1996

A bi-level local search algorithm for the melt scheduling problem in steel processing

Bhavin J. Doshi
Lehigh University

Doshi, Bhavin J.

A Bi-Level Local Search Algorithm for the Melt Scheduling Problem in Steel Processing

June 2, 1996

A Bi-Level Local Search Algorithm

for the Melt Scheduling Problem in Steel Processing

by

Bhavin J. Doshi

A Thesis

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial and Manufacturing Systems Engineering

Lehigh University

May 1996

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

5/3/96

Date

Thesis Advisor

Chairperson of Department

Acknowledgments

I would like to thank my thesis advisor, Dr. S. David Wu, for his guidance and constant support during the completion of this thesis and also throughout the entire duration of the project at BethForge. I would also like to thank Dr. Robert H. Storer for his helpful insights and enthusiasm during the course of this project. Thanks are also due to Mike Medei from the BethForge corporation for his patience and for the countless hours he spent working with us. I would like to extend a special note of thanks to my office mates at the Manufacturing Logistics Institute - Nabeela Al-Refai, Karel Zeithammer, Kedar Naphade, Erhan Kutanoglu, Jaime Bustos and Jayan Moorkanat. This thesis would not have been possible without their help and support.

Table of Contents

List of Tables

List of Figures

ABSTRACT

BethForge, a division of the Bethlehem Steel Corporation, is a manufacturer of custom-made heavy forgings. An increasingly competitive market and need for more responsive customer service have made planning and scheduling a top priority at BethForge. The ingot formation process at BethForge is a heavily constrained engineering problem which involves melting and pouring molten steel of specific chemical composition to a variety of ingot molds. Due to the capital intensive nature of the process and the complex technological and resource constraints in the melting/pouring process, this "melt scheduling" problem demands a considerable effort on part of the planner. This thesis examines in detail both - the melt scheduling problem and a bi-level local search algorithm designed to solve it. The melt scheduling problem is defined and a mixed integer non-linear programming formulation is developed for it. Next, a bi-level local search algorithm designed to solve this problem is described. Results of testing the algorithm on actual order data from BethForge are provided. The results show the tradeoff involved in trying to optimize the two conflicting objectives of waste minimization and tardiness reduction. They also indicate the flexibility of the algorithm as a scheduling tool that can used at BethForge.

Chapter 1

Introduction

1.1 Description of the BethForge Manufacturing Environment

The BethForge division of the Bethlehem Steel Corporation is a leading manufacturer of custom-made heavy steel forgings. Its products fall into five basic categories (1) hardened steel rolls (HSRs), (2) large diameter rolls (LDRs), (3) electrical power generation equipment (EPGs), (4) marine application equipment and (5) custom open die forgings. HSRs and LDRs are used in steel and aluminum rolling mills. EPGs include products such as large shafts and rotors. Marine application equipment includes products such as ship shafts. Most of the custom-made open die forgings are nuclear power plant components.

HSRs, LDRs, rotors and shafts follow similar processing since the major machining step for these products is the turning operation. These turned parts constitute a major portion of BethForge's production. The nuclear power plant products are discs or shells.

BethForge's customers include defense equipment manufacturers, the steel industry, the power generation industry and ship builders. BethForge is the largest U.S. manufacturer of heavy custom-made forgings. Its main competitors are companies in Europe and the Pacific rim. Under their current operating conditions, BethForge faces a number of problems in meeting their customers' requirements. Due to a long production lead time and an ever changing market, meeting desired delivery date while cutting back

on operating costs become increasingly difficult. These problems are often handled by costly solutions such as excess in-process inventory, over time and excess capacity. This has prompted management at BethForge to examine their planning and scheduling functions as well as their information system, hoping to strengthen its leadership in the competitive market.

1.2 Layout of Production Facilities at BethForge

There are six basic steps in the BethForge production process. These six steps involve processing at four facilities - melting, forging, heat treatment and machining. Production of a forging starts by formation of an ingot at the Steelton facility near Harrisburg, PA. Ingot formation involves two steps. First the right ingredients (steel and additives such as alloying elements) are melted in a furnace to give the desired grade of steel. Next, the molten metal is poured into molds to form the ingot of the required size and weight. The ingots are then transported from Steelton to Bethlehem via railroad using specially designed rail cars. Ingots are then forged at the press forge. The forged ingots then go to the treatment shop where they undergo a cycle of preliminary heat treatment. The so-called "hot-end" consists of all the shops up to and including the treatment shop. Ingots must be kept hot at all times while they are in the hot-end. From the treatment shop, the ingots visit one of two machining shops where they undergo rough machining. Unlike the hot-end, the products follow different paths through the machine shops. HSRs are machined at machine shop #1 and all other products are machined at machine shop #2.

Some of the products from machine shop #2 revisit the treatment shop for a final round of heat treatment, followed by further machining to their final dimensions. HSRs that need heat treatment after rough machining visit a separate heat treatment facility in machine shop #1. Thus, the flow of materials in BethForge is like a flow shop up to the hot-end. A reentrant flow situation exists at the end where ingots revisit the treatment shop from machine shop #2. Figure 1 shows the flow of materials through different sections in BethForge.

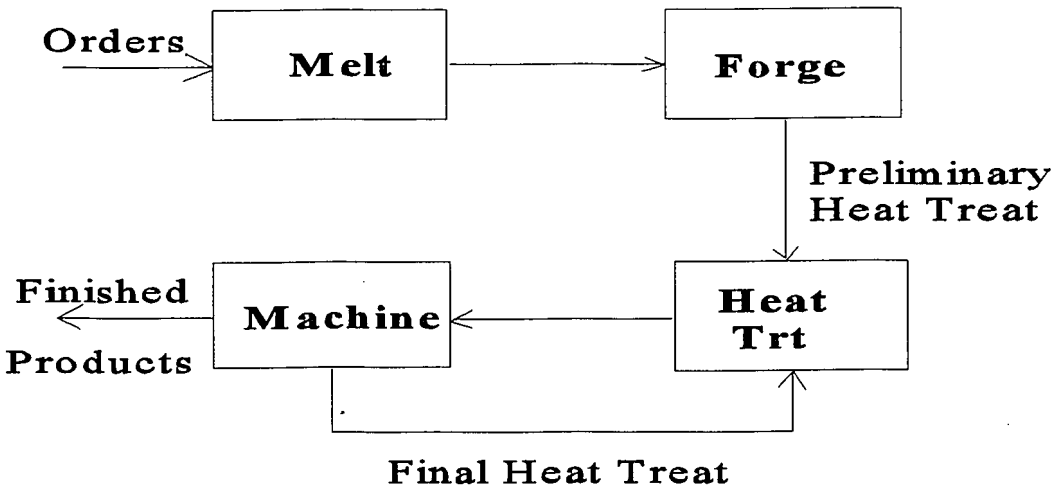


Figure 1. Layout of BethForge Production Facilities

Each of the four major manufacturing steps - melting, forging, heat treatment and machining- has a significant number of technological constraints which make scheduling a difficult and important issue at BethForge. The focus of this thesis will be the scheduling issues involved at the melt shop.

1.3 Thesis Outline

The thesis is organized as follows. Chapter 2 provides a detailed description of the Melt Shop. The melt scheduling problem is defined and a mixed integer non-linear programming formulation is provided. Chapter 3 discusses the bi-level local search heuristic developed to solve the melt scheduling problem. The results of using the algorithm on actual order data from BethForge are provided in Chapter 4. Chapter 5 outlines possible improvements and enhancements to the algorithm and offers concluding remarks.

Chapter 2

The Melt Scheduling Problem

2.1 Melt Shop Resources and Constraints

Ingot formation is the first step in the production of a forging. This process is carried out at the melt shop. As shown in Figure 2, the melt shop consists of the following resources: one 150 ton D.C. arc furnace, one 150 ton ladle refining furnace, one 150 ton tank degasser and two 290 ton vacuum stream degassing tanks. Production capacity at the melt shop is measured in terms of "heats". A heat is said to be formed when a certain

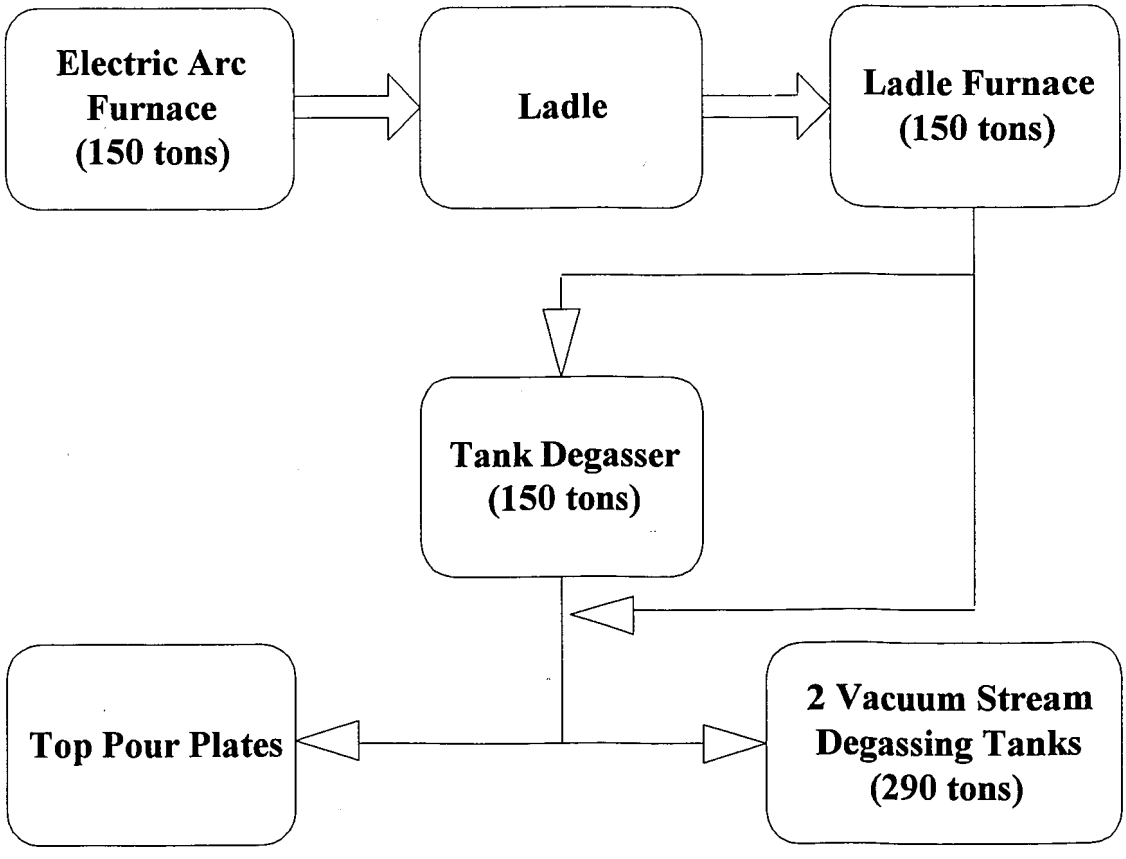


Figure 2. Overview of the Ingot Formation Process

quantity of steel is earmarked to be melted for pouring one or more ingots of a particular chemical composition called a "grade". Steel is first melted in the D.C arc furnace. It is then transferred to the ladle furnace using a ladle. The ladle furnace is used to refine the metal. Alloying elements are added to it and it undergoes desulphurization. Molten metal from the ladle furnace is either transferred to the tank degasser for further chemical control or is poured to form ingots. The tank degasser is used to remove unwanted elements such as hydrogen, sulphur, aluminum and oxygen and to thus impart better chemical properties. The desired chemical properties of a forging determine whether the ingot for that forging visits the tank degasser. Under current operating conditions, ingots are melted once a week with a capacity of up to 6 heats. There are several restrictions on how a heat can be formed. These are described below.

2.1.1 Heat Formation Constraints

(1) All ingots that are to be poured from a heat must be of the same grade since exactly one grade can be melted in a heat.

(2) A minimum of 125 tons of steel must be melted in each heat. This lower limit on the tonnage that can be melted in a heat is a result of the design of the Electric Arc Furnace (EAF). In this furnace, a certain minimum quantity of metal is required for the electrodes to make contact and strike an arc. This lower limit imposes a severe restriction on the flexibility in scheduling the melt shop. If the ingots of a certain grade have a combined weight of less than 125 tons, say 110 tons, then an additional 15 tons of steel

must be melted to meet the lower weight limit. These 15 tons constitute waste since no ingots are made from it.

