INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600
Continuous Caster Scheduling: An Optimization Approach Using Column Generation

by

Keith R. Jawahir

A thesis submitted in conformity with the requirements for the degree of
Doctor of Philosophy
Graduate Department of Mechanical and Industrial Engineering
University of Toronto

© Copyright by Keith R. Jawahir 1998
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/films, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-41555-4
Abstract

Scheduling of certain major production operations remains a chronic concern for integrated steel producers. One operation in particular, the twin strand Continuous Slab Caster, is a major contributor to overall plant profitability when utilized effectively. In this thesis, we describe the optimization based heuristic that has been implemented to support Continuous Caster scheduling needs at Dofasco Inc. (Hamilton, Ontario, Canada). The basic scheduling problem addressed is a variation of the multiple Travelling Salesman Problem with competing objectives and complex, cross tour synchronization constraints. An integer programming formulation for this scheduling problem is presented and a decomposition and column generation based heuristic solution strategy is described. The quality of the heuristic model is evaluated using statistical methods for optimal solution value estimation. In addition, economic benefits of using this approach, estimated at $1.7 million per year, are developed from direct comparison with real production schedules, using actual production system measures.

Keywords: Scheduling application, Steel production, Continuous Casting, Column generation, Decomposition, Optimization based heuristic, Multicriteria optimization.
Acknowledgements

The majority of this research effort was sponsored and supported by Dofasco Inc. of Hamilton, Ontario, Canada, with partial support from the Natural Sciences and Engineering Research Council of Canada (NSERC). My sincere gratitude is offered to the Dofasco Inc. management group of Ray Dibbley, David Parris, Bruce Gabel and Jere Hartnett for their extended support of this work over several years. In addition, I thank Daniel Buchko for providing essential assistance with all technical aspects of system development and implementation along with valuable modeling insights along the way.

I am greatly indebted to Professor Michael Carter for his academic guidance as well as for his mentoring and moral support provided throughout this developmental journey. Thanks also to Professors Francois Soumis, Mark Fox, Eric Pickett, Chelliah Sriskandarajah and Gilbert Laporte for their valuable feedback and guidance.

Finally, the greatest debt of gratitude is owed to my wife Diane, my son Alex and my family in general, whose constant love, support and patience not only inspires me but also allows me to indulge in academic and professional endeavours of this magnitude.
Table of Contents

CHAPTER 1

Introduction .................................................. 1
1.0 Integrated Steel Production Environment ................. 1
1.1 External Economic Influences ............................ 2
1.2 Primary Steel Production Process Description .......... 3
1.3 Synchronous Flow Manufacturing Issues .................. 7
1.4 Summary of Key Results .................................. 9
1.5 Thesis Organization .................................... 10

CHAPTER 2

Problem Description ........................................ 12
2.0 Caster Scheduling Requirements ......................... 12

CHAPTER 3

Literature Review ........................................... 20
3.0 Survey Focus ........................................ 20
3.1 Optimization Methods ................................ 20
3.2 Heuristic Methods ..................................... 24
3.3 Knowledge Based and Artificial Intelligence (AI) Approaches

.......................................................... 30

CHAPTER 4

Problem Formulation ........................................ 35
4.0 Mathematical Model ........................................ 35
4.1 Computational Complexity .................................. 42
4.2 Solution Challenge .......................................... 45

CHAPTER 5

Solution Methodology ........................................ 46
5.0 The Decomposition and Column Generation Technique .... 46
5.1 Applied Decomposition and Column Generation for Caster Scheduling
.......................................................... 56
5.2 Recommended Caster Scheduling System ................. 64
  5.2.1 Preprocessing Stage .................................... 64
  5.2.2 Iterative Improvement Stage .......................... 74
  5.2.3 Final Solution Stage ................................... 77

CHAPTER 6

Solution Evaluation ........................................... 79
6.0 Model Assessment Criteria ............................... 79
CHAPTER 1

Introduction

1.0 Integrated Steel Production Environment

An integrated steel plant is the traditional, multiple facility complex that has evolved over the last fifty to seventy-five years. During this period, process technology and product quality have been continuously improved at individual facilities but, with only a few exceptions, the standard metallurgical steps required for steel manufacturing have remained unchanged. Thus, modern integrated steel plants still exhibit their basic original designs and configurations, most notably, the existence of separate, large scale Primary and Finishing operations groups.

Primary steel production facilities provide the capability for progressive transformation of raw mineral products such as iron ore, limestone, coal and various alloying elements, into solid steel slabs as an intermediate step and finally, into flat rolled steel bands. Within the characteristic hot liquid or high temperature solid steel processing stages of Primary operations, customer specifications for order dimensions, chemistry, internal metallurgy and surface quality are addressed. After Primary production, the Finishing product streams process customized steel orders mainly at ambient temperatures, systematically satisfying final requirements for product thickness, ductility, surface strength and protective surface coatings.
1.1 External Economic Influences

For integrated producers, current market conditions have created extraordinary profitability demands on their Primary steel manufacturing divisions, with the traditional twin strand Continuous Slab Caster (Caster) facilities most seriously affected. Caster technology requires large, typically leveraged capital investments and must be utilized at maximum capacity to provide satisfactory annual returns. The emergence and global proliferation of new Compact Steel Production (CSP) technology, referred to as mini mills, have also had a profound impact on Caster operations. Due to inherent start up and operating cost advantages, CSP plants now dominate the market for plain carbon steel products. As a result, integrated producers have largely dedicated their resources to supplying the remaining, intensely competitive, high value added, high profit margin market sectors. Since any conventional slab casting facility has the capability for satisfying customer demand for high quality automotive, appliance, container or other critical end use material, little or no product differentiation is possible. The critical business success factor in this market environment is customer service excellence. Companies that routinely meet or exceed customer product and delivery requirements distinguish themselves from competitors and are acknowledged as the preferred suppliers within the industry. Participating in this type of market environment tends to create conflicting productivity and customer service goals for Caster operations.
Based on technical economy of scale requirements, Caster facilities are ideally suited for mass production featuring large batch sizes feeding long production runs, within which minimizing the number of product changes and maximizing produced slab dimensions provides the greatest productivity. However, customers place orders for small quantities that are typically much less than the minimum steel batch size and several orders of magnitude less than the economical Caster production run length. There is also great variety in ordered product type, dimension and delivery dates which further fragments the order base for the Caster. Consistently sound trade offs are essential for balancing the competing economic factors unique to the twin strand Continuous Slab Caster operation.

1.2 Primary Steel Production Process Description

Since customer requests are rarely supplied from stock material, the majority of orders received in an integrated steel plant are transformed into open requisitions for Primary production. This system is generally made up of the following production operations: Coke Ovens; Blast Furnaces; Melt Shops; Ladle Metallurgy and Vacuum Degas Stations; Continuous Casters and Hot Strip Mills (ref. Figure 1).
The main function of the first two units is to reduce coal, iron ore and other minerals to liquid iron in two steps. Coke Ovens convert coal to its purest hydrocarbon form called coke. Next, coke, iron ore and limestone are reduced to liquid iron following the complex, high pressure, exothermic reactions within a Blast Furnace. However, because liquid iron itself is considered a raw material for the subsequent process, detailed scheduling is not necessary in this area.

Liquid steel is the next major achievement in the sequential Primary process. Oxygen Furnaces within a Melt Shop convert liquid iron, scrap steel, various purifying fluxes and alloying elements into batches of liquid steel called *heats*. As every heat is made to satisfy a group of open order requisitions with identical metallurgical profiles, referred to as the steel *grade*, the customization process begins at this stage. Heats are produced in either 330 ton or 295 ton batches. The
smaller heat size is due to a capacity restriction at the Vacuum Degas station, a component process within the immediately succeeding Ladle Metallurgy operations group. Only a select group of extremely high quality steel grades, accounting for approximately ten percent of the overall product throughput, requires this Vacuum Degas treatment for removal of minute impurities.

Heats are transferred to the Ladle Metallurgy and Vacuum Degas area via a refractory brick lined steel ladle. In general, the processing of liquid steel through this operation group provides necessary product purification, refinement of the alloy compositions for the particular steel grade and ensures homogeneous chemistry and temperature throughout the ladle.

After the liquid steel chemistry and temperature aims are met, the next major processing step is to satisfy customer requirements for internal solid steel metallurgy, surface quality, ordered width and weight, all accomplished at the Continuous Caster. Ladles from the Ladle Metallurgy station are placed on a movable turret above the Caster. The turret is then rotated, positioning the ladle over the tundish. A tundish is a refractory lined vessel with two ceramic pouring nozzles at the bottom. This device provides the buffer between batch arrivals and the continuous flow of liquid steel necessary for the Caster. Tundishes stay in service for approximately ten heats, after which, repair and maintenance is necessary. Caster productivity is reduced but not fully interrupted during the ten
minute tundish replacement procedure also known as a flying tundish change. As liquid steel enters the two water cooled moulds of the Caster from the two tundish nozzles above, a solid shell begins to form over the surface of the steel. The top and bottom mould walls are fixed by design, producing all slabs at a thickness of 8.5 inches. However, side walls of each rectangular mould are dynamically and independently controlled, providing the capability for satisfying individual customer width requirements. Within the transit time from liquid steel entry into Caster moulds to its exit from the two 125 foot long parallel cooling lines or strands, the entire cross-section of each steel strand is solidified. At the exit end of the Caster, torches are activated at precise intervals, cutting the two continuous solid strands into slabs of specific ordered sizes.

The last major Primary production function is thickness reduction performed at the Hot Strip Mill. At this operation, the steel mass within a slab of any given width, length and thickness is transformed into a strip of equal or up to 4" less width, an average thickness of .08" and an elongation of several thousand feet. These flat steel strips or bands are wound into coils after the critical gauge reduction and water spray cooling processes. Other important product attributes controlled by this operation are surface quality, strip shape and internal properties dictated by the spray cooling practice selected. In fact, if any of gauge, internal or surface quality or band shape (i.e. cross-sectional or longitudinal profile) are out of their specification ranges, it is very difficult and often uneconomical to correct such coil
1.3 Synchronous Flow Manufacturing Issues

Both of the major value adding components within the Primary division, the Continuous Caster and the Hot Strip Mill, present complex and unique scheduling problems. Since the focus of this work is on optimization of Caster scheduling performance within the context of the integrated Primary production environment just described, certain critical steel flow coordination issues must be considered. To minimize the substantial cost of non productive time at the Caster, the Caster schedule dictates the pace for the entire Primary production system on its supply side. However, the Caster production rate can vary rapidly based on the standard casting speed developed for each grade of steel and the slab widths being produced at any particular time. The supporting Melt Shop, Ladle Metallurgy and, in particular, the Blast Furnace operation require several hours to adjust to changes in production volume. The options for dealing with short term surpluses or shortages of liquid iron or steel upstream of the Caster are extremely limited and costly. Therefore, to allow the Caster to maintain its planned production runs, the schedules must be constrained to reflect feasible liquid iron and steel supply rates.

Another costly process interaction effect that is controlled by Caster scheduling is that of grade sequencing. To allow the Caster to operate continuously and to produce the wide range of products required, heats of different steel grades must
be accommodated within any particular cast sequence. However, grade differences between successive heats in a Caster schedule result in a liquid steel mixed chemistry zone within the tundish, which further propagates into mixed chemistry slabs after solidification. The severity of the grade difference directly affects the associated transition cost incurred. Sequential casting of identical grades represents the ideal situation. When grade changes are necessary, transitions between similar or compatible grades result in a pair of mixed grade slabs on each strand, right at the heat boundary. These boundary slabs, the last two from a heat just completed and the first two from the immediately succeeding heat, must be taken out of the production flow and tested to establish exact chemistry. Mixed grade slabs are still satisfactory for customer applications with wider quality tolerances but long inventory holding times and smaller profit margins are the norm. Scheduling of radically different grades in succession is the least desirable alternative. This practice, although unavoidable at times, creates both production delays and a substantial amount of scrap material at the grade transition point.

In summary, these existing process chain interaction issues combined with evolving external economic factors and local technological and operational limitations produce a complex, daily Caster scheduling problem. Effective daily management of the Caster production stream provides both a difficult challenge and a significant opportunity for profitability improvement.
1.4 Summary of Key Results

The introductory material in this chapter covers general business conditions that both influence the Caster scheduling problem and motivate this solution effort. The solution approach was validated during the comprehensive testing and evaluation project stage which included benefit estimation using the traditional company effectiveness measures, as well as from statistically derived solution optimality assessment criteria.

A summary of optimization based model performance using the profitability related effectiveness measures is presented first. The basis for comparison is the percentage improvement attributed to model use, over and above the performance level exhibited by the existing system which relies exclusively on the decision making proficiency of the scheduling staff. Further quantification in terms of annual net dollar benefit was then provided for measurement categories where both company policy and company financial auditing capability allow. For the measures quantified in financial terms, a total annual benefit estimate of $1.7 million was established during model validation. In addition, during the same evaluation stage, model performance generated significant improvements within the delivery and productivity measurement categories.

For the assessment of solution quality, differences between model generated solution values and theoretical optimal solution values derived using several
statistical estimation methods were documented and analyzed. By the most conservative of these methods, solutions produced by this mathematical optimization technique deviate from the estimated optimal solution value by approximately 5%.

Considering all of the measurement criteria, the consensus among the approvers of this project is that the approach is sound and that it provides significant, measurable benefits. In addition, in less tangible areas such as: minimization of schedule decision making subjectivity; consistent management of competing decision making objectives; rapid adaptability to changing business conditions and the opportunity to perform off line experimentation to support improvement initiatives; the model provides insight and value that is beyond the capability of the existing scheduling system.

1.5 Thesis Organization

In the remaining chapters of this thesis, the relevant information describing the design, development and use of the mathematical optimization based solution approach for Caster scheduling is presented in detail, starting with a complete definition of the problem itself in Chapter 2. Chapter 3 provides a review of previous work in the area of Primary steel production scheduling for integrated plants. In Chapter 4 we present a mathematical formulation for this problem. This is followed by a description of the column generation solution method in Chapter
5. Validation of the mathematical properties of the solution heuristic along with estimated benefits of the model as a production tool are covered in Chapter 6. Finally, in Chapter 7, conclusions and insights derived from this problem solving effort are documented.
CHAPTER 2

Problem Description

2.0 Caster Scheduling Requirements

The complete Caster scheduling problem is an intricate combination of tasks that include constraint satisfaction, order selection, piece or micro sequencing, strand synchronization and the aforementioned grade or macro sequencing.

When creating a Caster schedule from the candidate order base, the due date and priority status of each ordered slab are the major selection criteria. This natural ranking scheme starts at the order acceptance stage. Each customer order is assigned a negotiated, promised delivery date based on when it's needed at the customer site and the latest feasible start date of each internal Finishing operation that would be required to honour the delivery commitment. During the delivery date negotiation process, a production week and day is also established for Caster scheduling purposes, referred to as the Hot Band Required Date (HBRD). This HBRD is the internal due date target for completion of all Primary processing operations, in order to satisfy the start date requirements within the Finishing division production plan.

As a rule, the HBRD indicates the desirability for including any particular ordered
slab in a Caster schedule. However, certain customer orders are also granted a priority status based on maintaining or enhancing strategic business relationships, urgent need for replacement orders that were originally produced outside of specification limits or because they belong to a special class of steel grades for which the production opportunities are limited (Ultra Low Carbon (ULC) and grades resistant to Hydrogen Induced Cracking (HIC)). A typical profile of the order base used for Caster scheduling will include late or backlogged, current and future ordered slabs, all identified by the HBRD attribute. In addition to the HBRD target, ordered slabs may also possess one or more of sales priority, replacement rush or special grade (ULC or HIC) priority statuses. During Caster schedule development, these delivery and priority based weighting factors are synthesized and used as a selection benefit estimator for the individual candidate slabs.

Once the relative slab values are established, priority slabs can be selected and aggregated into heats for liquid steel batch production. However, within each heat, slabs must also be assigned to specific production positions in either of two mutually exclusive queues representing the two Caster strands. This procedure is referred to as the piece or micro sequencing task and the goal is to minimize the magnitude and frequency of width changes between successive slabs.

Width changes are necessary to ensure product size compatibility and flexibility for the broad range of customer plant tooling specifications. The Caster has the
capability for expanding or reducing width at any time, on either strand, up to an operating limit set at 4 inches per slab. The larger the width change, the greater the chance of compromising the solidifying exterior slab shell resulting in a liquid steel \textit{break out} within the Caster. Unsuccessful width changes of this type are rare but represent a worst case scenario, requiring days of down time and substantial equipment repair cost. Minimizing the magnitude of any individual width change is considered a legitimate, proactive \textit{break out} control strategy. For the routinely successful width changes, transitional slabs are produced that are trapezoidal in shape. These transitional slabs require special processing procedures at the Hot Strip Mill to restore rectangular dimensions and additional production time and yield losses are incurred. These extra processing costs can best be reduced by minimizing the total number of width changes within a schedule.

Heat creation using the technique of designing and combining two mutually exclusive Caster strand micro sequences provides tangible customer service gains. This split strand Caster scheduling approach effectively doubles the number of available opportunities for including urgent orders and for building valuable heats when using those steel grades that historically feature large gaps between ordered slab widths. However, in order to ensure the economic viability of this or any other Caster scheduling practice, the following strand synchronization criterion must be satisfied as well. Caster strands must simultaneously complete processing the last slab in their respective queues coincident with the liquid steel heat
boundary. Since every heat is steel grade specific and metallurgical properties within the steel grade dictate a fixed and equal casting speed for both strands, the synchronization requirement is transformed into the problem of building two separate strand micro sequences, which are of equal casting duration and with combined assigned slab weights totalling 330 tons (295 tons for ULC and HIC grades). This safeguard precludes any slab from straddling a heat boundary and becoming a product of unusable quality.

To this point we have described the interdependence among order selection, micro sequencing and strand synchronization scheduling factors, all within the context of individual heat generation. However, a Caster schedule is typically made up of two or more strings of individual heats referred to as cast sequences. The main objective for cast sequence development is the minimization of total grade transition penalties within the sequence as previously indicated in the grade or macro sequencing task description. In addition, when cast sequences are assembled into a Caster schedule, a costly set up time of approximately forty minutes is incurred between successive sequences. Thus, the longest possible cast sequences are preferred.

