PRODUCTION SCHEDULING IN INTEGRATED STEEL MANUFACTURING

by

Wanxi Li

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Management Science

at

The University of Wisconsin-Milwaukee

August 2014

ABSTRACT
PRODUCTION SCHEDULING IN INTEGRATED STEEL MANUFACTURING

by

Wanxi Li

The University of Wisconsin-Milwaukee, 2014
Under the Supervision of Professor Anthony Ross, Ph.D.

Steel manufacturing is both energy and capital intensive, and it includes multiple production stages, such as iron-making, steelmaking, and rolling. This dissertation investigates the order schedule coordination problem in a multi-stage manufacturing context. A mixed-integer linear programming model is proposed to generate operational (up to the minute) schedules for the steelmaking and rolling stages simultaneously. The proposed multi-stage scheduling model in integrated steel manufacturing can provide a broader view of the cost impact on the individual stages. It also extends the current order scheduling literature in steel manufacturing from a single-stage focus to the coordinated multi-stage focus. Experiments are introduced to study the impact of problem size (number of order batches), order due time and demand pattern on solution performance. Preliminary results from small data instances are reported. A novel heuristic algorithm, Wind Driven Algorithm (WDO), is explained in detail, and numerical parameter study is presented. Another well-known and effective heuristic approach based on Particle Swarm Optimization (PSO) is used as a benchmark for performance comparison. Both algorithms are implemented to solve the scheduling model. Results show that WDO outperforms PSO for the proposed model on solving large sample data instances. Novel contributions and future research areas are highlighted in the conclusion.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The journey to complete this dissertation and my Ph.D. has been a long process. I would like to take the time to thank those who have helped me so much in this pursuit.

First, and foremost, I want to thank my dissertation chair and academic advisor, Dr. Anthony Ross. Dr. Ross, three years ago, I started this journey that I have no idea where it was leading. Your guidance and insights during my study have shaped me into a better scholar. I want to thank you for your unwavering patience, refined intellect, and your trust and vision. You encouraged me when I faced obstacles, and reminded me to forge ahead when I settled. As a mentor, your attitude in both work and life has influenced me professionally and personally.

I want to thank my minor advisor in Marketing, Dr. Sanjoy Ghose. Dr. Ghose, I have learned so much more from you than marketing. Your truly wisdom in both work and life instilled on me is priceless. I always feel like talking to you about anything, and you always have an open-mind, warm heart, and the real empathy in me. I am so lucky to have you in my Ph.D. study at UW-Milwaukee.

I would like to thank other members of my committee: Dr. Xiang Fang, Dr. Matthew Petering, and Dr. Kaan Kuzu. I learned several major courses from you during my study at UW-Milwaukee, and you helped me to learn and develop in my study and research. With all my heart, I thank you for the help you truly gave but never for any return. I appreciate your friendship and support.

Lastly, I would like to thank my family. My parents give me endless love and are always proud of me. Yiyuan, my wife, and Tiaotiao, my son, you brought me so much joy. We have always been so close to each other in heart. Being with you, there will be only courage, but no fear in life.

CHAPTER 1 INTRODUCTION

The global steel manufacturing industry remains an essential part of the industrial ecosystem, and is also extremely capital and energy intensive. For example, in 2006 the iron and steel industry accounted for 1.4% of primary energy consumption in the U.S., and 13.6% in China. Over the last two hundred years, metallurgists have improved the process and related-material technology in iron and steelmaking to produce better quality steel using more efficient methods. The basic oxygen furnace method produces 350 tons of steel every 39-40 minutes, while a traditional open hearth process requires 10 to 12 hours to produce the same output. As the steel manufacturing technology advances and the industry evolves, large integrated steel plants began to emerge during the late 19th Century, and gradually replaced smaller and special-purpose steel mills during the 20th Century as production scale was leveraged by the newer facilities. The 1980s saw significant merger and acquisition activity among steel manufacturers all over the world as firms pursued the major capacity targets of over 30 million tons. In the U.S., there have been over 20 acquisitions since 1985. This accounted for 30% of domestic steelmaking capacity. In 2007, the top 15 steel manufacturers produced one third of steel volume worldwide. As both the scale and the complexity of steel production have increased dramatically, even greater challenges are now imposed on managing the production operations in these large-scale facilities.

As the steel industry has advanced, so has the need for and importance of more sophisticated approaches to managing and scheduling steel production. Mathematical programming approaches gained attention after the seminal paper by Fabian (1958), who first proposed a model that examined the order batching policies. These policies coordinate multi-stage steelmaking operations. After that, research was being developed to improve the utilization and reduce production cost of a single stage by applying mathematical programming models. A paper by Tang et al. (2001) provides one of the most comprehensive reviews of schedule optimization research in the steel industry. Since then, a second generation of research emerged in which the focus shifted away from studying steelmaking operations in isolation to a focus on understanding how modern integrated steel production plants work and on developing integrated production schedules. In retrospect, the timeline of this shift in focus also coincided with the steel industry consolidation summarized earlier.

Because early researchers evaluated the production stages separately, the usual model objective focused on cost reduction for a single stage without fully exploring the impact on downstream stages. For example, production schedules that optimize the steel melting process to obtain stage-wise lowest unit cost may negatively impact the downstream rolling mill operations or annealing operations since these stages often focus on a different set of production factors. A coordinated production schedule in this case, synchronizes the three production stages and thus improves the efficiency of the overall production process. But as researchers examined the integration of multi-stage steel manufacturing processes, this coordinated scheduling model is critical to capacity utilization and overall cost reduction. Without the guidance of such scheduling model, coordination across different stages may result in operating inefficiently in each stage, and the overall production cost could increase substantially as noted by Lee et al. (1996).

In addition to the operational challenges describing the current state of integrated steel manufacturing, another hurdle some researchers attempted to address was the computational challenge associated with more sophisticated mathematical programming models. For example, Blocher et al. (1999) studied a changeover scheduling problem with product-dependent changeover costs and times. The problem is NP-complete (Non-deterministic Polynomial-time) because the time required to solve the model to optimality increases exponentially as the problem size increases. As the authors noted in their study, even relatively small to medium sized problems could not be solved within a practical time window. As a result, effective solutions to this type of problems often require sophisticated heuristic methods, or very powerful High-Performance Computing (HPCs) systems. Because HPCs are not as generally accessible, heuristic solution techniques have proved to be widely accepted for solving these scheduling problems today. The complex nature of modeling production schedules for most any small or large integrated steel manufacturer means that heuristic approaches represent an important and viable solution approach.

This study makes two important contributions to the literature. First, a production scheduling model for integrated steel manufacturing is proposed, and the model of a two-stage production scheduling process is simultaneously solved based upon our observation of current operations at a large US manufacturer. As the literature shows, this area is still in need of further study. Second, because of the well-known dimensionality curse of the problem that when the dimension (production items or number of production stages) of the problem increases, the size of the solution searching space increases exponentially. We devise a new heuristic approach to solve the proposed model, and offer effective solutions to the integrated scheduling problem. The problem formulation investigated here cannot be easily solved by any industrial grade commercial software package given the scale of larger problem scenarios. As a result, the heuristic must be efficient and can be executed within a short period of time. The rest of this chapter introduces the steel manufacturing process and the research problem examined.

1.1 Overview of Integrated Steel Manufacturing

In general, most integrated steel manufacturing operations usually consist of four production stages: (1) iron making, (2) steelmaking, (3) rolling, and (4) finishing processes. The Iron making stage extracts iron from ores, usually in blast furnace, and the intermediate iron product is called 'pig iron' and it generally contains 4-5% carbon and other impurities. Second, pig iron can be converted into steel during the steelmaking stage where iron is melted in either basic oxygen furnaces or electric arc furnaces. Though carbon and other impurities like sulfur are further removed, some metals such as chromium, manganese, may be added to produce alloy steel. Then the molten steel is cast into desired forms including slabs, ingots, or bars. In the rolling stage (stage three), slabs, ingots or bars are passed through rolling trains along rails, and are formed into various gauges of steel coil, wire, or into thinner slabs. The Finishing stage generally consists of one or more processes such as surface finish, mechanical properties, and coating (Lee et al. 1996). The final form of the finished steel depends on the type of items each individual plant was designed to produce. For example, Vasko et al. (1991) study a U.S. steel plant that produced steel plates. The plates are used to produce railroad cars, ships, or boilers. Dawande et al. (2004) examine a steel plant that produces steel sheets from large slabs. The steel sheets are commonly used in producing household appliances and machines.

Figure 1 (Tang and Wang 2009) illustrates the processes of a China-based integrated steel manufacturer's plant. Iron making, steelmaking and rolling are on the left, and the finishing stage is on the right. An integrated steel manufacturer may operate using some combination of, or all four stages. For example, a plant can be dedicated to producing only steel slabs, and its customers could have their own internal rolling and annealing capability, but none of the first two stages.

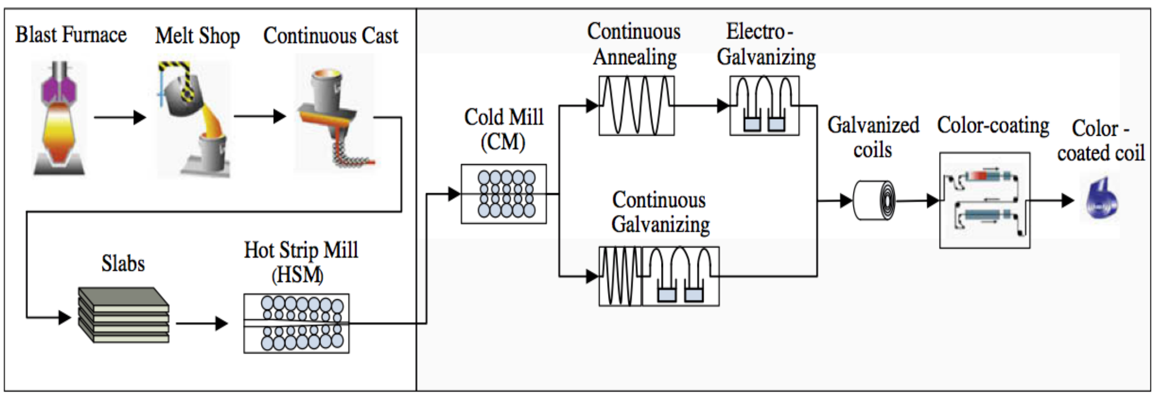

Figure 1: A Steel Manufacturer Plant Illustration

1.2 Motivation and Challenges

The previous section described the primary steelmaking environment (PSM). Another operational orientation found in the industry is referred to as secondary steelmaking (SSM). The secondary steelmaking process is distinguished from PSM in several important ways. First, SSM processes use scrapped steel as the major direct raw material input rather than iron ores as in PSM, and electric arc furnace (EAF) technology is used instead of basic oxygen furnace (BOF). Second, the Iron-making stage is not necessary in

SSM plants. Essentially, recycled or scrapped steel is melted in an EAF, and then cast into slabs or bars before rolling. However, using the less expensive scrap as raw material creates certain cost and quality trade-offs. The chemical impurities that are introduced to the SSM process are more difficult to control because of the wide variety of scrap material delivered to the SSM manufacturer on a daily basis. For example, in PSM iron is extracted from iron ore and impurities like carbon and sulfur can be removed in heating. Other metals can be added as needed. However, in SSM scrap material comes from junk cars, recycled cans, bottles, or recycled packaging and construction materials. In addition to the impurities such as carbon or sulfur, residuals of other metals will remain in the SSM and make the steel quality control much more challenging. Moreover, when the SSM manufacturer also offers its customers various product grades, the optimization of production schedules is further complicated. Therefore in order to better control the chemical composition and to avoid production mismatches, the production scheduler would like to avoid dramatic changes in the chemical composition between successive production sequences. Therefore an effective scheduling mechanism can have significant influence on the production efficiency and capacity utilization.

Another reason that production scheduling research in steel manufacturing plays so significant role is that coordination across the different manufacturing stages is more complex and strategically relevant as the industry structure evolved. Fierce market competition required today's firms to focus on integrated operations optimization instead of single-stage optimization. This is especially the case today given the intense consolidation in the steel industry occurring over the last decade. For SSM operations, the steelmaking and rolling stages often follow completely different planning disciplines:

one based upon on chemical composition, the other on physical dimensions. A scheduling mechanism that reconciles both interests can more effectively meet the company's aggregate requirements. This study is timely and relevant because SSM research on multi-stage scheduling problems is still very new and available studies are limited.

1.3 Research Problem

Production scheduling models using mathematical programming techniques can address various short-term production planning problems, and changeover scheduling represents the type of problem in which production setup times/costs varies depending on product types or production sequences (Blocher et al. 1999). Their model can be applied to different production scenarios such as processing industry (paper, steel manufacturing), or discrete product manufacturing (consumer electronics). More complex model can be developed based on the changeover scheduling structure to reflect complicated production processes, such as more than one production stage and production coordination. In this study, we try to address a production scheduling problem with sequence-dependent production setup times in a multiple production stages scenario. In particular, we formulate a model for a multi-stage production scheduling case in a SSM plant. Though specification of the proposed model is aimed to SSM production environment, our modeling method and solution approach can also be applied to other production scenarios where there are similar production characteristics.

The integrated steel plant structure focused in this study has one EAF used for secondary steelmaking, and one continuous casting facility used for casting molten steel into billets of uniform dimension. In hot rolling, the billets are re-heated before being put onto the rolling rail. A set of roller trains is then set up along the rail, and the steel billets are formed into steel wire according to the configured diameter sizes desired by customers and sequenced by the production scheduler. The end product is reels of industrial-grade steel wire that are characterized by the grade and (diameter) size. The model proposed here focuses on coordinating production schedules for both manufacturing stages, rather than trying to find separate optimal schedules for each operation. Because the throughput rate and operating interests are always found to be disparate between the steelmaking and rolling stages, an integrated schedule scheme provides an alternative, hopefully broader basis for studying the steel production problem here.

Scheduling production of the integrated steel plant is very challenging primarily due to the complexity of each individual production stage, the disparate nature of these stages, and the inherent challenges of secondary steelmaking. There is a rich history of schedule optimization research in the single-stage scenario by Tang et al. (2001), which summarizes scheduling applications in steelmaking and continuous casting processes. However, more recent research has been reported in Chen et al. (2012), Höhn et al. (2012), and Tang et al. (2011), which focus on different production stages and processes. On the other hand, there is very limited literature on multi-stage scheduling and order release. In fact, this stream of literature is only now beginning to appear. For example, Safaei et al. (2010) study a multi-site steel production-distribution planning problem, which focuses the integration of production and distribution. Tang and Gong (2009) present a model for scheduling production, the product transportation and a batch processing subsequently. The study proposed here is motivated by both the steel industry evolution (the evolution of industry practice) and the coordination research beginning to emerge in this area as a result of customer, product, and processing dynamics for SSMs.

The rest of this dissertation is organized as follows. Chapter 2 conducts a literature review. Then a production scheduling model and the data structure developed during the conduct of this research are presented in Chapter 3. Chapter 4 presents an experiment design and the solution experience of solving a series of small problems. Chapter 5 develops meta-heuristic approaches and discusses the implementation of the algorithms. Numerical parameter study and large sample performance comparison are presented in Chapter 6. The last chapter concludes this dissertation and outlines future areas.

CHAPTER 2 LITERATURE REVIEW

This chapter reviews the literature on production order scheduling in the integrated steel manufacturing industry. Based on the early research by Fabian (1958) and two review studies published in 2001 (Tang et al. 2001, Dutta and Fourer 2001), we detail the significant research published after 2001, and segment the research by the steel manufacturing stage to mirror published work. Models and heuristic methods applied in these studies are also summarized and tabulated.

Production scheduling research nesting in the steel industry is well established and dates back some seventy years. However, as both the steelmaking technology and industry structure have evolved so have the modeling and solution approaches used. Fundamentally, the scope of the research problem has progressed from the examination of a single-stage production activity to that of multiple-stage production and inventory coordination decisions. This is due to increasing need to understand the impact of production decisions in multi-stage operating environments rather than in isolation. The expanding problem focus is also the result of significant merger and acquisition activity, which resulted in dramatic increases in production capacity and the operational complexity among surviving firms. The first half of this literature review summarizes the scheduling research according to the focal production stages investigated. More importantly, research on multi-stage operations is reviewed in order to provide a more precise background of the research stream in which this dissertation nests. In the second half of this review, various heuristic techniques reported in the literature and used to solve the related scheduling models are summarized. This review of solution approaches explicates the fit and appropriateness of the selected family of heuristic solution methods devised for the problem investigated here.

