Optimization Models for Production Planning in Metal Sheet Manufacturing

by

Srimathy Gopalakrishnan

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1994

Author ....
Department of Electrical Engineering
and Computer Science
December 17, 1993

Certified by ....
Anantaram Balakrishnan
Associate Professor, Sloan School of Management
Thesis Supervisor

Accepted by ....
Richard C. Larson
Co-director, Operations Research Center

Optimization Models for Production Planning in Metal Sheet Manufacturing

by

Srimathy Gopalakrishnan

Submitted to the Department of Electrical Engineering and Computer Science on December 17, 1993 in partial fulfillment of the requirements for the Degree of Master of Science in Operations Research

Abstract

In this thesis, we address the tactical planning decision problem of ingot sizing in an aluminum sheet manufacturing facility. Ingots used for sheet manufacturing are made-to-stock, and used when necessary, to satisfy customer demands. The facility produces large ingots to exploit economies of scale in ingot casting, but customers order products frequently, and in small quantities. In this situation, the facility's current practice of dedicating an ingot to each order generates large amounts of scrap and increases processing costs. To prevent this, the facility is considering an alternate strategy of combining more than one order for production on a single ingot.

When we permit multiple orders to be jointly produced from the same ingot, what standard ingot sizes should the facility produce, and which orders should be combined to minimize total scrap? We group similar orders over a long planning horizon into one product. Given the forecast demand for each product, a set of candidate ingot sizes, and a set of feasible product combinations, we need to determine the standard set of ingot sizes, and the number of times each product combination is produced on the standard ingots to minimize total scrap, while satisfying demand for all products.

We formulate the ingot sizing problem as an integer program, and develop an efficient solution procedure. The solution procedure consists of dual ascent to obtain lower bounds, and two heuristics to provide good feasible solutions. We have implemented the dual ascent procedure and the heuristics, and tested them with data on actual orders received at a leading aluminum sheet manufacturing facility.

Our computational results indicate that the solutions obtained by the dual ascent and heuristic procedures are within 4% of optimality on an average. For the alloy that we studied, a comparison of the proposed set of standard sizes with the current set of ingots suggests that the proposed solution could reduce total scrap by an average of 9.5%. The reduction in total scrap could result in savings of up to $100,000 annually in scrap reprocessing and ingot casting costs.

Thesis Supervisor: Anantaram Balakrishnan
Title: Associate Professor, Sloan School of Management

To my parents

and my husband

Acknowledgments

I would like to thank my advisor, Anantaram Balakrishnan, for introducing me to this real world problem and motivating me to work on it. I very much appreciate his support and guidance for the past two years of my student life. He has been very understanding and has helped me in several aspects to manage my research work and my long distance marriage!! And finally, my heartfelt thanks to him for reading my thesis at such short notice.

I would like to thank all my friends at the ORC for making it almost home. I will cherish the special friendship that I have developed with Gina and Rodrigo forever. I would also like to recall the happy evenings spent with Kamala and her family. Thanks a million to Paulette and Laura for helping me take care of all the MIT requirements.

Special thanks to my cousin Lakshmi and her husband Venkat for their care and support, and many relaxing weekends. I would also like to thank my suite mates for the wonderful times we had together.

I do not know how to thank my husband. He has been a constant source of encouragement and inspiration for me. He has sacrificed a lot for me over the past few years, and I hope I can make it up to him someday. I cannot thank my family and my husband's family enough for supporting our decision to stay away from each other.

Contents

4. Computational Results

5. Conclusions and Recommendations

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Motivation

This thesis addresses a tactical planning decision problem in a make-to-order aluminum rolling facility. The facility manufactures sheet and plate products to customer specifications. Traditionally, the facility has treated each order independently, and assigned a customized ingot for each order, thus minimizing the total scrap during the process. But over the past few years, the company has upgraded its rolling facility and can now produce large size ingots in order to exploit economies of scale at the ingot casting stage. On the other hand, with an increase in the emphasis for just-in-time production, customers are ordering products more frequently, and in smaller quantities. In this situation, the previous practice of dedicating an ingot to a single order is uneconomical, since the facility could very well be using a 10,000 pound ingot for a customer order of 4,000 pounds. The remaining 6,000 pounds have to be either scrapped or stored in intermediate inventory. As a result, the facility would have to incur additional cost in scrap reprocessing or holding intermediate inventory.

In this situation, the plant is considering an alternate strategy of combining more than one order for production on a single ingot (Ventola [1992] and Gopalan [1992]). This strategy allows multiple orders to be combined on a single ingot, and allows the plant to exploit its production capabilities while reducing excess scrap or intermediate inventory.

Each order specifies the alloy, temper, width, thickness or gauge, and weight of the sheet product that the customer requires. Different orders can be combined within certain processing limits. Gopalan [1992] has shown that combining order can increase profits by hundreds of thousands of dollars per annum for just a single alloy type.

This thesis addresses the following tactical planning decision: when we permit a facility to jointly produce multiple orders from the same ingot, what standard ingot sizes should the facility produce, and which orders should be combined to optimize performance? Selecting "good" ingot sizes is important for several reasons. The facility makes ingots to stock, and satisfies customer orders from the stocked ingots. Changing ingot sizes requires considerable capital investment, and hence cannot be done frequently. Moreover, maintaining a large number of ingot sizes in stock increases inventory and material handling costs, and creates logistical problems (Vasko et al. [1989]). Hence, the facility can only maintain a small number (relative to the number of orders received) of ingot sizes in stock. Simulation experiments by Gopalan [1992] show that economic benefit from order combination is very sensitive to the ingot sizes.

1.2 Outline of Thesis

The objective of this thesis is to model the **ingot sizing problem** and develop an efficient solution procedure for this problem. The input to the problem is the physical characteristics and forecast demand over the planning horizon for each product type, a limited set of candidate ingot sizes, and the set of feasible product combinations. Given this data, we need to choose a set of a prespecified number of ingots and determine the optimal combination of orders to minimize total production and scrap reprocessing cost, while satisfying demand for all products.

We formulate the ingot sizing problem as an integer program. Since all costs are proportional to the weight of the ingot used, we minimize the total weight of all ingots used to satisfy demand for all products. The desired output is the set of standard ingot sizes and the optimal order combinations. The assumptions and approximations we make are explained in detail in chapter 2. We also discuss two special cases of the ingot sizing problem.

We have developed an efficient solution procedure using a dual ascent method and some heuristics. We use the dual ascent procedure to generate a lower bound as well as a heuristic solution for the problem. We have also developed two stand-alone heuristics to obtain good feasible solutions for the problem. Both the heuristics are greedy, and select ingots based on either the total order weight covered by an ingot, or the weight of order combinations. We have implemented the dual ascent procedure and heuristics, and tested them using data on actual orders received, and actual processing constraints at a leading aluminum sheet manufacturing company. We also use our model to perform sensitivity analyses related to width constraints, and the number of standard ingot sizes allowed.

This thesis focuses on an important practical problem facing an aluminum rolling facility. The problem addresses the issue of ingot sizing with order combination. The current literature either focuses on the sizing problem or the order combination problem. We have developed and tested an efficient solution procedure for solving the problem. Our computational results indicate that the solution procedure is quite effective (within 4% of optimality on an average) and, that the set of ingots suggested by the solution procedure reduces the total scrap by an average of 9.5% over the current set of ingots used by the manufacturing facility. For the alloy that we studied, the total reduction in scrap could result in savings of up to $100,000 annually.

1.3 Organization of Thesis

The remainder of the thesis is organized as follows. Chapter 2 describes the manufacturing process and the order combination process in detail. This chapter defines the ingot sizing problem, develops a mathematical formulation of the problem, and discusses a few interesting special cases. We also present a review of relevant literature in this chapter. Chapter 3 describes the dual ascent procedure and the heuristics in detail. Chapter 4 describes the input data analysis, and reports the computational results for the dual ascent and heuristic procedures. Finally, Chapter 5 provides conclusions and directions for future work.

Chapter 2

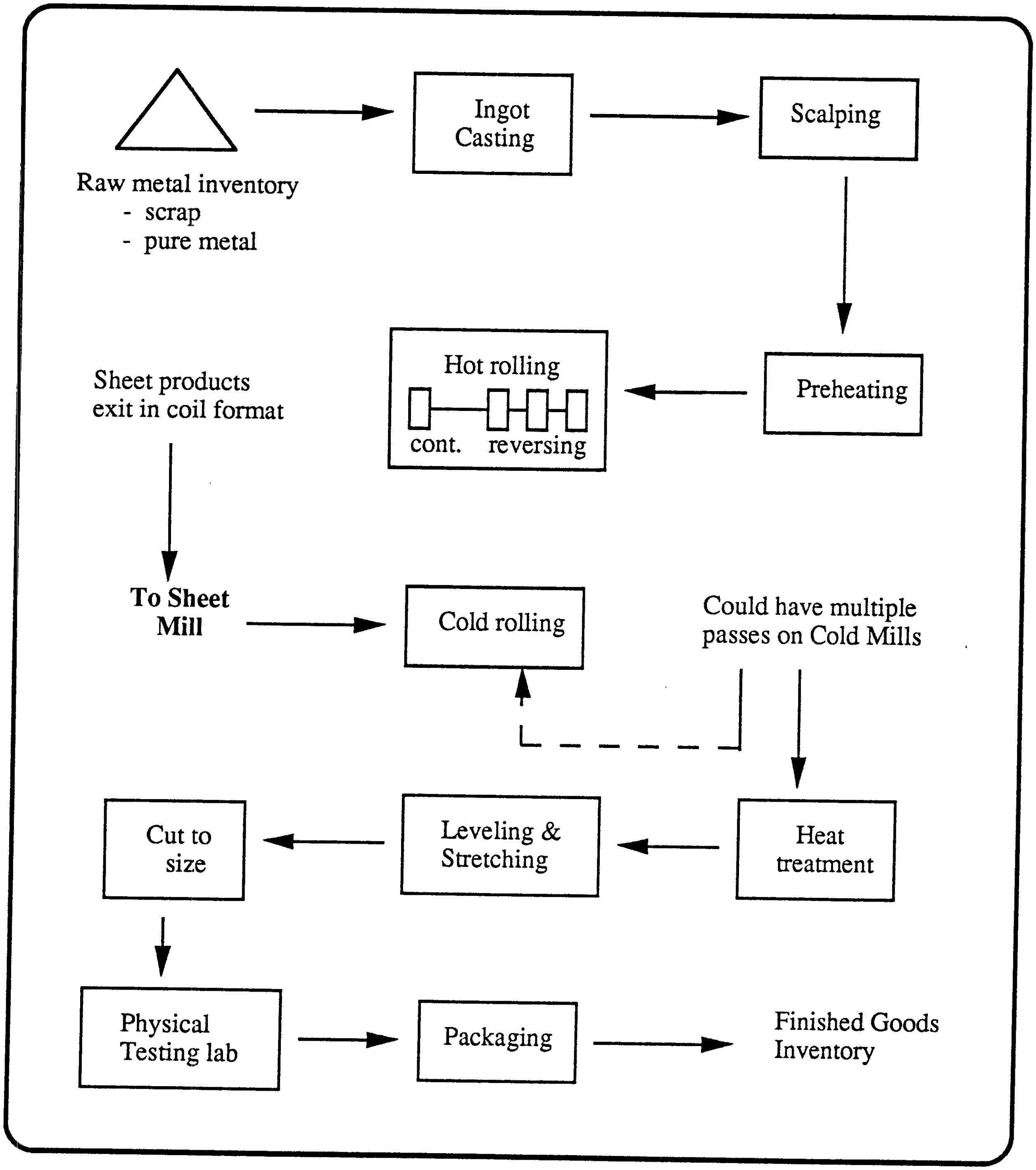
Problem Definition and Formulation

In this chapter, we first describe the aluminum sheet manufacturing process and the processing constraints of order combination. We then present the ingot sizing problem description and the modeling assumptions. Next, we develop a mathematical model of the problem, and discuss a few special cases of the problem. Finally, we discuss the relevant literature.