At the cast sequence building level, a number of critical feasibility criteria must also be resolved. These constraining factors cover a wide range of issues, from equipment limitations and operating policy decisions to complex process interaction
restrictions. One of the most difficult constraints imposed on the cast sequence development function involves steel flow pacing over a moving six hour operating time horizon. Considering that the production of wide slabs consumes heats of steel faster that the production of narrow slabs, operating for too long at either width extreme has serious operating cost consequences. For example, the Melt Shop and Blast Furnaces cannot sustain production at slab widths that consume more than eight heats in any six hour period. Scheduled demand at this rate directly results in a cast sequence break followed by a costly forty minute set up procedure. Conversely, the consequences of extended production of narrow slabs, requiring less than six heats in any six hour period, are also severe. That is, either Blast Furnace operations must be run at very inefficient levels, or a surplus of liquid iron must be poured into pig iron moulds before it freezes in the transport vehicles.

Another major concern is the availability and utilization of tundishes. Due to the number kept in service and the duration of the repair and maintenance cycle, a maximum of three tundish changes are allowed in a day. This restriction impacts on Caster scheduling in the following way. In order to fully utilize tundish life, radical grade changes that trigger a flying tundish change should be avoided or should occur in conjunction with or close to the planned tundish changes for maintenance.

Precedence restrictions exist and are important to consider. The special ULC and
HIC grades can only be scheduled at the start of a cast sequence after the forty minute set up time and, when scheduled, a minimum of three heats and a maximum of five heats are allowed. These grades are precluded from appearing elsewhere in the schedule due to both the risk of contamination in the tundish from other grades and the extra processing time requirement for Ladle Metallurgy and Vacuum Degas treatments.

A key condition affecting both ease of development and final sequence lengths is encountered during cast sequence building. A partial cast sequence can be extended only if a heat is available or can be created, such that, its strand widths are within the 4 in. width matching tolerance range of a target heat at either the head or tail end of the sequence under construction. Numerous other constraints are imposed on the Caster scheduling function originating not only from specific technical limitations such as the width matching requirement above, but also from a variety of preferences or potentially relaxable rules, empirically developed for operating cost reduction or productivity, quality and customer service improvement.

Effective constraint satisfaction is the basic prerequisite for Caster schedule feasibility. However, for a scheduling problem solution to be judged as high quality or ideal, it must also concurrently address the maximization of on time delivery and productivity goals while minimizing inherent grade and width transition penalties.
In Box and Herbe's (1990) heuristic solution proposal, they describe the Caster scheduling problem as analogous to two knapsack constrained salesmen travelling on two separate but interdependent itineraries. The goal is to accumulate the largest net benefit by maximizing due date or order priority value for visiting cities (i.e. selecting slabs), while minimizing the inter city grade and width change travel costs incurred in the tour. Ignoring, for the time being, the further complication that the two salesmen must arrive at certain cities at the same time because of the heat boundary synchronization issue and other major process interaction constraints, the core Caster scheduling problem is comparable to a family of problems which are all generalizations of the well known, \textit{NP-Hard} travelling salesman problem (TSP).

These TSP generalizations are similar in that there is no requirement for visiting all available cities. The common emphasis is on efficient tradeoffs among rewards accrued from visiting cities, transition costs incurred from inter city travel and, in some cases, lost opportunity costs for cities left unvisited. Balas (1989) discusses both theory and solution approaches for the variation of the problem identified as the prize collecting travelling salesman problem (PCTSP). Other closely related formulations are the multiobjective vending problem (MVP) of Keller (1989), Laporte and Martello's (1990) development of the selective travelling salesman problem (STSP) and the orienteering problem (OP) defined by Golden, Levy and Vohra (1987). This entire family of related problems have been demonstrated to
be *NP-Hard* in the respective prior submissions. Since a simplified version of the Caster scheduling problem (CP) shares the problem structure common to this group of TSP generalizations, we consider it to be *NP-Hard* as well. This *NP-Hard* supposition will be supported by a formal proof using the mathematical formulation developed in Chapter 4.
CHAPTER 3

Literature Review

3.0 Survey Focus

A significant amount of material dealing with steel industry planning and scheduling concerns currently exists. Kerr (1993) provides a comprehensive survey of steel industry scheduling and coordination work with emphasis on the most recent developments. Formulations and solution approaches are presented for a variety of problem sizes and scheduling time horizons. The objective of this review is to focus on previous application work that is relevant to the Caster scheduling problem. Prior efforts in this area cover several alternative modelling methodologies, including advanced optimization procedures, expert system or artificial intelligence techniques and specialized heuristic approaches.

3.1 Optimization Methods

Any effective optimization approach must successfully control the inherent intractability of the production scheduling problem. Often, the main contribution of an optimization method is derived from some unique aspect of the problem structure that is recognized and leveraged for combinatorial complexity reduction.

To our knowledge, the earliest research specifically dedicated to Caster scheduling
was contributed by Boctor and Trémolières (1981). In this era, the scheduling problem was much different due to the rudimentary state of Caster technology, as we will now illustrate. Unlike the current situation where width changes are allowed at any time during a cast sequence and are treated as a variable, sequence dependent cost factor, the previous generation of Casters did not have that capability. Every width change necessitated a casting interruption of approximately one hour for set up and that requirement presented a formidable scheduling challenge. For the example problem used in their paper, a feasible cast sequence involved fixing a pair of Caster width settings and producing up to five 300 ton heats of steel of any grade combination. After a five heat sequence, a tundish change was required, forcing a one hour set up, at which time the opportunity to fix a new pair of widths was available. If a width change was scheduled before a tundish maintenance change, a one hour set up delay would be incurred and the partly used tundish would be prematurely replaced as well. Because of this severe width change limitation, the order base had to be stratified by both grade and width. The result of the stratification procedure was the creation of numerous sub groups without sufficient orders to make either a single full heat or integer multiples of whole heats. To provide the necessary full heat batch resolution, unordered slabs had to be scheduled and subsequently managed by a costly stock slab inventory operation.

The resulting scheduling problem was mainly one of establishing a sequence of
width settings, such that, the number of required stock slabs were minimized. Other competing objectives were the minimization of casting interruptions and the minimization of the number of orders not made at the Caster and relegated for the more costly alternate steel production method of ingot casting. In addition, all candidate orders for scheduling were considered of equal priority. Three separate multiple criteria optimization problem formulations were studied and small test cases were solved using an exhaustive lexicographic enumeration method. This pioneering approach was very thorough in its recognition, analysis and development of the difficult Caster scheduling problem as it existed, but provided little in the vital area of complexity reduction for practical application.

The design and development of more recent optimization based scheduling models by Lally, Beigler and Henein (1989), Boukas, Haurie and Soumis (1990) and Takahashi et al. (1992), reflect computational efficiency concerns. These practical proposals cite necessary boundary conditions for partitioning the complex production scheduling problem into a manageable component problem. For example, a common assumption for these applications is that the combinatorial tasks of selecting and arranging orders into heats and of determining the processing order of heats at the Caster, are considered out of scope. Those difficult selection and sequencing tasks are assumed to be the responsibility of separate, higher level components of the overall planning and scheduling system. These applications focus instead on the problem of coordinating the steel flow across multiple primary
production facilities prior to and including the Caster, with the goal of developing a synchronized production timetable that minimizes global operating costs, while satisfying all the technological, logistic and temporal restrictions that tend to restrict long cast sequences.

A successful optimization based system for performing both of the difficult combinatorial tasks of order selection and order sequencing, albeit for Hot Strip Mill scheduling, is provided by Balas (1989). This application, formulated as a prize collecting travelling salesman problem (PCTSP), features several types of complementary solution techniques. Schedules are generated using specialized tour construction and improvement algorithms derived from standard travelling salesman problem (TSP) solution methods. In addition, investigation of the structural properties of the PCTSP and related component TSP and knapsack problems, led to the development of a family of facet inducing inequalities. These inequalities provide the foundation for an efficient solution method by their inclusion as cutting plane constraints within an LP relaxation algorithm. Implementation experience indicates that the overall scheduling tool is successful in delivering extremely good solutions, in minutes, as long as the number of candidate cells do not exceed an upper limit of approximately three hundred. In this context, the scheduling unit, a cell, usually refers to a group of ten to twenty slabs with similar metallurgical and dimensional characteristics along with comparable processing requirements.
3.2 Heuristic Methods

To provide practical solution strategies for steel production scheduling applications, the optimization approaches utilized various problem scope or data set bounding schemes. In each of those cases, critical assumptions were made regarding which of the necessary scheduling component tasks and rules were considered within the scope of the particular procedure and how large a scheduling time horizon to target. In contrast, heuristic solution methods allow a more comprehensive problem scope while providing solutions that are both timely and of good quality as judged by the clients of the implementations.

For example, both Box and Herbe (1988) and Lee et al. (1996) discuss the heuristic system for scheduling a twin strand continuous Caster at LTV Cleveland Works. This work is particularly relevant because both the production environment described and the defined scheduling problem are very similar in problem scope and operating conditions to the Dofasco Caster scheduling application.

The LTV system is designed to produce a cast sequence from customer orders while attempting to manage the following four objectives:

- *Maximization of on time delivery* by giving priority to orders with current due dates and to orders that are late.
- *Maximization of Caster productivity* by scheduling long cast sequences which are designed to fully utilize the in service life of available tundishes.
- **Satisfaction of quality requirements** by placing orders of the highest quality early in the cast sequence and by minimizing the number and magnitude of width transitions and casting speed changes.

- **Minimization of stock material** (i.e. slabs with no customer order) inserted into the cast sequence to prevent violations of width change, heat boundary or steel consumption rate constraints.

The major constraints considered within the scope of the LTV model includes:

- **Grade compatibility** restrictions to prevent radical grade changes within a cast sequences.

- **Width change** limitations reflecting the Caster capability for only gradual width changes within a cast sequence.

- **Heat batching** requirement ensuring that each allowable grade group within a cast sequence contains an integer multiple of full heats.

- **Strand balancing** or synchronization requirement for each strand to complete a whole slab at each heat boundary.

- **Steel supply** limitations forcing the Caster scheduling model to maintain an aggregate strand width (i.e. steel consumption rate) that is within the capability of the iron and steel production units upstream.

Recognizing the inherent complexity of this scheduling problem, the decision was made to design a solution approach based on partitioning the problem into a
manageable series of precedence related tasks or subproblems. After partitioning, a special purpose heuristic solution algorithm was developed for each problem component. Only some of the details of the actual algorithms used within each subproblem were documented in the submissions but the basic goal of each solution stage was well described.

An initial complexity reduction strategy to support successful problem solution at this site was to reengineer a portion of the scheduling and commercial business processes. This activity addressed the grade compatibility concerns by combining as many similar grades as possible into groups called cast families. The cast family was designed to be a superior grade, incorporating the tightest metallurgical specifications of each of the individual component grades. Complementing this grade rationalization procedure was a policy decision to consider a cast sequence feasible for production if and only if it was unique to a cast family. That is, orders for grades from different cast families could not be included in the same cast sequence. The tactical trade off that was made was to minimize the impact of grade change cost as a decision criteria in daily Caster scheduling at a cost of providing every customer with a higher quality product than ordered for the same price (i.e. a product made to the more stringent cast family specification at a higher production cost).

For the daily cast sequence scheduling procedure, the first step is to select the
candidate orders for casting, within each cast family, without concern for the exact sequence in which the orders will be produced at the Caster. Within the selection procedure, current and late orders are highest priority and are selected first. Other, lower priority orders are then added to avoid constraint violations on width changes and other technical Caster limitations, ensuring that the final collection of orders in any particular cast family can be feasibly cast.

The next solution step is to apply steel flow constraints and profitability measures to further reduce the number of candidate cast families and their potential cast sequences. The selected orders within each cast family are evaluated to ensure that they can sustain Caster production at aggregate widths between 110 and 120 inches, with one strand running from wide to narrow and the other from narrow to wide. In addition, cast families that do not have either sufficient orders to at least meet the minimum cast sequence length requirement or a sufficiently high average order priority are also eliminated from further consideration.

Among the remaining candidate cast families, a more detailed evaluation is conducted using a pseudo cost function. This function is designed to measure the relative benefit of one potential cast sequence over another and includes such variable sequence evaluation factors as savings for casting the selected orders versus relegation to the ingot production stream, hot strip mill penalty cost for correcting differences between planned cast slab widths and ordered slab widths,
penalty cost for deviations between planned cast slab lengths and hot strip mill reheat furnace hearth size, due date weights, sequence set up cost, predicted yield loss, productivity penalty for width changes and tundish utilization costs. Various alternative cast sequence templates are generated featuring unique pairs of starting strand widths and associated, feasible width change settings during casting. The available orders are mapped onto the cast width templates and the evaluation function measures the overall goodness of fit by accumulating the individual function contributions of each order. The best orders to template match as judged by the evaluation function is selected for final cast sequence scheduling.

The final cast sequence scheduling stage requires the best fit orders to be sorted in descending order of quality requirement and secondarily by descending due date priority as calculated for previous evaluation function use. A final schedule is then sequentially constructed by searching the sorted orders in top down fashion and assigning the first available order that matches the width setting on the strand with the lower cumulative casting time. Once the final cast sequence is assembled, a timetable for the required heat arrivals at the Caster is developed and issued to the Steelmaking department.

A comparable multiple criteria optimization problem is discussed in the submission by Neuwirth (1993). The daily Caster scheduling function at the Stahl Linz facility of Voest Alpine was formulated as a mixed integer problem (MIP) and several
solution methods were evaluated. Direct solution of the full MIP formulation using a branch and bound procedure was initially attempted, resulting in unacceptable solution time requirements. An alternative solution approach was developed featuring matrix transformation and decomposition. This method was also considered impractical because the quality level of the final solution did not consistently exceed the minimal expectations of the users. The recommended solution scheme in this case requires the representation of scheduling choices as branches within a decision tree and the application of a heuristic search strategy, with final schedule selection reflecting an acceptable compromise between search time and solution quality.

Another comprehensive and practical application of optimization based heuristics for Caster scheduling was developed for Geneva Steel, as described by Tong, Silverman and Clausen (1994). With this system, a highly constrained, multiple criteria schedule optimization task is effectively managed by a variety of sequence construction and improvement heuristics. Initial order selection and arrangement into Caster production lots, called strings in this case, is accomplished through the application of a heuristic clustering technique. The main goals for this Caster string generation procedure are to maximize Caster utilization (i.e. string lengths), to minimize grade transition costs and to satisfy scheduled maintenance requirements. After strings have been developed for all production periods within the target scheduling horizon, another heuristic routine constructs an initial, feasible series
of strings for casting. Main issues resolved at this stage include Caster set up cost, crew availability and utilization of limited tundish and nozzle life. The initial schedule is then transformed into what the authors refer to as a near optimal Caster schedule, using a pair of complementary travelling salesman problem tour improvement heuristics.

In each of these cases, heuristic approaches for Caster scheduling were successfully implemented because they addressed all of the required order selection, batching, sequencing and constraint satisfaction tasks. The complex problem was partitioned to allow a sequential approach with iterative improvement. As each sequential step of the solution procedure was executed, specific constraints were satisfied and certain objectives were achieved while ensuring that the gains made at prior stages were not lost. Unlike the solution method developed in this thesis, concurrent optimization of all objectives subject to the simultaneous application of all Caster scheduling constraints was not attempted in any of these heuristic applications.

3.3 Knowledge Based and Artificial Intelligence (AI) Approaches

Similar to the specialized heuristic methods, scheduling approaches employing knowledge based or other advanced AI search oriented techniques target the full scope and complexity of the highly constrained order selection and sequencing problem. Intractability concerns are addressed in some cases by designed interaction with the human decision maker for choices and solution direction. Other
successful AI based proposals focus on the development of solution search methods guided by known constraints to ensure schedule feasibility and by emulation of preferences, logic and rule relaxation strategies of highly effective schedulers to enhance schedule quality and profitability.

Numao and Morishita (1989) describe the cooperative method for Caster scheduling implemented at NKK's Keihin plant under the commercial name Scheplan. The problem addressed is that of generating a workable coordination plan for an operation with three Basic Oxygen Furnaces supplying nine Ladle Refining Stations, which, in turn, feed five Continuous Casters. Given a predetermined sequence of heats to be made at each Caster, the goal is to develop a timetable for all supporting operations that will minimize waiting time in the system and still meet the casting requirements. The implemented solution fully integrates model algorithms with user interaction to produce feasible and effective coordination plans. System architecture features a scheduling engine for satisfying global temporal precedence requirements and equipment capacity limitations; an accompanying rule based algorithm for applying local, domain specific constraint conflict resolution heuristics; and a user interface to incorporate the knowledge and experience required throughout the iterative schedule development, improvement and approval process.

The major advantage of the cooperative approach is that it solves the problem of
capturing, a priori, all of the logic, fixed rules, allowable rule relaxations and exceptions along with the intangible elements of expert judgement and discretion. Instead, synergy is achieved between a decision maker who manages the difficult trade offs for improving schedule quality and the efficient model algorithms that develop and maintain schedule feasibility in a highly constrained environment. Other measurable benefits of the Scheplan system include a reduction in schedule generation time from three hours to thirty minutes and a fifty percent reduction in costly coordination delays.

Another system developed using the cooperative scheduling concept is implemented under the name VAI-SchedEx at the VOEST-Alpine plant. Stohl and Snopek (1993) describe a Caster scheduling system in which user interaction, via a graphical interface, is both a necessity and a major asset for directing the iterative, multiple stage schedule development process. Surrounding this user decision making core are knowledge based schedule generation, constraint violation checking and various solution assessment, manipulation and improvement routines. The automated system components use intelligent search techniques for first creating, then combining heats to form meta groups capable of being sequentially cast. By accumulating a total score for each meta group from the individual multi criteria scores assigned to each component heat, the system provides the user with a relative measure for schedule quality evaluation, assessment of improvement opportunities and for justification of the final schedule selection.
An alternative to the Numao and Morishita (1989) cooperative approach for dealing with multiple facility scheduling and coordination is described by Dorn and Slany (1984). The application, developed for the Böhler steel company, creates coordinated schedules for a steelmaking complex featuring three Electric Arc Furnaces (EAF) supplying three Ladle Treatment furnaces, which service two Vacuum Decarburation units and subsequently, one twin strand Continuous Caster and five Ingot Teeming stations. The steel produced is of the highly alloyed variety and a major scheduling concern is the sequencing of heats at the EAF such that the residual alloying elements of one heat is compatible with the alloying requirements of the heat immediately following.