2.1 Production Scheduling in Steel Manufacturing

Fabian's (1958) seminal paper used linear programming to study the optimal product mix that results in lowest cost for an integrated iron and steel production plant. For its time, this state-of-art model optimized the different production stages (coke production, iron production, steel production, and rolling and finishing production). Each of the stages was assumed to be linked by the material consumption rate between them. The resulting formulation was a linear programming model, which is to optimize the economical consumption rate of input material at each stage of the steel manufacturing. The solution can satisfy the input and output material flow constraints at each production stage, and achieve the required production output rate. The main research focus in the study was not scheduling per se. Rather it was the modeling technique deployed and the suggestion that future work must broaden the problem scope given the changing industrial landscape. As a result, Fabian's (1958) work inspired a significant amount of subsequent research in the steel industry. The work was also the very first industrial application to integrated steelmaking scenarios involving mathematical programming (MP) techniques.

Since the work of Fabian (1958), research and applications in production scheduling for a single-stage in steel industry grew dramatically as researchers and practitioners examined the details of the production processes, and applied MP techniques to practice. For example, the scheduling problem in casting processes is studied by Vonderembse and Haessler (1982), Box and Herbe (1988). Lopez et al. (1998) study the rolling production schedule in a hot strip mill that produces steel coils from slabs. Tang et al. (2000) examine the steelmaking and casting scheduling in a large integrated steel manufacturer in China. The scheduling research reported between 1950-2000 is reviewed comprehensively by Tang et al. (2001). Dutta and Fourer (2001) study mathematical programming applications in steel industry in general. To recognize these two independent review studies and form the research ground of our paper, we highlight the contributions of the two simultaneous studies in our review.

In the review paper by Dutta and Fourer (2001), applications in steel manufacturing using mathematical programming models are categorized into six different application categories. They include steel-planning models for national economics, product mix models, blending models, scheduling, inventory and distribution models, set-covering applications, and cutting stock problems. Each of these categories adopts different perspectives, study several production stages, and use various approaches. The problem studied in our paper falls into the scheduling category.

During the same time period, Tang et al. (2001) published a review study for planning and scheduling of integrated steel production plants. The study examined specific operational stages including steelmaking, continuous casting, and hot rolling production. The authors also concisely summarize earlier research on scheduling challenges in the continuous casting context and which prevailed during the 1970s. They first outlined the characteristics of integrated steel production for continuous casting process scenarios. Then they introduced the production management systems prevailing in the industry at that time, primarily in Japan. The relevant planning and scheduling models reviewed were grouped by methodological orientation, namely OR-based model, AI-based model, and human-machine coordination. The most influential research in each group was also discussed. The authors conclude their review by identifying some key issues and future research directions. Of them, the multi-stage production scheduling for steelmaking is one of the main issues mentioned.

Given the research trajectory paved by these early researchers, this dissertation proposes to extend the understanding of the multi-stage production scheduling problem based upon collaboration with an SSM producer of industrial grade cable wire. The next part of this discussion reviews literature appearing since 2000, and that is germane to integrated steel manufacturing. The few studies in the multi-stage scheduling context are also discussed in this section.

2.1.1 Scheduling Research: Iron-making, Steelmaking, and Continuous Casting Processes

Naphade et al. (2001) study how to schedule the ingot formation process in steelmaking. A tactical mixed integer programming (MIP) scheduling model proposes which specific ingot batches should be produced from a certain production run (or production heat). It's structurally similar to allocation problems. The model is formulated to be executed on a weekly basis at it tries to balance the cost of wasted production against the total tardiness of all ingots. The solution procedure first decouples the scheduling problem into two levels. Then a neighborhood search heuristic is developed to solve each problem level. The steelmaking process under study is a traditional one as ingots are considered to be intermediate products feeding into a continuous casting process. Tang et al. (2002) also study a continuous casting steelmaking process. The cost trade-off between cast breaks, energy lost during production idle, and earliness/tardiness are all incorporated into the MIP model proposed. The authors use Lagrangian relaxation to decompose the model into sub-problems. Dynamic programming is then used to solve each sub-problem. In the last step, a two-phase heuristic is used to recover the original problem and identify feasible solutions. A similar solution method is detailed in the study by Tang and Xuan (2005).

Denton et al. (2003) study a make-to-stock (MTS) system in steelmaking and casting. The novelty here is the authors' focus on managing the product variety in an integrated steel mill. Their approach is to identify the slabs to produce for MTS in the first step. They then develop production schedule for chosen slabs, but purchase other slabs from outside sources. While the two-stage mathematical model is not presented in their paper, the authors do report on their model implementation at the steel mill where improved inventory utilization resulted.

Huegler and Vasko (2006) propose a machine scheduling model for the steelmaking stage of a meltshop. They discuss the efficacy of meta-heuristic solution procedures to solve the scheduling problem for different working stations that are upstream from continuous casting process. The desired schedule should satisfy the metallurgical optimized sequence in the casting process. Domain-specific heuristics for each process feed into the meta-heuristics, namely generational evolutionary programming, steady state evolutionary programming, and simulated annealing. The benefit of separating the domain-specific heuristics and the meta-heuristics is that either one can be improved or changed without impacting the others. For example, Atighehchian et al. (2009) develop a Hybrid Ant Colony and Non-linear Optimization heuristic (HANO) to determine the routing of jobs between different facilities and the sequence to process jobs. It is shown that the preferred schedule found through HANO would have lowest cost including cast interruption, energy loss, and waiting time penalty in all job cases tested. The authors show that, in 95% of the test cases, HANO also outperforms a competing Genetic Algorithm designed to solve the same problem.

The steelmaking in the continuous casting context is still a widely-used process orientation today. As a result, the scheduling research in this area continues to be reported. However, additional study is needed to understand the impact of effective scheduling across multiple production stages such as hot rolling. With this background, our research focus is to address this schedule coordination problem across production stages.

2.1.2 Scheduling Research: Rolling and Finishing Processes

Rolling processing is another major production stage in steel manufacturing and generally follows the steelmaking stage in most settings. The difference here is that the rolling process must consider the physical dimension of the steel product as the most important determinant of actual production rolling sequence. This decision influences cost and efficiency due to the need for major and minor setups between product sizes and product families. By contrast, the products' chemical composition influences cost and efficiency during the steelmaking stage. This product-dependent setup cost situation shares similar characteristics to the changeover scheduling study by Blocher et al. (1999). In this section, we summarize the 'rolling' production scheduling research appearing since the late 2001. This is the same timeframe as the steelmaking scheduling literature described in the previous section. Our aim is to provide a coherent view of the current state-of-the-art not only for each stage in isolation, but also to identify new research opportunities in multi-stage context.

An early study by Cowling (2003) demonstrated a coil rolling scheduling system. In that study, the author specifically accounted for physical coil dimensions and order due dates when generating cost-efficient schedules. Unfortunately, the mathematical formulation was not provided in the paper so it is not possible to assess specifics of the model. Nevertheless, the work does represent an early successful demonstration of the efficacy of Tabu Search heuristic to solve the model. The study also reported a semi-automatic scheduling procedure that can be easily used to schedule and reschedule production when disruptions occur. Tang and Huang (2007) then study the hot rolling process of seamless steel tube production. They model a flow-shop scheduling scenario described by the classic sequence-dependent setup times between product types. The objective finds the minimal makespan of the rolling batches. The authors first develop a Branch-and-Bound method to solve optimally a small-scale problem. Then they continue by evaluating a two-step heuristic using neighborhood search methods for their large-scale problems. An interactive scheduling procedure is also illustrated to explain how the model and solution method could be applied to the seamless steel tube manufacturer.

In a continuing study of steel tube manufacturing by Wang and Tang (2008) the steel slabs are then batched into a number of production “turns”. Each “turn” is scheduled based on the physical dimension, temperature, and early/late tardiness. The proposed mixed integer programming (MIP) model is solved by similar Tabu Search heuristics which also provide near-optimal solutions for large-scale problems. Their objective function structure is similar to that of the model developed in this proposed dissertation. But our focus here differs in terms of operational context. Recently, Chen et al. (2012) investigated a rolling system similar to Wang and Tang (2008). In the Chen et al. (2012)

context, steel slabs were converted into coils, which are reels of thin and flat steel sheet. The objective function of their model minimized the changeover cost and earliness/lateness penalty while satisfying production rolling constraints. The solution approach included aggregation of production orders into subgroups of the orders using special criteria such as setup/changeover costs and order completion priorities. A modified genetic algorithm is used to solve the order selection and sequencing scheduling problem. The solution is then coupled with extremal optimization as the local improvement algorithm. The solution method consistently generated better solution over simple genetic algorithm alone. Tang et al. (2012) further extended the steel rolling scheduling problem by incorporating stochastic demand and volume-dependent production cost. The model approximates schedules by recommending lot sizes for each planning period. The objective minimizes the combined production and inventory carrying cost via a mixed-integer non-linear (MINLP) formulation. The model is piecewise linearized and solved by the stepwise Lagrangian relaxation heuristics. The authors report near-optimal solutions.

Finishing processes in steel manufacturing are known to vary dramatically between plants due to the uniqueness of the final products. For example, color coating is used to satisfy customers' special color and metal requirements, and annealing is used to enhance the physical and surface quality of the final steel product. Tang and Wang's (2009) study developed a mixed integer nonlinear programming (MINLP) model to investigate the color coating process. The objective minimizes the total penalties incurred by order switching, changeovers of sizes, rollers, and embossing type, delivery tardiness, and setups. A tailored Tabu Search heuristics method is applied to find near-optimal solutions. Two strategies are implemented to improve the Tabu Search performance. The specialized heuristics utilize feasibility checking and coil batching, and are described in detail by the authors.

Tang et al. (2011) present an improved Lagrangian relaxation algorithm for order batching decisions in the steelmaking annealing process. Although this paper does not clearly fall into the scheduling research category, even though the modeling techniques, the solution procedures, and the problem context are closely-related to our proposed research. Höhn et al. (2011) extend Tang and Wang (2009) by investigating a more complex color-coating process described by shuttle coaters. This advanced form of processing addresses the need for changes in steel color, roller changes, and the concurrent setup of rollers and color change. They minimize the makespan of a set of steel coils being planned. The solution method is Genetic Algorithm-based heuristics and the authors report implementing the model at a German steel producer to reduce makespan.

To summarize, scheduling research in rolling and finishing production stages has largely focused on criteria such as changeover cost, earliness/tardiness penalties, batching, and makespan. Many of these modeling criteria have been captured as objective function components, decision variables, and data parameters. They are also reflected in this dissertation. Additionally, the broad array of heuristic methods appearing in this research stream provides strong justification for the solution approach proposed in this study as well. But the fundamental contribution offered to the literature by this research is the examination of coordinated scheduling schemes across two separate production stages. The model developed for this research may provide a better understanding of the overall production decision alternatives and the interplay among the problem parameters and schedule cost. While additional research on the rolling or finishing processes in isolation may improve stage-wise performance, such a singular focus could also impose potential impediments on downstream or upstream steel manufacturing processes. This could likely undermine overall production effectiveness given the current integrated nature of the industry.

2.1.3 Related Multi-stage Scheduling Research

Multi-stage scheduling research in steel manufacturing is scarce. Li and Shang (2001) then use a similar input-output modeling framework to synchronize the material flow between coke production, iron making, steelmaking, and plate/wire rolling. Planning guarantees energy and material requirement for each process, and tries to maximize production value while minimizing environmental emission. The authors consider input material availability between successive stages. By contrast, our proposed study focuses on cost of changeover, earliness/tardiness penalty details.

Tang and Liu (2007) present a production order scheduling system in the same spirit of production coordination across different steel manufacturing stages as our research proposes, but with a different focus. The proposed model routes production orders across several key operations in different steel production processes, and decides the exact starting and finishing date, and the fraction of orders to produce. The objective is to achieve the lowest weighted total completion time of all production orders. Tang and Liu (2007) introduce a combination of Lagrangian relaxation, LP, and subgradient optimization heuristic approach to solve the large-scale problem. They generate promising solutions. While the scope of the model covers operations from iron-making to the last finishing process, cold rolling, it is a much broader focus than study being proposed here. Yet this proposed research offers a novel contribution in that the operating processes are scheduled tactically (daily or hourly). Tactical scheduling implies that a dispatching heuristics is needed for each specific operation to determine the exact starting and ending run time of the production orders. In this sense, the models proposed in previous literature resemble a lotsizing approach rather than a pure scheduling model as in our research.

The second departure from Tang and Liu (2007) is focus on minimizing weighted total order completion time of a complex, high-volume production scenario. The purpose is to achieve better order fulfillment, on time completion, and better customer service level. While in our study, the combination of changeover cost and tardiness penalty drive the performance of the schedule. Vanhoucke and Debels (2009) use a similar modeling approach to schedule the orders across different operations of the steel plant. But they focus on minimizing the production cost, which includes utilization cost, assignment cost, and earliness/lateness cost. They first decompose the problem into machine assignment and machine scheduling sub-problems. Then a greedy local search algorithm is used to find the solution for the model. The study is based on a Belgian steel producer.

Tang and Wang (2011) study production scheduling problem for hot rolling process in steel industry. They formulate a flowshop scheduling model to minimize the makespan of all production jobs. The key element is the job processing time given different batching and sequencing on processing machines, and the objective is to improve the utilization by minimizing the makespan. The flowshop formulation does incorporate multiple machines into the model, but the structure is quite different from changeover scheduling in a multi-stage setting, which is the proposed model structure in this study.

This dissertation follows this research stream of coordinated steel production scheduling, and the problem under study is a tactical rather than strategic orientation. We investigate the exact production starting and ending time at the smallest incremental time unit (the minute level). This is done in order to control the production and related costs more precisely without losing sight of customer order fulfillment requirements. This “tight” control of schedule is considered critical especially when the different production stages have disparate production rates, and the speed of processing is production-batch dependent. For these reason, the research will also extend the literature in the coordinated scheduling area.

2.2 Solution Approach

The solution methods for scheduling model using mathematical programming are mostly heuristics, since the models are often found to be hard to solve. As Blocher et al. (1999) point out, changeover scheduling belongs to NP-hard (Non-deterministic Polynomial-time hard) problems that the time to solve the models increase exponentially with the problem size. And no known algorithms can solve medium to large-scale problems of this type to optimality within practical time. In order to obtain effective solutions within a reasonable time limit, researchers usually develop heuristic methods that find near optimal solutions. In the scheduling literature summarized above, large-scale problem instances are mostly solved through heuristic methods to near optimal, instead of global optimality (Tang et al. 2001, Cowling 2003, Vanhoucke and Debels 2009, etc.)

In order to quickly and efficiently obtain the best solution possible, researchers explore a variety of heuristic approaches to tackle the problems. As a result, we have found almost all heuristic approaches known to date are applied to the scheduling problems in steel manufacturing. Our aim is not to comprehensively review this vast body of heuristic approaches in the literature. Although this is not the aim of the work here, we do summarize the breadth of heuristic methods most closely-related to the proposed study. Table 1 clusters the research literature by each author's solution approach.

Table 1: Taxonomy of Solution Methods

Solution Method	Authors
Lagrangian Relaxation	Tang et al. (2002), Tang and Xuan (2005), Tang and Liu (2007), Tang et al. (2011), Tang et al. (2012)
Branch and Bound	Tang and Huang (2007)
Simulated Annealing	Huegler and Vasko (2006)
Artificial Intelligence	Dawande et al. (2004), Huegler and Vasko (2006), Vanhoucke and Debels (2009)
Greedy Approach	Vanhoucke and Debels (2009)
Neighborhood based heuristics	Naphade et al. (2001), Tang and Huang (2007)
Genetic Algorithm	Höhn et al. (2011), Chen et al. (2012)
Ant Colony Optimization	Atighehchian et al. (2009)
Evolutionary Optimization	Huegler and Vasko (2006)
Tabu Search heuristics	Cowling (2003), Wang and Tang (2008), Tang and Wang (2009)
Particle Swarm Optimization	Tang and Yan (2009), Tang and Wang (2010)

This summary of solution approaches serves for two purposes. First, it provides good overall foundation of the best approaches in the area. Second, as we devise the solution approaches for our own models, it will provide a good sense of the applicability to this two-stage problem and serve as a performance benchmark.

CHAPTER 3 THE MODEL

This chapter discusses the multi-stage scheduling model in detail and highlights the current challenges and research opportunities. Then the scheduling model specification is presented. The model has similar characteristics to the class of vehicle routing problem with time window (VRPTW), and is known to be NP-hard. However, the model proposed in this dissertation is quite different from VRPTW. In addition to the practical production constraints, the multi-stage nature of the model requires that production activity of downstream stage (vehicle) should be scheduled no earlier than upstream stage (vehicle). As a result, the proposed model is an extension of the VRPTW problem, and is also NP-hard. This chapter also illustrates the structure of the steel production batch data used in the scheduling model.