2.1 Process Description

This section describes the sheet manufacturing process at the aluminum rolling facility we studied. The sheet manufacturing process consists of five main stages: ingot casting, hot rolling, cold rolling, heat treatment, and finishing operations. Figure 1 depicts the various stages in the process flow. In the first stage, aluminum in the form of pure metal and scrap is cast into rectangular ingots. In this facility, ingot casting is a make-to-stock production process. The cast ingots are then "scalped" to provide a smooth uniform surface for the rolling operation. During the scalping process, a fixed depth of aluminum is removed off the top and bottom faces of the ingot. The scalped ingots are heated to the temperature required for the hot rolling operation. The hot rolling station consists of several rolling mills in series, that successively reduce the thickness of the ingot. The ingot comes off the hot rolling mills as a coiled sheet. The hot rolling operation can produce large reduction in the thickness of an ingot, but cannot maintain tight dimensional

Figure 1. Sheet Manufacturing Process



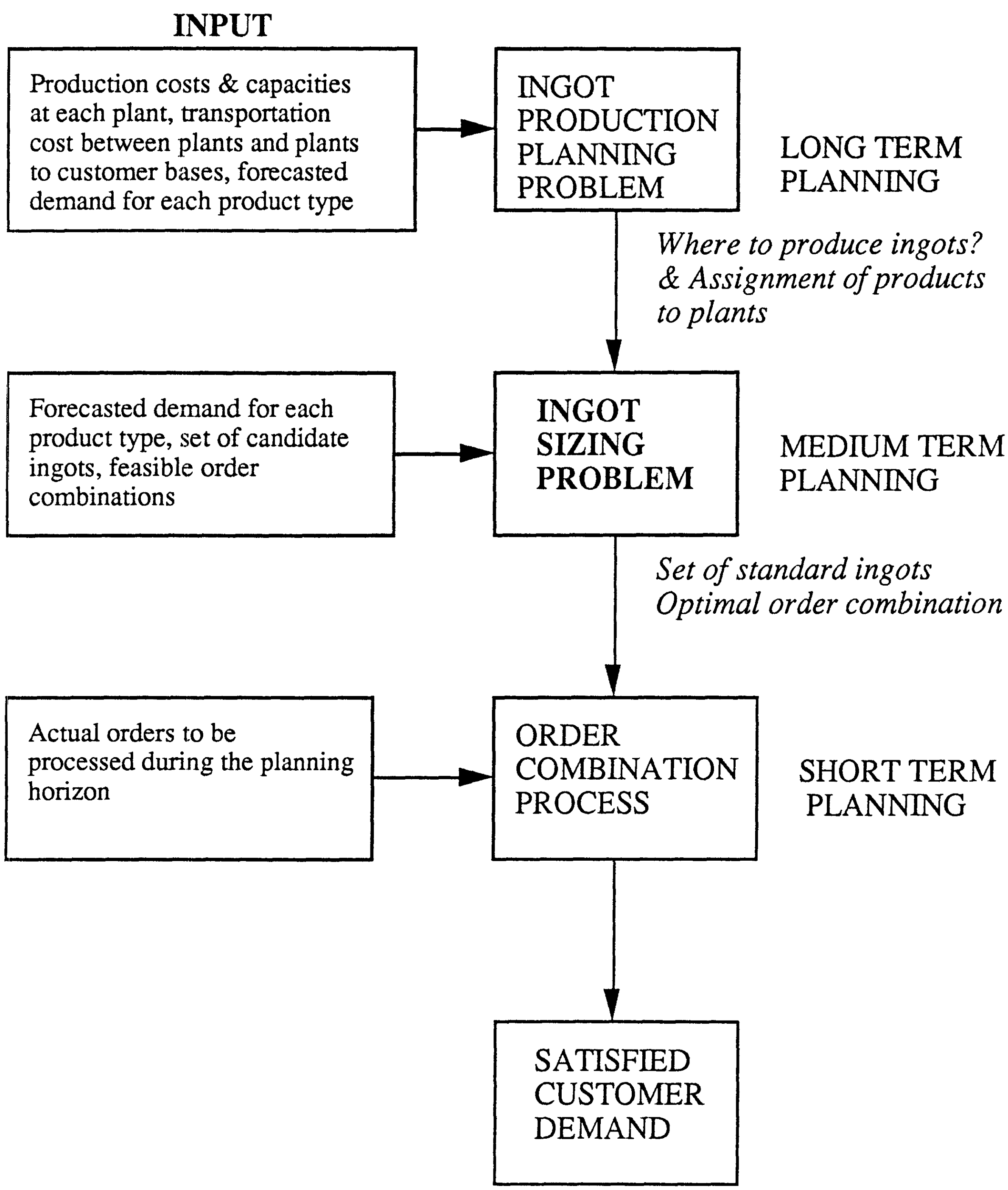
tolerances. So, the sheet of metal next goes through a cold rolling operation, that further reduces the thickness of the product. Some of the cold mills can change gauge on the fly within certain ranges. On these cold mills, we can dynamically adjust the spacing between the rollers while processing a coil to produce sheets with different gauges (within certain ranges) from the same coil. This allows us to process two orders as a single one until the last phase of the cold rolling stage, and is one of the processing flexibilities which makes order combination a feasible strategy. Cold rolling is followed by heat treatment and finishing operations (Balakrishnan, [1993]).

2.2 Problem Context

Figure 2 shows the hierarchy of decisions involved in production planning for metal sheet manufacturing. When a sheet manufacturing company has more than one plant where it can make ingots and final products, it must decide how to allocate ingot and sheet production to various plants to utilize capacities effectively while meeting customer requirements at minimum total production and transportation cost. Therefore, at the long term planning stage, we would decide which plants would produce what size of ingots, given the production costs and capacities at the various plants, the forecasted customer demands, and transportation costs between plants, and between customers and plants. The goal is to minimize total production and distribution costs, and the decision serves as an input to the medium term planning problem.

Given the long term decisions for each plant that produces ingots, we have the set of products whose demands must be satisfied from the ingots in stock at that plant. Given the forecasted demand for these products, and the set of available candidate ingot sizes, the medium term planning problem decides the standard ingot sizes to stock, assuming that the facility can produce more than one order using a single ingot. The objective at this stage

Figure 2. Hierarchy of Decisions



is to minimize the total production and scrap processing cost. For the short term planning problem, we have the actual set of orders to be processed during the planning horizon and the standard ingots. We must decide which specific orders to combine to optimize system performance (minimize total cost or maximize revenue from satisfied demand).

We focus on the medium term ingot sizing problem, assuming that multiple orders can be jointly produced from the same ingot. We first describe the constraints and requirements for combining two orders for production on the same ingot in Section 2.3, and then define the ingot sizing problem and discuss the assumptions in Section 2.4.

2.3 Order Combination Process

Planners at the facility that we studied indicated that combining more than two or three orders on an ingot requires many special instructions to operators, and poses challenging operational problems (Balakrishnan, [1993]). Moreover, combining more than two or three orders on an ingot increases the number of orders that need expediting, if the entire ingot has to be scrapped due to defects. Hence, we assume that at most two orders can be combined on an ingot. Given the set of orders, not all pairs of orders can be combined. They must meet certain processing constraints that limit the maximum differences in gauges, and we must be able to process them as a single job until the final phase of cold rolling. The processing path for each combination describes the various steps in the actual processing of that combination - the ingot used, the amount of reduction at the hot line, the number of passes at the cold mills, and finishing operation specifications. Order combination tries to group orders that share a common processing path until the final pass at the cold mill, and require the same alloy.

During the final pass at the cold mill, the difference in gauges achievable, by changing the spacing between the rollers, is limited. Hence, we can only combine orders whose gauges are compatible, i.e., the minimum and maximum gauge in a combination must not differ by more than a prespecified value. This maximum gauge differential depends on the final finished gauges of the combined orders. For each alloy, the facility has determined a set of intervals of gauges which can be used as a guideline to combine orders. Table 1 shows a representative for an alloy that we studied. The table has nine different overlapping intervals, each of which has its own characteristic processing path. All orders on a single ingot must have gauges that are in one of the nine intervals, and the width of each order must be less than or equal to the width specified in that interval.

We pick the order with the minimum gauge and identify the intervals into which this gauge falls. If the gauge of the other order is less than or equal to the thickest order gauge requirement on one of the intervals, and both orders have width less than or equal to the maximum cold mill entry width for that interval, then we can combine them. Suppose we have an order of width 48 inches and gauge 0.080 and another order of width 54 inches and gauge 0.250. We pick the thinner order and see that it satisfies gauge and width requirements on intervals 2, 4, and 5. Now the thicker order does not satisfy gauge requirement of intervals 2 and 5, but satisfies the requirements of interval 4. Hence, we can combine the two orders.

A combination that satisfies the maximum gauge differential is feasible only if it can be produced on one of the available ingots. We refer to the gross ingot weight minus planned and unplanned scrap as the final recovered weight. A pair of orders can be produced using an ingot if the total weight of the two orders is less than or equal to the weight of the ingot after planned and unplanned scrap. The planned scrap consists of the following: a certain percentage of the total weight removed during scalping, a fixed depth (a few inches) along the length of the ingot on either side (side trim), and a fixed amount along the width of ingot (head and tail scrap). Given the set of available ingot sizes, for each possible combination of orders, we can determine the set of ingots that can process the combination. If none of the available ingots can process the combination, the combination is not feasible. Figure 3 shows the various steps involved in the order combination process.