The solution method that was implemented featured an initial, single pass constraint satisfaction scheduling procedure followed by an iterative schedule evaluation, repair and improvement process. At the initial scheduling stage, all heats are assigned an importance or priority ranking based on a delivery date factor, unique processing restrictions, demand for bottleneck resources and potential alloying flexibility. The initial schedule is assembled by the sequential assignment of heats to feasible production opportunities, from highest rank or most restrictive to lowest rank or least constrained. For the schedule improvement stage, a goodness of fit constraint satisfaction score is estimated for each heat based on its position in the initial schedule and those values are accumulated for an overall evaluation measure. Another rule based procedure is then applied, allowing the
systematic interchange of heats within the schedule, progressing from lowest to highest score on the constraint satisfaction scale. The improvement process terminates whenever the overall evaluation measure stops increasing.

These knowledge based approaches focus on constraint satisfaction and schedule value assessment using various cumulative priority ranking or penalty functions. The main responsibility for controlling the iterative schedule improvement process and for resolution of conflicting objectives and constraints rests with the decision maker. Thus, while these systems provide good Caster schedules, the final outcome is still predisposed to scheduler subjectivity and variability.
CHAPTER 4

Problem Formulation

4.0 Mathematical Model

Based on the review of prior Caster scheduling work, it is evident that certain rule
based or directed search techniques are extremely effective for the constraint
satisfaction aspects of the scheduling problem (CP) but are not ideally suited for
satisfaction of problem needs in the area of multiple criteria decision making. A
mathematical optimization approach was adopted because it does provide an
inherent capability for handling competing objectives in the presence of numerous
operational constraints. An associated integer programming formulation now
follows, preceded by relevant notational definitions.

CP - Full Formulation

Indices

i,j  subscript identifying individual candidate slabs, $\in N$

h  subscript identifying caster strands $\in (1,2)$

k,l  subscript or superscript denoting heat lots created, $\in V$

m  subscript or superscript specifying cast sequences generated, $\in \Phi$
Determining, in advance, how many heats or cast sequences will be created and used (i.e. cardinality of sets V and Φ) is not possible. However, within the solution method discussed in the next chapter, the specific number of heats and cast sequences are established by a model initialization routine and those numbers are dynamically updated whenever new solution options are generated.

**Coefficients and Parameters**

- $p_i$: premium(+) or penalty (-) for selecting slab $i$
- $q_{ij}$: adjacency penalty for slab $j$ following slab $i$ on a strand within a heat
- $r_{kl}$: grade transition penalty for heat $l$ following heat $k$ in a cast sequence
- $t_i$: casting time for slab $i$ (min.)
- $d$: allowable synchronization gap (min.) between the two strands at any heat boundary
- $w_i$: weight of slab $i$ (tons)
$w_{d_i}$ width of slab i (ins.)

$s$ allowable strand width difference (ins.) between adjacent heats in a cast sequence

$c$ weighting factor for biasing solution procedure toward longer sequences

$H_u, H_l$ upper, lower limit (tons) for the two types of heats produced: for regular grades (refered to as 330 ton heats), $H_u=349$ & $H_l=320$; for ULC & HIC grades, $H_u=305$ & $H_l=290$

$I_u, I_l$ upper, lower bound on average Caster steel consumption rate in minutes per heat ($I_l=45$ & $I_u=55$). These are reference parameters used during cast sequence building to trigger a dynamic penalty for heats that, if selected, would result in an avg. cast sequence speed lower than $I_l$ or higher than $I_u$. Cast sequences with average cast speed between $I_u$ and $I_l$ will not violate the iron supply constraints.

$B_{h_k}, B_{h_k}^*$ starting slab width, ending slab width on strand h in heat k (ins.)
**Variables**

\[ X_{hij}^k = \begin{cases} 1 & \text{if slab } j \text{ follows slab } i \text{ on strand } h \text{ in heat } k; \\ 0 & \text{otherwise} \end{cases} \]

\[ X_{h0j}^k = \begin{cases} 1 & \text{if slab } j \text{ is the first slab on strand } h \text{ in heat } k; \\ 0 & \text{otherwise} \end{cases} \]

\[ X_{h0i}^k = \begin{cases} 1 & \text{if slab } i \text{ is the last slab on strand } h \text{ in heat } k; \\ 0 & \text{otherwise} \end{cases} \]

\[ Y_{kl}^m = \begin{cases} 1 & \text{if heat } l \text{ follows heat } k \text{ in cast sequence } m; \\ 0 & \text{otherwise} \end{cases} \]

\[ Y_{k0}^m = \begin{cases} 1 & \text{if heat } k \text{ is last in cast sequence } m; \\ 0 & \text{otherwise} \end{cases} \]

\[ Y_{0l}^m = \begin{cases} 1 & \text{if heat } l \text{ is first in cast sequence } m; \\ 0 & \text{otherwise} \end{cases} \]

\[ Z_{nik} = \begin{cases} 1 & \text{if slab } i \text{ is assigned to strand } h \text{ in heat } k; \\ 0 & \text{otherwise} \end{cases} \]

\[ A_{km} = \begin{cases} 1 & \text{if heat } k \text{ is assigned to cast sequence } m; \\ 0 & \text{otherwise} \end{cases} \]

**Objective**

\[
\text{Maximize} \quad \left\{ \sum_k \sum_i \sum_h P_i Z_{hik} + \sum_m (c_k \sum_k A_{km}) - \sum_m \sum_k \sum_l r_{kl} Y_{kl}^m - \sum_k \sum_i \sum_j \sum_h q_{ij} X_{hij}^k \right\}
\]

For the mathematical formulation, we reflect the multiple goals of Caster
scheduling by a linear combination of four objective function factors. Maximizing
the delivery goal is represented by the first term that accumulates contributions for
each slab \((i)\), used in a heat \((k)\). To address the productivity aim of favouring the
longest possible cast sequences, the scaling factor \((c)\) is used in the second term to
effect a direct relationship between the number of heats assigned to a cast sequence
and the associated objective function contribution of the cast sequence. Consistent
with the previous problem description, in order to accrue delivery and productivity
benefits, our formulation recognizes that resources must also be consumed. These
countervailing cost factors are represented by the last two objective function
components, accumulating grade and width sequencing penalties respectively.
Solutions generated using this multiple criteria objective function design will
address the simultaneously competing goals of the Caster scheduling problem in
a consistent way.
Constraints

\[ \sum_{j - (0, M)} X_{hij}^k - Z_{hik} = 0 \quad \forall i \neq j, h, k \] (1)

\[ \sum_{i - (0, M)} X_{hij}^k - Z_{hjk} = 0 \quad \forall j \neq i, h, k \] (2)

\[ \sum_{l, n \in (0, M)} Y_{kl}^m - A_{km} = 0 \quad \forall k, m \] (3)

\[ \sum_{k, n \in (0, M)} Y_{kl}^m - A_{lm} = 0 \quad \forall l, m \] (4)

\[ \sum_{k, n \in (0, M)} Z_{hik} \leq 1 \quad \forall i \] (5)

\[ \sum_{m} A_{km} = 1 \quad \forall k \] (5')

\[ H_l \leq \sum_{i} \sum_{k} w_i Z_{hik} \leq H_u \quad \forall k \] (6)

\[ \sum_{i} t_i Z_{1ik} - \sum_{i} t_i Z_{2ik} \leq d \quad \forall k \] (7)

\[ \sum_{i} t_i Z_{2ik} - \sum_{i} t_i Z_{1ik} \leq d \quad \forall k \] (7')

\[ B_h^k = \sum_{j} (wd_j) X_{h0j}^k \quad \forall h, k \] (8)

\[ E_h^k = \sum_{i} (wd_i) X_{h0i}^k \quad \forall h, k \] (9)

\[ (B_h^l - E_h^k) Y_{kl}^m \leq s \quad \forall h, k, l, m \] (10)

\[ X_{hij}^k, X_{h0j}^k, Y_{kl}^m, Y_{k0}^m, Y_{0l}^m, Z_{hik}, A_{km} = 0, 1 \quad \forall h, i, j, k, l, m \] (11)

\[ II \leq \text{Iron & Steel Consumption Rate in any 6 hr. period} \leq I_u \] (12)

Special Grade Precedence Constraints (13)
Effective constraint resolution is equally as important as the profitability evaluation capability of this mathematical optimization approach. Accordingly, model constraints (1) through (13) combine to ensure cast sequence feasibility. Constraints (1) and (2) ensure that for each slab (i) selected and assigned to heat \((k)\), only one slab can be assigned to precede it or to follow it within the particular strand micro sequence. Note that the summation covers the cases where no slab precedes the first or follows the last slab in a heat by using special variables \(X^k_{h0j}\) and \(X^k_{ni0}\). Similarly, restriction sets (3) and (4) ensure that for every heat \((k)\) assigned to a cast sequence \((m)\), only one heat is permitted to precede or succeed it and special variables \(Y^m_{o1}\) and \(Y^m_{k0}\) are again used to address the first and last heat cases. Constraints (5) and (5′) confirm that, in the final solution, each slab is allowed to be in at most one heat and that each selected heat will be in exactly one cast sequence. These initial constraints, in conjunction with the requirement that all variables must be integer \((0,1)\) from constraint (11), are requirements of the chosen optimization formulation.

The remaining restrictions are directly related to specific, major Caster scheduling feasibility criteria. Constraints (6), (7) and (7′) combine to allow compliance with the weight and strand synchronization requirements at heat boundaries. Constraints (8), (9), and (10) jointly define the necessary width compatibility condition between adjacent heats in a cast sequence. Finally, constraints (12) and (13),
although difficult to state mathematically, represent the critical iron and steel flow pacing and special grade precedence restrictions respectively. The grade precedence constraints, discussed earlier, require the insertion of ULC and HIC grades at the start of cast sequences and nowhere else. When scheduled, at least three heats and no more than five heats of either are allowed. Since these grades are sold at premium prices, any new cast sequences started without ULC or HIC grades represent a significant lost profit opportunity.

4.1 Computational Complexity

One of the main advantages of this mathematical formulation is that it can be used for detailed comparisons with other known, formal problem descriptions. We now use this comparative capability to develop the following proof by contradiction, thus supporting our earlier assumption that Caster scheduling is an NP-Hard problem.

(a) Assume that an algorithm exists that is capable of solving the twin strand Caster scheduling problem (CP) optimally, in polynomial time.

(b) Consider now the known NP-Hard prize collecting travelling salesman problem (PCTSP) as defined by Balas (1989). The overall goal of any PCTSP solution algorithm is to determine a tour of selected cities, from a larger total set of available cities, while considering that: a cost $c_{ij}$ is incurred for travel between cities; a penalty, $c_{ii}$, is incurred for cities left unvisited; a prize, $v_i$, is collected for visiting city $i$; and a certain minimum
or threshold total prize, $V_0$, must be collected for the tour to be deemed as feasible. An optimal PCTSP solution must therefore minimize the sum of all inter city travel costs, $c_{ij}$, and non selection penalties, $c_{ii}$, while including sufficient cities to at least obtain the predetermined threshold prize $V_0$.

(c) Returning to our proposed polynomial time CP solver from (a), we represent each of the PCTSP cities as a slab, with each slab weight, $w_i$, equal to the city selection prize, $v_i$. Next, the following CP solver parameter adjustments are made:

- set the objective function width change costs, $q_{ij}$, equal to the inter city travel costs, $c_{ij}$;
- for each city $i$, set the objective function slab selection value, $p_i$, equal to the negative of the city non selection penalty, $c_{ii}$, since this is a positive term under the maximization formulation;
- set the objective function grade change costs, $r_{i}$, to be prohibitively large. Thus, only slabs of the same grade will be considered and the special grade precedence constraints (13) become redundant;
- set the objective function sequence length bias factor, $c$, to zero, eliminating the need for multiple heats;
- in constraint (12), set the lower and upper iron & steel consumption rates to extremely small and large values respectively;
- insert one dummy city with a selection value, $v_4$, that is as large as
as the minimum total prize to be collected, \( V_0 \), and with a prohibitively large non selection penalty, \( c_{dd} \). This would force the solver to assign this "slab" to one of the two Caster strands, resulting in only one remaining strand to be sequenced by the solver;

- in constraint (6), set the lower heat size limit \( H_l \) to be equal to the sum of the minimum total prize to be collected and the large dummy city prize (i.e. \( V_0 + v_{dd} \)) and make the upper heat size limit \( H_u \) arbitrarily large;

The compound effect of these parameter changes focuses the solver on the task of developing a single heat, using a single sequence of slabs, that contains at least a minimum total weight (\( \sum w_i \geq H_l \)) and for which, the total of all of the width change costs and the non selection penalties are minimal.

(d) Run the recalibrated solver and an optimal PCTSP solution is exactly the slab sequence of the optimal heat created.

Thus, the CP solver with these parameter adjustments would be capable of providing optimal PCTSP solutions, in polynomial time, implying that a problem of known \( NP-Hard \) complexity classification can be reformulated to be equivalent to a problem in the complexity class \( P \). This implication directly contradicts the current consensus complexity conjecture that \( P \neq NP \). The conclusion then is that the Caster scheduling problem is at least \( NP-Hard \) since a solution of an instance
of the CP problem could be capable of solving any \textit{NP-Hard} PCTSP.

\section*{4.2 Solution Challenge}

This formulation featuring maximization of the stated multiple criteria objective function subject to all of the feasibility criteria defined, provides a good framework for representing the \textit{NP-Hard} Caster scheduling problem. However, if a direct integer solution method was applied to this model, in its current state, several difficulties would be encountered in addition to the computational complexity concern just presented. For example, the last two objective function terms, governing the accumulation of grade and width transition penalties, imply an a priori knowledge of all of the possible slab combinations that can form feasible heats and of all of the ways of combining heats into legal cast sequences. Additional direct solution method requirements would include the linearization of the width matching constraints and the development of a linear form for representing the special grade precedence and iron and steel consumption rate constraints. Thus, the most formidable challenge for any solution method is the resolution of these intractability and non linearity concerns, while accommodating all of the feasibility and profitability characteristics of the Caster scheduling problem. To overcome these problem difficulties, as described in the next chapter, we leverage the inherent capability of the column generation approach for dealing only with relevant and manageable subsets of all of the possible heat and cast sequence combinations.
CHAPTER 5
Solution Methodology

5.0 The Decomposition and Column Generation Technique

Based on our prior discussion of the inherently intractable Caster scheduling problem, a direct integer programming solution technique for the full mathematical model is considered impractical. The implemented solution method features a decomposition and column generation approach for computational complexity reduction. This version of the methodology, pioneered by Dantzig and Wolfe (1960), was originally developed and applied for resolution of large LP models. The same technique, referred to directly as Dantzig - Wolfe decomposition or indirectly as price decomposition or simply as column generation, is equally valid for LP relaxations of integer programming (IP) formulations.

The fundamental objective of a decomposition and column generation procedure, when applied to a scheduling problem, is to partition a large LP relaxation of an IP (or Mixed Integer) problem into a master problem that is capable of selecting optimal schedules while considering availability limits on common resources (i.e. jobs to be scheduled); and one or more independent subproblems, each designed to enforce a set of special or local schedule feasibility constraints. A master problem and component subproblems exhibit a unique, hierarchical mathematical
relationship that is exploited with an iterative and practical solution strategy. The master LP problem, when solved to optimality with a subset of the total number of columns, creates dual values representing the objective function effect for incremental changes in any constrained resource. Each subproblem requires the dual values or *prices* for determining whether any new local solution will be useful to the master problem. Optimal subproblem solutions that satisfy the improvement requirement are supplied to the master problem as new columns. The master problem is solved again to find a new optimal mix of column proposals. Master problem and subproblem interactions continue, producing progressively improved master problem solutions, until no beneficial subproblem solution can be found. This procedure is guaranteed to converge in a finite number of iterations.

As an illustration, consider the master problem with its current subset of columns:

\[
\text{Maximize} \quad c^T x \\
\text{subject to:} \quad Ax \leq b \\
x \geq 0
\]

For an optimal solution of this problem, \(x^*\), a dual value \(\pi_i^*\) is associated with each R.H.S. row resource, \(b_i\), representing the instantaneous rate of change of the objective function for an incremental change in \(b_i\). Consider a new column \(A_j\) generated by a subproblem with objective function value \(c_j\). This new column will improve the current master problem solution only if the condition \((c_j - \pi_i^*A_j) > 0\) is satisfied. An interpretation of this requirement is as follows. Suppose that the \(i^{th}\)
master problem row element is included in the new column (i.e. $a_{ij} = 1$ in $A_j$). From the duality principle described, we know that the objective function will increase by $\pi_i^*$ if the R.H.S. row resource, $b_i$, is increased by one unit. However, if we use the new column, with $a_{ij} = 1$, then row element $i$ is no longer available for the old LP solution. The net result of adding the new column $A_j$ is that the column value $c_j$ is added to the objective function, but the corresponding row elements within $A_j$ are now no longer available for the old LP and the objective function is reduced by $(\Sigma_{i \in A_j} \pi_i^*)$. Therefore, for any new column $j$ to be beneficial to the master problem, the net result of the $(c_j - \pi^*A_j)$ comparison must be positive.

This mathematical principle of decomposition and column generation provides an effective strategy for solving the large, complex scheduling problems identified across a broad range of application areas. Many of these successful applications are comparable to the core Caster scheduling problem in size and scope. That is, the fundamental Caster scheduling problem structure, analogous to a generalization of a multiple TSP with cross tour synchronization constraints, is similar in complexity to problems identified in the areas of military logistics, crew scheduling for airlines and urban mass transit systems, bulk cargo shipping and vehicle routing.

The integrated hierarchical nature of the decomposition and column generation method seems well suited for diverse scheduling applications. Typically, these real
problems involve maximizing the value or minimizing the cost of selecting and processing elemental scheduling units (e.g. slabs in the Caster scheduling case or flight legs, trip segments etc. in the crew scheduling cases), while obeying a number of rules. Within a decomposition and column generation model, a master problem is developed for the sole purpose of selecting the best solution, while ensuring that each elemental scheduling unit is constrained to be included in at least one, at most one or exactly one solution as dictated by the particular application. The corresponding master problem formulations defined by these three types of constraints are known as set covering, set packing or set partitioning models respectively.