3.1 Problem Description

The problem studied in this research is motivated from our industrial experience with a producer of industrial grade cable wire, who has two billion dollar annual revenue. The steel manufacturer headquarter in this study locates in the mid-western area of the United States, with plant operations throughout the Midwest. The plants are secondary steelmaking facilities in which recycled scrap metal from commercial grade products, such as junk cars, are the primary input raw material that is used to make various grades and sizes of steel wire cable. Electronic arc furnace (EAF) is used for melting recycled scrap materials. Despite the benefit of EAF on energy saving compared to traditional blast furnace (BF) and basic oxygen furnace (BOF), only an estimate of less than 5% tonnage produced in the US applying the technique in 2009. All top five steel producers in China use traditional BF and BOF technique. The EAF steel plant has both environmental and cost benefits, but it also provides challenges in production operations. Impurities in the scraps are hard to control and changes of steel grade in production must be well managed in order to minimize the impact of quality and efficiency. Given the recycled nature of raw materials and the technical specifications of each customer's order, other mixtures of pure metals, such like manganese, chromium, copper, etc., may help to formulate exact mixing/melting recipes. The recipes are engineered to customer specifications and melted during each melting cycle. The liquid metal is poured into a buffering tundish, which can hold the liquid metals for continuous casting process.

The precise grade or quality of steel is monitored during the melt process. Adjustments in the precise grade of molten steel can be made on-demand to match the required quality range of all assigned production batches. The single set of casting machines then forms the molten steel into fixed dimension steel billets. The billets are then moved to an open yard staging area for cooling. The next stage is the rolling process with an average throughput rate of over a hundred tons per hour. Before the rolling process begins, cooled billets will be moved from inventory or from the cooling area into a reheating tube according to the rolling production schedule. The cooled billets are reheated in preparation for processing through the hot rolling train where the billets are gradually forged into a specific diameter size cable wire. As an inventory buffer stage, the reheating process will pace the rolling schedule. As a result, only sufficient volume of billets is actually reheated prior to each rolling cycle. Depending on the specific orders scheduled and sequenced to roll, the rolling train lines may be subject to minor and major changeovers/setups. The rolled steel cable wire is then moved to an open yard where it is staged for final packaging, loading and shipment release.

The two processing stages, steelmaking and (billet) rolling, currently are scheduled separately, and the individual utilization rate of the two stages is considered quite good. The steelmaking stage generally paces the rolling production, as the usual scheduling priority is to provide what is needed for the rolling stage in a timely manner. However, the difficulty often facing schedulers is that the realized (average) throughput rate of the rolling facility is different than that of the steelmaking stage. This often disrupts the rolling schedule due to production batch shortages, or interrupts further upstream the melting schedule due to urgent need of specific billet types for a production batch cycle. Such occurrences are common in practice and create significant challenge. Once such an interruption occurs, the rolling production facility is shutdown completely. This requires the production manager to check for other available billet inventory, immediately schedule and change the roll train setups, and then restart the rolling process. By comparison similar intervention is required at the steelmaking stage. In particular, the steelmaking operator will clean up the current in-process tundish, and insert the desired grade of steel into the production sequence. This leads to costly production order expediting overtime and excess inventory.

3.2 Scheduling Challenges

Given the production environment described above, several challenges are summarized. The production interruptions possibly cause increased idle time before and after a special production batch run. Operationally speaking, both stages incur capacity loss, and result in significantly higher production cost according to our observations and discussions with managers in the company. In the extreme case, the rolling facility has been forced to a complete stoppage until the upstream steelmaking facility finishes the current production of needed product grades. Because the steel manufacturer often operates at or near full production capacity, any interruption likely results in delivery delays of other customer orders as well. And this further jeopardizes plant profitability and efficiency. Therefore, production schedules that coordinate different production stages can more effectively control the production cost on a plant-wise level, so that overall production can be synchronized (despite the product-specific throughput rates) at lowest production cost.

Another challenge facing steel manufacturers is competitiveness in the industry. Customer service expectations, namely responsiveness and satisfaction, are high. Steelmakers must maintain high customer service levels while evaluating the impact of delaying some customers' orders to satisfy others. Controlling the cost of delaying orders (lateness) is as important as reducing production cost. Because the throughput rate is product specific and major setups between different steel grades and production dimensions consume a significant proportion of available production time, the resulting production schedule will influence the order fulfillment rate. In practice, manufacturers are committed to reduce late order delivery to stay competitive. Our model addresses these challenges together and is discussed next.

3.3 Model Specification

The model proposed here is a Mixed Integer Linear Programming (MILP) model. The scheduling unit is a single production batch. Each production batch specifies the diameter sizes and desired grades of the steel. The steel grades are categorized into several product families by the metallurgist. In the steelmaking process, frontline production operators try to aggregate similar grade production batches to smooth the casting process. They try to avoid long lead-times and high changeover costs between different product families. In the rolling process, frontline operators prefer to batch together similar diameters in order to avoid roll train setups and save production time. Each individual batch also has a due time and a penalty cost is incurred if the production schedule violates the batch due time requirements. The MILP model proposed here aims to minimize the costs incurred by the capacity loss during setup and changeovers, and the penalty cost of batch lateness.

The notations used in the model are listed and defined as follows:

Parameters

I_i: Inventory indicator for batch i. 0 for no billets inventory, 1 for sufficient billet inventory to fulfill current batch orders.

N: Total number of production batches.

i, j: Production batch index, $i, j = 1, \ldots, N$. $i, j = 0$ is a dummy batch index, it is explicitly specified in the model when needed.

L_i: Lateness penalty of a unit period of time for batch i.

C_{ij}: Steelmaking changeover time required between batch i and j.

S_{ij}: Rolling changeover time between batch i and j.

T_1: Total available steelmaking time during the scheduling period.

T_2: Total available rolling time during the scheduling period.

D_i: Due time for production batch i.

T_{1i}: Processing time for batch i in steelmaking.

T_{2i}: Processing time for batch i in rolling.

T_{break}: Minimum required break time between steelmaking and rolling for the same batch.

M: Auxiliary parameter.

α: Opportunity cost per unit of time for steelmaking process.

β: Opportunity cost per unit of time for rolling process.

Decision Variables

x_{ij}: Elements in changeover time matrix of steelmaking process.

y_{ij}: Elements in changeover time matrix of rolling process.

t_{1i}: Start time of batch i in steelmaking process.

t_{2i}: Start time of batch i in rolling process.

l_i: Total tardiness of batch i.

k_i: Auxiliary variable of batch i.

The model specification is given in equations (1) - (12). Constraint set (5) is equivalent to constraint sets (5-1) and (5-2). In order to convert the model into a MILP, the two sub-constraints replace equation (5) in the implementation. Decision variable k_i and parameter M are auxiliary for the purpose of model implementation.

The Model (SSM)

$$\min Z = \alpha \sum_i \sum_j C_{ij} x_{ij} + \beta \sum_i \sum_j S_{ij} y_{ij} + \sum_i L_i * l_i \quad (1)$$

$s.t.$

$$(t_{1i} + T_{1i} + T_{break}) * (1 - I_i) \leq t_{2i} \quad \forall i \quad (2)$$

$$t_{1i} + T_{1i} + C_{ij} - (1 - x_{ij}) \cdot T_1 \leq t_{1j} \quad \forall\, i, j;\ i \neq j \tag{3}$$

$$t_{2i} + T_{2i} + S_{ij} - (1 - y_{ij}) \cdot T_2 \leq t_{2j} \quad \forall\, i, j;\ i \neq j \tag{4}$$

$$l_i = \max\{0, t_{2i} + T_{2i} - D_i\} \quad \forall\, i, i \tag{5}$$

$$0 \leq l_i \leq 0 + M \cdot k_i \quad \forall\, i \tag{5-1}$$

$$t_{2i} + T_{2i} - D_i \leq l_i \leq t_{2i} + T_{2i} - D_i + M \cdot (1 - k_i) \quad \forall\, i \tag{5-2}$$

$$\sum_j x_{ij} = 1 - I_i \quad i = 1, \dots, N; j = 0,1, \dots, N \tag{6}$$

$$\sum_i x_{ij} = 1 - I_j \quad j = 1, \dots, N; i = 0,1, \dots, N \tag{7}$$

$$\sum_i y_{ij} = 1 \quad j = 1, \dots, N; i = 0,1, \dots, N \tag{8}$$

$$\sum_j y_{ij} = 1 \quad i = 1, \dots, N; j = 0,1, \dots, N \tag{9}$$

$$0 \leq t_{1i} \leq T_1 - T_{1i}, \quad \forall\, i \tag{10}$$

$$0 \leq t_{2i} \leq T_2 - T_{2i}, \quad \forall\, i \tag{11}$$

$$k_i, x_{ij}, y_{ij}\ are\ binaries \tag{12}$$

The objective function (1) minimizes the total cost of production batch changeovers and batch tardiness penalties. Constraint (2) enforces the rule that any production batch scheduled for rolling must complete steelmaking production first. Otherwise there should be sufficient billet inventory for rolling. In other words, the constraint prevents any schedule conflicts and possible production interruptions. The inventory index, I_i, shows whether the production batch in the cooling yard is ready for rolling. Constraints (3) and (4) restrain the batch sequence and schedule of the two production processes respectively. They require that if batch *j* is scheduled after batch *i*, then the starting time of batch *j* must be later than or equal to the starting time of batch *i* plus the necessary processing and changeover time. Constraint (5) computes the lateness of batch *i*. l_i shows how long production batch *i* has been delayed. Constraints (6) - (9) are production batch sequencing requirements. They ensure that each production batch will be scheduled for production once and only once when needed. If certain batches have billet inventory available in the cooling yard, they don't need to be scheduled for steelmaking production. Index 0 is for the dummy production batch. Constraints (10) - (11) enforce that each production batch must be processed during the current planning period. Constraint (12) shows the types of decision variables.

The structure of the model proposed here shares similarities with the classic vehicle routing problem with time window (VRPTW). VRPTW is known to be NP-hard (Fisher et al. 1997, Tang and Wang 2009), and consequently this scheduling model is NP-hard. This means that finding an optimal solution to this model requires unrealistic time for practical production scheduling. The scheduling problem proposed to solve here is tactical rather than strategic, so finding the solution in a short period of time is crucial for implementation. As a result, heuristic solutions must be proposed. Detailed problem instances are explained in the following subsection.

3.4 Description of Data Used

The operational data used in our scheduling model is described by the characteristics of a typical order batch. This includes product diameter size, steel grade, and batch processing time for both steelmaking and rolling stages. The production batch is the smallest scheduling unit for Model (SSM).

The details of the data item are shown in Table 2. There are twelve elements in each production batch. "ID" is a unique identifier for the production batch. T1 and T2 are both start times for steelmaking and rolling, respectively. S1 and S2 are the batch sequence numbers for steelmaking and rolling process, respectively. "Quant" specifies the product requirements during the horizon. "Grade" is the required quality of the batch. It specifies the chemical composition of the steel, such as concentration of carbon or copper. For example, more than 0.6% carbon is considered high carbon steel, and the grade shows the specific carbon range. Steelmaking process determines the grade of the batch. Impurities like carbon or sulfur are further removed through heating, and other desired compositions like chromium or manganese are added. "Size" specifies the wire diameter for the batch of steel. A typical range is 0.2 to nearly 2.0 inches. "Inv" is binary indicator showing whether or not a particular batch has billet inventory in the cooling yard. Billet inventory is in-process inventory that the steel billets are produced in the steelmaking stage, and have the required grade already. They are waiting to be cooled down and used for rolling production. "Process Time 1" and "Process Time 2" records the total time required to finish the steelmaking and rolling for the batch, respectively. They both depend on the aggregate volume of the batch, the grade and size of the batch, since the throughput rates differ by product.

Table 2: Data Structure

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
ID	T1	T2	S1	S2	Quant	Grade	Size	Due	Inv	Process Time 1	Process Time 2

Two changeover matrices are used when considering candidate schedules. Once a set of production batch instances is generated, a separate look-up routine retrieves the corresponding grades' and sizes' changeover time, then two changeover matrices are constructed for the scheduling model. This "look-up and construct" procedure is necessary because the set of production batch instances usually doesn't include all the available steel products. For the purpose of implementing the scheduling model in a convenient way, only a subset of the changeover matrices is selected. A sample of the changeover matrix for rolling production is shown in Table 3.

Table 3: Changeover Time (minutes) Matrix for Rolling Production

	Size 1	**Size 2**	**Size 3**	**Size 4**	…
Size 1	*	87	86	83	…
Size 2	59	*	86	83	…
Size 3	60	64	*	82	…
Size 4	63	60	64	*	…
…	…	…	…	…	…

The production batch instances are simulated to reflect different characteristics of production quantity, and how urgent the batches are. As shown in the literature (Tang et al. 2011, Tang and Wang 2009, Fisher et al. 2003), the size of the instance set has the most significant impact on the solution time, and even the leading MILP software can only find optimal solutions for fairly small problem instances. In order to explore the solution experience given different number of production batches, we design an experiment and use small problem instances to further study the model.

CHAPTER 4 NUMERICAL EXPERIMENTATION AND SOLUTION

In order to investigate the solution time required to solve the scheduling model proposed in Chapter 3, this Chapter discusses the experimental design and the exploratory results of solving small-scale problems. The experimental factors include, the data instance size, demand pattern (high or low), and due time (tight or slack). Small data instances are solved using CPLEX, and we report results of solving the sample data instances.

4.1 Problem Instance Size

In general, NP-hard problem with a large-scale data instance cannot be solved to optimality within limited time, and the size of the problem generally decides the degree of complexity for this type of model. In our model, we assume N batches in the problem instance, and assume that all batches need to be processed through steelmaking and then rolling. In this case, there are $2(N+1)^2$ decision variables for the sequencing decision. There also are another 2N decision variables needed to assign the start time of each batch, N decision variables for the lateness of each batch, then finally N auxiliary decision variables for model implementation. In other words, there are $2(N+1)^2 + 4N$ decision variables for an instance set with N production batches. The number of constraints is also similarly determined. Tang and Yan (2009) concluded that a batching problem such as this cannot be reliably solved to optimality within 3600 seconds using CPLEX on a capable computer when the planning horizon exceeds 160 days. In their experiment, only five out of twenty random problem instances are solved within 3600 seconds using CPLEX. Overall, fifteen out of twenty-five instances were not solvable within the time limit. Another way to understand the NP-hard nature of the our scheduling model is that it needs to investigate separate schedules for two different stages, and both the sequence of the batches and the exact time of production need to be determined. So it shares similar characteristics like VRPTW as previously noted.

Small size problem instances are generated to better understand the solution time. Six levels of problem size are selected, with the number of batches in each instance range from 5 to 15. The problems generated are comparable in complexity to those studies in the literature. As'ad and Demirli (2011) use 10 data instances to analyze the solution of the rolling mill planning model. Their solution times grew to more than 20 hours for problems containing over 600 decision variables and more than 1200 constraints. The heuristic solution they proposed required more than 16 hours to execute. Based on these observations, we believe the batch instance sizes selected here provide a good basic understanding of the complexity of the scheduling model.

4.2 Demand Pattern

Demand pattern of the problem instance has also been studied in the literature. Tang et al. (2012) generate data instances with demand pattern from a uniform distribution, and the demands for different instances are generated independently. In their study, the product type range is relatively small and concentrated, and each of the products maintains a stable level of demand. The steel plant under their study is among top three strip steel producers in China, and has a high volume of outputs. So they introduce a high average, small variance uniformly distributed demand pattern. In a different paper, Vanhoucke and Debels (2009) used random order quantities between 20 and 40 tons in the experiment to study the machine routing and assignment problem of a steel plant in Belgium.

The steel manufacturer in our study is a small to medium scale steel company, who competes in a niche market with large global steel producers. They offer a wide range of products with grade and diameter size combinations. Small order sizes (say of a few tons) of steel wire are not uncommon for the company. So there is a need to manage these small, highly variable orders in a way that can guarantees high customer service on a cost-competitive basis. So a random or uniformly distributed demand pattern with high demand volume does not describe in our exact case, but it is possible to consider such a scenario. We use a two-level design to reflect the real world demand scenarios. Low volume scenario has an average of ten tons volume for each production batch. And in the high volume scenario, 20% of the production batches have an average volume of 160 tons.

4.3 Order Due Time

Order due time is also a factor that has been included in scheduling models. It is included as another factor as we associate a penalty cost for late order. Order lateness is introduced as a trade-off. In many scenarios, a steel manufacturer faces very strict order due time requirements in production planning. To guarantee better than average customer service level, on-time completion is a priority. In such situations, an overdue order incurs severe lateness penalties. Clearly, production batch scheduling would be tightly constrained as the number of order batches to schedule increases. This is due to the resulting tighter production windows driven by due time of the orders and available production time. There are two levels of order due time in our experiment. One is the slack due time scenario, where all production batches are due at the end of the planning period. The other is the tight due time scenario, where 50% of the production batches are due before the end of the planning period. The due time of these batches are generated with an average due time at half the planning horizon, and standard deviation of one eighth of the planning horizon.

Table 4: Factors and Levels in the Experiment

	Factor		
	Data Instance Size	**Demand Pattern**	**Due Time**
Levels	5, 7, 9, 11, 13, 15	High Low	Tight Slack

The complete design of experiments is summarized in Table 4. In the full factorial experiments, there are 6 levels of instance size, each with 2 levels of demand volume and 2 levels of due time. This provides 6×2×2 = 24 scenarios in total with each scenario replicated 10 times for a total of 240 runs.

