\sigma^E(\cdot)$ are utilized if and only if the breakdown attacks at any one of EAFs.

To use k-mean clustering, a characteristic representation of the objects has to be defined. In the scheduling problem dealing with the uncertain breakdown, different individual schedule can be affected differently by same disruption statistics (different slack in the schedule differently absorbs the effect of uncertainty). Thus, the uncertainty characteristics (UC_m) for each machine is calculated from the statistics of the large number of the schedules as follows:

$$UC_m = \frac{E[\sigma^m]}{E[\mu^m]} \tag{4.23}$$

Based on UC_m, the difference of the uncertainty between the machines can be measured in terms of a distance. The Euclidean distance is the one of the most commonly used methods to measure a pair (Choi and Wang, 2012). It is applied as follows:

$$D_{m,n} = \|UC_m - UC_n\| = \sqrt{(UC_m - UC_n)^2} \tag{4.24}$$

where UC_m and UC_n are a pair of machines.

There are 2 problems for using k-mean clustering; 1) how to choose a suitable cluster number (*k* number) and 2) how to validate the resulting clusters. For the first problem, the trial and error method may be applied for a small cluster number and iterative algorithms (e.g., GA) are applied for a large cluster number (choosing a suitable *k* number of both methods resides on comparing the cluster validity indices, which are assigned in the second problem). For the second problem, to know how well-separated the resulting clusters are, a silhouette plot can be utilized by using the distance measures (such as Euclidean) from k-means (Rousseeuw, 1987). The silhouette value is between -1 and 1. The value that is close to 1 is suggested.

4.2.6 GA and Search Procedure

With GA, the procedure of the proposed methodology is given in Figure 4.20. The details of GA formulation and its operations for SCC scheduling are given as follows:

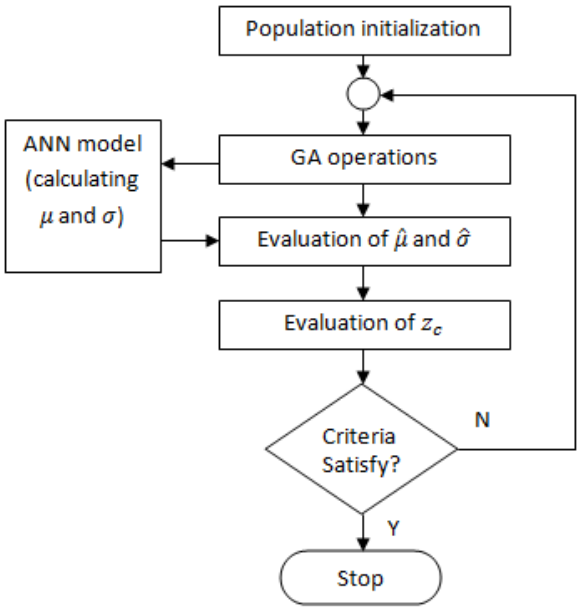


Figure 4.20 Procedure of the proposed methodology

This procedure starts from the initialization of the objective function variables $\mathbf{X}^{(k)}$, $\mathbf{Y}^{(m)}$, and $st_i^{(m)}$, which are formulated into a chromosome form. The chromosome contains 3 representations; 1) binary string that contains $\mathbf{X}^{(k)}$ and $\mathbf{Y}^{(m)}$ and 2) real number string that contains the starting time $st_i^{(m)}$. Then, the main GA operations such as the crossover and mutation are performed separately for each chromosome. The standard one or multiple-point crossovers can be adopted. For mutation, a bit mutation is applied to the binary string while the random mutation (defined in Man et al., 1999) is applied to the real number string. After GA operation, the statistical values of the system performance in each chromosome will be calculated by ANN model and then the robust measure is evaluated. The iteration will be executed until the maximum iteration is reached or a convergence criterion about successive iteration is reached.

4.2.7 Experiment and Results

The experiment is performed in 3 steps; (1) investigating the effect of uncertainty, (2) constructing ANN models, and (3) demonstrating the methodology. The first step is to formulate the uncertainty and assess its effect against the best deterministic schedules. The performance of the deterministic schedules will be used as an upper bound and lower bound for assessing the methodology performance later. The optimization parameters are summarized in Table 4.14. In the table, the parameters of GA are defined by trial and error based on suggestion from Grefenstette et al. (1986) and Man et al. (1999) while the parameters of the system performance, the robust measure, and the processing time of SCC production are defined from a real factory situation. The system performance is defined in terms of the production costs in USD. The percentile α (at 0.8) is set via experience of operators.

Table 4.14 Optimization model parameters for GA-decomposed ANN

Optimization model		GA parameters	
Parameters	Value	Parameters	Value
f_1, f_2, f_3, f_4	{600, 14400, 0.07, 1.6}	Representation	Binary, Real
C_W, C_B, C_L, C_E	{6.7, 14400, 16.2, 5.4}	Population size	400
N, P, L	{12, 2, 4}	Max. iteration	200
α	0.80	Crossover	2 point (rate 0.8)
β_1, β_2	0,1	Selection	Stochastic unif.
		Mutation	Bit (rate 0.01) for binary Ramdom (rate 0.01) for real

4.2.7.1 *Uncertainty Assessment*

Three breakdown characteristics; BM, BT, and BD, are formulated from the maintenance data of the period of 15 months that approximately contain 150 samples (for all processes). BM is defined as Eq. 4.23 that returns the integer value in between 1 to 6 (represented 6 machines) while BT and BD are fitted in terms of Gamma and Log-normal distribution functions as illustrated Figure 4.21 and summarized in . Since the raw data is investigated per a production lot containing 12-14 charges, the breakdown time is represented in the charge number unit. These breakdown characteristics are used for both steel production lines.

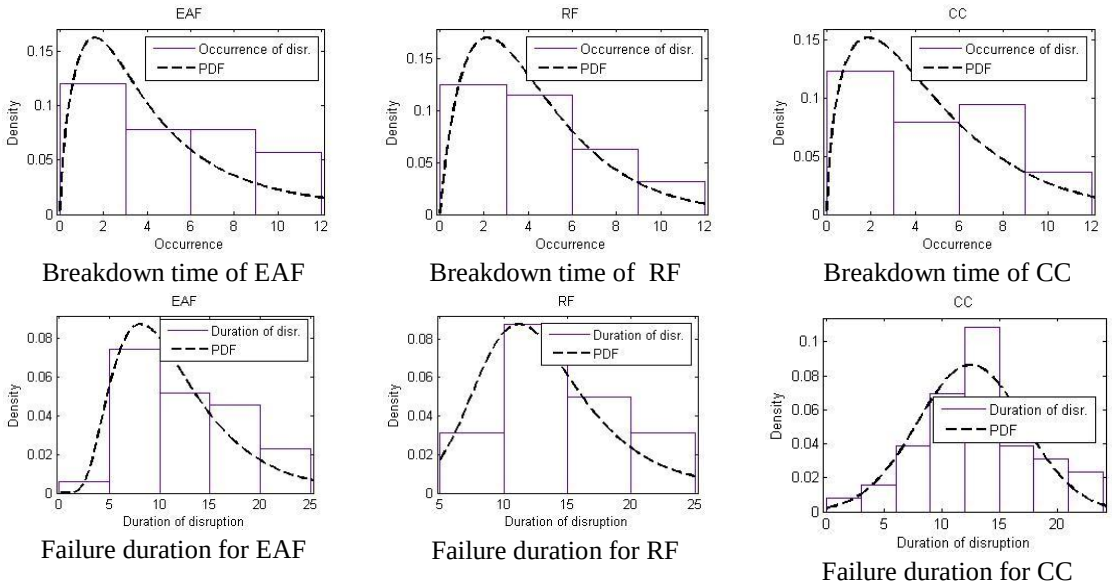


Figure 4.21 Fitted PDF of machine breakdown from a steel factory

Table 4.15 Summary of disruption PDF

Process	Breakdown Parameter	Range	Expression
Melting	*BT*	1-12 (charge no.)	Gamma(1.37,3.55)
	BD	5-25 (min)	Logn (2.39, 0.50)
Refining	*BT*	1-12 (charge no.)	Gamma(1.98,2.17)
	BD	5-25 (min)	Logn (2.50, 0.42)
Casting	*BT*	1-12 (charge no.)	Gamma(1.64,2.89)
	BD	4.5-25 (min)	Norm (12.50, 4.61)

In order to investigate the effect of the uncertainty, the disruption in Table 4.15 is tested with a best deterministic schedule. MCS is shown in Figure 4.22(a). The best deterministic schedule provides a high cost when the disruption attacks, especially, in case of the cast break event. This behavior causes its mean (μ) and standard deviation (σ) from MCS at 10,000 replications to be higher when comparing with the minimum value as shown in Table 4.16.

Table 4.16 Statistic results of MCS with the best deterministic schedule

Statistic	Value
Min. value ($)	2.1504×10^4
μ ($)	4.9471×10^4
σ ($)	4.1105×10^4

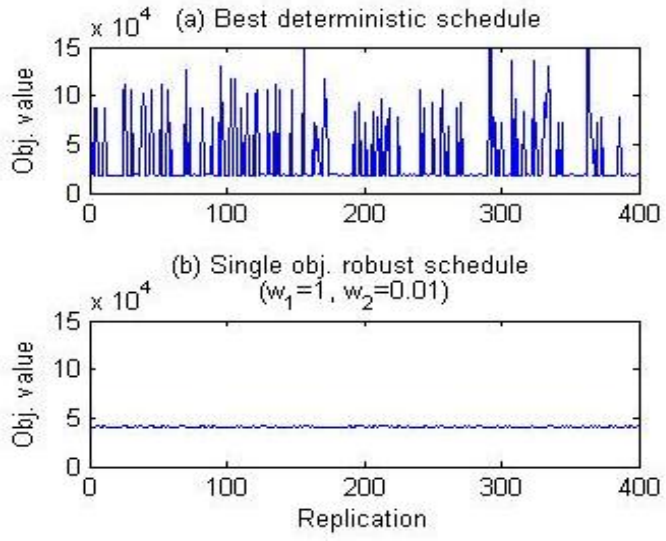


Figure 4.22 (a) MCS of the best deterministic schedule and (b) robust schedule with single objective robust measure (z_s)

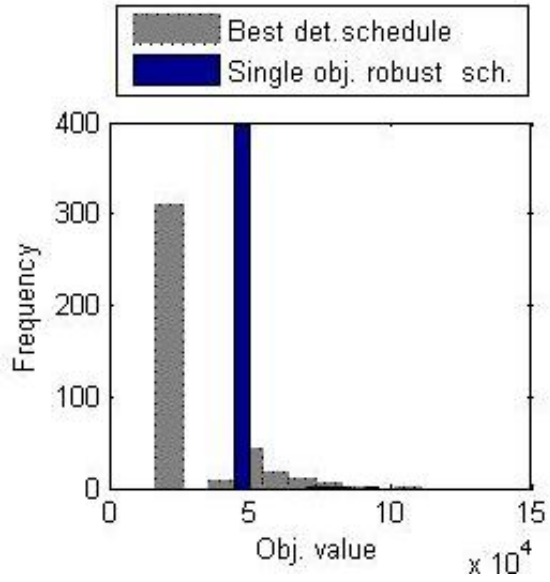


Figure 4.23 Histograms of MCS in each case

In another case, a typical robust schedule is also tested under the same uncertainty. The mock up case of a single objective robust measure $z_s = w_1\mu + w_2\sigma$, where w_1 and w_2 are the weighting factors, is investigated. The result shows that even the variance is weighted with a very small value, this robust measure provides a very high mean with a very low variance as shown in Figure 4.22(b). It becomes an impractical schedule that has too much slack according to the histograms in Figure 4.23. Obviously, it is not flexible enough to use this robust measure. From Figure 4.23, the proposed methodology is designed to compromise a result between the deterministic schedule and traditional robust measure case.

4.2.7.2 *Design of the decomposed ANN model*

Based on the procedure in section 4.2.4, the design steps of the decomposed ANN architecture are given as follows:

- Step 1: Creating an initial sample set of the relation between design variables and the output variables.
- Step 2: Clustering the machines by *k*-mean clustering and validating the resulting clusters.
- Step 3: Constructing and tuning a topology of the ANN models.
- Step 4: Validating the decomposed ANN models.

In the first step, the set of input variables ($\mathbf{X}$, $\mathbf{Y}$, st) is firstly created by the LHD and then the set of output variables (μ and σ) are achieved by performing MCS. The binary variables $\mathbf{X}$ and $\mathbf{Y}$ are coded into real number before applying LHD. In this study, 100 samples (P) of LHD and 10^5 replications for each MCS are initially used. In the second step, the samples of the schedule are tested for each breakdown and then μ^m

and σ^m are evaluated for each machine as shown in Table 4.17. The *k*-mean clustering is performed by MATLAB toolbox. To optimize the number of clusters, 2 and 3 clusters are trialed. The results are shown in Table 4.18.

Table 4.17 Calculation of $\boldsymbol{UC_m}$

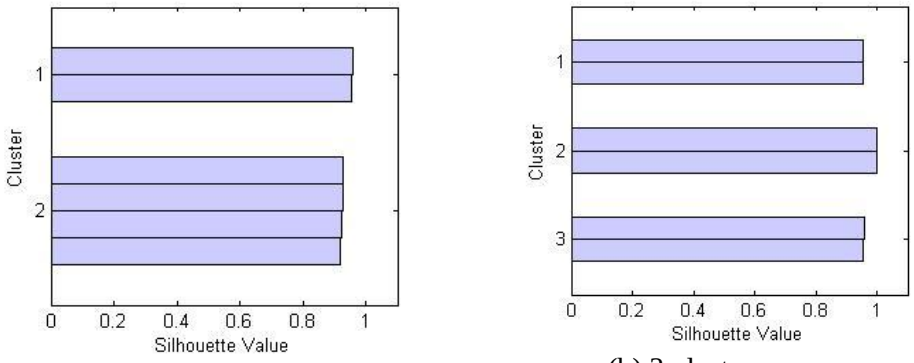
Breakdown point	$E[\mu^m]$	$E[\sigma^m]$	UC_m
EAF#1	1.1142E6	0.0267E6	0.0240
EAF#2	1.1153E6	0.0263E6	0.0236
RF#1	1.1706E6	0.1191E6	0.1017
RF#2	1.1642E6	0.1143E6	0.0982
CC#1	1.1030E6	0.0162E6	0.0147
CC#2	1.1027E6	0.0162E6	0.0147

Table 4.18 Resulting clusters

Case	Cluster	Centroid
2 clusters	{EAF#1, EAF#2, CC#1, CC#2}	0.0192
	{RF#1, RF#2}	0.0999
3 clusters	{EAF#1, EAF#2}	0.0238
	{RF#1, RF#2}	0.0999
	{CC#1, CC#2}	0.0147

To validate the resulting clusters, the silhouette plot is made as illustrated in Figure 4.24. From the figures, the silhouette value of both figures is acceptable (close to 1). However, the clustering with 3 clusters is fitter than 2 clusters because of a higher silhouette value. On the other hand, it means the clustering with 3 clusters provides a longer distance between the neighboring clusters (outer distance) while provides a shorter distance between the cluster members (inner distance) comparing with 2 cluster case. If a greater number of clusters (such as 4 or 5) has been used, it is expected to perform better (than 3 clusters). However, due to the acceptable performance of 3 clusters already observed, there is no need to use a greater number. Based on 3 clusters, the decomposed ANN models will be constructed as follows:

- Models when breakdown attacks at EAF#1 or EAF#2: $\mu^E = g_\mu^E(\mathbf{X}, \mathbf{Y}, st)$ and $\sigma^E = g_\sigma^E(\mathbf{X}, \mathbf{Y}, st)$.
- Models when breakdown attacks at RF#1 or RF#2: $\mu^R = g_\mu^R(\mathbf{X}, \mathbf{Y}, st)$ and $\sigma^R = g_\sigma^R(\mathbf{X}, \mathbf{Y}, st)$.
- Models when breakdown attacks at CC#1 or CC#2: $\mu^C = g_\mu^C(\mathbf{X}, \mathbf{Y}, st)$ and $\sigma^C = g_\sigma^R(\mathbf{X}, \mathbf{Y}, st)$.



(a) 2-cluster case (b) 3-cluster case

Figure 4.24 Silhouette plot

In the third step, 4 main parameters of the ANN architecture, i.e., the transfer functions, the number of hidden layers, the number of neurons, and training algorithm, must be trialed. The ANN toolbox of MATLAB is utilized in this study. The summary of trial and error is shown in Table 4.19. Due to the input and output variables should be appropriately scaled before training (Demuth and Beale, 1999), this paper scales the input and output variables into the range 5 and 3, respectively (tuning by trial and error).

Table 4.19 Summary of trial and error on decomposed ANN models

Topology	Setting
Hidden layer	2
Neuron in each layer	{300,100}
Transfer function in each layer	logsig/logsig/purelin
Training algorithm	Trainrp

Finally, the validation of the decomposed ANN models is shown in Figure 4.25. The MAE indicates that the estimation by the decomposed ANN models provides a better accuracy than the model without decomposition ($\mu = g_{\mu}(\mathbf{X}, \mathbf{Y}, st)$ and $\sigma = g_{\sigma}(\mathbf{X}, \mathbf{Y}, st)$) in Figure 4.26.

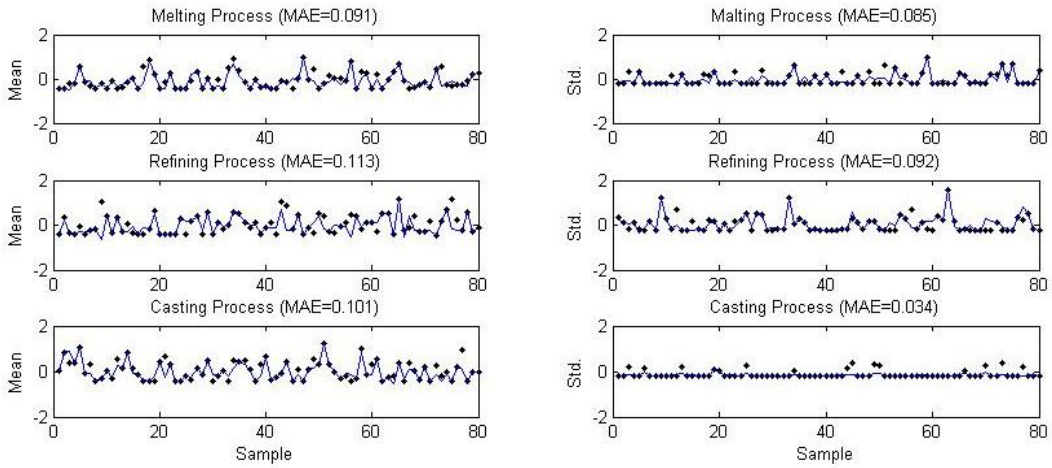


Figure 4.25 Results of the decomposed models

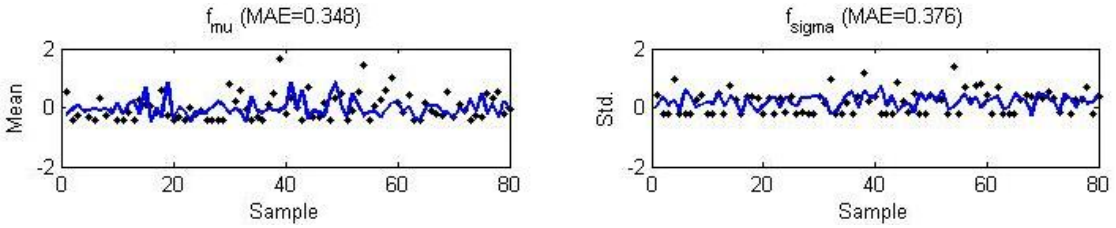


Figure 4.26 Results of the model without decomposition (g_μ and g_σ)

4.2.7.3 *Methodology Performance*

The performance of the proposed methodology will be compared with both cases in Figure 4.27. The result in this figure shows that the proposed methodology can provide a lower mean of the system performance than the single objective robust measure while the frequency of the worst scenario occurrence is acceptable (less than 5% of all scenarios). The result is confirmed by the histogram in Figure 4.28 that the PDF curve of the proposed robust approach is consistent with the desired curve in Figure 4.29.

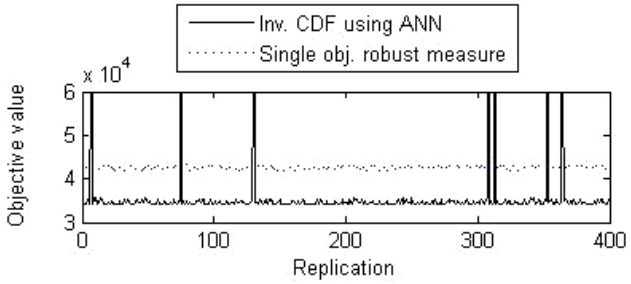


Figure 4.27 MCS of optimal schedule from proposed approach and single objective robust approach

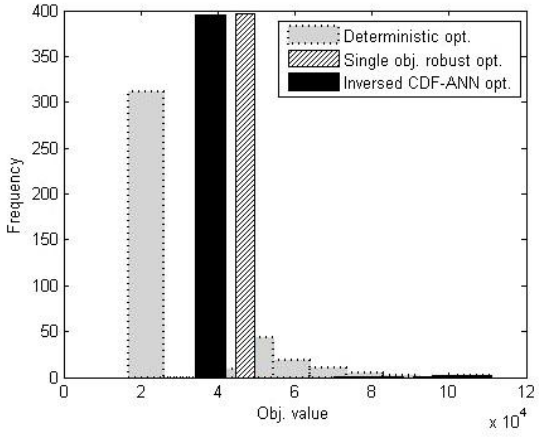


Figure 4.28 Comparison between the inversed CDF based ANN and other cases

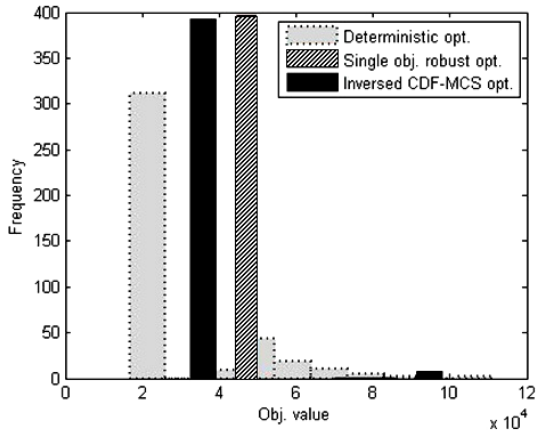


Figure 4.29 Comparison between the inversed CDF based MCS and other cases

The accuracy of the methodology is also demonstrated by replacing the decomposed ANN model via MCS with 10,000 replications. The result in Figure 4.29 indicates that the optimal schedule from the proposed methodology is acceptable and accurate when ANN is compared with MCS.

Table 4.20 Average of computational time consumption

Approach	Average of computational time (min)
Inversed CDF using ANN	22.1
Inversed CDF using MCS	212.3

Regarding the computational time, the inversed CDF robust measure using ANN model is compared with utilizing MCS. The average time of each case is shown in Table 4.20. The approach based on MCS needs more computational time than the proposed approach based on ANN model at about 10 times. It indicates that the proposed approach is more likely to provide a practical schedule coping when the disruption happens in a real situation.

CHAPTER 5 PLAN EXECUTION: OPTIMUM SPRAY COOLING IN CONTINUOUS SLAB CASTING PROCESS

In plan execution, the robust or optimal schedule from the previous chapter is effectively carried on as shown in Figure 5.1, which is a case study of a spray cooling optimization in CC plant under a real situation. This case study shows the potentials of MES with the proposed framework. The system can streamline the CC process from designing an effective schedule under the uncertain disruption to execution of the process.

In Figure 5.1 (from a steel factory), the solidification model is used to optimize the spray cooling (secondary cooling) in purpose of the quality improvement. The process flow based on the figure is as follows:

1. The interface module (PROD) receives the robust schedule from Level 3 (in Fig. 3). In each charge (job), some schedule data (e.g., mold size) are directly sent to set the solidification model while another data (e.g., steel grade) are used to select the set of the related model parameters (e.g. casting speed) from the database (the process knowledge table).
2. The solidification model calculates the slab surface temperature and sends the result to the quality management module (QMGT). This module optimizes the optimal spray cooling patterns based on GA and sent to the actual set-point interface module (ASET) to set the regulators/controllers in the automation (Level 1).

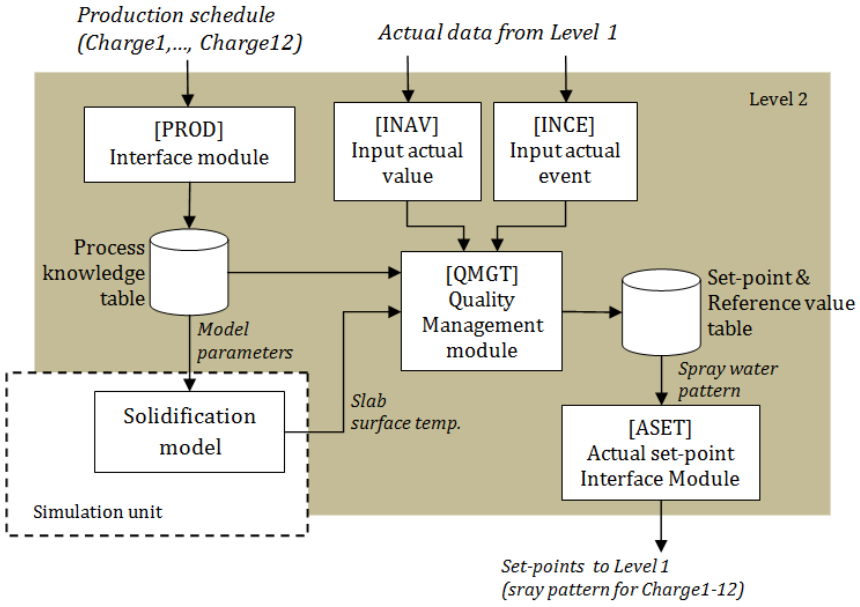


Figure 5.1 Process of the quality management in cooling control of a steel factory

5.1 Simulation Modeling for Execution Phase

To find an optimal cooling rate and pattern of the mold or secondary cooling system, a solidification model based on a finite difference technique and objective function, which satisfies the metallurgical criteria and the resource consumption, is developed here.

5.1.1 Primary and Secondary Cooling in CC

The continuous casting process starts from opening the ladle to pour the molten steel into the tundish that is used as the buffer. Then, the liquid steel is poured through the submersed entry nozzle (SEN) into the mold. The liquid steel level in the mold is regulated by an according control between the flow rate and the casting speed. At the mold, the liquid steel is continuously solidified by the primary cooling (mold water cooling); heat transfer from the convection at the mold wall and the conduction in the steel. The solidified shell will be pulled out from the mold with the optimal speed and is secondly cooled via the spay cooling. After that, the surface temperature of the solid steel (strand) will be naturally decreased in the air (radiation cooling and free convection) before cutting out into the slab.

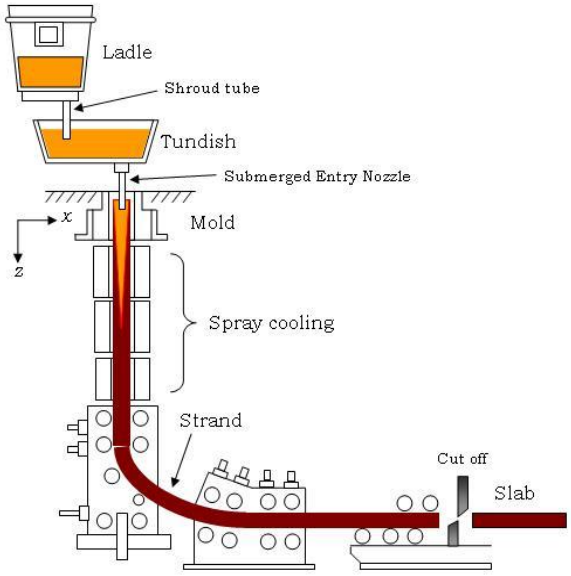


Figure 5.2 Cooling zones in CC

5.1.2 Solidification Model

In order to derive the solidification model of the continuous slab casting process, the governing equation of heat conduction in an unsteady state is shown in Eq. (5.1), which is given for heat flux under the assumption as follows: 1) the thermal conductivity is consistent along heat flux and with internal heat generation, 2) the heat flux is unidirectional from the center to the surface and can be considered negligible along the vertical direction.

$$\rho c \frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2}\right) + \dot{q} \tag{5.1}$$

where T is temperature (K); t is time (s); ρ is the material density (kg/m^3); c is the specific heat (J/kgK); k is the thermal conductivity (W/mK); x is a Cartesian coordinate (m), and $\dot{q}$ represents the heat flux in term of the latent heat source (W/m^3). Then, Eq. (5.1) is approximated by finite difference scheme as follows:

$$\rho c \frac{T_i^{n+1} - T_i^n}{\Delta t} = k \frac{(T_{i+1}^n - 2T_i^n + T_{i-1}^n)}{\Delta x} + \dot{q} \tag{5.2}$$

where T_i^n is the temperature of point i at time n. The latent heat is generated during phase change between liquid and solid, which can be described by $\dot{q} = \rho L(\partial f_s / \partial t)$, where L is the latent heat of fusion (J/kg) and f_s is the solid friction. By taking the chain rule, $\dot{q}$ can be rewritten as

$$\dot{q} = \rho L \frac{\partial f_S}{\partial T} \frac{\partial T}{\partial t} \tag{5.3}$$

By replacing $\dot{q}$ in Eq. (22), the specific heat can be written as $c' = c - L(\partial f / \partial T)$, which implies that the latent heat is compensated by the specific heat. For simplicity, Eq. (5.2) is formulated by using thermal resistance (Santos *et al*., 2003; Santos *et al*., 2005; Spuy *et al*., 2006). If the thermal resistance of element i is defined as $R_i = \Delta x_i / k_i A$, where A_t is the heat flux area and $A_t = \Delta x \Delta z$. Then, the resistances between nodes shown on Fig. 5 and can be defined as follows:

$$R_{i,i-1} = \frac{\Delta x_i}{2k_i A_t} + \frac{\Delta x_{i-1}}{2k_{i-1} A_t}, \quad R_{i,i+1} = \frac{\Delta x_i}{2k_i A_t} + \frac{\Delta x_{i+1}}{2k_{i+1} A_t} \tag{5.4}$$

Since we define C_t such that $C_t = A_t \Delta x \rho c'$, Eq (5.2) is multiplied by A_t and Δx, then the equation can be finally written as

$$T_i^{n+1} = \left(\frac{\Delta t}{C_t R_{i,i-1}}\right) T_{i-1}^n + \left(1 - \frac{\Delta t}{C_t R_{i,i-1}} - \frac{\Delta t}{C_t R_{i,i+1}}\right) T_i^n + \left(\frac{\Delta t}{C_t R_{i,i+1}}\right) T_{i+1}^n \tag{5.5}$$

where Δt depends on the casting speed by $V_{\text{casting}} = \Delta z / \Delta t$. Due to the conditional stability of the model, Δt needs to be small enough.

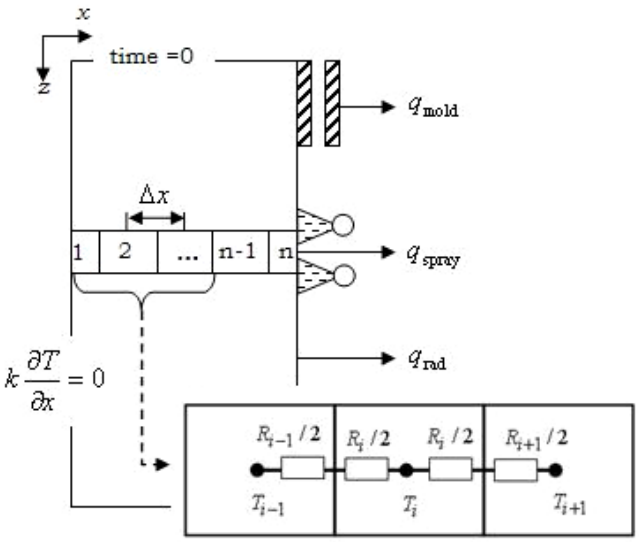


Figure 5.3 Thermal resistance in each element

5.1.3 Boundary Conditions

- The turbulence in the liquid metal is considered by defining an effective thermal conductivity as $k_{eff} = k_{lid} \times A$, where A varies between 3 and 7 (Santos et al., 2003).
- At $t = 0$, the temperature profile of a slice at meniscus is equal to the pouring temperature (T_p).
- Heat flux between the elements at the center line of the slab is neglected, thus $k(\partial T / \partial x)\big|_{x=center} = 0$.
- Heat flux at the outer surface of slab in the different regions; mold, spray cooling, and air cooling, are defined as $-k(\partial T / \partial x)\big|_{x=0} = q_{out}$, where q_{out} is given in Table 5.1 (Hardin et al., 2003).