(3) A maximum of 145 tons of steel may be melted in each heat. This is a result of the capacity of the arc furnace.

It is also possible to pour molten metal from two successive heats to form an ingot but in this case, additional restrictions apply. Ingots weighing more than 145 tons must be poured by combining two heats. In order to pour an ingot this way, the metal from the first heat is superheated and is transferred to a ladle where it begins to cool down. Meanwhile, the second heat is prepared and by timing the operations correctly, the molten metal from both heats become available at the correct temperature and at the same time. Since only one ladle is available to store the molten metal, at most two successive heats can be combined to pour an ingot. If the heats are numbered successively from 1 to 6 and if two of them - say heats 2 and 3 - are used to pour an ingot, then heats 3 and 4 cannot be used to pour an ingot. In other words, if combined heats are used to pour an ingot, then any two such combinations must be disjoint. This may be called the Ladle Constraint.

Once the molten metal is ready in the ladle furnace or the tank degasser, it must be poured into a mold to form an ingot. An ingot can either be bottom-poured or top-poured. Pouring constraints are described next.

2.1.2 Pouring Constraints

Bottom-pouring utilizes two resources, viz., plates and stools. A plate consists of

a system of runners and down fountains through which the molten metal flows. A runner connects a mold to a down fountain. A stool is used at the base of each plate. Molten metal is poured into a reservoir attached to a down fountain from where it flows into runners and rises up into the molds. One or more molds can be assembled on a plate, depending on the capacity of the plate used. Each plate also has a limit on the total weight of ingots that can be supported on it. Some plates have a facility to "block" the runners. Blocking of runners allows ingots of different grades and different heights to be poured on the same plate by using a separate down fountain for each ingot. Plates that do not have this blocking facility must have all the ingots of the same grade and height poured on them. The common height requirement arises from the fact that the molten metal must rise to the same level in each mold since all molds are interconnected by runners. Interconnection of the molds by runners also requires all the ingots to be of the same grade. The number of plates, molds, stools and down fountains available each week is limited. However, only the plates are considered a critical resource. Sufficient inventory of molds, stools and down fountains exists at the present time in order to fully utilize all available plates.

Top pouring is done using the vacuum stream degassing tanks. Each of the two tanks can be used for pouring one ingot. Top pouring also requires molds and stools.

Molten metal from a heat must be poured into molds of the correct shape and size to form the ingot. Ingot size is determined by its diameter. There are restrictions on the diameter ranges of the ingots that can be accommodated on different plates or tanks. Only certain combinations of ingots can be poured on the same plate. The ingots produced at

Ingot Diameter	Possible Resources
40"-48"	3
54"-69"	1,2B
69"-78"	1,2A,2B
78"-92"	1,2A
>92"	1

Table 1 Ingot-Plate Compatibility

Resource Type	Diameter	No.of Plates/Tanks	No. of Ingots per Plate/Tank
1	>54"	2	1
2A	69"-92"	2	2
2B	54"-78"	2	2
3	40"-48"	4	6

Table 2. Ingot Pouring Resources

the melt shop can be classified into five diameter ranges. Table 1 shows the compatibility of the pouring resources with each of these ranges. Ingots larger than 92" in diameter are always top poured. In addition, any ingot with diameter 54" or more can also be top poured. Table 2 summarizes the data on ingot pouring resources. Resource type 1 refers to the vacuum stream degassing tanks used for top pouring. Types 2A, 2B and 3 refer to the bottom pour plates. The second column indicates the diameter ranges that can be accommodated on each resource type. The third column indicates the number of available

tanks or plates of each type. The last column contains the capacity of each resource in terms of the number of ingots that can be accommodated on each plate or tank.

2.1.3 Zero WIP Constraint

In general, any metal that is melted in a heat must be poured before the next heat is melted. Molten metal cannot be stored. The only exception to this rule is when two heats are being combined to pour an ingot. In this case, the first heat can be stored in the ladle while the second heat is being melted and then metal from both heats must be poured. This constraint thus imposes a zero work-in-process (WIP) restriction between the melt and pour stages of the ingot formation process.

The melt shop is thus a very heavily constrained environment for scheduling. With a weekly capacity of six heats, a maximum of 900 tons can be melted each week. However, any schedule developed for the melt shop must be feasible in terms of both the melting and pouring constraints. Compatibility of grades within a heat and the lower weight limit on each heat impose severe restrictions on the ability to utilize this capacity. Finding a feasible melt schedule that makes effective use of available capacity is the first challenge for any scheduler.

2.2 Melt Shop Performance Measures

Based on the due date for the order, a "melt-by" date is set for each ingot by the planners based on estimates of processing times through the various stages of the

manufacturing process. The melt-by date serves as the due date for the melt shop. In order to meet customer delivery dates, it is important for each stage in the manufacturing process to have good due-date performance. To this end, minimization of tardiness becomes an important objective in scheduling the melt shop. High cost of raw materials makes waste minimization another important objective. BethForge often gets orders for products that need unique grades, not common to those of any other product. In order to make these odd grade ingots, it is often necessary to scrap a considerable amount of steel if the ingot weight is less than the lower limit for heat formation. Even in cases when the grade of an ingot is not uncommon, there may not be a sufficient number of orders of the same grade to make a heat without wastage. The ingot then has to wait till a new order for the same grade is received. However, excessive delay is clearly undesirable and often an ingot must be melted even with considerable waste in order to meet delivery requirements. Thus there is a tradeoff between the two conflicting objectives.

2.3 Problem Statement and Model Formulation

This section defines the melt scheduling problem and describes a non-linear integer programming formulation for it. This is followed by a discussion of the constraints modeled in the formulation.

A customer order contains the following information that is relevant to the melt scheduling problem formulation: ingot weight, ingot diameter, grade and due date. In the model formulation, the vacuum stream degassing tanks are considered as "plates" for

simplicity. Bottom pour plates are numbered 1 through 8. Plate numbers 9 and 10 indicate top pour resources. The following terms are used in the problem formulation:

Data:

N total number of ingots to be melted and poured

Wt_k weight of ingot k

G_k grade of ingot k

D_k diameter of ingot k

dd_k due date of ingot k

P total number of plates

M two dimensional matrix of compatibility between ingot diameter and plate; element M(d,p) is 1 if diameter d can be accommodated on plate p, 0 otherwise.

C one dimensional array of plate capacities in terms of number of ingots that can be poured on each plate; C(p) is the number of ingots that can be poured on plate p

T total number of weeks in the planning horizon

H total number of heats available each week

MAXWT upper limit on the weight that can be melted in each heat; equals 145 tons

MINWT lower limit on the weight that can be melted in each heat; equals 125 tons

POURWT maximum weight that a bottom pour plate can hold; equals 145 tons

Decision Variables:

X_{kht} fraction of ingot k melted in heat h of week t

IX_{kht} a zero-one variable that indicates whether metal from heat h of week t is used to pour ingot k

Y_{kt} a zero-one variable that indicates whether ingot k is poured in week t

Z_{kpt} a zero-one variable that indicates whether ingot k is poured on plate p in week t

f_k finish time of ingot k; it is the week in which ingot k is melted

$waste_{ht}$ amount of waste in heat h of week t

2.3.1 Problem Statement

The melt scheduling problem consists in finding a schedule to melt and pour all ingots from the available order pool subject to the restrictions imposed on the melting and pouring of ingots. The objective is to minimize a weighted sum of total waste and tardiness. The output is an assignment of ingots to weeks and within each week to specific heats and plates subject to all the constraints.

2.3.2 Model Formulation

$$\min \alpha * \sum_{t=1}^{T}\sum_{h=1}^{H} waste(h,t) + \beta * \sum_{k=1}^{N} (f_k - dd_k)^+ \qquad (1)$$

subject to:

$$IX_{kht} = \begin{cases} 0 \text{ if } X_{kht} = 0 \\ 1 \text{ if } X_{kht} > 0 \end{cases} \quad \forall\ k,h,t \qquad (2)$$

$$f_k \geq t * LX_{kht} \quad \forall\ k,h,t \tag{3}$$

$$Y_{kt} \in \{0,1\} \quad \forall\ k,t \tag{4}$$

$$\Sigma_{h=1}^{H} X_{kht} = Y_{kt} \quad \forall\ k,t \tag{5}$$

$$\Sigma_{t=1}^{T} Y_{kt} = 1 \quad \forall\ k \tag{6}$$

$$Z_{kpt} \in \{0,1\} \quad \forall\ k,p,t \tag{7}$$

$$\Sigma_{p=1}^{P} Z_{kpt} = Y_{kt} \quad \forall\ k,t \tag{8}$$

Melt Constraints:

$$\Sigma_{h=1}^{H} LX_{kht} \leq 2 \quad \forall\ k,t \tag{9}$$

$$\left.\begin{array}{l} LX_{kht} + LX_{k,h+2,t} \leq 1 \\ LX_{kht} + LX_{k,h+3,t} \leq 1 \\ \quad . \quad . \quad . \quad . \quad . \quad . \quad . \\ LX_{kht} + LX_{k,H,t} \leq 1 \end{array}\right\} \forall\ k,t \quad \Big\} \text{for } h=1,2,\ldots,H-2 \tag{10}$$

$$\Sigma_{k=1}^{N} Wt_k * X_{kht} \leq MAXWT \quad \forall\ h,t \tag{11}$$

$$waste_{ht} = \max\{\ 0\ ,\ MINWT - \Sigma_{k=1}^{N} Wt_k * X_{kht}\ \} \tag{12}$$

$$IX_{kht} + IX_{mht} \leq 1 \quad \forall\ k,m,h,t \text{ where } G_k \neq G_m \tag{13}$$

Pour Constraints:

$$Z_{kpt} + Z_{mpt} \leq 1 \quad \text{for } p=1,2,3,4\ ;\ \forall\ k,m,t \text{ where } G_k \neq G_m \tag{14}$$

$$\Sigma_{k=1}^{N}\ Wt_k * Z_{kpt} \leq POURWT \quad \forall \text{ for } p=1,2,...,8\ ;\ \forall\ t \tag{15}$$

$$\Sigma_{k=1}^{N}\ Z_{kpt} \leq C(p) \quad \forall\ p,t \tag{16}$$

$$Z_{kpt} \leq M(D_{kp}p) \quad \forall\ k,p,t \tag{17}$$

The objective function (1) is a weighted sum of two terms. The first term is the total waste accumulated in all the heats across the entire planning horizon. The second term is the total tardiness of all the ingots. Constraint (2) defines the integer variables IX_{kht} in terms of the variables X_{kht}. (3) defines the finish time for each ingot. (4) and (5) together ensure that an ingot must be melted completely during one week, i.e., we cannot melt part of an ingot in one week and the remaining in another. Equation (5) is the link between the melt and pour variables. (6) ensures that an ingot is poured exactly once in the planning horizon. (7) and (8) imply that if an ingot is poured in some week, it must go on exactly one plate. (9) through (13) model the melt constraints: (9) imposes the limit of a maximum of two heats to melt an ingot in any week. The set of inequalities (10) enforce the ladle

constraints for the case of an ingot that is poured from two heats. If two heats are used to pour a single ingot, (10) forces the two heats to be consecutive. In addition, constraint (10) ensures that such pairs of heats are disjoint. Constraint (11) imposes the upper weight limit on each heat. (12) defines the waste for each heat. (13) forces different grades to be poured in different heats. The pour constraints are modeled in (14) through (17). (14) applies to the plates numbered 1 through 4. These are the plates that cannot have their runners blocked and therefore must have all ingots of the same grade. (14) takes care of this restriction. (15) imposes the limit on the tonnage that can be accommodated by the bottom pour plates (numbered 1 through 8). (16) ensures that the number of ingots poured on any plate does not exceed the capacity of that plate. The last constraint, (17), ensures compatibility of ingot diameters with the plates on which they are poured. If the above model is solved, the values of the decision variables X_{kht} define the schedule.