4.4 Initial Solution Experience

Model (SSM) is solved by using CPLEX 12.5 on a 2.1 GHz dual core CPU computer with 4GB memory. We constrain the running time to a maximum of 3600 seconds for each small problem instance. In practice, tactical production planning models like Model (SSM) must be solved several times weekly as schedules are revised. In a competitive commercial market, the schedule evaluation time constraint is even more strict and demanding. When changes in order priority and quantity, or unexpected production disruptions occur, a feasible and cost-effective plant schedule must be generated. In this experimental phase, some problem instances of Model (SSM) could not be solved within the time limit.

Table 5: Due Time and Demand Scenario

	Case 1	Case 2	Case 3	Case 4
Demand Pattern	Low	Low	High	High
Due Time	Slack	Tight	Slack	Tight

The CPU utilization and memory utilization are being tracked during each run of data instance. In cases that require relative long solution time, CPU utilization exceeds 90% while memory utilization increases gradually over time. For cases not solvable within 3600 seconds, both the CPU and memory utilization approach 100%.

Table 6: Solution Time (seconds) For Model (SSM)

Size	Case 1	Case 2	Case 3	Case 4	Average
5	0.269	0.085	0.075	0.038	0.117
7	0.135	0.104	0.097	0.141	0.119
9	2.475	1.398	0.336	0.239	1.112
11	15.043*	4.443	5.228	4.37	4.680**
13	8.037	7.454	1.775	2.87	5.034
15	150.199*	-	168.559	1761.048	964.804**

- Not solved within time limit. * Solution with numerical tolerance. ** Case 1 and Case 2 for "Size 15" are excluded.

Table 5 reports the experimental design scenario of due time and demand pattern. All instances are solved to optimality except for Case 1 and Case 2 with fifteen batches. The average solution time in our initial computational study is reported in Table 6. All time values are reported in seconds. Among the 24 instances, Case 2 with instance size 15 is not solvable. Case 1 with instance sizes of 11 and 15 report solution but with numerical tolerance, which is a sign of growing complexity as the instance size grows. The relationship is visualized in Figure 2. It is clear that computing time increases along with problem size, and it shows a non-linear increasing pattern. This is consistent with the analysis of the model structure, which shows the model falls into the strongly NP-hard category.

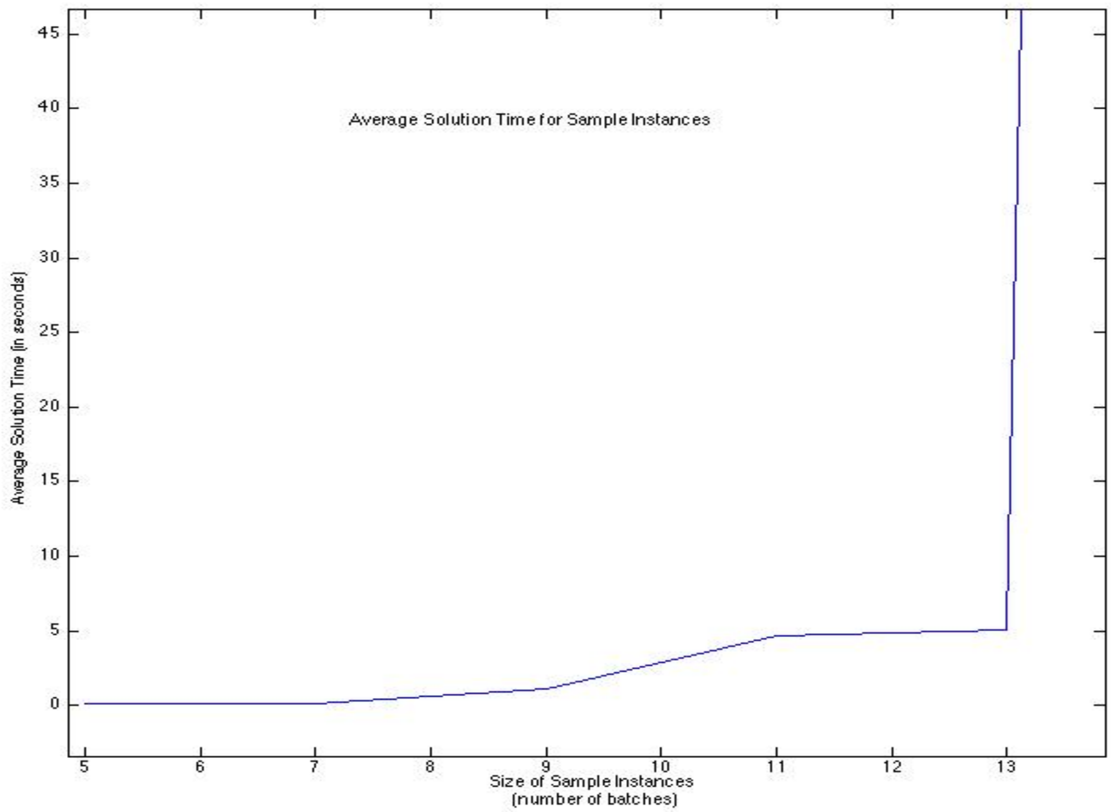

Figure 2: Average Solution Time for Sample Instances

Table 7: Optimal Solution Values of Sample Instances

Size	Case 1	Case 2	Case 3	Case 4
5	269	112	161	75
7	225	176	116	265
9	396	130	180	381
11	226	145	391	151
13	398	191	473	170

Table 7 summarizes the optimal solutions found for the sample instances. Sample instances with fifteen production batches are not included in the table, because two cases are not solved to optimality. The sample instances and their optimal solutions are used in following chapters, where meta-heuristic algorithms and computational study are discussed. In the next chapter, we discuss the meta-heuristic approaches proposed to solve Model (SSM).

CHAPTER 5 HEURISTIC APPROACHES

The scheduling literature shows that for many classes of NP-hard problems, heuristic approaches are usually required to solve practical-sized problems. In this study, we found that leading commercial software packages could not solve Model (SSM) for even fairly small problem instances. This limitation may be due to the multi-stage nature and the complex problem structure consisting of linking constraints. As a result, heuristic approaches are needed to solve large-scale problem instances.

In this chapter, we study a new heuristic approach, called Wind Driven Optimization (WDO). It was recently applied to solve antenna optimization problems in engineering (Bayraktar et al. 2013). To the best of our knowledge, WDO research is scant. As the first study of WDO heuristic applied to production scheduling, or scheduling research in general, we focus on characterizing the WDO mechanism. We discuss the method and adapt it to production scheduling research. In addition, another heuristic method, Particle Swarm Optimization (PSO), is also discussed and compared to WDO.

5.1 WDO

Nature-inspired optimization techniques, such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO), have proved to be effective and efficient in solving production scheduling models in the literature (Table 1). Wind Driven Optimization (WDO) is a new global optimization methodology inspired by how small air parcels navigate over the three-dimensional space. It follows Newton’s second law of motion, unlike other optimization methods mentioned.

"Wind Driven" refers to how the wind blows in the earth's atmosphere in an attempt to balance air pressure differences. Small air parcels travel along the wind from high pressure areas to low pressure areas. The pressure value changes as the air parcel's spatial position changes. There are four forces that influence how air parcels move spatially. The pressure gradient force ($F_{pg} = -\nabla P \delta V$), which is a major cause, relates to pressure gradient, ∇P, and finite mass and volume of the air parcel δV. The pressure gradient force drags air parcels from high pressure areas to low pressure areas. The pressure gradient ∇P is related to RT, which are the universal gas constant and the temperature. A friction force ($F_f = -\rho \alpha v$), which comes from the resistance from surrounding air parcels, applies a counter directional force to change direction. It relates to friction parameter α and the air parcel velocity v. A gravity force ($F_g = \rho \delta V g$) represents the earth's gravitational field, and g is the gravity parameter. It causes air parcels fall towards the earth surface. A Coriolis force ($F_c = -2\Omega \times v$), which contributes to the deflection of the wind (air parcels) from the current path, is mainly caused by the rotation of the earth Ω. v is the air parcel velocity. These four forces together govern the velocity and position of air parcels when they move in the space. Equations of velocity change can be set up according to Newton's Second Law using the four forces. The velocity and position of air parcel change according to the relationships shown in equations (13) and (14).

$$v_{new} = (1-\alpha)v_{cur} - g \cdot x_{cur} + \left(RT \left| \frac{1}{i} - 1 \right| \cdot \left(x_{cur\_opt} - x_{cur} \right) \right) + \frac{cv_{cur}^{other\ dim}}{i} \quad (13)$$

$$x_{new} = x_{cur} + v_{new} \quad (14)$$

v_{new} in equation (13) represents the new velocity of the air parcel. α is the friction parameter contributing to the change of velocity by reducing the current velocity v_{cur}. The gravity force (parameter g) applied to the current position changes the velocity as well. The position difference between current air parcel and the current optimal air parcel is $(x_{cur\_opt} - x_{cur})$. The current optimal position locates at the lowest pressure position, and i is the pressure ranking of the current air parcel within the air parcel population. So RT parameter comes from the pressure gradient force. The last part of equation (13) shows the influence of Coriolis force on the new velocity, and it includes parameter c and a velocity from a different dimension. Equation (14) shows that the new air parcel position x_{new} is the sum of its current position x_{cur} and new velocity v_{new}.

A pressure value is associated with each position of the air parcel, and is used to evaluate whether the position under evaluation is the best (lowest pressure). The updating mechanism defined in (13) and (14) guides air parcels moving from period to period, and each time obtained positions are evaluated by their pressure values to decide where should be the best position. In order to better understand the WDO mechanism, we describe the PSO technique in following section, and highlight the similarities and differences.

5.2 PSO

The second heuristic approach examined in this study is the Particle Swarm Optimization (PSO). Kennedy and Eberhart (1995) developed this population-based optimization method. PSO is inspired by the social system of a flock of flying birds. In the social structure of a flock each bird monitors its social behavior and spatial position in the flock guided by both its own objectives, its current best position, and relative to the best (or lead) position in the flock. The velocity and position of each bird in the swarm is updated iteratively. A fitness value is then associated with the position of each bird, and is used to locate the best position. The velocity and position updating mechanisms are summarized in equation (15) and (16).

$$v_{new} = w \cdot v_{cur} + c_2 rand() \cdot (gbest_{cur} - x_{cur}) + c_1 rand() \cdot (pbest_{cur} - x_{cur}) \qquad (15)$$

$$x_{new} = x_{cur} + v_{new} \qquad (16)$$

Coefficient w represents the inertia parameter and preserves a proportion of the current velocity for the updating new velocity. The best position found in the entire swarm and best position found by each bird both influence the velocity update process. The member birds in the swarm are influenced to fly towards these two social positions. In this context, parameters c_2 and c_1 are associated with the position impact. Random numbers are generated by $rand()$ and included in the updating. This reflects the randomness and relative importance of the global best position versus the individual best position during the velocity update stage for each member bird.

The PSO inertia parameter plays a similar role to the friction parameter used by WDO algorithms. Although the WDO and PSO methods do share conceptual similarities with the usage of populations, velocities, and (relative) positions, there are fundamental differences between the methods. For example, WDO is derived from ideas of air flow movement. PSO is developed from the way a social structure and movement of a system of bird swarms. This difference in origin results in the updating mechanism unique to each method. Parameters such as g, RT, and c in WDO cannot be interpreted in a meaningful way in the PSO, so the PSO parameter values reported in previous research cannot be translated into WDO settings. Therefore, a numerical study of WDO parameters is needed in order to shed new insights and help understand the algorithm's application to other problems.

5.3 Heuristic Algorithms Implementation

WDO and PSO algorithm procedures include population initialization, iteratively evaluating the pressure (WDO) or fitness (PSO), and updating position and velocity. Figure 3 shows a flowchart of the algorithms implementation of both WDO and PSO. The best solution found in this procedure is reported as the solution to the problem. In order to unveil the potential power of either WDO or PSO to solve the scheduling problem at hand, there are two issues need to be addressed. First, interpret the WDO and PSO algorithm in the scheduling context. Second, develop a method so that the "position" and "velocity" in the scheduling context can be updated, and the objective function can be evaluated.

5.3.1 Interpretation of WDO and PSO in the Context of Scheduling

The components of WDO and PSO must be interpreted into the scheduling context in order to apply it to solve Model (SSM). The position in both algorithms is associated with value (pressure value for WDO, fitness value for PSO), which can be used to evaluate the position. This relationship is similar to the model solution and objective function value in our scheduling context. As a result, position in PSO or WDO represents a complete production schedule in Model (SSM), and the pressure value (WDO) or fitness value (PSO) is the objective function value of the model. Other analogy of terminology in our scheduling problem context, WDO, and PSO are summarized in Table 8. This interpretation will help bridge the understanding of PSO or WDO in the context of scheduling problems like Model (SSM).

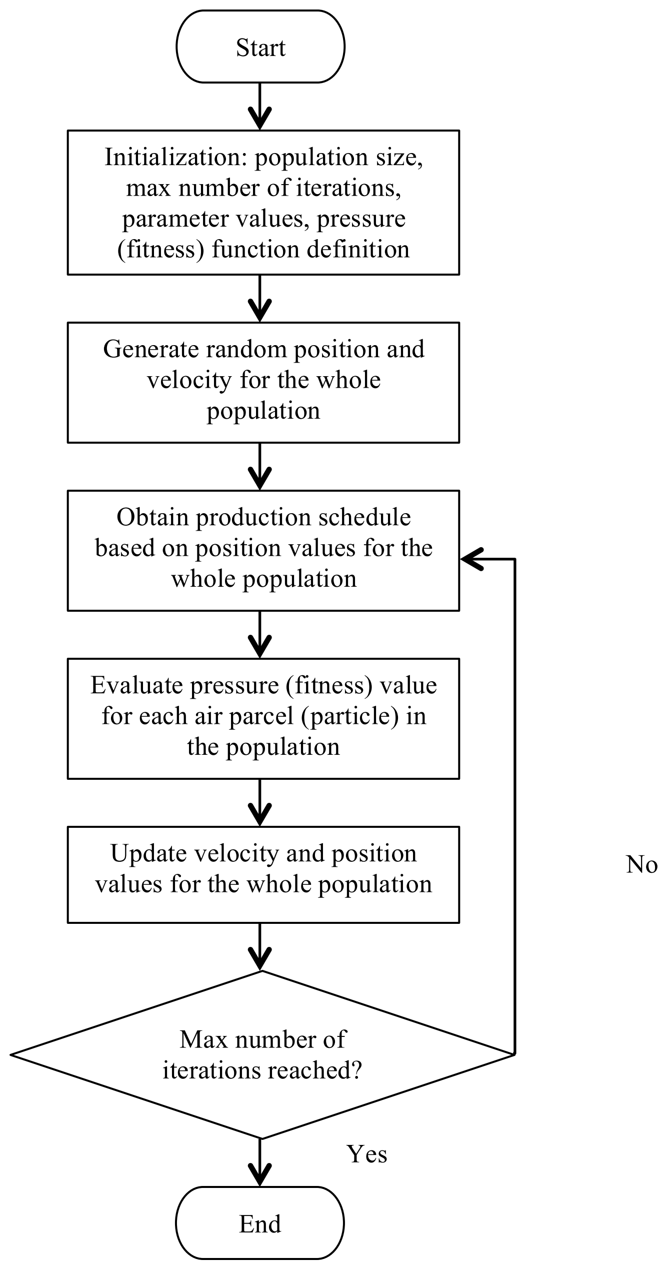

Figure 3: Flowchart Showing Algorithm Implementation

Table 8: Analogy of Scheduling Terminology in the Context of WDO and PSO

	Terminology Description				
Scheduling context	A production schedule	A set of schedules	A production batch	Objective function value	Production schedule change
WDO	Position of an air parcel	Population	One dimension in position	Pressure value	Velocity
PSO	Position of a particle	Swarm	One dimension in position	Fitness value	Velocity

5.3.2 Random Keys Method

The important issue in implementing WDO and PSO in the scheduling context is how to make the computation of schedules possible, and how to update the "position" and "velocity". A study applying PSO to traveling salesman problem (Zhong et al. 2007) proposed a method to compute on traveling sequences directly. Their method uses the travelling sequence as the "position", and defines a set of "plus", "minus", and "multiplication" operators directly on the sequences. While it solved the traveling salesman problem quite effectively, a pitfall of computing on sequences directly is that it is additional computational overhead required to manipulate discrete-valued sequences. By comparison, PSO and WDO use real numbers. Thus, the computational burden will only be magnified when the problem structure is more complicated as is the case for Model (SSM). In order to overcome this problem, it is necessary to define the computation of "positions" and "velocities" on continuous values, rather than on discrete-valued sequences. The real number is used as a unique surrogate of the actual position in the sequence, and the sequence is recovered through a mapping scheme from the set of real numbers. Based on this concept, "position" and "velocity" are both defined as sets of real numbers. The PSO and WDO updating mechanism are therefore more efficient.