Table 5.1 Heat flux in each position of strand

Location	Boundary condition
Mold	$q_{mold} = 2 \times \left[(8.5 \times 10^5) e^{-0.386(t)} + 4 \times 10^3 (t) - 3.6 \times 10^4 \right]$
Spray zone	$q_{spray} = h_s (T - T_w), h_s = \frac{1570.0 W^{0.55} [1 - 0.0075 (T_W - 273.15)]}{\alpha}$
Natural cooling	$q_{rad} = \sigma \varepsilon (T^4 - T_{air}^4)$

where t is dwell time in the mold (s), h is the heat transfer coefficient (W/m^{2}K), T_w is water temperature (K), T_{air} is air temperature (K), W is the spray cooling flux (L/m^{2}s), α is a machine dependent calibration factor (set to 4 (Mizikar, 1967)), σ is Boltzmann's constant (5.67×10^{-8} Wm^{-2}K^{-4}), and ε is emissivity (0.8).

5.1.4 Model Validation

To simulate the solidification, the steel slab casting with carbon 0.1% and speed 0.65 m/min is used as the case study. Since the slab is symmetry, the simulation is performed by only a half of thickness of the slab. Therefore, by the mathematical model developed on MATLAB and the parameters on Table 5.2, the simulation result is shown in Figure 5.4. To validate, the result in each depth is compared with Ishiguro and Itaoka (1974)'s result, which was validated by a real factory data (the spots around the line).

Table 5.2 Input parameters for continuous slab casting

Thermo physical properties of steel		Geometry of slab caster	
Parameter	Value	Parameter	Value
Liquidus temp. (°C)	1520	Section size (m)	1.90 x 0.25
Solidus temp. (°C)	1480	Mould length (m)	0.625
Heat of fusion (J/kg)	265000	Unbending point (m)	18
Pouring temp. (°C)	1530	Mould material	Copper
Density (kg/m^3)	7850	Spray zones (1-8)	
Heat capacity (J/kg K) (Meng and Thomus, 2003)	$\begin{cases} C_p = 1431 & T < 750 \\ C_p = 3849 - 3.766T(^\circ C) & 750 \le T < 850 \\ C_p = 648 & 850 \le T < 1100 \\ C_p = 268 + 0.334T(^\circ C) & 1100 \le T < T_{sol} \\ C_p = 772 & T_{sol} \le T < T_{liq} \\ C_p = 787 & T \ge T_{liq} \end{cases}$	Pos. (m)	Flow (l/m^{2}s)
		1) 1	6.1
		2) 3.5	2.3
		3) 5	1.7
		4) 7.7	1.0
		5) 11	1.2
		6) 14.5	0.6
Ther. cond. (W/m K)	$k_{liq} = 6 \times 43.0$, $k_{sol} = 33.0$	7) 16.3	1.5
Steel emissivity (ε)	0.8	8) 17	0.6

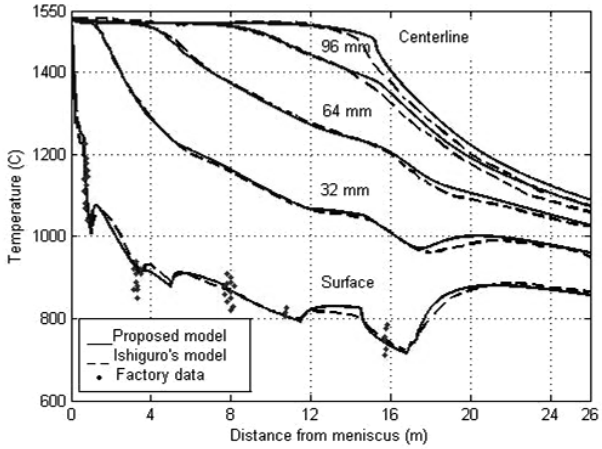


Figure 5.4 Comparison of the proposed model and Ishiguro and Itaoka (1974)

5.2 Cost function specification

To define the objective function, the key success factors of the mass customization production as the quality and resource constraints must be taken into account. For quality, the casting condition and metallurgical criteria need to be satisfied while the water flux needs to be reduced for the resource constraint. Practically, the tradeoff between quality and resource is necessary and frequently occurs. The final decision depends on the company policy and the current market situation.

5.2.1 Quality criteria

- *Shell thickness at mold exit*: In order to avoid breakout, shell thickness has to be greater than the minimum value. For simplicity, the minimum shell thickness will be transformed in term of the temperature, which can be approximated by 10% of the value of the half slab thickness. Therefore, the sub-objective function can be written by Eq. (5.6), where $T_{10\%}$ is the temperature at 10% of the half slab thickness and T_S is the solidus temperature.

$$J_1 = \max\left[0, T_{10\%} - T_S\right] \tag{5.6}$$

- *Metallurgical length*: In order to avoid the internal and transversal cracking, the steel have to be completely solidified before the unbending point. By this constraint, the temperature at the center of the strand at the bending point (T_{center}) must be lower than the solidus temperature. Therefore, the sub-objective function can be written by Eq. (5.7).

$$J_2 = \max\left[0, T_{\text{center}} - T_S\right] \tag{5.7}$$

- *Surface temperature at unbending point*: In order to avoid the transverse surface cracking due to bending under the lower ductility, the strand surface temperature at the bending point (T_{surface}) should be kept out of the lower ductility zone,

which is the interval 700-900℃ for low carbon steel (Santos et al., 2003). Therefore the sub-objective function can be defined in Eq. (5.8), where T_{upper} and T_{lower} are the upper and lower limits of the low ductility zone, respectively.

$$J_3 = \begin{cases} \max\left[0, T_{\text{surface}} - T_{\text{upper}}\right] & T_{\text{surface}} \geq T_{\text{upper}} \\ \max\left[0, T_{\text{lower}} - T_{\text{surface}}\right] & T_{\text{surface}} \leq T_{\text{lower}} \end{cases} \tag{5.8}$$

- *Reheating temperature*: To avoid the midway surface cracking, the reheating temperature, which leads to the development of the tensile stress at the solidification front, must be limited at 100℃ (Spuy et al., 1999). Therefore, the sub-objective function can be written in Eq. (5.9), where n is the number of spray zone and the reheat temperature in each zone is $\Delta T_{\text{zone}} = T_{\max \text{ surface}} - T_{\min \text{ surface}}$.

$$J_4 = \sum_{i=1}^{n} \max\left[0, \Delta T_{\text{zone } i} - \Delta T_{\max}\right] \tag{5.9}$$

5.2.2 Resource Criteria

- *Water flow rate*: Typically, only physical constraints as the upper and lower bounds of the water flow rate for each spray zone will be taken into account in case of quality improvement. However, the water flow rate is also minimized in case of continuous casting. Therefore, the summation of the flow rate is used as the sub-objective function, shown in Eq. (5.10).

$$J_5 = \sum_{i=1}^{n} W_i \tag{5.10}$$

Finally, the objective function is the summation of the normalized value in each sub-objective function and is compromised by the penalty weight (p_i) as shown in Eq. (5.11).

$$J_{\text{total}} = \sum_{i=1}^{k} \frac{J_i - J_{i\min}}{J_{i\max} - J_{i\min}} \, p_i \tag{5.11}$$

where $J_i^{\min}$ and $J_i^{\max}$ are the minimum and maximum values of the sub-objective function i, respectively.

5.3 Spray Cooling Optimization

The optimization based on the solidification model for the spray cooling is performed by off-line. After the quality management system receives the production schedule, the schedule data such as the mold size, steel grade, chemical analysis, etc. is utilized to set the solidification model in each steel grade. Then, the optimal spray patterns are calculated based on the solidification model and sent to set the regulators/controllers in the automation level. For the optimization, two case studies on the improvement of quality and productivity are achieved with GA. The parameters of GA as shown in Table 5.3 are defined by a Trial and Error method.

Table 5.3 Parameters of GA in spray cooling optimization

Parameters	Value
Representation	Binary
Population size	60
Max. iteration	40
Crossover	Two points (rate 0.8)
Selection	Stochastic uniform
Mutation	Random (rate 0.01)

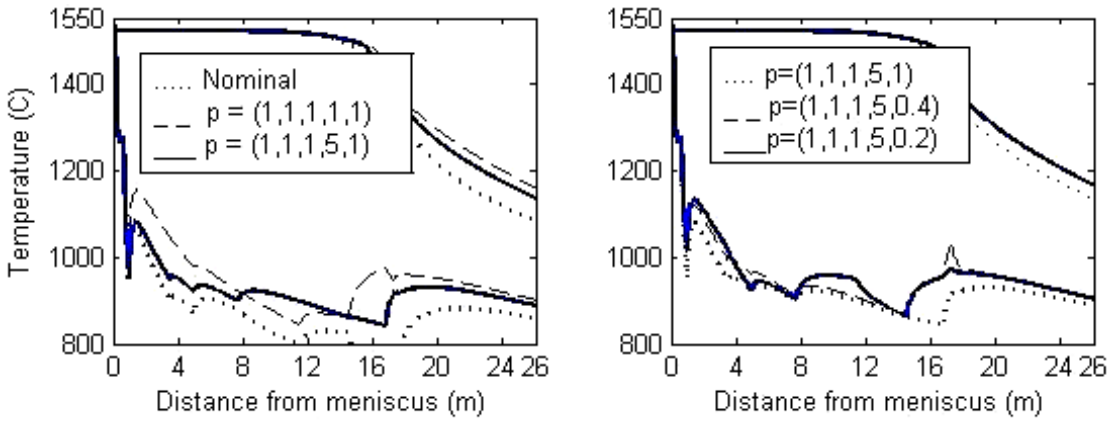
5.3.1 Case I: Quality Improvement

In this experiment, quality will be improved by tuning the penalty weights under the constant casting speed (0.65 m/min). The weight is trialed by starting from $p_i = \{1, 1, 1, 1, 1\}$ until all of the quality constrains are satisfied (J_1 to J_4). Then, the tuned weights will be used in Case II.

The results of the penalty weight tuning are shown in Table 5.4 and Figure 5.5(a) and (b). By the starting weights, T_{surface} and $\sum W_i$ are satisfied (can be reduced) while the reheating at zone 1 and 5 are not satisfied. Therefore, p_4 should be increasingly penalized for reducing the reheating at zone 1 and 5. After increasing p_4 until equal 5, only zone 5 is satisfied while zone 1 is not satisfied because the spray cooling rate is limited by p_5. After decreasing p_5 to 0.2, the reheating of zone 1 is almost satisfied. However, decreasing trend can be noticed. Thus, this pattern is used as the initial conditions in Case II. For $T_{10\%}$, it is noted that the patterns of the cooling rate are not affected. The trend of objective value in each generation is shown in Figure 5.6 and the optimal spray patterns can be shown in Figure 5.7.

Table 5.4 Optimum temperature in each position

Control variables	Original cooling rate	Optimal cooling rate in each p_i	
		$\{1,1,1,1,1\}$	$\{1,1,1,5,0.2\}$
$T_{10\%}$	1503.3	1503.3	1503.3
T_{center}	1290.9	1359.9	1336.5
T_{surface}	833.71	968.53	972.91
ΔT_{zone}	{170.3, 40.8, 34.1, 80.8, 33.6, 107.5, 32.9}	{121.8, 68.9, 60.3, 77.5, 26.8, 115.7, 28.4}	{116.3, 94.4, 43.0, 57.0, 85.1, 92.2, 19.2}
W_i	{6.1, 2.3, 1.7, 1.0, 1.2, 0.6, 1.6, 0.6}	{5.1, 1.4, 1.4, 1.0, 0.9, 0.6, 1.2, 0.3}	{5.9, 1.4, 1.6, 1.1, 0.5, 0.7, 0.2, 0.15}
$\sum W_i$	15.1	10.5	11.7



(a) Tuning p_4 (b) Tuning p_5

Figure 5.5 Comparison of surface/center temperature with tuning p_i

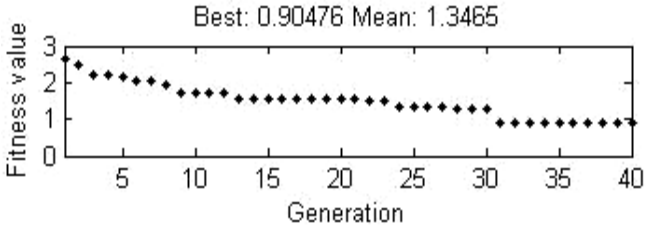


Figure 5.6 Objective value in each generation for $p_i = \{1,1,1,5,0.2\}$

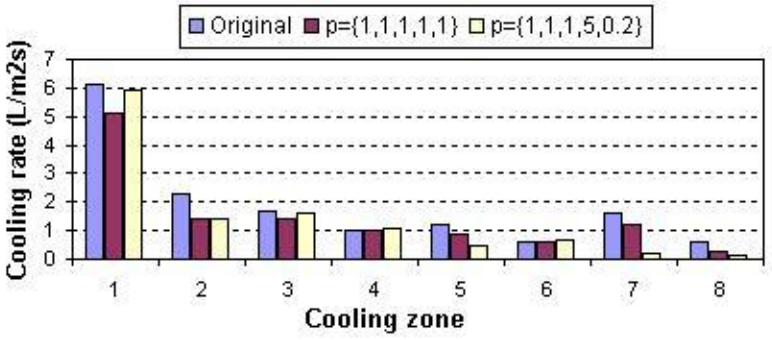
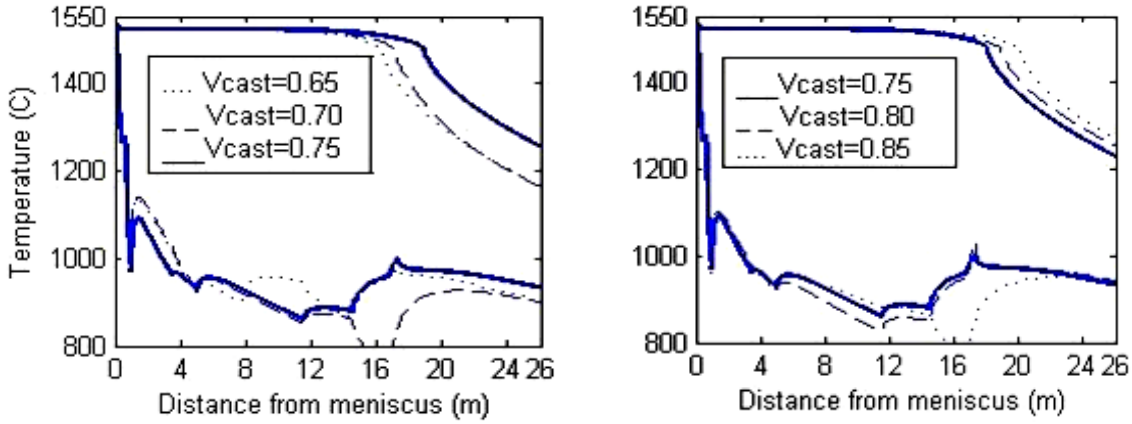


Figure 5.7 Optimal spray pattern in each p_i

5.3.2 Case II: Productivity Improvement

In this experiment, the productivity will be improved by increasing the casting speed (from 0.65 m/min with 0.5 per step) under the constant penalty weight defined in Case I until the quality constraints are infringed. However, the reheating at zone 1 is permitted to use the limit of 116.3 ($\Delta T_{\text{zone }1} \le 116.3$), which is the result from the optimum spray cooling rate in Case I.



(a) Varying speed between 0.65-0.75 m/s (b) Varying speed between 0.75-0.85 m/s

Figure 5.8 Comparison of surface/center temperature with tuning p_i

Table 5.5 Optimum temperature in each casting speed

Control variables	Optimal cooling rate in each casting speed		
	V_{cast} =0.75	V_{cast} =0.80	V_{cast} =0.85
$T_{10\%}$ T_{center} T_{surface}	1506.2 1479.2 986.7	1509.2 1492.2 989.17	1513.9 1502.8 910.11
ΔT_{zone}	{113.8, 89.8, 48.8, 70.4, 25.2, 99.5, 80.6}	{123.4, 59.2, 37.0, 76.4, 31.9, 99.1, 78.5}	{130.3, 41.6, 36.2, 53.5, 22.8, 85.2, 71.8}
W_i	{6.2, 1.5, 1.7, 1.2, 1.1, .7, 2.3, 0.1}	{8.7, 2.0, 1.8, 1.2, 1.2, 0.7, 0.2, 0.1}	{9.9, 2.3, 1.8, 1.1, 1.0, 0.8, 1.3, 0.2}
$\sum W_i$	14.8	15.9	18.4

After varying the casting speed, the results shown in Figure 5.8 and Table 5.5 point that the maximum casting speed is 0.75 m/min while the quality criterion can be satisfied. For the higher speed such as 0.80 and 0.85 m/min, not only the reheating limit at the zone 1 is infringed but T_{center} is also infringed. When considering the resource consumption, it is noted that a higher casting speed will consume a high water rate.

In concluding, this chapter on the optimization of the secondary cooling of the slab caster was performed based on the solidification model. Two experiments on improvement of quality and productivity were raised as the case studies and solved by GA. The results reveal that the quality improvement was achieved by increasing the penalty weight of the reheating zone and decreasing the weight of the flow rate. However, although the adjustment of those weights leads to the increment of the water resource consumption, it could be admitted (less than the original case). In case of the productivity improvement, the casting speed could be increased to 0.75 m/min while it still satisfied the quality constraints.

CHAPTER 6 COMMERCIALIZATION PLAN

In this chapter, the commercialization plan is presented. The proposed framework from previous chapters is developed to be a business model and a business plan, which explain strategies, marketing plan, and expenses. The business focuses on the planning software for steel factories.

6.1 Business Model

The proposed product is a predictive steel planning system on web service application. This system can be a plug-in module that is compatible to MES or other factory management systems.

6.1.1 Value Proposition

Customers

- **Target**: The target is a niche market, where is on the segment of metal, steel, aluminum, and copper manufactories/plants.
- **Customer problems and needs**: The steel making factories needs an effective planner tailored for the steel making factories. The current problem is that there are a few specific planning softwares for the steel making factories. The existing software products have high cost due to the customization.

Value proposition

The product offers the following list to the potential target customers.

- *Cost reduction*: With the product, the cost-oriented planning or productivity-oriented planning can be achieved. The product will enable our clients to optimize their output.
- *Simple operation*: It is easy to set up the daily operation. It can fulfill, and make the right service/information to the right machine in the production line. The product provides the customized fit to each production technology of the steel and iron factories segment.
- *Leveraging from a large pool of expertise*: By the alliances with technological and training partners, skills and intellectual capacity of these partners will be in the fields of product support, implementation, and execution.

Competition

Two types of competitors that also provide customized consultancy, web-based, and access to the underlying information for the production management are follows:

- Full-line provider and the developers of numerous innovative processes such as steelmaking, i.e., Siemens and ABB.

- Other production planning software companies with specialty in steel manufacturing, i.e., Quint and ATEAK Automation.

To differentiate our product from competitors' products, the product will be built to support an integrated enterprise software that the operations management is performed in real-time. The product can be updated the production data (e.g., machine available status and disruption status) in real-time.

6.1.2 Profit Formula

6.1.2.1 *Revenue Estimation*

Revenue is estimated from selling units in 5 years. Total estimated revenue for years 1–5 is around 87.8 million Baht. It is expected that since the 1st year to the 5th year, the application of 4, 9, 18, 31 and 47 units can be sold and the revenues are 3.2, 7.2, 14.4, 25.2, and 37.2 million Baht, respectively (the detail is presented in next section). The revenue estimation is as follows:

$$\text{Revenue} = \text{Application License} \times \text{Project Implementation} \quad (7.1)$$

Remark: an application is combination of web and database applications.

6.1.2.2 *Cost Estimation*

The cost estimation of 5 years is calculated based on Marginal costing. It contains both fixed components and variable components as follows:

- Mixed costs include the cost of labor and training. The estimated costs from years 1–5 are around 5.5, 6.8, 9.4, 10.8 and 13.7 million Baht, respectively.
- Marketing and sale costs include advertisement and travel expenses that are estimated around 0.3, 0.55, 0.6, 0.8 and 0.8 million Baht, respectively.
- Management expenses of years 1–5, e.g., salary, telecommunication, rent, and public utility are estimated around 5.5, 6.7, 9.4, 10.5, 13.4 million Baht, respectively.

6.1.2.3 *Profit Estimation*

Total accumulative profit of year 1-5 is around -8.7, -6.7, -4.3, 2.3 and 7.0 million baht, respectively

6.1.3 Key Application Workflow Processes

Key application workflow is illustrated in Figure 6.1. It is to submit the predictive automate plant. The process flow starts from the planning by using the real-time database and the chosen simulation model to manage resources and machines. The simulation is run from the data each plant’s production and quality database in the DBMS. The key processes of application development are as follows:

1. Improve production line management in terms of technology performance.
2. Develop production capacity such as to support more steel factory types.

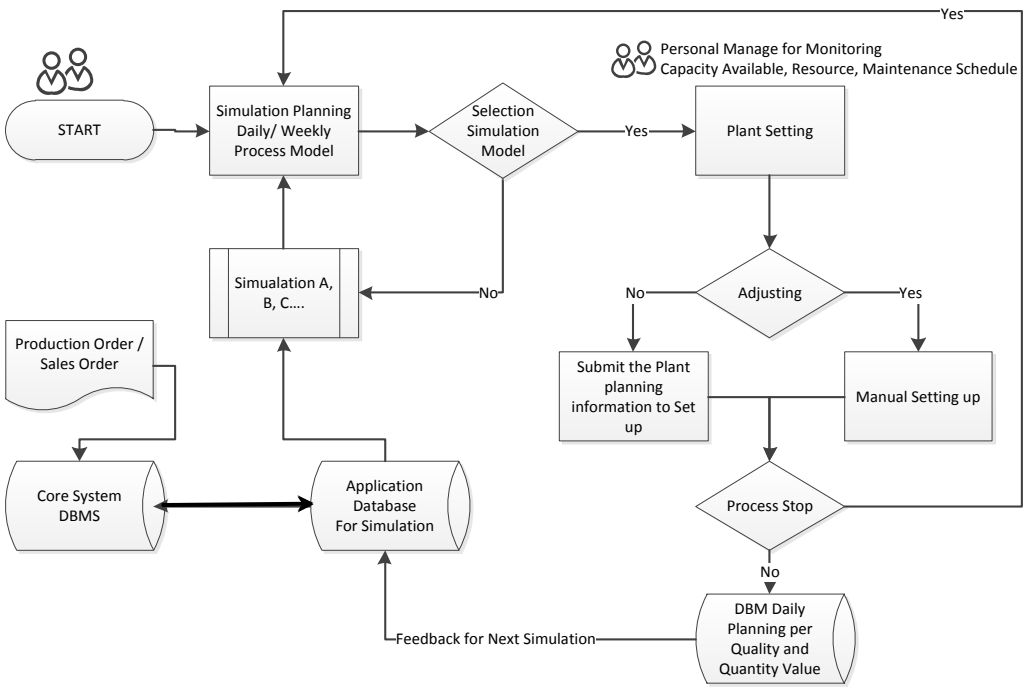


Figure 6.1 Key application workflow processes

- *Process simulation model for daily/weekly planning*: The process simulation model, which can select types of simulations such as deterministic (standard) or stochastic mode.

- *Application database*: The application continuously communicates with application database to real-time gather the input information, e.g., sale orders and available machine status. With this database, the application can collaborate with the core system (MES) and others as follows:

  - The application works together with Master Production Schedule (MPS) to process and simulate production management for setting up machines and illustrates as the graphical user interface (GUI).
  - Application supports the real time updated for the machine maintenance data and rescheduling the process by less time consuming process at this stage.
  - Application supports the mobile or tablet devices in with every operation systems (OS), thus using in the production line to re-check and set up process will be easily process and manage.

- *Plant set up and data control*: The current optimal plan will not only be automatically derived from application to setup the plant automation but the manual adjusting can be also obtained. In addition quantity and quality from the plant output in the production line DBMS will feedback in real time to the planning plugin module application for the next simulation processes.

6.1.4 Resources

6.1.4.1 *Human Resources*

The expected company is shown in Figure 6.2. The organizational chart contains 5 sections under a managing director. The general management section includes financial and human resource controlling while the sales management section includes the sales and marketing team. The detail of roles in this organization is explained as follows:

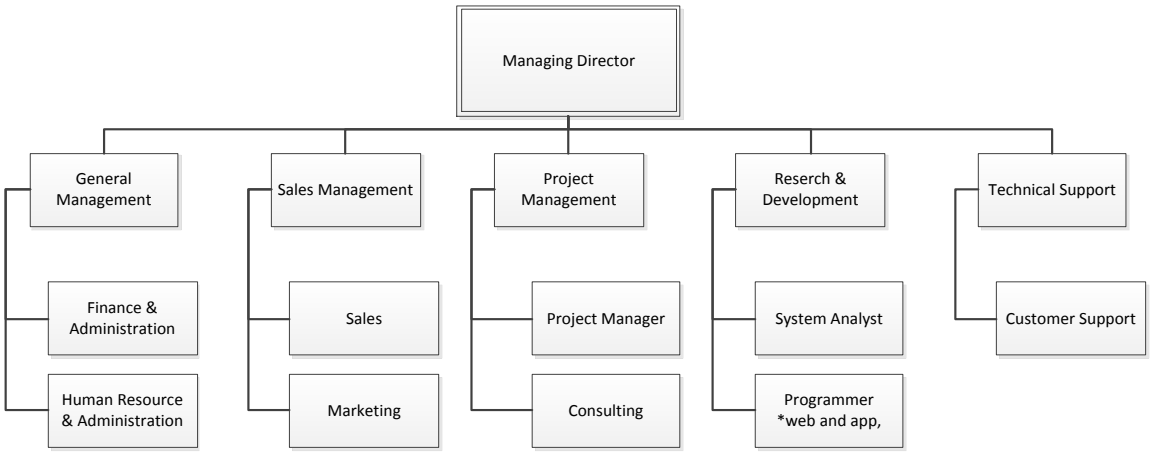


Figure 6.2 Organization structure

1. Managing director is responsible in the performance, policy, and strategy of the company.
2. The general management function is to support the internal office operations on financial and resource management as follows:

    a. Finance section aspect is responsible to manage financial health, budgeting, and financial reporting, and monitoring for the management level.
    b. Human resource activities are mainly taking care of compensation and benefits, training, employee relations, and recruitment.

3. Sales management is responsible to overseeing that all sales representative meet their sales targets of the organization through business trends, costs, revenues,

financial commitments, to project future revenues while the marketing section is responsible for brand promotion.

4. Project management role is to manage a project through its life cycle, including controls and monitors of the project scope, time, and cost in managing project requirements. The project team members are responsible to on-site implement and customize the application according to each client's requirement as per guided and controlled by the project manager.
5. Technical support deals with taking the technical issues from customers to analyse the troubleshooting for customers.

Table 6.1 HR cost in year 1 to 5.

Yr	Position	No. of staff	Salary Rate	Grand Total	Salary Expense				
					2015	2016	2017	2018	2019
				83,814,862	4,812,000	5,628,600	8,010,030	8,410,532	10,871,058
1	Managing director	1	90,000	14,213,246	1,080,000	1,134,000	1,190,700	1,250,235	9,558,311
1	Project manager	1	60,000	11,786,437	720,000	756,000	793,800	833,490	8,683,147
1	Sales & marketing	1	30,000	9,797,210	360,000	378,000	396,900	416,745	8,245,565
1	Finance and Accounting	1	30,000	9,797,210	360,000	378,000	396,900	416,745	7,807,982
1	HR & admin	1	20,000	8,550,691	240,000	252,000	264,600	277,830	7,516,261
1	System analyst	1	30,000	8,630,324	360,000	378,000	396,900	416,745	7,078,679
1	Programmer	3	25,000	10,593,139	900,000	945,000	992,250	1,041,863	6,714,027
1	Consulting	3	18,000	9,142,336	648,000	680,400	714,420	750,141	6,349,375
1	Customer support	1	12,000	1,304,270	300,000	151,200	330,750	347,288	175,033
2	Sales & marketing	1	30,000	1,454,783		360,000	330,750	347,288	416,745
2	Consulting	1	18,000	954,234		216,000	238,140	250,047	250,047
3	Sales & marketing	1	30,000	872,596			360,000	250,047	262,549
3	Programmer	2	25,000	1,891,500.00			600,000	630,000	661,500
3	Consulting	1	18,000	1,324,050.00			20,000	441,000	463,050
3	Project manager	1	60,000	2,269,800.00			720,000	756,000	793,800
5	Consulting	2	18,000	960,000.00					960,000
5	Sales & marketing	1	30,000	360,000.00					360,000
5	Project manager	1	60,000	720,000.00					720,000

6.1.4.2 *Office and Facilities*

The workplace with the decoration and furniture is provided to support the business activities and running. The total expense in the first year is estimated around 300,000 Baht.

6.2 Business Plan

The business plan for establishing this software business is roughly implemented on a market growth strategy with the following phases:

- Market penetration strategy in the 1st year by local selling.
- Market development strategy in the 2nd year is on finding the new business to business (partners) in global market (China and Korea).
- Product development strategy is to support the expansion in new market at the 4th year onward. (Europe and UAE Zones).
- Net margin is expected around 17.12%. The expense on labor cost is increased by 5% and other expenses are based on activities in each year.

6.2.1 Marketing and Sales Plan

The company needs effective market and business services, which can push the company to be at top of our clients' minds. Focusing on proactive market, the target market seeks the companies that can provide an efficient and effective IT system ensuring their business objectives. The clear product and marketing positioning will effectively drive the sales approach to sell to the target market via the company sales representatives and supporting from overseas business alliances. The target companies should be large enough to require a high-quality, cost effective, attractive development to the production and quality management in the steel making.

The market is separated into local market (in Thailand) implemented by our company and the overseas market, especially, in the Europe, China and Korea, through business partners. The marketing strategy is on the influencer marketing, which searches or builds the influencers for the on-line audiences. By this marketing strategy, the advertisment on the on-line media (e.g., Facebook, blog, and websites) and printing media (e.g., magazines) are essential. The marketing action plan for 5 years is shown in Table 6.2.

A sales plan (that is the results from activities in Table 6.1) will cover strategy what the leaders want to accomplish, and which battles they choose to fight. The tactics will determine how an individual battle is fought. The sales action plan is described as Table 6.3.

Table 6.2 Marketing action plan

Year	Marketing action	Freq.	Budget (Baht)	Outcome Expectations
1	Website creation with performing Google analytics	1	50,000	Create company profile and branding awareness
	Documentation in magazine and brochure	1	125,000	Create company profile and branding awareness
	Social marketing, e.g., Facebook, Twitter, and LinkedIn	∞	25,000	Increase in the number of perceptual in target and potential segment by 30%
	Economic diversity by doing network visitation for local factory visits	20	40,000	Increase customer engage and trust, get feedback and relationship
	Tradeshow coverage in manufacturing industries	2	60,000	Create company profile and branding awareness
	Total Marketing Spending Year#1		300,000	
2	Website creation in English/Chinese language version for international Business	1	50,000	Create company profile and branding awareness
	Website maintenance	∞	5,000	Potential segment by 30%
	Documentation in magazine and brochure categorized by customer segment (Thai and international marketing)	10	200,000	Create company profile and branding awareness
	Social marketing, e.g., Facebook, Twitter, and LinkedIn	∞	15,000	Increase in the number of perceptual in target and potential segment by 30%
	Economic diversity by doing network visitation for local factory visits *(souvenir inc.)	10	60,000	Increase customer engage and trust, get feedback and relationship
	Tradeshow coverage in manufacturing industries – Local, and software industries	2	70,000	Create company profile and branding awareness and attention
	Tradeshow –International coverage in manufacturing industries	2	150,000	Create company profile and branding awareness and find targeted business alliance
	Total Marketing Spending Year#1		550,000	

Year	Marketing action	Freq.	Budget (Baht)	Outcome Expectations
3	Website (Thai/English) maintenance	∞	5,000	Potential segment by 30%
	Documentation in magazine and brochure categorized by customer segment (English language for international marketing)	1	100,000	Increase in the number of perceptual in target and potential segment by 30%
	Social marketing, e.g., Facebook, Twitter, and LinkedIn	∞	5,000	Increase in the number of perceptual in target and potential segment by 30%
	Economic diversity by doing network visitation for local factory *(souvenir inc.)	15	50,000	Increase customer engage and trust, get feedback and create relationship
	Economic diversity by doing network visitation for overseas factory and business partners *(souvenir inc.)	4	180,000	Increase customer engage and trust, get feedback and create relationship
	Tradeshow coverage in manufacturing industries – oversea alliance, and software industries	2	80,000	Create company profile and branding awareness and attention
	Tradeshow –International coverage in manufacturing industries	3	180,000	Create company profile and branding awareness and attention, find targeted business alliance
	Total Marketing Spending Year#3		600,000	
4	Website (Thai/English) maintenance	∞	5,000	Potential segment by 30%
	Documentation in Magazine and brochure categorize by customer segment (Korea language for international marketing)	1	160,000	Increase in the number of perceptual in target and potential segment by 30%
	Social marketing, e.g., Facebook, Twitter, and LinkedIn	∞	25,000	Increase in the number of perceptual in target and potential segment by 30%
	Economic diversity by doing network visitation for local factory *(souvenir inc.)	15	100,000	Increase customer engage and trust, get feedback and create relationship
	Economic diversity by doing network visitation for overseas factory and business partners *(souvenir inc.)	4	250,000	Increase customer engage and trust, get feedback and create relationship

Year	Marketing action	Freq.	Budget (Baht)	Outcome Expectations
	Tradeshow coverage in manufacturing industries – oversea alliance, and software industries	2	60,000	Create company profile and branding awareness
	Tradeshow –International coverage in manufacturing industries	3	200,000	Create company profile and branding awareness and find targeted business alliance
	Total Marketing Spending Year#4		800,000	
5	Website (Thai/English) maintenance	∞	5,000	Potential segment by 30%
	Documentation in magazine and brochure categorize by customer segment (Chinese language for international marketing)	1	160,000	Increase in the number of perceptual in target and potential segment by 30%
	Social marketing, e.g., Facebook, Twitter, and LinkedIn	∞	25,000	Increase in the number of perceptual in target and potential segment by 30%
	Economic diversity by doing network visitation for local factory *(souvenir inc.)	15	100,000	Increase customer engage and trust, get feedback and create relationship
	Economic diversity by doing network visitation for overseas factory and business partners *(souvenir inc.)	4	250,000	Increase customer engage and trust, get feedback and create relationship
	Tradeshow coverage in manufacturing industries – oversea alliance, and software industries	2	60,000	Create company profile and branding awareness and attention
	Tradeshow –International coverage in manufacturing industries	3	200,000	Create company profile and branding awareness and find targeted business alliance
	Total Marketing Spending Year#5		800,000	

Table 6.3 Sales action plan

Year	Sales action	Main target	Sale unit
1	Create a website that contains the product and service information	Local Customers	4 applications
	Telephone call	Thai metal companies	
	Contact and visit local factories	Organizations that is interesting in the solution	
	Tradeshow 2 times	Thai metal companies	
	Advertise in industrial magazine	Thai metal companies	
2	Telephone call	Thai metal companies	9 applications
	Contact and visit local factories	Organizations that is interesting in the solution	
	Contact and visit overseas business partners and factories	Organizations that is interesting in the solution	
	Tradeshow 4 times	Thai and overseas metal companies	
3	Telephone call	Thai metal companies	18 applications
	Contact and visit local factories	Organizations that is interesting in the solution	
	Contact and visit overseas business partners and factories	Organizations that is interesting in the solution	
	Tradeshow 5 times	Thai and overseas metal companies	
4	Telephone call	Thai metal companies	31-32 applications
	Contact and visit local factories	Organizations that is interesting in the solution	
	Contact and visit overseas business partners and factories	Organizations that is interesting in the solution	
	Tradeshow 5 times	Thai and overseas metal companies	
5	Telephone call	Thai metal companies	47-48 applications
	Contact and visit local factories	Organizations that is interesting in the solution	
	Contact and visit overseas business partners and factories	Organizations that is interesting in the solution	
	Tradeshow 5 times	Thai and overseas metal companies	

6.2.2 Management Plan

There is no absolute rule regarding as a right framework because there are many different frameworks and methodologies for strategic planning and management. Based on consideration of resources and an assessment of the internal and external environments, the resource in each year will be increased to support the marketing planned.