The above formulation, complex as it is, captures only those aspects of the real problem that can be modeled algebraically. As in many real world problems, there are numerous qualitative and logically complex constraints at the melt shop that cannot be easily captured by a mathematical programming model. More importantly, the melt schedule has a considerable impact on the overall performance of the production system since it affects the workload of the entire production downstream. This is not accounted for by the formulation. The "true" optimal solution to the melt scheduling problem may be untenable when considering the performance of the entire manufacturing facility. Furthermore, as one can quickly conclude from the formulation, even this "simplified"

mixed integer program is too complex for realistic size problems. The above model was implemented in LINGO with standard constraint reduction and tightening methods. Unfortunately, even for small problems, the number of integer variables and constraints become prohibitively large. For instance, a test problem consisting of 40 ingots with a 4 week planning horizon resulted in 20512 variables and 39659 constraints. As one would expect, a melt schedule could not be obtained within a "reasonable" computer time (i.e., several hours on a RISC workstation). However, despite the disappointing computational overlook of this model, it serves as an important basis for developing a computationally efficient heuristic algorithm. The next chapter summarizes the major components of the algorithm and its relationship to the integer programming model.

Chapter 3

A Local Search Heuristic for the Melt Scheduling Problem

From a practical point of view, providing one "optimized" melt schedule does not satisfy the needs for BethForge planner. It is more important to have a decision tool which provides alternative solutions making use of all the information available. This allows the engineers to participate in the decision process by applying their expertise to make the most informed decisions for melt scheduling. To this end, a heuristic procedure is proposed which generates a family of schedules for each week. In this family of schedules each schedule represents a different trade-off between the two scheduling criteria: minimizing material waste and order tardiness. The following sections provide details of the bi-level local search scheme for solving the melt scheduling problem.

3.1 Related Literature

It has been observed by researchers and practitioners that classical shop-scheduling or resource allocation models often provide a poor fit to real industrial problems. Industrial scheduling problems are complicated by enormous technological and operational constraints, and conflicting objectives. Simplifying assumptions and linear, additive single objectives are common in classical models. The melt scheduling problem is formulated as a bi-criterion optimization problem where on-time delivery and resource efficiency are considered simultaneously.

Several methods have been developed in the literature for solving bi-criterion scheduling problems. The earliest bi-criteria procedures (c.f., Smith (1956), Baker (1974)) use the approach where one criterion is fixed at its optimal value, or more generally at an arbitrary value (Emmons (1975)), while a secondary objective is optimized. In more recent development researchers try to optimize both criteria simultaneously. Sen and Gupta (1983) use a linear combination of multiple criteria and solve the resulting single criterion problem. French (1982) considers two criteria independently. He further defines an efficient schedule as one which is not dominated by another schedule for the two criteria under inspection. Wu et. al. (1993) and Liao (1993) use this concept and generate all efficient schedules for the criteria under consideration. In Wu et. al. (1993) a local search method is developed to optimize two conflicting objectives in a single-machine re-scheduling problem. Daniels (1994) and Bernardo and Lin (1994) develop interactive procedures in which the decision maker interacts with the algorithm in order to steer it fast to the most preferred efficient schedule. Chen and Bulfin (1993) analyze the complexity of several single-machine multi-criteria scheduling problems.

Most of the research mentioned above has been devoted to classical single machine scheduling problems. Nevertheless, the solution methodology for the melt scheduling problem shares some commonalities with the above approaches. As in Daniels (1994) and Wu et. al. (1993) a local search heuristic is used to generate a set of efficient schedules each demonstrating a different trade-off between the two criteria. This approach allows the decision maker the final decision about which schedule to choose.

3.2 General Heuristic Structure

The melt scheduling problem is a single-stage, multi-resource, multi-product, cyclic scheduling problem. The scheduling time period is one week since ingots are melted once a week. At the beginning of each period, an order pool of ingots contains all active orders waiting to be processed. A subset of ingots must be scheduled from the order pool to be completed during the current period at the production facility. The ingot schedule must satisfy resource capacity constraints, and, more importantly, all technological and operational constraints. The objective is to optimize up to two independent criteria specified by the decision maker viz. waste and order tardiness. At the end of each period the set of scheduled ingots are removed from the order pool while new ingot orders may arrive which are added to the order pool for future consideration.

Figure 3 shows the outline of the bi-level local search scheme. The appendix provides a pseudo-code for the algorithm. First, a subset of orders in the current order pool with more recent due-dates is considered (Steps 1 and 2). The remainder of the decisions are divided into two levels Steps 4 and 5. In a higher level (Step 4) only the *order selection* decision is made to determine the set of ingots for a single period based on aggregate capacity estimates. This decision may be characterized as a classical knapsack problem where the size of the knapsack is an estimate of total resource capacity. Since many products could share a particular resource the "effective" resource capacity varies depending on the product mix. After a subset of orders are picked from the Knapsack model, level II checks whether this subset can be processed at the facility. Level

II considers the *detailed resource allocation* and *constraint satisfaction* decisions (Step 5). A feasible schedule is built, given the set of orders selected in Level I. Next, the capacity estimate for the knapsack is updated according to the results of the detailed scheduling. The algorithm iterates between the two levels while updating the estimate of capacity in each iteration until a certain termination condition is met (Step 6).

STEP 1 : Update Order Pool

The order pool is updated by adding orders that arrived during the last period.

STEP 2 : Pool Reduction by Due-date Cutoff

From the updated order pool select a subset of products that have earlier due dates than a certain due date cutoff. This cutoff is typically 10 or 15 periods into the future.

Repeat the following for S iterations

STEP 3 : Perturb due date for each product.

STEP 4 : Order Selection Module using Knapsack **(LEVEL I)**

Select a set of products for the current period by heuristically solving a modified KP. Use estimated capacity of the facility as the size of the knapsack.

STEP 5 : Detailed Scheduling Module **(LEVEL II)**

Build a feasible resource allocation for the selected product set. Record objective function values. If all selected products are processed, calculate the capacity slack. If all products not processed, calculate overestimation of capacity.

STEP 6 : Check Termination Condition

The adjustment to the size of the knapsack is positive if there is a capacity slack and negative if there is an over-estimation. If the size of the adjustment has changed signs during the last two iterations, return to STEP 1 Else return the capacity adjustment to STEP 4.

Figure 3. The Bi-Level Local Search Algorithm

The above bi-level solution approach is motivated by several factors. Firstly, order selection and constraint satisfaction are two separable (though not independent) sets of decisions. Secondly, the set of constraints related to the two decisions is decomposable. This is because some of the constraints are applicable only to individual product types whereas other constraints apply jointly to several different product types. This constraint structure suggests the following decomposition: the constraints that are associated with a particular product type and can be isolated from the other products are evaluated as part of the order selection decisions while the constraints coupled across different products are left to be evaluated in the lower level when the detailed resource allocation and constraint satisfaction are considered. An important feature of this decomposition is that a different criterion can be used in each of the two decision levels providing a convenient structure for trading-off the two criteria.

3.3 Implementation of the Bi-level Approach

This section explains in detail the optimization problems at each level of the solution approach and the adaptive feature of the bi-level decomposition.

3.3.1 Level I Problem Description

Level I (STEP 4 in Figure 3) corresponds to ingot selection. A modified knapsack problem representation is used to select the ingots. The objective is to minimize a due-date based function and the volume of the items is the ingot weight. The total capacity of the

knapsack is the estimated total capacity of the melt-shop. The upper bound on this is 6 x 145 = 870 tons as six heats each with a maximum weight of 145 tons are available on the EAF every week. However the pour constraints and ladle furnace constraints in conjunction with the product mix selected will result in a lesser actual capacity usage. The capacity of the knapsack is initially set to an arbitrary value of 80% of maximum (corresponding to b = 0.8) and is adaptively changed as the algorithm iterates between the two levels.

Level II (STEP 5 in Figure 3) is the detailed resource allocation level. Here heats are formed by appropriately combining the ingots selected by the knapsack model. The ingots that can be poured together are then grouped into heats. All the pour resources (e.g., plates, molds) need to be checked for feasibility. The objective at this level is to obtain a "good" feasible resource allocation: one with small percentage of waste and one that is able to process as many ingots as possible from the orders selected in the knapsack. Because of the discrete nature of the heat formation and ladle furnace constraints and the overlapping diameter ranges in the pour resources, obtaining a good resource allocation for the selected ingots is a complex combinatorial problem. The burden in feasibility checking can be somewhat reduced by having some constraints "migrate" to the upper level. This is achieved as follows. The constraints are partitioned into two categories: those that can be checked at the product selection stage (independent of the detailed resource allocation) and those that can be considered only at the pouring stage.

The total weight constraint obviously falls into the first category. Two additional sets of constraints are in this category: (1) the restriction on the total number of ingots allowed in diameter ranges of 40" - 48" (max 24 ingots allowed), and > 92" (max 2 ingots allowed) and (2) the total number of different grades among ingots can not be more than six as only six heats are available per week. The objective function at Level I is minimization of order tardiness. This along with the constraints mentioned above stipulates the optimization problem for Level I. Figure 4 contains a summary of the Level I optimization problem. The objective function (1) is to minimize the penalty caused by

Objective :

$$\min \beta * \Sigma_{k=1}^{N} (f_k - dd_k)^+ \qquad (1)$$

Constraints :

$$\Sigma_{k=1}^{N} \Sigma_h Wt_k * X_{khit} \le b \quad \forall t \qquad (2)$$

$$\Sigma_k Y_{kt} \le 24 \quad \forall t \text{ where } D_k \in \{40''\text{-}48''\} \qquad (3)$$

$$\Sigma_k Y_{kt} \le 2 \quad \forall t \text{ where } D_k \in \{> 92''\} \qquad (4)$$

$$n_{gt} = \begin{cases} 0 \text{ if } Y_{kt} = 0 \ \forall k \text{ where } G_k = g \\ 1 \text{ otherwise} \end{cases} \quad \forall t \qquad (5)$$

$$\Sigma_g n_{gt} \le 6 \quad \forall t \qquad (6)$$

AND constraints (2) through (6) in section 2.3.2

Figure 4. The Level I Optimization Problem

tardiness. (2) imposes the constraint that the total weight of ingots should not exceed the knapsack capacity b. (3) and (4) put a limit on the number of ingots in the diameter ranges 40"-48" and >92" respectively. (5) introduces a 0/1 variable n_{gt} which indicates whether grade g is included in period t. (5) and (6) ensure that at most 6 grades are included in the knapsack during any week t.

3.3.2 Level I Solution Approach

This modified knapsack problem is solved using a simple one-pass heuristic. For every candidate product, a desirability index is calculated based on its due date and weight. The index used was $(\text{due-date})^x(\text{weight})^y$. The order pool is sorted in the ascending order of this index. The algorithm makes one pass through the order pool keeping track of the total weight of included ingots, number of grades included and the number of ingots in the 40"-48" and > 92" diameter ranges. The next candidate ingot is included in the knapsack only if it does not violate any of the constraints described in Figure 4. As mentioned previously, the capacity of the sack is initialized to 80% of the 870 ton EAF upper bound. This number is adapted as the algorithm iterates between the two levels. It should be evident at this point that the actual EAF capacity used will depend heavily on the product mix which ultimately governs which ingots are combined into which heats.

3.3.3 Level II Problem Description

The input to Level II is the solution from Level I which is a set of ingots with

different weights, and belonging to different grades and diameter ranges. The objective is to assign heats and plates to these ingots in a manner that minimizes the waste. A trivial solution is not to process any ingot. However the tardiness criterion does not allow that. We force the model to process as many ingots as possible at the lower level. This approach is quite effective since the ingots that have reached Level II are expected to have relatively early due dates as they have filtered down through the tardiness-minimizing knapsack

Objective :

$$\min \alpha * \sum_{t=1}^{T}\sum_{h=1}^{H} waste(h,t) - \beta * \sum_{t=1}^{T}\sum_{h=1}^{H} Y_{kt} \quad (1)$$

Constraints :

$$LX_{kht} + LX_{mht} \le 1 \quad \forall\, k,m,h,t \text{ where } G_k \ne G_m \quad (2)$$

$$\sum_{k=1}^{N} Wt_k * X_{kht} \le MAXWT \quad \forall\, h,t \quad (3)$$

$$Z_{kpt} \le M(D_{kp}) \quad \forall\, k,p,t \quad (4)$$

$$\left.\begin{array}{l} LX_{kht} + LX_{k,h+2,t} \le 1 \\ LX_{kht} + LX_{k,h+3,t} \le 1 \\ \quad\vdots \\ LX_{kht} + LX_{k,H,t} \le 1 \end{array}\right\} \forall\, k,t \quad \text{for } h=1,2,\ldots,H-2 \quad (5)$$

$$\sum_{h=1}^{H} LX_{kht} \le 2 \quad \forall\, k,t \quad (6)$$

$$waste_{ht} = \max \{ 0 , MINWT - \sum_{k=1}^{N} Wt_k * X_{kht} \} \quad (7)$$

AND constraints (2), (4) through (8) and (14) through (16) in section 2.3.2

Figure 5. Level II Optimization Problem

model. The different constraints for the Level II problem have already been detailed in sections 2.1.1 and 2.1.2. Figure 5 provides a summary of the Level II optimization problem. The objective function (1) seeks to minimize the waste and maximize the number of ingots scheduled. (2) ensures that ingots assigned to the same heat have the same grade. (3) imposes the upper heat weight limit. (4) guarantees compatibility of ingots with plates. (5) and (6) imply that an ingot can be split in at most two heats which are also required to be successive. (7) defines the waste.