For each feasible combination, the planner chooses an ingot whose width exceeds the width of the order (including the necessary side trim), and whose recovered weight after scalping and head and tail scrap removal is greater than or equal to the weight of the order. The planner then decides the hot line exit gauge, which helps decide the reduction required at each hot rolling station. He finally determines the number of cold mill passes required to achieve the final required gauge, and assigns the combination for processing. For a detailed discussion of the costs and benefits of order combination, see Ventola [1991] and Balakrishnan and Brown [1992].

2.4 Problem Definition and Assumptions

We have focused thus far on the short term planning problem and described the various aspects of order combination. We will now define the higher level planning decision of choosing the standard ingots and the optimal order combinations.

2.4.1 Problem Definition

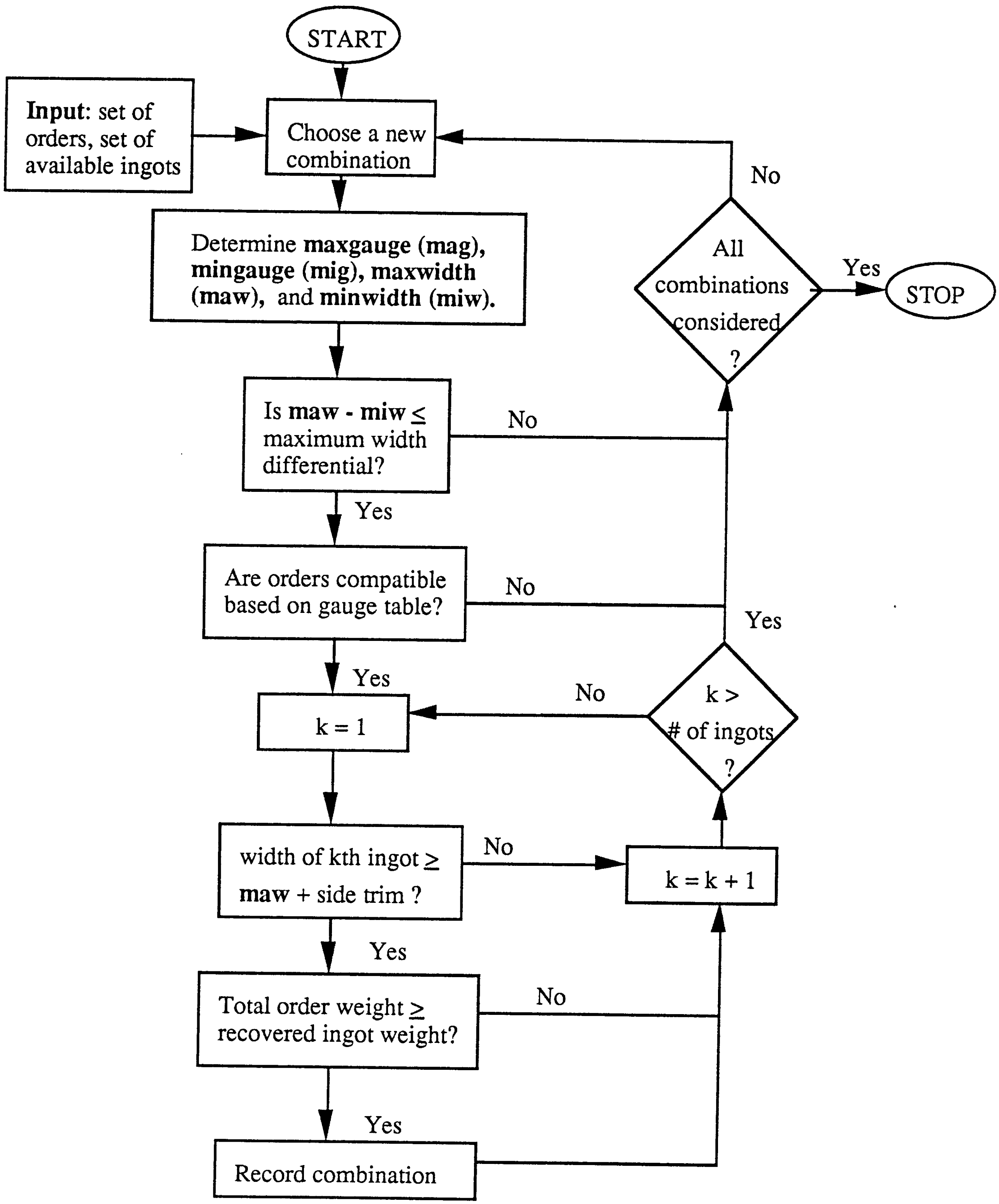
Customers place orders for sheets of a particular alloy, temper, gauge, width, and weight. Similar orders might be placed several times over a long planning horizon, say one year. We define a product as a collection of similar orders. Thus each product is

Table 1. Gauge Combination Table

Processing path	Finish gauge for thinnest order ≥	<	Maximum cold mill entry width	Finish gauge for thickest order ≤
1	0.039*	0.071	60	0.229
2	0.071	0.082	60	0.229
3	0.082	0.229	60	0.229
4	0.071	0.114	75	0.257
5	0.071	0.157	75	0.214
6	0.157	0.214	75	0.286
7	0.214	0.286	75	0.357
8	0.286	0.357	75	0.450
9	0.357	0.500	75	0.500

* The numbers have been disguised to preserve confidentially of data.

Figure 3. Rules for the Order Combination Process



characterized by its temper, gauge, width, weight, and frequency. The frequency of the product corresponds to the number of times the product is demanded over the year or equivalently, the number of orders grouped under the product type. In order to define the **ingot sizing problem** formally, we first describe the inputs to the model. We are given the following input data:

- Forecast demand for each product over the planning horizon.
- The set of candidate ingot sizes. Each ingot size is characterized by its weight, width, and length. We consider only the width and weight of the ingots. We assume that the planner specifies a discrete set of candidate sizes in the range of possible sizes that the ingot plant can produce. The actual choice of candidate sizes depends on the width and demand of the products. We need to choose a set of candidate sizes that can produces all the products. We choose candidate widths and weights in proportion to the widths and weights of the products.
- The set of feasible order combinations, based on order combination rules and constraints described in the Section 2.3. We assume that at most two orders can be combined on an ingot.
- The maximum number of standard sizes, p, that we can choose. Several factors such as, storage capacity, ease of tracking inventory, scrap reprocessing costs, and inventory costs play a role in determining this number. The higher the value for p, the lower the scrap will be. However, an increase in the number of stock sizes results in higher inventory costs and more detailed inventory tracking systems. We do not incorporate the inventory costs in our model.

The ingot sizing problem chooses a subset of p or less candidate ingots as standard ingots, and the number of times a product combination is produced on a standard ingot, to satisfy demand for all products at the minimum total processing and scrap reprocessing costs. The processing cost is the sum of the operating cost (equipment and labor) at each station for all the processed ingots. The scrap reprocessing cost consists of the ingot casting and melting cost for the total amount of scrap. Since all the costs that we use here are proportional to the number of pounds of metal rolled, we use the total weight of the ingots used to satisfy demand, as a surrogate for the costs.

The ingot sizing decision decomposes by alloy, since finished products of a particular alloy have to be produced from an ingot of the same alloy. Only certain tempers of an alloy can be combined since they have similar processing paths up to the final pass of cold rolling. Thus, we have to solve the ingot sizing problem for each alloy and group of tempers that can be combined together.

2.4.2 Modeling Assumptions

The input data for the ingot sizing problem consists of forecast demand for the products. For a particular realization of demand, two products that must be combined might not occur simultaneously. For example, the solution to the ingot sizing problem might suggest that product A (frequency = 2) and product B (frequency = 2) must be combined twice. However, if demand for product A occurs during months 1 and 2 and for product B during months 4 and 5, then we cannot combine them. In this case, we would have to choose an alternate combination. Hence, there is some loss of generality in not considering the due dates explicitly for each order in a product. However, if product frequencies are relatively high, then we can assume that two products that have to be combined, occur together most of the time.

We have mentioned that during the order combination process, orders can be combined only based on the limits given by Table 1. Also, it is advantageous to combine orders of similar widths, since this would minimize scrap. Though the objective function would minimize trim loss, and hence combine orders whose widths are comparable, we explicitly limit the difference in the widths of orders combined to a pre-specified maximum width differential. This also reduces the number of feasible combinations, and the problem is easier to solve.

2.5 Model Formulation

The ingot sizing problem can be formulated as an integer program as follows. We first provide the required definitions.

I = set of products

K = set of candidate ingots

J(i) = set of all products with which product i can be combined

IJ(k) = set of all feasible combinations on ingot k

A combination is a pair (i,j) and without loss of generality, we assume that $i \leq j$ for all combinations. Combination (i,i) denotes producing two orders of product i. A combination can contain just one unit of a product, to allow dedication of an ingot to an order. In this case, we denote the combination as (i,0).

K(i,j) = set of ingots which can produce combination (i,j), $i \leq j$, i.e., set of ingots for which maximum width of the two products + side trim $\leq$ width of ingot, and total weight of products $\leq$ recovered weight of ingot.

Given the width, weight, and gauge of all the products, and the width and weight of the candidate ingots, we first determine the feasible combinations for each ingot using the order combination rules explained in Section 2.3. The actual parameters used by the ingot sizing model are:

w_k = weight of ingot k,

f_i = frequency (total demand) of product i (as number of orders for product i),

p = maximum number of standard ingots, and

λ_{ij} = $\min\left\{f_i, f_j\right\}$.

The decision variables are:

y_{ijk} = number of times the combination (i,j) will be produced during the year using ingot k, and

$$z_k = \begin{cases} 1 & \text{if ingot k is chosen, and} \\ 0 & \text{otherwise.} \end{cases}$$

We want to minimize the total weight of the ingots used to satisfy demand for all the orders. The ingot sizing model can be formulated as follows:

$$\textbf{(ISP)} \qquad \text{Min} \sum_{k \in K} \sum_{(i,j) \in IJ(k)} w_k\, y_{ijk} \tag{2.1}$$

subject to :

Demand constraints

$$2 * \sum_{k \in K(i,i)} y_{iik} + \sum_{j \in J(i)} \sum_{k \in K(i,j)} y_{ijk} \geq f_i \qquad \text{for all } i \in I \tag{2.2}$$

Forcing constraints

$$y_{ijk} \leq \lambda_{ij}\, z_k \quad \text{for all } (i,j) \in IJ(k),\ k \in K \tag{2.3}$$

p-median constraint

$$\sum_{k \in K} z_k \leq p \tag{2.4}$$

Integrality constraints

$$y_{ijk} \in \{0,1,2,\ldots\} \text{ for all } (i,j) \in IJ(k),$$
$$z_k \in \{0,1\} \qquad k \in K \tag{2.5}$$

The objective function (2.1) is the total weight of all the ingots used to satisfy the total demand for all orders. Constraint (2.2) requires that the demand for each product be completely satisfied. The forcing constraint (2.3) ensures that two products i, j are combined on an ingot k, only if ingot k is a standard ingot, and the number of times the two products are combined is less than or equal to the minimum of the demands of the two products. When we combine products i and j, we have to make sure that we do not combine them more often than necessary. The p-median constraint (2.4) restricts the number of chosen ingots to less than or equal to the prespecified value p. We can vary the value of p parametrically to determine the number of standard ingots to produce, to minimize scrap and inventory costs. The higher the value for p, the lower the scrap will be. However, an increase in the number of stock sizes results in higher inventory costs. We do not incorporate the inventory costs in our model. But we can solve the ISP for different values of p, and let the planner select the best value of p by weighing the reduced scrap against the increase in inventory cost and inventory tracking efforts.