- Starting the business on 1st year - to the compact steel factories in Thailand. The target companies are large enough to require the high-quality web-based software for industrial automation.
- Running the 2nd year onward - our most important group of potential customers will be high level business executives in large corporations on global segment. Then the business is expanded to others countries steel factories in both integrated and compact factories.
- From the 4th year – the business is expanded to other manufacturing such as coppers and aluminum factories in the global market segmentation.

The management activity plan for years 1–5 is summarized in Table 6.4 while the business management plan for years 1–5 is summarized in Table 6.5.

Table 6.4 Management activity plan for years 1–5

Year	Sales and Marketing Activity Management activity
1	Sales and Marketing - Launch the product in local market. - Introduce the company and prepare the marketing material e.g., website, documentation, and brochure. Management - Prepare office and facilities e.g., office rental, computer, and network. - Recruit staffs. - Train staffs (in-house training).
2	Sales and Marketing - Launch the product in international market. - Perform quality and efficiency improvement. - Build the business partners in Asia regional. Management - Build relationship on the existing and new target customers. - Perform strategic and resource planning to lead employees to meet the organization's expectations on productivity, quality, and goal accomplishment. - Perform product development and preparation for higher target volume in the coming year.
3	Sales and Marketing - Launch the product in international market (Asia regional). - Research and develop the product for the new factories type. - Build the business partners in Europe regional. Management

Year	Sales and Marketing Activity Management activity
	- Plan the operation and function for higher production target. - Perform the job satisfactorily and receive feedback on a regular basis. - Perform product development and preparation for new market. - Train staff and partners for new product technology and the new market.
4	Sales and Marketing - Launch the product in international market (EU Regional). - Research and develop the product for new factories type. - Build the business partners in UAE regional. Management - Prepare materials for high production volume. - Build relationship on the existing and new target customers.
5	Sales and Marketing - Launch the product in international market (UAE regional). - Research the new market and new product for other type of factories. Management - Reduce cost. - Expand international market. - Purchase materials and equipment according to sales target.

Table 6.5 Business management plan for years 1–5

Description	Year 1	Year 2	Year 3	Year 4	Year 5
	2015	2016	2017	2018	2019
Sales and Marketing cost					
Advertisement	300,000	550,000	600,000	800,000	800,000
Miscellaneous (Entertainment and Travelling)	120,000	300,000	480,000	1,200,000	1,680,000
Total	420,000	850,000	1,080,000	2,000,000	2,480,000
Management expense					
Salary* inc. R&D team	4,812,000	5,628,000	8,010,030	8,410,532	10,871,058
Rent	240,000	240,000	240,000	264.000	264.000
Public utility	48,000	60,000	72,000	84,000	96,000
Office disposables	6,000	6,000	10,500	6,000	6,000
Total	5,106,000	5,934,000	8,332,530	8,500,796	10,973,322
Grand total cost	5,526,000	6,784,000	9,412,530	10,500,796	13,453,322

6.2.3 Investment Cost

The primary funding source is estimated about 1,207,000 Baht from shareholders. The investment includes facilities services such as the expenses of building the office, decoration, and furniture as the 1st year in Table 6.6.

Table 6.6 List of investment cost

Details	Quantity	Estimated cost (Baht)	Life time (year)
Office decoration and office workstation	-	300,000	5
HW: Server	2	150,000	5
HW: Notebook for Developer	6	240,000	5
HW : Notebook	14	280,000	5
HW : Smart Devices	4	117,000	5
Software	1	100,000	3
Printer	1	20,000	5
Total		1,207,000	

6.2.4 Financial Plan

This financial planning starts with estimating the project cost, expenses, and project income. Evaluation of business is calculated from the financial statements such as the pro forma income statement in Table 6.7 and the pro forma balance sheet in Table 6.8.

Table 6.7 Pro forma income statement year 2012–2016

Description	Year 1	Year 2	Year 3	Year 4	Year 5
	2015	2016	2017	2018	2019
Revenue Company + Partners Implementation					
Total revenue	3,200,000	7,200,000	14,400,000	25,200,000	37,800,000
Production cost	6,088,000	6,889,600	9,537,530	10,764,532	13,717,058
Gross profit	-2,888,000	310,400	4,862,470	14,435,468	24,082,942
Management cost	5,586,000	6,844,600	9,472,530	10,824,532	13,777,058
Profit	-8,474,000	-6,534,200	- 4,610,060	3,610,937	10,305,884
Depreciation	254,733	254,733	254,733	254,733	254,733
Profit before tax	-8,728,733	-6,788,933	-4,864,793	3,356,204	10,051,151
30 % Tax	0	0	0	0	0
Net profit	-8,728,733	-6,788,933	-4,864,793	3,356,204	10,051,151

Table 6.8 Pro forma balance sheet year 2012–2016

Description	Year 1	Year 2	Year 3	Year 4	Year 5
	2012	2013	2014	2015	2016
Current assets					
Cash	7,831,460.00	14,964,444.00	19,622,700.00	17,814,253.00	12,197,974.00
Account receivable(A/R)	2,240,002	6,000,007	12,240,015	21,960,027	34,020,042
Total current assets	8,631,461.00	15,564,444.75	20,822,701.50	19,914,255.63	15,347,977.94
Fixed assets					
Equipment & computer	1,207,000.00	1,207,000.00	1,207,000.00	1,207,000.00	1,207,000.00
Less depreciation	(254,728.47)	(254,728.47)	(253,739.47)	(221,398.80)	(221,398.80)
Net assets	952,271.53	952,271.53	953,260.53	985,601.20	985,601.20
Total assets	9,583,732.53	16,516,716.28	21,775,962.03	20,899,856.83	16,333,579.14
Current liability					
Account payable (A/P)	555,000.00	699,050.23	1,093,502.95	1,559,878.41	2,020,922.69
30% Tax payable	-	-	-	-	-
Total liability	555,000.00	699,050.23	1,093,502.95	1,559,878.41	2,020,922.69
Owner equity					
Capital	300,000.00	300,000.00	300,000.00	300,000.00	300,000.00
Accumulation profit	8,728,733.00	15,517,666.00	20,382,459.00	19,039,978.00	14,012,656.00
Total equity	9,028,733.00	15,817,666.00	20,682,459.00	19,339,978.00	14,312,656.00
Total equity & Liability	9,583,733.00	16,516,716.23	21,775,961.95	20,899,856.41	16,333,578.69

CHAPTER 7 DISCUSSION

The discussion of results from all experiments is provided in this chapter. It includes overall perspective, practical perspective, and limitation of the frameworks and methodologies in this work. The topics are ranged from the deterministic scheduling for CC process, the robust scheduling for CC process, the deterministic scheduling for SCC process, the robust scheduling for SCC process, and the plan execution.

Deterministic scheduling for CC process

In all experiments of Section 3.2, an optimal caster schedule was obtained under the fixed charges in the melting and refining processes seen as Gantt chart in Figure 3.8. Change of the charge allocation in the melting and refining processes will affect the cast break and waiting time in the casting process, consequently the optimal caster schedule will be changed. Therefore, the steel factories practically design whole production schedule after completing the caster schedule.

In *Experiment I* of Section 3.2, the optimal number of the populations and iterations was defined by considering the computational time into account. For a real factory, a faster computational time is very important. Because it is not only the scheduling task being optimized but the rescheduling task must be urgently revised. It is noted that the case of 50 iterations provides a little better frequency of the achievement while it spends more significantly computational time. In *Experiment II*, although the gradient-based method normally spends shorter total computational time than GA does, the total computational time of both may be close in some cases. The reason is that the main computational time of both GA and gradient-based method cases are spent for running in part of the process simulation. Therefore, if some bad initial solutions are given and lead to bad direction for gradient-based method, it will possibly spend more time.

Robust scheduling for CC Process

In the validation of the uncertainty model from the processing time in Section 3.3, 240 examples (charges or heats), which are used to compare with the MCS results, are selected from the real production data in a half year period (with the mean capacity around 60,000 tons/month). Many charges were discarded because the processing time was extended due to unaware disruptions. For instance, the waiting time due to the machine breakdown disruptions (e.g., breakout at caster and the melting electrode is broken) and the extended melting or refining time due to the contaminated material. By those situations, the processing time is deviated from the normal distribution.

In the first experiment in Section 3.3.4.1 that defined the rescheduling strategy, 5 replications were performed to observe the behavior of the significant factors in the objective function; the queue time, idle time, efficiency, and stability. Although the small numbers of replications were performed, an obvious trend could be noticed. For

the stability factor in each strategy, it may be assigned differently by finely tuning each plant.

In *Case II* of Section 3.3.4.2 that rescheduling from machine failure, it is noted that although the best deterministic objective value is continuously reduced in Figure 3.18, some fluctuations due to the effect of the uncertainty distinctly appeared. The reduction of the objective value reflects the existence of a significantly better schedule but it is not enough in some scenarios. The fluctuation rarely appeared in the 2 remaining cases, which are formulated into the minimax problem. The rescheduling approach in *Case II* can be a solution, which covers both situations of the machine failures; 1) Some equipments are damaged but the caster can be run with a slower casting speed 2) The caster must be shut down for maintenance. However, it does not cover the case of the unpredicted maintenance time (that is later raised and solved in Section 3.4). Regarding the computational time, the optimization without minimax approximately spends around 20 minutes while *Case II* and *III* (minimax optimization) approximately spend around 80 minutes (with the 1.66 GHz Intel(R) Core 2 CPU computer). More computational time in *Case II* and *III* resides on the minimax formulation, which every scenario is considered for evaluating the objective value.

For scheduling under disruption in Section 3.4, handling disruption via the best worst case scheduling provides the results in the same way in Section 3.3. Although the robust schedule from this approach can handle the uncertain disruption or processing time, it provides an unnecessarily high cost. The results address that there are more slacks in this robust schedule. It seems that the best worst case scheduling is not practical enough. Factories need a better choice, which provides a production cost lower than the best worst case scheduling while still keeps an acceptable robustness. It is later raised in Section 4.2.

Deterministic scheduling for SCC process

In designing the optimization model for SCC plant in Section 4.1, it is noted that this case study and many steel factories are based on a compact strip plant (CSP), which has no buffer between the steel making plant and the rolling mill. Therefore, a JIT concept is necessary. Consequently, the earliness and lateness penalties in the optimization model are practically emphasized and the consignment date is set as the due time delivering the slabs to the rolling mill. It should be noted that applications of this methodology on other steel production types producing only slabs or billets can be slightly different. In slab or billet factories, the earliness penalty is often relaxed or approximated from the warehouse cost while the lateness penalty is approximated from the marine cargo delay expense. In addition, the cast break penalty may be relaxed in some cases. In practice, the caster operators can avoid the cast break by reducing the casting speed to wait the charge from the refining furnace. However, this methods (reducing casting speed) generally works for the waiting time less than 30 minutes.

Robust scheduling for SCC Process

From the results of the case study on the peak function in Section 4.2.4.2, it is noted that the single robust measure and the best worst case performance measure may work with the problem that provides low variance as the peak function. However, the single robust measure and best worst case performance measure will likely provide the similar result that is a conservative solution when apply to the high variance problem such as the SCC scheduling as seen in Section 4.2.7.1. The high variance is caused by the high penalty factor of the cast break C_B and the conservative solution is caused by the single objective robust measures tries to avoid the cast break event. Because the cast break penalty is in a part of objective function not in constraints, the reduction of this penalty to decrease the variance is not suggested. Therefore the new robust measure is unavoidable.

Considering the uncertainty assessment by ANN, this dissertation separately created 2 single-output feed-forward backpropagation ANN models; $g_{\mu}^{m}(\cdot)$ and $g_{\sigma}^{m}(\cdot)$. However, the multi-output feed-forward backpropagation can be adopted. Regarding *k*-mean clustering in Section 4.2.5, the number of clusters has to be defined prior assigning machines to each cluster. This dissertation can manually define the initial number of clusters (to 2 and 3) because of having a few numbers of machines in the production. However, for large size production, a searching method for the optimal number of clusters is needed.

For overall perspective of the simulation model for scheduling SCC plant, the mix-integer model in Section 4.2.5 (that combines 2 phases of scheduling) and the proactive scheduling with the new robust measure in Section 4.2.6 have the potentials to be a main simulation model for the proposed framework. However, the proposed proactive scheduling has limitation. It can handle only the small disruptions, which delays the production less than 30 minutes. To handle large disruption or uncertainty that can stop the production, rescheduling or proactive-reactive scheduling as proposed in Section 3.2 is needed.

Plan execution

In executing phase in Chapter 5, it is noted that the penalty weights p_i are manually tuned in experiments (Section 5.3). Although it can include the weights into the process variables in this optimization problem, it may make the problem very difficult due to increasing nonlinearity. Thus, this work takes care of them separately. To adopt the solidification model, an accuracy of model is on setting of the boundary condition and model parameters, including validating the model. However, the model validation in this work is performed by comparing with a relevant independent case in literature (due to lacking of the actual production data from a factory). If possible, it should be directly compared with the primary source such as the actual production data.

Considering overall perspective of the simulation model for executing task, the solidification model is only a model for executing the optimal plan from the earlier phase. Practically, there are several subsystems supporting the casting process such as mold level control and mold oscillator. To extend the proposed framework, the simulations supporting the executing task should also contain other necessary models.

CHAPTER 8 CONCLUSIONS AND FUTURE WORKS

This dissertation proposes the development of MES in a steel making process using simulation. Main contributions of this work are in the area of scheduling model, stochastic scheduling, and collaboration between scheduling and plan executing. A new framework of MES that utilizes 2 simulation models is presented. Firstly, simulation model is designed to effectively support on the production scheduling while another is utilized for executing the schedule from first phase. To design the first simulation model, 2 tiers of production scheduling are focused; 1) cast planning for continuous caster 2) extend to the whole schedule for steelmaking-continuous casting plant. In planning caster, an optimization model is design by taking the technological constraint and economic restricts into consideration. The deterministic scheduling is evaluated by both Genetic Algorithm and Gradient-based method. In the stochastic scheduling for the continuous caster, this dissertation focuses on scheduling under the uncertain processing time and uncertain disruption.

In scheduling under the uncertain processing time, the proactive-reactive scheduling is used. A suitable rescheduling for continuous slab casting can be achieved by the proposed rescheduling system and the performance criterion based on stability. To perform this approach, uncertainty of the processing time is analyzed and modeled. The rescheduling strategy, which is a compromise between the efficiency and stability, is studied for 2 supporting cases; (1) a machine failure and (2) a rush order comming. This problem is formulated to the minimax optimization and solved by GA with simulation. The results show that the new schedules of both cases are the best worst cases that are robust enough to the uncertainty and can provide an accepted performance.

In the case of the caster scheduling under uncertain disruption, a worst case performance scheduling is proposed. It is achieved by the minimax optimization that is solved via GA. The uncertain disruption is characterized and modeled by the actual failure distribution. The result show that besides the charges is suitably allocated into the casts and machines, the best worst case schedule also retains an acceptable performance when the worst disruption occurs. The Gantt-chart revealed that the late starting time of the casters made the result schedule be able to flexibly handle the worst disruption.

The optimization model of the caster plan is extended to the whole schedule for the steelmaking-continuous casting plant. The problem is characterized as a flexible flow shop scheduling that is NP-complete and needs to meet special requirements of the steelmaking-continuous casting plant. The proposed methodology can optimize both sequencing and scheduling phases simultaneously via a proposed model, which assigns the jobs to the steel-making and casting machines and determines the timing and cast sequencing. The hierarchical genetic algorithm with the specific operations is proposed

to search an optimal schedule. The efficiency of the proposed methodology is illustrated by the simulation of the real industrial world case. The experimental results are compared with the standard GA in 3 criteria; (1) the quality of solution, (2) the quantity of achievement in terms of the probabilistic assessment, and (3) the computational time consumption. It can be seen that the proposed methodology can satisfy all criteria, especially in the medium and large scale of population.

This optimization model (whole schedule) is used to study in thecase of uncertain disruption (machine breakdown). The robust optimal schedule is designed under the proposed mix-integer model in terms of the actual costs and the actual machine failure characteristics causing uncertainty from a real factory (the case study showed that the single objective robust measure provided the robust schedule that has too much slack and it is not suitable with the problem). The inversed CDF-based robust measure, which accepts bad scenarios at a low frequency, is a more practical choice to handle disruptions for steel production. To obtain the proposed approach, ANN model was adopted for estimating the mean and variance of the system performance. The machine partition by *k*-mean clustering is the success key in the ANN design. Applying ANN without clustering can result inaccurate prediction from ANN. The methodology was demonstrated with the single objective robust method and confirmed the performance in terms of the computational time by replacing the ANN model with MCS. It addresses that the proposed robust measure with developed ANN model can be implemented effectively and efficiently in a real steelmaking-continuous casting plant.

For plan execution, the optimization on the quality management function is used to be a mockup. The optimal schedule is executed on the secondary cooling of the slab caster. The spray pattern is optimized based on the solidification model, which is constructed by the optimal schedule. Two experiments on improvement of quality and productivity are raised as the case studies and solved by GA. The case study shows the potentials of MES with this framework. The system can streamline the steelmaking-continuous casting plant from designing an effective schedule under the uncertain disruption to execution of the process.

Future works:

- Since the extension time or the maintenance time and their uncertainty affect on the stability, the relation of them should be studied.
- In part of the uncertainty assessment, other clustering (e.g., Fuzzy clustering) should be studied.
- The uncertainty should be investigated in the part of plan execution such as in spray cooling system.
- Other significant quality criteria in the spray cooling optimization should be studied.
- Other processes for the plan execution should be incorporated.
- Extension of the MES framework to related industry should be studied.

REFERENCES

Adolfsson, J.K., Ng, A. and Moore, P.R., 2000, "Modular Machine System Design using Graphical Simulation", **33rd CIRP International Seminar on Manufacturing Systems**, pp. 335-340.

Akturk, M. S. and Gorgulu, E., 1999. "Match-up Scheduling under a Machine Breakdown", **European Journal of Operational Research**, Vol. 122, pp. 81-97.

Al-Hinai, N. and ElMekkawy, T.Y., 2011, "Robust and Stable Flexible Job Shop Scheduling with Random Machine Breakdowns using a Hybrid Genetic Algorithm", **International Journal of Production Economics**, Vol. 132, pp. 279-291.

Atighehchian, A., Bijari, M. and Tarkesh, H., 2009. "A Novel Hybrid Algorithm for Scheduling Steel-Making Continuous Casting Production", **Computer & Operations Research,** Vol. 36, pp. 2450-2461.

Aufenanger, M., Varnholt, H. and Dangelmaier, W., 2009, "Adaptive Flow Control in Flexible Flow Shop Production System-A Knowledge-Based Approach, **Proceedings of the 2009 Winter Simulation Conference**, pp. 2164-2175.

Aytag, H., Lawley, M. A., Mckay, K., Mohan, S. and Uzsoy, R., 2005, "Executing Production Schedules in the Face of Uncertainties: A Review and Some Future Directions", **European Journal of Operational Research**, Vol. 161, pp. 86-110.

Azadeh, A. and Maghsoudi, A., 2010, "Optimization of Production Systems Through Integration of Computer Simulation, Design of Experiment, and Tabu Search: The Case of a Large Steelmaking Workshop", **The International Journal of Advanced Manufacturing Technology**, Vol. 48, No. 5-8, pp. 785-800.

Bayer, F., Strasser, J. and Trenkler, H., 2000, "The New Integrated Flat-Steel Production Facility of HADEED", **La Revue de Métallurgie-CIT**, Saudi Arabia, pp. 1371-1389.

Bellabdaoui, A., Fiordaliso, A. and Teghem, J., 2005, "A Heuristic Algorithm for Scheduling the Steel Making Continuous Casting Process", **Pacific Journal Optimization**, Vol. 1, pp. 1-18.

Bellabdaoui, A., Teghem, J., 2006, "A Mixed-Integer Linear Programming Model for the Continuous Casting Planning", **International Journal of Production Economics**, Vol. 104, pp. 260-270.

Beyer, H.G., and Sendhoff, B., 2006, "Robust Optimization-A Comprehensive Survey", **Computer Methods in Applied Mechanics and Engineering**, Vol. 196, pp. 3190--3218.

Bierwirth, C., and Mattfeld, D. C., 1999, "Production Scheduling and Rescheduling with Genetic Algorithms", **Evolutionary Computation**, Vol. 7, No. 1, pp. 1-17.

Blanc, P., Demongodin, I. and Castagna, P., 2008, "A Holonic Approach for Manufacturing Execution System Design: An Industrial Application", **Engineering Applications of Artificial Intelligence**, Vol. 21, No. 3, pp. 71-78.

Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G. and Weglarz., J., 2007, **Handbook of Scheduling from Theory to Applications,** Heidelberg: Springer, Berlin, pp. 291-311.

Boyes,W., 2002, "The Market for Control System Integrators", **White Paper of the Control System Integrators Association (CSIA)**, 25 March, pp. 1-27.

Brimacombe, J. K., Argarwal, P. K., Baptista, L. A., Hobbins, S. and Prabhakar, B., 1980, "Spray Cooling in the Continuous Casting of Steel", **Steelmaking Proceedings of the NOH-BOS Conference,** pp. 109-123.

Brucker, P., 2007. **Scheduling algorithms**, 5th ed., Springer, Berlin, pp. 1-10.

Buranathiti, T., Cao, J., Chen, W., Baghdasaryan, L. and Xia, Z. C., 2006, "Approach for Model Validation: Methodology and Illustration on a Sheet Metal Flanging Process", **Journal of Manufacturing Science and Engineering**, Vol. 128, pp. 588-597.

Bürvenich, H. P., 1999, "CSP Sequence Planning and Optimization" **Metals, Mining & More**, Vol. 2, pp. 4-5.

Carrasco, J. A. and Paljakka, M., 2004, **Simulation Aids in the Automation of Industrial Processes**, White Paper of VTT Industrial System, pp. 1-13.

Chang, S. Y., Chang, M.-R. and Hong, Y., 2000, "A Lot Grouping Algorithm for a Continuous Slab Caster in an Integrated Steel Mill", **Production Planning & Control**, Vol. 11, pp. 363-368.

Chen, W., Wiecek, M. and Zhang, J., 1999, "Quality Utility – a Compromise Programming Approach to Robust Design", **ASME Journal of Mechanical Design**, Vol. 121, No. 4, pp.179-187.

Cheng, T. C. E. and Sin, C. C. S., 1990, "A State-of-the-Art Review of Parallel-Machine Scheduling Research", **European Journal of Operational Research**, Vol. 47, pp. 271-292.

Choi, S.H., and Wang, K., 2012, "Flexible Flow Shop Scheduling with Stochastic Processing Time: A Decomposition-based Approach", **Computers & Industrial Engineering**, Vol. 63, pp. 362--373.

Cho, K. H. and Kim, B. M., 2008, "Numerical Analysis of Secondary Cooling in Continuous Slab Casting", **Journal of Materials Science and Technology**, Vol. 24, pp. 389-390.

Cowling, P. and Johansson, M., 2002. "Using Real Time Information for Effective Dynamic Scheduling", **European Journal of Operational Research**, Vol. 139, pp. 230–244.

Cowling, P. I., Ouelhadj, D. and Petrovic, S., 2004, "Dynamic Scheduling of Steel Casting and Milling using Multi-Agents", **Production Planning & Control**, Vol. 15, No. 2, pp. 178-188.

Defang, L., Liu, L., Zhu, W. and Rong, G., 2008, "Material-flow Modeling Technology and Its Application in Manufacturing Execution Systems of Petrochemical Industry", **Chinese Journal of Chemical Engineering**, Vol. 16, No. 1, pp. 71-78.

Demuth, H. and Beale, M., 2001, **Neural Network Toolbox User's Guide**, The Math Works, Inc., pp. 5-15.

Douglas, G.W., 1998, Integrating Simulation based Scheduling with MES in Semi-Conductor Fab, **Winter Simulation Conference**, pp.1713-1715.

Dutta, G. and Fourer, R., 2001, "A Survey of Mathematical Programming Applications in Integrated Steel Plants". **Manufacturing & Service Operations Management**, Vol. 3, No. 4, pp. 387-400.

Du, X., and Chen, W., 2000, "Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design", **ASME Journal of Mechanical Design**, Vol. 122, No. 2, pp.385-394.

Feng, S. C., 2000, "Manufacturing Planning and Execution Software Interfaces", **Journal of Manufacturing Systems**, pp. 1-17.

Fillipic, B. and Laitinen, E., 2005, "Model-Based Tuning of Process Parameters for Steady-State Steel Casting". **Informatic**, Vol. 29, pp. 491-496.

Goren, S. and Sabuncuolu, I., 2008, "Robustness and Stability Measures for Scheduling: Single-Machine Environment", **IIE Transactions**, Vol. 40, No. 1, pp.66-83.

Gravel, M., Price, W. L. and Gagne, C., 2002, "Scheduling Continuous Casting of Aluminum using a Multiple Objective Ant Colony Optimization Metaheuristic", **European Journal of Operational Research**, Vol. 143, pp. 218-229.

Grefenstette, J.J., 1986, "Optimization of Control Parameters for Genetic Algorithms", **IEEE Trans Systems, Man, and Cybernetics**, SMC-16, No. 1, pp. 122--128.

Guo, B. and Nonaka, Y., 1999, "Rescheduling and Optimization of Scheduling Considering Machine Failures", **Journal of Production Economics**, Vol. 60, pp. 503-513.

Guo, D. and Li, T., 2007, "Rescheduling algorithm for steelmaking-continuous casting", **Proceedings of 2nd IEEE Conference on Industrial Electronics and Applications,** pp. 1421-1425.

Hardin, R. A., Liu, K., Kapoor, A. and Beckermann, C., 2003, "A transient simulation and dynamic spray cooling control model for continuous steel casting", **Metallurgical and Materials Transactions B**, Vol. 34B, pp. 297-306.

Holland, J. H., 1975, **Adaptation in Natural and Artificial Systems**, University of Michigan Press, Ann Arbor, pp. 1-96.

Hopkomp, S. J., Mockus, L. and Reklaitis, G. V., 1997, "Robust Scheduling with Processing Time Uncertainty", **Computers & Chemical Engineering**, Vol. 21, pp. s1055-s1060.

Hwa, G. P., Jong, M.B. and Sang, B. P., 1999, "A Development of Object-Oriented Simulator for Manufacturing Execution Systems", **Computer & Industrial Engineering**, Vol. 37, pp. 239-242.

Ishiguro, M. and Itaoka, I., 1974, **Tetsu-to-Hagane**, vol. 60, pp. 63-65.

Jensen, M. T., 2001, **Robust and flexible scheduling with evolutionary computation**. Doctoral Dissertation, Faculty of Science, University of Aarhus, Denmark, pp. 151-194.

Jian, W., Xue, Y. C. and Yang, Q. W., 2004, "Optimum Cast Plan for Steelmaking-Continuous Casting Production Scheduling using Artificial Fish Swarm Optimization Algorithm", **3rd International Conference on Machine Learning and Cybernetics**, pp. 2339-2341.

Jin, R., Du, X. and Chen, W, 2003, "The Use of Metamodeling Techniques for Optimization under Uncertainty", **Structural and Multidisciplinary Optimization**, Vol. 25, No. 2, pp. 99-116.

Jung, J.R. and Yum, B.J., 2011, "Artificial Neural Network based Approach for Dynamic Parameter Design", **Expert Systems with Applications**, Vol. 38, pp. 504–510.

Kim, W. C. And Mauborgne, R., 2000, **Knowing a Winning Business Idea When You See One**. Harvard Business School Press, Boston.

Kletti, J., 2007. **Manufacturing Execution System-MES**, Springer, pp. 1-13.

Kulkarni, M. S. and Babu, A. S., 2005, "Managing Quality in Continuous Casting Process using Product Quality Model and Simulated Annealing", **Journal of Materials Processing Technology**, Vol. 166, pp. 294-306.

Kyparisis, G. J. and Koulamas. C., 2006, "Flexible flow shop scheduling with uniform parallel machines", **European Journal of Operational Research**, Vol. 168, No. 3, pp. 985–997.

Lee, H. S., Murthy, S. S., Haider, S. W. and Morse, D. V., 1996, "Primary Production Scheduling at Steelmaking Industries", **IBM Journal of Research Development**, Vol. 40, pp. 231-352.

Lee, K., Chang, S. Y. and Hong, Y., 2004, "Continuous Slab Caster Scheduling and Interval Graphs", **Production Planning & Control**, Vol. 15, pp. 495-501.

Leitão, P., Colombo, A. W. and Restivo, F. J., 2005, "ADACOR: A Collaborative Production Automation and Control Architecture", **IEEE Intelligent System**, Vol. 20, No. 1, pp. 58-66.

Li, Y. C. E. and Shaw, W. H., 1998, "Simulation modeling of a dynamic job shop rescheduling with machine availability constraints", **Computers & Industrial Engineering**, Vol. 35, pp. 117-120.

Liu, X., Bo, H., Ma, Y. and Meng, Q., 2007, "Research on Hybrid Distributed Manufacturing Execution System in Multi-Location Enterprises Environment", In **CSCW in Design III, LNCS 4402**, Shen, W. (Ed.), Springer-Verlag, pp. 247-256.

Luecke, R. and Katz, R., 2003, **Managing Creativity and Innovation**, Harvard Business School Press, Boston, pp. 25-32.

Luh, P. B. and Hoitomt, D. J., 1993, "Scheduling of manufacturing systems using the lagrangian relaxation technique", **IEEE Transactions on Automation Control**, Vol. 38, No., 7, pp. 1066-1079.

Man, K.F., Tang, K.S. and Kwong, S., 1999, **Genetic Algorithms: Concepts and Designs**, Springer-Verlag, London, pp. 26-33.

Martin, J., and Simpson, T., 2006, **A Monte Carlo Method for Reliability-based Design Optimization**, Tech. Rep., American Institute of Aeronautics and Astronautics AIAA.

Melouk, S. H., Freeman, N. K., Miller, D., Dunning, M., 2013, "Simulation Optimization-Based Decision Support Tool for Steel Manufacturing", **International Journal of Production Economics**, Vol. 141, pp. 269–276.

Meng, Y., Thomas, B. G., 2003, "Heat Transfer and Solidification Model of Continuous Slab Casting: CON1D," **Metallurgical and Materials Transactions B**, Vol. 34B, pp. 685-705.

Mezgar, I., Egresits, C.S. and Monostori, L., 1997, "Design and Real-Time Reconfiguration of Robust Manufacturing Systems by Using Design of Experiments and Artificial Neural Networks", **Computers in Industry**, Vol. 33, pp. 61–70.

Michael, M., 2000, "Introduction to Manufacturing Execution Systems", **MES Conference & Exposition**, Arizona, pp. 1-12.