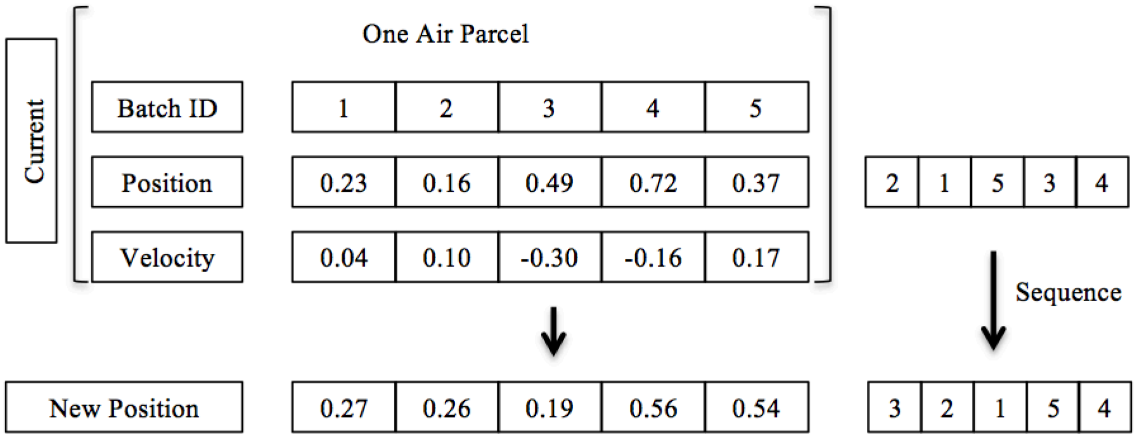

Figure 4: Random Key Method Example with One Air Parcel

This mapping method is quite similar to the random keys method introduced by Bean (1994). Bean presented a general genetic algorithm using "random keys" method to address a variety of sequencing and optimization problems. Random keys are continuous values randomly generated for each discrete element in the problem solution. Each continuous value is called a "key", which represents its original discrete element. In the PSO and WDO context, each random key represents one element in the position or velocity, and is updated in each iteration of the algorithm. The actual solution (sequence) to the problem is obtained by sorting random keys from smallest to largest.

Figure 4 illustrates how random key method works with an example of one air parcel (or particle). The air parcel has five batch IDs in it, and has a pair of position and velocity. Each batch ID has a value in position and in velocity respectively. The current air parcel position and velocity values are shown in the same column of the batch ID they are corresponding to. The current production sequence can be recovered from the position values by ranking them from the smallest to the largest. It shows in Figure 4 that the current sequence is "2 1 5 3 4". New position values are obtained by adding current position and velocity values according to the algorithm. In Figure 4, add the position and velocity values for each batch, and then new position values of the air parcel can be obtained. The corresponding new sequence is “3 2 1 5 4” after ranking the position values in the same manner. New velocity values can be obtained by following the updating mechanism of the algorithm. New production sequence is generated after updating the position values for each production batch in the air parcel.

In our implementation, position and velocity are represented as real number values, and randomly generated initially. Production sequences are converted from the ranking position values, from smallest to largest. Based on the random keys method, the formal definitions of particles in PSO and air parcels in WDO are presented below.

Definition 1: A particle or an air parcel is a set of random key values in which each value corresponds to a production batch. Each production batch to be scheduled in Model (SSM) has two key values, one is used for scheduling in the steelmaking stage, and the other is used for the rolling stage.

For example, assume there are three production batches, namely P1, P2, P3, and one particle or air parcel has its random values as [0.1 0.2 0.3, 0.2 0.1 0.3]. We develop an ordered data structure for each production stage. So After ordering the key values from smallest to largest for each stage, the combined multi-stage production sequence can be recovered as [P1 P2 P3, P2 P1 P3].

Definition 2: The fitness value of a particle (for the PSO), or pressure value of an air parcel (for the WDO) refers to the objective function value of the corresponding production sequence.

The updating mechanism of WDO and PSO can be applied to the air parcel or particle structure defined for the Model (SSM), and using the real number key value for a batch, the two heuristic algorithms can be effectively implemented as solution procedures to the scheduling problem. The updating mechanism is applied to the entire population in each iteration until the maximum number of iterations is reached. The algorithm pseudo codes for of PSO and WDO are given in Figure 5 and Figure 6, respectively.

```
%% Algorithm starts
% Initialization
Set population size, number of iterations, parameters;

% Assign random position and velocity
pop(0) = particle_gen(n);  % generate n random particles (position)
vel(0) = initial_vel(n);  % generate initial velocity for each particle

% Position evaluation, and update position, velocity
while (i <= maximum iteration not reached)
    lateness = late(pop(i));  % compute the lateness;

    % Evaluate the value of current population
    value(i) = eval_pres(pop(i),lateness);

    for (each particle j in the swarm)
        pbest(i)(j) = min(value(i)(j),pbest(i-1)(j));
        gbest(i)(j) = min(value(i)(j),gbest(i-1)(j));

        % Update velocity based on PSO updating equation
        vel(i+1) = update_vel(pop(i),vel(i),pbest(i),gbest(i));

        % Update position based on PSO updating equation
        pop(i+1) = update_pop(pop(i), vel(i+1));
    end  % end of for loop

end  % end of while iteration
Report the current best particle;
%% Algorithm ends
```

Figure 5: PSO Heuristic Pseudo-code

```
%% Algorithm starts
% Initialization
Set population size, number of iterations, parameters;

% Assign random position and velocity
pop(0) = particle_gen(n);  % generate n random particles (position)
vel(0) = initial_vel(n);  % generate initial velocity for each particle

% Pressure evaluation, and update position, velocity
while (i <= maximum iteration not reached)
    lateness = late(pop(i));  % compute the lateness;

    % Evaluate the pressure of current population
    pressure(i) = eval_pres(pop(i),lateness);
    rank(i) = ranking(pressure(i));  % ranking based on pressure

    % Record the current best particle
    best_particle(i) = min(rank(i),best_particle(i-1));

    % Update velocity based on WDO updating equation
    vel(i+1) = update_vel(pop(i),vel(i),rank(i),best_particle(i));

% Update position based on WDO updating equation
    pop(i+1) = update_pop(pop(i), vel(i+1));

end  % end of while iteration
Report the current best particle;
%% Algorithm ends
```

Figure 6: WDO Heuristic Pseudo-code

To the best of our knowledge, there is no other study applying WDO to the scheduling literature. As the first study using WDO and PSO in the SSM scheduling literature, our goal is to evaluate the relative computational performance of PSO and WDO when applied to this scheduling problem. The next chapter details the computational study of both WDO and PSO.

CHAPTER 6 COMPUTATIONAL STUDY

The literature shows that PSO methods can solve both production scheduling models and classic traveling salesman models quite effectively. However, no studies that examine the efficacy of WDO applied to the SSM or PSM scheduling have appeared yet. As a result, we first conduct an exploratory numerical study of WDO parameter settings. This is needed in order to tune WDO-specific parameter settings for solving large-scale instances of SSM. The chapter consists of three sections. The first section details our experience with the parameter values for both WDO and PSO using small-scale problems. The next section reports results of the computational analysis carried out. Three different methods are deployed to analyze WDO and PSO performance, and to identify the preferred parameter setting for each algorithm. Finally, the last section reports the relative performance comparison of WDO and PSO when solving large-scale data instances of Model (SSM).

6.1 Numerical Parameter Study

Bayraktar et al. (2013) apply WDO to solve three optimization problems relating to antenna design. They perform a numerical parameter study of WDO using four different geometrical functions as benchmark functions. The optimal solution and value is known for each function, and parameters are selected based on the best average solutions found. In their study, large friction parameter value in the range of (0.8, 0.9) is recommended. A middle range gravity parameter of (0.6, 0.7) is suggested even though smaller values also provide good performance in solving the three different antenna design problems. Suggested ranges for parameters “RT” (from Pressure Gradient Forces) and “c” (from

Coriolis Forces) fall into (1.0, 2.0) and (0.05, 3.6) respectively. The production scheduling problem (Model SSM) investigated in our study has totally different structure, so another approach is developed to examine the WDO parameters.

Table 9: WDO Numerical Parameter Study

Data Instance Size	5, 7, 9, 11, 13
Case	Case 1, Case 2, Case 3, Case 4
α	0.1, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9
g	0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9
RT	0.5, 1, 1.5, 2, 2.5, 3, 3.5
c	0.1, 0.3, 0.5, 0.7, 0.9, 1.0, 2.0, 3.0

This numerical parameter study examines a wide range of parameter settings. Similar number of parameter levels is chosen for each parameter so that the four parameters are equally emphasized in the study. Because there is no previous study suggesting which parameter is more important than others. Table 9 specifies the parameter setting values considered. The data instances solved in the previous chapter are used for performance comparison here. The size of data instance varies from 5 to 13 production batches. Solutions obtained for the 15-batch size problems were not tractable so they were excluded from consideration in the parameter study.

To compare the performance of WDO with PSO, we used the parameter settings listed in Table 10. The inertia parameter w seems to play a key role in PSO algorithm performance while c_1 and c_2 (with settings in the range 1.5 or 2) do not have much impact. We set c_1 at 1.5 and details the impact of w and c_2. WDO and PSO techniques were used in all problem instances.

Table 10: PSO Numerical Parameter Study

Data Instance Size	5, 7, 9, 11, 13
Case	Case 1, Case 2, Case 3, Case 4
w	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
c_1, c_2	1.5, 2

Both PSO and WDO algorithms are implemented in Matlab©, and executed in the same computing environment as in section 4.4. For performance reasons, the population size is ten times the number of batches in a given data instance, and the maximum number of iterations allowed is 1000. Each data instance is solved 10 times with the best solution found in each repetition recorded. Tables 11 and 12 show a sample of the parameter study results of WDO and PSO. The columns include number of batches, case id, and repetition run of the problem instance, parameter value, and best solution found for the particular run. The solution value varies noticeably with different parameter value settings. For example, the optimal solution of one problem instance is 225, and the solution value found in a particular run could be between 225 and 275 given different combinations of parameter setting. This solution value variation exhibited in the results may imply that parameter settings have significant impact on solution values obtained from the heuristics. The following section investigates such influences.

Table 11: A Sample of WDO Numerical Parameter Study Result

Batch	Case	Repetition	α	g	RT	c	Best Value
5	1	1	0.1	0.1	2	0.7	269
7	1	4	0.6	0.7	2.5	0.5	225
9	2	1	0.8	0.1	3	2	130
11	3	2	0.1	0.7	3	0.5	392
13	2	7	0.1	0.7	1.5	2	199

Table 12: A Sample of PSO Numerical Parameter Study Result

Batch	Case	Repetition	w	c_1	c_2	Best Value
5	1	1	0.7	1.5	1.5	269
7	2	6	0.6	1.5	1.5	176
9	3	7	0.4	1.5	1.5	180
11	4	10	0.2	1.5	2	115
13	1	8	0.1	1.5	2	427

6.2 Analysis of Result

In order to select the proper parameter values leading to good algorithm performance, a three-step parameter selection process is used. This includes ANOVA analysis, top performing parameter setting, and response surface observation. Each of the three processes is detailed in the following sub-sections.

6.2.1 ANOVA Analysis

ANOVA is a useful tool for investigating the underlying structure of factors associated with conducting experiments. The method explains, with statistical reliability, the overall effect of factors' levels on the variables under consideration. We perform ANOVA analysis of WDO and PSO solution quality performance on the parameter levels introduced earlier. In this study, we evaluate solution gap and solution time as influenced by these parameter settings.

The solution gap is computed for each run of WDO and PSO in the parameter study. It is defined as the percentage of difference between the obtained solution value and the optimal value (equation 17). Solution quality performance is defined as 1 – Solution Gap (equation 18). Larger solution quality performance value (close to or equal to 1) is considered as better heuristic performance.

$$Solution\ Gap = \frac{(obtained\ value - optimal\ value)}{optimal\ value} \times 100\% \qquad (17)$$

$$Solution\ Quality\ Performance = 1 - Solution\ Gap \qquad (18)$$

Solution quality performance could be negative if the (best) obtained solution value is too large (more than 2 times optimal value), since Model (SSM) is a minimization problem. But in our study, we don't observe such phenomenon. All solution quality performance fell between 0 and 1.

Table 13: WDO ANOVA Analysis on Solution Quality Performance

	Df	SumSq	MeanSq	F	P-value	Pr(>F)
Factor: α	6	95	15.762	581.6	<2e-16	***
Residuals	548793	14872	0.027			
Factor: g	6	92	15.326	565.5	<2e-16	***
Residuals	548793	14875	0.027			
Factor: RT	6	116	19.366	715.7	<2e-16	***
Residuals	548793	14850	0.027			
Factor: c	7	45	6.486	238.5	<2e-16	***
Residuals	548792	14921	0.027			

Significant codes: '***' 0.001, '**' 0.01, '*' 0.05, '.' 0.1

Table 13 reports the ANOVA main effect results for WDO solution quality performance while using the four parameters as factors. All four WDO parameters are found to have a significant influence on algorithm's quality performance for Model SSM. It suggests that the value settings on α, g, RT, and c significantly influence the quality performance of WDO algorithm. To further explore these differences, a Tukey multiple comparison test on all four parameters was performed. The results are summarized in Tables 14 - 17. The first column tells which two parameter levels are compared. For example in Table 14, "Level 6 vs. Level 7" in the first row indicates that "average solution quality performance difference between parameter α setting at level 6 and level 7". Level 7 in α setting is 0.9, and Level 6 corresponds to 0.8. The "Difference" column tells the average solution performance difference. "Lower Bound" and "Upper Bound" present the lower and upper bound of the difference. The last column shows the p-value at 95% confidence level. Small p-value (>0.05) indicates significant difference between the average solution performances under comparison. Statistically it means solution performance under one parameter level setting is better than the other if the difference is positive. In Table 14 – 17, better-performed parameter levels are shown in the bottom of the table.

Table 14: Tukey Multiple Comparisons of Solution Quality Performance (WDO, α)

*Parameter	Difference	Lower Bound	Upper Bound	P-Value
Level 6 vs. Level 7	0.0010	-0.0014	0.0035	0.8732
Level 4 vs. Level 7	0.0019	-0.0006	0.0043	0.2603
Level 3 vs. Level 7	0.0063	0.0039	0.0088	0.0000
Level 2 vs. Level 7	0.0161	0.0136	0.0185	0.0000
Level 1 vs. Level 7	0.0385	0.0360	0.0409	0.0000
Level 4 vs. Level 6	0.0008	-0.0016	0.0033	0.9510
Level 3 vs. Level 6	0.0053	0.0028	0.0077	0.0000
Level 2 vs. Level 6	0.0151	0.0126	0.0175	0.0000
Level 1 vs. Level 6	0.0374	0.0350	0.0399	0.0000
Level 7 vs. Level 5	0.0002	-0.0023	0.0026	1.0000
Level 6 vs. Level 5	0.0012	-0.0012	0.0037	0.7652
Level 4 vs. Level 5	0.0021	-0.0004	0.0045	0.1665
Level 3 vs. Level 5	0.0065	0.0041	0.0090	0.0000
Level 2 vs. Level 5	0.0163	0.0138	0.0187	0.0000
Level 1 vs. Level 5	0.0387	0.0362	0.0411	0.0000
Level 3 vs. Level 4	0.0044	0.0020	0.0069	0.0000
Level 2 vs. Level 4	0.0142	0.0118	0.0167	0.0000
Level 1 vs. Level 4	0.0366	0.0342	0.0391	0.0000
Level 2 vs. Level 3	0.0098	0.0073	0.0122	0.0000
Level 1 vs. Level 3	0.0322	0.0297	0.0346	0.0000
Level 1 vs. Level 2	0.0224	0.0199	0.0248	0.0000

*α: Level 1 - 0.1, Level 2 - 0.3, Level 3 - 0.5, Level 4 - 0.6, Level 5 - 0.7, Level 6 - 0.8, Level 7 - 0.9

WDO study in antenna optimization problems suggests using large values of α (0.8 to 0.9). However, in our study, the result shows that small α values seem to generate better solutions to Model (SSM). This finding could be explained by the fundamentally different problem structure in our SSM scheduling study. The Tukey multiple comparison results also show that the mechanisms of WDO and PSO are in fact quite different, because all parameters in WDO have a significant impact on solution performance.