3.3.4 Level II Solution Approach

The resource allocation problem can be split into two: Heat Formation and Plate Allocation. Of these the Heat Formation problem is a difficult combinatorial problem as it involves partitioning the ingots of a certain grade into groups that have a total weight of 125 to 145 tons or 250 to 290 tons. The Plate Allocation problem is relatively easier to handle since it is easy to check whether a given set of ingots can be completely poured into the available plates. This section describes the algorithms used for each of these functions and then establishes the link between them.

Heat Formation Ingots of the same grade are combined into heats. Hence a subproblem can be created for every grade in the selected ingot set. These subproblems can be solved independently. Every subproblem consists of a set of ingots with associated weights and due dates. A lower bound k on the total heats required is calculated as the smallest integer

greater than or equal to the ratio of the total of ingot weights and 145. If the total weight is less than or equal to 290 tons this bound can be achieved. If not, k may underestimate the heats required to pack all ingots. The reason for this is the ladle furnace constraint: two ingots may not be split between three successive heats. For example for an ingot weight set {200, 200} the estimate yields k = 3. However the first ingot would need to be split between heat 1 and 2 and the second between heat 2 and 3 which is not permitted technologically. To include this rationale in the heat formation algorithm, "bins" are used. A bin is an imaginary ingot container that can have two sizes: 145 tons and 290 tons. All ingots must be completely accommodated in a bin. The ingot set for a certain grade is now packed into these bins. It is easy to verify that this ensures feasibility of the ladle furnace constraint described earlier. The issue that remains is how many bins and of what size. This question is addressed by enumerating all possible combinations of bins for a given k. For example if k=4, the different combinations are as follows: {145, 145, 145, 145}, {145, 290, 145}, {290, 290}. Distinct permutations of bins are also relevant. The reason is that the heat formation function uses a due-date sorted list of ingots and places them in the bins in a single pass. The function moves on to the next bin if the ingot under consideration cannot be accommodated in the current bin. Thus for k=4, two more bin permutations : {290, 145, 145} and {145, 145, 290} are relevant. Each of these permutations is tested to see whether all ingots can be accommodated.

If none of the permutations accommodate all ingots the number of heats is increased by one, and the process continues (refer to Figure 6) . This process usually stops by the time 4 or 5 heats are reached. There are two reasons for that. First, the set of ingots is restricted by the total weight knapsack capacity constraint. This set is further partitioned into sub-problems for every grade reducing the problem size for the individual sub-problems. When all the sub-problems are solved the total number of heats formed is found. If this total is greater than six, the heats are sorted in ascending order of the smallest ingot due-date, and the first six heats are selected, thus implementing a minmax tardiness policy.

Plate Allocation This is a fairly simple function. First the ingots in the 40" - 48" range are assigned to resource 3 and > 92" diameter ingots to resource 1. The remaining ingots are then assigned in the following order. First the 54" - 69" ingots are assigned to resource 2B if available else assigned to resource 1. 78"-92" ingots are assigned to resource 2A if

1. Create sub-problems by partitioning ingots by grade. Sort ingots by due date within a grade.

2. For each subproblem calculate lower bound k on number of heats required

3. Generate all permutations of bins of 1 or 2 heats that add up to a total of k heats

4. Make one pass through the ingots assigning them to bins in first permutation.

5. Repeat Step 4. until a permutation is found in which all ingots are assigned to heats . Exit if such a permutation is found.

6. If no such permutation found increment k by 1 and goto step 3.

Figure 6. The Heat-Formation Function

available else assigned to resource 1. Finally the 69"-78" ingots are assigned to either 2A or 2B. From Tables 1 and 2 one can easily verify that the above assignment policy will always assign the maximum number of ingots. Within each diameter range, the ingots are ordered by due date. Hence the ingots rejected will have the latest due dates among ingots competing for the same resources.

It should be noted that the ingot set at Level II is the output of a modified knapsack problem with an estimated measure of aggregate capacity. An overestimation of capacity or a skewed product mix could both result in one or more of the selected ingots being rejected in the Heat Formation and/or Plate Allocation functions. These two functions are linked by the zero WIP constraint: whatever is melted must be poured, and by the same token, if a certain ingot cannot be poured, it must not be melted. This introduces a cyclic nature into constraint checking for the heat formation and plate allocation functions and raises an important issue: which of the two functions must be performed first ?

Suppose the heat-formation function is executed first. Some ingots may be rejected while doing this. The reduced pool is now passed through the plate allocation function. It is possible that some more ingots may be rejected in plate allocation. By the zero WIP rule, these ingots must be withdrawn from their heats. This results in a possible increase in material waste if the total heat weight goes below 125 tons. Moreover, there may be an excess capacity in the heat which could be used to accommodate ingot(s) that were rejected previously in the heat formation. It is not difficult to make a similar argument for

the case where the plate allocation function is performed first. It is difficult to justify rigorously any particular order in executing the two functions. However, after testing several case problems using real data it was found that the plate allocation constraints are, in most cases, the binding constraints. An ingot set that has been filtered through the plate allocation function is highly likely to remain intact with no further ingots rejected in the heat formation and EAF capacity check. Therefore, in the implementation, the plate allocation function is executed first followed by the heat formation function.

Problem Space Search Problem Space Search (Storer et. al. 1993) is an efficient method for generating good alternative solutions for a large variety of combinatorial optimization problems. The basic idea is that by perturbing the problem data we can use the same one-pass heuristic to generate alternative solutions. The space of problems formed by perturbing original problem data is called problem space. In this implementation of problem space search, the order due dates are perturbed at two different steps: in the Level I knapsack solver, and in the Level II Heat Formation function. Due dates perturbed once in the knapsack solver are further perturbed in the heat-formation function. In both cases the objective is to generate a large variety of solutions and search for those that are efficient with respect to tardiness and waste.

The knapsack solver and the heat formation function are essentially single pass heuristics which assign ingots pre-sorted in a due date order. Perturbation of due dates in both functions in essence shuffles the sorted order of ingots. This shuffling leads to

different sets of ingots being combined in the heat-formation function providing solution diversity. The "degree" of shuffling can be controlled by the amount of perturbation. Results of testing the algorithm show that this method permits exploration of different areas of the problem space resulting in generation of a large number of efficient schedules.

Adaptive Feedback The result of resource allocation in Level II (or more precisely, the ingots rejected in the process) provides feedback information to the Level I optimization problem. Specifically, this information is used to adjust the capacity of the knapsack in Level I. The need for adjustment is driven by two factors. If initially the knapsack capacity is underestimated, the facility may operate with a lot of slack, causing higher average tardiness then necessary. On the other hand, if the knapsack capacity is grossly overestimated, a large set of ingots will be passed down to Level II for detailed melt scheduling. This may cause a significant increase in computer time and affect the algorithm effectiveness. Hence the objective is to solve the knapsack problem with an "appropriate" level of capacity corresponding to the product mix.

There are three possible outcomes which may result from the Level II computation: (1) No ingots rejected, (2) Ingot(s) rejected in the Plate Allocation function, and (3) Ingots rejected in the Heat Formation function. Important inferences can be derived from these outcomes. If no ingot was rejected, then there is a possible under estimation of capacity. We compute a "capacity slack" associated with every heat as (145 tons - total heat weight). The maximum slack among the six heats in a given week is a relevant figure. If

an additional ingot is to be accommodated in the heats, then its weight cannot exceed the maximum slack. Note that this is true regardless of some ingots being rejected. If entire heats have been eliminated in the heat-formation function then there is likely an over-estimation of capacity which can be corrected by providing a negative adjustment equal to the sum eliminated heat weights. Finally if ingots have been rejected in the plate allocation routine, the reason is most likely a diameter range conflict and may not be related to heat-formation.

Based on the above outcomes, two pieces of information are fed back to Level I: a capacity adjustment and a "tabu" list. The capacity adjustment is equal to (max heat slack) - (tonnage of unmelted ingots). The rationale is that reducing the capacity by unmelted tonnage will eliminate those ingots from the next knapsack and incrementing by maxslack may add a meltable ingot to the knapsack. Any ingot that is rejected by the plate allocation function is made "tabu" for that week by not permitting it to enter the knapsack until the next week. The algorithm thus iterates between Level I and level II unless a termination condition is satisfied. The termination condition stipulated in the algorithm is a change in the sign of the capacity adjustment. A positive adjustment is caused by under estimation of capacity while a negative adjustment signifies over estimation. Hence a sign change in either direction is an indication of "correct" capacity estimation. Figure 7 contains a block diagram of the overall algorithm indicating the roles played by each feature of the bi-level decomposition procedure.

3.4 Enhancements to the Basic Algorithm

Section 3.3 provided details of the basic algorithm that was first developed to be used as a scheduling tool at BethForge. It is clear that the algorithm in this form does not give any special consideration to either of the two criteria - waste and tardiness. Both the

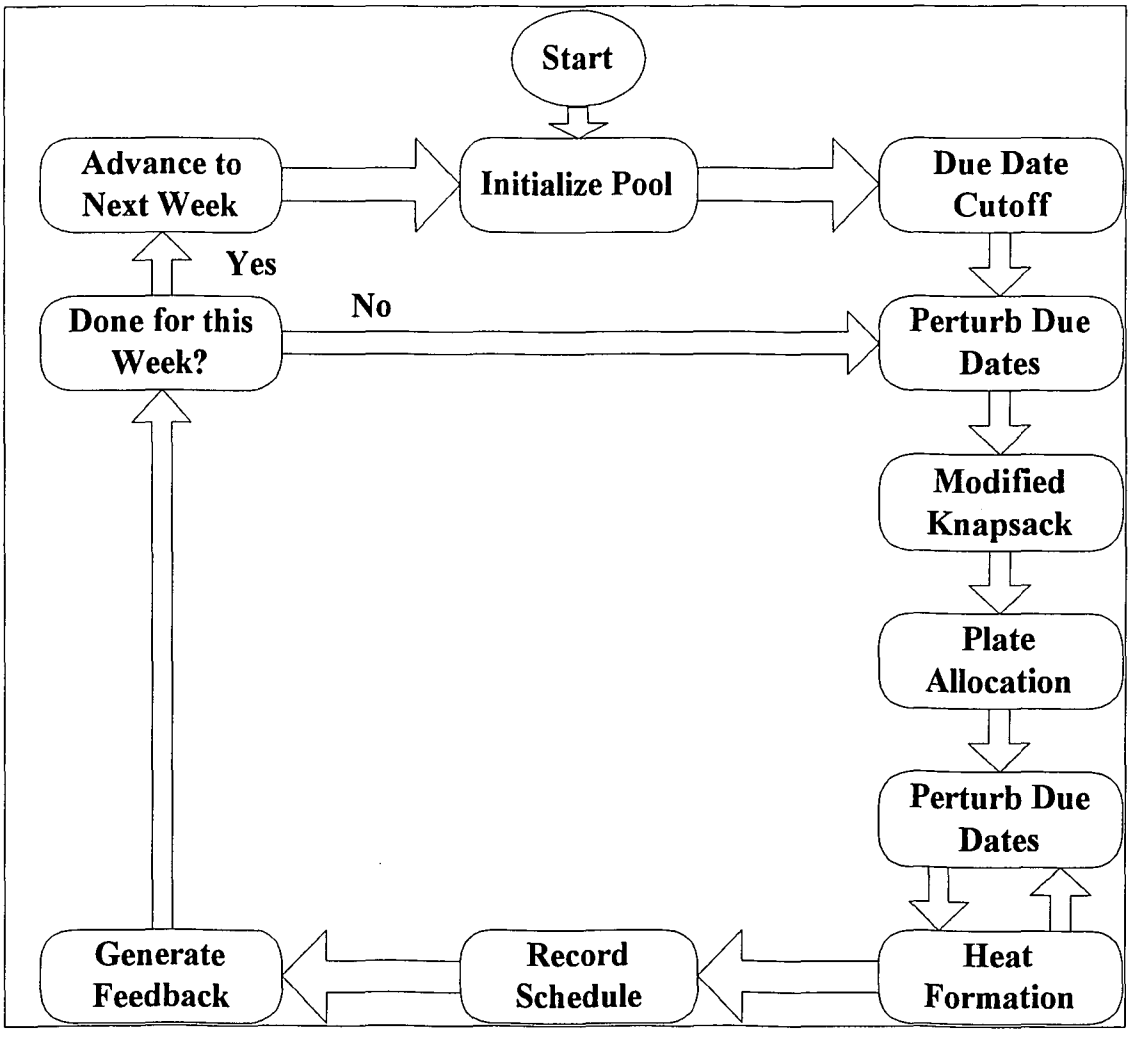


Figure 7. Block Diagram of the Algorithm

objectives are considered more or less equally important and the emphasis of the algorithm

is on producing a large *variety* of solutions to each week's scheduling problem. The variety arises from the algorithm's ability to search the problem space and come up with schedules having different values of waste and tardiness and also a different mix of ingots.