Mizikar, E. A., 1967, "Mathematical Heat Transfer Model for Solidification of Continuously Cast Steel Slabs", **Transactions of the metallurgical Society of AIME**, Vol. 239, pp. 1747-1753.

Mönch, L, Rose, O. and Sturm, R., 2003, "A Simulation Framework for the Performance Assessment of Shop-floor Control Systems", **Simulation**, Vol. 79, No. 3, pp. 163-170.

Morel, G., Panetto, H., Zaremba, M. and Mayer, F., 2003, "Manufacturing Enterprise Control and Management System Engineering: Paradigms and Open Issues", **Annual Reviews in Control**, Vol. 27, pp. 199-209.

Murray, R. M., Astrom, K. J., Boyd, S. P., Brockett, R. W., and Stein, G.., 2003, "Future Directions in Control in an Information-Rich World", **IEEE Control Systems Magazine**, pp. 20-33.

Nagasaka Y., 1999. "Computer Simulation and Information Management Systems for Material Processing", **Intelligent Processing and Manufacturing of Materials**, 10 Jul 1999, Honolulu, pp. 85-90.

Namao, M., and Morishita, S., 1991, "Cooperative Scheduling and Its Application to Steelmaking Processes". **IEEE Transactions on Industry Electronic**, Vol. 38, No.2, pp. 150-155.

Nawaz, M., Emory, E. E. Jr. and Ham, I., 1983, "A Heuristic Algorithm for the m-Machine, n-Job Flowshop Sequencing Problem", **Omega, The International Journal of Management Science**, Vol. 11, No. 1, pp. 91-95.

Olofsgård, P., Ng, A.H.C., Moore, P., Pu, J., Wong, C.B. and De Vin, L.J., 2002, "Distributed Virtual Manufacturing for Development of Modular Machine Systems", **Journal of Advanced Manufacturing Systems**, Vol. 1, No 2, pp. 141-158.

Ouelhadj, D., 2003. **A multi-agent system for the integrated dynamic scheduling of steel production**. Doctoral Dissertation, University of Nottingham, United Kingdom, pp. 40-85.

Pacciarelli, D. and Pranzo, M., 2004, "Production Scheduling in a Steelmaking-Continuous Casting Plant", **Computers and Chemical Engineering**, Vol. 28, pp. 2823–2835.

Pfeiffer, A, Kadar, K. and Monostori, L., 2007, "Stability-Oriented Evaluation of Rescheduling Strategies by using Simulation", **Computers in Industry**, Vol. 58, pp. 630–643.

Ponsignon, T. and Mönch, L., 2014, "Simulation-Based Performance Assessment of Master Planning Approaches in Semiconductor Manufacturing, **Omega**, Vol. 46, pp. 21–35.

Preissl, H., Obermann, W., Hubner, N., Juza, P. and Schrack, P., 2001, "Advances and Future Aspects in Continuous Casting Automation", **STEEL TIMES**, pp. 98-100.

Qiu, R. G. and Zhou, M., 2004, "Mighty MESs: state-of-the-art and future manufacturing systems", **IEEE Robotics and Automation Magazine**, Vol. 11, No. 1, pp. 19-40.

Rangsaritratsamee, R., Ferrell, W. G. and Kurz, M. B., 2004, "Dynamic Rescheduling that Simultaneously Considers Efficiency and Stability", **Computers & Industrial Engineering**, Vol. 46, pp. 1–15.

Reeves, C. A., 1995, "A Genetic Algorithm for Flow Shop Scheduling", **Computers and Operation Research**, Vol. 22, pp. 5-13.

Rolón, M. and Martinez, E., 2012a, "Agent-based Modeling and Simulation of an Autonomic Manufacturing Execution System", **Computers in Industry**, Vol. 63, pp. 53-78.

Rousseeuw, P.J., 1987, “Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis”, **Computational and Applied Mathematics**, Vol. 20, pp. 53-–65.

Santos, C. A., Spim, A. and Garcia, A., 2003, “Mathematical Modeling and Optimization Strategies (Genetic Algorithm and Knowledge Base) Applied to the Continuous Casting of Steel”, **Engineering Applications of Artificial Intelligence**, Vol. 16, pp. 511-527.

SanZ, R., and Årzén, K.E., 2003, “Trends in Software and Control”, **IEEE Control Systems Magazine**, pp. 12-15.

Sethanan, K., 2001, **Scheduling flexible flowshops with sequence dependent setup times**, Doctoral Dissertation, College of Engineering and Mineral Resources, West Virginia University, pp. 7-9.

Shen, H., Hardin, R., Mackenzie, R. and Beckermann, C., 2002, “Simulation using realistic spray cooling for the continuous casting of muiti-componant steel”, **Journal of Materials Science and Technology**, Vol. 18, pp 311-314.

Siemens, 2006, **Why Integrate MES and ERP: Because you cannot afford not to**, Siemens Whitepaper, Siemens Energy & Automation Inc., pp. 1-8.

Simao, J.M., Stadzisz, P. C. and Morel, G., 2006, “Manufacturing Execution Systems for Customized Production”, **Journal of Materials Processing Technology**, Vol. 179, No.1-3, pp. 268-275.

Sandgren, E. and Cameron, T., 2002, “Robust Design Optimization of Structures through Consideration of Variation”, **Computer & Structures**, Vol. 80, pp. 1605--1613.

Santos, C. A., Fortaleza, E. L., Ferreira, C. R. F., Spim, J. A. and Garcia, A., 2005, “A solidification heat transfer model and neural network based algorithm applied to the continuous casting of steel billets and blooms”, **Modelling and Simulation in Materials Science and Engineering**, Vol. 13, pp. 1071–1087.

Santos, C. A., Spim, A. and Garcia, A., 2003, “Mathematical modeling and optimization strategies (genetic algorithm and knowledge base) applied to the continuous casting of steel”, **Engineering Applications of Artificial Intelligence**, Vol. 16, pp. 511-527.

Sargent, R. G., 1998, “Verification and validation of simulation models”, **Proceedings of the Winter Simulation Conference**, pp. 121-130.

Shaojian, Q., Hongyou, Y. and Kecum, Z., 2006, “A global optimization algorithm using linear relaxation”, **Applied Mathematics and Computation**, Vol. 178, pp. 510-518.

Spuy, D. D., Craig, I. K. and Pistorius, P. C., 1999, "An optimization procedure for the secondary cooling zone of a continuous billet caster", **The Journal of the South African Institute of Mining and Metallurgy**, pp. 49-54.

Sriskandarajah, C. and Sethi., S., 1989, "Scheduling Algorithms for Flexible Flowshops: Worst and Average Case Performance", **European Journal of Operational Research, Vol.** 43, No. 2, pp. 143–160.

Sunkara, R. and Rao, R., 2003, "Dynamic Scheduling I: Use of Discrete Event Simulation to Analyze Dispatch Policies of an Equipment Group in Semiconductor Feb", **Proceedings of the 2003 Winter Simulation Conference**, pp. 1474-1479.

Tang, L. X. and Wang, G., 2008, "Decision Support System for the Batching Problems of Steelmaking and Continuous-Casting Production", **Omega**, Vol. 36, pp. 976-991.

Tang, L. X., Luh, P. B., Liu, J. and Fang, L., 2000, "A Mathematical Programming Model for Scheduling Steel-Making-Continuous Casting Production", **European Journal of Operational Research**, Vol. 120, pp. 55-70.

Tang, L. X., Luh, P. B., Liu, J. and Fang, L., 2002, "Steelmaking process scheduling using Lagrangian relaxation", **International Journal of Production Research**, Vol. 40, pp. 55-70.

Taguchi, G., 1992, **Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream**, ASME Press, New York.

Valckenaers, P., Brussel, H. V., Verstraete, P., Germain, B. S. and Hadeli, 2007, "Schedule execution in autonomic manufacturing execution systems", **Journal of Manufacturing Systems**, Vol. 26, No. 2, pp. 75-84.

Vieira, G., Hermann, J. and Lin, E., 2000b, "Predicting the Performance of Rescheduling Strategies for Parallel Machine System", **Journal of Manufacturing System**, Vol. 19, No. 4, pp. 256-266.

Vieira, G., Herrmann, J. and Lin, E., 2003. Rescheduling Manufacturing System: A Framework of Strategies, Policies, and Methods, **Journal of Scheduling**, Vol. 6, pp. 39-62.

Wang, H., 2005, "Flexible flowshop Scheduling: Optimum, Heuristics and Artificial Intelligence Solutions", **Expert Systems**, Vol. 22, No. 2, pp. 78–85.

Wang, X., Yu, X., Zheng, B. and Chai, T., 2006, "Intelligent Scheduling System of steel making and continuous casting based on ERP/MES/PCS", **The Sixth World Congress on Intelligent Control and Automation**, pp. 7381-7384.

Worapradya, K., 2006, “Front-End of Development of Integrated Control System”, **Qualifying Examination Paper**, King Mongkut’s University of Technology Thonburi, 8 March 2006, pp. 1-56.

Xiong, J., Xing, L.N. and Chen, Y.W., Inpress, “Robust Scheduling for Multi-Objective Flexible Job-shop Problems with Random Machine Breakdowns”, **International Journal of Production Economics**.

Xue, Y., Yang Q. and Shao, H., 2004, “Optimum Cast Plan for Steelmaking-Continuous Casting Production Scheduling”, **IEEE International Conference on Control Applications**, pp. 1394-1397.

Yagmahan, B. and Yenisey, M. M.., 2008, “Ant colony optimization for multi-objective flow shop scheduling problem”, **Computers & Industrial Engineering**, Vol. 54, pp. 411-420.

Yan-hai, H., Junqi, Y., Fei-fan, Y. and Jun-he, Y., 2005, “Flow Shop Rescheduling Problem under Rush Order”, **Journal of Zheiang University SCIENCE**, Vol. 10, pp. 1040-1046.

Zhong, R.Y., Dai, Q.Y., Qu, T., Hu, G.J. and Huang G. Q., 2013, “RFID-Enabled Real-Time Manufacturing Execution System for Mass-Customization Production”, **Robotics and Computer Integrated Manufacturing**, Vol. 29, No. 2, pp. 283-292.

Zobel, C.W. and Keeling, K. B., 2008, “Neural Network-based Simulation Metamodels for Predicting Probability Distributions”, **Computers and Industrial Engineering**, Vol. 54, pp. 879--888.

APPENDIX A PRODUCTION DATA

A.1. Example of Raw Data for Processing Time Modeling

Heat	Grade	No.	EAF (Melting)		Delay	LHF (Refining)				Delay	Caster		
			Dept	Inter-Arr		Line	Arr	Dept	LHF-Time		Arr	Dept	CT time
19648	1008-MNDK2	1	12/12/2007 00:09	0:00	3:17	2	12/12/2007 3:26	12/12/2007 7:49	4:23	0:11	12/12/2007 8:00	12/12/2007 09:02	1:02
19649	1007DKGXA	2	12/12/2007 04:21	4:12	0:09	1	12/12/2007 4:30	12/12/2007 8:38	4:08	0:24	12/12/2007 9:02	12/12/2007 10:48	1:45
19650	1007DKGXA	3	12/12/2007 08:48	4:27	0:11	1	12/12/2007 8:59	12/12/2007 10:30	1:31	0:18	12/12/2007 10:48	12/12/2007 11:43	0:55
19651	1007DKGXA	4	12/12/2007 09:48	1:00	0:10	2	12/12/2007 9:58	12/12/2007 11:22	1:24	0:21	12/12/2007 11:43	12/12/2007 12:39	0:56
19652	1007DKGXA	5	12/12/2007 10:52	1:04	0:08	1	12/12/2007 11:00	12/12/2007 12:17	1:17	0:22	12/12/2007 12:39	12/12/2007 13:37	0:58
19653	1007DKGXA	6	12/12/2007 11:54	1:02	0:10	2	12/12/2007 12:04	12/12/2007 13:15	1:11	0:22	12/12/2007 13:37	12/12/2007 14:30	0:52
19654	1007DKGXA	7	12/12/2007 13:02	1:08	0:09	1	12/12/2007 13:11	12/12/2007 14:10	0:59	0:20	12/12/2007 14:30	12/12/2007 15:27	0:57
19655	1007DKGXA	8	12/12/2007 14:01	0:59	0:11	2	12/12/2007 14:12	12/12/2007 15:11	0:59	0:16	12/12/2007 15:27	12/12/2007 16:23	0:56
19656	1007DKGXA	9	12/12/2007 14:55	0:54	0:10	1	12/12/2007 15:05	12/12/2007 16:05	1:00	0:18	12/12/2007 16:23	12/12/2007 17:19	0:55
19657	1007DKGXA	10	12/12/2007 15:46	0:51	0:08	2	12/12/2007 15:54	12/12/2007 16:58	1:04	0:21	12/12/2007 17:19	12/12/2007 18:12	0:52
19658	1007DKGXA	11	12/12/2007 16:53	1:07	0:09	1	12/12/2007 17:02	12/12/2007 17:53	0:51	0:19	12/12/2007 18:12	12/12/2007 19:16	1:04
19659	1007DKGXA	12	12/12/2007 17:50	0:57	0:08	2	12/12/2007 17:58	12/12/2007 18:57	0:59	0:19	12/12/2007 19:16	12/12/2007 20:01	0:44
19660	1007DKGXA	1	12/12/2007 18:46	0:00	0:09	1	12/12/2007 18:55	12/12/2007 21:33	2:38	0:11	12/12/2007 21:44	12/12/2007 23:01	1:17
19661	1007DKGXA	2	12/12/2007 20:31	1:45	0:19	2	12/12/2007 20:50	12/12/2007 22:43	1:53	0:18	12/12/2007 23:01	12/13/2007 0:13	1:11
19662	1007DKGXA	3	12/12/2007 22:34	2:03	0:12	1	12/12/2007 22:46	12/12/2007 23:54	1:08	0:19	12/13/2007 0:13	12/13/2007 1:23	1:10
19663	1007DKGXA	4	12/12/2007 23:32	0:58	0:12	2	12/12/2007 23:44	12/13/2007 1:03	1:19	0:20	12/13/2007 1:23	12/13/2007 2:30	1:06
19664	1007DKGXA	5	12/13/2007 00:36	1:04	0:11	1	12/13/2007 0:47	12/13/2007 2:09	1:22	0:21	12/13/2007 2:30	12/13/2007 3:38	1:08
19665	1007DKGXA	6	12/13/2007 01:43	1:07	0:11	2	12/13/2007 1:54	12/13/2007 3:13	1:19	0:25	12/13/2007 3:38	12/13/2007 4:40	1:02
19666	1007DKGXA	7	12/13/2007 02:49	1:06	0:10	1	12/13/2007 2:59	12/13/2007 4:16	1:17	0:24	12/13/2007 4:40	12/13/2007 5:55	1:15
19676	1007DKGXA	1	12/14/2007 01:37	0:00	0:15	1	12/14/2007 1:52	12/14/2007 3:33	1:41	0:11	12/14/2007 3:44	12/14/2007 4:57	1:12
19677	1007DKGXA	2	12/14/2007 02:45	1:08	0:10	2	12/14/2007 2:55	12/14/2007 4:37	1:42	0:20	12/14/2007 4:57	12/14/2007 6:02	1:04
19678	1007DKGXA	3	12/14/2007 04:03	1:18	0:10	1	12/14/2007 4:13	12/14/2007 5:39	1:26	0:23	12/14/2007 6:02	12/14/2007 7:09	1:07
19679	1007DKGXA	4	12/14/2007 05:06	1:03	0:10	2	12/14/2007 5:16	12/14/2007 6:48	1:32	0:21	12/14/2007 7:09	12/14/2007 8:15	1:06
19680	1007DKGXA	5	12/14/2007 06:11	1:05	0:10	1	12/14/2007 6:21	12/14/2007 7:55	1:34	0:20	12/14/2007 8:15	12/14/2007 9:32	1:16
19681	1007DKGXA	6	12/14/2007 07:10	0:59	0:08	2	12/14/2007 7:18	12/14/2007 9:12	1:54	0:20	12/14/2007 9:32	12/14/2007 10:43	1:11
19682	1007DKGXA	7	12/14/2007 09:04	1:54	0:10	1	12/14/2007 9:14	12/14/2007 10:20	1:06	0:23	12/14/2007 10:43	12/14/2007 11:59	1:16
19683	1007DKGXA	8	12/14/2007 10:02	0:58	0:09	2	12/14/2007 10:11	12/14/2007 11:36	1:25	0:23	12/14/2007 11:59	12/14/2007 13:19	1:19
19684	1007DKGXA	9	12/14/2007 11:03	1:01	0:09	1	12/14/2007 11:12	12/14/2007 12:58	1:46	0:21	12/14/2007 13:19	12/14/2007 14:40	1:21
19685	1007DKGXA	10	12/14/2007 12:00	0:57	0:10	2	12/14/2007 12:10	12/14/2007 14:24	2:14	0:16	12/14/2007 14:40	12/14/2007 15:54	1:13
19686	1007DKGXA	11	12/14/2007 13:07	1:07	0:09	1	12/14/2007 13:16	12/14/2007 15:35	2:19	0:19	12/14/2007 15:54	12/14/2007 17:07	1:13
19687	1007DKGXA	12	12/14/2007 15:52	2:45	0:10	2	12/14/2007 16:02	12/14/2007 16:46	0:44	0:21	12/14/2007 17:07	12/14/2007 18:22	1:14
19688	1007DKGXA	13	12/14/2007 16:57	1:05	0:10	1	12/14/2007 17:07	12/14/2007 18:03	0:56	0:19	12/14/2007 18:22	12/14/2007 19:27	1:05
19689	1007DKGXA	14	12/14/2007 17:57	1:00	0:11	2	12/14/2007 18:08	12/14/2007 19:12	1:04	0:15	12/14/2007 19:27	12/14/2007 20:30	1:02
19690	1007DKGXA	15	12/14/2007 18:50	0:53	0:10	1	12/14/2007 19:00	12/14/2007 20:06	1:06	0:24	12/14/2007 20:30	12/14/2007 21:35	1:04
19691	1007DKGXA	16	12/14/2007 20:00	1:10	0:11	2	12/14/2007 20:11	12/14/2007 21:11	1:00	0:24	12/14/2007 21:35	12/14/2007 22:51	1:15
19692	1007DKGXA	1	12/14/2007 21:16	0:00	0:10	1	12/14/2007 21:26	12/14/2007 23:09	1:43	0:12	12/14/2007 23:21	12/15/2007 0:23	1:02
19693	1007DKGXA	2	12/14/2007 22:13	0:57	0:11	2	12/14/2007 22:24	12/15/2007 0:05	1:41	0:18	12/15/2007 0:23	12/15/2007 1:23	0:59
19694	1007DKGXA	3	12/14/2007 23:10	0:57	0:11	1	12/14/2007 23:21	12/15/2007 1:10	1:49	0:13	12/15/2007 1:23	12/15/2007 2:22	0:58
19695	1007DKGXA	4	12/15/2007 00:11	1:01	0:10	2	12/15/2007 0:21	12/15/2007 2:02	1:41	0:20	12/15/2007 2:22	12/15/2007 3:14	0:52
19696	1007DKGXA	5	12/15/2007 01:16	1:05	0:09	1	12/15/2007 1:25	12/15/2007 2:57	1:32	0:17	12/15/2007 3:14	12/15/2007 4:07	0:52
19697	1007DKGXA	6	12/15/2007 02:19	1:03	0:11	2	12/15/2007 2:30	12/15/2007 3:46	1:16	0:21	12/15/2007 4:07	12/15/2007 5:04	0:57
19698	1007DKGXA	7	12/15/2007 03:16	0:57	0:10	1	12/15/2007 3:26	12/15/2007 4:43	1:17	0:21	12/15/2007 5:04	12/15/2007 6:02	0:57
19699	1007DKGXA	8	12/15/2007 04:16	1:00	0:10	2	12/15/2007 4:26	12/15/2007 5:45	1:19	0:17	12/15/2007 6:02	12/15/2007 6:58	0:55
19700	1007DKGXA	9	12/15/2007 05:37	1:21	0:09	1	12/15/2007 5:46	12/15/2007 6:40	0:54	0:18	12/15/2007 6:58	12/15/2007 7:52	0:54
19701	1007DKGXA	10	12/15/2007 06:34	0:57	0:10	2	12/15/2007 6:44	12/15/2007 7:29	0:45	0:23	12/15/2007 7:52	12/15/2007 8:50	0:58
19702	1007DKGXA	11	12/15/2007 07:41	1:07	0:11	1	12/15/2007 7:52	12/15/2007 8:33	0:41	0:17	12/15/2007 8:50	12/15/2007 9:49	0:58
19703	1007DKGXA	12	12/15/2007 08:38	0:57	0:09	2	12/15/2007 8:47	12/15/2007 9:30	0:43	0:19	12/15/2007 9:49	12/15/2007 10:43	0:54
19704	1007DKGXA	13	12/15/2007 09:32	0:54	0:10	1	12/15/2007 9:42	12/15/2007 10:25	0:43	0:18	12/15/2007 10:43	12/15/2007 11:37	0:53
19705	1007DKGXA	14	12/15/2007 10:25	0:53	0:15	1	12/15/2007 10:40	12/15/2007 11:17	0:37	0:20	12/15/2007 11:37	12/15/2007 12:44	1:06
19707	1007DKGXA	1	12/15/2007 13:11	0:00	0:16	2	12/15/2007 13:27	12/15/2007 14:41	1:14	0:11	12/15/2007 14:52	12/15/2007 15:59	1:06
19708	1007DKGXA	2	12/15/2007 13:55	0:44	0:10	1	12/15/2007 14:05	12/15/2007 15:43	1:38	0:16	12/15/2007 15:59	12/15/2007 17:07	1:07
19709	1007DKGXA	3	12/15/2007 14:48	0:53	0:11	2	12/15/2007 14:59	12/15/2007 16:43	1:44	0:24	12/15/2007 17:07	12/15/2007 18:17	1:10
19710	1007DKGXA	4	12/15/2007 16:14	1:26	0:10	1	12/15/2007 16:24	12/15/2007 17:53	1:29	0:24	12/15/2007 18:17	12/15/2007 19:22	1:05
19711	1007DKGXA	5	12/15/2007 17:14	1:00	0:11	2	12/15/2007 17:25	12/15/2007 19:01	1:36	0:21	12/15/2007 19:22	12/15/2007 20:23	1:00
19712	1007DKGXA	6	12/15/2007 18:21	1:07	0:10	1	12/15/2007 18:31	12/15/2007 20:01	1:30	0:22	12/15/2007 20:23	12/15/2007 21:38	1:14
19713	1007DKGXA	7	12/15/2007 19:34	1:13	0:10	2	12/15/2007 19:44	12/15/2007 21:15	1:31	0:23	12/15/2007 21:38	12/15/2007 22:45	1:06
19714	1007DKGXA	8	12/15/2007 20:30	0:56	0:10	1	12/15/2007 20:40	12/15/2007 22:25	1:45	0:20	12/15/2007 22:45	12/15/2007 23:52	1:07
19715	1007DKGXA	9	12/15/2007 21:33	1:03	0:11	2	12/15/2007 21:44	12/15/2007 23:30	1:46	0:22	12/15/2007 23:52	12/16/2007 0:53	1:00
19716	1007DKGXA	10	12/15/2007 22:34	1:01	0:10	1	12/15/2007 22:44	12/16/2007 0:32	1:48	0:21	12/16/2007 0:53	12/16/2007 1:58	1:04
19717	1007DKGXA	11	12/15/2007 23:34	1:00	0:09	2	12/15/2007 23:43	12/16/2007 1:45	2:02	0:13	12/16/2007 1:58	12/16/2007 3:08	1:10
19718	1007DKGXA	12	12/16/2007 00:49	1:15	0:09	1	12/16/2007 0:58	12/16/2007 2:48	1:50	0:20	12/16/2007 3:08	12/16/2007 4:20	1:11
19719	1007DKGXA	13	12/16/2007 01:53	1:04	0:10	2	12/16/2007 2:03	12/16/2007 3:58	1:55	0:22	12/16/2007 4:20	12/16/2007 5:34	1:13
19720	1007DKGXA	14	12/16/2007 02:52	0:59	0:11	1	12/16/2007 3:03	12/16/2007 5:14	2:11	0:20	12/16/2007 5:34	12/16/2007 6:36	1:02
19721	1007DKGXA	15	12/16/2007 04:14	1:22	0:10	2	12/16/2007 4:24	12/16/2007 6:18	1:54	0:18	12/16/2007 6:36	12/16/2007 7:45	1:08
19722	1007DKGXA	16	12/16/2007 05:20	1:06	0:09	1	12/16/2007 5:29	12/16/2007 7:28	1:59	0:17	12/16/2007 7:45	12/16/2007 9:08	1:23
19734	1007DKGXA	1	12/17/2007 00:28	0:00	0:10	2	12/17/2007 0:38	12/17/2007 4:24	3:46	0:12	12/17/2007 4:36	12/17/2007 5:49	1:13
19735	1007DKGXA	2	12/17/2007 01:29	1:01	0:10	1	12/17/2007 1:39	12/17/2007 5:29	3:50	0:20	12/17/2007 5:49	12/17/2007 6:52	1:03
19736	1007DKGXA	3	12/17/2007 03:00	1:31	0:26	1	12/17/2007 3:26	12/17/2007 6:32	3:06	0:20	12/17/2007 6:52	12/17/2007 7:52	1:00
19737	1007DKGXA	4	12/17/2007 06:12	3:12	0:09	2	12/17/2007 6:21	12/17/2007 7:34	1:13	0:18	12/17/2007 7:52	12/17/2007 9:02	1:09
19738	1007DKGXA	5	12/17/2007 07:14	1:02	0:09	1	12/17/2007 7:23	12/17/2007 8:38	1:15	0:24	12/17/2007 9:02	12/17/2007 10:10	1:08
19739	1007DKGXA	6	12/17/2007 08:22	1:08	0:10	2	12/17/2007 8:32	12/17/2007 9:48	1:16	0:22	12/17/2007 10:10	12/17/2007 11:17	1:07
19740	1007DKGXA	7	12/17/2007 09:20	0:58	0:12	1	12/17/2007 9:32	12/17/2007 10:55	1:23	0:22	12/17/2007 11:17	12/17/2007 12:19	1:02
19741	1007DKGXA	8	12/17/2007 10:18	0:58	0:10	2	12/17/2007 10:28	12/17/2007 11:59	1:31	0:20	12/17/2007 12:19	12/17/2007 13:27	1:07
19742	1007DKGXA	9	12/17/2007 11:11	0:53	0:09	1	12/17/2007 11:20	12/17/2007 13:09	1:49	0:18	12/17/2007 13:27	12/17/2007 14:38	1:10
19743	1007DKGXA	10	12/17/2007 12:32	1:21	0:09	2	12/17/2007 12:41	12/17/2007 14:17	1:36	0:21	12/17/2007 14:38	12/17/2007 16:01	1:23
19744	1007DKGXA	11	12/17/2007 13:21	0:49	0:10	1	12/17/2007 13:31	12/17/2007 15:42	2:11	0:19	12/17/2007 16:01	12/17/2007 17:14	1:12
19745	1007DKGXA	12	12/17/2007 14:21	1:00	0:12	2	12/17/2007 14:33	12/17/2007 16:52	2:19	0:22	12/17/2007 17:14	12/17/2007 18:25	1:10
19746	1007DKGXA	13	12/17/2007 16:11	1:50	0:08	1	12/17/2007 16:19	12/17/2007 18:04	1:45	0:21	12/17/2007 18:25	12/17/2007 19:35	1:10
19747	1007DKGXA	14	12/17/2007 17:14	1:03	0:09	2	12/17/2007 17:23	12/17/2007 19:12	1:49	0:23	12/17/2007 19:35	12/17/2007 20:45	1:10
19748	1007DKGXA	15	12/17/2007 18:20	1:06	0:08	1	12/17/2007 18:28	12/17/2007 20:25	1:57	0:20	12/17/2007 20:45	12/17/2007 21:57	1:11
19749	1007DKGXA	16	12/17/2007 19:20	1:00	0:11	2	12/17/2007 19:31	12/17/2007 21:34	2:03	0:23	12/17/2007 21:57	12/17/2007 23:18	1:20
19766	1007DKGXA	1	12/18/2007 15:41	0:00	0:09	1	12/18/2007 15:50	12/18/2007 18:36	2:46	0:13	12/18/2007 18:49	12/18/2007 20:00	1:11
19765	1007DKGXA	2	12/18/2007 16:50	1:09	0:10	2	12/18/2007 17:00	12/18/2007 19:40	2:40	0:20	12/18/2007 20:00	12/18/2007 21:11	1:11
19767	1007DKGXA	3	12/18/2007 19:17	2:27	0:09	1	12/18/2007 19:26	12/18/2007 20:55	1:29	0:16	12/18/2007 21:11	12/18/2007 22:30	1:19
19768	1007DKGXA	4	12/18/2007 20:27	1:10	0:09	2	12/18/2007 20:36	12/18/2007 22:09	1:33	0:21	12/18/2007 22:30	12/18/2007 23:41	1:10
19769	1007DKGXA	5	12/18/2007 21:26	0:59	0:09	1	12/18/2007 21:35	12/18/2007 23:21	1:46	0:20	12/18/2007 23:41	12/19/2007 0:54	1:13
19770	1007DKGXA	6	12/18/2007 22:41	1:15	0:10	2	12/18/2007 22:51	12/19/2007 0:39	1:48	0:15	12/19/2007 0:54	12/19/2007 2:04	1:09
19771	1007DKGXA	7	12/18/2007 23:45	1:04	0:09	1	12/18/2007 23:54	12/19/2007 1:42	1:48	0:22	12/19/2007 2:04	12/19/2007 3:26	1:21
19772	1007DKGXA	8	12/19/2007 01:02	1:17	0:09	2	12/19/2007 1:11	12/19/2007 3:06	1:55	0:20	12/19/2007 3:26	12/19/2007 4:33	1:07
19773	1007DKGXA	9	12/19/2007 02:08	1:06	0:08	1	12/19/2007 2:16	12/19/2007 4:11	1:55	0:22	12/19/2007 4:33	12/19/2007 5:48	1:14
19774	1007DKGXA	10	12/19/2007 03:26	1:18	0:09	2	12/19/2007 3:35	12/19/2007 5:29	1:54	0:19	12/19/2007 5:48	12/19/2007 7:03	1:15
19775	1007DKGXA	11	12/19/2007 05:03	1:37	0:09	1	12/19/2007 5:12	12/19/2007 6:41	1:29	0:22	12/19/2007 7:03	12/19/2007 8:11	1:07
19776	1007DKGXA	12	12/19/2007 06:02	0:59	0:10	2	12/19/2007 6:12	12/19/2007 7:53	1:41	0:18	12/19/2007 8:11	12/19/2007 9:15	1:04
19777	1007DKGXA	13	12/19/2007 07:20	1:18	0:10	1	12/19/2007 7:30	12/19/2007 8:51	1:21	0:24	12/19/2007 9:15	12/19/2007 10:18	1:03
19778	1007DKGXA	14	12/19/2007 08:21	1:01	0:09	2	12/19/2007 8:30	12/19/2007 9:57	1:27	0:21	12/19/2007 10:18	12/19/2007 11:25	1:06
19779	1007DKGXA	15	12/19/2007 09:16	0:55	0:09	1	12/19/2007 9:25	12/19/2007 11:04	1:39	0:21	12/19/2007 11:25	12/19/2007 12:27	1:02
19780	1007DKGXA	16	12/19/2007 10:14	0:58	0:09	2	12/19/2007 10:23	12/19/2007 12:06	1:43	0:21	12/19/2007 12:27	12/19/2007 13:38	1:11