Table 15: Tukey Multiple Comparisons of Solution Quality Performance (WDO, g)

*Parameter	Difference	Lower Bound	Upper Bound	P-Value
Level 2 vs. Level 1	0.0041	0.0016	0.0065	0.0000
Level 3 vs. Level 1	0.0134	0.0110	0.0159	0.0000
Level 4 vs. Level 1	0.0207	0.0182	0.0231	0.0000
Level 5 vs. Level 1	0.0290	0.0266	0.0315	0.0000
Level 6 vs. Level 1	0.0334	0.0309	0.0358	0.0000
Level 7 vs. Level 1	0.0348	0.0323	0.0372	0.0000
Level 3 vs. Level 2	0.0093	0.0069	0.0118	0.0000
Level 4 vs. Level 2	0.0166	0.0142	0.0191	0.0000
Level 5 vs. Level 2	0.0249	0.0225	0.0274	0.0000
Level 6 vs. Level 2	0.0293	0.0268	0.0318	0.0000
Level 7 vs. Level 2	0.0307	0.0282	0.0331	0.0000
Level 4 vs. Level 3	0.0073	0.0048	0.0097	0.0000
Level 5 vs. Level 3	0.0156	0.0132	0.0181	0.0000
Level 6 vs. Level 3	0.0200	0.0175	0.0224	0.0000
Level 7 vs. Level 3	0.0214	0.0189	0.0238	0.0000
Level 5 vs. Level 4	0.0083	0.0059	0.0108	0.0000
Level 6 vs. Level 4	0.0127	0.0102	0.0151	0.0000
Level 7 vs. Level 4	0.0141	0.0116	0.0165	0.0000
Level 6 vs. Level 5	0.0044	0.0019	0.0068	0.0000
Level 7 vs. Level 5	0.0057	0.0033	0.0082	0.0000
Level 7 vs. Level 6	0.0014	-0.0011	0.0038	0.6413

*g: Level 1 – 0.05, Level 2 – 0.1, Level 3 – 0.2, Level 4 – 0.3, Level 5 – 0.5, Level 6 – 0.7, Level 7 – 0.9

Table 16: Tukey Multiple Comparisons of Solution Quality Performance (WDO, RT)

*Parameter	Difference	Lower Bound	Upper Bound	P-Value
Level 2 vs. Level 1	0.0029	0.0004	0.0053	0.0102
Level 3 vs. Level 1	0.0084	0.0060	0.0109	0.0000
Level 4 vs. Level 1	0.0198	0.0173	0.0222	0.0000
Level 5 vs. Level 1	0.0311	0.0286	0.0335	0.0000
Level 7 vs. Level 1	0.0363	0.0338	0.0387	0.0000
Level 6 vs. Level 1	0.0365	0.0340	0.0389	0.0000
Level 3 vs. Level 2	0.0056	0.0031	0.0080	0.0000
Level 4 vs. Level 2	0.0169	0.0145	0.0194	0.0000

Level 5 vs. Level 2	0.0282	0.0257	0.0306	0.0000
Level 7 vs. Level 2	0.0334	0.0310	0.0359	0.0000
Level 6 vs. Level 2	0.0336	0.0311	0.0360	0.0000
Level 4 vs. Level 3	0.0113	0.0089	0.0138	0.0000
Level 5 vs. Level 3	0.0226	0.0202	0.0251	0.0000
Level 7 vs. Level 3	0.0278	0.0254	0.0303	0.0000
Level 6 vs. Level 3	0.0280	0.0256	0.0305	0.0000
Level 5 vs. Level 4	0.0113	0.0088	0.0137	0.0000
Level 7 vs. Level 4	0.0165	0.0141	0.0190	0.0000
Level 6 vs. Level 4	0.0167	0.0142	0.0191	0.0000
Level 7 vs. Level 5	0.0052	0.0028	0.0077	0.0000
Level 6 vs. Level 5	0.0054	0.0029	0.0078	0.0000
Level 6 vs. Level 7	0.0002	-0.0023	0.0026	1.0000

*RT: Level 1 – 0.5, Level 2 – 1.0, Level 3 – 1.5, Level 4 – 2.0, Level 5 – 2.5, Level 6 – 3.0, Level 7 – 3.5

Table 17: Tukey Multiple Comparisons of Solution Quality Performance (WDO, c)

*Parameter	Difference	Lower Bound	Upper Bound	P-Value
Level 3 vs. Level 1	0.0009	-0.0018	0.0036	0.9693
Level 4 vs. Level 1	0.0018	-0.0009	0.0045	0.4473
Level 5 vs. Level 1	0.0036	0.0009	0.0063	0.0015
Level 6 vs. Level 1	0.0045	0.0018	0.0072	0.0000
Level 7 vs. Level 1	0.0171	0.0144	0.0198	0.0000
Level 8 vs. Level 1	0.0262	0.0235	0.0289	0.0000
Level 1 vs. Level 2	0.0008	-0.0019	0.0035	0.9838
Level 3 vs. Level 2	0.0017	-0.0010	0.0044	0.5103
Level 4 vs. Level 2	0.0026	0.0000	0.0053	0.0587
Level 5 vs. Level 2	0.0044	0.0017	0.0071	0.0000
Level 6 vs. Level 2	0.0053	0.0026	0.0080	0.0000
Level 7 vs. Level 2	0.0179	0.0152	0.0206	0.0000
Level 8 vs. Level 2	0.0270	0.0243	0.0297	0.0000
Level 4 vs. Level 3	0.0009	-0.0018	0.0036	0.9721
Level 5 vs. Level 3	0.0027	0.0000	0.0054	0.0572
Level 6 vs. Level 3	0.0036	0.0009	0.0063	0.0017
Level 7 vs. Level 3	0.0162	0.0135	0.0189	0.0000
Level 8 vs. Level 3	0.0253	0.0226	0.0280	0.0000
Level 5 vs. Level 4	0.0018	-0.0009	0.0045	0.5038
Level 6 vs. Level 4	0.0026	-0.0001	0.0053	0.0591
Level 7 vs. Level 4	0.0153	0.0126	0.0180	0.0000
Level 8 vs. Level 4	0.0244	0.0217	0.0271	0.0000
Level 6 vs. Level 5	0.0009	-0.0018	0.0036	0.9738
Level 7 vs. Level 5	0.0135	0.0108	0.0162	0.0000
Level 8 vs. Level 5	0.0226	0.0199	0.0253	0.0000
Level 7 vs. Level 6	0.0126	0.0099	0.0153	0.0000
Level 8 vs. Level 6	0.0217	0.0190	0.0244	0.0000
Level 8 vs. Level 7	0.0091	0.0064	0.0118	0.0000

*c: Level 1 - 0.1, Level 2 - 0.3, Level 3 - 0.5, Level 4 - 0.7, Level 5 - 0.9, Level 6 - 1, Level 7 - 2, Level 8 - 3

The analysis of parameter g also shows quite different result from prior studies. In contrast to the previous study, the results show that large rather than small values of g are suggested for solving the problem we proposed. The analysis also shows that higher levels of parameter RT and c are preferred. In the updating mechanism, RT contributes to influencing the air parcel to move towards the incumbent best position, and c helps introduce randomness to the current air parcel as it introduces position information from other nearby air parcels in the population.

Table 18: PSO ANOVA Analysis on Solution Quality Performance

	Df	SumSq	MeanSq	F	P-value	Pr(>F)
Factor: *w*	2	0.2300	0.1150	9.96	<.0001	***
Factor: c_2	1	0.0148	0.0148	1.28	0.2573	
Interaction	2	0.0008	0.0004	0.04	0.9649	

Significant codes: '***' 0.001, '**' 0.01, '*' 0.05, '.' 0.1

Results of the PSO ANOVA are summarized in Table 18. The nine levels of w are grouped into 3 categories, namely small (0.1, 0.2, 0.3), medium (0.4, 0.5, 0.6), and large (0.7, 0.8, 0.9). The ANOVA doesn't include c_1 because all factor ANOVA result shows that c_1 is not significant. In Table 18, it shows that w has significant influence on solution quality performance, and that parameter c_2 has no noticeable influence on PSO performance. The Tukey multiple comparison test (Table 19) also suggests PSO achieves significantly better performance on larger values of w. Higher w values mean that new velocity preserves more of current velocity into it. PSO can obtain best solution value faster with high value w for the same problem instance. This result is consistent with the finding in previous studies that w is a key factor in PSO algorithm and usually has significant impact on the performance.

Table 19: Tukey Multiple Comparisons of Solution Quality Performance (PSO, *w*)

Parameter	Difference	Lower Bound	Upper Bound	Pr(>F)
Large - Medium	0.0125	0.0022	0.0227	***
Large - Small	0.0193	0.0090	0.0296	***
Medium - Small	0.0069	-0.0034	0.0171	

These analyses help us to characterize preferred parameter values that, in turn, provide a foundation of further investigation of heuristic performance (time or quality). The results for parameters α and g highlight the importance of the parameter study when applying WDO to different classes of problems as in our study. To fine tune the proper parameter values for our proposed Model (SSM), the top-performing parameter values are examined next.

6.2.2 Top Performing Parameter Settings

We conducted analyses of a grand total of twenty problem instances in our experimental design. The top forty quality performance values from the 2744 parameter combinations for solving each case are selected for analysis so that we could identify the overall trends or patterns leading to top solution performance quality. In order to help differentiate the top performing parameter values from others, cases with batch number smaller than 9 are excluded, because most runs in such cases found optimal solutions. For each parameter in WDO and PSO, frequency counts of parameter values are recorded. There are a total of 480 counts for each parameter. This provides an alternative perspective to studying the parameter settings.

Figures 7-10 illustrate the historical plots of the levels of four parameters among the top performing WDO runs. It shows that friction parameter α has the highest frequency count at value 0.1, and gravity parameter $g = 0.9$ appears most frequently. The results are consistent with the founding in ANOVA analysis, and provide more evidence for selecting proper parameter values. It also can be observed that RT and c prefers medium to large values. Historical plots of the four parameters on each problem instance size show similar patterns as Figures 7-10. Based on the results of both ANOVA analysis and historical plot of top performing parameters, we select $\alpha = 0.1$ and $g = 0.9$ to further explore the combined results of all four parameters in the next sub-section.