Complementing the selection oriented master problem are one or more subproblems supplying beneficial solution options in the form of columns. Since subproblems ensure development of valid combinations and arrangements of elemental scheduling units, the majority of the difficult scheduling and sequencing requirements are incorporated at this level. However, simply distributing the combinatorial work inherent in large scale optimization problems does not guarantee success, unless efficient subproblem solution methods are developed. Efficient subproblem solution algorithms are essential because subproblems are solved routinely and repeatedly throughout the solution process, providing as many improved column combinations as required by the master problem.
Goodman (1985) developed an application for assigning ships to fulfil U.S. Atlantic Fleet naval commitments, given fixed start and completion dates and specific ship resource demands. The objective of the decomposition and column generation solution strategy was to satisfy all event requirements while providing an equitable rotation of ships and even workload distributions. A set covering master problem was utilized for selecting the ideal event schedule for each ship in the fleet after feasible event schedules were generated by a subproblem and included as columns. The distinctive aspect of this approach was that the subproblem solution algorithm enumerated all feasible event schedules and supplied them to the master problem as an initial condition. A similar single step approach was developed for a bulk cargo ship scheduling problem by Fisher and Rosenwein (1989). A set packing master problem was formulated and was populated with all columns judged to be advantageous by the subproblem. Again, the subproblem solution algorithm fully enumerated all feasible options, then reduced the set by excluding all columns that were not considered beneficial by a solution cost estimation function. Such single step or so called once and for all column generation methods are only practical if the final number of solutions (columns) under consideration is relatively small. Some application specific conditions that restrict the number of possible solutions and permit a single step column generation solution strategy include: highly constrained problems, for which, effective constraint satisfaction algorithms exist; problem instances where the number of items to be scheduled are limited to several hundred at most; and problem solution
methods featuring fast and efficient enumeration of the set of feasible solution options which is then effectively pruned by pseudo cost assessment and exclusion strategies.

A more typical decomposition and column generation solution strategy is discussed by Lima (1988). The objective of this application was to create optimal ship schedules to support military crisis deployment plans. An LP relaxation of an integer set partitioning master problem supplied dual values to a column generating subproblem. The iterative procedure was executed until no more columns could be generated to improve the LP master problem, at which point, the relaxed set partitioning constraints were reintroduced and the integer zero-one problem was solved to select the best ship schedules. An interesting result from this paper was that 94% of the time, the LP relaxation phase terminated with the best integer solution (ship schedules) available and the final IP zero-one solution step was unnecessary.

To date, decomposition and column generation methods have provided significant advantages over alternative approaches for solving the complex, large scale crew scheduling applications encountered in airline and urban mass transit sectors. Successful applications in these areas provide minimal cost crew schedules that respect both the collective bargaining contract agreements defining allowable workload assignments and the fixed transportation itineraries designed for servicing
the public. For these applications, the decomposition and column generation approach typically features a master problem mathematically formulated as a set partitioning or set covering model, complemented by one or more column generating subproblems that can be solved optimally using a constrained shortest path network flow algorithm. The basic or elemental scheduling unit discussed is a *flight leg* in the airline crew scheduling case and a *trip segment* in transit crew scheduling terminology, each defining a minimum contiguous work period for a vehicle operations crew (i.e. time between potential crew change opportunities). Each basic scheduling unit is represented by a row constraint in the master problem. Correspondingly, master problem columns define valid operations crew work assignments to one or more of the basic scheduling units by setting the appropriate row and column intersection value in the constraint matrix to a value of 1. However, both the feasibility with respect to specific collective bargaining constraints and the cost of every column within the master problem is the responsibility of the unique subproblem column creation algorithms. Typically, a subproblem optimization algorithm uses dual values from a current solution of an LP relaxation of the master problem to determine if any feasible column can be created such that the current master problem crew scheduling solution will be improved. If an appropriate column exists, the subproblem supplies it to the master problem and the iterations continue. Otherwise, the current selection of master problem columns represents an LP optimal solution to the master problem. A functional crew schedule is then derived by solving the final master problem one
more time after the zero-one integrality constraints on all decision (column) variables are restored.

As an illustration, consider the successful decomposition and column generation approach for urban mass transit crew scheduling developed by Desrochers and Soumis (1988), using the modified set covering model shown below. This application produces minimal cost crew schedules defined by a subset of the (column) variables \( x_j \), each column representing a valid bus driver workday, which must cover all the trip segments represented by row constraints (1). A constraint matrix element \( a_{ij} \) is set to 1 if trip segment \( i \) is assigned to bus driver workday \( j \) and is 0 otherwise. This method, forcing constraints (1) to be \( \geq 1 \), guarantees that every trip segment will be in at least one selected workday \( x_j \). The additional constraints (2) restrict the number of workdays of special types (e.g. rush hour split shifts etc.) included in the crew schedule.

\[
\text{Min} \quad \sum_{j=1}^{n} c_j x_j
\]

subject to

\[
\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad i = 1, \ldots, m \quad (1)
\]

\[
\sum_{j=1}^{n} d_{kj} x_j \leq D_k \quad k = 1, \ldots, K \quad (2)
\]

\[
x_j = 0, 1 \quad j = 1, \ldots, n \quad (3)
\]
Critical to the success of this method is an efficient subproblem that creates minimal marginal cost bus driver workdays (based on a current set of master problem dual values) and returns them as columns for the set covering master problem. The subproblem is formulated as a constrained shortest path network flow problem and is solved optimally using a primal-dual dynamic programming method.

For airline crew scheduling problems, successful column generation approaches have been submitted by Lavoie, Minoux and Odier (1988), Vance (1993) and many others. The solution methods have centred around either the standard set covering master problem formulation (i.e. the same model defined above with constraints (2) removed) or the standard set partitioning model shown below. From the overall set of feasible column variables \( x_j \) generated, each representing a valid airline crew pairing or round trip work assignment, an optimal airline crew schedule is defined as the minimal cost subset of these column variables that includes each flight leg exactly once (row constraints (4)).

\[
\text{Min } \sum_{j=1}^{n} c_j x_j
\]

subject to
The distinguishing feature among the alternative solution methods based on similar set partitioning or set covering formulations is found in the unique subproblem algorithms developed for generating minimal cost pairings or round trips (i.e. beginning and ending at a crew base with a sequence of flight legs, rest stops, overnight stays etc.). Pairing generation is a complex task requiring specialized subproblem solution methods because it consists of nested combinatorial activities. A unique subproblem algorithm is usually designed to enumerate possible sequences of daily work responsibilities for an airline crew, referred to as duty periods, which include briefing and debriefing periods, flightlegs assigned and necessary rest stops, all governed by strict work contract limitations. Valid duty periods then become the building blocks for another subproblem solution algorithm developed to handle the combinatorial task of assembling duty periods into legal pairings for individual crews, again governed by specific work contract agreements. Duty periods can either be enumerated once, prior to column generation, and used as a fixed set of available options for a pairing creation subproblem during the column generation step, or a duty period subproblem can be utilized concurrently with a pairing subproblem to create minimal marginal cost duty periods for assembly into minimal marginal cost pairings, during each column generation iteration, as proposed by Vance (1993).
Problems of comparable complexity are also solved by column generation for vehicle routing with various types of difficult scheduling constraints. Examples of successful applications from this extensive body of vehicle routing work are described by Desrosiers, Soumis and Desrochers (1984), Skitt and Levary (1984), Desrosiers and Dumas (1988) and Ferland and Michelon (1988). In addition to the airline crew scheduling approach previously described, Vance (1993) also provides an effective column generation solution technique for the binary cutting stock problem along with a comprehensive general survey of prior decomposition and column generation methods in all of the application areas discussed. A concluding observation on application papers featuring the column generation technique is that the main contribution of each is derived from development of unique and efficient subproblem solution algorithms. Since these subproblems contain the majority of application specific scheduling constraints, comparison of solution effectiveness measures across multiple applications is difficult. Authors typically demonstrate the benefit of an approach by direct comparison with existing system performance measures.

5.1 Applied Decomposition and Column Generation for Caster Scheduling
The prohibitive combinatorial demands of Caster scheduling were addressed by the following unique decomposition and column generation procedure. For the task of selecting the best combination of cast sequences to make up the Caster schedule, a master problem was developed using a set packing mathematical formulation. We
supplement the notation previously defined in Chapter 4 to show this reformulation.

\[ \Phi \] the set of cast sequences satisfying scheduling constraints (5) - (10), (12) and (13) from the CP - Full Formulation.

\[ \beta_{im} \] a binary constant equal 1 if slab \( i \) is in cast sequence \( m \); 0 otherwise.

\[ \pi_i \] dual value associated with the \( i^{th} \) row constraint in the LP relaxation of the master problem.

\[ \sigma_k \] value of heat \( k \); sum of the individual slab values (\( p_i \)) within heat \( k \) minus the sum of width transition penalties between adjacent slabs (\( q_{ij} \)).

\[ \tau_m \] value of cast sequence \( m \); sum of the individual heat values (\( \sigma_k \)) minus the sum of grade transition penalties between adjacent heats in the cast sequence (\( r_{kl} \)). An additional, positive bias factor is added that reflects sequence length value (i.e. number of heats in the sequence times constant \( c \)).

\[ \theta_m \] binary decision variable equal 1 if cast sequence \( m \) is selected; 0 otherwise.
Master Problem

Maximize \[ \sum_{\Phi} \tau_m \theta_m \]

Subject to:
\[ \sum_{\Phi} \beta_{im} \theta_m \leq 1 \quad i \in N \quad (14) \]
\[ \theta_m = 0,1 \quad m \in \Phi \quad (15) \]

The master problem has a row constraint (14) for each slab i, ensuring its use in, at most, one sequence within the final solution. In effect, row constraints (14) and (15) within this reformulation provide the identical functionality as constraints (5), (5') and (11) from the CP - Full Formulation. For the LP version of the master problem that is fundamental to the column generation procedure, constraint (15) is relaxed, allowing the decision variables \( \theta_m \) to assume any positive value \( \leq 1 \). These decision variables correlate to the master problem columns representing feasible and beneficial cast sequences. Columns are generated, as required, by a pair of linked subproblems (ref. Figure 2).
Subproblems

After each solution of the LP relaxation of the master problem, the dual values $\pi_i$ are available for use by the heat creation subproblem and the cast sequence building subproblem. For new subproblem solutions, the value of each slab $i$ is temporarily adjusted by subtracting the associated dual value $\pi_i$. All previously created heats are then reevaluated based on the adjusted slab values. Net positive adjusted slab and heat values represent potential cast sequence components that could contribute to an improvement of the most recent master problem objective function value. All adjusted values are local in nature, specific to the particular column generation iteration stage and are used for acceptance testing of new column proposals. That is, we perform an evaluation that ensures the $(c_j - \pi^j A_j) > 0$ condition is satisfied for all newly created columns, as discussed earlier.

The objective of the first of the two subproblems, the heat creation subproblem,
is to maximize heat value, $\sigma_k$, while complying with the heat specific micro sequencing and synchronization constraints. Ideally, a subproblem within a decomposition and column generation method is formulated such that an efficient, exact solution method can be applied. For example, Vance (1993) describes optimal solution methods for constrained shortest path subproblems common to column generation approaches for airline crew scheduling and vehicle routing problems. However, even when subproblems remain complex, columns can be generated successfully by heuristic methods as in the cutting stock problem solutions developed by Haessler and Sweeney (1991) and Roodman (1986). Similarly, for this Caster scheduling application, both the heat creation subproblem and the cast sequence building subproblem discussed later remain highly complex and heuristic solution approaches were developed.

The core of the solution heuristic for heat creation is based on a *weighted nearest neighbour* or *cheapest addition* TSP tour construction method similar to an approach developed by Rosenkrantz, Stearns and Lewis (1977) (ref. APPENDIX I, Create Heat Function, section A10.0). Parallel strand micro sequences are simultaneously constructed with this heuristic, by selecting the best available slab and extending the queue on the strand with the lower cumulative casting time. The best available slab is defined as the unselected slab with the largest delivery value to width change penalty ratio, when placed in the position of either preceding the first slab or following the last slab in the target strand queue. This greedy benefit
to relative cost selection ratio was used for several reasons. First, with this method, the number and magnitude of width changes observed within heats were reduced compared to alternative heat construction methods based on simpler net benefit (i.e. benefit minus relative width change cost) slab selection criteria. During model validation, this method was also judged to be superior by the scheduling experts because it performed effectively for the difficult or limiting cases involving the selection value of a slab versus the maximum amount of width change disruption that would be willingly accepted for scheduling that slab in that particular position within the heat. With respect to the issue of solution effectiveness, the nearest neighbour approach works well for this type of application because a legal heat contains only a small subset of the total set of candidate slabs. This is an extremely important problem characteristic because it allows a greedy heuristic to avoid the prohibitively large costs typically associated with selection of final tour outliers and for returning to the place of origin, both of which are mandatory feasibility criteria for classic TSP problem solutions. Finally, since this is a subproblem solution algorithm that will be executed repeatedly during column generation improvement iterations, the computational efficiency (i.e. $O(n^2)$) of this approximate TSP solution method is a major asset.

Overall, this parallel, typically alternating heat construction strategy maintains a relatively even casting time distribution between the two strands. Balanced strand casting time becomes an essential heat feasibility condition when the combined
weight of slabs assigned to both strands are within the bounds of a full heat. Within the final schedule, compliance with the difficult heat boundary constraints, (6), (7) and (7') from the original CP - Full Formulation, is accomplished by a few simple post process adjustments to the slab lengths. That is, slab lengths on the strand with higher casting time are shortened and slab lengths on the strand with the lower casting time are increased, while ensuring that all slab weights remain between their minimum and maximum ordered limits (ref. APPENDIX I, Iterative Column Generation Stage, section A9.0).

The cast sequence building subproblem is formulated to maximize cast sequence value (adjusted $\tau_m$), while satisfying cast sequence specific scheduling constraints (8), (9), (10), (12) and (13) from the original CP - Full Formulation. Similar to the greedy heuristic developed for heat creation, the design of this solution algorithm is also a weighted nearest neighbour or cheapest addition TSP tour construction heuristic (ref. APPENDIX I, Build Sequence Function, section A11.0). As discussed previously for the heat creation subproblem, the greedy heuristic is well adapted for solution of this subproblem because only a small subset of heats are required to form a valid cast sequence.

With this method, long cast sequences are progressively developed by the selection and addition of the best available heat, either at the start or at the end of a particular partial sequence. The heat judged as the best available is a currently
unselected heat with the largest adjusted heat value to grade change penalty ratio, that satisfies all of the relevant scheduling constraints. For example, in compliance with constraints (8), (9) and (10), only those heats meeting the specific width matching criteria at either the head or tail end of the current partial sequence can be considered for extending the sequence. For candidate heats targeted for insertion at the start of a partial sequence, the special grade precedence constraints (13) are applied. In addition, a moving average of the iron and steel consumption rate required by the partial sequence is maintained. If the average consumption rate dictated by the partial sequence is within the allowable limits of constraint (12), no further action is warranted. However, if the consumption rate approaches either the upper or lower supply limit, only candidate heats with casting times that move the average in the desired, corrective direction maintain their full selection value.

The construction heuristic is considered complete either when the sequence length meets a predefined scheduling target, usually set at twenty heats or two tundish changes; when no available heat can satisfy the strict sequence extension requirements; or when no heat specifically designed to fit sequence extension requirements can be dynamically created by the heat building subproblem. Before insertion in the master problem as new columns, each cast sequence that satisfies the aim sequence length requirement is evaluated to ensure the net master problem contribution value ($c_j - \pi^* A_j$) is positive.
5.2 Recommended Caster Scheduling System

The system that was developed to solve this Caster scheduling problem consists of a preprocessing stage, the iterative column generation algorithm and postprocessing activities that include the integer programming (IP) solution step (ref. Figure 3). The algorithmic detail for this solution method is provided in APPENDIX I.

5.2.1 Preprocessing Stage

A combination of procedures are used to gather vital information for the column generation solution process. An initial, interactive routine establishes and verifies user controlled initial scheduling conditions, such as, the desired number of cast sequences, minimum and maximum cast sequence lengths, activation of special grade indicators and identification of the critical schedule effective date used to categorize backlog, current and future orders.
The next task in the preprocessing stage is the creation of two reference matrices containing the relative cost of all possible grade and width transitions respectively. Tables I and II summarize the grade and width change penalty values followed by descriptions of the processes and criteria used to achieve those consensus values. Another important initial task is the estimation of a delivery value for each candidate slab based on due date (HBRE) and priority attributes. These important customer service factors are included in Table III followed by a discussion of the assessment criteria. To address the Caster productivity issue, sequence value is increased as a function of sequence length. A linear relationship, $-15000 + c \cdot (\text{sequence length})$ where $c = 1250$, has been empirically developed to capture this vital objective function component.

### Table I

<table>
<thead>
<tr>
<th>Grade Difference Between Adjacent Heats (Δg)</th>
<th>Penalty (t_i units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Δg = no change</td>
<td>.9</td>
</tr>
<tr>
<td>Δg = <em>like</em> (<em>compatible</em>)</td>
<td>1</td>
</tr>
<tr>
<td>Δg = <em>like grade, test required</em> (<em>transition slabs applicable for some quality critical orders</em>)</td>
<td>8</td>
</tr>
<tr>
<td>Δg = <em>mixed</em> (<em>transition slabs used for less critical end use applications</em>)</td>
<td>1500</td>
</tr>
<tr>
<td>Δg = <em>severe</em> (<em>delay &amp; transition zone scrap</em>)</td>
<td>4000</td>
</tr>
</tbody>
</table>
The development of the consensus, steady state penalty and premium values initially applied in this preprocessing stage and used extensively in the two solution stages to follow, required a substantial experimentation effort over a period of several months in late 1994 and early 1995. Exhaustive sensitivity testing and scenario analyses were necessary, with full involvement of both Primary Scheduling and Caster operations staff, in order to derive the best configuration of model parameters under current, normal operating conditions. During this period, the procedure that was followed for parameter setting and validation in general was to execute the model once early in each week, creating three cast sequences. The schedulers and Caster operations staff were then jointly requested to analyze the schedule in detail, considering the delivery implications of the slabs included in the schedule compared to any valuable slabs excluded; the number and types of grade changes; the number and magnitudes of width transitions; utilization of special grade processing opportunities; and compliance with iron and steel supply limitations. Whenever the model result seemed to deviate from a standard scheduling or operating practice result, analysis and negotiation would yield a joint determination regarding whether the change represented an improvement or a deficiency. For each deficiency, the corrective action involved repeated model runs, systematically varying parameters that directly affected the desired outcome, individually at first and then collectively. When parameter settings provided suitable results for the approval group, the new values were established as the benchmark for the next calibration run at the start of the following week.
Parameter testing activity continued throughout the period, with fewer and fewer corrective actions necessary, until the values converged such that model results were consistently free of deficiencies. Although time consuming, this negotiation and consensus building process worked well because it clarified several contentious issues such as operations staff wanting minimal grade changes and thus higher penalties while scheduling staff preferred lower penalties enabling more grade change flexibility in order to schedule more priority orders.