19784	1007DKGXA	1	12/19/2007 15:45	0:00	0:11	1	12/19/2007 15:56	12/19/2007 21:04	5:08	0:12	12/19/2007 21:16	12/19/2007 22:25	1:09
19785	1007DKGXA	2	12/19/2007 16:43	0:58	0:12	1	12/19/2007 16:55	12/19/2007 22:08	5:13	0:17	12/19/2007 22:25	12/19/2007 23:34	1:08
19786	1007DKGXA	3	12/19/2007 17:18	0:35	0:25	2	12/19/2007 17:43	12/19/2007 23:20	5:37	0:14	12/19/2007 23:34	12/20/2007 0:33	0:59
19787	1007DKGXA	4	12/19/2007 22:42	5:24	0:11	1	12/19/2007 22:53	12/20/2007 0:15	1:22	0:18	12/20/2007 0:33	12/20/2007 1:36	1:02
19788	1007DKGXA	5	12/19/2007 23:50	1:08	0:11	2	12/20/2007 0:01	12/20/2007 1:13	1:12	0:23	12/20/2007 1:36	12/20/2007 2:38	1:02
19789	1007DKGXA	6	12/20/2007 00:53	1:03	0:11	1	12/20/2007 1:04	12/20/2007 2:24	1:20	0:14	12/20/2007 2:38	12/20/2007 3:42	1:03
19790	1007DKGXA	7	12/20/2007 01:52	0:59	0:11	2	12/20/2007 2:03	12/20/2007 3:22	1:19	0:20	12/20/2007 3:42	12/20/2007 4:52	1:10
19791	1007DKGXA	8	12/20/2007 02:47	0:55	0:10	1	12/20/2007 2:57	12/20/2007 4:30	1:33	0:22	12/20/2007 4:52	12/20/2007 5:57	1:05
19792	1007DKGXA	9	12/20/2007 03:43	0:56	0:10	2	12/20/2007 3:53	12/20/2007 5:36	1:43	0:21	12/20/2007 5:57	12/20/2007 7:02	1:04
19793	1007DKGXA	10	12/20/2007 04:58	1:15	0:11	1	12/20/2007 5:09	12/20/2007 6:43	1:34	0:19	12/20/2007 7:02	12/20/2007 8:14	1:12
19794	1007DKGXA	11	12/20/2007 06:13	1:15	0:11	2	12/20/2007 6:24	12/20/2007 7:45	1:21	0:29	12/20/2007 8:14	12/20/2007 9:22	1:08
19795	1007DKGXA	12	12/20/2007 07:11	0:58	0:10	1	12/20/2007 7:21	12/20/2007 9:02	1:41	0:20	12/20/2007 9:22	12/20/2007 10:24	1:01
19796	1007DKGXA	13	12/20/2007 08:00	0:49	0:10	2	12/20/2007 8:10	12/20/2007 10:03	1:53	0:21	12/20/2007 10:24	12/20/2007 11:23	0:59
19797	1007DKGXA	14	12/20/2007 09:28	1:28	0:11	1	12/20/2007 9:39	12/20/2007 10:58	1:19	0:25	12/20/2007 11:23	12/20/2007 12:30	1:07
19798	1007DKGXA	15	12/20/2007 10:42	1:14	0:11	2	12/20/2007 10:53	12/20/2007 12:13	1:20	0:17	12/20/2007 12:30	12/20/2007 13:32	1:01
19799	1007DKGXA	16	12/20/2007 11:23	0:41	0:11	1	12/20/2007 11:34	12/20/2007 13:08	1:34	0:24	12/20/2007 13:32	12/20/2007 14:42	1:09
19800	1007DKGXA	1	12/20/2007 13:48	0:00	0:09	2	12/20/2007 13:57	12/20/2007 15:01	1:04	0:14	12/20/2007 15:15	12/20/2007 16:29	1:13
19801	1007DKGXA	2	12/20/2007 14:47	0:59	0:10	1	12/20/2007 14:57	12/20/2007 16:08	1:11	0:21	12/20/2007 16:29	12/20/2007 17:41	1:12
19802	1007DKGXA	3	12/20/2007 15:43	0:56	0:10	2	12/20/2007 15:53	12/20/2007 17:21	1:28	0:20	12/20/2007 17:41	12/20/2007 18:42	1:01
19803	1007DKGXA	4	12/20/2007 16:56	1:13	0:10	1	12/20/2007 17:06	12/20/2007 18:24	1:18	0:18	12/20/2007 18:42	12/20/2007 19:50	1:07
19804	1007DKGXA	5	12/20/2007 18:15	1:19	0:09	2	12/20/2007 18:24	12/20/2007 19:33	1:09	0:17	12/20/2007 19:50	12/20/2007 20:52	1:02
19805	1007DKGXA	6	12/20/2007 19:10	0:55	0:10	1	12/20/2007 19:20	12/20/2007 20:33	1:13	0:19	12/20/2007 20:52	12/20/2007 21:55	1:02
19806	1007DKGXA	7	12/20/2007 20:12	1:02	0:10	2	12/20/2007 20:22	12/20/2007 21:34	1:12	0:21	12/20/2007 21:55	12/20/2007 22:58	1:02
19807	1007DKGXA	8	12/20/2007 21:12	1:00	0:08	1	12/20/2007 21:20	12/20/2007 22:39	1:19	0:19	12/20/2007 22:58	12/21/2007 0:05	1:07
19808	1007DKGXA	9	12/20/2007 22:11	0:59	0:08	2	12/20/2007 22:19	12/20/2007 23:46	1:27	0:19	12/21/2007 0:05	12/21/2007 1:13	1:08
19809	1007DKGXA	10	12/20/2007 23:12	1:01	0:09	1	12/20/2007 23:21	12/21/2007 0:52	1:31	0:21	12/21/2007 1:13	12/21/2007 2:28	1:14
19810	1007DKGXA	11	12/21/2007 00:15	1:03	0:19	2	12/21/2007 0:34	12/21/2007 2:07	1:33	0:21	12/21/2007 2:28	12/21/2007 3:36	1:08
19811	1007DKGXA	12	12/21/2007 01:12	0:57	0:09	1	12/21/2007 1:21	12/21/2007 3:18	1:57	0:18	12/21/2007 3:36	12/21/2007 4:41	1:04
19812	1007DKGXA	13	12/21/2007 02:08	0:56	0:17	2	12/21/2007 2:25	12/21/2007 4:18	1:53	0:23	12/21/2007 4:41	12/21/2007 5:46	1:05
19813	1007DKGXA	14	12/21/2007 03:57	1:49	0:09	1	12/21/2007 4:06	12/21/2007 5:26	1:20	0:20	12/21/2007 5:46	12/21/2007 6:51	1:05
19814	1007DKGXA	15	12/21/2007 05:03	1:06	0:08	2	12/21/2007 5:11	12/21/2007 6:35	1:24	0:16	12/21/2007 6:51	12/21/2007 8:04	1:12
19815	1007DKGXA	16	12/21/2007 06:00	0:57	0:09	1	12/21/2007 6:09	12/21/2007 7:47	1:38	0:17	12/21/2007 8:04	12/21/2007 9:22	1:18
19816	1007DKGXA	1	12/21/2007 07:50	1:50	0:09	2	12/21/2007 7:59	12/21/2007 9:41	1:42	0:10	12/21/2007 9:51	12/21/2007 11:06	1:15
19817	1007DKGXA	2	12/21/2007 08:53	1:03	0:09	1	12/21/2007 9:02	12/21/2007 10:48	1:46	0:18	12/21/2007 11:06	12/21/2007 12:12	1:05
19818	1007DKGXA	3	12/21/2007 09:46	0:53	0:09	2	12/21/2007 9:55	12/21/2007 11:50	1:55	0:22	12/21/2007 12:12	12/21/2007 13:18	1:05
19819	1007DKGXA	4	12/21/2007 11:23	1:37	0:10	1	12/21/2007 11:33	12/21/2007 12:57	1:24	0:21	12/21/2007 13:18	12/21/2007 14:22	1:04
19820	1007DKGXA	5	12/21/2007 12:17	0:54	0:09	2	12/21/2007 12:26	12/21/2007 14:02	1:36	0:20	12/21/2007 14:22	12/21/2007 15:24	1:01
19821	1007DKGXA	6	12/21/2007 13:07	0:50	0:10	1	12/21/2007 13:17	12/21/2007 15:02	1:45	0:22	12/21/2007 15:24	12/21/2007 16:25	1:01
19822	1007DKGXA	7	12/21/2007 14:26	1:19	0:17	2	12/21/2007 14:43	12/21/2007 16:02	1:19	0:23	12/21/2007 16:25	12/21/2007 17:33	1:07
19823	1007DKGXA	8	12/21/2007 15:31	1:05	0:09	1	12/21/2007 15:40	12/21/2007 17:13	1:33	0:20	12/21/2007 17:33	12/21/2007 18:39	1:05
19824	1007DKGXA	9	12/21/2007 16:24	0:53	0:16	2	12/21/2007 16:40	12/21/2007 18:18	1:38	0:21	12/21/2007 18:39	12/21/2007 19:41	1:02
19825	1007DKGXA	10	12/21/2007 17:23	0:59	0:07	1	12/21/2007 17:30	12/21/2007 19:21	1:51	0:20	12/21/2007 19:41	12/21/2007 20:42	1:00
19826	1007DKGXA	11	12/21/2007 18:27	1:04	0:08	2	12/21/2007 18:35	12/21/2007 20:19	1:44	0:23	12/21/2007 20:42	12/21/2007 21:47	1:04
19827	1007DKGXA	12	12/21/2007 19:45	1:18	0:12	1	12/21/2007 19:57	12/21/2007 21:29	1:32	0:18	12/21/2007 21:47	12/21/2007 22:52	1:05
19828	1007DKGXA	13	12/21/2007 20:42	0:57	0:12	2	12/21/2007 20:54	12/21/2007 22:35	1:41	0:17	12/21/2007 22:52	12/21/2007 23:56	1:03
19829	1007DKGXA	14	12/21/2007 21:47	1:05	0:12	1	12/21/2007 21:59	12/21/2007 23:35	1:36	0:21	12/21/2007 23:56	12/22/2007 1:00	1:04
19830	1007DKGXA	15	12/21/2007 22:54	1:07	0:12	2	12/21/2007 23:06	12/22/2007 0:40	1:34	0:20	12/22/2007 1:00	12/22/2007 2:11	1:10
19831	1007DKGXA	16	12/21/2007 23:41	0:47	0:11	1	12/21/2007 23:52	12/22/2007 1:51	1:59	0:20	12/22/2007 2:11	12/22/2007 3:27	1:16
19843	1008-MNDK-M1	1	12/22/2007 20:17	0:00	0:10	1	12/22/2007 20:27	12/22/2007 23:19	2:52	0:10	12/22/2007 23:29	12/23/2007 0:37	1:08
19844	1008-MNDK-M1	2	12/22/2007 21:22	1:05	0:10	2	12/22/2007 21:32	12/22/2007 23:34	2:02	1:03	12/23/2007 0:37	12/23/2007 1:40	1:02
19845	1008-MNDK-M1	3	12/22/2007 22:03	0:41	0:21	2	12/22/2007 22:24	12/23/2007 1:19	2:55	0:21	12/23/2007 1:40	12/23/2007 2:33	0:53
19846	1008-MNDK-M1	4	12/23/2007 00:41	2:38	0:10	1	12/23/2007 0:51	12/23/2007 2:15	1:24	0:18	12/23/2007 2:33	12/23/2007 3:38	1:04
19847	1008-MNDK-M1	5	12/23/2007 01:39	0:58	0:10	2	12/23/2007 1:49	12/23/2007 3:20	1:31	0:18	12/23/2007 3:38	12/23/2007 4:36	0:58
19848	1008-MNDK-M1	6	12/23/2007 02:42	1:03	0:10	1	12/23/2007 2:52	12/23/2007 4:14	1:22	0:22	12/23/2007 4:36	12/23/2007 5:35	0:59
19849	1008-MNDK-M1	7	12/23/2007 03:38	0:56	0:10	2	12/23/2007 3:48	12/23/2007 5:15	1:27	0:20	12/23/2007 5:35	12/23/2007 6:27	0:51
19850	1008-MNDK2	8	12/23/2007 04:35	0:57	0:10	1	12/23/2007 4:45	12/23/2007 6:13	1:28	0:14	12/23/2007 6:27	12/23/2007 7:29	1:02
19851	1008-MNDK2	9	12/23/2007 05:34	0:59	0:09	2	12/23/2007 5:43	12/23/2007 7:09	1:26	0:20	12/23/2007 7:29	12/23/2007 8:29	0:59
19852	1008-MNDK2	10	12/23/2007 06:34	1:00	0:17	1	12/23/2007 6:51	12/23/2007 8:08	1:17	0:21	12/23/2007 8:29	12/23/2007 9:27	0:57
19853	1008-MNDK2	11	12/23/2007 07:30	0:56	0:09	2	12/23/2007 7:39	12/23/2007 9:07	1:28	0:20	12/23/2007 9:27	12/23/2007 10:25	0:57
19854	1008-MNDK2	12	12/23/2007 08:26	0:56	0:09	1	12/23/2007 8:35	12/23/2007 10:04	1:29	0:21	12/23/2007 10:25	12/23/2007 11:43	1:17
19857	1007DKGXA	1	12/23/2007 12:46	0:00	0:09	2	12/23/2007 12:55	12/23/2007 14:06	1:11	0:11	12/23/2007 14:17	12/23/2007 15:30	1:13
19858	1007DKGXA	2	12/23/2007 13:41	0:55	0:08	1	12/23/2007 13:49	12/23/2007 15:14	1:25	0:16	12/23/2007 15:30	12/23/2007 16:47	1:16
19859	1007DKGXA	3	12/23/2007 15:01	1:20	0:08	2	12/23/2007 15:09	12/23/2007 16:32	1:23	0:15	12/23/2007 16:47	12/23/2007 17:49	1:02
19860	1007DKGXA	4	12/23/2007 15:54	0:53	0:09	1	12/23/2007 16:03	12/23/2007 17:28	1:25	0:21	12/23/2007 17:49	12/23/2007 18:53	1:03
19861	1007DKGXA	5	12/23/2007 17:07	1:13	0:09	2	12/23/2007 17:16	12/23/2007 18:34	1:18	0:19	12/23/2007 18:53	12/23/2007 19:55	1:02
19862	1007DKGXA	6	12/23/2007 18:31	1:24	0:12	1	12/23/2007 18:43	12/23/2007 19:34	0:51	0:21	12/23/2007 19:55	12/23/2007 21:04	1:09
19863	1007DKGXA	7	12/23/2007 19:50	1:19	0:12	2	12/23/2007 20:02	12/23/2007 20:45	0:43	0:19	12/23/2007 21:04	12/23/2007 22:11	1:06
19864	1007DKGXA	8	12/23/2007 20:53	1:03	0:12	1	12/23/2007 21:05	12/23/2007 21:51	0:46	0:20	12/23/2007 22:11	12/23/2007 23:11	1:00
19865	1007DKGXA	9	12/23/2007 22:05	1:12	0:12	2	12/23/2007 22:17	12/23/2007 22:59	0:42	0:12	12/23/2007 23:11	12/24/2007 0:24	1:12
19866	1007DKGXA	10	12/23/2007 23:19	1:14	0:12	1	12/23/2007 23:31	12/24/2007 0:11	0:40	0:13	12/24/2007 0:24	12/24/2007 1:35	1:11
19867	1007DKGXA	11	12/24/2007 00:20	1:01	0:10	1	12/24/2007 0:30	12/24/2007 1:16	0:46	0:19	12/24/2007 1:35	12/24/2007 2:46	1:11
19868	1007DKGXA	12	12/24/2007 01:18	0:58	0:11	1	12/24/2007 1:29	12/24/2007 2:32	1:03	0:14	12/24/2007 2:46	12/24/2007 3:53	1:07
19869	1007DKGXA	13	12/24/2007 02:14	0:56	0:11	2	12/24/2007 2:25	12/24/2007 3:32	1:07	0:21	12/24/2007 3:53	12/24/2007 4:55	1:02
19870	1007DKGXA	14	12/24/2007 03:07	0:53	0:11	1	12/24/2007 3:18	12/24/2007 4:33	1:15	0:22	12/24/2007 4:55	12/24/2007 5:55	0:59
19871	1007DKGXA	15	12/24/2007 04:15	1:08	0:11	2	12/24/2007 4:26	12/24/2007 5:36	1:10	0:19	12/24/2007 5:55	12/24/2007 6:58	1:02
19872	1007DKGXA	16	12/24/2007 05:15	1:00	0:11	1	12/24/2007 5:26	12/24/2007 6:35	1:09	0:23	12/24/2007 6:58	12/24/2007 8:12	1:14
19990	1008-MNDK2	1	12/30/2007 09:58	0:00	0:11	2	12/30/2007 10:09	12/30/2007 12:15	2:06	0:11	12/30/2007 12:26	12/30/2007 13:29	1:02
19991	1008-MNDK2	2	12/30/2007 10:59	1:01	0:11	1	12/30/2007 11:10	12/30/2007 13:07	1:57	0:22	12/30/2007 13:29	12/30/2007 14:26	0:57
19992	1008-MNDK2	3	12/30/2007 12:17	1:18	0:10	2	12/30/2007 12:27	12/30/2007 14:07	1:40	0:19	12/30/2007 14:26	12/30/2007 15:20	0:54
19993	1008-MNDK-M1	4	12/30/2007 13:29	1:12	0:10	1	12/30/2007 13:39	12/30/2007 14:59	1:20	0:21	12/30/2007 15:20	12/30/2007 16:15	0:55
19994	1008-MNDK-M1	5	12/30/2007 14:46	1:17	0:10	2	12/30/2007 14:56	12/30/2007 15:54	0:58	0:21	12/30/2007 16:15	12/30/2007 17:19	1:03
19995	1008-MNDK-M3	6	12/30/2007 15:55	1:09	0:10	1	12/30/2007 16:05	12/30/2007 16:58	0:53	0:21	12/30/2007 17:19	12/30/2007 18:15	0:56
19996	1008-MNDK-M3	7	12/30/2007 16:55	1:00	0:14	2	12/30/2007 17:09	12/30/2007 17:57	0:48	0:18	12/30/2007 18:15	12/30/2007 19:11	0:55
19997	1008-MNDK-M3	8	12/30/2007 17:47	0:52	0:09	1	12/30/2007 17:56	12/30/2007 18:49	0:53	0:22	12/30/2007 19:11	12/30/2007 20:05	0:53
19998	1008-MNDK2	9	12/30/2007 18:41	0:54	0:09	2	12/30/2007 18:50	12/30/2007 19:47	0:57	0:18	12/30/2007 20:05	12/30/2007 21:04	0:59
19999	1007DKGXA-00	10	12/30/2007 19:47	1:06	0:11	1	12/30/2007 19:58	12/30/2007 20:45	0:47	0:19	12/30/2007 21:04	12/30/2007 22:05	1:00
20000	1007DKGXA-00	11	12/30/2007 20:44	0:57	0:12	2	12/30/2007 20:56	12/30/2007 21:44	0:48	0:21	12/30/2007 22:05	12/30/2007 23:03	0:57
20001	1007DKGXA-00	12	12/30/2007 21:37	0:53	0:10	1	12/30/2007 21:47	12/30/2007 22:47	1:00	0:16	12/30/2007 23:03	12/31/2007 0:01	0:58
20009	1007DKGXA-00	1	12/31/2007 07:28	0:00	0:11	1	12/31/2007 7:39	12/31/2007 11:19	3:40	0:12	12/31/2007 11:31	12/31/2007 12:42	1:10
20010	1007DKGXA-00	2	12/31/2007 09:05	1:37	0:08	2	12/31/2007 9:13	12/31/2007 12:24	3:11	0:18	12/31/2007 12:42	12/31/2007 13:50	1:08
20011	1007DKGXA-00	3	12/31/2007 09:34	0:29	0:18	2	12/31/2007 9:52	12/31/2007 13:35	3:43	0:15	12/31/2007 13:50	12/31/2007 14:53	1:02
20012	1007DKGXA-00	4	12/31/2007 13:03	3:29	0:10	1	12/31/2007 13:13	12/31/2007 14:37	1:24	0:16	12/31/2007 14:53	12/31/2007 15:59	1:06
20013	1007DKGXA-00	5	12/31/2007 14:07	1:04	0:09	2	12/31/2007 14:16	12/31/2007 15:39	1:23	0:20	12/31/2007 15:59	12/31/2007 17:04	1:04
20014	1007DKGXA-00	6	12/31/2007 15:01	0:54	0:08	1	12/31/2007 15:09	12/31/2007 16:42	1:33	0:22	12/31/2007 17:04	12/31/2007 18:10	1:05
20015	1007DKGXA-00	7	12/31/2007 16:09	1:08	0:19	2	12/31/2007 16:28	12/31/2007 17:51	1:23	0:19	12/31/2007 18:10	12/31/2007 19:17	1:07
20016	1007DKGXA-00	8	12/31/2007 17:29	1:20	0:08	1	12/31/2007 17:37	12/31/2007 18:56	1:19	0:21	12/31/2007 19:17	12/31/2007 20:23	1:05
20017	1007DKGXA-00	9	12/31/2007 18:29	1:00	0:09	2	12/31/2007 18:38	12/31/2007 20:01	1:23	0:22	12/31/2007 20:23	12/31/2007 21:27	1:04
20018	1007DKGXA-00	10	12/31/2007 19:40	1:11	0:08	1	12/31/2007 19:48	12/31/2007 21:07	1:19	0:20	12/31/2007 21:27	12/31/2007 22:36	1:09
20019	1007DKGXA-00	11	12/31/2007 20:47	1:07	0:08	2	12/31/2007 20:55	12/31/2007 22:19	1:24	0:17	12/31/2007 22:36	12/31/2007 23:42	1:05
20020	1007DKGXA-00	12	12/31/2007 21:42	0:55	0:08	1	12/31/2007 21:50	12/31/2007 23:23	1:33	0:19	12/31/2007 23:42	1/1/2008 1:03	1:21

A.2. Example of Raw Data for Disruption Modeling

	No	HeatNo	Total Delay	EAF	Sintering	LHF	Oth-RF	CT/HSM	Delay/Lot
1	1	37251	112	13					
	2	37252							
	3	37253	79				79		
	4	37254	65			36		29	
	5	37255	30			30			
	6	37256	18			18			
	7	37257							
	8	37258	37			37			
	9	37259							
	10	37260							
	11	37261	3			3			
	12	37262	24			24			368.00
2	13	37263	7			7			
	14	37264	118	21	10	17	45		
	15	37265							
	16	37266	14			14			
	17	37267	19			19			
	18	37268							
	19	37269							
	20	37270							
	21	37271	15						
	22	37272							
	23	37273							
	24	37274							173.00
3	25	37275	12						
	26	37276	104					61	
	27	37277							
	28	37278							
	29	37279							
	30	37280	28			24		4	
	31	37281							
	32	37282							
	33	37283							
	34	37284							
	35	37285	22			22			
	36	37286	19			19			185.00
4	37	37287	16			16			
	38	37288	17			17			
	39	37289							
	40	37290	25			25			
	41	37291							
	42	37292							
	43	37293							
	44	37294							
	45	37295							
	46	37296	20		20				
	47	37297							
	48	37298							78.00
5	49	37299							
	50	37300							
	51	37301	8						
	52	37302							
	53	37303							
	54	37304							
	55	37305	9						
	56	37306							
	57	37307							
	58	37308							
	59	37309							
	60	37310							17.00
6	61	37311	37						
	62	37312							
	63	37313							
	64	37314							
	65	37315							
	66	37316	10		10				
	67	37317							
	68	37318							
	69	37319							
	70	37320							
	71	37321							
	72	37322							47.00
7	73	37323							
	74	37324							
	75	37325							
	76	37326							
	77	37327	70	64					
	78	37328							
	79	37329							
	80	37330							
	81	37331							
	82	37332	104						
	83	37333							
	84	37334	38	30					212.00
8	85	37335							
	86	37336							
	87	37337							
	88	37338							
	89	37339	54					54	
	90	37340	124					124	
	91	37341							
	92	37342							

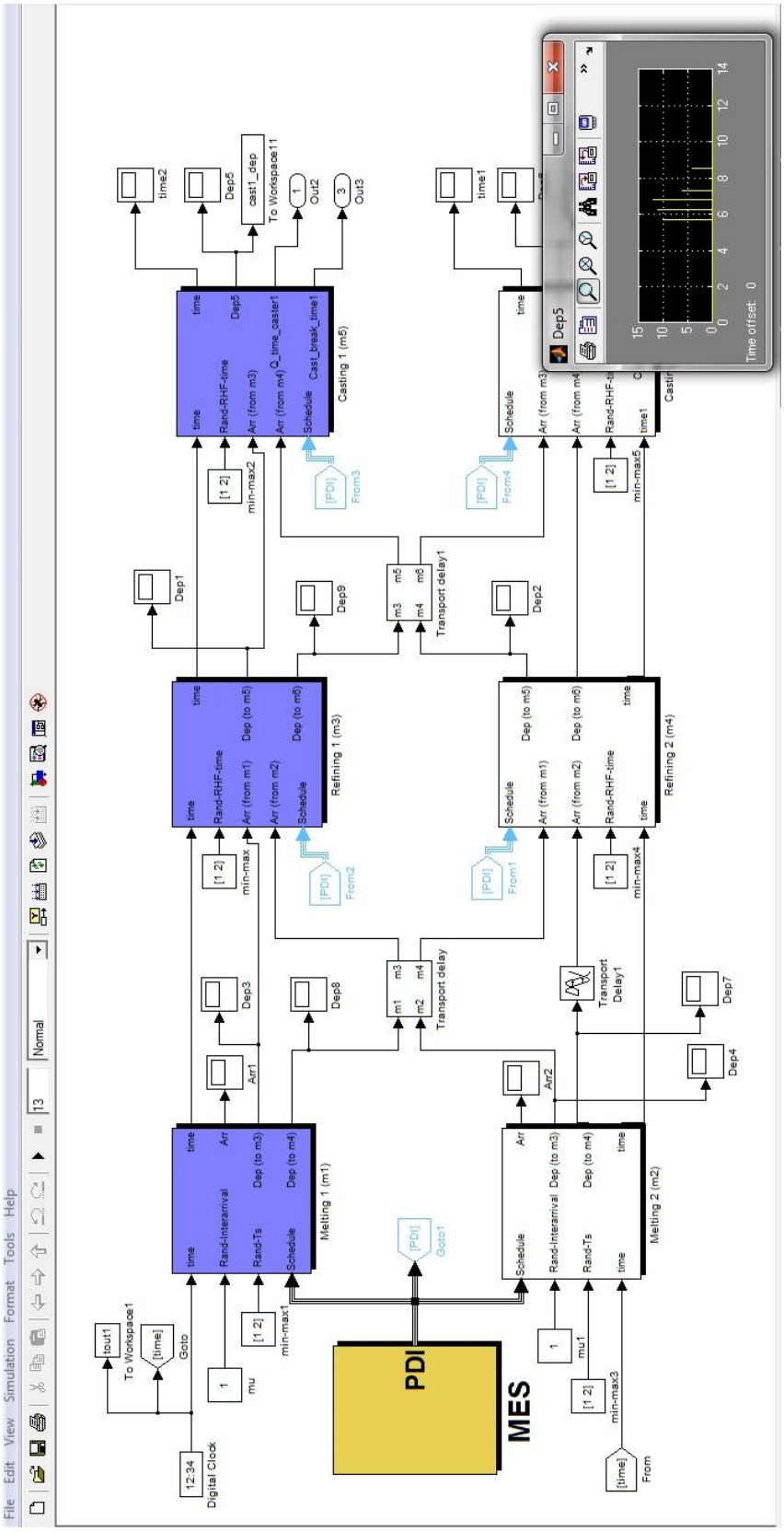
	No	HeatNo	Total Delay	EAF	Sintering	LHF	Oth-RF	CT/HSM	Delay/Lot
	93	37343	31						
	94	37344	14						
	95	37345	9	9					
	96	37346							232.00
9	97	37347							
	98	37348							
	99	37349							
	100	37350	45					45	
	101	37351							
	102	37352							
	103	37353	61					61	
	104	37354	20			20			
	105	37355	26					26	
	106	37356							
	107	37357	46					46	
	108	37358							198.00
10	109	37359							
	110	37360	37					37	
	111	37361							
	112	37362							
	113	37363							
	114	37364							
	115	37365	34					34	
	116	37366							
	117	37367							
	118	37368	21					21	
	119	37369							
	120	37370	80					80	172.00
11	121	37371							
	122	37372	7						
	123	37373							
	124	37374	36					36	
	125	37375							
	126	37376							
	127	37377	59					30	
	128	37378							
	129	37379							
	130	37380	4					4	
	131	37381							
	132	37382							106.00
12	133	37383	26					26	
	134	37384	15		15				
	135	37385	52	35					
	136	37386							
	137	37387							
	138	37388							
	139	37389							
	140	37390							
	141	37391							
	142	37392							
	143	37393	7						
	144	37394							100.00
13	145	37395							
	146	37396							
	147	37397	9						
	148	37398							
	149	37399							
	150	37400							
	151	37401							
	152	37402	88					88	
	153	37403	44						
	154	37404							
	155	37405							
	156	37406	10						151.00
14	157	37407							
	158	37408							
	159	37409							
	160	37410							
	161	37411							
	162	37412							
	163	37413	1						
	164	37414							
	165	37415							
	166	37416							
	167	37417	14			14			
	168	37418							15.00
15	169	37419							
	170	37420	198					187	
	171	37421	15	15					
	172	37422							
	173	37423							
	174	37424							
	175	37425	12						
	176	37426							
	177	37427							
	178	37428							
	179	37429	6						
	180	37430	28						259.00
16	181	37431							
	182	37432							
	183	37433	103		20			83	
	184	37434							

	No	HeatNo	Total Delay	EAF	Sintering	LHF	Oth-RF	CT/HSM	Delay/Lot
	185	37435							
	186	37436	39					39	
	187	37437	73					73	
	188	37438							
	189	37439	31					31	
	190	37440	20					20	
	191	37441							
	192	37442	65						331.00
17	193	37443							
	194	37444							
	195	37445							
	196	37446							
	197	37447							
	198	37448	24					24	
	199	37449	9				9		
	200	37450							
	201	37451	41					41	
	202	37452	5				5		
	203	37453							
	204	37454	75					75	154.00
18	205	37455							
	206	37456	35						
	207	37457							
	208	37458							
	209	37459							
	210	37460	41					41	
	211	37461							
	212	37462							
	213	37463							
	214	37464							
	215	37465	104					104	
	216	37466	10					10	190.00
19	217	37467	26			26			
	218	37468	30		30				
	219	37469							
	220	37470							
	221	37471							
	222	37472							
	223	37473							
	224	37474							
	225	37475	17					17	
	226	37476	22			22			
	227	37477	37			37			
	228	37478	2						134.00
20	229	37479	14						
	230	37480							
	231	37481							
	232	37482	22			22			
	233	37483	12			12			
	234	37484	14			14			
	235	37485	16			16			
	236	37486							
	237	37487	45			45			
	238	37488	35	24					
	239	37489							
	240	37490							158.00
21	241	37491	20			20			
	242	37492	18			18			
	243	37493	4			4			
	244	37494							
	245	37495							
	246	37496							
	247	37497							
	248	37498	2						
	249	37499							
	250	37500							
	251	37501							
	252	37502							44.00
22	253	37503	7						
	254	37504							
	255	37505							
	256	37506							
	257	37507	8					8	
	258	37508	38					38	
	259	37509	20		20				
	260	37510							
	261	37511	42					42	
	262	37512							
	263	37513	30						
	264	37514							145.00
23	265	37515							
	266	37516							
	267	37517							
	268	37518							
	269	37519							
	270	37520							
	271	37521							
	272	37522	80					80	
	273	37523							
	274	37524							
	275	37525							
	276	37526	5						85.00

	No	HeatNo	Total Delay	EAF	Sintering	LHF	Oth-RF	CT/HSM	Delay/Lot
24	277	37527							
	278	37528							
	279	37529							
	280	37530							
	281	37531							
	282	37532	31					31	
	283	37533							
	284	37534							
	285	37535	43					43	
	286	37536							
	287	37537							
	288	37538							74.00
25	289	37539	82					82	
	290	37540							
	291	37541							
	292	37542							
	293	37543	39					39	
	294	37544	36	36					
	295	37545							
	296	37546							
	297	37547							
	298	37548	10						
	299	37549							
	300	37550							167.00
26	301	37551							
	302	37552							
	303	37553							
	304	37554							
	305	37555							
	306	37556	32					32	
	307	37557	25						
	308	37558							
	309	37559	26			26			
	310	37560	40		40				
	311	37561							
	312	37562							123.00
27	313	37563							
	314	37564							
	315	37565							
	316	37566	20					7	
	317	37567							
	318	37568							
	319	37569							
	320	37570							
	321	37571							
	322	37572							
	323	37573							
	324	37574							20.00
28	325	37575	7						
	326	37576	53					53	
	327	37577	3						
	328	37578							
	329	37579							
	330	37580							
	331	37581	6						
	332	37582							
	333	37583	5						
	334	37584	27			27			
	335	37585	7			7			
	336	37586	10			10			118.00
29	337	37587	11						
	338	37588							
	339	37589							
	340	37590	10						
	341	37591							
	342	37592							
	343	37593	3			3			
	344	37594	19			19			
	345	37595							
	346	37596	17			17			
	347	37597	10		10				
	348	37598							70.00
30	349	37599							
	350	37600	21	21					
	351	37601	10						
	352	37602	65						
	353	37603							
	354	37604							
	355	37605	8						
	356	37606	6						
	357	37607	5						
	358	37608							
	359	37609							
	360	37610	69	19				45	184.00
31	361	37611							
	362	37612							
	363	37613							
	364	37614	10			10			
	365	37615	7			7			
	366	37616	12			12			
	367	37617							
	368	37618	116					116	