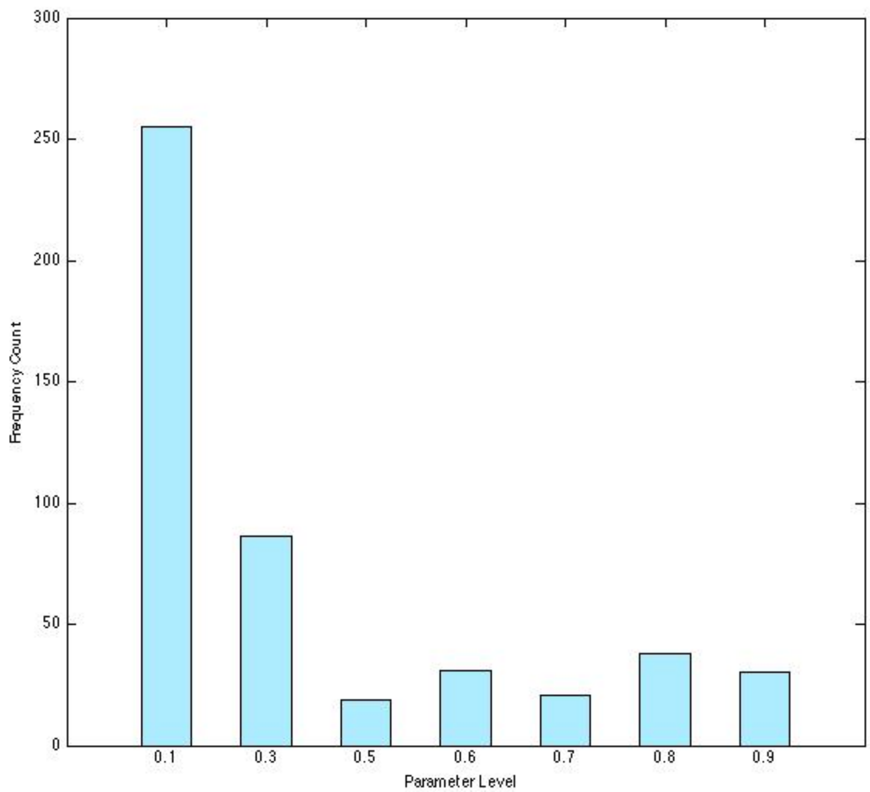

Figure 7: (WDO) Frequency Plot of α values

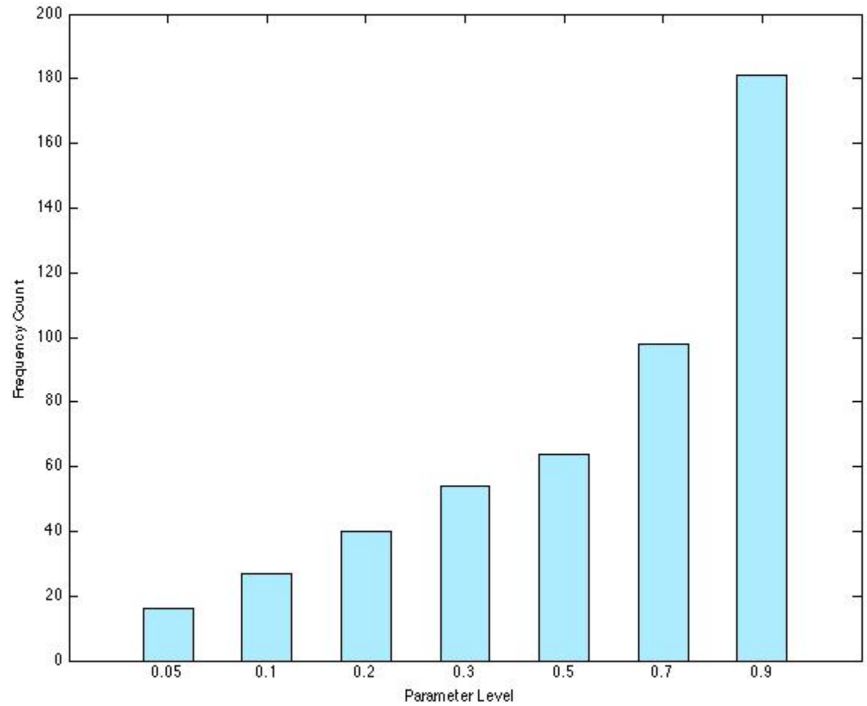

Figure 8: (WDO) Frequency Plot of g values

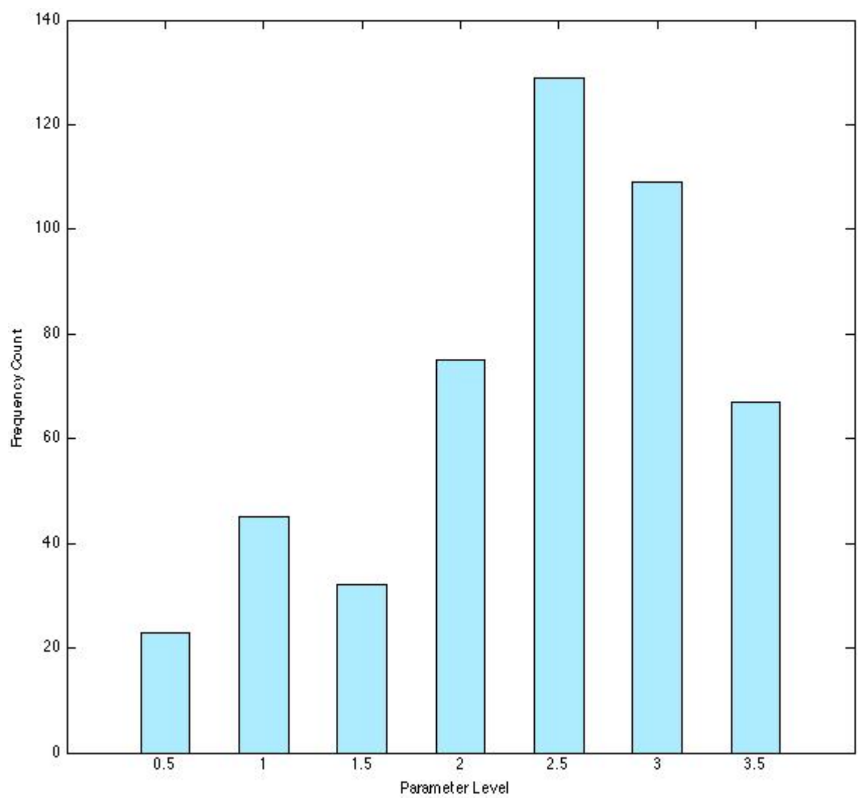

Figure 9: (WDO) Frequency Plot of RT values

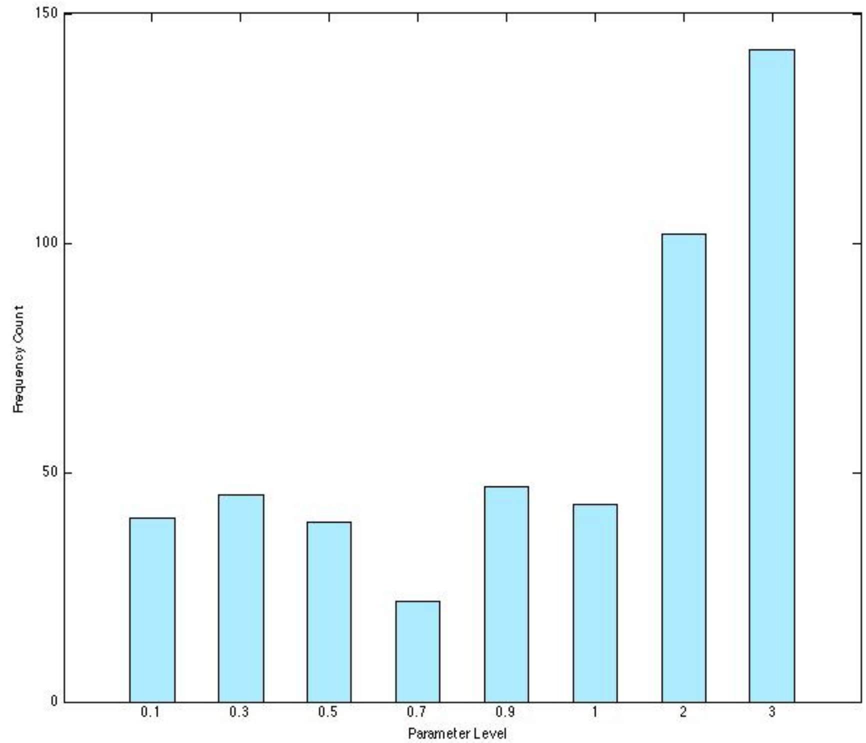

Figure 10: (WDO) Frequency Plot of c values

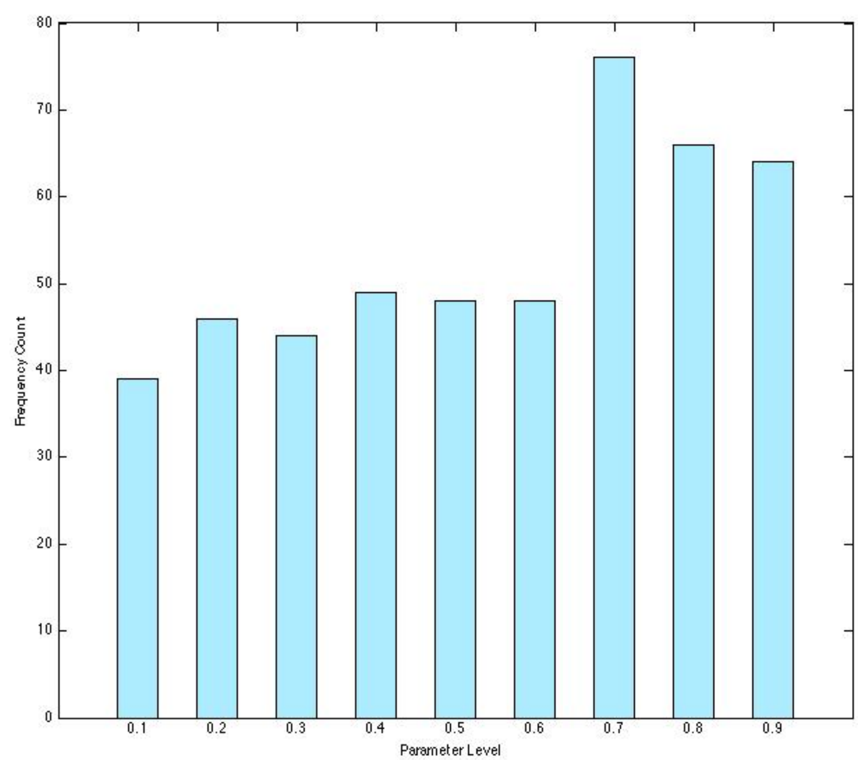

Figure 11: (PSO) Frequency Plot of *w* values

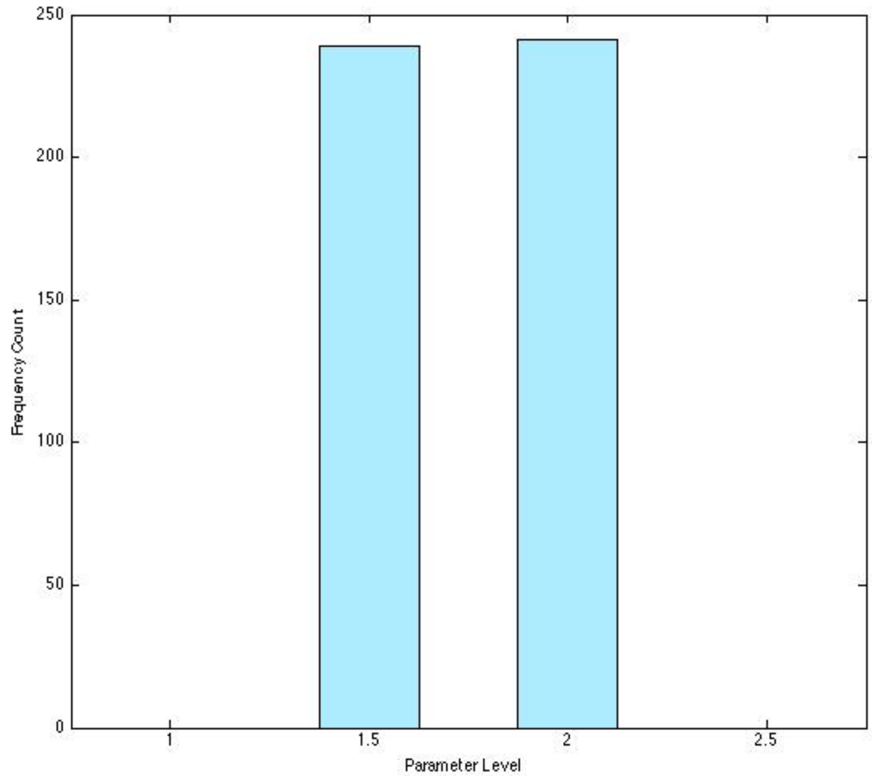

Figure 12: (PSO) Frequency Plot of c_2 values

Figures 11 and 12 report the frequency plots of parameters w and c_2 for the PSO heuristic. They are both consistent with the ANOVA analysis results suggesting that large values of w are important, while levels of c_2 (Figure 12) seem to have no noticeable influence. As a result, we chose settings of $w = 0.7$, $c_2 = 2$ for use in solving the large-scale data instances.

6.2.3 Response Surface

In order to interpret the combined results of all parameter settings in WDO, we set $\alpha = 0.1, g = 0.9$, and plot a three-dimensional the response surface of average "Solution Gap" defined in Equation (17). And all levels of RT and c are included. In Figure 13, the lowest point indicates the preferred parameter settings as small solution gap means better heuristic performance. It can be observed that lowest solution gap in the figure appears at $RT = 2.5, c = 2$. The combined results (of the ANOVA analysis, the top-forty analysis, and the response surface) provided considerable confidence in selecting the WDO parameter values for $(\alpha, g, RT, c) = (0.1, 0.9, 2.5, 2)$ for solving large-scale problem instances.

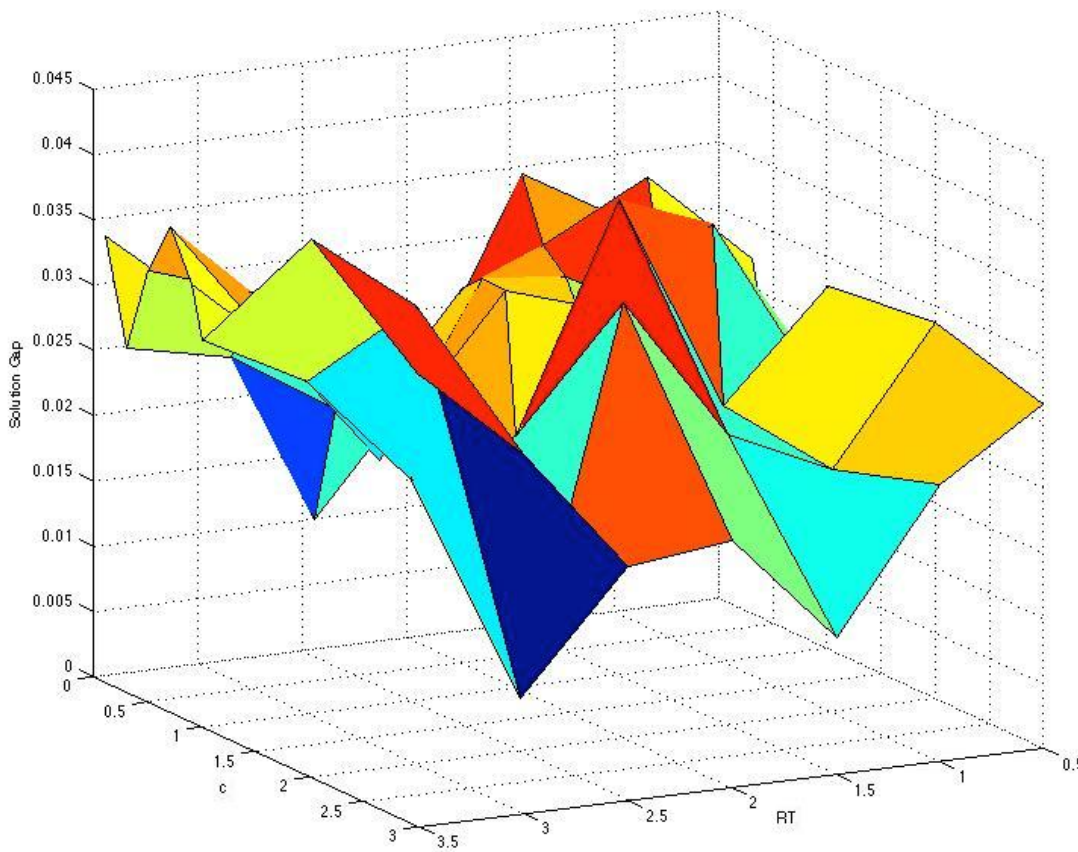

Figure 13: WDO Response Surface of Solution Gap (RT and c)

The corresponding response surface of solution time is presented in Figure 14. It can be discovered that the average solution time for each run is around one second. The best solution time observed is at parameter levels $(\alpha, g, RT, c) = (0.1, 0.9, 1, 3)$. Parameter levels that have lowest solution gap in Figure 13 also show short solution time, which is less than one second. Given the fact that the solution quality is relatively more important, and the solution time difference is fairly small and acceptable for usual computing environment, we choose the parameter levels that minimize solution gap in solving large-scale problem instances.

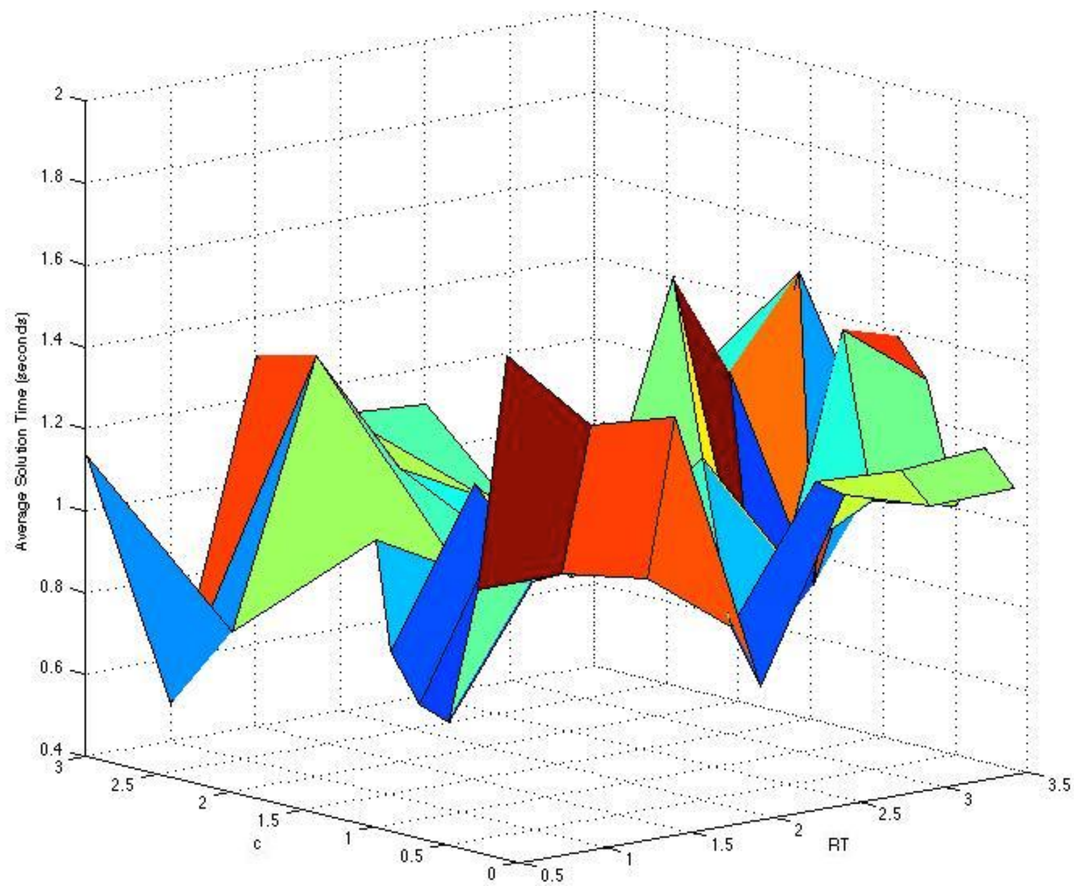

Figure 14: WDO Response Surface of Solution Time in Seconds (RT and c)

6.3 Performance Comparison on Large Data Instances

Large data instances are generated based on the 4 cases shown in Table 6. The number of batches in the instances is set to 150, 160, 170, 180, and 190. Problems as this scale, even using the leading commercial packages such as CPLEX, cannot be solved in the usual computing environment practically. Two distinct problem instances are generated at each data instance size, each case. There are forty large problem instances in total. In terms of production planning horizon, the total production time available (production capacity) reflects a seven-day horizon, and assumes 24-hour continuous production. We discuss experience with problem instances and the population structure used in WDO and PSO in this section.

Because we randomly generated our problem instances, infeasible solutions can result when there is insufficient production capacity available. In such cases, the problem instance cannot be used to evaluate the performance of the heuristics for our purposes here. To mitigate this problem however, we pre-process (pre-asses and adjust) problem instances in order to ensure planning horizon feasibility. We performed a two-step pre-assessment, and adjustment to the production capacity as needed. In the first step, the total aggregate capacity requirements check is performed for both steelmaking and rolling. In the second step, the ten replications of each problem instance are solved using PSO. For these larger problems we now use a 14-day planning horizon rather than a seven-day horizon as before. The obtained changeover times by PSO for both production stages are used. In either stage, if the batch processing time plus the changeover time is longer than the planned production horizon, then the problem instance is treated as infeasible. We found that problem instances with high demand pattern required more processing time. As a result, we augmented the planning horizon up to the minimum requirement during the capacity adjustment phase discussed earlier. In some cases, we extended the production capacity buffer by one week. We consider this to be consistent with actual practice, as production schedulers usually want tactical insight into the minimum horizon length required for a set of jobs at any given time. All problem instances are solved by both WDO and PSO.

The number of columns in random keys matrix is set to be ten times the number of batches for both heuristics. Each column represents one air parcel (WDO) or particle (PSO). As problem complexity increases with problem size (number of batches), varying the size of random keys matrix proportionally with the problem size may potentially help to search the solution space more effectively though at the expense of increased computing time. The initial keys used for position and velocity value are generated randomly. They are then perturbed and re-sequenced in order to search more effectively and quickly. For problem instances with very low due flexibility, the initial random keys matrix is divided evenly. For example, we know that the production batches with tight due time are generally assigned high priority by schedulers since the lateness penalty can be very costly to manufacturer. Therefore, in these cases one-half the random keys matrix is allocated to smaller random keys that are used for these high priority batches. The other half of the matrix has completely random keys as the normal case. In this way, we can add priority information into the initial random keys matrix to make it search more effectively, and at the same time maintain the randomness of the whole matrix.

PSO and WDO are implemented and executed using Matlab© on a 2.0 GHZ quad-core personal computer with 4-gigabyte memory. All problem instances are solved ten times using a unique, randomly generated initial population. The maximum iteration is now 2000. The best solution found in each run is recorded along with solution time. The results are summarized in Table 20. First two columns show the problem size and case id, respectively. The third and fourth columns record the best solution values found in ten repetition runs by PSO and WDO, respectively. The fifth column reports the calculated percent of solution improvement of WDO over PSO. For example, the first row, this improvement is (6225 – 5995) / 6225 = 3.69%. The sixth (Average PSO) and seventh (Average WDO) columns list the average of best solutions found in each set of ten repetitions. This can be interpreted as the overall solution quality of across all PSO and WDO runs. The last column lists the improvement in average solution values across all PSO and WDO runs. Figure 15 illustrates the average solution value improvements by cases across different problem instance sizes. It shows that WDO consistently outperforms PSO on all cases and problem instance sizes.