With respect to the assignment of specific penalty values, only certain individual cost parameters could be quantified in absolute terms. These parameters were used as reference points, from which, other penalty and premium values were set by comparison. For example, the operating cost of a severe (unlike) grade change was calculated to be $8750 per occurrence and a mixed grade transition was established at $1050 per occurrence. Working from this cost base, the project stakeholders estimated model penalties of 0, 1, 10, 1050 and 8750 units for grade transitions of the equal, like, like grade - test required (LGTR), mixed and severe types respectively. As the testing and validation stage progressed, it became evident that not enough grade transitions of the severe type were being done, adversely affecting the delivery performance measure. Thus, the penalty for severe grade changes was systematically lowered until the delivery concern was addressed. Likewise, at different times, each of the penalty parameters were scrutinized individually and then collectively, until the final consensus parameter settings
shown in Table I were established.

Table II

<table>
<thead>
<tr>
<th>Width Change Amount: In or Out (Δw)</th>
<th>Penalty (qij units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Δw = 0</td>
<td>no penalty</td>
</tr>
<tr>
<td>0 &lt; Δw ≤ 100 mm.</td>
<td>Absolute value</td>
</tr>
<tr>
<td>Δw &gt; 100 mm.</td>
<td>99999</td>
</tr>
</tbody>
</table>

The true cost of width changes is difficult to measure and is not currently tracked within the existing system. However, the following significant width change concerns are shared by operations staff. Movement of the Caster mould walls in or out to allow width changes is one of several contributing risk factors for the rare but extremely costly disruption known as Caster *breakout*. This condition occurs whenever the integrity of the solidifying shell around the liquid steel core within a Caster strand is broken, allowing liquid steel to run out and to subsequently solidify over mechanical and electronic components of the Caster. Another concern at the Caster is that, in conjunction with certain product and processing factors, mould movement for width changes can result in cracking and other surface quality defects during slab solidification. After solidification, the trapezoidal shape of a width change slab also adversely affects downstream operations. When processing a tapered slab instead of the normal rectangular shaped slab, reheat furnaces at the Hot Strip Mill are less productive in terms of throughput and exhibit more
inconsistent temperature distribution within the slab. Finally, extra processing passes are required at the Roughing stand in the Hot Strip Mill to eliminate the taper within the width change slab. The greater number of passes through the Roughing stand not only takes up more production time but, when combined with inconsistent temperature distribution within a slab, exaggerates the naturally occurring undulating profile across the front and back ends of the piece, sometimes referred to as a *dog bone* or *fish tail* defect, increasing the amount of steel that must be cut off.

To address the increased risk factors associated with width changes during slab casting, the existing system imposes a combination of hard restrictions precluding width changes in certain casting situations and penalties to discourage both the frequency and magnitude of scheduled width changes. However, while schedulers considered the minimization of width penalties as a valid objective, the tendencies that emerged over the parameter validation period were that they would generally accept more frequent, small (25 or 50 mm.) width changes to select slabs that improved the delivery measure and that they would definitely incur any legal width change in order to avoid a mixed or severe grade change within a cast sequence. To reflect these preferences, a slab selection method was devised that considered delivery and grade transition objectives as dominant factors but still simultaneously incorporated the effects of the sequence dependent width change penalties. That is, as discussed during the prior description of the heat creation subproblem, all slab
selection decisions are based on the highest delivery value to slab width change ratio. This strategy was extensively tested and use of the normalized measure, delivery value divided by the absolute value of the slab width change in millimetres, provided the required consistency for assessing the joint impact of slab selection and width sequencing concerns.

<table>
<thead>
<tr>
<th>Order Characteristic</th>
<th>Premium ((p_i) units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog</td>
<td>150</td>
</tr>
<tr>
<td>Current</td>
<td>100</td>
</tr>
<tr>
<td>Future</td>
<td>1</td>
</tr>
<tr>
<td>Special Grade</td>
<td>300</td>
</tr>
<tr>
<td>Priority Customer</td>
<td>100</td>
</tr>
</tbody>
</table>

The last major area of parameter estimation and validation were the slab delivery based values. Initially, a linear scale factor was proposed, calculated as the schedule effective date minus the slab due date using the HBRD date format (i.e. two digits for year, two digits for production week and one digit for production day: YYWWD). Under this scheme, backlog orders would become increasingly valuable with age, slabs due within the current week would be of nominal value and future orders would be penalized in direct proportion to their forelog position. Follow up testing using this delivery valuation method demonstrated that small samples of very old orders that typically existed in any order base snapshot were
having a very pronounced, undesirable effect. Very costly schedules were being built around these dominant, most valued orders, at the expense of more desirable current orders. The schedulers' standard practice for dealing with the backlog, current and forelog order situation is to acknowledge backlog orders as a priority item as a group but, since these orders are already late, current customer deliveries would not be jeopardized simply to expedite the backlog orders. With this value system in mind, three simple delivery value weights were developed for orders categorized as members of the backlog, current or future order groups. In addition, for orders belonging to either priority customers or special grade groups, priority values was established to further increment the applicable delivery weight parameter. As a guide or reference point for validating the compound effect of these customer service parameters, the schedulers also provided a preferential list of cast sequence types, shown in Table IV, reflecting trade offs among the three delivery categories, the two priority factors and cast sequence length. Parameter weights for the delivery criteria and scaling factors for cast sequence length were systematically adjusted until model results indicated that schedules were being created in accordance with the breakpoints shown in the cast sequence valuation template.
Table IV

<table>
<thead>
<tr>
<th>Desirability Rank</th>
<th>Cast Sequence Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15 heats current orders, 5 heats ULC/HIC</td>
</tr>
<tr>
<td>1a</td>
<td>15 heats backlog orders, 5 heats ULC/HIC</td>
</tr>
<tr>
<td>3</td>
<td>20 heats current orders</td>
</tr>
<tr>
<td>3a</td>
<td>20 heats backlog orders</td>
</tr>
<tr>
<td>5</td>
<td>10 heats current orders</td>
</tr>
<tr>
<td>5a</td>
<td>10 heats backlog orders</td>
</tr>
<tr>
<td>7</td>
<td>15 heats future orders, 5 heats ULC/HIC</td>
</tr>
<tr>
<td>8</td>
<td>3 heats backlog orders</td>
</tr>
<tr>
<td>9</td>
<td>3 heats ULC/HIC</td>
</tr>
<tr>
<td>10</td>
<td>3 heats current orders</td>
</tr>
<tr>
<td>11</td>
<td>20 heats future orders</td>
</tr>
<tr>
<td>12</td>
<td>1 heat current orders</td>
</tr>
<tr>
<td>13</td>
<td>1 heat backlog orders</td>
</tr>
<tr>
<td>14</td>
<td>10 heats future orders</td>
</tr>
<tr>
<td>15</td>
<td>3 heats future orders</td>
</tr>
<tr>
<td>16</td>
<td>1 heat future orders</td>
</tr>
</tbody>
</table>

After focusing on each of the major parameter areas as discussed, a final round of scenario testing was completed in order to validate the concurrent interaction effects of all coefficients and parameters on final schedule quality. From these comprehensive trials, grade transition and delivery weighting factors were adjusted and fine tuned, producing the final values summarized in Tables I, II and III.
Those parameter values are considered the standard settings for model use under normal operating and commercial conditions. When a key business driver changes temporarily, the model would adapt by adjustment of the relevant parameter affected. To allow for long term model evolution reflecting the cumulative effect of many permanent individual changes, or for coping with major fundamental changes within the customer base (i.e. product mix) or within the Caster production stream, an off line concurrent recalibration of all model parameters would be recommended. The objective of such periodic analyses would be to provide updated parameter setting strategies.

After all of the required coefficient and parameter matrix comparisons are established, the solution preprocessing stage concludes with the development of a sample of initial heats (ref. APPENDIX I, Create Initial Heats, section A7.0) and cast sequences (ref. APPENDIX I, Create Initial Sequences, section A8.0). The primary objective of this procedure is to provide feasible heats and cast sequences in order to initialize and expedite the subsequent column generation process. Initial heats, maintained in an accessible heat inventory, provide a ready source of feasible, elemental scheduling options for cast sequence building. The cast sequence building subproblem is then executed to create initial columns using all initial heats as starting seeds. At this preprocessing stage starting heats are selected sequentially, in descending order of heat value, beginning with heats in the ULC and HIC grade category before exploring the remaining grades.
5.2.2 Iterative Improvement Stage

Initial cast sequences from the preprocessing stage, within an observed range of 200 to 500 columns depending on seasonality, production cycles and other factors that caused the order mix to change between model tests, provide a stable starting position for the iterative column generation method. That is, once the working set of 10,000 slabs is extracted from any snapshot of the unscheduled order file, the unique distributions of slab widths within grades affects both the number of starting slab seeds used for initial heat creation and the number of feasible initial heats that are found. The number of initial heats created then affects the number of feasible initial sequences that can be built.

For the first iteration of the column generation procedure, both the LP relaxation and the IP versions of the original master problem are solved, using LP barrier and IP branch and bound methods respectively. We capitalize on the opportunity to find an initial integer feasible solution while the associated computational demand is minimal because any reasonable integer feasible solution provides a valuable lower cut off, or search fathoming criteria, accelerating the final IP solution search and selection procedure at the conclusion of the postprocessing stage. All subsequent column generation iterations solve the LP relaxation of the master problem only, using the dual simplex method to control problem degeneracy. During this continuous improvement stage, any master problem LP solution that is also integer feasible is guaranteed to be at least as good as any integer solutions
previously found, thus providing a stronger integer solution bound.

At the core of the column generation approach, iterative solution improvements are facilitated by the reciprocal relationship between master problem LP solutions supplying needed dual value information, allowing the subproblems to produce additional columns that are guaranteed to improve the current master problem LP solution (ref. APPENDIX I, Iterative Column Generation Stage, section A9.0). The two subproblems in this application use the dual values to generate new, beneficial columns for the master problem in a very coordinated and integrated way. The cast sequence building subproblem selects the starting heat with the highest dual adjusted heat value then systematically adds on heats at either end until the growing partial sequence satisfies one of the cast sequence termination conditions. For identifying the best heat to extend a partial cast sequence, at either its lead or tail end, dual value based net benefit comparisons are conducted for all of the unselected heats within the existing heat inventory, that satisfy the strand width matching conditions. The starting widths of the first heat and the ending widths of the last heat of the partially built sequence are also simultaneously used to trigger the heat building heuristic. The goal of this dynamic heat building task is to create the highest value new heats that are specifically designed to extend the current partial cast sequence at its lead and tail ends. The final selection is made by identifying the single heat with the largest net benefit after comparing the best of the existing heats and the best of the two newly created heats. Concurrent use
of the heat building procedure within cast sequence construction for creation of custom fit heats is repeated at every sequence extension decision point. The existing heat inventory is systematically expanded to include the newly created heats, providing more feasible options during future cast sequence building evaluations.

The current target for subproblem column generation is to return up to six new and beneficial columns, per iteration, for master problem assimilation. This target for new columns per iteration, an application specific variable ranging between one and several hundred in the literature, was set through experimentation aimed at balancing time spent solving subproblems, growth of and associated time spent solving the master problem, rate of improvement of the master problem solutions and the overall time duration for the column generation process to reach one of the termination criteria. The ideal termination condition occurs when no subproblem solution is capable of further improving the master problem and thus, no columns are returned. However, in the interest of application expedience, our procedure also recognizes a compromise termination condition. After a minimum of 200 columns have been created, the improvement procedure will terminate if the total objective function increase is less than 1.5% over six consecutive column generation cycles.
5.2.3 Final Solution Stage

After the column generation process has successfully terminated, the concluding IP solution step is expedited in several ways. As discussed, to restrict the IP search space, the solution method provides an initial integer branch and bound cut off solution that is on average approximately 12% less than the final IP solution value. The initial integer cut off solution is established by solving a reduced version of the integer problem, based on initial columns only, and is improved opportunistically during column generation whenever an LP relaxation solution also happens to be a feasible integer solution. IP solution times are approximately 10% faster when using an initial cut off solution. In fact, for approximately 49% of the test cases, the column generation procedure terminated with the best integer solution and the final IP solution stage was avoided altogether.

In addition to the use of a lower cut off value to reduce the search space for the final branch and bound Caster schedule selection procedure, certain effective, problem specific search techniques and parameters were also identified. For example, use of a branching priority scheme, based on the largest ratio of objective function coefficient value to number of non zero column entries, is a major asset. Similarly, a hybrid search strategy is utilized to great advantage. Initially, depth first search is used until at least one improved IP solution is found. Thereafter, if no improved solution is found over a period of 100 depth first iterations, the search mode is switched to best node estimate for the remainder of the IP solution effort.
The combination of *depth first* sequential search early in the procedure, followed by the flexibility of jumping around and exploring multiple options at higher levels in the search tree inherent in the *best node estimate* strategy, provides a robust general solution approach for this variable, order base dependent, scheduling application.
CHAPTER 6

Solution Evaluation

6.0 Model Assessment Criteria

Results generated from extensive off line and on line testing are used to validate both heuristic solution quality and overall model capability as a production tool. The heuristic quality measure is established by comparing optimization based heuristic solution values with optimal solution values derived by statistical inference. For these model runs, the available orders for scheduling were held constant, using an actual unscheduled order file from April 9th., 1996.

For assessing the value of the model within the real operating environment, quantitative comparisons are made between model based scheduling performance and existing scheduling system performance, using the current standard production measurement criteria. In all of the comparison tests, the real unscheduled order file on the day of the test was captured, used for generation of the model based schedules and saved in a compressed form for reuse if needed. Each unscheduled order file snapshot is large, approximately .3 MB, containing all outstanding orders requiring one or more slabs, in all of the approximately 150 steel grades. A total of approximately 250,000 tons of unscheduled orders (16,000 to 20,000 slabs) are contained in a typical unscheduled order file instance, structured as
defined by the record layout shown in APPENDIX I, Section A3.0. Forty five of these comparison runs were made in 1995 using unscheduled order file snapshots from the dates shown in APPENDIX II.

6.1 Heuristic Solution Quality Analysis

The approach used for analysis of heuristic quality is based on statistical estimation of a confidence interval that would contain the true optimal solution value. That is, based on repeated independent sampling of heuristic solutions, the objective is to be able to predict with $100(1 - \alpha)\%$ confidence, or with probability $(1 - \alpha)$, that an optimal solution value is within an interval $(z_L, z_U)$. For a minimization problem, if $z'$ is the objective function value achieved for the solution heuristic under normal starting and steady state conditions, then an accepted conservative measure of the heuristic quality is:

$$r = \frac{(z_L - z')}{z_L} \quad \text{for} \quad (z_L \leq z' < 0)$$

or

$$r = \frac{(z' - z_L)}{z_L} \quad \text{for} \quad (0 < z_L \leq z')$$

In general, statistical techniques for estimation of optimal solution value, or for estimation of confidence limits around an optimal solution value as considered in this case, are based on the study of order statistics and related asymptotic extreme value analysis. Pioneering work in this area was provided by Fisher and Tippett.
(1928) with significant later contributions due to Gumbel (1956) and Galamos (1976). Successful applications of these fundamental principles for estimation of optima in combinatorial optimization problems are presented by Dannenbring (1977), Golden and Alt (1979) and Hu and Carter (1992). The following procedure for measuring heuristic solution quality is based on the similar approaches used by both Golden and Alt (1979) and Hu and Carter (1992).

Consider N independent samples, each of size m, from some parent distribution that is bounded on the left (i.e. has a lower bound). Let \( x_{i(i)} \) \((i=1,2,\ldots,N)\) be the smallest value in sample \( i \), and \( x_{(1)} \) be the overall minimal value of these order statistics (i.e. \( x_{(1)} = \min\{x_{i(i)} : 1 \leq i \leq N\}\)).

When the sample size, \( m \), becomes large, the distribution of \( x_{(1)} \) approaches the three parameter (shifted) Weibull distribution with parameter \( \theta \) defining location, parameter \( \sigma \) defining scale and parameter \( c \) defining shape (Fisher and Tippett (1928), Gumbel (1956)). The cumulative Weibull distribution for a random variable \( x \) is given by:

\[
F_x (x) = \begin{cases} 
1 - \exp\left[-\left((x-\theta)/\sigma\right)^c\right] & \text{for } x > \theta \\
0 & \text{for } x \leq \theta 
\end{cases}
\]

Using this asymptotic property of extreme value sample distributions, a maximum likelihood estimate for the location parameter \( \theta \) provides an estimate of the original parent distribution minimal value without detail knowledge of the specific parent
distribution. This implies not only that: for N independent heuristic solution samples of a reasonable size m, a limiting three parameter Weibull distribution can be generated using only the extreme (minimum) solution values from each sample \((z_{(1)}, z_{(2)}, \ldots, z_{(m)})\); but also that the estimate of its location parameter, \(\theta\), provides an estimate of the true optimal solution value \(z^*\). Further, based on the results of previous work by Golden and Alt (1979), a \(100(1 - e^{-N})\%\) confidence interval containing \(z^*\) is given by \((z_{(1)} - \sigma, z_{(1)})\) where \(\sigma\) is the maximum likelihood estimate of the scale parameter of the limiting Weibull distribution.