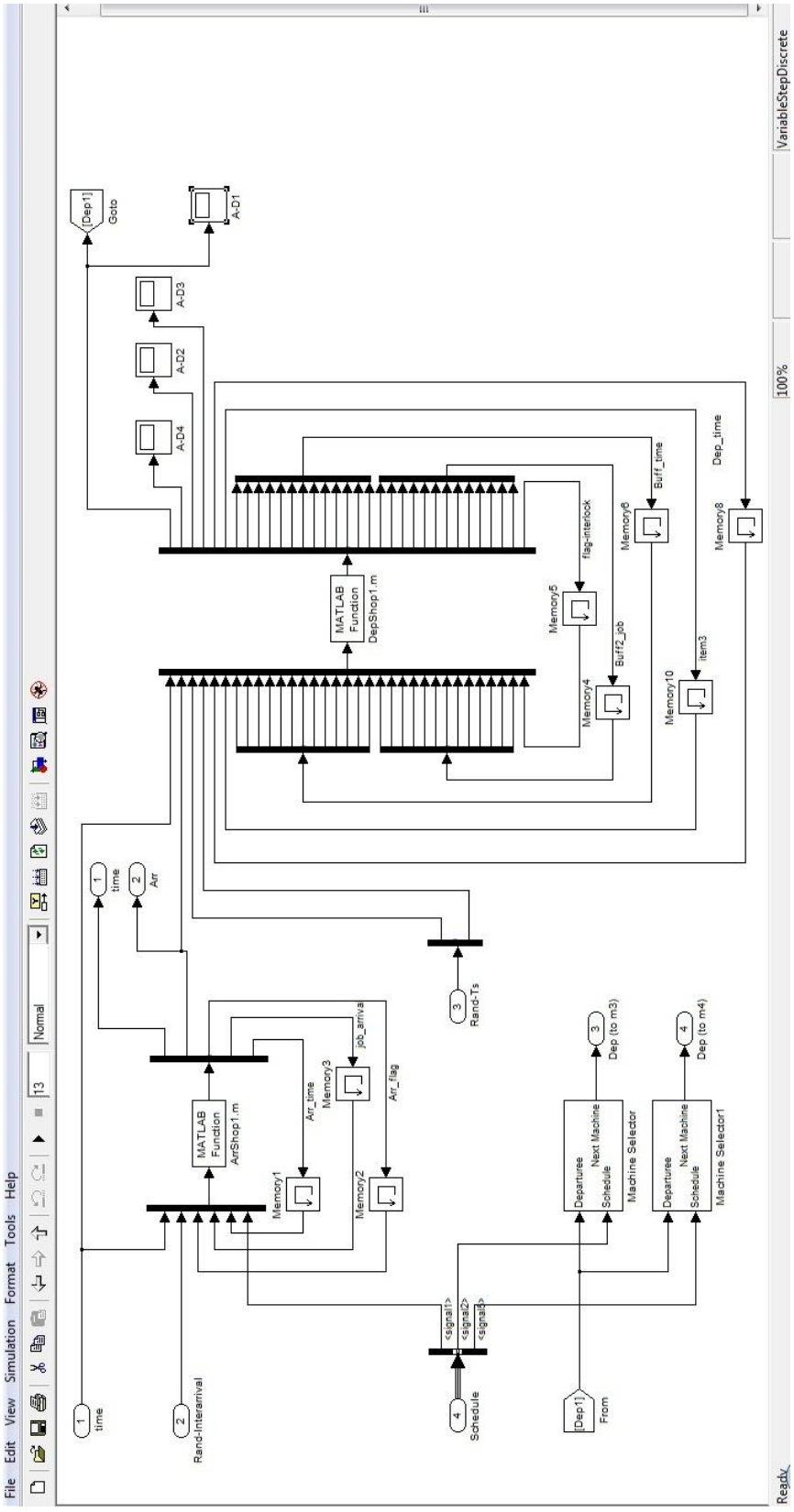
APPENDIX B SIMULATION INTERFACE VIA MATLAB

B.1. Parallel Production Line

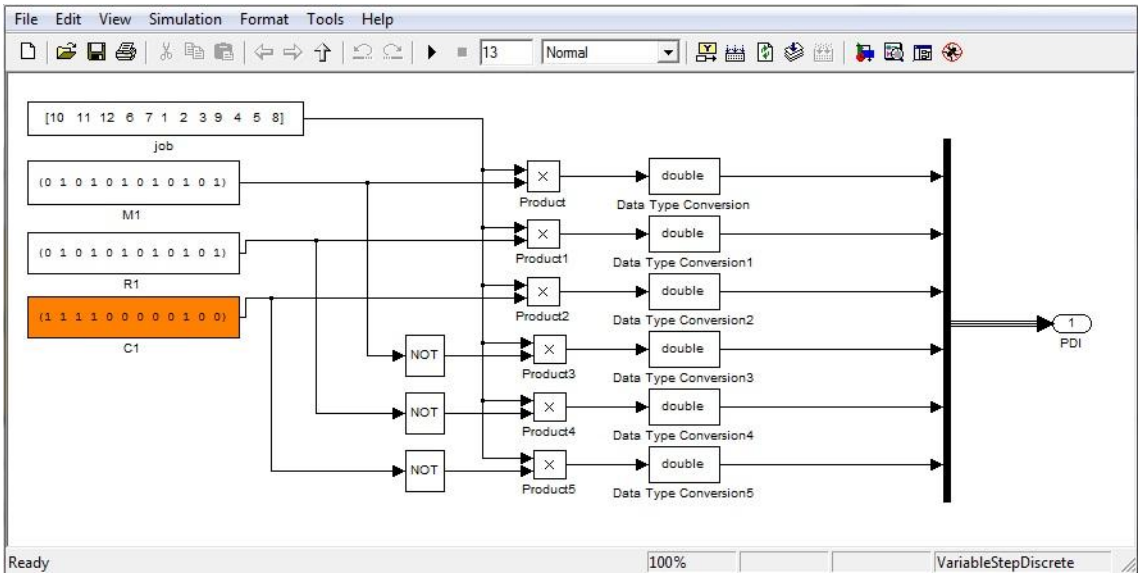


2-lines SCC production

B.2. Detail in Each Simulation Block



Melting block


File Edit View Simulation Format Tools Help
13
Normal
[10 11 12 6 7 1 2 3 9 4 5 8]
job
(0 1 0 1 0 1 0 1 0 1 0 1)
M1
(0 1 0 1 0 1 0 1 0 1 0 1)
R1
(1 1 1 1 0 0 0 0 0 1 0 0)
C1
NOT
NOT
NOT
Product
Product1
Product2
Product3
Product4
Product5
double
Data Type Conversion
Data Type Conversion1
Data Type Conversion2
Data Type Conversion3
Data Type Conversion4
Data Type Conversion5
1
PDI
Ready
100%
VariableStepDiscrete

MES block

APPENDIX C MATLAB CODE

C.1. Code of 2 Lines SCC Production Simulation (with UI)

File: Arrshop1.m

```
function [Event]=GenEventArr(input)

base_clock=input(1)*10000;
mu=input(2)*10000;

Arr_flag=input(3);
job_arrival=input(4);
Arr_time=input(5);
sizePDI=size(input,1)-5;

item2=0;
for item1=6:size(input,1
    if input(item1)>0
        item2=item2+1;
        PDI(item2)=input(item1);
    else
        PDI=0;
    end
end
max_job=size(PDI,2);

if Arr_flag == 1
   job_arrival=job_arrival+1;
   if base_clock==0
      Int_time=0;

   else
      Int_time=mu;
   end
   Arr_time=Arr_time+Int_time;
   Arr_flag=0;

else
   Arr_time=Arr_time;
   Arr_flag=0;
end

Event(1)=base_clock/10000;

if base_clock >= Arr_time
   if job_arrival <= max_job
      Event(2)=PDI(job_arrival);
      Arr_flag=1;
   else
      Event(2)=0;
      Arr_flag=0;
   end
else
   Event(2)=0;
   Arr_flag=0;
end

Event(3)=Arr_flag;
Event(4)=job_arrival;
```

```
Event(5)=Arr_time;
```

File: DepShop1.m

```
function [Event]=GenEventDep(input)

base_clock=input(1);
Arrival=input(2);
min_survice=input(3);
max_survice=input(4);
Dep_time=input(5);
item3=input(6);
flag_interlock=input(33);

for item1=7:19
    Buff_t1(item1-6)=input(item1);
end
for item5=20:32
    Buff_Arr1(item5-19)=input(item5);
end

if Arrival>0

   if base_clock==0
      Arr_t=0;
      Que_time=0;
      Ser_time=GetSurviceTime(Arrival,1)+3;
   else
      Arr_t=base_clock;
      Que_time=max(0,(Dep_time-Arr_t));
      Ser_time=GetSurviceTime(Arrival,1);
   end
   Dep_time=Arr_t+Que_time+Ser_time; %--
   Buff_t1(item3)=Dep_time;
   Buff_Arr1(item3)=Arrival;
   item3=item3+1;
   flag_interlock=1;

else
   Dep_time=Dep_time;
   Que_time=0;
   flag_interlock=0;
end

base_clock=single(base_clock);
Buff_t1=single(Buff_t1);

if (10000*base_clock) >= (10000*Buff_t1(1)) & Buff_t1(1)>0 %&
      Event(1)=Buff_Arr1(1);

      for item4=1:(item3-1)
          Buff_t1(item4)=Buff_t1(item4+1);
          Buff_Arr1(item4)=Buff_Arr1(item4+1);
      end
      item3=item3-1;
      flag_interlock=0;
```

```
else
      Event(1)=0;
end

Event(2)=Buff_t1(1);
Event(3)=Que_time;
Event(4)=0;
Event(5)=Dep_time;
Event(6)=item3;
Event(33)=flag_interlock;

for item2=7:19
    Event(item2)=Buff_t1(item2-6);
end
for item6=20:32
    Event(item6)=Buff_Arr1(item6-19);
end
```

File: DepCaster1_castbreak.m

```
function [Event]=DepShop2(input)
time_ratio=100;

base_clock=input(1);
Arrival=input(2);
min_survice=input(3);
max_survice=input(4);
Dep_time=input(5);
item3=input(6);
Arrival2=input(33);
flag_interlock=input(46);

for item1=7:19
    Buff_t2(item1-6)=input(item1);
end
for item5=20:32
    Buff_Arr2(item5-19)=input(item5);
end

uppersizePDI=size(input,1)-1;
item7=0;
for item8=34:uppersizePDI
    if input(item8)>0
        item7=item7+1;
        PDI_dep(item7)=input(item8);
    end
end

if (Arrival>0) | (Arrival2>0)

   for item9=1:size(PDI_dep,2)
       Arrival=int8(Arrival);
       Arrival2=int8(Arrival2);
       if Arrival==PDI_dep(item9)
          Buff_t2(item9)=base_clock;
          Buff_Arr2(item9)=Arrival;
```

```
        elseif Arrival2==PDI_dep(item9)
           Buff_t2(item9)=base_clock;
           Buff_Arr2(item9)=Arrival2;
        end
    end

else
    Buff_t2=Buff_t2;
    Buff_Arr2=Buff_Arr2;
end

if (Buff_Arr2(item3)>0)&(flag_interlock==0)
     if item3==1
         Dep_time=0
         Cast_break=0;
     else
         Dep_time=Dep_time;
         Cast_break=abs(min(0,Dep_time-Buff_t2(item3)));
     end

     Ser_time=GetSurviceTime(Buff_Arr2(item3),3);
     Que_time=max(0,(Dep_time-Buff_t2(item3)));
     Dep_time=Buff_t2(item3)+Que_time+Ser_time;
     flag_interlock=1;

else
     Dep_time=Dep_time;
     Que_time=0;
     Que_time_max=0;
     Cast_break=0;

end

base_clock=single(base_clock);
Dep_time=single(Dep_time);

if ((10000*base_clock) >= (10000*Dep_time))&(Dep_time > 0)&
flag_interlock==1

     Event(1)=Buff_Arr2(item3);
     item3=item3+1;
     flag_interlock=0;
else
     Event(1)=0;
end

Event(2)=Buff_t2(1);
if (Que_time*time_ratio)<40
     Event(3)=0;
else
     Event(3)=(Que_time*time_ratio)-40;
end
if (Cast_break*time_ratio)<10
     Event(4)=0;
else
     Event(4)=Cast_break*time_ratio;
end
Event(5)=Dep_time;
Event(6)=item3;
Event(33)=flag_interlock;
```

```
for item2=7:19
    Event(item2)=Buff_t2(item2-6);
end
for item6=20:32
    Event(item6)=Buff_Arr2(item6-19);
end
```

File: GetServiceTime.m

```
function servicetime = GetSurviceTime(arrival,resource)

time_ratio=100;
iniPlan=[100  100  100  110  110  130  130  130  100  001  001  001;
         1594 1594 1594 1594 1570 1570 1570 1570 1570 1564 1564 1564;
         60   60   60   60   60   60   60   60   60   50   50   50;
         4    4    4    4    4    4    4    4    4    4    4    4];

if resource==1
    servicetime_melt=(54 + gamrnd(2.03, 4.6))/time_ratio;%for test
    servicetime=servicetime_melt;

elseif  resource==2
    servicetime_refine=(50.5 + 49*betarnd(1.4, 1.01))/time_ratio;
    servicetime=servicetime_refine;
else
    servicetime=getservicetimeCasting(arrival,iniPlan)/time_ratio;
end

function ser_time_melt = getservicetimeMelting(arrival,iniPlan)
if iniPlan(1,arrival)==001
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
elseif iniPlan(1,arrival)==100
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
elseif iniPlan(1,arrival)==110
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
elseif iniPlan(1,arrival)==120
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
elseif iniPlan(1,arrival)==130
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
end

function ser_time_refine = getservicetimeRefining(arrival,iniPlan)
if iniPlan(1,arrival)==001
    ser_time_refine=unifrnd(2,2.5,1,1);
elseif iniPlan(1,arrival)==100
    ser_time_refine=unifrnd(2,2.5,1,1);
elseif iniPlan(1,arrival)==110
    ser_time_refine=unifrnd(2,2.5,1,1);
elseif iniPlan(1,arrival)==120
    ser_time_refine=unifrnd(2,2.5,1,1);
elseif iniPlan(1,arrival)==130
    ser_time_refine=unifrnd(2,2.5,1,1);
end

function ser_time_caster = getservicetimeCasting(arrival,iniPlan)
if iniPlan(1,arrival) < 500
```

```
    if iniPlan(2,arrival) < 1350
        if iniPlan(3,arrival)==60
            Cast_speed=5.2;
        else
            Cast_speed=5.5;
        end
    elseif (iniPlan(2,arrival) >= 1350)&(iniPlan(2,arrival) <= 1450)
        if iniPlan(3,arrival)==60
            Cast_speed=5.0;
        else
            Cast_speed=5.2;
        end
    elseif iniPlan(2,arrival) > 1450
        if iniPlan(3,arrival)==60
           Cast_speed=4.8;
        else
           Cast_speed=5.0;
        end
    end %check width
else
     if iniPlan(2,arrival) < 1350
        if iniPlan(3,arrival)==60
            Cast_speed=4.8;
        else
            Cast_speed=5.0;
        end
    elseif (iniPlan(2,arrival) >= 1350)&(iniPlan(2,arrival) <= 1450)
        if iniPlan(3,arrival)==60
            Cast_speed=4.6;
        else
            Cast_speed=4.8;
        end
    elseif iniPlan(2,arrival) > 1450
        if iniPlan(3,arrival)==60
            Cast_speed=4.4;
        else
            Cast_speed=4.6;
        end
    end
end

servicetime_cast=(normrnd(165,7.56)*10^6)/(Cast_speed*iniPlan(2,arriva
l)*iniPlan(3,arrival)*7.6);
ser_time_caster=servicetime_cast;
```

C.2. Code of Deterministic Scheduling using GA

List of files:

1. runGA_Detnew_Matlab7.m
2. genXlhd.m
3. ObjxQx_New3_Caster.m
4. Steel_Process.m
5. GetService.m

File: runGA_Detnew_Matlab7.m

```
function
[X,Sq,dW,vec_fval]=runGA_DetNew(nPops,nGens,initGA,initGA_sq,initGA_dW
,flagInit)

fitFunc = @ObjxQx_New2;

nReschedCast = 8;
nReschedHeat = 12;

nVars = nReschedCast*nReschedHeat;
nVars2 = 3*12;

%---------Define Options Outter Loop-----------------------
options = gaoptimset;

options = gaoptimset(options,'PopInitRange' ,[0 0 0 0;1 1 1 1]);
options = gaoptimset(options,'Generations' ,nGens);
options = gaoptimset(options,'PopulationSize' ,nPops);
options = gaoptimset(options,'StallGenLimit' ,500000);
options = gaoptimset(options,'StallTimeLimit' ,1000000);
options = gaoptimset(options,'EliteCount' ,1);
options = gaoptimset(options,'CrossoverFcn' ,@crossover_Resched);
options.CrossoverFraction=0.8;
options = gaoptimset(options,'MutationFcn' ,{ @mutationuniform 0.01
});
options = gaoptimset(options,'Display' ,'off');

if flagInit==1
    [initGA,initGA_sq,initGA_dW] = genXlhd(nPops,nReschedHeat,2,4);
else
    initGA    = initGA;
    initGA_dW = initGA_dW;
    initGA_sq = initGA_sq;
end

options  = gaoptimset(options,'InitialPopulation' ,initGA);

%-----------Find initial pop of starting time chromosome ----
options2 = gaoptimset;
options2 = gaoptimset(options2,'Generations' ,nGens);
options2 = gaoptimset(options2,'PopulationSize' ,nPops);
options2 = gaoptimset(options2,'StallGenLimit' ,500000);
options2 = gaoptimset(options2,'StallTimeLimit' ,1000000);
options2 = gaoptimset(options2,'EliteCount' ,1);
options2 = gaoptimset(options2,'CrossoverFcn' ,@crossoversinglepoint);
```

```
options2 = gaoptimset(options2,'MutationFcn' ,{ @mutationuniform 0.01
});
options2 = gaoptimset(options2,'Display' ,'off');
options2 = gaoptimset(options,'InitialPopulation' ,initGA_dW);
options3 = options2;
options3 = gaoptimset(options,'InitialPopulation' ,initGA_sq);

%---------Run GA-------------------------------------------
GenomeLengthX   = nVars;
GenomeLength_sq  = nVars2;
GenomeLength_dW = nVars2;

nextPopulationX   = options.InitialPopulation;
nextPopulation_sq = options3.InitialPopulation;
nextPopulation_dW = options2.InitialPopulation;

totalPop = options.PopulationSize;

%-------find out this score--------------------
for pop = 1:totalPop

[nextScoreX(pop),q_time(pop),castbr_time(pop)]=fitFunc(nextPopulationX
(pop,:),nextPopulation_sq(pop,:),nextPopulation_dW(pop,:));
end
disp('-----------------------------------------------------');

%--------GA LOOP----------------------------------
maxiters=int16(options.Generations);
for i=1:maxiters
    %iterate=i
    thisScoreX=nextScoreX;
    thisPopulationX   = nextPopulationX;
    thisPopulation_sq = nextPopulation_sq;
    thisPopulation_dW = nextPopulation_dW;

%------define no. of next population-----------
    nEliteKidX  = options.EliteCount;
    nXoverKidX  = round(options.CrossoverFraction *
(size(thisPopulationX,1) - nEliteKidX));
    nMutateKidX = size(thisPopulationX,1) - nEliteKidX - nXoverKidX;
    nParentX    = 2 * nXoverKidX + nMutateKidX;

    nEliteKid_sq  = options3.EliteCount;
    nXoverKid_sq  = round(options3.CrossoverFraction *
(size(thisPopulation_sq,1) - nEliteKid_sq));
    nMutateKid_sq = size(thisPopulation_sq,1) - nEliteKid_sq -
nXoverKid_sq;
    nParent_sq    = 2 * nXoverKid_sq + nMutateKid_sq;

    nEliteKid_dW  = options2.EliteCount;
    nXoverKid_dW  = round(options2.CrossoverFraction *
(size(thisPopulation_dW,1) - nEliteKid_dW));
    nMutateKid_dW = size(thisPopulation_dW,1) - nEliteKid_dW -
nXoverKid_dW;
    nParent_dW    = 2 * nXoverKid_dW + nMutateKid_dW;

%-------Scalling------------------------------
    ExpectationX = fitscalingrank(thisScoreX,nParentX);
    parentX      = selectionroulette(ExpectationX,nParentX,options);
    parentX      = parentX(randperm(length(parentX)));
```

```
    [unused,k]    = sort(thisScoreX);
    Expectation_sq = fitscalingrank(thisScoreX,nParent_sq);
    parent_sq = selectionroulette(Expectation_sq,nParent_sq,options3);
    parent_sq = parent_sq(randperm(length(parent_sq)));
    [unused3,k3] = sort(thisScoreX);

    Expectation_dW = fitscalingrank(thisScoreX,nParent_dW);
    parent_dW = selectionroulette(Expectation_dW,nParent_dW,options2);
    parent_dW = parent_dW(randperm(length(parent_dW)));
    [unused2,k2] = sort(thisScoreX);

 %-------GA Operations-------------------------
    eliteKidX  = thisPopulationX(k(1:options.EliteCount),:);
    t1=thisScoreX(k(1))
    xoverKidX  =
crossoverResched(parentX(1:(2*nXoverKidX)),options,GenomeLengthX,fitFu
nc,unused,thisPopulationX);
    mutateKidX =
mutationuniform(parentX((1+2*nXoverKidX):end),options,GenomeLengthX,fi
tFunc,thisScoreX,thisPopulationX,0.01);
    nextPopulationX = [eliteKidX ; xoverKidX ; mutateKidX]
    eliteKid_sq  = thisPopulation_sq(k3(1:options3.EliteCount),:);
    xoverKid_sq  =
crossoversinglepoint_sq(parent_sq(1:(2*nXoverKid_sq)),options3,GenomeL
ength_sq,fitFunc,unused3,thisPopulation_sq)
    mutateKid_sq =
mutationMatrix_sq(parent_sq((1+2*nXoverKid_sq):end),thisPopulation_sq,
3,12,0.10);
    nextPopulation_sq = [eliteKid_sq ; xoverKid_sq ; mutateKid_sq];

    eliteKid_dW  = thisPopulation_dW(k2(1:options2.EliteCount),:);
    xoverKid_dW  =
crossoversinglepoint(parent_dW(1:(2*nXoverKid_dW)),options2,GenomeLeng
th_dW,fitFunc,unused2,thisPopulation_dW);
    mutateKid_dW =
mutationuniform(parent_dW((1+2*nXoverKid_dW):end),options2,GenomeLengt
h_dW,fitFunc,thisScoreX,thisPopulation_dW,0.01);
    nextPopulation_dW = [eliteKid_dW ; xoverKid_dW ; mutateKid_dW];

%-------find out this score--------------------
for pop = 1:totalPop

[nextScoreX(pop),q_time(pop),castbr_time(pop)]=fitFunc(nextPopulationX
(pop,:),nextPopulation_sq(pop,:),nextPopulation_dW(pop,:));

end

%-------find answer----------------------------
    [fval_sort,i_sort] = sort(nextScoreX);
    iterate=i
    x_min(i,:)=nextPopulationX(i_sort(1),:);
    sq_min(i,:)=nextPopulation_sq(i_sort(1),:);
    dw_min(i,:)=nextPopulation_dW(i_sort(1),:);

    fval_min=fval_sort(1)
    status = sprintf('eff= %0.2f q= %0.2f cb=
%0.2f',nextScoreX(i_sort(1)),q_time(i_sort(1)),castbr_time(i_sort(1)))

%-------Record obj. value in each iters--------
    vec_fval(i)=fval_sort(1);
```

```
    state.Generation=i;
    state.Score=nextScoreX;

disp('-------------------------------------------------------');
disp('                   end iteration                       ');
disp('-------------------------------------------------------');

end

X  = x_min(end,:);
Sq = sq_min(end,:);
dW = dw_min(end,:);
FVAL = vec_fval(end);

function [expectation] = fitscalingrank(scores,nParents)
nParents
[unused,i] = sort(scores)
expectation = zeros(size(scores))
%expectation(i) = 1./((1:length(scores)).^ 0.5)
expectation = 1./((1:length(scores)).^ 0.5)
expectation = nParents*expectation./ sum(expectation)

%*************************************************************
function parents = selectionroulette(expectation,nParents,options)
wheel = cumsum(expectation) / nParents;

parents = zeros(1,nParents);
for i = 1:nParents
    r = rand;
    for j = 1:length(wheel)
        if(r < wheel(j))
            parents(i) = j;
            break;
        end
    end
end
%*************************************************************
function xoverKids  =
crossoversinglepoint(parents,options,GenomeLength,FitnessFcn,unused,th
isPopulation)

nKids = length(parents)/2;
xoverKids = zeros(nKids,GenomeLength);
index = 1;

for i=1:nKids
    % get parents
    parent1 = thisPopulation(parents(index),:);
    index = index + 1;
    parent2 = thisPopulation(parents(index),:);
    index = index + 1;
    xOverPoint = ceil(rand * (length(parent1) - 1));
    xoverKids(i,:) = [ parent1(1:xOverPoint),parent2(( xOverPoint + 1
):  end )  ];
end

%*************************************************************
function mutationChildren =
mutationuniform(parents,options,GenomeLength,FitnessFcn,thisScore,this
Population,mutationRate)
```

```
if(nargin < 8)
    mutationRate = 0.01;
end

mutationChildren = zeros(length(parents),GenomeLength);
for i=1:length(parents)
    child = thisPopulation(parents(i),:);
    mutationPoints = find(rand(1,length(child)) < mutationRate);
    child(mutationPoints) = ~child(mutationPoints);
    mutationChildren(i,:) = child;
end
%************************************************************
function state = gaplotbestf(options,state,flag)

if(flag==1)
    set(gca,'xlim',[1,options.Generations]);
    xlabel Generation
    ylabel('Fitness value');

end

hold on;
generation = state.Generation;
best = min(state.Score);
plot(generation,best, '.k');
m = mean(state.Score);
plot(generation,m,'.b');
title(['Best: ',num2str(best),' Mean: ',num2str(m)])
hold off;

%************************************************************
function xoverKids  =
crossoverResched(parents,options,GenomeLength,FitnessFcn,unused,thisPo
pulation)

nKids = length(parents)/2;
xoverKids = zeros(nKids,GenomeLength);
index = 1;

for i=1:nKids
    % get parents
    parent1 = thisPopulation(parents(index),:);
    index = index + 1;
    parent2 = thisPopulation(parents(index),:);
    index = index + 1;

    k3=0;
    for k1=1:nCast
        for k2=1:nHeat
            k3=k3+1;
            mat_parent1(k1,k2)=parent1(k3);
            mat_parent2(k1,k2)=parent2(k3);
        end
    end
    mat_parent1;
    mat_parent2;

    xOverPoint = ceil(rand*nHeat);
    mat_xoverKids =
[mat_parent1(:,1:xOverPoint),mat_parent2(:,(xOverPoint+1):end)];
```

```
    k3=0;
    for k1=1:nCast
        for k2=1:nHeat
            k3=k3+1;
            xoverKids(i,k3)=mat_xoverKids(k1,k2);
        end
    end
end

%*****************************************************************
function xoverKids  =
crossoversinglepoint_sq(parents,options,GenomeLength,FitnessFcn,unused
,thisPopulation)

nKids = length(parents)/2;
xoverKids = zeros(nKids,GenomeLength);
index = 1;

for i=1:nKids
    parent1 = thisPopulation(parents(index),:);
    index = index + 1;
    parent2 = thisPopulation(parents(index),:);
    index = index + 1;

    xOverPoint = (ceil(rand * (3 - 1)))*12; %random between 12,24
    xoverKids(i,:) = [ parent1(1:xOverPoint),parent2(( xOverPoint + 1
):  end )  ];
end
%*****************************************************************
function mutationChildren =
mutationMatrix_sq(parents,thisPopulation,nProcess,nHeat,mutationRate)

mutationChildren = zeros(length(parents),nProcess*nHeat);
pRandom=round(rand(1)*100);

for k4=1:length(parents)% no. of muatation pop

    if pRandom <=round(100*mutationRate)

        selectedPop  = thisPopulation(parents(k4),:);
        row_selected = randperm(3);
        rowMut  = row_selected(1);
        col_selected = randperm(nHeat);
        colMut1 = col_selected(1);%
        colMut2 = col_selected(2);
        k3=0;
        for k1=1:nProcess
            for k2=1:nHeat
                k3=k3+1;
                mat_pop(k1,k2)=selectedPop(k3);
            end
        end

        mat_pop_new = mat_pop;
        mat_pop_new(rowMut,colMut1)= mat_pop(rowMut,colMut2);
        mat_pop_new(rowMut,colMut2)= mat_pop(rowMut,colMut1);
```

```
        k3=0;
        for k1=1:nProcess
            for k2=1:nHeat
                k3=k3+1;
                vec_pop_new(k3)=mat_pop_new(k1,k2);
            end
        end

        mutationChildren(k4,:)=vec_pop_new;

    else
        mutationChildren(k4,:)=thisPopulation(parents(k4),:);
    end
end
```

File: genXlhd.m

```
function [X,sqX,dW]=genXlhd(nSample,nJob,nMPP,nCast)

intv_1=nJob;
intv_2=2*nJob;
intv_3=3*nJob;

%----------gen X-------------------------------------
X1=lhsdesign(nJob*2,nSample)';

for i=1:intv_1
    Xscale(:,i)=ceil((X1(:,i)*100)*nMPP/100);
end

for i=(intv_1+1):intv_2   %i=7:12
    Xscale(:,i)=ceil((X1(:,i)*100)*nMPP/100);
end

Xeaf = decodeX2( Xscale(:,1:intv_1), nMPP , nJob);
Xlhf = decodeX2( Xscale(:,intv_1+1:intv_2) , nMPP , nJob);
Xcasts=genCasts(nCast,nJob,nSample)
X=[Xeaf Xlhf Xcasts];
%-----------gen sqX---------------------------------
for i=1:nSample
    sqXeaf(i,:)=randperm(nJob);
end

for i=1:nSample
    sqXlhf(i,:)=randperm(nJob);
end

for i=1:nSample
    sqXcasts(i,:)=randperm(nJob);
end
sqX=[sqXeaf sqXlhf sqXcasts];
%------------gen dW-----------------------------------
genW_eaf=lhsdesign(nSample,nJob);
genW_lhf=lhsdesign(nSample,nJob);
genW_cc =lhsdesign(nSample,nJob);

for i=1:nSample
```

```
    dW_eaf(i,:)=ceil((genW_eaf(i,:)*100)*30/100);
    dW_lhf(i,:)=ceil((genW_lhf(i,:)*100)*30/100);
    dW_cc(i,:) =ceil((genW_cc(i,:)*100)*30/100);
end
dW=[dW_eaf dW_lhf dW_cc];

%*****************************************************
%                   Subfunction
%*****************************************************
function [Xnew]=decodeX(X,nOutRow,nOutCol)
nSample=size(X,1);
for i=1:nSample
    x_new=zeros(nOutRow,nOutCol);
    for k=1:nOutCol
        x_new(X(i,k),k)=1;
    end
    Xnew{i}=x_new;
end

%*****************************************************
function [Xnew]=decodeX2(Xscale,nMPP,nJob)
nSample=size(Xscale,1);
for i=1:nSample
    x_new=zeros(nMPP,nJob);
    X_new=zeros(1,nMPP*nJob);

    for k=1:nJob
        x_new(Xscale(i,k),k)=1;
    end
    j_st=1;
    for j=1:nMPP
        j_st;
        j_end=j*nJob;
        Xnew(i,j_st:j_end)=x_new(j,:);
        j_st=1+j_end;
    end
end

%*****************************************************
function [Xcasts]=genCasts(nCasts,nJob,totalpopulation)
range = [0 0 0 0;1 1 1 1];
lowerBound = range(1,:);
upberBound = range(2,:);

sizeVAR=(nCasts*nJob)/size(range,2);
for k2=1:totalpopulation

     cast1=zeros(1,sizeVAR);%4 casts
     cast2=zeros(1,sizeVAR);
     cast3=zeros(1,sizeVAR);
     cast4=zeros(1,sizeVAR);
     while(sum(cast1)==0 | sum(cast1)>(sizeVAR/3))
           cast1=round(rand(1,sizeVAR));
     end

     while(sum(cast2)==0 | sum(cast2)>(sizeVAR/3) |
sum(cast1.*cast2)>0)
           cast2=round(rand(1,sizeVAR));
     end
```

```
     while(sum(cast3)==0 | sum(cast3)>(sizeVAR/4) |
sum(cast1.*cast2)>0 | sum(cast1.*cast3)>0 | sum(cast3.*cast2)>0)
           cast3=round(rand(1,sizeVAR));
     end

     for k1=1:sizeVAR
         if sum(cast1(k1)+cast2(k1)+cast3(k1)+cast4(k1))==0
            cast4(k1)=1;
         else
            cast4(k1)=0;
         end
     end

     pop2(k2,:)=[cast1 cast2 cast3 cast4];

end
Xcasts=[pop2];
```