We next show that when we dedicate ingots to orders, [ISP] reduces to the p-median location problem which is NP-complete (Garey and Johnson [1979]). We also show that when the set of standard sizes are given, the problem of determining the optimal product combinations can be solved as a non-bipartite matching problem.

2.5.1 Special Cases

Ingots Dedicated to Orders

If we allow only one order per ingot, then the ingot sizing problem reduces to a p-median location problem. In this case we always satisfy demand for a product from the same ingot, whereas in (ISP) we might satisfy demand for a product using more than one ingot. For this special case, we would incorporate the frequency of the orders into the objective function, and determine the assignment of orders to the subset of selected ingots. The model for the special case can be represented as follows. Let

$\mathbf{K(i)}$ = set of ingots that can satisfy demand for product i

$C_{ik} = f_i \, w_k$

The decision variable y_{ik} is defined as

$$y_{ik} = \begin{cases} 1 & \text{if ingot k satisfies demand for product i} \\ 0 & \text{otherwise} \end{cases}$$

The remaining parameters and decision variables are as defined in [ISP]. The formulation of the special case is as follows.

(ISP-SC) $$\text{Min} \sum_{i \in I} \sum_{k \in K(i)} C_{ik} \, y_{ik} \qquad (2.6)$$

subject to :

Assignment constraints

$$\sum_{k \in K(i)} y_{ik} = 1 \qquad \text{for all } i \in I \qquad (2.7)$$

Forcing constraints

$$y_{ik} \leq z_k \qquad \text{for all } i \in I, \; k \in K(i) \qquad (2.8)$$

p-median constraint

$$\sum_{k \in K} z_k \leq p \qquad (2.9)$$

Integrality constraints

$$z_k \in \{0, 1\} \qquad \text{for all } i \in I,$$
$$y_{ik} \in \{0, 1\} \qquad k \in K(i) \qquad (2.10)$$

This is a standard p-median location problem and to solve this problem, we can use any of the specialized p-median algorithms. (See Mirchandani and Francis [1990] for a review of p-median solution algorithms.)

Determining Optimal Combinations

When we are given the set of standard ingots to be maintained in stock, then the determination of the y_{ijk} values can be solved as a non-bipartite matching problem (Papadimitriou and Steiglitz, [1982]). Note that this is a special of [ISP], where the number of candidate ingot sizes is equal to the number of required standard sizes. Given the set of standard ingots, we can determine a priori the ingot to be used for each possible combination: this is the lowest weight feasible ingot. We can also determine if combining an order with itself (if it is feasible) is more economical than assigning a dedicated ingot to each of the two orders. Given these cost parameters, we can the transform to the matching problem as follows. We refer to the example in Figure 4 to explain this transformation. For each product i with frequency f_i, we create f_i nodes in the graph. If products i and j can be combined at a cost of w_k, then from each node corresponding to product i, we add an edge to each node corresponding to product j, at a cost of w_k. For example, in Figure 4, we create three nodes corresponding to product 1 which has a frequency of three, and two nodes for product 2. Since combination (1,2) is feasible and the cost is w_2, we add two edges from each node of product 1 to the nodes corresponding to product 2 with a cost of w_2.

For all products with f_i equal to 1, we add a dummy node with a cost of the corresponding edge equal to the cost of dedicating an ingot to product i. If the frequency of product i is greater than 1, and it is more economical to combine two orders of this product than dedicating it to an ingot, then we just add one dummy node. On the other hand, if it is better to dedicate ingots to product i, we add f_i dummy nodes and connect one to each one of the nodes corresponding to order i. Note that in the example one dummy node for product 1, since combining two orders is better than dedication. On the other hand, we add two different dummy nodes for product 2. All the dummy nodes are connected with each other at a cost of 0. A minimum cost matching on this non-bipartite graph produces a solution to (ISP) when the standard sizes are given. We use this transformation to solve the sub problem of determining the y_{ijk} values in our solution procedure. An arc (i,j) in the matching solution indicates that we combine the product corresponding to node i and the product corresponding to node j once.

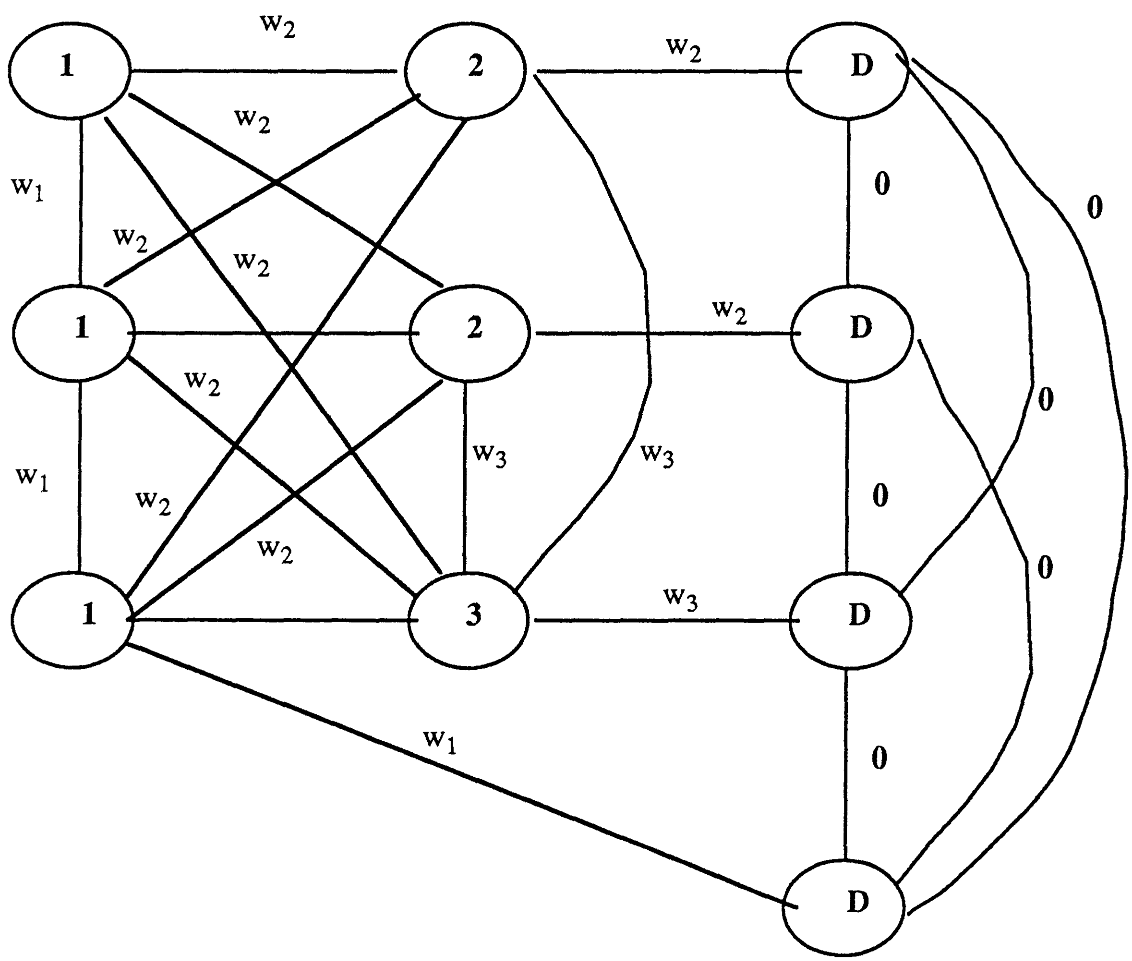
We can reduce the number of arcs in the graph by making the following simplification. Assume order i can be combined with orders j and k. We sort orders j and k in ascending order of their frequencies. Let j be the first node and k the second in the sorted list. Then each node corresponding to order j is connected to min $\{ f_i, f_j \}$ nodes corresponding to order i, and each node of order k is connected to min $\{ f_i, f_j + f_k \}$ nodes corresponding to order i. This simplification will be useful when we have a few orders with very large frequencies and the remaining orders with relatively smaller frequencies.

2.6 Related Literature

We have described the ingot sizing problem and formulated it as an integer program. Now, we discuss some of the relevant literature and the unique features of the ingot sizing problem. We first present an actual application in a steel manufacturing facility. We then describe two closely related problems and, the similarities and differences between these problems and the ingot sizing problem. Vasko et al. [1986, 1987, 1989] describe an actual application of a set covering approach for choosing an optimal set of ingot sizes for Bethlehem steel. Although developed for the steel industry, their model

Figure 4. Matching Transformation

Products:	1	2	3				
Frequency:	3	2	1				
Combining 2 orders of product better than dedication?	Yes	No	--				
Feasible Combinations:	(1,1)	(1,2)	(1,3)	(2,3)	(1,0)	(2,0)	(3,0)
Cost:	w_1	w_2	w_2	w_3	w_1	w_2	w_3



shares with our model features such as generating feasible assignments of orders to ingots based upon processing constraints. The main difference is that they allow only one order on an ingot, and do not consider frequencies of the products. In other words, frequency for all products is 1. Thus their model is a special case of the ingot sizing problem. They do not explicitly impose the restriction of selecting at most p standard sizes. Given a set of products, and set of ingots, they determine the set of standard ingots and the assignment of products to ingots, to minimize the number of standard sizes, and the total yield loss from assigning orders to ingots as a secondary objective. They use a set covering heuristic to solve the problem and report that the results have produced millions of dollars in savings for the company. They define "inflexible" orders as orders that can be satisfied by only a few ingots. The heuristic initially selects ingots to cover as many inflexible orders as possible, and then switches to selecting ingots that can cover as many orders as possible. Further, they also use neighborhood search to locally improve the solution.