Initial attempts at using the algorithm at BethForge resulted in valuable feedback from the planners at BethForge about some of the enhancements that needed to be made to the algorithm. The following sections describe the enhancements made to the algorithm based on this feedback.

3.4.1 Freezing of Orders

The planning requirements at the melt shop dictate that the first two weeks of the schedule be "frozen". Changing the orders in the first two weeks can be detrimental to melt shop planning since they need sufficient lead time to set up the resources for building the ingot molds. In addition, the BethForge planners also like to "freeze" certain ingots to specific weeks so that timely deliveries for important customers can be ensured. To implement this, the algorithm was modified so as to allow ingots to be flagged as "frozen" to a particular week. The schedule for that week would then be built to be compatible with the frozen ingot(s). The freezing of ingots is achieved by a due date modification. Suppose an ingot is to be frozen to week number 5. At the beginning of week 5, the due date of this ingot is set to a sufficiently small value (smaller than the smallest due date amongst the set of unscheduled ingots). This in effect makes the ingot a high priority item for being scheduled. As a result, this ingot will be selected first both in the knapsack at Level I and

in the detailed scheduling at Level II.

3.4.2 Zero - Waste Schedules

Under current operating environment, the BethForge planners desire to build schedules that have little or no waste due to the high cost of steel and energy. To incorporate this criterion into the algorithm, a parameter called *allowable_waste_per_heat* was introduced. It is the maximum allowable waste per heat that a schedule can have for it to be considered a feasible schedule. This parameter allows the user to specify the level of waste per heat that can be tolerated. In order to implement this, some changes were made in the algorithm. In level II, an additional check was added after the heat formation function that evaluates each "bin" (as defined in section 3.3.4) to see if its waste confirms to that permitted by the *allowable_waste_per_heat* (thus, bins of capacity two heats will have twice as much allowable waste). If the actual waste in a bin exceeds the allowable limit, one of the ingots from that bin is made "tabu" (as explained in section 3.3.4) and the bin is repacked. If the repacked bin has more than the allowable waste, this process is repeated. The procedure continues until either a packing is found that confirms to the waste limit or until all the unscheduled ingots have been made tabu for this bin. The latter results in a schedule that does not confirm to the allowable limits. However, the use of problem space search increases the likelihood of finding an allowable solution if one exists.

In demanding zero-waste schedules every week, the tradeoff between waste and

tardiness is brought into focus. Certain "odd-grade" ingots that cannot be combined with any other ingots tend to get pushed off way beyond their due dates, thus affecting the overall due date performance. Sometimes there are multiple zero-waste schedules possible for a given week. In such a situation, it is important for the planner to get some kind of an estimate of the effects of selecting a particular schedule. To provide such an estimate, the algorithm was further modified to calculate a measure called *remaining tardiness* given that a particular schedule, say schedule S, is selected. The tardiness estimation is simply the total tardiness that would result if all the *unscheduled* (i.e. remaining) ingots were scheduled that week. Clearly, the *remaining tardiness* is a lower bound on the actual minimum tardiness that can be achieved in scheduling all the ingots given that we choose to use schedule S for the current week.

Chapter 4

Computational Testing

The following section describes the experimental design used to determine the important factors affecting the performance of the bi-level local search algorithm. Section 4.2 presents the computational results obtained from actual order data from BethForge.

4.1 Design of Experiments for Testing the Algorithm

We first identify several algorithmic parameters in the local search that need to be determined for the data set. These are:

(1) The total number of iterations (ITER) performed. This is a measure of the total computing resource that is allocated to the algorithm in order to achieve a reasonable performance. Fixing this parameter to a certain value allows a meaningful comparison of the performance of the algorithm when tuning other parameters.

(2) The ratio (RATIO) of the number of iterations used for due date perturbations at Level I and those at Level II. A low value indicates that a small percentage of the total computing time is spent at Level I perturbations and more time is spent at Level II perturbations.

(3) The due date power (X) and the weight power (Y) used in the calculation of the desirability index of ingots in the modified knapsack problem at Level I. Parameters X and Y determine the relative importance given to due date performance and capacity

utilization.

(4) The allowable waste per heat (A_W_P_H) as determined by the user. A_W_P_H could have a considerable impact in the scheduling flexibility. It allows the user to specify whether zero-waste schedules are desired (A_W_P_H set to 0) or whether some wastage can be tolerated.

The main performance measures of interest are the average waste (calculated as a percentage of the total weight of ingots melted) and the average tardiness per ingot (calculated as the total tardiness averaged over the entire ingot pool). In addition, in the experiments performed, the variance of tardiness and the CPU time taken were also tabulated.

4.1.1 The 2^5 Design for Screening

A 2^5 design with two replicates was used to first identify which of the 5 parameters had significant effects on the performance measures. Tables 3 and 4 show the results of the ANOVA carried out on this design with the different levels of the parameters. After this initial screening experiment a full-factorial design with two replicates was used to assess the impact of different levels of the significant parameters on the performance

	ITER	RATIO	X	Y	A_W_P_H
Low	20	1:5	1	0	0
High	125	5:1	3	2	15

Table 3 A 2^5 Design for Initial Screening Experiment

measures. The information from this experiment is useful in tuning the algorithm for producing solutions with different emphasis on the performance measures.

Table 3 shows the actual values of the parameters corresponding to the High and Low settings in the 2^5 design. In Table 4 the mean (μ) and standard deviation (σ) for each performance measure is tabulated. The mean and variance are collected across all the settings of the remaining parameters. Thus, each mean is calculated from 32 observations. The column labeled DT indicates the result of Duncan's Multiple Range test with a probability of type I error (α) of 0.05. A 'Y' in this column indicates that the means for the two settings of the corresponding parameters are significantly different. 'N' indicates that the means are not significantly different. The results are based on a design with two replicates and hence a total of $2^5 * 2 = 64$ experiments. The data for these experiments was drawn from two different real data sets from BethForge. In each table, A_W_P_H is measured in tons.

At this point, it is important to note that the above experiments were not conducted on a *rolling horizon* basis. Since data on arrival dates of orders was not available, the experiments were performed assuming a static pool of ingots with no new order arrivals over time. It should be understood that the flexibility and full potential of the algorithm are constrained by this restriction. In the actual scheduling environment, new orders are added to the pool on a regular basis. Doing this allows the planners to continuously have updated data and consequently more flexibility in forming heats and trading off waste with tardiness. The relevance of this will become clear when the results

of scheduling entire order pools are presented.

The following conclusions can be drawn from table 4. Only parameter A_W_P_H has a significant effect on the average tardiness. A_W_P_H affects the flexibility available in scheduling the ingots. Setting A_W_P_H to zero leaves no room for trading off waste

Factor	Level	Performance Measures											
		Tardiness (weeks)			σ of Tardiness (weeks)			%Waste			CPU Time (secs)		
		μ	σ	DT	μ	σ	DT	μ	σ	DT	μ	σ	DT
ITER	L	2.89	1.16	N	5.36	1.59	N	9.71	2.09	N	126	112	Y
	H	2.66	0.76		5.18	1.35		9.18	1.89		428	262	
RATIO	L	2.63	0.90	N	5.13	1.50	N	9.46	2.09	N	182	170	Y
	H	2.92	1.05		5.41	1.44		9.43	1.92		372	284	
X	L	2.64	1.27	N	5.18	1.80	N	9.92	2.12	N	317	252	Y
	H	2.91	0.55		5.36	1.05		8.97	1.76		237	248	
Y	L	2.73	1.19	N	5.10	1.57	N	9.48	1.99	N	268	253	N
	H	2.82	0.73		5.44	1.36		9.41	2.03		286	253	
AWPH	L	3.51	0.77	Y	6.52	0.72	Y	9.00	1.81	N	376	288	Y
	H	2.05	0.51		4.03	0.82		9.90	2.09		178	159	

Table 4 Results of 2^5 Design for Screening Parameters

in favor of tardy ingots. Odd grade ingots tend to get pushed back whenever alternate zero-waste schedules can be found. The interaction effect between X and A_W_P_H was also found to be significant for average tardiness. X is the due date power in the desirability index calculation. Hence, it is natural to expect X to have some influence on the average tardiness. Likewise, the variance of tardiness for a given pool of ingots (calculated based on the tardiness values of each ingot in the pool) is affected only by A_W_P_H.

The average waste is not affected significantly by any factor. At first sight, this seems to be a somewhat surprising result. However, it should be borne in mind that the waste figures being tabulated are *averaged* across all the ingots in the pool. Since the experiments were not conducted on a rolling horizon basis, the total amount of waste for a given order pool remains fairly constant, subject to a particular heat packing algorithm (since no new ingots are added to the pool). Hence, it is clear that the average waste will be almost constant irrespective of the parameter values.

The computational time, as expected, depends on the total number of iterations (ITER) and also on how the computation effort is divided between the two levels of the algorithm (RATIO). As in the case of tardiness, the CPU time depends on both X and A_W_P_H. Both these factors affect the scheduling flexibility by impacting the tardiness and the allowable waste. As scheduling flexibility reduces, it is natural to expect longer computation times to build schedules.

4.1.2 The Full Factorial Design

Based on an analysis of the 2^5 experiments, a full factorial experiment was designed to test the effects of various settings of the parameters X and A_W_P_H which were found to be significant factors affecting the two primary performance measures viz. tardiness and waste. The other factors were set to levels best suited for minimizing the CPU time. Tables 5 summarizes the results of the ANOVA.

In the full factorial experiment, 8 levels of X were investigated, each for 4 levels of A_W_P_H. Again, two replicates were used and this gave rise to 8 * 4 * 2 = 64 experiments. As before, table 5 shows mean and standard deviation of the average tardiness, the standard deviation of tardiness, the average percentage waste and the CPU time. The mean for each cell for the factor X is calculated across the 4 levels of A_W_P_H. With two replicates for each A_W_P_H level, this gives 8 observations. Similarly, the mean for each A_W_P_H cell is based on two replicates each of the 8 levels of X, and thus on 16 observations. The same holds true for the standard deviations. The column DT indicates the result of Duncan's Multiple range test applied with a probability of type I error of 0.05. In the Duncan grouping, treatment means that are significantly different are assigned different letters. Means with the same letters are not significantly different.