File: ObjxQx_New3_Caster.m

```
function z = ObjxQx_New3_Caster(input)

input1 = input(1:48);
input2 = input(49:144);
input3 = input(145:288);

%-------Set initial and constant variables-----------------
dim_ip   = size(input1);
nVar     = 4;
nProcess = 3;
nHeat    = 12;
nCast    = 4;

%-------Set initial for Disruption Scenario----------------
iniPlan =[100  100  100  110  110  130  130  130  100  001  001  001;
          1272 1258 1252 1230 1230 1272 1552 1245 1240 1272 1255 1275
          60   60   50   60   60   60   50   50   60   60   60   50;
          320  320  320  320  320  350  420  420  450  450  480 500];%
penalty_para = [600 0.077 1.612 14400 6.7 12.6 ]; %30 min=0.5 hr,

%-------formulate inputs to X, sqX, dW---------------------
X  = decodeX(input1,nVar,nHeat);
X1 = round(X);
eaf_PDI = X1(1:2,:)
lhf_PDI = X1(3:4,:)
cc_PDI = [1 1 1 0 0 0 0 0 1 0 0 0;
          0 0 0 0 0 1 1 1 0 0 0 0;
          0 0 0 1 1 0 0 0 0 0 0 0;
          0 0 0 0 0 0 0 0 0 1 1 1]
[sqX1 aux1] = decodeSeq(input2,nHeat,nProcess-1);
sqEAF = sqX1(1,:)
sqLHF = sqX1(2,:)
sqCC  = [2 1 3 9 7 8 6 4 5 10 11 12]
aux2 dW]= decodeSeq(input3,nHeat,nProcess);
dW

%--------Check constraint-----------------------------------
```

```
for k1=1:nHeat
    if sum(cc_PDI(:,k1))~=1
        check_st1(k1)=0;
    else
        check_st1(k1)=1;
    end

    if sum(eaf_PDI(:,k1))~=1
        check_st2(k1)=0;
    else
        check_st2(k1)=1;
    end

    if sum(lhf_PDI(:,k1))~=1
        check_st3(k1)=0;
    else
        check_st3(k1)=1;
    end
end

for k1=1:nVar
    if sum(X1(k1,:))>1
        check_st4(k1)=0;
    else
        check_st4(k1)=1;
    end
end

if sum(check_st1)==nHeat & sum(check_st2)==nHeat&sum(check_st3)==nHeat
& sum(check_st4)==0

    [br_job, resource, br_duration, br_point]= setBrRandom();
    br_time=[br_job resource 0 0];
    [q_time_c,q_time_el,castbr_time,tDep_cc] =
Steel_Process(eaf_PDI(1,:),lhf_PDI(1,:),cc_PDI,sqEAF,sqLHF,sqCC,dW,br_
time);
    q_time    = q_time_c+q_time_el
    cost_q    = (penalty_para(5)*q_time_c)+(penalty_para(5)*q_time_el)

    if castbr_time < 10
        cost_br = 0;
    else
        cost_br = penalty_para(4)*0.6*castbr_time;
    end

    for i1=1:nCast
        DueDate = cc_PDI(i1,:).*iniPlan(4,:);
        Late_per_cast(i1)= sum(max(0,tDep_cc(i1,:)-DueDate));
    end
    cost_late = sum(Late_per_cast)*penalty_para(6);
    Qc = costQ(iniPlan,penalty_para,2);
    Qe = costQ(iniPlan,penalty_para,1);

    for k1=1:nCast
        cost_each_cast1(k1) = 0.5*sum(cc_PDI(k1,:)*Qc*cc_PDI(k1,:)');
    end

    for k1=1:2
        cost_each_cast2(k1) =
0.5*sum(eaf_PDI(k1,:)*Qe*eaf_PDI(k1,:)');
```

```
    end

Efficiency = sum(cost_each_cast1)+sum(cost_each_cast2) + cost_q +
cost_br + cost_late;
    z = Efficiency

else
    z=20*penalty_para(4)
    Efficiency =20*penalty_para(4);
    q_time=20*penalty_para(4);
    castbr_time=20*penalty_para(4);
end

disp('----------------------------');

%*************************************************************
%                  subfunction
%*************************************************************
function y = costQ(iniPlan,penalty_para,noMachine)
nHeat=size(iniPlan,2);
for k1=1:nHeat
    for k2=1:nHeat
        if k1==k2
            Q(k1,k2)=0;
        else
            if abs(iniPlan(1,k1)-iniPlan(1,k2))==0
                c_g = 0;
            elseif abs(iniPlan(1,k1)-iniPlan(1,k2))>50
                c_g = penalty_para(4)/10;%loss 1 slab, not 1 heat
            else
                c_g = penalty_para(1);
            end

            if noMachine==1
                Q(k1,k2) = c_g/3;
            else
                c_w  = penalty_para(2)*abs(iniPlan(2,k1)-
iniPlan(2,k2));
                c_th = penalty_para(3)*abs(iniPlan(3,k1)-
iniPlan(3,k2));
                Q(k1,k2) = c_g + c_w + c_th;
            end
        end
    end
end
y = Q;
%*************************************************************
function y=findCasts(pCast,iniPlan)
for i=1:size(pCast,2)
    y(i)=iniPlan(pCast(i));
end
%*************************************************************
function X_decoded = decodeX(input, nVar, nHeat)
k3=0;
for k1=1:nVar
    for k2=1:nHeat
        k3=k3+1;
        newPDI(k1,k2)=input(k3);
    end
end
```

```
X_decoded=newPDI;
%*************************************************************
function [br_job, resource, br_duration, br_point]= setBrRandom()

resource = 1+floor((rand(1,1)*10)*3/10);%random machine (EAF,LHF,CT)
    if resource==1
        br_pos_in_heat = 1 + 10.8*betarnd(0.59,0.57);
        br_duration = round(5 + 38*betarnd(3.75,1.15));
    elseif resource==2
        br_pos_in_heat = 1 + 10.7*betarnd(0.57,0.71);
        br_duration = round(5 + 38*betarnd(3.75,1.15));
    else
        br_pos_in_heat = 1 + 11.8*betarnd(0.72,0.70);
        br_duration = round(5 + 38*betarnd(3.75,1.15));
    end
    br_job   = floor(br_pos_in_heat);
    br_point = round((br_pos_in_heat-br_job)*100);
%*************************************************************
function [x_new,x_new2] = decodeSeq(x,nHeat,nRow)
iniX=[1 2 3 4 5 6 7 8 9 10 11 12];
y=iniX;
x_decoded = decodeB2Mat(x,nRow,nHeat,4);
x_decoded = ceil((x_decoded+1)*nHeat/17);
x_new2= x_decoded;
for k2=1:nRow
    x_row = x_decoded(k2,:);
    for k1=1:nHeat
        pSq=x_row(k1);
        index=[k1,pSq];
        buff1=y(k1);
        y(k1)=iniX(pSq);
        y(pSq)=buff1;
        iniX=y;
    end
    x_new(k2,:)= y;
end
%-------------------------------------------------------------
function X_decoded = decodeB2Mat(input, nRow, nHeat,resInput)
k3=0;
for k1=1:nRow
    for k2=1:nHeat*resInput
        k3=k3+1;
        binaryMat(k1,k2)=input(k3);%devide long row (input1) to matrix
    end
end

for k1=1:nRow
    st_point=1;
    end_point=1;
    for k2=1:nHeat
        end_point=k2*resInput;
        str_vec=binaryMat(k1,st_point:end_point);
        bin_str=int2str(str_vec);
        dec_str=bin2dec(bin_str);
        dec_mat(k1,k2)=dec_str;
        st_point=end_point+1;
    end
end
X_decoded =  dec_mat;
%*************************************************************
```

File: Steel_Process.m

```
function [totalQ_ct,totalQ_EafLhf,totalCBr,tDep_cc] =
Steel_Process(eaf1_PDI,lhf1_PDI,casts_PDI,sqEAF,sqLHF,sqCC,dW,br_time)
%-------Initial setting-----------------------
time_ratio=200;
if br_time(2)==1
   setBr=[br_time(1) br_time(3) br_time(4); 0 0 0; 0 0 0];
elseif br_time(2)==2
   setBr=[0 0 0; br_time(1) br_time(3) br_time(4); 0 0 0];
else
   setBr=[0 0 0; 0 0 0; br_time(1) br_time(3) br_time(4)];
end

tArr_eaf1 = 0;
tArr_eaf2 = 0;
tDep_eaf1_old = tArr_eaf1;
tDep_eaf2_old = tArr_eaf2;
cntLHF1=0;
cntLHF2=0;
cntCT1 =0;
cntCT2 =0;
cntCast1=0;
cntCast2=0;
cntCast3=0;
cntCast4=0;
%-------Run Simulation--------------------------

for job=1:12
    jobEaf=sqEAF(job);
    if eaf1_PDI(jobEaf)==1
    %--eaf1 run---------------------
        tQue_eaf1(jobEaf) = max(0,(tDep_eaf1_old -
tArr_eaf1))+dW(1,jobEaf);
        [Ser_time_eaf1, tempoVar] =
GetSurviceTime(jobEaf,1,setBr(1,1),setBr(1,2),setBr(1,3));
        tSer_eaf1(jobEaf) = Ser_time_eaf1*time_ratio;
        tDep_eaf1(jobEaf) = tArr_eaf1 + tQue_eaf1(jobEaf) +
tSer_eaf1(jobEaf);

        tDep_eaf1_old = tDep_eaf1(jobEaf);
        tArr_eaf1     = tDep_eaf1_old;
        tDep_eaf2(jobEaf)= 0;
        tQue_eaf2(jobEaf)= 0;
        tSer_eaf2(jobEaf)= 0;
    else
    %--eaf2 run---------------------
        tQue_eaf2(jobEaf) = max(0,(tDep_eaf2_old -
tArr_eaf2))+dW(1,jobEaf);
        [Ser_time_eaf2, tempoVar] =
GetSurviceTime(jobEaf,1,setBr(1,1),setBr(1,2),setBr(1,3));
        tSer_eaf2(jobEaf) = Ser_time_eaf2*time_ratio;
        tDep_eaf2(jobEaf) = tArr_eaf2 + tQue_eaf2(jobEaf) +
tSer_eaf2(jobEaf);
        tDep_eaf2_old = tDep_eaf2(jobEaf);
        tArr_eaf2     = tDep_eaf2_old;
        tDep_eaf1(jobEaf)= 0;
        tQue_eaf1(jobEaf)= 0;
        tSer_eaf1(jobEaf)= 0;
    end
```

```
end

for job=1:12
    jobLhf=sqLHF(job);
    if lhf1_PDI(jobLhf)==1
    %--lhf1 run----------------------
        if eaf1_PDI(jobLhf)==1
            tArr_lhf1 = tDep_eaf1(jobLhf);
        else
            tArr_lhf1 = tDep_eaf2(jobLhf);
        end
        cntLHF1=cntLHF1+1;
        if cntLHF1==1
            tDep_lhf1_old = tArr_lhf1;
        else
            tDep_lhf1_old = tDep_lhf1_old;
        end
        tQue_lhf1(jobLhf)  = max(0,(tDep_lhf1_old -
tArr_lhf1))+dW(2,jobLhf);
        tIdle_lhf1(jobLhf) = abs(min(0,(tDep_lhf1_old - tArr_lhf1)));
        [Ser_time_lhf1, tempoVar] =
GetServiceTime(jobLhf,2,setBr(2,1),setBr(2,2),setBr(2,3));
        tSer_lhf1(jobLhf)  = Ser_time_lhf1*time_ratio;
        tDep_lhf1(jobLhf)  = tArr_lhf1 + tQue_lhf1(jobLhf) +
tSer_lhf1(jobLhf);
        tDep_lhf1_old = tDep_lhf1(jobLhf);
        tDep_lhf2(jobLhf)= 0;
        tQue_lhf2(jobLhf)= 0;
        tIdle_lhf2(jobLhf)=0;
        tSer_lhf2(jobLhf) =0;
    else
    %--lhf2 run----------------------
        if eaf1_PDI(jobLhf)==1
            tArr_lhf2 = tDep_eaf1(jobLhf);
        else
            tArr_lhf2 = tDep_eaf2(jobLhf);
        end
        cntLHF2=cntLHF2+1;
        if cntLHF2==1
            tDep_lhf2_old = tArr_lhf2;
        else
            tDep_lhf2_old = tDep_lhf2_old;
        end
        tQue_lhf2(jobLhf)  = max(0,(tDep_lhf2_old -
tArr_lhf2))+dW(2,jobLhf);
        tIdle_lhf2(jobLhf) = abs(min(0,(tDep_lhf2_old - tArr_lhf2)));
        [Ser_time_lhf2, tempoVar] =
GetServiceTime(jobLhf,2,setBr(2,1),setBr(2,2),setBr(2,3));
        tSer_lhf2(jobLhf)  = Ser_time_lhf2*time_ratio;
        tDep_lhf2(jobLhf)  = tArr_lhf2 + tQue_lhf2(jobLhf) +
tSer_lhf2(jobLhf);
        tDep_lhf2_old = tDep_lhf2(jobLhf);
        tDep_lhf1(jobLhf)= 0;
        tQue_lhf1(jobLhf)= 0;
        tIdle_lhf1(jobLhf)=0;
        tSer_lhf1(jobLhf) =0;
    end
end

tDep_cc1=0*casts_PDI(1,:);
tDep_cc2=0*casts_PDI(1,:);
```

```
tDep_cc3=0*casts_PDI(1,:);
tDep_cc4=0*casts_PDI(1,:);

tQue_cc1=0*casts_PDI(1,:);
tQue_cc2=0*casts_PDI(1,:);
tQue_cc3=0*casts_PDI(1,:);
tQue_cc4=0*casts_PDI(1,:);

tCastbreak1=0*casts_PDI(1,:);
tCastbreak2=0*casts_PDI(1,:);
tCastbreak3=0*casts_PDI(1,:);
tCastbreak4=0*casts_PDI(1,:);

tSer_cc1=0*casts_PDI(1,:);
tSer_cc2=0*casts_PDI(1,:);
tSer_cc3=0*casts_PDI(1,:);
tSer_cc4=0*casts_PDI(1,:);

Caster_setuptime=20;

%-------------CAST#1----------------------------
[activHeat_cast1,pSqActHeat_cast1]=
findPos_inCast(casts_PDI(1,:),sqCC);
nActHCast=size(activHeat_cast1,2);
for i=1:nActHCast

    job_incast = activHeat_cast1(pSqActHeat_cast1(i));
    if lhf1_PDI(job_incast)==1
       tArr_cc1 = tDep_lhf1(job_incast);
    else
       tArr_cc1 = tDep_lhf2(job_incast);
    end

    cntCast1=cntCast1+1;
    if cntCast1==1
       tDep_cc1_old = tArr_cc1;
    else
       tDep_cc1_old = tDep_cc1_old;
    end

    tQue_cc1(job_incast)  = max(0,(tDep_cc1_old -
tArr_cc1))+dW(3,job_incast);
    tCastbreak1(job_incast) = abs(min(0,(tDep_cc1_old - tArr_cc1)));
    [Ser_time_cc1, tempoVar] =
GetServiceTime(job_incast,3,setBr(3,1),setBr(3,2),setBr(3,3));
    tSer_cc1(job_incast)  = Ser_time_cc1*time_ratio;
    tDep_cc1(job_incast)  = tArr_cc1 + tQue_cc1(job_incast) +
tSer_cc1(job_incast);
    tDep_cc1_old = tDep_cc1(job_incast);
    tDep_cc2(job_incast)   = 0;
    tQue_cc2(job_incast)   = 0;
    tCastbreak2(job_incast)= 0;
    tSer_cc2(job_incast)   = 0;
    tDep_cc3(job_incast)   = 0;
    tQue_cc3(job_incast)   = 0;
    tCastbreak3(job_incast)= 0;
    tSer_cc4(job_incast)   = 0;
    tDep_cc4(job_incast)   = 0;
    tQue_cc4(job_incast)   = 0;
    tCastbreak4(job_incast)= 0;
```

```
    tSer_cc4(job_incast)   = 0;

end

%-------------CAST#2----------------------------
[activHeat_cast2,pSqActHeat_cast2]=
findPos_inCast(casts_PDI(2,:),sqCC);
nActH_inCast=size(activHeat_cast2,2);
for i=1:nActH_inCast
    job_incast = activHeat_cast2(pSqActHeat_cast2(i));
    if lhf1_PDI(job_incast)==1
       tArr_cc2 = tDep_lhf1(job_incast);
    else
       tArr_cc2 = tDep_lhf2(job_incast);
    end
    cntCast2=cntCast2+1;
    if cntCast2==1
       tDep_cc2_old = tDep_cc1_old + Caster_setuptime;
    else
       tDep_cc2_old = tDep_cc2_old;
    end

    tQue_cc2(job_incast)  = max(0,(tDep_cc2_old -
tArr_cc2))+dW(3,job_incast);
    tCastbreak2(job_incast) = abs(min(0,(tDep_cc2_old - tArr_cc2)));
    [Ser_time_cc2, tempoVar] =
GetSurviceTime(job_incast,3,setBr(3,1),setBr(3,2),setBr(3,3));
    tSer_cc2(job_incast)  = Ser_time_cc2*time_ratio;
    tDep_cc2(job_incast)  = tArr_cc2 + tQue_cc2(job_incast) +
tSer_cc2(job_incast);
    tDep_cc2_old = tDep_cc2(job_incast);
    tDep_cc1(job_incast)   = 0;
    tQue_cc1(job_incast)   = 0;
    tCastbreak1(job_incast)= 0;
    tSer_cc1(job_incast)   = 0;
    tDep_cc3(job_incast)   = 0;
    tQue_cc3(job_incast)   = 0;
    tCastbreak3(job_incast)= 0;
    tSer_cc3(job_incast)   = 0;
    tDep_cc4(job_incast)   = 0;
    tQue_cc4(job_incast)   = 0;
    tCastbreak4(job_incast)= 0;
    tSer_cc4(job_incast)   = 0;
end

%-------------CAST#3----------------------------
[activHeat_cast3,pSqActHeat_cast3]=
findPos_inCast(casts_PDI(3,:),sqCC);
nActH_inCast=size(activHeat_cast3,2);
for i=1:nActH_inCast
    job_incast = activHeat_cast3(pSqActHeat_cast3(i));
    if lhf1_PDI(job_incast)==1
       tArr_cc3 = tDep_lhf1(job_incast);
    else
       tArr_cc3 = tDep_lhf2(job_incast);
    end
    cntCast3=cntCast3+1;
    if cntCast3==1
       tDep_cc3_old = tArr_cc3;
    else
       tDep_cc3_old = tDep_cc3_old;
```

```
    end

    tQue_cc3(job_incast)  = max(0,(tDep_cc3_old -
tArr_cc3))+dW(3,job_incast);
    tCastbreak3(job_incast) = abs(min(0,(tDep_cc3_old - tArr_cc3)));
    [Ser_time_cc3, tempoVar] =
GetServiceTime(job_incast,3,setBr(3,1),setBr(3,2),setBr(3,3));
    tSer_cc3(job_incast)  = Ser_time_cc3*time_ratio;
    tDep_cc3(job_incast)  = tArr_cc3 + tQue_cc3(job_incast) +
tSer_cc3(job_incast);
    tDep_cc3_old = tDep_cc3(job_incast);
    tDep_cc1(job_incast)   = 0;
    tQue_cc1(job_incast)   = 0;
    tCastbreak1(job_incast)= 0;
    tSer_cc1(job_incast)   = 0;
    tDep_cc2(job_incast)   = 0;
    tQue_cc2(job_incast)   = 0;
    tCastbreak2(job_incast)= 0;
    tSer_cc2(job_incast)   = 0;
    tDep_cc4(job_incast)   = 0;
    tQue_cc4(job_incast)   = 0;
    tCastbreak4(job_incast)= 0;
    tSer_cc4(job_incast)   = 0;
end

%-------------CAST#4----------------------------
[activHeat_cast4,pSqActHeat_cast4]=
findPos_inCast(casts_PDI(4,:),sqCC);
nActH_inCast=size(activHeat_cast4,2);
for i=1:nActH_inCast
    job_incast = activHeat_cast4(pSqActHeat_cast4(i));
    if lhf1_PDI(job_incast)==1
       tArr_cc4 = tDep_lhf1(job_incast);
    else
       tArr_cc4 = tDep_lhf2(job_incast);
    end

    cntCast4=cntCast4+1;
    if cntCast4==1
       tDep_cc4_old = tDep_cc3_old+ Caster_setuptime;
    else
       tDep_cc4_old = tDep_cc4_old;
    end

    tQue_cc4(job_incast)  = max(0,(tDep_cc4_old -
tArr_cc4))+dW(3,job_incast);
    tCastbreak4(job_incast) = abs(min(0,(tDep_cc4_old - tArr_cc4)));
    [Ser_time_cc4, tempoVar] =
GetServiceTime(job_incast,3,setBr(3,1),setBr(3,2),setBr(3,3));
    tSer_cc4(job_incast)  = Ser_time_cc4*time_ratio;
    tDep_cc4(job_incast)  = tArr_cc4 + tQue_cc4(job_incast) +
tSer_cc4(job_incast);
    tDep_cc4_old = tDep_cc4(job_incast);
    tDep_cc1(job_incast)   = 0;
    tQue_cc1(job_incast)   = 0;
    tCastbreak1(job_incast)= 0;
    tSer_cc1(job_incast)   = 0;
    tDep_cc2(job_incast)   = 0;
    tQue_cc2(job_incast)   = 0;
    tCastbreak2(job_incast)= 0;
    tSer_cc2(job_incast)   = 0;
```

```
    tDep_cc3(job_incast)   = 0;
    tQue_cc3(job_incast)   = 0;
    tCastbreak3(job_incast)= 0;
    tSer_cc3(job_incast)   = 0;
end
tQue_eaf=[tQue_eaf1;tQue_eaf2];
tSer_eaf=[tSer_eaf1;tSer_eaf2];
tQue_lhf=[tQue_lhf1;tQue_lhf2];
tIdle_lhf=[tIdle_lhf1;tIdle_lhf2];
tSer_lhf=[tSer_lhf1;tSer_lhf2];
tQue_cc=[tQue_cc1;tQue_cc2;tQue_cc3;tQue_cc4];
tCastbreak=[tCastbreak1;tCastbreak2;tCastbreak3;tCastbreak4];
tSer_cc=[tSer_cc1;tSer_cc2;tSer_cc3;tSer_cc4];
tDep_eaf=[tDep_eaf1;tDep_eaf2];
tDep_lhf=[tDep_lhf1;tDep_lhf2];
tDep_cc=[tDep_cc1;tDep_cc2;tDep_cc3;tDep_cc4];

totalQ_ct1 = sum( tQue_cc(1,:) )+ sum( tQue_cc(2,:) );
totalQ_ct2 = sum( tQue_cc(3,:) )+ sum( tQue_cc(4,:) );
totalQ_ct  = totalQ_ct1 + totalQ_ct2;
totalCBr   = sum(sum(tCastbreak));

totalQ_EafLhf = sum(sum(tQue_eaf))+sum(sum(tQue_lhf));

function [activHeat,pSqActHeat]= findPos_inCast(cast_PDI,sqCC)
activHeat=find(cast_PDI);
sizActH=size(activHeat,2);
[newSqCC,pSqCC]=sort(sqCC);
for i=1:sizActH
    sqActHeat(i)=pSqCC(activHeat(i));
end
[newArrHeat,pSqActHeat]=sort(sqActHeat);
```

File: GetService.m

```
function [servicetime, avgTundWeight] =
GetSurviceTime(arrival,resource,br_job,br_time,br_point)

time_ratio=200;
iniPlan=[100  100  100  110  110  130  130  130  100  001  001  001;
       1594 1594 1594 1594 1570 1570 1570 1570 1570 1564 1564 1564;
         60   60   60   60   60   60   60   60   60   50   50   50;
          4    4    4    4    4    4    4    4    4    4    4    4];
if resource==1
    servicetime_melt=50/time_ratio;
    avgTundWeight=0;
    servicetime=servicetime_melt;
elseif  resource==2
    servicetime_refine=50/time_ratio;
    avgTundWeight=0;
    servicetime=servicetime_refine;
else
    avgTundWeight=180;
servicetime=getservicetimeCasting(arrival,iniPlan,avgTundWeight)/time_
ratio;

end
```

```
if arrival==br_job & br_time>0
    if resource==3
        if br_point>60
           servicetime = (60*servicetime/100)+(br_time/time_ratio);
        else
           servicetime = servicetime+(br_time/time_ratio);
        end
    else
        servicetime = servicetime+(br_time/time_ratio);
    end
else
    servicetime = servicetime;
end

function ser_time_melt = getservicetimeMelting(arrival,iniPlan)
if iniPlan(1,arrival)==001
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
elseif iniPlan(1,arrival)==100
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
elseif iniPlan(1,arrival)==110
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
elseif iniPlan(1,arrival)==120
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
elseif iniPlan(1,arrival)==130
    ser_time_melt=1+unifrnd(-0.2,0.2,1,1);
end

function ser_time_refine = getservicetimeRefining(arrival,iniPlan)
if iniPlan(1,arrival)==001
    ser_time_refine=unifrnd(2,2.5,1,1);
elseif iniPlan(1,arrival)==100
    ser_time_refine=unifrnd(2,2.5,1,1);
elseif iniPlan(1,arrival)==110
    ser_time_refine=unifrnd(2,2.5,1,1);
elseif iniPlan(1,arrival)==120
    ser_time_refine=unifrnd(2,2.5,1,1);
elseif iniPlan(1,arrival)==130
    ser_time_refine=unifrnd(2,2.5,1,1);
end

function ser_time_caster =
getservicetimeCasting(arrival,iniPlan,TundW)
if iniPlan(1,arrival) < 500
    if iniPlan(2,arrival) < 1350
        if iniPlan(3,arrival)==60
            Cast_speed=5.2;
        else
            Cast_speed=5.5;
        end
    elseif (iniPlan(2,arrival) >= 1350)&(iniPlan(2,arrival) <= 1450)
        if iniPlan(3,arrival)==60
            Cast_speed=5.0;
        else
            Cast_speed=5.2;
        end
    elseif iniPlan(2,arrival) > 1450
        if iniPlan(3,arrival)==60
           Cast_speed=4.8;
        else
           Cast_speed=5.0;
```

```
        end
    end
else
     if iniPlan(2,arrival) < 1350
        if iniPlan(3,arrival)==60
            Cast_speed=4.8;
        else
            Cast_speed=5.0;
        end
    elseif (iniPlan(2,arrival) >= 1350)&(iniPlan(2,arrival) <= 1450)
        if iniPlan(3,arrival)==60
            Cast_speed=4.6;
        else
            Cast_speed=4.8;
        end
    elseif iniPlan(2,arrival) > 1450
        if iniPlan(3,arrival)==60
            Cast_speed=4.4;
        else
            Cast_speed=4.6;
        end
    end
end

servicetime_cast =
(TundW*10^6)/(Cast_speed*iniPlan(2,arrival)*iniPlan(3,arrival)*7.6);
ser_time_caster = servicetime_cast;
```

C.3. Code of Robust Scheduling with ANN

List of files:

1. runGA_CDF
2. genXlhd_steel2.m
3. tuneANN_steel.m
4. simANN_steel.m

File: runGA_CDF

```
function
[X,Sq,dW,vec_fval,vec_cdf,vec_mode]=runGA_DisrMCS(nPops,nGens,initGA,i
nitGA_sq,initGA_dW,flagInit)

tStart = tic;
global z_t_old;
z_t_old = zeros(1,400);
fitFunc = @MultiObj_CDF;

%---------Get parameters-------------------------------------
nReschedCast = 8;
nReschedHeat = 12;
nVars = nReschedCast*nReschedHeat;
nVars2 = 3*12;
%---------Define Options Outter Loop-----------------------
options = gaoptimset(@ga);
%options = gaoptimset;
options = gaoptimset(options,'PopInitRange' ,[0 0 0 0;1 1 1 1]);
options = gaoptimset(options,'Generations' ,nGens);
options = gaoptimset(options,'PopulationSize' ,nPops);
options = gaoptimset(options,'StallGenLimit' ,500000);
options = gaoptimset(options,'StallTimeLimit' ,1000000);
options = gaoptimset(options,'EliteCount' ,1);
options = gaoptimset(options,'CrossoverFcn' ,@crossover_Resched);
options.CrossoverFraction=0.8;
options = gaoptimset(options,'MutationFcn' ,{ @mutationuniform 0.01
});
options = gaoptimset(options,'Display' ,'off');
if flagInit==1
    [initGA,initGA_sq,initGA_dW] = genXlhd(nPops,nReschedHeat,2,4);
else
    initGA    = initGA;
    initGA_dW = initGA_dW;
    initGA_sq = initGA_sq;
end

options  = gaoptimset(options,'InitialPopulation' ,initGA);
options2 = gaoptimset(@ga);
options2 = gaoptimset(options2,'Generations' ,nGens);
options2 = gaoptimset(options2,'PopulationSize' ,nPops);
options2 = gaoptimset(options2,'StallGenLimit' ,500000);
options2 = gaoptimset(options2,'StallTimeLimit' ,1000000);
options2 = gaoptimset(options2,'EliteCount' ,1);
options2 = gaoptimset(options2,'CrossoverFcn' ,@crossoversinglepoint);
options2 = gaoptimset(options2,'MutationFcn' ,{ @mutationuniform 0.01
});
options2 = gaoptimset(options2,'Display' ,'off');
```

```
options2 = gaoptimset(options,'InitialPopulation' ,initGA_dW);
options3 = options2;
options3 = gaoptimset(options,'InitialPopulation' ,initGA_sq);

GenomeLengthX   = nVars;
GenomeLength_sq  = nVars2;
GenomeLength_dW = nVars2;
nextPopulationX   = options.InitialPopulation;
nextPopulation_sq = options3.InitialPopulation;
nextPopulation_dW = options2.InitialPopulation;
totalPop = options.PopulationSize;

for pop = 1:totalPop
      [nextScoreX(pop),z_cdf(pop),z_mode(pop)]=fitFunc(nextPopulationX
(pop,:),nextPopulation_sq(pop,:),nextPopulation_dW(pop,:));
end
disp('-----------------------------------------------------');

%--------GA LOOP---------------------------------
maxiters=int16(options.Generations);
for i=1:maxiters
    thisScoreX=nextScoreX;
    thisPopulationX   = nextPopulationX;
    thisPopulation_sq = nextPopulation_sq;
    thisPopulation_dW = nextPopulation_dW;
    nEliteKidX  = options.EliteCount;
    nXoverKidX  = round(options.CrossoverFraction *
(size(thisPopulationX,1) - nEliteKidX));
    nMutateKidX = size(thisPopulationX,1) - nEliteKidX - nXoverKidX;
    nParentX    = 2 * nXoverKidX + nMutateKidX;
    nEliteKid_sq  = options3.EliteCount;
    nXoverKid_sq  = round(options3.CrossoverFraction *
(size(thisPopulation_sq,1) - nEliteKid_sq));
    nMutateKid_sq = size(thisPopulation_sq,1) - nEliteKid_sq -
nXoverKid_sq;
    nParent_sq    = 2 * nXoverKid_sq + nMutateKid_sq;
    nEliteKid_dW  = options2.EliteCount;
    nXoverKid_dW  = round(options2.CrossoverFraction *
(size(thisPopulation_dW,1) - nEliteKid_dW));
    nMutateKid_dW = size(thisPopulation_dW,1) - nEliteKid_dW -
nXoverKid_dW;
    nParent_dW    = 2 * nXoverKid_dW + nMutateKid_dW;
    ExpectationX = fitscalingrank(thisScoreX,nParentX);
    parentX      = selectionroulette(ExpectationX,nParentX,options)
    parentX      = parentX(randperm(length(parentX)));
    [unused,k]   = sort(thisScoreX);
    Expectation_sq = fitscalingrank(thisScoreX,nParent_sq);
    parent_sq = selectionroulette(Expectation_sq,nParent_sq,options3);
    parent_sq = parent_sq(randperm(length(parent_sq)));
    [unused3,k3] = sort(thisScoreX);
    Expectation_dW = fitscalingrank(thisScoreX,nParent_dW);
    parent_dW = selectionroulette(Expectation_dW,nParent_dW,options2);
    parent_dW = parent_dW(randperm(length(parent_dW)));
    [unused2,k2] = sort(thisScoreX);
    eliteKidX  = thisPopulationX(k(1:options.EliteCount),:);
    t1=thisScoreX(k(1))
    xoverKidX  =

crossoverResched(parentX(1:(2*nXoverKidX)),options,GenomeLengthX,fitFu
nc,unused,thisPopulationX);
```

```
    mutateKidX =
mutationuniform(parentX((1+2*nXoverKidX):end),options,GenomeLengthX,fi
tFunc,thisScoreX,thisPopulationX,0.01);
    nextPopulationX = [eliteKidX ; xoverKidX ; mutateKidX];
    eliteKid_sq  = thisPopulation_sq(k3(1:options3.EliteCount),:);
    xoverKid_sq  =
crossoversinglepoint_sq(parent_sq(1:(2*nXoverKid_sq)),options3,GenomeL
ength_sq,fitFunc,unused3,thisPopulation_sq);
    mutateKid_sq =
mutationMatrix_sq(parent_sq((1+2*nXoverKid_sq):end),thisPopulation_sq,
3,12,0.10);
    nextPopulation_sq = [eliteKid_sq ; xoverKid_sq ; mutateKid_sq];
    eliteKid_dW  = thisPopulation_dW(k2(1:options2.EliteCount),:);
    xoverKid_dW  =
crossoversinglepoint(parent_dW(1:(2*nXoverKid_dW)),options2,GenomeLeng
th_dW,fitFunc,unused2,thisPopulation_dW);
    mutateKid_dW =
mutationuniform(parent_dW((1+2*nXoverKid_dW):end),options2,GenomeLengt
h_dW,fitFunc,thisScoreX,thisPopulation_dW,0.01);
    nextPopulation_dW = [eliteKid_dW ; xoverKid_dW ; mutateKid_dW];

for pop = 1:totalPop
[nextScoreX(pop),z_cdf(pop),z_mode(pop)]=fitFunc(nextPopulationX(pop,:
),nextPopulation_sq(pop,:),nextPopulation_dW(pop,:));
end

%-------find answer-----------------------------
    [fval_sort,i_sort] = sort(nextScoreX);
    [fval_cdf,i_sort] = sort(z_cdf);
    [fval_mode,i_sort] = sort(z_mode);
    iterate=i
    x_min(i,:)=nextPopulationX(i_sort(1),:);
    sq_min(i,:)=nextPopulation_sq(i_sort(1),:);
    dw_min(i,:)=nextPopulation_dW(i_sort(1),:);
    fval_min=fval_sort(1)
    status = sprintf('eff= %0.2f ',nextScoreX(i_sort(1)))

%-------Record obj. value in each iters--------
    vec_fval(i) = fval_sort(1);
    vec_cdf(i)  = fval_cdf(1);
    vec_mode(i) = fval_mode(1);
    state.Generation=i;
    state.Score=nextScoreX;

disp('------------------------------------------------------');
disp('                   end iteration                      ');
disp('------------------------------------------------------');

end

X  = x_min(end,:);
Sq = sq_min(end,:);
dW = dw_min(end,:);
FVAL = vec_fval(end);
toc(tStart);

function [expectation] = fitscalingrank(scores,nParents)
nParents;
[unused,i] = sort(scores);
expectation = zeros(size(scores));
```

```
expectation = 1./((1:length(scores)).^ 0.5);
expectation = nParents*expectation./ sum(expectation);

function parents = selectionroulette(expectation,nParents,options)
wheel = cumsum(expectation) / nParents;
parents = zeros(1,nParents);
for i = 1:nParents
    r = rand;
    for j = 1:length(wheel)
        if(r < wheel(j))
            parents(i) = j;
            break;
        end
    end
end

function xoverKids  =
crossoversinglepoint(parents,options,GenomeLength,FitnessFcn,unused,th
isPopulation)
nKids = length(parents)/2;
xoverKids = zeros(nKids,GenomeLength);
index = 1;
for i=1:nKids
    parent1 = thisPopulation(parents(index),:);
    index = index + 1;
    parent2 = thisPopulation(parents(index),:);
    index = index + 1;
    xOverPoint = ceil(rand * (length(parent1) - 1));
    xoverKids(i,:) = [ parent1(1:xOverPoint),parent2(( xOverPoint + 1
):  end )  ];
end

function mutationChildren =
mutationuniform(parents,options,GenomeLength,FitnessFcn,thisScore,this
Population,mutationRate)
if(nargin < 8)
    mutationRate = 0.01;
end
mutationChildren = zeros(length(parents),GenomeLength);
for i=1:length(parents)
    child = thisPopulation(parents(i),:);
    mutationPoints = find(rand(1,length(child)) < mutationRate);
    child(mutationPoints) = ~child(mutationPoints);
    mutationChildren(i,:) = child;
end

function state = gaplotbestf(options,state,flag)
if(flag==1)
    set(gca,'xlim',[1,options.Generations]);
    xlabel Generation
    ylabel('Fitness value');
end
hold on;
generation = state.Generation;
best = min(state.Score);
plot(generation,best, '.k');
m = mean(state.Score);
plot(generation,m,'.b');
title(['Best: ',num2str(best),' Mean: ',num2str(m)])
hold off;
```

```
function xoverKids  =
crossoverResched(parents,options,GenomeLength,FitnessFcn,unused,thisPo
pulation)
nCast=8
nHeat=12;
nKids = length(parents)/2;
xoverKids = zeros(nKids,GenomeLength);
index = 1;
for i=1:nKids
    parent1 = thisPopulation(parents(index),:);
    index = index + 1;
    parent2 = thisPopulation(parents(index),:);
    index = index + 1;
    k3=0;
    for k1=1:nCast
        for k2=1:nHeat
            k3=k3+1;
            mat_parent1(k1,k2)=parent1(k3);
            mat_parent2(k1,k2)=parent2(k3);
        end
    end
    mat_parent1;
    mat_parent2;
    xOverPoint = ceil(rand*nHeat);
    mat_xoverKids =
[mat_parent1(:,1:xOverPoint),mat_parent2(:,(xOverPoint+1):end)];
    k3=0;
    for k1=1:nCast
        for k2=1:nHeat
            k3=k3+1;
            xoverKids(i,k3)=mat_xoverKids(k1,k2);
        end
    end
end

function xoverKids  =
crossoversinglepoint_sq(parents,options,GenomeLength,FitnessFcn,unused
,thisPopulation)

nKids = length(parents)/2;
xoverKids = zeros(nKids,GenomeLength);
index = 1;

for i=1:nKids
    parent1 = thisPopulation(parents(index),:);
    index = index + 1;
    parent2 = thisPopulation(parents(index),:);
    index = index + 1;
    xOverPoint = (ceil(rand * (3 - 1)))*12; %random between 12,24
    xoverKids(i,:) = [ parent1(1:xOverPoint),parent2(( xOverPoint + 1
):  end )  ];
end

function mutationChildren =
mutationMatrix_sq(parents,thisPopulation,nProcess,nHeat,mutationRate)
mutationChildren = zeros(length(parents),nProcess*nHeat);
pRandom=round(rand(1)*100);
for k4=1:length(parents)
    if pRandom <=round(100*mutationRate)
        selectedPop  = thisPopulation(parents(k4),:);
        row_selected = randperm(3);
```

```
        rowMut  = row_selected(1);
        col_selected = randperm(nHeat);
        colMut1 = col_selected(1);
        colMut2 = col_selected(2);
        k3=0;
        for k1=1:nProcess
            for k2=1:nHeat
                k3=k3+1;
                mat_pop(k1,k2)=selectedPop(k3);
            end
        end
        mat_pop_new = mat_pop;
        mat_pop_new(rowMut,colMut1)= mat_pop(rowMut,colMut2);
        mat_pop_new(rowMut,colMut2)= mat_pop(rowMut,colMut1);
        k3=0;
        for k1=1:nProcess
            for k2=1:nHeat
                k3=k3+1;
                vec_pop_new(k3)=mat_pop_new(k1,k2);
            end
        end
        mutationChildren(k4,:)=vec_pop_new;
    else
        mutationChildren(k4,:)=thisPopulation(parents(k4),:);
    end
end
```