The **cutting stock** problem, which is common in glass, paper, and steel bar manufacturing, is similar to the ingot sizing problem in some ways. In the cutting stock problem, large sheets or rolls are maintained in intermediate inventory and they are cut to size to satisfy customer demands. Here, all the final products have the same quality and differ only in their dimensions. So the problem reduces to determining the best patterns to use to minimize scrap. In the ingot sizing problem, in addition to combining final products of varying quality (such as temper), we have the additional decision of selecting the optimal stock sizes. The generation of feasible combinations is similar to the generation of feasible patterns for the cutting stock problem.

Gilmore and Gomory [1961, 1963] have used a column generation approach to solve the linear programming formulation of the one dimensional cutting stock problem. They have also presented a solution approach for the two dimensional problem which is based on the one-dimensional technique (Gilmore and Gomory, [1965]). Many researches have addressed the single and multi-dimensional cutting stock problems. Most of the work deals with determining optimal cutting patterns for a given set of stock sizes. (Christofides and Whitlock [1977], Goulimis [1990], Stadler [1990], Wang [1983]). A few articles (Chambers and Dyson [1976], Beasley [1985], and Farley [1990]) examine the question of optimal dimensions for the fixed stocked sizes, and the optimum number of stocked sizes. Chambers and Dyson [1976] address a two-dimensional cutting stock problem with stock size selection. They develop a two stage heuristic algorithm, where they first decide the single best width to stock during the first stage. In the second stage, they determine the k best lengths for the selected width. Beasley [1985] presents a two stage heuristic algorithm for the deciding the best stock sizes of rectangular plates and the best cutting patterns for each of these plates. These papers consider frequency of the products, but the solution procedure is different from the solution procedure for ISP. Beasley [1985] determines the cutting patterns first, and selects the stock sizes which are used the maximum number of times in the cutting patterns. Dyson [1976] allow only one width for the set of standard sizes.

The **assortment** problem deals with selecting the best sizes to stock from a give set of sizes, in order to satisfy demand for all products at a minimum cost. In this problem, only one final product is obtained from a standard size. So this model does not deal with the order combination issues addresses in our ingot sizing problem. Given a set of products with known demands, Wolfson [1965] uses a dynamic programming approach to select the best lengths to stock to minimize scrap costs incurred while satisfying demand for sizes not in stock from the standard sizes. An important property of the solution which facilitates the use of dynamic programming is that all products (including the stocked sizes) can be ordered based on the single dimension with which we are dealing, and demand for any unstocked size is satisfied from the closest stocked size. This property does not hold for the ingot sizing problem since we allow one unit of the stocked size to satisfy demand for more than one product. Pentico [1976] extends the work of Wolfson to the case with probabilistic demand for the products. Pentico [1988] has also developed heuristic procedures for the two dimensional assortment problem. In the two dimensional problem also, one unit of a stocked size can satisfy demand for only one unit of any product. We once again highlight the fact that while the problems presented here have a lot of similarities to the ingot sizing problem, they also have distinct differences, which makes it worthwhile to study the ingot sizing problem.

In this chapter, we have provided a description of the metal sheet production process and order combination. Next we defined the ingot sizing problem and stated the assumptions of our model. We have also developed a mathematical formulation of the ingot sizing problem and presented two special cases of the problem. Finally, we have discussed some of the relevant literature and highlighted the similarities and differences to the ingot sizing problem.

Chapter 3

Dual Ascent Procedure and Heuristics

This chapter describes a solution procedure for the ingot sizing problem. The solution procedure that we have developed uses a combination of dual ascent to generate lower bounds and heuristic methods to generate good solutions to the problem. We have developed an optimization-based approach that heuristically solves the dual problem and generates lower bounds and heuristic solutions. We next describe the dual ascent procedure.

3.1 Dual Ascent Procedure

In order to generate lower bounds, we approximately solve the dual of the linear programming relaxation of [ISP] using dual ascent. We use dual variable u_i for the demand constraint (2.2) of [ISP], dual variables v_{ijk} for the forcing constraints (2.3), and the variable α for the p-median constraint (2.5). The dual problem can be written as follows:

[DISP]

$$\text{Max} \sum_{i \in I-\{0\}} f_i u_i - p\alpha \tag{3.1}$$

subject to:

$$u_i + u_j - v_{ijk} \quad \leq \quad w_k \qquad \text{for all } (i,j) \in IJ(k),\ k \in K \tag{3.2}$$

$$\sum_{(i,j) \in IJ(k)} \lambda_{ij} v_{ijk} - \alpha \quad \leq \quad 0 \qquad \text{for all } k \in K, \text{ and} \qquad (3.3)$$

$$u_i \geq 0, v_{ijk} \geq 0, \alpha \geq 0 \qquad \text{for all } (i,j) \in IJ(k), \; k \in K \qquad (3.4)$$

The set IJ(k) includes combinations (i,i) also. For combinations of type (i,i), equation (3.2) reduces to

$$2\, u_i - v_{iik} \quad \leq \quad w_k \qquad \text{for all } (i,i) \in IJ(k), \; k \in K \qquad (3.5)$$

3.1.1 Lower Bound for [ISP]

The objective value of [DISP] is a lower bound for the ingot sizing problem. We start with an initial solution for the dual problem and try to improve it iteratively, to obtain a good lower bound. We next describe the method to construct the initial solution to the dual problem.

Initialization

We construct the initial dual solution using the following greedy method. We start with $v_{ijk} = 0$ for all $(i, j) \in IJ(k)$, and $k \in K$, and $\alpha = 0$. We note that $u_i = 0$ for all $i \in I$ is a feasible dual solution, but we want to start with better starting values for the u_i variables. Having fixed the value of the v_{ijk} variables at 0, inequalities (3.2) reduce to:

$$u_i + u_j \quad \leq \quad w_k \qquad \text{for all } (i, j) \in IJ(k), \; k \in K \qquad (3.6)$$

For each feasible combination (i,j), define

$$\delta_{ij} \quad = \quad \min_{k \in IJ(k)} [w_k] \qquad (3.7)$$

i.e., α_{ij} is the minimum weight ingot on which combination (i,j) is feasible. Then, inequalities (3.5) reduce to

$$u_i + u_j \quad \leq \quad \delta_{ij} \qquad \text{for all orders (i,j) that can be combined} \tag{3.8}$$

The initialization procedure is greedy and tries to increase u_i as much as possible in each iteration. Starting with $u_i = 0$ for all products, we iteratively increase one u value at a time using the following procedure. We first increase the dual variable u_i corresponding to the product i with the maximum frequency f_i, since this variable has the maximum dual objective function coefficient. We set this variable equal to its possible maximum value by checking inequalities (3.8) which contain this dual variable. Having fixed the first variable, we pick the dual variable corresponding to the second most frequent product and repeat the procedure. The procedure can be expressed formally as follows.

Step 1: Index the products from 1 to n in the descending order of their frequencies, i.e., $f_1 \geq f_2 \geq \ldots \geq f_n$. Let $(u_1, u_2, \ldots, u_n)$ be the dual variables corresponding to the indexed list of products. Initially, $u_i = 0$ for all i, and the objective value = 0.

Step 2: Set m = 1.
For all $j \in J(m)$, we have constraints of the type $u_m \leq \delta_{mj} - u_j$. If the combination (m,m) is feasible on some ingot, then we have an additional constraint of the type $2u_m \leq \delta_{mm}$. Note that δ_{mj} is the minimum weight ingot on which combination (m,j) is feasible. Now, set

$$u_m \quad = \quad \text{Min}\left[\min_{j \in J(m)}\left[\delta_{mj} - u_j\right], \frac{\delta_{mm}}{2}\right].$$

Step 3: Increase the objective function value by $f_m u_m$. Set $m = m+1$, and go to step 2 if m is less than or equal to n.

The initial lower bound is the objective value of the dual with the initial values of the variables. Next, we try to increase the lower bound using dual ascent techniques.

3.1.2 Dual Ascent Techniques

Dual ascent refers to a broad class of heuristic strategies to iteratively change the values of the dual multipliers in an effort to monotonically improve the dual lower bound. Several authors have successfully used this technique to obtain very good bounds for hard problems. Erlenkotter [1978] has applied dual ascent to the uncapacitated facility location problem, and Fisher and Kedia [1990] have applied it to the set packing problem. Two other successful applications of dual ascent include Balakrishnan, Magnanti, and Wong [1989] for the network design problem, and Wong [1984] for the Steiner tree problem. We use two heuristic adjustment techniques to increase the lower bound. The first technique attempts to increase the value of one v_{ijk} at a time, and the second increases the values of two v_{ijk} simultaneously, if possible.

Increasing one dual variable at a time

We now describe the first procedure to increase the dual objective by increasing one v_{ijk} at a time. Increasing v_{ijk} permits us to increase u-values and hence the dual objective value; but α might also need to increase to satisfy constraints (3.3), reducing the objective value. So, we must judiciously select the v_{ijk} value that produces a net increase in the objective function value. In this procedure, we pick a v_{ijk} variable to increase, and calculate the net change in the objective function if we increase the v_{ijk} variable by Δ units. We evaluate this change for all v_{ijk} variables, and increase the variable that results in the maximum increase in the objective function. The changes in the v_{ijk} variables are directly related to changes in the u_i variables, and hence we develop a procedure to keep track of the changes and determine the best candidate for increase.

We can potentially increase the value of the objective function only if we increase a v_{ijk} on a tight constraint (3.2). When we increase v_{ijk} on a tight constraint (3.2), we have to increase either u_i or u_j to maintain feasibility of the dual solution. This might increase the value of the objective function. On the other hand, if we increase v_{ijk} on a constraint (3.2) that is not tight, we do not affect feasibility of the dual solution and hence we do not have to increase the u-values. Hence, we try to increase v_{ijk} for some constraint (3.2) which is tight. We use the following set definitions to describe the procedure.

A = set of tight constraints in (3.2). This set consists of triplets (i,j,k).

B = set of tight constraint in (3.3)

T(i) = $\{j \mid (i,j,k) \in T1\}$, i.e., the set of products that can be combined with product i, and the constraint (3.2) for this combination is tight for some ingot k. T(i) does not contain i, if (i,i) is a feasible combination and (i,i,k) $\in$ T1 for some ingot k. This set consists of values for j.

NT(i) = $\{(j, k) \mid (i,j,k) \notin T1\}$, i.e., the set of products that can be combined with product i, and the constraint (3.2) for this combination is not tight for any ingot k. This set consists of pairs (j,k).