Negative percentage values in the two improvements columns indicate instances where PSO outperforms WDO. Among all the problem instances, only one instance (problem size 190, case 3) resulted in a solution value found by PSO was better than that was found by WDO, and the improvement is only 0.04%. For the same problem instance, the average solution quality of WDO was 3.24% better than that of PSO, and it is a noticeable difference. For all other problem instances, WDO outperforms PSO by a very remarkable margin. WDO shows an average improvement of 5.70% (best solution value) and 5.16% (average solution value) on all problem instances. Given the dominant performance results reported, we find that WDO does seem to be superior to PSO in solving the production scheduling Model (SSM) proposed in this study.

Table 20: WDO and PSO Large Sample Performance Comparison

Size	Case	Best PSO Solution	Best WDO Solution	$\frac{B_{pso} - B_{wdo}}{B_{pso}}$	Average PSO	Average WDO	$\frac{A_{pso} - A_{wdo}}{A_{pso}}$
150	1	6225	5995	3.69%	6656.4	6425.3	3.47%
150	1	7356	6789	7.71%	7704.0	7221.5	6.26%
150	2	6477	6019	7.07%	6657.7	6234.1	6.36%
150	2	5552	5031	9.38%	5859.2	5520.0	5.79%
150	3	6169	5936	3.78%	6448.7	6133.0	4.90%
150	3	5911	5303	10.29%	6075.3	5625.4	7.41%
150	4	5816	5520	5.09%	6041.1	5796.1	4.06%
150	4	6241	5390	13.64%	6389.3	5974.7	6.49%
160	1	7094	6654	6.20%	7344.7	6945.6	5.43%
160	1	6426	6152	4.26%	6764.1	6593.8	2.52%
160	2	6173	6014	2.58%	6545.7	6386.5	2.43%
160	2	6559	6140	6.39%	6800.7	6397.4	5.93%
160	3	6584	6474	1.67%	6980.8	6730.7	3.58%
160	3	7146	6632	7.19%	7427.8	7093.3	4.50%
160	4	6461	6028	6.70%	6739.4	6442.2	4.41%
160	4	6888	6521	5.33%	7261.2	6799.4	6.36%
170	1	7395	7047	4.71%	7861.8	7328.4	6.78%
170	1	7454	6908	7.32%	7730.8	7388.9	4.42%
170	2	8130	7476	8.04%	8257.9	7783.5	5.74%
170	2	7545	7371	2.31%	8168.9	7682.3	5.96%
170	3	7437	6967	6.32%	7686.8	7389.5	3.87%
170	3	6850	6246	8.82%	7133.7	6909.4	3.14%
170	4	7412	6812	8.09%	7734.4	7309.1	5.50%
170	4	6310	6289	0.33%	6985.5	6625.1	5.16%
180	1	7806	7346	5.89%	8204.7	7703.5	6.11%
180	1	7286	6981	4.19%	7884.8	7370.4	6.52%
180	2	7894	7285	7.71%	8281.7	7667.4	7.42%
180	2	7910	7125	9.92%	8171.4	7551.0	7.59%
180	3	7748	7190	7.20%	7945.1	7542.2	5.07%
180	3	7140	6410	0.22%	7452.8	7008.5	5.96%
180	4	7346	6956	5.31%	7528.9	7281.2	3.29%
180	4	7519	7428	1.21%	7978.7	7582.2	4.97%
190	1	7676	7228	5.84%	7882.2	7572.9	3.92%
190	1	7824	7222	7.69%	8163.9	7663.8	6.13%
190	2	8668	8069	6.91%	8988.3	8472.5	5.74%
190	2	8700	8276	4.87%	9072.2	8645.6	4.70%
190	3	7731	7213	6.70%	7862.7	7575.8	3.65%
190	3	7427	7430	-0.04%	7924.5	7667.8	3.24%
190	4	7385	7158	3.07%	7988.2	7427.7	7.02%
190	4	7253	6943	4.27%	7634.0	7279.4	4.65%

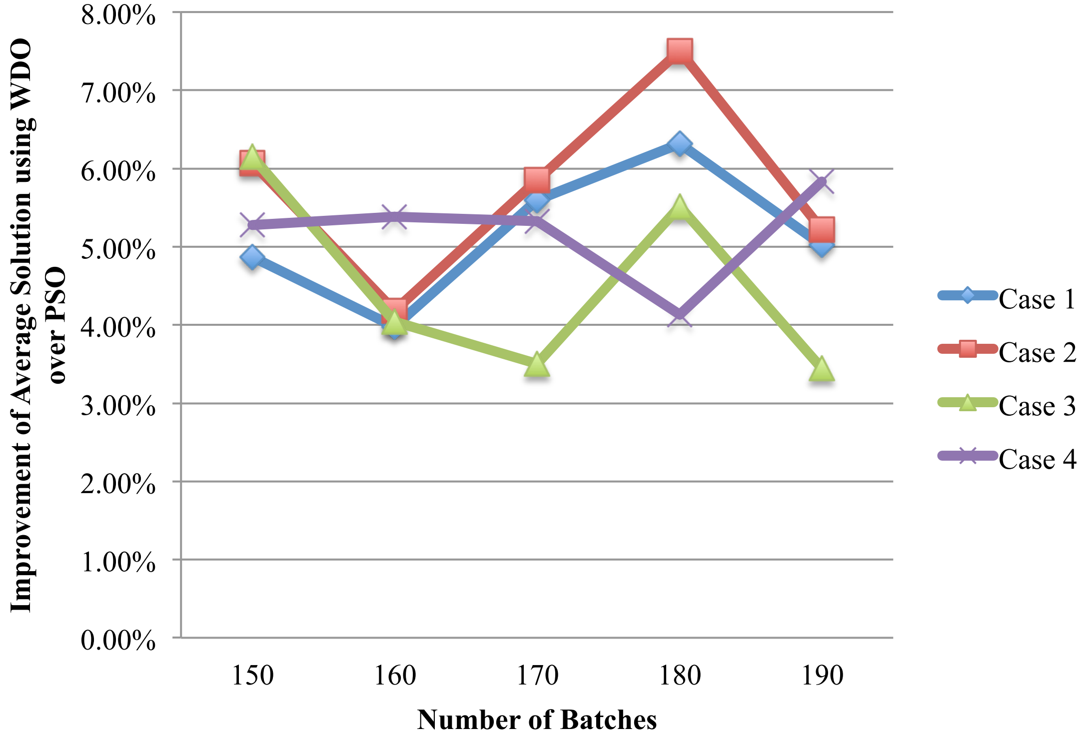

Figure 15: Percentage of Average Solution Value Improvement using WDO over PSO

CHAPTER 7 CONCLUSION

In this dissertation, we investigate a multi-stage production scheduling problem in the integrated steel manufacturing industry, and our contribution is two-fold. First, we proposed a MILP Model (SSM) to address the tactical scheduling problem for the two-stage steel manufacturing scenario (namely steelmaking and rolling decisions). The complicated nature of operations at steel plants worldwide, and the fierce market competition require more comprehensive control of cost and efficiency. It is this exact issue we are trying to address here. Second, it is well-known that sequence-depending changeover scheduling problems are difficult to solve optimally even for certain small-scale instances. Therefore we develop heuristic methods to solve large-scale problem scenarios observed in practice.

We applied a new heuristic method, WDO, to solve the proposed Model (SSM). To the best of our knowledge, this is the first study to apply WDO to production scheduling research, or any research in scheduling. Numerical parameter study is presented before solving large-scale problem instances. It provides new insights on the parameter settings when applying WDO to scheduling problems. The new findings on parameter preference for our model compared to what has been suggested by prior studies refresh the understanding of WDO, and it provides a foundation for future WDO research in different scheduling problems.

The performance of WDO is compared with a well-studied heuristic method, PSO. WDO outperforms PSO by a noticeably margin in all forty large problem instances randomly generated, except for one instance in which the best solution found by the two heuristics tied closely. The average improvement of WDO over PSO is 5.70% for the best solutions found. Average solutions found by WDO improve 5.16% over those by PSO. More research and applications are needed to make the general the conclusion that which heuristic method is superior because research on WDO is scant.

Future research areas can be developed in two streams. From the application perspective, the scheduling model could be extended to even more complex production processes, or more production stages such as the finishing stage in integrated steel production. On the heuristic side, WDO is far from being fully explored and studied. There are many open areas such as parameter settings for other applications, performance comparison with other heuristics on different problems. PSO has been studied for almost 20 years since it is originally introduced in 1995, but WDO hasn't even taken off yet if comparing it to the PSO development period. So more studies and applications are needed in the future, and they will contribute to further explore the WDO heuristic, and uncover more of its potential.

REFERENCE

As'ad, R. & Demirli, K. A bilinear programming model and a modified branch-and-bound algorithm for production planning in steel rolling mills with substitutable demand. *International Journal of Production Research*, 49, 3731–3749 (2011).

Atighehchian, A., Bijari, M. & Tarkesh, H. A novel hybrid algorithm for scheduling steelmaking continuous casting production. *Computers & Operations Research* 36, 2450–2461 (2009).

Bayraktar, Z., Komurcu, M., Bossard, J. A. & Werner, D. H. The Wind Driven Optimization Technique and its Application in Electromagnetics. *IEEE Trans. Antennas Propagat.* 61, 2745–2757 (2013).

Bean, J. Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing* 6, 154-160 (1994).

Billington, P. J., McClain, J. O. & Thomas, L. J. Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction. *Management Science* 1126–1141 (1983).

Blocher, J. D., Chand, S. & Sengupta, K. The changeover scheduling problem with time and cost considerations: analytical results and a forward algorithm. *Operations Research* 559–569 (1999).

Box, R. E. & Herbe, D. G., Jr. A scheduling model for LTV steel's Cleveland works' twin strand continuous slab caster. *Interfaces* 42–56 (1988).

Chen, Y. W., Lu, Y. Z., Ge, M., Yang, G. K. & Pan, C. C. Development of hybrid evolutionary algorithms for production scheduling of hot strip mill. *Computers & Operations Research* 39, 339–349 (2012).

Cowling, P. A flexible decision support system for steel hot rolling mill scheduling. *Computers & Industrial Engineering* 45, 307–321 (2003).

Dawande, M. *et al.* The slab-design problem in the steel industry. *Interfaces* 215–225 (2004).

Denton, B., Gupta, D. & Jawahir, K. Managing increasing product variety at integrated steel mills. *Interfaces* 41–53 (2003).

Dobson, G. & Nambimadom, R. S. The batch loading and scheduling problem. *Operations Research* 52–65 (2001).

Drexl, A. & Kimms, A. Lot sizing and scheduling—survey and extensions. *European Journal of Operational Research* 99, 221–235 (1997).

Dror, M. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research* (1994).

Dutta, G. & Fourer, R. A Survey of Mathematical Programming Applications in Integrated Steel Plants. *Manufacturing & Service Operations Management* 3, 387–400 (2001).

Fabian, T. A linear programming model of integrated iron and steel production. *Management Science* 415–449 (1958).

Fisher, M. L., Jörnsten, K. O. & Madsen, O. B. G. Vehicle routing with time windows: Two optimization algorithms. *Operations Research* 488–492 (1997).

Höhn, W., König, F. G. & Möhring, R. H. Integrated sequencing and scheduling in coil coating. *Management Science* 57, 647–666 (2011).

Huegler, P. A. & Vasko, F. J. Metaheuristics for meltshop scheduling in the steel industry. *Journal of the Operational Research Society* 58, 791–796 (2006).

Jans, R. & Degraeve, Z. Modeling industrial lot sizing problems: a review. *International Journal of Production Research* 46, 1619–1643 (2008).

Jans, R. & Degraeve, Z. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research* 177, 1855–1875 (2007).

Jones, H. G. Early OR in the Steel Company of Wales. *The Journal of the Operational Research Society* 43, 563–567 (1992).

Kennedy, J & Eberhart R.C. Particle Swarm Optimization. Proceeding of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, 1942-1948 (1995)

Lambrecht, M. R., Ivens, P. L. & Vandaele, N. J. ACLIPS: A capacity and lead time integrated procedure for scheduling. *Management Science* 1548–1561 (1998).

Lee, H. S., Murthy, S. S., Haider, S. W. & Morse, D. V. Primary production scheduling at steelmaking industries. *IBM Journal of Research and Development* 40, 231–252 (1996).

Li, H. & Shang, J. Integrated model for production planning in a large iron and steel manufacturing environment. *International Journal Of Production Research* 39, 2037–2062 (2001).

Lopez, L., Carter, M. W. & Gendreau, M. The hot strip mill production scheduling problem: A tabu search approach. *European Journal of Operational Research* 106, 317–335 (1998).

Naphade, K. S., Wu, S. D. & Storer, R. H. Melt scheduling to trade off material waste and shipping performance. *Operations Research* 49, 629–645 (2001).

Petersen, C., Sørensen, K. & Vidal, R. Inter-process synchronization in steel production. *International Journal Of Production Research* 30, 1415–1425 (1992).

Safaei, A., Husseini, S. M. M., RZ-Farahani, Jolai, F. & Ghodsypour, S. Integrated multi-site production-distribution planning in supply chain by hybrid modelling. *International Journal Of Production Research* 48, 4043–4069 (2010).

Tang, L., Che, P. & Liu, J. A stochastic production planning problem with nonlinear cost. *Computers & Operations Research* 39, 1977–1987 (2012).

Tang, L. & Huang, L. Optimal and near-optimal algorithms to rolling batch scheduling for seamless steel tube production. *International Journal of Production Economics* 105, 357–371 (2007).

Tang, L. & Liu, G. A mathematical programming model and solution for scheduling production orders in Shanghai Baoshan Iron and Steel Complex. *European Journal of Operational Research* 182, 1453–1468 (2007).

Tang, L., Liu, J., Rong, A. & Yang, Z. A mathematical programming model for scheduling steelmaking-continuous casting production. *European Journal of Operational Research* 120, 423–435 (2000).

Tang, L., Liu, J., Rong, A. & Yang, Z. A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex. *European Journal of Operational Research* 124, 267–282 (2000).

Tang, L., Liu, J., Rong, A. & Yang, Z. A review of planning and scheduling systems and methods for integrated steel production. *European Journal of Operational Research* 133, 1–20 (2001).

Tang, L., Luh, P. B., Liu, J. & Fang, L. Steelmaking process scheduling using Lagrangian relaxation. *International Journal Of Production Research* 40, 55–70 (2002).

Tang, L., Meng, Y. & Liu, J. An improved Lagrangean relaxation algorithm for the dynamic batching decision problem. *International Journal Of Production Research* 49, 2501–2517 (2011).

Tang, L. & Wang, X. Simultaneously scheduling multiple turns for steel color-coating production. *European Journal of Operational Research* 198, 715–725 (2009).

Tang, L. & Wang, X. An Improved Particle Swarm Optimization Algorithm for the Hybrid Flowshop Scheduling to Minimize Total Weighted Completion Time in Process Industry. *Control Systems Technology, IEEE Transactions on,* 18, 1303–1314 (2010).

Tang, L. & Wang, X. A Scatter Search Algorithm for a Multistage Production Scheduling Problem With Blocking and Semi-Continuous Batching Machine. *Control Systems Technology, IEEE Transactions on*, 99, 1–14 (2011).

Tang, L. & Xuan, H. Lagrangian relaxation algorithms for real-time hybrid flowshop scheduling with finite intermediate buffers. *Journal of the Operational Research Society* 57, 316–324 (2005).

Tang, L. & Yan, P. Particle Swarm Optimization Algorithm for a Batching Problem in the Process Industry. *Ind. Eng. Chem. Res.* 48, 9186–9194 (2009).

Vanhoucke, M. & Debels, D. A finite-capacity production scheduling procedure for a Belgian steel company. *International Journal of Production Research* 47, 561–584 (2009).

Vasko, F. J., Wolf, F. E. & Pflugrad, J. A. An efficient heuristic for planning mother plate requirements at Bethlehem Steel. *Interfaces* 1–7 (1991).

Vonderembse, M. A. & Haessler, R. W. A mathematical programming approach to schedule master slab casters in the steel industry. *Management Science* 1450–1461 (1982).

Wang, X. & Tang, L. Integration of batching and scheduling for hot rolling production in the steel industry. *Int J Adv Manuf Technol* 36, 431–441 (2008).

Zhong, W., Zhang, J. & Chen, W. A novel discrete particle swarm optimization to solve traveling salesman problem. *Evolutionary Computation* (2007).

CURRICULUM VITAE

Wanxi Li

Education

Ph.D., Management Science, 2014
University of Wisconsin Milwaukee, Milwaukee, WI
Major: Supply Chain & Operations Management
Minor: Marketing

B.S., Computer Science, 2003
Zhejiang University, Hangzhou, China

Research Interests

Supply Chain Management, Optimization, Supply Risk, Marketing and Information System Interface

Born in Jilin, China. Married, and has one child.