File: genXlhd_steel2.m

```
function [initial_set]=genXlhd_steel2(initial_set_old,nSample)

for i=1:size(initial_set_old.vec_X(1:nSample,:),1)
    input1 = initial_set_old.vec_X(i,:);
    nVar   = 8;
    nHeat  = 12;
    X0 = decodeX(input1,nVar,nHeat);
    X1 = round(X0);
    X2 = X1';
    for j=1:size(X2,1)
        X_real(i,j)= bi2de(X2(j,:),'left-msb');
    end
end

for k=1:nSample
    X  = initial_set_old.vec_X(k,:);
    Sq = initial_set_old.vec_Sq(k,:);
    dW = initial_set_old.vec_dW(k,:);
    [St_all] = ObjxQx_New4(X,Sq,dW);
    St_all_pro(k,:)=[St_all(1,:), St_all(2,:), St_all(3,:)];
end

initial_set=initial_set_old;
initial_set.new_ip_Xreal= X_real;
initial_set.new_ip_St=St_all_pro;

function [St_all_pro]=ObjxQx_New4(input1,input2,input3)
```

```
dim_ip   = size(input1);
nVar     = 8;
nProcess = 3;
nHeat    = 12;
nCast    = 4;

iniPlan = [100  100  100  110  110  130  130  130  100  001  001  001;
           1272 1258 1252 1230 1230 1272 1552 1245 1240 1272 1255 1275
           60   60   50   60   60   60   50   50   60   60   60   50;
           320  320  320  320  320  350  420  420  450  450  480  500];
penalty_para = [600 0.077 1.612 14400 6.7 12.6 ];
X  = decodeX(input1,nVar,nHeat);
X1 = round(X);
eaf_PDI = X1(1:2,:);
lhf_PDI = X1(3:4,:);
cc_PDI  = X1(5:8,:);
sqX1  = decodeX(input2,nProcess,nHeat);
sqEAF = sqX1(1,:);
sqLHF = sqX1(2,:);
sqCC  = sqX1(3,:);
dW  = decodeX(input3,nProcess,nHeat

for k1=1:nHeat
    if sum(cc_PDI(:,k1))~=1
        check_st1(k1)=0;
    else
        check_st1(k1)=1;
    end

    if sum(eaf_PDI(:,k1))~=1
        check_st2(k1)=0;
    else
        check_st2(k1)=1;
    end

    if sum(lhf_PDI(:,k1))~=1
        check_st3(k1)=0;
    else
        check_st3(k1)=1;
    end
end

for k1=1:nVar
    if sum(X1(k1,:))>1
        check_st4(k1)=0;
    else
        check_st4(k1)=1;
    end
end

if sum(check_st1)==nHeat & sum(check_st2)==nHeat&sum(check_st3)==nHeat
    [br_job, resource, br_duration, br_point]= setBrRandom();
    br_time=[br_job resource 0 0];
    [q_time_c,q_time_el,castbr_time,tDep_cc,St_all_pro] =
Steel_Process(eaf_PDI(1,:),lhf_PDI(1,:),cc_PDI,sqEAF,sqLHF,sqCC,dW,br_
time);

else
    z=1*penalty_para(4)
    Efficiency =1*penalty_para(4);
```

```
    q_time=1*penalty_para(4);
    castbr_time=1*penalty_para(4);
end

disp('----------------------------');

function X_decoded = decodeX(input, nVar, nHeat)
k3=0;
for k1=1:nVar
    for k2=1:nHeat
        k3=k3+1;
        newPDI(k1,k2)=input(k3);
    end
end
X_decoded=newPDI;

function [br_job, resource, br_duration, br_point]= setBrRandom()
resource = 1+floor((rand(1,1)*10)*3/10);
    if resource==1
        br_pos_in_heat = gamrnd(1.37,3.55);
        if br_pos_in_heat>12
            br_pos_in_heat=12;
        else
            br_pos_in_heat=12;
        end
        br_duration = round(lognrnd(2.39,0.5));
    elseif resource==2
        br_pos_in_heat = gamrnd(1.98,2.17);
        if br_pos_in_heat>12
            br_pos_in_heat=12;
        else
            br_pos_in_heat=12;
        end
        br_duration = round(lognrnd(2.5,0.42));
    else
        br_pos_in_heat = gamrnd(1.64,2.89);
        if br_pos_in_heat>12
            br_pos_in_heat=12;
        else
            br_pos_in_heat=12;
        end
        br_duration = round(normrnd(12.5,4.61));
    end
    br_job   = floor(br_pos_in_heat);
    br_point = round((br_pos_in_heat-br_job)*100);

function [totalQ_ct,totalQ_EafLhf,totalCBr,tDep_cc,St_all_pro]  =
Steel_Process(eaf1_PDI,lhf1_PDI,casts_PDI,sqEAF,sqLHF,sqCC,dW,br_time)

time_ratio=200;
if br_time(2)==1
   setBr=[br_time(1) br_time(3) br_time(4); 0 0 0; 0 0 0];
elseif br_time(2)==2
   setBr=[0 0 0; br_time(1) br_time(3) br_time(4); 0 0 0];
else
   setBr=[0 0 0; 0 0 0; br_time(1) br_time(3) br_time(4)];
end

tArr_eaf1 = 0;
tArr_eaf2 = 0;
```

```
tDep_eaf1_old = tArr_eaf1;
tDep_eaf2_old = tArr_eaf2;
cntLHF1=0;
cntLHF2=0;
cntCT1 =0;
cntCT2 =0;
cntCast1=0;
cntCast2=0;
cntCast3=0;
cntCast4=0;

for job=1:12
    jobEaf=sqEAF(job);
    if eaf1_PDI(jobEaf)==1
       tQue_eaf1(jobEaf) = max(0,(tDep_eaf1_old -
tArr_eaf1))+dW(1,jobEaf);
        [Ser_time_eaf1, tempoVar] =
GetServiceTime(jobEaf,1,setBr(1,1),setBr(1,2),setBr(1,3));
        tSer_eaf1(jobEaf) = Ser_time_eaf1*time_ratio;
        tDep_eaf1(jobEaf) = tArr_eaf1 + tQue_eaf1(jobEaf) +
tSer_eaf1(jobEaf);

        tDep_eaf1_old = tDep_eaf1(jobEaf);
        tArr_eaf1     = tDep_eaf1_old;
        tDep_eaf2(jobEaf)= 0;
        tQue_eaf2(jobEaf)= 0;
        tSer_eaf2(jobEaf)= 0;
    else
        tQue_eaf2(jobEaf) = max(0,(tDep_eaf2_old -
tArr_eaf2))+dW(1,jobEaf);
        [Ser_time_eaf2, tempoVar] =
GetServiceTime(jobEaf,1,setBr(1,1),setBr(1,2),setBr(1,3));
        tSer_eaf2(jobEaf) = Ser_time_eaf2*time_ratio;
        tDep_eaf2(jobEaf) = tArr_eaf2 + tQue_eaf2(jobEaf) +
tSer_eaf2(jobEaf);
       tDep_eaf2_old = tDep_eaf2(jobEaf);
        tArr_eaf2     = tDep_eaf2_old;
        tDep_eaf1(jobEaf)= 0;
        tQue_eaf1(jobEaf)= 0;
        tSer_eaf1(jobEaf)= 0;
    end
end

for job=1:12
    jobLhf=sqLHF(job);
    if lhf1_PDI(jobLhf)==1
       if eaf1_PDI(jobLhf)==1
            tArr_lhf1 = tDep_eaf1(jobLhf);
        else
            tArr_lhf1 = tDep_eaf2(jobLhf);
        end
        cntLHF1=cntLHF1+1;
        if cntLHF1==1

            tDep_lhf1_old = tArr_lhf1;
        else

            tDep_lhf1_old = tDep_lhf1_old;
        end
        tQue_lhf1(jobLhf)  = max(0,(tDep_lhf1_old -
tArr_lhf1))+dW(2,jobLhf);
```

```
        tIdle_lhf1(jobLhf) = abs(min(0,(tDep_lhf1_old - tArr_lhf1)));
        [Ser_time_lhf1, tempoVar] =
GetSurviceTime(jobLhf,2,setBr(2,1),setBr(2,2),setBr(2,3));
        tSer_lhf1(jobLhf)  = Ser_time_lhf1*time_ratio;
        tDep_lhf1(jobLhf)  = tArr_lhf1 + tQue_lhf1(jobLhf) +
tSer_lhf1(jobLhf);
        tDep_lhf1_old = tDep_lhf1(jobLhf);
        tDep_lhf2(jobLhf)= 0;
        tQue_lhf2(jobLhf)= 0;
        tIdle_lhf2(jobLhf)=0;
        tSer_lhf2(jobLhf) =0;
    else
        if eaf1_PDI(jobLhf)==1
            tArr_lhf2 = tDep_eaf1(jobLhf);
        else
            tArr_lhf2 = tDep_eaf2(jobLhf);
        end
        cntLHF2=cntLHF2+1;
        if cntLHF2==1
            tDep_lhf2_old = tArr_lhf2;
        else
            tDep_lhf2_old = tDep_lhf2_old;
        end
        tQue_lhf2(jobLhf)  = max(0,(tDep_lhf2_old -
tArr_lhf2))+dW(2,jobLhf);
        tIdle_lhf2(jobLhf) = abs(min(0,(tDep_lhf2_old - tArr_lhf2)));
        [Ser_time_lhf2, tempoVar] =
GetSurviceTime(jobLhf,2,setBr(2,1),setBr(2,2),setBr(2,3));
        tSer_lhf2(jobLhf)  = Ser_time_lhf2*time_ratio;
        tDep_lhf2(jobLhf)  = tArr_lhf2 + tQue_lhf2(jobLhf) +
tSer_lhf2(jobLhf);
        tDep_lhf2_old = tDep_lhf2(jobLhf);
        tDep_lhf1(jobLhf)= 0;
        tQue_lhf1(jobLhf)= 0;
        tIdle_lhf1(jobLhf)=0;
        tSer_lhf1(jobLhf) =0;
    end
end

tDep_cc1=0*casts_PDI(1,:);
tDep_cc2=0*casts_PDI(1,:);
tDep_cc3=0*casts_PDI(1,:);
tDep_cc4=0*casts_PDI(1,:);
tQue_cc1=0*casts_PDI(1,:);
tQue_cc2=0*casts_PDI(1,:);
tQue_cc3=0*casts_PDI(1,:);
tQue_cc4=0*casts_PDI(1,:);
tCastbreak1=0*casts_PDI(1,:);
tCastbreak2=0*casts_PDI(1,:);
tCastbreak3=0*casts_PDI(1,:);
tCastbreak4=0*casts_PDI(1,:);
tSer_cc1=0*casts_PDI(1,:);
tSer_cc2=0*casts_PDI(1,:);
tSer_cc3=0*casts_PDI(1,:);
tSer_cc4=0*casts_PDI(1,:);

Caster_setuptime=20;

[activHeat_cast1,pSqActHeat_cast1]=
findPos_inCast(casts_PDI(1,:),sqCC);
nActHCast=size(activHeat_cast1,2);
```

```
for i=1:nActHCast
    job_incast = activHeat_cast1(pSqActHeat_cast1(i));
    if lhf1_PDI(job_incast)==1%Check PDI that com from LHF1 or LHF2
       tArr_cc1 = tDep_lhf1(job_incast);
    else
       tArr_cc1 = tDep_lhf2(job_incast);
    end
    cntCast1=cntCast1+1;
    if cntCast1==1
       tDep_cc1_old = tArr_cc1;
    else
       tDep_cc1_old = tDep_cc1_old;
    end

    tQue_cc1(job_incast)  = max(0,(tDep_cc1_old -
tArr_cc1))+dW(3,job_incast);
    tCastbreak1(job_incast) = abs(min(0,(tDep_cc1_old - tArr_cc1)));
    [Ser_time_cc1, tempoVar] =
GetSurviceTime(job_incast,3,setBr(3,1),setBr(3,2),setBr(3,3));
    tSer_cc1(job_incast)  = Ser_time_cc1*time_ratio;
    tDep_cc1(job_incast)  = tArr_cc1 + tQue_cc1(job_incast) +
tSer_cc1(job_incast);

    tDep_cc1_old = tDep_cc1(job_incast);
    tDep_cc2(job_incast)   = 0;
    tQue_cc2(job_incast)   = 0;
    tCastbreak2(job_incast)= 0;
    tSer_cc2(job_incast)   = 0;
    tDep_cc3(job_incast)   = 0;
    tQue_cc3(job_incast)   = 0;
    tCastbreak3(job_incast)= 0;
    tSer_cc4(job_incast)   = 0;
    tDep_cc4(job_incast)   = 0;
    tQue_cc4(job_incast)   = 0;
    tCastbreak4(job_incast)= 0;
    tSer_cc4(job_incast)   = 0;
end

activHeat_cast2,pSqActHeat_cast2]=
findPos_inCast(casts_PDI(2,:),sqCC);
nActH_inCast=size(activHeat_cast2,2);
for i=1:nActH_inCast
    job_incast = activHeat_cast2(pSqActHeat_cast2(i));
    if lhf1_PDI(job_incast)==1
       tArr_cc2 = tDep_lhf1(job_incast);
    else
       tArr_cc2 = tDep_lhf2(job_incast);
    end

    cntCast2=cntCast2+1;
    if cntCast2==1
       tDep_cc2_old = tDep_cc1_old + Caster_setuptime;
    else
       tDep_cc2_old = tDep_cc2_old;
    end

    tQue_cc2(job_incast)  = max(0,(tDep_cc2_old -
tArr_cc2))+dW(3,job_incast);
    tCastbreak2(job_incast) = abs(min(0,(tDep_cc2_old - tArr_cc2)));
```

```
    [Ser_time_cc2, tempoVar] =
GetSurviceTime(job_incast,3,setBr(3,1),setBr(3,2),setBr(3,3));
    tSer_cc2(job_incast)  = Ser_time_cc2*time_ratio;
    tDep_cc2(job_incast)  = tArr_cc2 + tQue_cc2(job_incast) +
tSer_cc2(job_incast);
    tDep_cc2_old = tDep_cc2(job_incast);
    tDep_cc1(job_incast)   = 0;
    tQue_cc1(job_incast)   = 0;
    tCastbreak1(job_incast)= 0;
    tSer_cc1(job_incast)   = 0;
    tDep_cc3(job_incast)   = 0;
    tQue_cc3(job_incast)   = 0;
    tCastbreak3(job_incast)= 0;
    tSer_cc3(job_incast)   = 0;
    tDep_cc4(job_incast)   = 0;
    tQue_cc4(job_incast)   = 0;
    tCastbreak4(job_incast)= 0;
    tSer_cc4(job_incast)   = 0;
end

[activHeat_cast3,pSqActHeat_cast3]=
findPos_inCast(casts_PDI(3,:),sqCC);
nActH_inCast=size(activHeat_cast3,2);
for i=1:nActH_inCast
    job_incast = activHeat_cast3(pSqActHeat_cast3(i));
    if lhf1_PDI(job_incast)==1%Check PDI that com from LHF1 or LHF2
       tArr_cc3 = tDep_lhf1(job_incast);
    else
       tArr_cc3 = tDep_lhf2(job_incast);
    end
    cntCast3=cntCast3+1;
    if cntCast3==1
       tDep_cc3_old = tArr_cc3;
    else
       tDep_cc3_old = tDep_cc3_old;
    end

    tQue_cc3(job_incast)  = max(0,(tDep_cc3_old -
tArr_cc3))+dW(3,job_incast);
    tCastbreak3(job_incast) = abs(min(0,(tDep_cc3_old - tArr_cc3)));
    [Ser_time_cc3, tempoVar] =
GetSurviceTime(job_incast,3,setBr(3,1),setBr(3,2),setBr(3,3));
    tSer_cc3(job_incast)  = Ser_time_cc3*time_ratio;
    tDep_cc3(job_incast)  = tArr_cc3 + tQue_cc3(job_incast) +
tSer_cc3(job_incast);
    tDep_cc3_old = tDep_cc3(job_incast);
    tDep_cc1(job_incast)   = 0;
    tQue_cc1(job_incast)   = 0;
    tCastbreak1(job_incast)= 0;
    tSer_cc1(job_incast)   = 0;
    tDep_cc2(job_incast)   = 0;
    tQue_cc2(job_incast)   = 0;
    tCastbreak2(job_incast)= 0;
    tSer_cc2(job_incast)   = 0;
    tDep_cc4(job_incast)   = 0;
    tQue_cc4(job_incast)   = 0;
    tCastbreak4(job_incast)= 0;
    tSer_cc4(job_incast)   = 0;
end
```

```
[activHeat_cast4,pSqActHeat_cast4]=
findPos_inCast(casts_PDI(4,:),sqCC);
nActH_inCast=size(activHeat_cast4,2);
for i=1:nActH_inCast
    job_incast = activHeat_cast4(pSqActHeat_cast4(i));
    if lhf1_PDI(job_incast)==1
       tArr_cc4 = tDep_lhf1(job_incast);
    else
       tArr_cc4 = tDep_lhf2(job_incast);
    end
    cntCast4=cntCast4+1;
    if cntCast4==1
       tDep_cc4_old = tDep_cc3_old+ Caster_setuptime;
    else
       tDep_cc4_old = tDep_cc4_old;
    end

    tQue_cc4(job_incast)  = max(0,(tDep_cc4_old -
tArr_cc4))+dW(3,job_incast);
    tCastbreak4(job_incast) = abs(min(0,(tDep_cc4_old - tArr_cc4)));
    [Ser_time_cc4, tempoVar] =
GetServiceTime(job_incast,3,setBr(3,1),setBr(3,2),setBr(3,3));
    tSer_cc4(job_incast)  = Ser_time_cc4*time_ratio;
    tDep_cc4(job_incast)  = tArr_cc4 + tQue_cc4(job_incast) +
tSer_cc4(job_incast);
    tDep_cc4_old = tDep_cc4(job_incast);
    tDep_cc1(job_incast)   = 0;
    tQue_cc1(job_incast)   = 0;
    tCastbreak1(job_incast)= 0;
    tSer_cc1(job_incast)   = 0;
    tDep_cc2(job_incast)   = 0;
    tQue_cc2(job_incast)   = 0;
    tCastbreak2(job_incast)= 0;
    tSer_cc2(job_incast)   = 0;

    tDep_cc3(job_incast)   = 0;
    tQue_cc3(job_incast)   = 0;
    tCastbreak3(job_incast)= 0;
    tSer_cc3(job_incast)   = 0;
end

tQue_eaf=[tQue_eaf1;tQue_eaf2];
tSer_eaf=[tSer_eaf1;tSer_eaf2];
tQue_lhf=[tQue_lhf1;tQue_lhf2];
tIdle_lhf=[tIdle_lhf1;tIdle_lhf2];
tSer_lhf=[tSer_lhf1;tSer_lhf2];
tQue_cc=[tQue_cc1;tQue_cc2;tQue_cc3;tQue_cc4];
tCastbreak=[tCastbreak1;tCastbreak2;tCastbreak3;tCastbreak4];
tSer_cc=[tSer_cc1;tSer_cc2;tSer_cc3;tSer_cc4];
tDep_eaf=[tDep_eaf1;tDep_eaf2];
tDep_lhf=[tDep_lhf1;tDep_lhf2];
tDep_cc=[tDep_cc1;tDep_cc2;tDep_cc3;tDep_cc4];
tDep_eaf_all=tDep_eaf1+tDep_eaf2;
tSer_eaf_all=tSer_eaf1+tSer_eaf2;
St_eaf_all=tDep_eaf_all-tSer_eaf_all;
tDep_lhf_all=tDep_lhf1+tDep_lhf2;
tSer_lhf_all=tSer_lhf1+tSer_lhf2;
St_lhf_all=tDep_lhf_all-tSer_lhf_all;
tDep_cc_all=tDep_cc1+tDep_cc2+tDep_cc3+tDep_cc4;
tSer_cc_all=tSer_cc1+tSer_cc2+tSer_cc3+tSer_cc4;
St_cc_all=tDep_cc_all-tSer_cc_all;
```

```
St_all_pro=[St_eaf_all; St_lhf_all; St_cc_all];

totalQ_ct1 = sum( tQue_cc(1,:) )+ sum( tQue_cc(2,:) );
totalQ_ct2 = sum( tQue_cc(3,:) )+ sum( tQue_cc(4,:) );
totalQ_ct  = totalQ_ct1 + totalQ_ct2;
totalCBr   = sum(sum(tCastbreak));
totalQ_EafLhf = sum(sum(tQue_eaf))+sum(sum(tQue_lhf));

function [activHeat,pSqActHeat]= findPos_inCast(cast_PDI,sqCC)
activHeat=find(cast_PDI);
sizActH=size(activHeat,2);
[newSqCC,pSqCC]=sort(sqCC);
for i=1:sizActH
    sqActHeat(i)=pSqCC(activHeat(i));
end
[newArrHeat,pSqActHeat]=sort(sqActHeat);
```

File: tuneANN_steel.m

```
function
[net,vecPerf]=tuneANN_steel(initial_set,flag_sig,nTuned,flag_con)

nSample=1:1000;
scaling_i1=5;
scaling_i2=5;
scaling_o=3;
p1 = (initial_set.new_ip_Xreal(nSample,:)-120)*scaling_i1/(200-120);
p2 = (initial_set.new_ip_St(nSample,:)-500)*scaling_i2/(2000-500);
p  = [p1 p2]';
t1=(initial_set.vec_mu(nSample)-1000000)*scaling_o/(7000000-1000000);
t2=(initial_set.vec_sig(nSample)-13000)*scaling_o/(200000-13000);
if flag_sig==1

net_sig=newff(p,t2,[800,300,1],{'logsig','logsig','purelin'},'trainrp'
);
    net_sig.trainParam.show = 1000;
    net_sig.trainParam.lr = 20;
    net_sig.trainParam.epochs = 1000;
    net_sig.trainParam.goal = 0.0001;
    net_sig.trainParam.max_fail = 40;
    net_sig.performFcn= 'mae';
    for i=1:nTuned
        [net_sig_trained,tr]=train(net_sig,p,t2);
        [Y]=sim(net_sig_trained,p);
        err=Y-t2;
        perf=mae(err);
        str_net_sig{i}=net_sig_trained;
        vecPerf(i)=perf
    end
    [perfmin ix]=min(vecPerf);
    net=str_net_sig{ix};
else
    if flag_con==1

net_mu=newff(p,t1,[800,300,1],{'logsig','logsig','purelin'},'trainrp')
;
```

```
    net_mu.trainParam.show = 1000;
    net_mu.trainParam.lr = 20;
    net_mu.trainParam.epochs = 1000;
    net_mu.trainParam.goal = 0.0001;
    net_mu.trainParam.max_fail = 40;
    net_mu.performFcn= 'mae';
    end
    for i=1:nTuned
        [net_mu_trained,tr]=train(net_mu,p,t1);
        [Y]=sim(net_mu_trained,p);
        err=Y-t1;
        perf2=mae(err);
        str_net_mu{i}=net_mu_trained;
        vecPerf(i)=perf2
    end
    [perfmin ix]=min(vecPerf);
    net=str_net_mu{ix};
end

function X_decoded = decodeX(input, nVar, nHeat)
k3=0;
for k1=1:nVar
    for k2=1:nHeat
        k3=k3+1;
        newPDI(k1,k2)=input(k3);
    end
end
X_decoded=newPDI;
```

File: simANN_steel.m

```
function [sim_result]=simANN_steel(test_set,net)

scaling_i1=5;
scaling_i2=5;
scaling_o=3;
p1 = (test_set.new_ip_Xreal(interval_sample,:)-120)*scaling_i1/(200-
120);
p2 = (test_set.new_ip_St(interval_sample,:)-500)*scaling_i2/(2000-
500);
p  = [p1 p2]'
[vec_Sim]= sim(net,p);
sim_result=vec_Sim;
perf_mae_sig = mae( (test_set.vec_sig(interval_sample)-
13000)*scaling_o/(200000-13000) -sim_result);
perf_mae_mu  = mae( (test_set.vec_mu(interval_sample)-
1000000)*scaling_o/(7000000-1000000) -sim_result)

%*************************************************************
```

CURRICULUM VITAE

Name	Mr. Kiatkajohn Worapradya
Date of Birth	15 May 1978
Educational Record	
Bachelor's Degree	Bachelor of Engineering (Electrical Engineering) King Mongkut's Institute of Technology Ladkrabang (2001)
Master's Degree	Master of Engineering (Electrical Engineering) Chulalongkorn University (2004)
Doctoral Degree	Doctor of Engineering (Integrated Product Design and Manufacturing) King Mongkut's University of Technology Thonburi (2013)
Work Experience	Senior automation engineer *GJ* Steel Public Company. (2004-2010) Lecturer Dhurakij Bundit University (2010-Present)

Publications:

International Journals

1. Worapradya, K. and Thanakijkasem, P., 2014, "A Production Scheduling Methodology in Steelmaking-Continuous Casting Plant Using Hierarchical Genetic Algorithm", **South African Journal of Industrial Engineering**, (accepted).
2. Worapradya, K. and Thanakijkasem, P., 2014, "Proactive Scheduling for Steelmaking-Continuous Casting Plant with Uncertain Machine Breakdown Using Distribution-Based Robustness and Decomposed Artificial Neural Network", **Asia Pacific Journal of Operational Research**, (accepted).

International Conferences

1. Worapradya, K. and Thanakijkasem, P., 2013, "Uncertainty Analysis in Robust Scheduling Problem: Case Study in Steel Making Continuous Casting Plant", **Asian Simulation and Modeling**, 20 January 2013, Bangkok, pp. 55-63.
2. Worapradya, K. and Thanakijkasem, P., 2012, "An Investigation of Robust Optimal Design Using Artificial Neural Network and Genetic Algorithm" **IEEE International Conference on Industrial Engineering and Engineering Management**, 12 December 2012, Hong Kong, pp 75-79.

3. Worapradya, K. and Thanakijkasem, P., 2010, "Worst Case Performance Scheduling facing Uncertainty Disruption in a Continuous Slab Casting Process" **IEEE International Conference on Industrial Engineering and Engineering Management**, 10 December 2010, Macau, pp. 291-295.
4. Worapradya, K. and Thanakijkasem, P., 2009, "Optimum Spray Cooling in Continuous Slab Casting Process under Productivity Improvement", **IEEE International Conference on Industrial Engineering and Engineering Management**, 9 December 2009, Hong Kong, pp. 120-124.
5. Worapradya, K. and Buranathiti, T., 2009, "Production Rescheduling based on Stability under Uncertainty for Continuous Slab Casting", **Asian Simulation and Modeling**, 22 January 2009, Bangkok, pp. 170-178.
6. Worapradya, K. and Buranathiti, T., 2007, "Integration of Manufacturing Execution System and Simulation" **Asian Simulation and Modeling**, 9 January 2007, Chiangmai, pp. 15-21.

National Conferences

1. Worapradya, K. and Thanakijkasem, P., 2009, "An Optimization of Secondary Cooling in Continuous Slab Casting Process Using Solidification Model" **Operations Research Network of Thailand**, 4 September 2009, Bangkok, pp. 234-241. (in Thai)
2. Worapradya, K. and Buranathiti, T., 2008, "Rescheduling based on Stability Assessments for Continuous Slab Casting" **Operations Research Network of Thailand**, 24 July 2008, Bangkok. (in Thai)
3. Worapradya, K. and Buranathiti, T., 2008, "Modeling and Optimum Production Scheduling for Continuous Slab Casting" **Modeling and Simulation Conference**, 26 Aug 2007, Bangkok, pp. 245-254. (in Thai)
4. Worapradya, K. and Buranathiti, T., 2007, "Production Scheduling Optimization for a 2 Line Continuous Slab Casting Process" **Operations Research Network of Thailand**, 6 September 2007, Bangkok, pp. 356-363. (in Thai)