During the first phase of the dual ascent, we want to increase only one v_{ijk} variable at a time. So, we pick a feasible combination (i,j) for which only one constraint in (3.2) is tight. If we increase this v_{ijk} variable by Δ units, then can increase either u_i or u_j by Δ units. By increasing u_i or u_j by Δ units, we contribute $f_i\,\Delta$ or $f_j\,\Delta$ to the objective function. We would profit most by increasing the u_i corresponding to the product with the higher frequency among products i and j. Let us assume that $f_i > f_j$, and so we wish to increase u_i by Δ units. If i = j in the combination that we choose, then we can only increase u_i by $\Delta/2$ units.

Now, for every unit of increase in u_i, every $u_m \in T(i)$ must decrease by Δ units. If any u_m value is equal to 0, we cannot increase v_{ijk} without increasing v_{imk} also. But we might be able to increase u_j instead of u_i. If the value of any $u_m \in T(j)$ is also equal to 0, then we do not consider this v_{ijk} as a candidate for increase, since we are increasing only one v_{ijk} during this ascent phase. If on the other hand, if $u_m > 0$ for all $m \in T(i)$ or T(j), then we can increase v_{ijk}. WLOG, assume that we are increasing u_i. Thus, the net contribution from all the u values to the objective function, due an increase of v_{ijk} by Δ units is

$$\Delta_{ijk} = \left[f_i - \sum_{\substack{m \in T(i) \\ m \neq j}} f_m \right] \Delta \tag{3.9}$$

The dual variable we are increasing, v_{ijk}, corresponds to combination (i,j) on ingot k. If $k \in B$, i.e., the constraint (3.3) corresponding to this ingot is tight, then the increase in v_{ijk} will increase the value of α by $\min\{f_i, f_j\}\Delta$. This contributes to the net change in the objective function too. Hence, if $k \in B$,

$$\Delta_{ijk} = \left[f_i - \sum_{\substack{m \in T(i) \\ m \neq j}} f_m \right] \Delta - \lambda_{ij}\, p\Delta \tag{3.10}$$

If the net change, Δ_{ijk}, in the objective function given by (3.9) or (3.10) is greater than 0, then we can increase the lower bound by increasing v_{ijk}. If it is less than or equal to 0, then it is not advantageous to increase v_{ijk}. The actual amount of change will depend on several factors.

For every unit of increase of v_{ijk}, u_i has to increase by a unit also (or by half a unit if i = j in the combination). So the amount of change of v_{ijk} is indirectly controlled by the set of u values which have to change as a result. Hence, we have to examine all the constraints of (3.2) which contains u_i. If the constraint is not tight, then the change allowed by that constraint is equal to the value of the current slack. We first determine the maximum possible increase in u_i while considering the constraints in NT(i), as

$$\Delta_1 = \min_{(m,k) \in NT(i)} \left[w_k + v_{imk} - u_i - u_m \right] \tag{3.11}$$

If constraint (i,j,k) of (3.2) is tight, then the change allowed by that constraint is equal to the current value variable u_j. We define the maximum possible increase in u_i allowed by the constraints in T(i), as

$$\Delta_2 = \min_{m \in T(i)} \left[u_m \right] \tag{3.12}$$

If we are increasing v_{ijk}, and $k \notin B$, we can increase v_{ijk} such that the slack in constraint k of (3.3) reduces to 0. In other words, we do not increase the value of α while increasing v_{ijk}. This change is defined as

$$\Delta_3 = \left[\frac{\text{slack of constriant k of (3.3)}}{\lambda_{ij}} \right] \tag{3.13}$$

Finally, the increase in variable v_{ijk} is determined as

$$\Delta \quad = \quad \begin{cases} \min[\Delta_1, \Delta_2] & \text{if } k \in B \\ \min[\Delta_1, \Delta_2, \Delta_3] & \text{if } k \notin B \end{cases} \tag{3.14}$$

The objective function value increases by Δ_{ijk} as a result.

Considering each $v_{ijk} \in A$ as the variable to increase, we evaluate the final change in objective function and pick the (i,j,k) with the maximum change as the dual variable to be increased. At the end of an iteration of the first procedure, if we increased the objective function, then the set of tighr constraints might change. And this implies that we might be able to identify other dual variables corresponding to the new tight constraints, if any, which can be increased. So we update the values of the dual variables which changed, and the set of tight constraints, and repeat the procedure until no further improvement. When we cannot improve the objective function any more using this procedure, we try to increase two dual variables simultaneously. We summarize the first procedure for dual ascent formally below.

The sets A and B are defined as before. For every $(i,j,k) \in A$

Initialize: Pass = 0. WLOG, $f_i > f_j$, and we choose to increase u_i in the first pass.

Set l = i.

Step 1: Define the sets **T(l)** and **NT(l)** as:

T(l) = set of products that occur with product l, except the other product in the combination (i,j), in the all the tight constraints in (3.2)

NT(l) = set of products that occur with product i in the constraints that are not tight in (3.2). This set contains pairs (j, k).

Pass = Pass + 1

Step 2: If for any $(m,k) \in$ T(l), $u_m = 0$, then if Pass = 1, we can try to increase u_j. Let $l = j$, go to step 1.

if Pass = 2, go to initialization step and evaluate the next $(i,j,k) \in A$.

Else

We can increase u_l. WLOG, assume $l = i$.

If $k \in B$,

$$\Delta_{ijk} = \left[f_i - \sum_{\substack{m \in T(i) \\ m \neq j}} f_m \right] \Delta,$$

else,

$$\Delta_{ijk} = \left[f_i - \sum_{\substack{m \in T(i) \\ m \neq j}} f_m \right] \Delta - \lambda_{ij}\, p\Delta$$

Step 3: If $\Delta_{ijk} > 0$, then go to step 4.

Else,

if $u_m > 0$ for all $(m,k) \in T(j)$, set $l = j$ and go to step 1.

else go to initialization step and evaluate next $(i,j,k) \in A$.

Step 4: Determine the maximum allowable value of Δ.

$$\Delta_1 = \min_{(m,k) \in NT(i)} \left[w_k + v_{imk} - u_i - u_m \right]$$

$$\Delta_2 = \min_{m \in T(i)} \left[u_m \right]$$

$$\Delta_3 = \left[\frac{\text{slack of constriant k of (3.3)}}{\lambda_{ij}} \right]$$

$$\Delta = \begin{cases} \min[\Delta_1, \Delta_2] & \text{if } k \in B \\ \min[\Delta_1, \Delta_2, \Delta_3] & \text{if } k \notin B \end{cases}$$

Step 5: Repeat steps 1 through 4 for all $(i,j,k) \in A$, and pick the combination (i,j,k) with the maximum value of Δ_{ijk}. Let this be combination (i,j,k).

Update: $u_i \leftarrow u_i + \Delta$

$u_m \leftarrow u_m - \Delta$ for every $m \in T(i)$

$v_{ijk} \leftarrow v_{ijk} + \Delta$

If $k \in B$, then $\alpha \leftarrow \alpha + \lambda_{ij} \Delta$

Bound $\leftarrow$ Bound + Δ_{ijk}

Update sets A and B, and go to step 1.

Increasing two dual variables simultaneously

When we can no longer increase a single v_{ijk} variable, we try to increase two of them simultaneously. If there is a combination (i,j) which is feasible on more than one ingot, and exactly two of the constraints (3.2), say k and k', are tight for this combination, then we can increase the dual variables v_{ijk} and $v_{ijk'}$. As a result, we will still be increasing one of the dual variables u_i and decreasing one or more u variables. The rest of the procedure is similar to the first procedure, except that we have to consider two constraints of (3.3) when calculating the values of Δ and $\Delta_{ijkk'}$. When both constraints k and k' of (3.3) are tight, then we increase the value of the variable α by increasing v_{ijk} and $v_{ijk'}$, and we have to take this into consideration when calculating the change in the objective function due to an increase in the v_{ijk} variables. On the other hand, when neither the constraints k and k' of (3.3) are not tight, or atmost one constraint k or k' of (3.3) is not tight, we do not have to increase the value of α. We consider the slack in the constraints to determine the maximum amount by which we can increase the v_{ijk} variables in this case. The changes are shown in the formal description of the procedure below.

Let **A** = set of combinations (i,j) with exactly two tight constraints in (3.2). This set consists of (i, j, k, k').

B = set of tight constraint in (3.3).

For every (i, j, k, k') $\in$ A

Initialize: Pass = 0. WLOG, $f_i > f_j$, and we choose to increase u_i in the first pass.

Set l = i.

Step 1: Define the following sets.

T(l) = set of products that occur with product i, except product j, in the all the tight constraints in (3.2)

NT(l) = set of products that occur with product i in the constraints that are not tight in (3.2). This set contains pairs (j, k).

Pass = Pass + 1

Step 2: If for any $m \in T(l)$, $u_m = 0$, then

if Pass = 1, we can try to increase u_j. Let l = j, go to step 1.

if Pass = 2, go to initialization step and evaluate next $(i,j,k,k') \in A$.

Else

We can increase u_l. WLOG, assume l =i.

If both k and $k' \in B$,

$$\Delta_{ijkk'} = \left[f_i - \sum_{\substack{m \in T(i) \\ m \neq j}} f_m \right] \Delta,$$

else

$$\Delta_{ijkk'} = \left[f_i - \sum_{\substack{m \in T(i) \\ m \neq j}} f_m \right] \Delta - \lambda_{ij} p \Delta$$

Step 3: If $\Delta_{ijkk'} > 0$, then go to step 4.

Else,

if $u_m > 0$ for all $(m,k) \in T(j)$, set l = j and go to step 1.

else go to initialization step and evaluate next (i,j,k,k') $\in$ A.

Step 4: Determine the maximum allowable value of Δ.

$$\Delta_1 = \min_{(m,k) \in NT(i)} \left[w_k + v_{imk} - u_i - u_m\right]$$

$$\Delta_2 = \min_{m \in T(i)} \left[u_m\right]$$

If at least one of k or k' $\notin$ B, then between k and k', determine the constraint with the minimum slack. Let the index of this constraint be k*.

$$\Delta_3 = \left[\frac{\text{slack of constriant } k^* \text{ of (3.3)}}{\lambda_{ij}}\right]$$

$$\Delta = \begin{cases} \min[\Delta_1, \Delta_2] & \text{if both k and k'} \in B \\ \min[\Delta_1, \Delta_2, \Delta_3] & \text{if both k and k'} \notin B \end{cases}$$

Step 5: Repeat steps 1 through 4 for all values of (i,j,k,k') $\in$ A, and pick the combination (i,j,k,k') with the maximum value of $\Delta_{ijkk'}$.

Update: $u_i \leftarrow u_i + \Delta$

$u_m \leftarrow u_m - \Delta$ for every m $\in$ T(i)

$v_{ijk} \leftarrow v_{ijk} + \Delta$

$v_{ijk'} \leftarrow v_{ijk'} + \Delta$

If k and k' $\in$ T2,

then $\alpha \leftarrow \alpha + \lambda_{ij}\,\Delta$

Bound $\leftarrow$ Bound + $\Delta_{ijkk'}$

Update sets A and B, and go to step 1.