Factor	Level	Performance Measures											
		Tardiness (weeks)			σ of Tardiness (weeks)			% Waste			CPU Time (secs)		
		μ	σ	DT	μ	σ	DT	μ	σ	DT	μ	σ	DT
X	0.0	2.86	0.72	A	5.81	1.42	A	11.0	1.08	A	12.8	5.02	B
	0.5	2.89	0.92	A	5.83	1.76	A	10.5	0.59	A	13.3	5.62	B
	1.0	2.93	1.23	A	5.99	2.00	A	11.4	0.95	A	19.6	7.79	A
	1.5	2.93	1.23	A	5.99	2.00	A	11.4	0.95	A	19.3	7.65	A
	2.0	2.93	1.23	A	5.99	2.00	A	11.4	0.95	A	19.6	7.65	A
	2.5	2.86	0.72	A	5.81	1.42	A	11.0	1.08	A	13.1	5.08	B
	3.0	2.86	0.72	A	5.81	1.42	A	11.0	1.08	A	13.6	5.18	B
	3.5	2.86	0.72	A	5.81	1.42	A	11.0	1.08	A	13.2	4.94	B
AWPH	0	4.02	0.47	A	7.81	1.02	A	10.3	0.62	A	23.7	5.17	A
	5	3.14	0.48	B	6.28	0.88	B	10.5	0.43	A	16.6	4.78	B
	10	2.40	0.19	C	5.20	0.70	C	12.1	0.75	B	12.5	3.11	C
	15	2.00	0.69	D	4.23	1.01	D	11.5	0.68	C	9.5	2.00	D

Table 5 Results of ANOVA on Full Factorial Design

From table 5 it is clear that A_W_P_H has a significant impact on each performance measure. The general trend is an improvement in the performance measures as A_W_P_H increases from 0 to 15 tons. This clearly demonstrates the increased scheduling flexibility and simplification of the problem as A_W_P_H is allowed to slacken. The mean tardiness reduces to almost half its value as A_W_P_H goes from 0 to 15 tons. However, the increase in waste is not as drastic. Again, the reason for this is the fact that the waste figures represent the average waste for a non-rolling horizon experiment. The impact of A_W_P_H on the CPU time is even more dramatic than that on the average tardiness. The factor X has a significant effect only on the CPU time. The ANOVA for this design indicated absence of significant second order interactions.

4.2 Algorithm Performance on Test Problems

The 2^5 and full factorial experiments provided valuable information on the effect of different parameter levels on the performance criteria. This section details the additional tests carried out on the algorithm to demonstrate two important features of the algorithm. The first set of tests show its ability to generate a wide variety of schedules (with different waste and tardiness values) for each week. This is one of the requirements of the BethForge planners since they require a tool that will allow them to look at alternative schedules to schedule the melt shop. The second set of tests is designed to demonstrate how the algorithm can be tuned to specific requirements. Three cases are examined here. The algorithm is tuned to provide (I) zero-waste solutions with frozen ingots, (ii) zero-

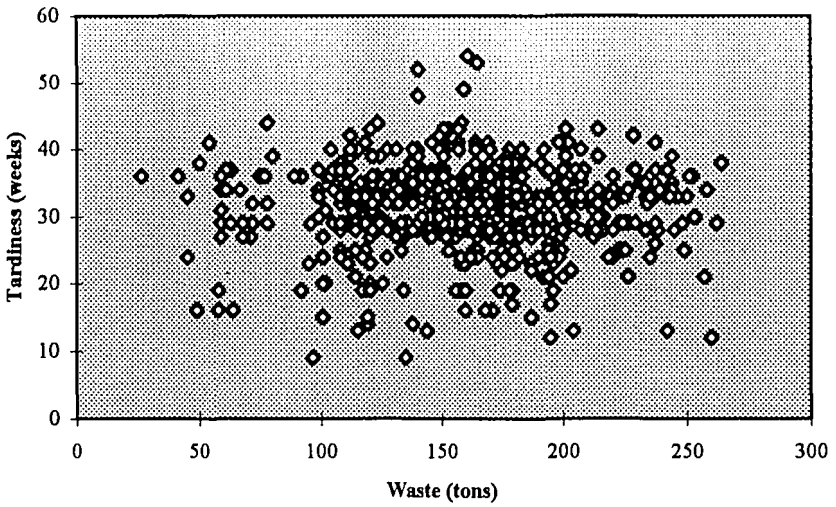
waste solutions without frozen ingots and (iii) low-tardiness solutions without frozen ingots.

The tests have been carried out on three sets of data form BethForge. These data sets are different from the ones used for the parameter tuning experiments. The results of these tests are shown graphically in figures 8 through 13. The following sections discuss these results in detail.

4.2.1 Bicriteria Plots

Figures 8 through 10 each show two representative bi-criteria plots. Here the waste is represented in tons and the mean tardiness in weeks. These plots visually demonstrate the trade-off between the two objectives. An efficient frontier of non-dominated solutions can be observed in each of the plots. These frontiers suggests an obvious tradeoff between waste and tardiness. Another important conclusion that can be drawn from these plots is the high data dependency of the performance of the algorithm. In certain weeks, the mix of ingots is such as to enable a large number of desirable solutions (e.g. the second plot in figure 10). In some other cases, there may be only one or two "good" solutions found by the heuristic.

Data Set 1, Week 3, Zero Waste with Frozen Ingots



Data Set 1, Week 4, Zero Waste with Frozen Ingots

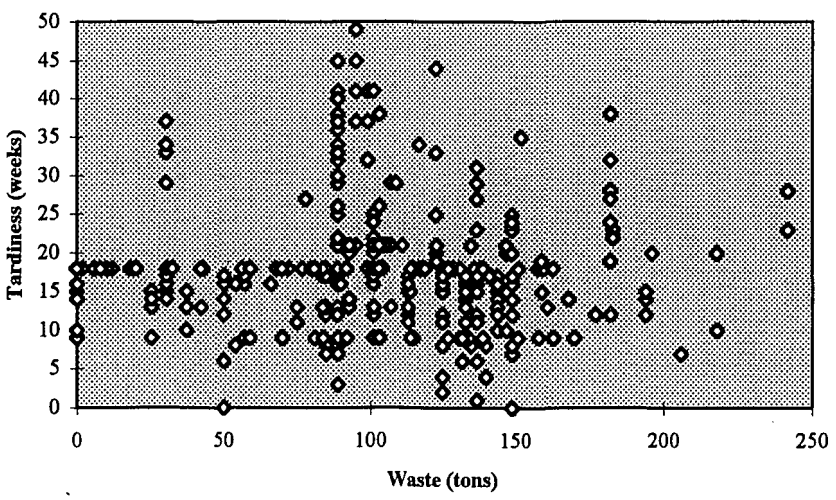
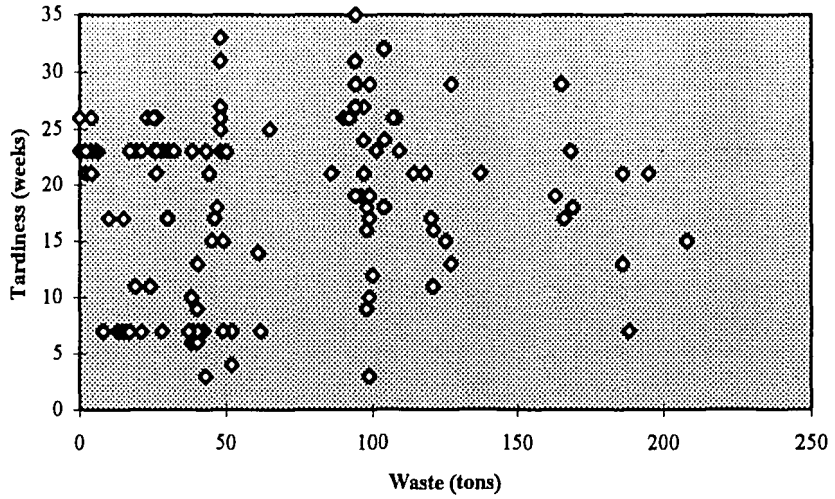


Figure 8. Bicriterion Plots for Zero Waste with Frozen Ingots

Data Set 3, Week 4, Zero Waste without Frozen Ingots



Data Set 3, Week 3, Zero Waste without Frozen Ingots

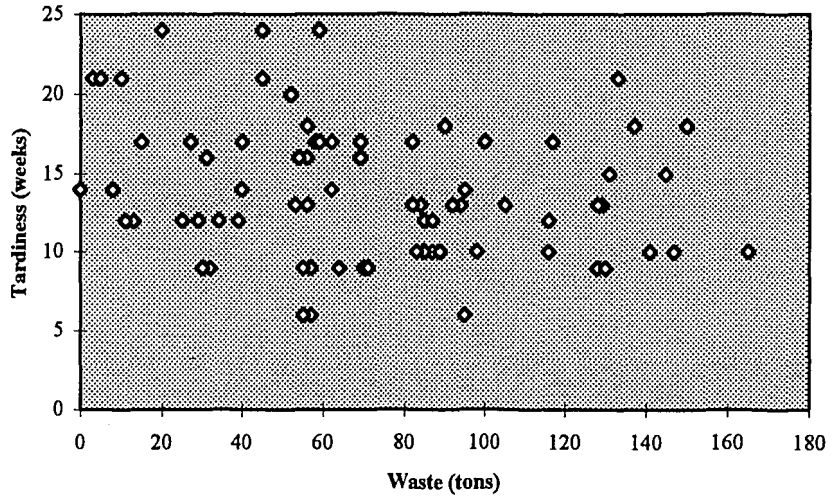
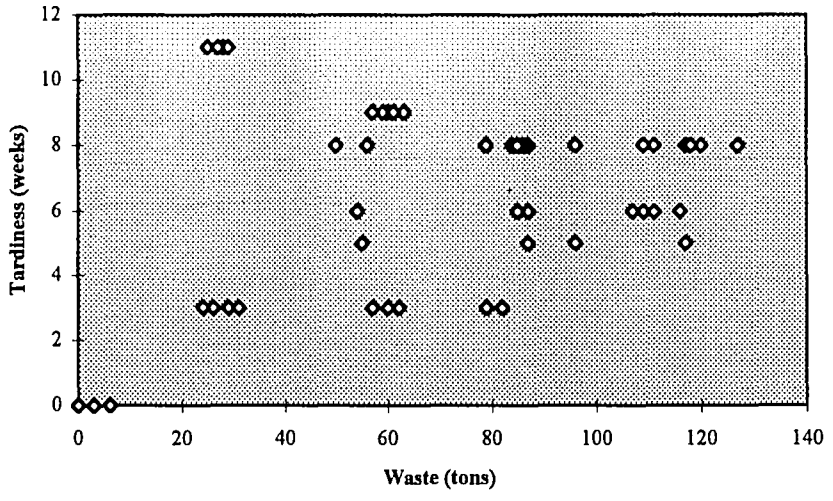


Figure 9. Bicriterion Plots for Zero Waste without Frozen Ingots

Data Set 3, Week 2, Low Tardiness without Frozen Ingots



Data Set 3, Week 4, Low Tardiness without Frozen Ingots

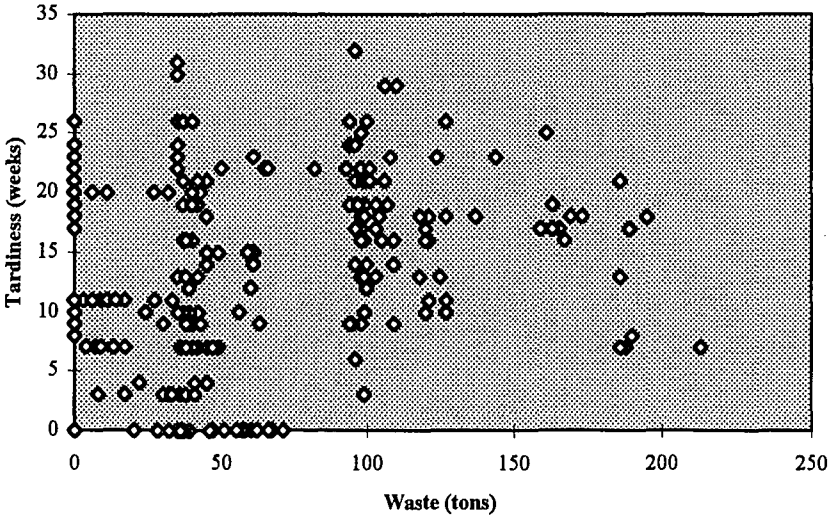


Figure 10. Bicriterion Plots for Low Tardiness without Frozen Ingots

4.2.2 Bar Graphs

Figures 11 through 13 show three bar graphs for each of the test data sets. These graphs plot the average waste (in tons) and the average tardiness per ingot (in weeks)for each week. The three graphs for each data set illustrate how the algorithm can be tuned to specific planning requirements. The first graph in each set is based on data containing frozen ingots and the emphasis is on generating zero-waste schedules. The second graph also emphasizes zero-waste schedules but now the restriction on freezing some ingots is removed. The third graph emphasizes low tardiness without frozen ingots.