6.1.1 A Confidence Interval Estimation Method

The procedure for confidence limit development around the optimal solution value (i.e. the pre-requisite tasks for the desired heuristic quality evaluation) is now presented:

1. Generate a benchmark heuristic solution, \(z'\), using the normal, fixed heuristic starting conditions.

2. Generate independent heuristic solutions based on certain randomized starting and solution building conditions. For N sample groups, each of size m, a total of \(N \times m\) randomized solutions are created. The specific changes introduced for randomization of our solution heuristic is covered in the following section describing how this confidence interval estimation method is applied.
Identify the N extreme value (minimal) solution values, $z_{i0}$ $[1 \leq i \leq N]$, and arrange in increasing order \((z_{(1)} \leq z_{(2)} \leq \ldots \leq z_{(N-1)} \leq z_{(N)})\) where \(z_{(1)}\) is the overall minimum extreme value and \(z_{(2)}\) is the next lowest, etc..

Fit a limiting three parameter Weibull distribution to the N order statistics using the Maximum Likelihood Estimation methodology and programme provided by the SAS® System (© SAS® Institute Inc., Cary, NC 27513, USA).

Test and validate the Maximum Likelihood Estimate parameters for \(\theta, \sigma\) and \(c\) using Chi-Square, Anderson-Darling and Cramer-von Mises goodness of fit statistics for empirical distribution functions (EDF).

Use \(\theta\) as a point estimate of the parent population minimal value and calculate the \(100(1 - e^{-N})\)% confidence interval for \(z^*\) as:

\[
\begin{align*}
Z_L &= z_{(1)} - \sigma; \\
Z_U &= z_{(1)}. 
\end{align*}
\]

The conservative measure of heuristic quality, \(r\), is then calculated using \(z'\) and \(z_L\) as previously described.

When applying this procedure for the following Caster scheduling heuristic
evaluation, the original objective function is transformed from a maximization function to a minimization function by multiplying through by -1. This exactly equivalent formulation is bounded from below by the limiting Weibull location parameter, \( \theta \), as demonstrated in step 5 of the estimation and validation process now summarized.

(1) Benchmark heuristic solution, \( z' = -138,815 \).

(2) For this step, \( N = 34 \) sample groups, each of size \( m = 5 \), for a total of 170 randomized heuristic solution runs were generated. To allow for randomization, heuristic choices were replaced with random selections at several points within our solution procedure. At the start of each of the 170 heuristic runs, the thousands of candidate heats that were created by the initial heat construction algorithm used initial slabs selected at random. That is, in the two initial heat creation methods described in APPENDIX I (Create Initial Heats, section A7.0), the choice of starting slab was made at random instead of using heuristic choices based on the widest available slab or the most valuable slab within a homogeneous width group. In a similar fashion, when dynamically creating customized heats as required during column generation cast sequence building, a starting slab is picked at random from among the candidates slabs meeting the width matching criteria instead of by highest dual adjusted value as shown in APPENDIX.
I (Create Heat Function, section A10.0). Finally, when selecting a heat to extend a partially built cast sequence, a random choice is made from among the final four heats instead of using the highest adjusted heat value to grade change penalty ratio selection heuristic (ref. APPENDIX I. Build Sequence Function, section A11.0). As a result of replacing heuristic criteria with random choices for initial slabs in all heat creation activity and again for heat selection during cast sequence building, the sample 170 randomized heuristic solutions can be considered independent as discussed in the similar TSP tour construction heuristic evaluation of Golden and Alt (1979).

(3) The 34 extreme values (order statistics), for sample groups of 5, from the 170 independent heuristic solutions are provided in Table V. A notable result from 170 solutions of the randomized version of our heuristic was that six solutions were better than the benchmark solution produced without randomization (i.e. $z' = -138,815$). This suggests that a possible improvement strategy, although probably impractical because it takes approximately an hour to generate each solution, would be to run several hundred replications of the randomized version of the solution heuristic each time a schedule was needed. Approximately 3.5% of the multiple replication solutions could be improvements over the standard heuristic solution (i.e. be closer to the unknown optimal solution).
Table V

<table>
<thead>
<tr>
<th>-140701</th>
<th>-137641</th>
<th>-135156</th>
</tr>
</thead>
<tbody>
<tr>
<td>-140113</td>
<td>-137233</td>
<td>-135052</td>
</tr>
<tr>
<td>-139861</td>
<td>-136900</td>
<td>-134254</td>
</tr>
<tr>
<td>-139414</td>
<td>-136872</td>
<td>-133922</td>
</tr>
<tr>
<td>-138865</td>
<td>-136807</td>
<td>-133223</td>
</tr>
<tr>
<td>-138853</td>
<td>-136677</td>
<td>-133029</td>
</tr>
<tr>
<td>-138737</td>
<td>-136632</td>
<td>-132891</td>
</tr>
<tr>
<td>-138721</td>
<td>-136318</td>
<td>-132873</td>
</tr>
<tr>
<td>-138481</td>
<td>-135991</td>
<td>-132213</td>
</tr>
<tr>
<td>-138421</td>
<td>-135757</td>
<td>-131767</td>
</tr>
<tr>
<td>-138397</td>
<td>-135451</td>
<td></td>
</tr>
<tr>
<td>-138126</td>
<td>-135394</td>
<td></td>
</tr>
</tbody>
</table>

(4) A Weibull distribution with \( \theta \) (location parameter) = -141746, \( \sigma \) (scale parameter) = 5940.5 and \( c \) (shape parameter) = 2.3545 was fitted to the order statistics using SAS\(^a\) (ref. Figure 4).
For evaluating goodness of fit for this Weibull distribution, the calculated Chi-Square statistic value of 1.06009 (4 d.f.) provides a *p value* of 0.9006 indicating that null hypothesis (i.e. \( H_0 \): the Weibull distribution with the MLE parameter estimates as shown is a good fit for the heuristic solution extreme values) cannot be rejected at any reasonable \( \alpha \) value (e.g. 0.1). That is, the calculated *p value* from the test statistic would have to have been less than 0.10 to reject \( H_0 \) at the \( \alpha = 0.10 \) significance level. Further evidence for adopting Weibull \((-141746, 5940.5, 2.3545)\) as the best fit limiting distribution are the Anderson-Darling (A-Square = .2651) and
Cramer-von Mises (W-Square = 0.03696) goodness of fit test statistics, both providing $p$ values $> .25$. These two quadratic EDF tests validate the selected limiting distribution since, unlike the Chi-Square test, these test statistics and associated $p$ values are not subject to variability depending on the midpoints and grouping intervals chosen for the data.

(6) Using this Weibull distribution, the $100(1-e^{-34})\% = 100(1 - 1.71\times10^{-15})\% = 100\%$ confidence interval around the unknown optimal solution value, $z^*$, is:

$$z_L = -140701 - 5940.5 = -146641.5;$$
$$z_U = -140701.$$

In the final analysis of heuristic quality (i.e. relative error with respect to $z^*$), we compare our reference (benchmark) heuristic value of $z' = -138815$ to several optimal solution value estimates. Two of the optimal solution value estimators have been discussed, the limiting distribution MLE parameter $\theta$ and the conservative $100\%$ confidence interval lower bound $z_L$. The final optimal solution value estimators, also derived by statistical inference, are the jackknife estimators from Nydick and Weiss (1994). The same sample of 34 independently generated order statistics are used in conjunction with $z'$ for the jackknife estimates. Four jackknife estimators are defined in terms of the two lowest order statistics ($z_{(1)} = -140,701$ and $z_{(2)} = -140,113$) from Table V as follows:
SAMP = 2z_{(1)} - z_{(2)}

2z' - z_{(1)} \quad \text{if } z' < z_{(1)}

HSAMP = 2z_{(1)} - z' \quad \text{if } z_{(1)} < z' < z_{(2)}

2z_{(1)} - z_{(2)} \quad \text{if } z_{(1)} < z_{(2)} < z'

MINHS = \text{Min} \{\text{SAMP, HSAMP}\}

AVG = \frac{\text{SAMP} + \text{MINHS}}{2}

For instances where z_{(1)} < z_{(2)} < z', as is true in this case, all of the jackknife estimators are the same (i.e. estimator SAMP = estimator HSAMP = estimator MINHS = estimator AVG), all of which are equal to $z_{\text{jackknife}} = 2z_{(1)} - z_{(2)}$. The three optimal solution value estimators used to measure heuristic solution quality are summarized below in Table VI.

<table>
<thead>
<tr>
<th>Base Case Heuristic Value</th>
<th>Optimal Sol. Value Est.</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z' = -138815$</td>
<td>$z_L = -146641.5$</td>
<td>$r = 0.0534$</td>
</tr>
<tr>
<td>$z' = -138815$</td>
<td>$z_\theta = -141746$</td>
<td>$r_\theta = 0.0207$</td>
</tr>
<tr>
<td>$z' = -138815$</td>
<td>$z_{\text{jackknife}} = -141289$</td>
<td>$r_{\text{jackknife}} = 0.0175$</td>
</tr>
</tbody>
</table>

The most conservative measure of relative error for the Caster scheduling solution heuristic is for the confidence interval estimate $z_L$ at approximately 5.3%.
6.2 Estimation of Model Benefits in Production Environment

Performance of the existing computer assisted Caster scheduling system has historically been difficult to measure. Due to the stochastic nature of each of the integrated primary production processes, a round the clock schedule maintenance effort is initiated immediately after a schedule becomes active. Responding to unpredicted events by adding and deleting scheduled items is routine, often creating a substantial difference between planned versus actual production just twenty four hours later. Nevertheless, attempts are made to track actual production system performance in five areas considered to be directly affected by scheduling decisions.

The traditional system assessment factor has always been Caster productivity, measured by both average number of heats per cast sequence and by total tonnage produced per shift, day, week, month or year. However, due to existing restrictions on tundish availability, scheduling and operations management decision makers have agreed that twenty heat sequences best meet current business objectives. Although we have demonstrated that the optimization based system is ideally suited for developing longer cast sequences, with average cast sequence lengths in excess of the twenty heat per sequence target, a consensus was reached on limiting effectiveness comparisons to twenty heat sequences only. This restriction was added to the cast sequence building heuristic for all performance tests, essentially eliminating the productivity factor from the basis of comparison.
The remaining four effectiveness measurement criteria are: delivery performance, grade transition penalties, utilization of special grade scheduling opportunities and iron and steel supply constraint satisfaction. Delivery performance is measured by calculating the percentage of priority, backlog and current slabs resident within any schedule. For evaluating the impact of grade changes, the appropriate delay, inventory and lost sales costs are assigned to each occurrence of severe, mixed, LGTR or like grade transition within the model based schedules. To assess the degree of use of special grade processing opportunities at the start of cast sequences, a direct count of the number of applicable heats within the schedule is kept. In the fourth evaluation category, measuring the iron and steel supply constraint satisfaction level, the final schedule is analyzed in sequential six hour blocks (i.e. a day would be divided and analyzed in 4 production blocks of 6 hours each). Any occurrence of less than six heats or more than eight heats within any six hour block is considered a violation and is tallied.

Extensive model performance testing was completed during the last half of 1995. Testing over such a prolonged period was advantageous because it provided a variety of order base samples reflecting inherent seasonal variability. It also provided ample opportunity for thorough model validation and verification. The first analysis compared the optimization model performance against production summary statistics in the four relevant categories. Forty five sample model runs were completed during the period, generating two twenty heat cast sequences each,
for a total of ninety cast sequences. The existing system created two hundred and forty six cast sequences in the same time period. Measurement criteria for the two systems, calculated on a per cast sequence basis, are summarized in Figure 5 by percent improvement attributed to the optimization based approach.

From this initial comparison we concluded that the recommended system was robust with respect to order base variability and was capable of providing schedules that were considered to be of high quality based on the accepted measures. Another interesting finding was that delivery and productivity goals did not conflict as much as originally thought. That is, with the productivity target fixed at the acceptable twenty heat per sequence level, there was still room to improve the delivery based scheduling measure. However, since this was a comparison of average model performance versus average existing system performance over the test period, it was considered inappropriate to presume that the magnitude of improvement observed for the delivery measure was, in fact, a
consistent and reliable estimator of model effectiveness. That is, the discrete design of the testing procedure could have inadvertently introduced a bias toward positive model results. Each test solution was generated from a unique snapshot of the order base and consecutive tests were separated by several days. The real scheduling system operates continuously, dealing with an order base that tends to degrade or fragment between batch updates. The existing system's delivery measure includes these recurring periods, during which, even the best schedules are typically ranked lower.

A second, more direct comparison of the two approaches was also completed in late 1995. For this comparison, identical order base snapshots were used for both the existing and recommended systems. The schedulers created two twenty heat sequences as did the optimization based model. The two schedules were compared in detail and the same effectiveness measures were calculated. This parallel trial
was repeated seven times. The results are summarized in Figure 6, again, from the perspective of percent improvement attributable to model use.

In three out of four categories, model results were favourable. Grade transition costs were reduced by approximately 20 percent. Approximately 24 percent more heats with special grade designation were included. No instances of iron and steel supply constraint violations occurred in the model generated schedules compared with five occurrences within the existing system schedules.

For the important delivery performance measure, the existing system provided an improved result by approximately 2 percent. Several factors contributed to this improvement. The main reason is that the schedulers are highly skilled individuals who, when focusing on any of the individual criteria, can produce a schedule that is difficult to improve. This factor was quite evident during the period of direct testing when an unusually large percentage of the order base consisted of backlog and priority customer orders, allowing the schedulers to exceed their own, year to date, average delivery performance by a large margin. The second contributing factor is inherent in the model based trade off decision making process. To maximize the use of limited opportunities for scheduling special grades at the start of cast sequences, the model assigns the highest premium to applicable orders as was shown in Table III. This model weighting scheme allows future orders of the designated special grades to be considered more valuable than priority or backlog
orders of the remaining grades. The model based improvement of approximately 24 percent, for this measure, was achieved partly at the expense of backlog and priority orders within the regular grade categories. However, when the measurement criteria were considered simultaneously, the consensus by all participants was that the optimization based approach provided significant overall benefit for Caster scheduling.

In terms of converting the percentage improvements attributed to model use into tangible benefits, impact assessments were developed for three of the five measures during the test period. No further financial quantification for the delivery measure was possible since this is not currently done and Dofasco’s Capacity Management group can not provide even a conservative estimate. The productivity measure was eliminated from the comparison as discussed earlier. Cost reduction potential for grade transition and 6 hr. supply violation measures and additional revenue generation potential from the utilization of special grade processing opportunities measure were estimated as summarized in Table VII.
<table>
<thead>
<tr>
<th>Measure</th>
<th>Annual Cost (1995 est.)</th>
<th>Est. Annual Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grade Transition</td>
<td>$1,902,600</td>
<td>$380,520</td>
</tr>
<tr>
<td>6 Hr. Supply Violations</td>
<td>$1,755,000</td>
<td>$842,400</td>
</tr>
<tr>
<td>Utilization of Special Grade Opportunities</td>
<td>N/A</td>
<td>$453,020</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>$1,675,940</td>
</tr>
</tbody>
</table>

The estimated annual benefit for the grade transition measure was derived using the approximately 20 percent reduction of annual cost attributed to the optimization based model in both the direct comparison and large sample tests.

When considering the model impact on supply violations, it was estimated that approximately 20 percent of the violations included in the annual cost estimate were due to unplanned events. These random delays are inherent to the process and cannot be affected by improved scheduling practices. For the optimization based model benefit, the 60 percent reduction observed in the large sample test was applied to the remaining 80 percent of the annual cost estimate. The more conservative improvement level from the large sample test, 60 percent, was considered more realistic than the 100 percent reduction level attained during the much smaller direct comparison test.
For the **utilization of special grade processing opportunities** measure, the benefit estimate is derived from the 24% increase in annual production of this material attributed to model based scheduling. The projected increase in production tonnage was converted into an annual additional revenue estimate by using the average difference in selling price between the special grades and other steel grades. The price differential was used because the additional heats of the special grades included in model generated schedules simply replaced heats of other grades in actual schedules. Finally, to estimate the annual benefit value shown in Table VII, the ratio of net income to revenue achieved in 1995 (6.4%) was applied to the annual additional revenue estimate.

### 6.3 Execution Time

This optimization based approach is implemented using Watcom C++ 32 (© WATCOM International Corporation) in conjunction with the MIP callable subroutine libraries of CPLEX® Optimization Inc. (© CPLEX® Optimization Inc., Incline Village, NV 89451-9436, USA). The application is executed on a Pentium 120 Mz. platform under OS/2® (© IBM Corporation).

Execution times for the optimization based system over both test periods averaged 48 minutes, with a minimum of 16 minutes and a maximum of 100 minutes. Overall response times in this range are considered a major advantage, compared to the existing Caster schedule development method that requires approximately 8
hours of scheduler effort. The existing system routinely ties up resources of this magnitude because all of the major decisions are made manually. Existing computer assist tools simply perform data manipulation and summarization tasks at the direction of the scheduler. The overall process that has evolved is cumbersome and features a great deal of backtracking and iteration before a workable solution is found. In addition, past problems in applying rules considered critical for product quality and operating efficiency, have resulted in regularly recurring schedule approval meetings. This requirement demands both plant and scheduling department resources for several additional hours each week. On an annual basis, model based reduction in schedule generation time would save the equivalent of one full time employee, estimated at $50,000.
CHAPTER 7

Conclusions and Implementation Issues

7.0 Summary

In this thesis, the multiple criteria optimization problem of scheduling a twin strand Continuous Slab Caster was described. For solution of this computationally complex primary steel production application, details of an effective optimization based heuristic were presented.

Specifically, for management of the main, traditionally conflicting, problem objectives of delivery and productivity, model results consistently indicate that mutually valuable solutions are attainable. This finding provides a flexibility advantage to the corporation when responding to changes in business conditions. Relevant delivery and productivity objectives for scheduling are easily set or adjusted within the model, either to reflect equal importance and contributions of both factors or such that either individual component is treated preferentially. The other three scheduling system measures: grade transition penalties incurred, use of special grade processing opportunities, and iron and steel supply constraint satisfaction, are also concurrently improved by the optimization based solution heuristic. Model based improvements of these evaluation criteria provide significant direct benefits to the company, in both cost reduction and additional
profit opportunity.

With respect to response time, effective solutions for problems with ten thousand candidate slabs are achieved in forty eight minutes on average. Compared to existing system requirements, this represents a performance improvement of ninety percent on average and eighty percent for the worst case observed. The net benefit to the company in this area is faster reaction to and recovery from unplanned events, as well as increased availability of highly skilled Capacity Management staff for working on various major customer service and strategic planning improvement initiatives.