Once again, we repeat the second procedure until no further improvement is possible. If we increased any dual variables, the sets A and B would have changed at the end of the second procedure, and there might be an opportunity to repeat the first procedure. Hence, we go back to the first procedure, and repeat the two procedures until we can make no further improvements from both the procedures. The final value of the objective function when the procedure terminates is the lower bound to the ingot sizing problem. We can use the final dual solution to construct a primal feasible solution for the ingot sizing problem. This process is explained in the next section.

3.2 Upper Bounds for [ISP]

We develop several heuristic solutions for the ingot sizing problem. The dual solution at the end of the dual ascent phase provides one starting solution. We have also developed two stand-alone heuristic procedures, which provide additional upper bounds. In the following section, we describe the heuristic solution approaches.

3.2.1 Dual Heuristic Solution

When the dual ascent procedure terminates, we try to construct a primal feasible solution using complementary slackness conditions. If the kth constraint in (3.3) is tight, then by complementary slackness, the corresponding ingot size k is a candidate for being included in the set of standard sizes. Hence, if the dual ascent procedure ends with p or less constraints of (3.3) being tight, then we choose the ingots corresponding to these tight constraints as the standard ingots. Given the standard ingot sizes, we then determine the optimal combinations and the actual solution value, by transforming the problem to a non-bipartite matching problem as described in the Section 2.5.1. If more than p constraints of (3.3) are tight at the end of the dual ascent phase, we select all the ingots corresponding to the tight constraints as candidates for standard sizes. We then apply the heuristics explained below for this restricted set of ingots to select a subset as standard ingots.

The two stand-alone heuristics that we have developed are greedy heuristics. One of them selects ingots based on the utilization of the available ingots, and the other based on the total demanded weight of orders.

3.2.2 Ingot Utilization Heuristic

This heuristic picks ingots based on the total weight of the orders an ingot can satisfy. As an input parameter, we specify a threshold utilization level of $\beta\%$. The purpose of this threshold utilization level is to reduce the combinations that we consider to only those that utilize the ingots well. We consider only feasible combinations that occupy at least $\beta\%$ of an ingot's weight, and refer to them as "β-effective" combinations. Using this strategy, we attempt to minimize the amount of scrap generated while satisfying demand. Given a set of candidate ingots and their weights, the threshold β for the minimum acceptable utilization level, and the set of all products with their weights and frequencies, we need to pick a set of at most p standard ingots. Once we determine the set of standard ingots, we find the optimal allocation of order combinations to the ingots.

For each available candidate ingot, we determine a measure of its "flexibility", M(k), as the total weight of all β-effective combinations that the ingot can satisfy. We pick the ingot with the maximum measure of flexibility as a standard ingot, and assign the corresponding feasible combinations (i,j) λ_{ij} times to this ingot. We then update the frequencies of the products, and repeat the procedure until we have chosen p ingots, or all product demands are satisfied. Consider the situation when we have chosen less than p ingots, and all the product demands are not yet satisfied. If we do not have any product combinations that occupy at least $\beta\%$ of any of the available ingots, then we reduce the value of β by a fixed percentage and continue with the heuristic. This prevents the heuristic from stopping prematurely. In our computations, we reduce β by 10%. We present the formal description of the heuristic below.

Initialization: Set $N = 0$

Step 1: Set $M(k) = 0$ for every ingot k. Let F_k be the set of all feasible order combinations (i,j) on ingot k such that $\text{weight}_i + \text{weight}_j \geq \beta\, w_k$.

Step 2: For every ingot k that has not been chosen as a standard ingot, calculate

$$M(k) = \sum_{m \in F_k} b_m \left(\text{weight}_m\right),$$

where each element m of F_k corresponds to two orders i and j, and $b_m = \min\left\{f_i, f_j\right\} = \lambda_{ij}$, and weight_m = sum of the weight of the products in the combination.

Step 3: If $M(k) = 0$ for all ingots, then set $\beta = 0.90 * \beta$ and go to step 1. Else, pick the ingot with the maximum value of M(k) as a standard ingot.

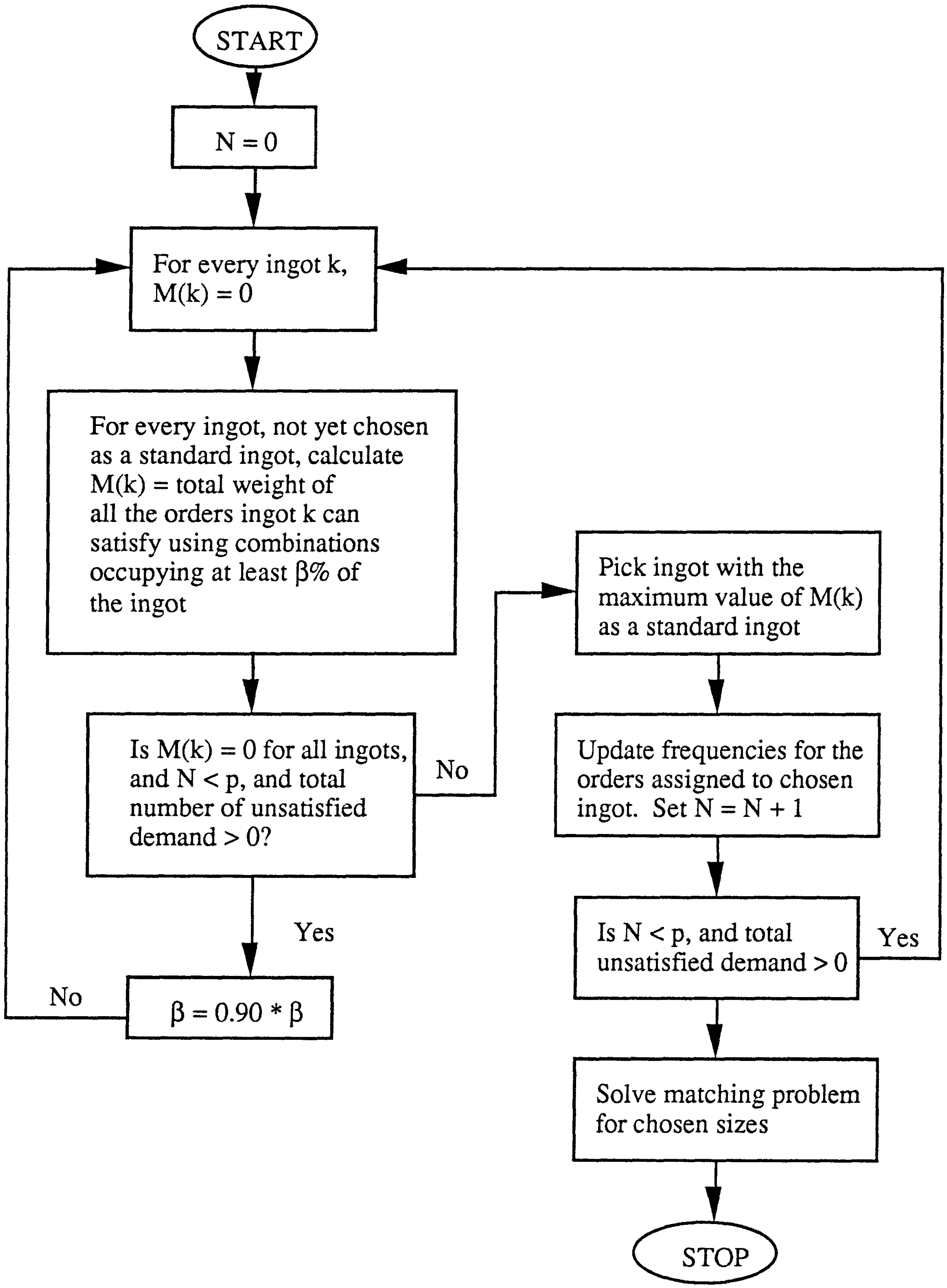
Step 4: Set $N = N + 1$ and update frequencies of all the orders that have been used to calculate the measure of the chosen ingot.

Step 5: If $N < p$ and the total unsatisfied demand > 0, then go to step 1. Else go to step 6.

Step 6: For the chosen set of ingots, solve the non-bipartite matching problem to determine the optimal combinations and the actual cost of the selection.

Figure 3 provides a flowchart of the ingot utilization heuristic.

Figure 3. Flowchart for Ingot Utilization Heuristic



3.2.3 Order Based Heuristic

This is also a greedy heuristic which first selects the order combination with the maximum total ordered weight, and assigns it to its best feasible ingot. The best feasible ingot for a combination is the ingot which can produce the combination with minimum scrap. This best feasible ingot is then chosen as the first standard ingot. We assign the combination to the ingot λ_{ij} times, and update the frequencies of products i and j of the combination.

This heuristic attempts to satisfy demand for the highest volume combinations with minimum scrap. We continue to assign the high volume orders to their best ingots until we have chosen p standard sizes, or satisfied demand for all the products. We do not use any threshold values to make the selection decision, since we always assign the current combination to its best ingot, irrespective of the percentage of the ingot the combination utilizes. We provide a formal description of the heuristic below.

Initialization: Set N = 0

Step 1: For each feasible order combination (i,j), set M(i,j) = 0.

Step 2: For each feasible combination (i,j) calculate
$M(i,j) = \lambda_{ij} * \left(\text{weight}_i + \text{weight}_j\right).$

Step 3: Pick the combination with the maximum value of M(i,j) and assign it to the best feasible ingot, i.e., the ingot which minimizes scrap for this combination. Update the frequencies for the orders in the chosen combination. Set N = N + 1.