Several important inferences can be drawn from these graphs. First, it must be noted that the experiments were not performed on a rolling horizon basis. This explains the high waste figures towards the end weeks. Once most of the ingots have been scheduled, there is very little flexibility available to schedule the few remaining ingots. Hence waste reduction becomes increasingly difficult as one progresses through the weeks. As we move from the first to the second graph, the scheduling flexibility increases since now none of the ingots are required to be frozen in any week. This allows the algorithm to search a larger solution space and consequently results in better waste figures. It is clear from the graphs that freezing the ingots improves the due date performance since tardiness is generally lower. This is easily explained by the fact that the planners at BethForge freeze certain important customer orders to minimize the delays in shipping to these customers. From the third graph it is clear that as the emphasis shifts to due date performance, the waste figures increase.

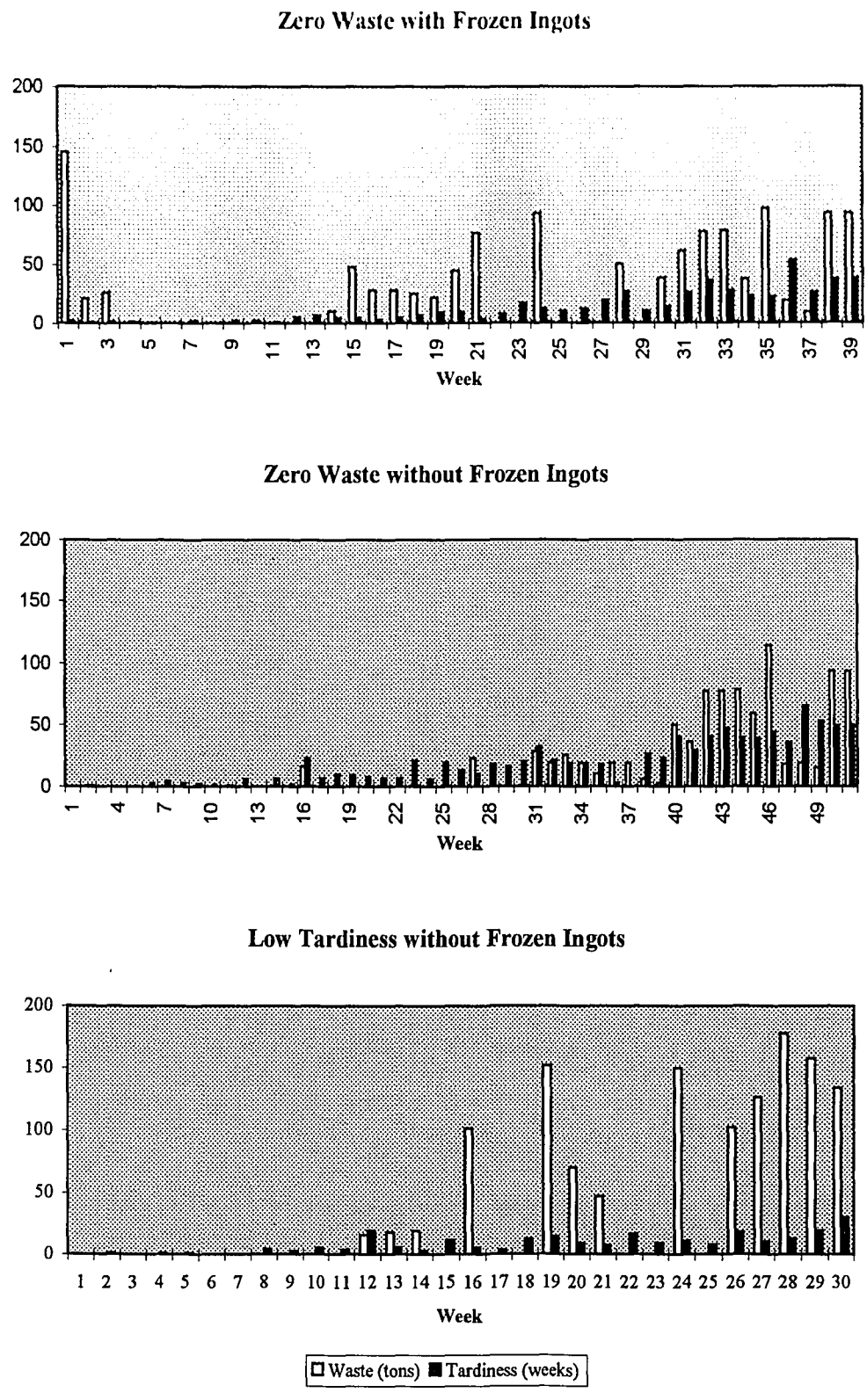


Fig 11. Bar Graphs for Data Set 1

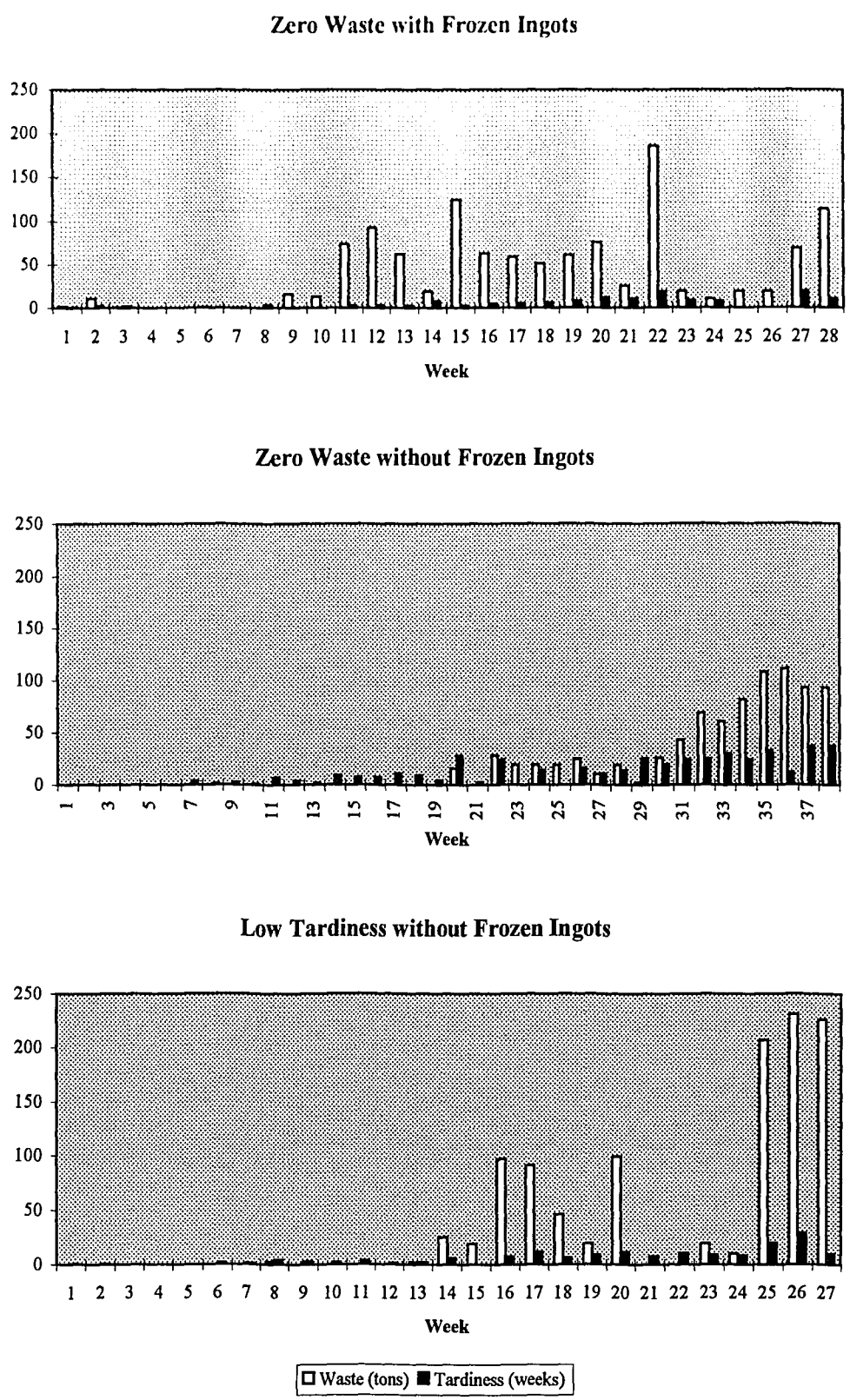


Fig 12. Bar Graphs for Data Set 2

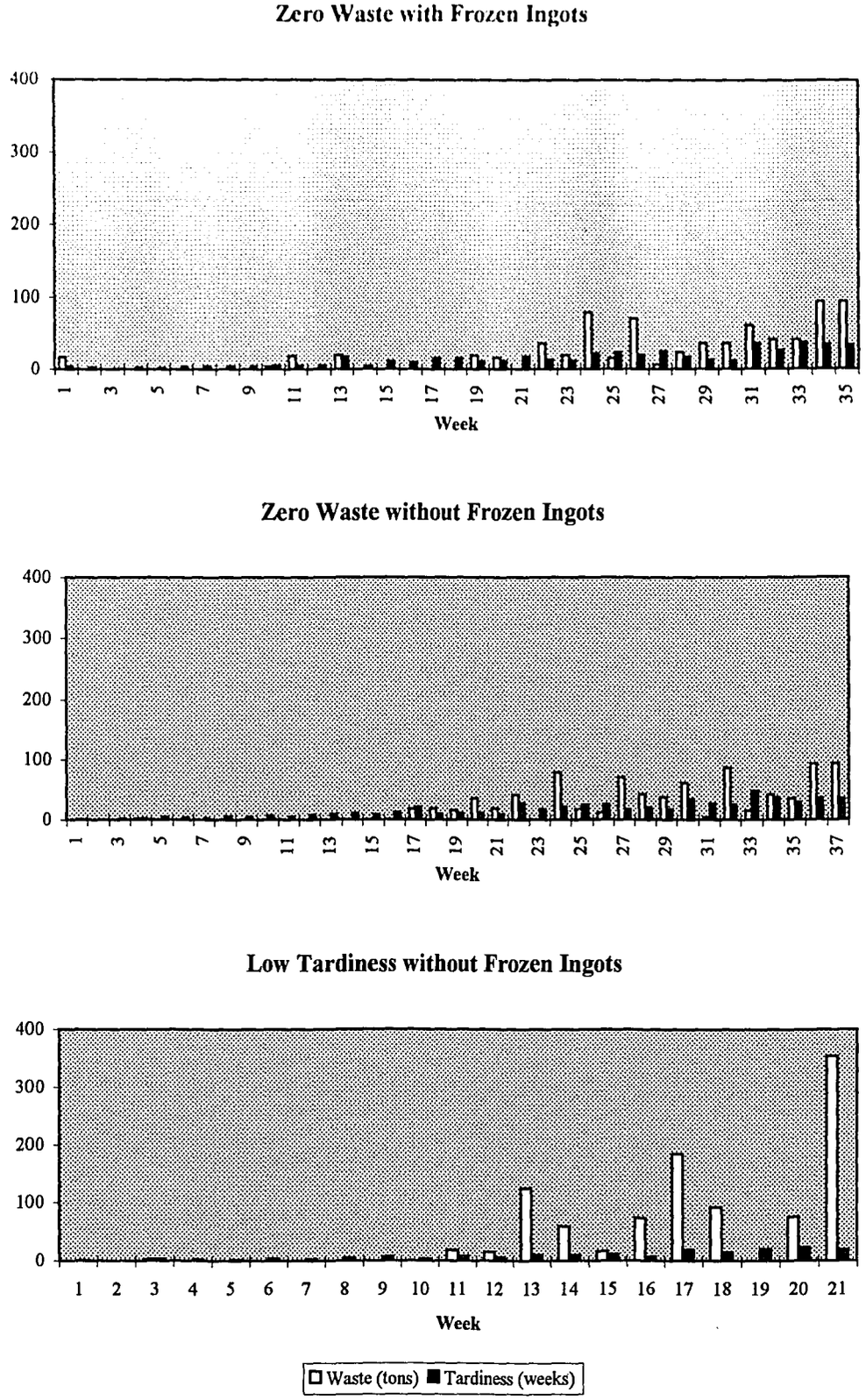


Fig 13. Bar Graphs for Data Set 3

Chapter 5

Conclusion

5.1 Inferences From Tests on the Algorithm

The melt scheduling problem examined in this thesis is an example of a heavily constrained scheduling environment where even finding a feasible solution can be a challenging task. A mixed integer programming formulation was developed for it which models some of the constraints that can be captured in standard algebraic form. A local search procedure was developed for this problem. The procedure was tested on actual order data and its utility as a decision support tool was shown.