In general, based on experience with this application, the decomposition and column generation approach could be more widely utilized for industrial production scheduling problems. Many of these problems are an intricate combination of multiple criteria selection, sequencing, synchronization and constraint satisfaction tasks. This technique for managing the computational complexity of such large scale applications has been successfully applied for a wide array of logistic type scheduling problems but, with the exception of some well known cutting stock problem applications, relatively few production scheduling problems have been solved in this advantageous way. Decomposition and column generation provides a natural framework for separating complicating, feasibility related real world constraints from the optimality oriented, solution selection problem component.
The method, when executed to its normal termination, provides globally optimal solutions. However, depending on the solution time demands of the particular industrial application, specially designed stopping criteria can be triggered once some target solution quality measure has been exceeded.

The Caster scheduling method developed in this thesis capitalizes on the inherent advantages of the decomposition and column generation approach. Using the existing scheduling system as a benchmark, results from rigorous testing and evaluation demonstrated that the model based approach is very beneficial. Based on these evaluations, Dofasco’s Capacity Management department has approved the routine use of this tool for improving the quality of both production planning and production scheduling decision making.

7.1 Implementation Issues

The Caster scheduling system has been used for several purposes since the original version was released in July 1996. At that time, the implementation plan was designed around a daily model run, executed immediately after the order file had been updated to reflect all adjustments and transactions due to the previous day’s actual production and sales activities. The schedule that was generated would form the basis of a 3 day Caster production schedule, that would take effect the following day. Thus, every day the active 24 hour portion of the schedule would be frozen and the next 72 hours would be refreshed. However, two significant
corporate initiatives emerged causing the implementation plan to be revised.

The first priority item was that a new slab production stream, a single strand Caster fed by an Electric Arc Furnace (EAF) steelmaking facility, was implemented in the late fall of 1996. The main scheduling implication associated with the new facilities start up concerned the productivity of the new Caster. Since the EAF stream would initially use scrap steel instead of liquid iron as its main raw material, the quality capability would be limited, restricting the single strand Caster to produce only a small subset of steel grades. However, the existing twin strand Caster was to be scheduled first to ensure maximum utilization and profitability. The remainder of the orders in the order base, that were in the appropriate grade categories, would then be used for scheduling the single strand Caster. To assist the schedulers with planned, off line simulations of this emerging scheduling problem, a new single strand heat creation function was created based on similar heuristics as the original twin strand version. The same cast sequence building function and column generation solution procedures could then be used to create equally effective single strand Caster schedules. Both versions of the model were then utilized to evaluate the impact of scheduling the twin strand Caster before and after the single strand Caster. During this period, in addition to the valuable knowledge and scheduling experience gained prior to the actual start up of the new production facility, senior Capacity Management staff gained greater confidence in the capabilities of the optimization based Caster scheduling system.
Further research work is required in this area to test the feasibility of scheduling both Casters simultaneously, in other words, building a sequence for a *triple strand* Caster where two of the strands must cast the same grade and obey all of the current twin strand casting rules. Also, the EAF is capable of using liquid iron as raw material as well and would then be capable of producing the same range of grades as the twin strand Caster. The implications of these future operation states also require further investigation.

The current high priority application for this Caster scheduling system is to support several strategic planning project teams with corporate mandates requiring major delivery performance improvement and manufacturing cycle time reduction over the next three to five years. The initial group of analyses completed were focused on the cost assessment of manufacturing cycle time minimization, by operating the twin strand Caster in a flexible, make to order mode. For this work, three days of actual slab production from the Finishing Division product streams were used as order requirements for casting. This order mix contained only approximately 1,500 slabs from across a broad range of steel grades. With all current casting restrictions in place, model based schedules were created and productivity costs at the Caster were estimated. The procedure was repeated with each of the main Caster scheduling constraints relaxed in isolation (e.g. minimum number of heats per sequence set to 1, reduction in grade change penalties, relaxation of width change limitations, removal of steel consumption rate limitations, grade rationalization
(i.e. grouping of similar grades into individual super grades), removal of order priorities, etc.). Combinations of rules were then subsequently relaxed. In all cases, both the cost implication of the rule relaxations and the potential cycle time improvement were documented. Current work now focuses on the cycle time and operating cost implications of using the Caster in a make to stock mode. That is, once a steel grade is scheduled for casting, some minimum number of heats will be made with all excess slabs held and used from stock. As the stock diminishes, a reorder point is reached and another minimum multiple heat batch would be ordered.

The capability of the current Caster scheduling models is being demonstrated by the growing number of these successful project applications. As the analytical work continues, senior strategists are finding more challenging applications for the Caster scheduling models while gaining valuable knowledge and insight into potential future operating states and configurations.
References


APPENDIX I

Solution Procedure Detail

A1.0 Read initialization parameters;

See section A1.1 Initialization Parameter Detail at end of this Appendix;

A2.0 Read Grade Transition Penalty Matrix;

Grade transition penalties directly impact the cast sequence building process. These relative cost parameters are maintained within a matrix structure containing both a row and a column for every grade currently produced. The matrix is used as a lookup table to determine the penalty \( r_{kl} \) of switching from a heat \( k \) of one grade to a heat \( l \) of another grade by finding the intersection value for the row containing the current heat \( k \) and the column of the heat \( l \) to follow. The matrix is not symmetric with respect grade transition penalties (i.e. \( r_{kl} \neq r_{lk} \)) as shown in the matrix excerpt shown below. Current penalties are 0.9 if the grades are equal, 1 if the grades are 'like', 8 for 'LGTR' grades, 1500 for 'mixed' grades, 4000 for 'unlike' grades, and 49999 for 'illegal' grade changes.
<table>
<thead>
<tr>
<th>Grade</th>
<th>. . .</th>
<th>CC016</th>
<th>CC018</th>
<th>CC020</th>
<th>CC030X</th>
<th>CC040</th>
<th>CC040A</th>
<th>. . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
</tr>
<tr>
<td>CC016</td>
<td>. . .</td>
<td>0.9</td>
<td>1500</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
</tr>
<tr>
<td>CC018</td>
<td>. . .</td>
<td>4000</td>
<td>0.9</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
</tr>
<tr>
<td>CC020</td>
<td>. . .</td>
<td>1500</td>
<td>1500</td>
<td>0.9</td>
<td>4000</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>CC030X</td>
<td>. . .</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
<td>0.9</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
</tr>
<tr>
<td>CC040</td>
<td>. . .</td>
<td>4000</td>
<td>4000</td>
<td>8</td>
<td>4000</td>
<td>0.9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CC040A</td>
<td>. . .</td>
<td>4000</td>
<td>4000</td>
<td>8</td>
<td>4000</td>
<td>1</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

A3.0 Read in Order File;

A batch routine is run every day extracting all unscheduled orders from the corporate order base. The result is an ASCII (flat) file containing all the essential product information for this scheduling procedure as summarized below.
<table>
<thead>
<tr>
<th>Field</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>REFERENCE</td>
<td>NUM</td>
<td>4</td>
<td>Unique order identifier</td>
</tr>
<tr>
<td>2</td>
<td>LOT</td>
<td>NUM</td>
<td>4</td>
<td>* not used for Caster Scheduling</td>
</tr>
<tr>
<td>3</td>
<td>PROD_BLOCK</td>
<td>NUM</td>
<td>5</td>
<td>* not used for Caster Scheduling</td>
</tr>
<tr>
<td>4</td>
<td>HM_ORDER</td>
<td>NUM</td>
<td>7</td>
<td>* not used for Caster Scheduling</td>
</tr>
<tr>
<td>5</td>
<td>HB_REQXDT</td>
<td>NUM</td>
<td>3</td>
<td>Hot Band Req. Date (WWD)</td>
</tr>
<tr>
<td>6</td>
<td>HB_REQXYR</td>
<td>NUM</td>
<td>2</td>
<td>Hot Band Req. Year (YY)</td>
</tr>
<tr>
<td>7</td>
<td>HB_REQ_CD</td>
<td>CHAR</td>
<td>1</td>
<td>* not used for Caster Scheduling</td>
</tr>
<tr>
<td>8</td>
<td>PRIORITY</td>
<td>CHAR</td>
<td>1</td>
<td>Sales Order Priority Code</td>
</tr>
<tr>
<td>9</td>
<td>GRADE_CD</td>
<td>CHAR</td>
<td>1</td>
<td>Ordered Grade</td>
</tr>
<tr>
<td>10</td>
<td>GRADE_QL</td>
<td>NUM</td>
<td>2</td>
<td>* not used for Caster Scheduling</td>
</tr>
<tr>
<td>11</td>
<td>AGGREGATE FORCE</td>
<td>CHAR</td>
<td>5</td>
<td>* not used for Caster Scheduling</td>
</tr>
<tr>
<td>12</td>
<td>SPEED_X</td>
<td>NUM</td>
<td>3</td>
<td>Casting Speed (mm/min)</td>
</tr>
<tr>
<td></td>
<td>COLUMN NAME</td>
<td>DATA TYPE</td>
<td>LENGTH</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>----------------</td>
<td>-----------</td>
<td>--------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>13</td>
<td>WIDTH_X</td>
<td>NUM</td>
<td>4</td>
<td>Ordered coil width</td>
</tr>
<tr>
<td>14</td>
<td>GAUGE_X</td>
<td>NUM</td>
<td>4</td>
<td>* not used for Caster Scheduling</td>
</tr>
<tr>
<td>15</td>
<td>COIL_CHAR</td>
<td>CHAR</td>
<td>3</td>
<td>* not used for Caster Scheduling</td>
</tr>
<tr>
<td>16</td>
<td>COIL_WGHX1</td>
<td>NUM</td>
<td>5</td>
<td>Coil Weight (kg.) min.</td>
</tr>
<tr>
<td>17</td>
<td>COIL_WGHX2</td>
<td>NUM</td>
<td>5</td>
<td>Coil Weight (kg.) max.</td>
</tr>
<tr>
<td>18</td>
<td>PIECES_X</td>
<td>NUM</td>
<td>5</td>
<td>Number of slabs</td>
</tr>
<tr>
<td>19</td>
<td>SCHEDXTONN</td>
<td>NUM</td>
<td>5</td>
<td>Total Tonnage (metric) to schedule</td>
</tr>
<tr>
<td>20</td>
<td>ABBREV_CUST_NAME</td>
<td>CHAR</td>
<td>12</td>
<td>Customer Name</td>
</tr>
<tr>
<td>21</td>
<td>ORDERED_DATE</td>
<td>CHAR</td>
<td>3</td>
<td>Promised delivery date to customer (WWD)</td>
</tr>
<tr>
<td>22</td>
<td>ORDERED_YEAR</td>
<td>CHAR</td>
<td>4</td>
<td>Promised delivery year to customer (YYYY)</td>
</tr>
</tbody>
</table>
A4.0 Get target scheduling horizon from Scheduler;
Scheduler answers system prompt to define a due date range, within which, orders are valued as current (in HBRD format = YYWWD);

A5.0 Get any Priority Grades from scheduler
Scheduler accepts or deletes Priority Grade Code indicators currently assigned by default to ULC / HIC grades;
Scheduler assigns Priority Grade Code indicator to additional ordered grades if warranted;

A6.0 Create Working Set from Order File;
Get all priority orders first (using Priority Grade Code and Sales Order Priority Code fields);
if an order is for more than one slab, create a unique Working Set record for each slab by cloning the original order characteristics;
Get as many non priority orders as required to fill out Working Set upper limit (typically 10,000 slabs);
Loop A: for each slab in Working Set assign a delivery based value;
    if {slab due date (HBRD) < lowest date in target scheduling horizon};
    then;
\[
\text{Slab Value} = \text{Slab Value} + \text{Backlog Priority Slab Premium parameter;}
\]

else if \{slab due date (HBRD) > highest date in target scheduling horizon\};
then;
\[
\text{Slab Value} = \text{Slab Value} + \text{Forelog Priority Slab Premium parameter;}
\]
else;
\[
\text{Slab Value} = \text{Slab Value} + \text{Current Priority Slab Premium parameter;}
\]
end if;

if \{slab Priority Grade Code field is not blank\};
\[
\text{Slab Value} = \text{Slab Value} + \text{Special Grade Priority Slab Premium parameter;}
\]
end if;

if \{slab Sales Order Priority Code field is not blank\};
\[
\text{Slab Value} = \text{Slab Value} + \text{Special Customer Priority Slab Premium parameter;}
\]
end if;

End Loop A:
A7.0 Create Initial Heats

Two methods are used. Method 1 features a sequential heat construction approach (based on descending slab widths) guaranteeing that all heats contain mutually exclusive slabs. Method 2 features heat construction based on using starting slabs that have the highest value within each group of candidate slabs of the same width. Heats developed by Method 2 are guaranteed to have unique starting slab widths and are designed to cover all width groups within the Working Set.

Method 1:

Loop A: for each grade group in Working Set;

Loop B: select the first 2 available slabs, one for each strand;

call Create Heat function \{selected slab width, grade parameters\} (see section A10.0 below);

if successful, mark all slabs assigned to Heat as selected;

End Loop B:

End Loop A:

End Method 1:

Method 2:
Loop A: for each Grade within Working Set;

Loop B: for each homogeneous width group;

select the two slabs with highest Slab Value, one for each strand;

call Create Heat function {selected slab width, grade parameters} (see section A10.0 below);

End Loop B:

End Loop A:

End Method 2:

A8.0 Create Initial Sequences

Loop A: for each Initial Heat;

Starting Heat = current Initial Heat;

if grade is ULC or HIC;

call Build Sequence function {Starting Heat data record} (see section A11.0 below);

if successful add cast sequence as a column to Master Problem LP version;

End Loop A:

Initial sequences beginning with non ULC / HIC heats are less valuable (reflecting company policy) and are less likely to be part of the final solution. However, we do generate initial sequences with this kind of profile just in case there are not
sufficient ULC / HIC orders available. We use a separate loop during initial sequence building because this is one area of flexibility for reducing overall solution time. That is, we can attempt to create initial sequences with each initial non ULC / HIC heat as shown below or by using a simple sampling criteria (e.g. every second, third, etc. heat in the initial heat file). The larger the offset, fewer heats are selected for starting seeds and fewer initial non ULC / HIC sequences are created resulting in faster execution times.

Loop B: for each Initial Heat

Starting Heat = current Initial Heat;

if grade is not ULC or HIC;

   call Build Sequence function {Starting Heat data record} (see section A11.0 below);

   if successful add cast sequence as a column to

   Master Problem LP version;

End Loop B:
A9.0 Iterative Column Generation Stage

LP Solve counter = 0;

Number of Sequence = Initial Sequences;

Plateau Count = 0;

Loop CG:

    if Number of Sequence < 500;
        solve Master Problem LP version using CPLEX
        CPXbarrier callable library routine;
    else;
        solve Master Problem LP version using CPLEX
        CPXdualopt callable library routine;
    end if;

Stopping Criteria Check:

    if CPLEX Iteration Count returned = 0 then exit
Loop CG:

    if Number of Sequences > Max Sequences
parameter then exit Loop CG:

    if percentage increase of LP Objective Function
value < Min Objective Increase parameter (i.e. 1.5%);
        increment Plateau Count;
    else;
Plateau Count = 0;

end if;

if Plateau Count > Max Objective Plateau parameter (i.e. 6); exit Loop CG:

End Stopping Criteria Check:

if LPSolve counter = 0;

solve initial Master Problem IP version using CPLEX CPXmipoptimize callable library routine;

end if;

Start Heat Count = 0

Column Count = 0;

for each slab in Working Set:

Adjusted Slab Value = Slab Value - Dual Value from most recent LP solution;

for each Heat:

Adjusted Heat Value = Heat Value + Σ Adjusted Slab Value for all slabs within the Heat;

create Sorted Heat List by sorting all heats in descending order of Adjusted Heat Value;

Loop A: create additional columns while Column Count < 6;

Starting Heat = Sorted Heat List (Start Heat Count);
call Build Sequence function {Starting Heat data record} (see section A11.0 below);

if successful add cast sequence as a column to Master Problem LP version;

else;

increment Start Heat Count;

go to Loop A:

end if;

increment Column Count counter;

increment LPSoolve counter;

increment Number of Sequences counter;

increment Start Heat Count;

End Loop A:

End Loop CG:

add constraint to Master Problem LP version enforcing:

Σ columns selected ≤ Max Sequences in Solution;

solve final Master Problem LP version using CPLEX CPXdualopt callable library routine;

solve final Master Problem IP version using CPLEX CPXmipoptimize callable library routine;
Solution Improvement:

for the cast sequences (zero-one decision variables) selected by final Master Problem IP:

enumerate all the scheduling combinations (e.g. if Max Sequences in Solution = 2 and cast sequences A and B are selected, both AB and BA are possible schedules. If Max Sequences in Solution = 3, the combinations ABC, ACB, BAC, BCA, CAB, CBA are all valid schedules, etc.);

select the Final Schedule as the combination that has the lowest number of violations of the Minimum Number Of Heats Per Double Block (i.e. 12 hour) and the Minimum Number Of Heats Per Block (i.e. 6 hour) parameters;

for each slab selected within the Final Schedule:

exchange positions with any non selected slab in the Working Set that is of identical size and grade but with greater Slab Value;

End Solution Improvement:

write out Schedule Detail, Schedule Summary and Unscheduled Slab files;

for each heat within the Final Schedule:
if total strand casting time \{strand 0\} not equal total strand casting time \{strand 1\};

Scheduler shortens slab lengths on the strand with the higher total casting time and lengthens slab lengths on the strand with the lower total casting time until the two total strand casting times are equal (any adjustment to a slab length must be within the allowable range keeping the slab weight within the customer's minimum ordered weight and maximum ordered weight specifications);

end if;

End Solution Procedure;
A10.0 Create Heat Function {target width, grade parameters}

This function is used in two ways. If called during the Create Initial Heats stage, a starting slab has already been identified for each strand and the pertinent slab and grade characteristics are available. If called from the Build Sequences function, only target start or end strand widths along with a target grade are identified and the first step is to find starting slabs that satisfy those parameters.