The results presented in sections 4.2.1 and 4.2.2 clearly bring out the tradeoff between the two conflicting objectives of waste and tardiness minimization. The bicriteria plots visually demonstrate this tradeoff. It is also clear from the plots and the bar graphs that the melt scheduling algorithm developed in this thesis provides a decision tool that can be used by the BethForge planners to examine alternate solutions to each week's melt scheduling problem. This tool can be especially helpful when equipment failures or other unforseen events require a complete rescheduling of the ingot pool. In such situations, the algorithm can be used to quickly establish good starting solutions for building a new schedule. Thus, problem space search embedded within a heuristic scheduling scheme can be an effective technique for generating alternate schedules for otherwise hard to solve bicriteria optimization problems.

The information contained in the bar graphs in section 4.2.2 also allows the planners to examine long term effects of current scheduling policies such as zero-waste scheduling or "order freezing". For instance, if one insists on zero-waste schedules in the earlier weeks, certain "odd grade" and "odd sized" ingots may be forced to be unreasonably tardy and the effects of this can be observed graphically.

It may be possible to generalize the melt scheduling algorithm to other bicriterion optimization problems having a similar structure to this problem. Specifically, if a problem involves separable constraint sets, it might be possible to extend some features of this algorithm to develop an effective heuristic procedure applicable to that problem. The success of the approach in this case relies in no small part on the effective exploitation of the special structure of constraints at the melt shop.

5.2 Future Work and Improvements to the Algorithm

Several avenues are still open to extend the melt scheduling algorithm to make it a more useful decision making aid. One such direction is to incorporate the downstream effects of the melt schedule. As described earlier, the melt facility serves to load the entire BethForge production facility following the melt shop. The next logical step in the development of the algorithm should therefore be to assess the impact of different types of melt schedules on the forge resources. Based on the information available at this stage, it is already clear that the forge shop has a certain "preferred" mix of ingot types. A desirable situation, therefore, would be to be able to construct melt schedules that not only

optimize the melt shop but are also "forge friendly".

Another direction of enhancement would be to incorporate into the algorithm some rules to prescribe *grade consolidation*. This means suggesting different (but similar and acceptable) grades for certain odd varieties of grades that cause wastage. This is one of the options already being considered by the planners at BethForge. Another useful feature that could be used in conjunction with grade consolidation is the ability to requote due dates based on revised estimates of finish times at the melt shop. Very often, disruptions cause the original schedules to be invalid. As a result, many orders often end up being late beyond their expected due dates. By making appropriate changes to the scheduling algorithm, it should be possible to get revised estimates of completion times in such cases of rescheduling.

Finally, some of the implementation aspects of the algorithm itself can be redesigned. One such possibility is to use a clustering algorithm to cluster together ingots of the same grade and then attempt to generate pour-feasible heats from these clusters.

REFERENCES

Baker, K.R., *Introduction to Sequencing and Scheduling,* Wiley, New York, 1974.

Bernardo, J.J. and Kun-Si Lin, "An interactive procedure for bi-criteria production scheduling", *Computers and Operations Research*, 21(6), 677-688, 1994.

Chen, C., and R.L.Bulfin, "Complexity of single-machine multi-criteria scheduling problems", *European Journal of Operational Research,* 70, 115-125, 1993.

Daniels, R.L., "Incorporating preference information into multi-objective scheduling", *European Journal of Operational Research*, 77, 272-286, 1994.

Dileepan P., and T. Sen, "Bi-criterion static scheduling for a single machine", *Omega*, 16, 53-59, 1988.

Emmons, H., " A note on a scheduling problem with dual criteria", *Naval Research Logistics Quarterly*, 22, 615-616, 1975.

French S., *Sequencing and Scheduling*, Ellis Horwood, Chichester, 1982.

Liao, Ching-Jong, "Tradeoff between setup times and carrying costs for finished items", *Computers and Operations Research*, 20(7), 697-705, 1993.

Nelson, R.T., R. K. Sarin and R. L. Daniels, "Scheduling with multiple performance measures: the one-machine case", *Management Science*, 32, 464-479, 1986.

Sen T., and S. K. Gupta, "A branch-and-bound procedure to solve a bi-criterion scheduling problem", *AIIE Transactions*, 15, 84-88, 1983.

Storer R.H., S. David Wu and R. Vaccari, "New search spaces for sequencing problems with application to job shop scheduling", Management Science, 38(10), 1495-1509, 1992.

Smith W.E., "Various Optimizers for single stage production", *Naval Research Logistics Quarterly*, 22, 585-592, 1975.

Wu, S.D., R.H. Storer and P. Chang, "One-machine rescheduling heuristics with efficiency and stability as criteria", *Computers and Operations Research*, 20(1), 1-14, 1993.6.

Appendix

This appendix provides a pseudo-code for the bi-level local search algorithm described in chapter 3. The computer code for this algorithm is available at:

H. S. Mohler Laboratory

200 W. Packer Avenue

Lehigh University

Bethlehem, PA 18015

Phone: (610) 758-4050

Notation :

w : indexes the week being considered for scheduling

I : indexes ingots

due_date(I) : due date of ingot *I*

-INF : a large negative number

waste(P) : the amount of waste incurred in scheduling pool P

tonnage(P) : the tonnage of steel melted in pool P

b : capacity of the knapsack

MAXCAPACITY : maximum capacity of the melt shop

adjustment : change in knapsack capacity recommended by procedure *lolevel*

CUTOFF : a due date cutoff to consider only a subset of ingots for scheduling

hsrs : number of HSRs included in knapsack

ldrs : number of LDRs included in knapsack

others : number of other type of ingots (neither HSRs nor LDRs) included in knapsack

priority(I) : a priority index for ingot *I*

X : a parameter governing the importance of due date in calculating the priority index

Y : a parameter governing the importance of weight in calculating the priority index

MAXHSRS : maximum number of HSRs that can be accommodated in knapsack

MAXLDRS : maximum number of LDRs that can be accommodated in knapsack

MAXOTHERS : maximum number of other ingots that can be accommodated in knapsack

tardiness(P) : the total tardiness of ingots scheduled in pool P

tot_heats : total number of heats needed to melt all ingots in a pool

MAXHEATS : maximum number of heats allowed per week

num_elim : number of heats eliminated by procedure *eliminate_heats*

Pseudo-Code :

procedure *main* :

update order file with new data;

set $w = 1$;

while there are more ingots to be scheduled do:

    procedure *search*;

procedure *search* :

for every ingot *I* in the pool do:

if ingot *I* is to be frozen in week *w* do:

set *due_date(I)* = -INF;

do procedure *adapt* and save ingot pool as *bestP*;

for S iterations, using dummy pool *dP*, do:

for each ingot *I* perturb *due_date(I)*;

procedure *adapt*;

if *waste(dP)* < *waste(bestP)*

replace *bestP* by *dP*;

if [*waste(dP)* = *waste(bestP)*] AND [*tonnage(dP)* > *tonnage(bestP)*]

replace *bestP* by *dP*;

record *bestP* as the best ingot pool for week *w*;

w = *w* + 1;

procedure *adapt* :

set knapsack capacity *b* = 0.8 * MAXCAPACITY;

do procedure *select_ingots*;

do procedure *knapsack*;

do procedure *lolevel*;

while *adjustment* does not show a sign change, do:

adapt knapsack capacity thus: *b* = *b* + *adjustment* ;

do procedure *knapsack*;

do procedure *lolevel*;

if a better ingot pool is found, replace *dP* by it;

procedure *select_ingots* :

for each unscheduled ingot *I* do:

if *due_date(I)* < CUTOFF

include *I* in pool;

procedure *knapsack* :

set *hsrs* = *ldrs* = *others* = 0;

for each ingot *I* in the pool:

calculate the priority index: $priority(I) = [due\_date(I)]^X * [weight(I)]^Y$;

sort ingots in ascending order of *priority(I)* (most urgent ingots first);

for each ingot *I* (in sorted order) do:

classify the ingot as an HSR, LDR or Other based on diameter;

if *I* is an HSR do:

if *hsrs* < MAXHSRS

include *I* in knapsack;

hsrs = *hsrs* + 1;

if *I* is an LDR do:

if *ldrs* < MAXLDRS

include *I* in knapsack;

ldrs = *ldrs* + 1;

if *I* is Other do:

if *others* < MAXOTHERS

include *I* in knapsack;

others = *others* + 1;

procedure *lolevel* :

sort ingots by due dates;

do procedure *check_pour*;

sort ingots by due dates;

do procedure *check_melt*;

save current ingot pool as *bestP'*;

calculate *waste(bestP')*, *tardiness(bestP')* and *tonnage(bestP')*;

for P iterations do:

for each ingot *I* perturb *due_date(I)*;

label the new pool thus generated as *dP'*;

sort ingots by due dates;

do procedure *check_pour*;

sort ingots by due dates;

do procedure *check_melt*;

calculate *waste(dP')*, *tardiness(dP')* and *tonnage(dP')*;

if *dP'* better than *bestP'* save *dP'* as *bestP'*;

calculate *adjustment* = slack or overestimation of *b* for pool *bestP'*;

procedure *check_pour* :

for each ingot *I* (in the sorted pool) do:

assign *I* to a compatible plate if capacity exists on the plate;

if *I* cannot be assigned to a plate, make it tabu;

procedure *check_melt* :

for each grade of steel in the ingot pool, do:

calculate the total tonnage of all ingots of that grade;

calculate the aggregate number of heats required for that grade;

procedure *check_split_feasibility*;

calculate *tot_heats* = total number of heats required to melt all ingots in the current pool;

if *tot_heats* > MAXHEATS

do procedure *eliminate_heats*;

procedure *check_split_feasibility* :

for the grade under consideration, attempt to pack all ingots of the grade into bins of one and two heats in the following order, continuing until either all ingots of the grade have been packed or it is necessary to use more than six heats to pack all ingots:

pack in 1 bin of 1 heat;

pack in 1 bin of 2 heats;

pack in 1 bin of 1 heat and 1 bin of 2 heats;

pack in 1 bin of 2 heats and 1 bin of 1 heat;

pack in 2 bins of 2 heats;

pack in 1 bin of 1 heat, 1 bin of 2 heats, 1 bin of 1 heat;

pack in 2 bins of 2 heats, 1 bin of 1 heat;

pack in 1 bin of 2 heats, 1 bin of 1 heat, 1 bin of 2 heats;

pack in 1 bin of 1 heat, 2 bins of 2 heats;

pack in 3 bins of 2 heats;

pack in 1 bin of 1 heat, 1 bin of 2 heats, 1 bin of 1 heat, 1 bin of 2 heats;

pack in 1 bin of 2 heats, 1 bin of 1 heats, 1 bin of 2 heats, 1 bin of 1 heat;

pack in 1 bin of 1 heat, 2 bins of 2 heats, 1 bin of 1 heat;

procedure *eliminate_heats* :

set *num_elim* = 0;

calculate *excess_heats* = *tot_heats* - MAXHEATS;

while *num_elim* < *excess_heats* do:

eliminate heat with earliest due date ingot;

num_elim = *num_elim* + 1;

Vita

Bhavin J. Doshi

Place of Birth: Bombay, India **Date of Birth:** 31st January, 1973

Names of Parents: Mr. J. K. Doshi and Mrs. P.J. Doshi

Education:

Lehigh University, Bethlehem, PA

MS Industrial & Manufacturing Systems Engineering, June 1996

Concentration in Operations Research

Indian Institute of Technology - Bombay, India

B Tech Mechanical Engineering, July 1994

Positions Held:

Secretary, Graduate Student Council, Lehigh, 1995-96

Treasurer, Indian Students Association, Lehigh, 1995-96

END OF
TITLE