Heat Weight = 0;
Invert = 0;
Heat Value = 0;
Heat File Number = 0;
Adjusted Heat Value = 0;
Accumulated Strand Casting Time (0) = 0;
Accumulated Strand Casting Time (1) = 0;

Loop HT:

    if Accumulated Strand Casting Time (strand 0) > Accumulated Strand Casting Time (strand 1);
        Current Strand = strand 1;
    else;
        Current Strand = strand 0;
    end if;
if Accumulated Strand Casting Time (Current Strand) = 0, this is the first slab on Current Strand:

then;

*When the Create Heat function is called from Create Initial Heats, a specific slab with all defining characteristics has been sent:*

Current Grade = Slab Grade;

go to Accrue;

*When the Create Heat function is called from the Build Sequence function, we check if the heat is required for the head (i.e. to precede the heat currently first) or the tail end (i.e. to follow the heat currently in the last position) of the sequence.*

*NOTE:* we build a heat the same way in either case but we simply invert the heat so that it will fit at the start of the sequence when required.

if \{target widths are First in Sequence Start Widths\} Invert = 1;

find a first slab that fits target parameters:

starting with slabs within the target grade then exploring remaining grades in ascending order of grade change penalty (i.e. like, LGTR, mixed and unlike grade change categories in succession) until at least one previously unselected candidate slab is found that matches the target width criterion (i.e. target strand widths - 100 mm. \(\leq\) candidate slab width \(\leq\) target strand widths + 100 mm.):\n
Current Grade = Slab Grade;
select the candidate slab within Current Grade with largest Adjusted Slab value;

retrieve pertinent slab characteristics from Working Set record (i.e. Slab Grade, Slab Casting Time, Slab Weight, etc.)

go to Accrue:

else this is not a first slab;

for every unselected slab in the Current Grade:

calculate Ratio = Adjusted Slab Value / absolute value of (Last Strand Width - Slab Width);

select the candidate slab with the largest Ratio;

Accrue:

Accumulated Strand Casting Time (Current Strand) = Accumulated Strand Casting Time (Current Strand) + Slab Casting Time;

Heat Weight = Heat Weight + Slab Weight;

Last Strand Width = Slab Width;

Heat Value = Heat Value + Slab Value;

Adjusted Heat Value = Adjusted Heat Value + Adjusted Slab Value;

Heat Casting Time = Max {Accumulated Strand Casting Time (0), Accumulated Strand Casting Time (1)};

end if;
Check heat acceptance criteria:

if Current Grade is Non-ULC / HIC:

if (Heat Weight ≥ 320 and Heat Weight ≤ 349)

exit Loop HT;

else;

go to Loop HT;

end if;

else;

if (Heat Weight ≥ 290 and Heat Weight ≤ 305)

exit Loop HT;

else;

go to Loop HT;

end if;

end if;

End Loop HT:

We have a legitimate new heat. Even if Adjusted Heat Value is not > 0, we keep the heat for potential future use since it will lose the current ratio comparison in the calling procedure anyway.

if Invert = 1 then sort candidate heat in descending order of current slab position;

append Heat to indexed heat file;

return to calling procedure;
A11.0 Build Sequence Function {Starting Heat data record}

This function is used in two different steps of the solution procedure, during the Create Initial Sequences stage, and repeatedly during the Column Generation procedure. In either case, a Starting Heat has been identified along with its important detail characteristics for cast sequence building: grade; start widths on strand 0 and strand 1; end widths on strand 0 and strand 1 etc.. Both the start and end width of each strand are important because a sequence can be extended by either appending a heat after the heat currently last in the sequence (i.e. at the tail) or by preceding the heat currently in the first position in the sequence (i.e. at the head).

Thus, from a single Starting Heat, a cast sequence is iteratively developed by adding the best of the final four candidate heats, after considering the Adjusted Heat Value / Grade Change Penalty ratio for each (ties are broken at random). The final four candidates are products of exhaustive selection processes aimed at finding the heat with the maximum Adjusted Heat Value / Grade Change Penalty ratio under the following four sequence building conditions (any ties are again broken at random): (1) selecting from all the existing heats that can feasibly follow the last heat in the partial sequence; (2) selecting from the all existing heats that can feasibly precede the first heat in the partial sequence; (3) dynamically creating a heat that can feasibly follow the last heat in the partial sequence; (4) dynamically
creating a heat that can feasibly precede the first heat in the partial sequence. After each heat addition decision is made, critical steel consumption rate and grade change restrictions are checked along with the sequence length and overall sequence value related stopping criteria.

First in Sequence Start Width (strand 0) = Starting Heat Start Width (strand 0);
First in Sequence Start Width (strand 1) = Starting Heat Start Width (strand 1);
First in Sequence Grade = Starting Heat Grade;
Current Partial Sequence consists of only 1 heat, the Starting Heat, therefore the First in Sequence and Last in Sequence information initially relates to the same heat.

Last in Sequence End Width (strand 0) = Starting Heat End Width (strand 0);
Last in Sequence End Width (strand 1) = Starting Heat End Width (strand 1);
Last in Sequence Grade = Starting Heat Grade;

Number of Heats in Seq. = 1;
Number of Plates = 0;
Number of Mixed = 0;
Total Casting Time = Starting Heat Casting Time;
Accumulated Reduced Cost = Adjusted Heat Value of Starting Heat;
Sequence Value = Heat Value of Starting Heat;
if Starting Heat grade is ULC, then ULC = 'true';

if Starting Heat grade is HIC, then HIC = 'true';

Loop CS:

Avg. Steel Consumption Rate = Total Casting Time / Number of Heats in Seq.;

**Condition 1:** find best existing heat to follow last heat in sequence, restrict search to feasible heats (i.e. with Start Width (strand 0) within ± 100 mm. of Last in Sequence End Width (strand 0) and with Start Width (strand 1) within ± 100 mm. of Last in Sequence End Width (strand 1) and (if Number of Heats in Seq. ≤ 4 and either ULC or HIC = 'true', only consider heats of ULC or HIC grades respectively, otherwise exclude all heats of these special grades from candidate heat list)).

maxval (1) = 0;

maxratio = 0;

candidate heat (1, maxratio) = 0;

Loop A:

for each Candidate Heat complying with Condition 1 criteria;

    if \{Avg. Steel Consumption Rate ≥ Upper Heat Time Aim parameter\}:
if {casting time of Candidate Heat < Avg. Steel Consumption Rate};

    increase heat value of Candidate Heat by Hot Metal Bias Factor parameter;

end if;
end if;

if {Avg. Steel Consumption Rate < Lower Heat Time Aim parameter};

    if {casting time of Candidate Heat > Avg. Steel Consumption Rate};

        increase heat value of Candidate Heat by Hot Metal Bias Factor parameter;

    end if;
end if;
end if;

Candidate Heat id = Heat File index number for current heat;

grade change penalty = Grade Penalty File table entry at row {Last in Sequence Grade} and column {Candidate Heat grade};

ratio = Adjusted Heat Value of Candidate Heat / grade change penalty;

if {ratio = maxval(1)};

The current candidate heat ratio has tied the previous best ratio, add to the list for later random tie breaking if necessary;
candidate heat (1, maxratio) = Candidate Heat id;

increment maxratio;

end if;

if \{ratio > maxval(1)\};

The current candidate heat is the new highest ratio;

maxratio = 0;

candidate heat (1, maxratio) = Candidate Heat id;

maxval(1) = ratio;

end if;

End Loop A;

if (maxratio > 0);

candidate heat (1,0) = one of candidate heat (1, maxratio) chosen at random;

end if;

Condition 2: find best existing heat to precede the first heat in sequence, restricting search to feasible heats (i.e. with End Width (strand 0) within ± 100 mm. of First in Sequence Start Width (strand 0) and with End Width (strand 1) within ± 100 mm. of First in Sequence Start Width (strand 1) and (if Number of Heats in Seq. ≤ 4 and either ULC or HIC = 'true', only consider heats of ULC or HIC grades respectively, otherwise exclude all heats of these special grades from candidate heat list)).

maxval (2) = 0;
maxratio = 0;
candidate heat (2, maxratio) = 0;

Loop B:

for each Candidate Heat complying with Condition 2 criteria;

    if {Avg. Steel Consumption Rate ≥ Upper Heat Time Aim parameter};

        if {casting time of Candidate Heat < Avg. Steel Consumption Rate};

            increase heat value of Candidate Heat by Hot Metal Bias Factor parameter;

        end if;

    end if;

if {Avg. Steel Consumption Rate ≤ Lower Heat Time Aim parameter};

    if {casting time of Candidate Heat > Avg. Steel Consumption Rate};

        increase heat value of Candidate Heat by Hot Metal Bias Factor parameter;

    end if;

end if;

Candidate Heat id = Heat File index number for current heat;
grade change penalty = Grade Penalty File table entry at row
{Candidate Heat grade} and column {First in Sequence Grade};

ratio = Adjusted Heat Value of Candidate Heat / grade change penalty;

if {ratio = maxval(2)};

*The current candidate heat ratio has tied the previous best ratio, add to the list for later random tie breaking if necessary;*

candidate heat (2, maxratio) = Candidate Heat id;

increment maxratio;

end if;

if {ratio > maxval(2)};

*The current candidate heat is the new highest ratio;*

maxratio = 0;

candidate heat (2, maxratio) = Candidate Heat id;

maxval(2) = ratio;

end if;

End Loop B;

if (maxratio > 0);

candidate heat (2,0) = one of candidate heat (2, maxratio) chosen at random;

end if;
Condition 3: create the highest value heat to follow the last heat in the sequence.

The Create Heat function is called and returns the heat file index of the newly created heat.

call Create Heat function {Last in Sequence End Width (strand 0), Last in Sequence End Width (strand 1), Last in Sequence Grade};

Candidate Heat id = Heat File index number for current heat;
grade change penalty = Grade Penalty File table entry at row {Last in Sequence Grade} and column {Candidate Heat grade};
ratio = Adjusted Heat Value of Candidate Heat / grade change penalty;
candidate heat (3,0) = Candidate Heat id;
maxval (3) = ratio;

Condition 4: create the highest value heat to precede the first heat in the sequence.

The Create Heat function is called and returns the heat file index of the newly created heat.

call Create Heat function {First in Sequence Start Width (strand 0), First in Sequence Start Width (strand 1), First in Sequence Grade};

Candidate Heat id = Heat File index number for current heat;
grade change penalty = Grade Penalty File table entry at row {Candidate Heat grade} and column {First in Sequence Grade};
ratio = Adjusted Heat Value of Candidate Heat / grade change penalty;

candidate heat (4,0) = Candidate Heat id;

maxval (4) = ratio;

Select the best of the final four candidate heats

count = 1;

best = 0;

max = 0;

Loop C:

if {maxval(count) > max};

max = maxval(count);

best = count;

end if;

Check additional feasibility criteria

reject = 'false';

if {grade change penalty = Current Illegal Grade Penalty parameter};

reject = 'true';

end if;

if {Avg. Steel Consumption Rate ≥ Upper Heat Time Aim parameter and casting time of Candidate Heat ≥ Avg. Steel Consumption Rate};

reject = 'true';
end if;

if \{Avg. Steel Consumption Rate \leq Lower Heat Time Aim parameter and
casting time of Candidate Heat \leq Avg. Steel Consumption Rate\};

    reject = 'true';

end if;

end if;

if \{grade change cost = Current Unlike Grade Penalty\};

    Number of Plates = Number of Plates + 1;

    if \{Number of Plates > Max Plates Per Tundish parameter\};

        reject = 'true';

    end if;

end if;

end if;

if \{grade change cost = Current Mixed Grade Penalty\};

    Number of Mixed = Number of Mixed + 1;

    if \{Number of Mixed > Max Mixed Per Tundish parameter\};

        reject = 'true';

    end if;

end if;

end if;

if \{reject = 'true'\}

    mark the slabs in the candidate heat \text{(best,0)} as selected since we
don't want to try this heat again;

    go to End Loop CS:

end if;
Number of Heats in Seq. = Number of Heats in Seq. + 1;

Accumulated Reduced Cost = Accumulated Reduced Cost + Adjusted Heat Value of selected heat - grade change penalty;

Sequence Value = Sequence Value + Heat Value for selected heat - grade change penalty;

End Loop CS:

if {Accumulated Reduced Cost ≤ 0 or Sequence Value < 0 or Number of Heats in Seq. < Min Heats per Sequence};

    return to calling procedure with status flag = unsuccessful;

end if;

return new cast sequence to calling procedure with status flag = successful;
A1.1 Initialization Parameter Detail

Name of comma-delimited file of output grade costs.

Grade Penalties File Name = GRDCOST.CSV

Maximum initial columns to generate. If negative, generate as many as possible.

Max Initial Sequences = -1

Maximum number of columns allowed in cast sequence generation program.

Max Sequences = 2000

Objective function must improve by at least this percentage before CS_PLATEAU_MAX is reached.

Min Objective Increase = 0.015

Max. # of times obj. fcn. can improve by less than OBJ_THRESHOLD.

Max Objective Plateau = 6

Max. / Min. / Current penalty for grade changes to be considered as 'LGTR'

Max LGTR Grade Penalty = 49
Min LGTR Grade Penalty = 5
Current LGTR Grade Penalty = 8

Max. / Min. / Current penalty for grade changes to be considered as ‘Mixed’
Max Mixed Grade Penalty = 3499
Min Mixed Grade Penalty = 50
Current Mixed Grade Penalty = 1500

Max. / Min. / Current penalty for grade changes to be considered as 'unlike' or 'severe' requiring 'plate' insertion
Max Unlike Grade Penalty = 24999
Min Unlike Grade Penalty = 3500
Current Unlike Grade Penalty = 4000

Max. / Min. / Current penalty for grade changes to be considered as 'illegal'
Max Illegal Grade Penalty = 99999
Min Illegal Grade Penalty = 25000
Current Illegal Grade Penalty = 49999

Current penalty for grade changes considered to be 'like' = 1
Current penalty for grade changes between exactly 'equal' grades = 0.9

Lower / Upper aim for the average heat time (min.)

Lower Heat Time Aim = 45
Upper Heat Time Aim = 55

If 'Lower / Upper Heat Time Aim' is exceeded, the amount that the heat generation routine will favour narrower or wider heats respectively.

Hot Metal Bias Factor = 400.0

Max. # of integer solves before final one.

Max Intermediate IP Solutions = 1

Max. number of additional columns to add per column generation iteration.

Max Additional Columns = 6

Total cols. generated will be the lesser of

('Max Total Columns Factor'*num_original_cols) and ('Max Sequences')

Max Total Columns Factor = 100

Maximum number of heats that can be generated

Max Heats Generated = 65000
Maximum number of heats in a sequence before a tundish change is required.

Max Heats Per Tundish = 10

Max. number of plates allowed on a tundish.

Max Plates Per Tundish = 1

Max. number of mixed grade changes allowed on a tundish.

(Number of plates + number of mixed) must be \( \leq \) this number.

Max Mixed Per Tundish = 3

Max. number of slabs to use in model.

Max Unscheduled Pieces = 10000

Max. number of slabs allowed in a heat.

Max Slabs Per Heat = 40

Name of the last allowable ULC grade.

Max ULC Grade = CC009ZZ

Min. number of times to Master Problem (LP) before starting to check objective function plateau stopping criteria.
Min LP Solutions = 34

Min. / Max. number of heats in a given cast sequence.

Min Heats Per Sequence = 20
Max Heats Per Sequence = 20

Max. allowable strand width for an unlike grade change.

Max Plate Insertion Width = 1375

Penalty if a plate or FTC is done when width > 'Max Plate Width',

Unsuccessful Plate Penalty = 2000

Slope an Intercept of sequence length penalty function.

Sequence Length Penalty Function Slope = 2500
Sequence Length Penalty Function Intercept = 0.00

Heat size limits in tons.

Min Non-ULC / HIC Heat Weight = 320
Max Non-ULC / HIC Heat Weight = 349
Min ULC Heat Weight = 290
Max ULC Heat Weight = 305
Maximum number of ULC heats allowed at start of sequence.

Max ULC Heats Per Sequence = 5

Number of ULC heats allowed at start of sequence, if the maximum width falls between ULC_WIDTH_LB and ULC_WIDTH_UB.

Mid ULC Heats Per Sequence = 4

Minimum number of ULC heats allowed at start of sequence.

Min ULC Heats Per Sequence = 3

Maximum number of very-low carbon heats per sequence.

Max VLC Heats = 3

Maximum number of HIC heats per sequence

Max HIC Heats = 3

Minimum strand width difference allowed in a heat (for split strand).

Min Strand Width Difference = 0

Delivery value weights

Backlog Priority Slab Premium = 150
Current Priority Slab Premium = 100

Future Priority Slab Premium = 1

Special Grade Priority Slab Premium = 300

Special Customer Priority Slab Premium = 100

Lower / Upper boundary of width for # of ULC heats at start of seq. in mm.

ULC Slab Width Lower Boundary (ULC_WIDTH_LB) = 1100

ULC Slab Width Upper Boundary (ULC_WIDTH_UB) = 1300

Time of heat groupings (i.e. blocks) broken out in summary report, in minutes.

Block Time = 360

Minimum number of heats desired in 2 consecutive 'Block Time' periods.

Minimum Number Of Heats Per Double Block = 13

Minimum number of heats desired in a 'Block Time' period.

Minimum Number Of Heats Per Block = 6

Delay time factors in minutes per occurrence

Time Required For A Flying Tundish Change = 10
Time Required For An Unlike (Plate) Grade Change = 10
Extra Time Required For Starting An HIC Sequence = 20
Time Required For Starting A New Sequence = 60
Extra Time Required For Starting A ULC Sequence = 10

If 'YES', then HBRD's will be used if 'NO', then ordered dates will be used.

Use HBRD = YES

Maximum number of sequences in the final solution.

Max Sequences in Solution = 2

Maximum width change between 2 consecutive pieces within a heat, in mm.

Max Width Change = 100

Density of slab in grams / cubic centimeter

Slab Density = 7.65

Thickness of slab in cm.

Slab Thickness = 21.6
## APPENDIX II

### Unscheduled Order File Instances Saved

| 1995 Unscheduled Order File Data (identified by month and day - mmdd) |
|-----------------|----------------|----------------|
| 0102            | 0404           | 0808           |
| 0117            | 0408           | 0812           |
| 0120            | 0411           | 0815           |
| 0124            | 0415           | 0819           |
| 0127            | 0418           | 0928           |
| 0210            | 0422           | 1005           |
| 0214            | 0427           | 1012           |
| 0217            | 0623           | 1019           |
| 0221            | 0627           | 1026           |
| 0224            | 0701           | 1102           |
| 0302            | 0706           | 1109           |
| 0309            | 0725           | 1116           |
| 0325            | 0729           | 1123           |
| 0328            | 0801           | 1130           |
| 0401            | 0805           | 1207           |