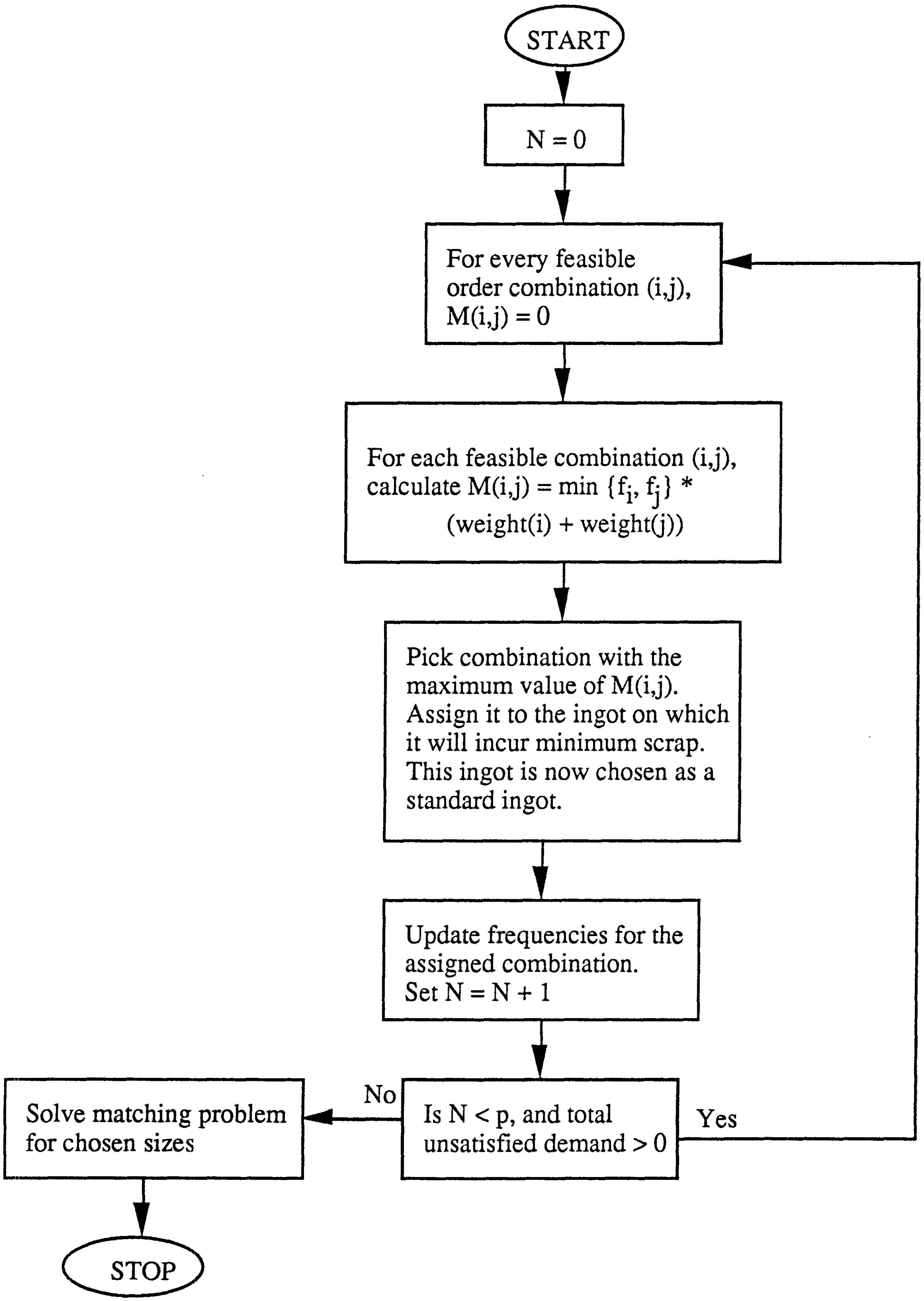
Step 4: If $N < p$, and total unsatisfied demand > 0, then go to step 1. Else solve the non-bipartite matching problem with the selected set of standard sizes to determine actual cost of the solution.

This heuristic strategy might be useful when the demand for a few orders dominate the demand of all the other orders. In this case, we want to pick good ingots to satisfy the demand for the orders with the maximum weight and then satisfy demand for the remaining products from the selection that we have made. Figure 4 shows a flowchart of the order based heuristic for the ingot sizing problem.

A local improvement would attempt to move from a standard ingot to its neighbor that is not chosen and determine the cost of that selection. In doing so, we have make sure that the new set of ingots can satisfy demand for all the products. If the matching problem for the new set of ingots has a feasible solution, then the new set of ingots is feasible. If the solution value with this new set of ingots is lesser than, the previous solution value, then we can make the swap permanent. We have not implemented the local improvement procedure, since initial computational results showed that the gaps between the heuristic solution values and the lower bounds were not very high.

In this chapter, we have presented the dual ascent procedure which generates lower bounds and heuristic solution for the ingot sizing problem. We have also described two greedy heuristics for generating upper bounds for the problem. In the next chapter , we test both the heuristics and measure their quality with the lower bound generated by the dual ascent procedure.

Figure 4. Flowchart for Order Based Heuristic



Chapter 4

Computational Results

We implemented the dual ascent procedure and the ingot utilization and order based heuristics, and tested them with data on actual orders received over a year at a leading aluminum sheet manufacturer. In this chapter, we present the implementation details and the results obtained from the heuristic and the dual ascent procedure.

4.1 Implementation Details and Data Analysis

We implemented the dual ascent procedure and the heuristic in FORTRAN on an IBM 4381 computer. For each problem, we first generate all the feasible combinations based on the order combination rules. This set of feasible combinations serves as an input to the heuristics and the dual ascent procedure. Given the set of feasible combinations, and the set of products and candidate ingot sizes, we obtain the dual ascent lower bound and heuristic solutions for each problem. For solving the non-bipartite matching problem, we use a FORTRAN implementation of a matching algorithm developed by Derigs [1988].

We obtained data on actual orders received at an aluminum sheet manufacturing facility over a period of one year, for one important product group. The data set contains a one record for each order placed. Orders with similar gauge, width and weight specifications are grouped into a single product type. We use the following rules to group orders into product types.

- Width of all orders are rounded up to the closest integer value.
- Orders of the same width can be grouped if they satisfy gauge and weight requirements explained below.
- For each width, if the gauges of two or more orders fall within a non-overlapping interval of the gauge combination table (Table 1), then we can group the orders. For example, we see that in the first row of Table 1, we allow any order of gauge between 0.039 inches and 0.071 inches to be combined with an order of gauge less than or equal to 0.229 inches. The interval (0.039, 0.071) does not over lap with any other interval. Now, if we group all orders with gauges in the interval (0.039, 0.071) into a single product type, we can still combine the orders of this product with orders of gauge less than or equal to 0.229 inches.
- For each width and gauge interval, we group all orders with weight of ±1000 pounds into a single product type.

The weight of a product is the average weight of the orders that have been grouped into that product, and its frequency is the number of orders that have been grouped into that product. The original data set had 638 orders, ranging in width from 30[1] inches to 90 inches. The weight of the orders ranged from 2,500 pounds to 115,000 pounds. Since the plant can only produce ingots weighing up to 40,000 pounds, we eliminated 100 orders that required ingots weighing more than 40,000 pounds. The remaining orders were grouped into products, using the rules above.

We grouped the products into 4 data sets. The first data set contains products with width ranging from 30 to 42 inches, and the second has products with widths from 48 to 53 inches. The third data set consists of a single width (54 inches) which accounts for

[1] All numbers from the original data have been disguised to preserve confidentiality of data.

approximately 50% of the individual orders and 40% of the total pounds of metal ordered. The fourth data set has products with width ranging from 55 to 90 inches.

Currently, the facility has 6 standard sizes. The standard sizes are of 3 widths - 60, 72 and 84 inches. The weight of the standard ingots range from 12,000 to 40,000 pounds. For our experiments, we generate candidate ingot sizes from 5,000 to 40,000 pounds. The actual weights and widths of the candidate ingot sizes depend on the product widths and demand. We have candidate ingot sizes to satisfy demand for all products, and allow more sizes for higher volume widths. For example, in problem 1 products with a width of 42 inches account for more than 50% of the total volume. So we pick 46 inch candidate ingots ranging in weight from 10,000 to 40,000 pounds. We have a candidate ingot size for every 2500 pounds. Other than these 13 candidate ingots, we also have ingots of widths 35, 39, and 43 inches. We have four sizes (10,000, 20,000, 30,000, and 40,000 pounds) for each of the three widths. For each of the four problems, we have a limited set and an extended set of candidate ingot sizes. Table 2 contains data on the number of products, the total frequency, and number of candidate ingots for each of the four problems, and Table 3 presents details on the actual candidate ingot sizes for all the four problems. We solve the ingot sizing problem with the candidate sizes we have chosen for each problem, and the current standard sizes and compare the two solutions. We solve relatively large problems - the largest problem that we solve has 11096 integer variables and 11166 constraints. The following sections describe the various steps involved in detail.

4.2 Framework for Experimentation

This section describes the various computational tests that we performed on the four problems. For each of the four problems, we have the set of products and their frequencies and physical dimensions. We obtain heuristic solutions and lower bound for all four problems using two different sets of candidate ingot sizes for each one of them. We do this to study the impact of the candidate ingot sizes on the total pounds of metal used to satisfy demand for all products. For the first run, we use only a few (the maximum is 10) candidate sizes, while in the second run, we increase the number of candidate ingot sizes. We discuss the effect of the number of candidate sizes on the solution in Section 4.4.

For each set of candidate sizes, we compare the upper bounds from the two heuristics in Section 4.3. For the ingot utilization heuristics, we vary the threshold utilization level β, from 50% to 90%. Initial experiments revealed that the heuristic solution with $\beta = 65, 70$ or 80 is always better than the solutions with $\beta = 50$ or 90. So, we report results for all five levels for problem 3 (in Table 4), and for the remaining results, we use only the 65%, 70% and 80% levels.

When we generate feasible combinations, we restrict the difference in the widths of combined orders to a prespecified level, which we refer to as the maximum width differential. For each problem, we use three different maximum width differential (2, 4, and 6 inches) and repeat the computations.

The ingot sizing problem has an explicit constraint limiting the number of standard sizes to at most p. We can obtain solutions for the different values of p, and make a choice by considering other factors such as ease of tracking inventory storage space, and the inventory costs. We use four different values of p - 3, 4, 5 and 6 - for each problem in our computational study.

Finally, in order to validate the benefits of order combination, we solve the ingot sizing problem for the following two special cases : (i) all orders are dedicated to ingots, and (ii) orders can either be dedicated or combined with one other order of the same type.

We also compare our proposed solution with the set of standard sizes currently stocked by the facility. The results of all these experiments are presented in the following sections.

4.3 Comparison of the Upper Bounds

For all the four problems, we obtained heuristic solutions using the ingot utilization heuristic and the order based heuristic. Tables 5-8 present the heuristic solution values for the four problems when the maximum width differential is 2 inches. The results shown are for the limited set of candidate ingot sizes.

The first observation is that in most of the cases the best solution from the ingot utilization heuristic has less scrap than the solution from the order based heuristic. One reason for this is that the ingot utilization heuristic considers all feasible combinations on an ingot, and chooses the ingot which can satisfy the maximum demand. On the other hand, the order based heuristic could get stuck because of a choice made during the initial stages of the algorithm, based on only one combination with the maximum ordered weight.

The performance of the ingot utilization heuristic with varying utilization levels merits an explanation. When we choose β–effective combinations for an ingot, we compare the weight of the orders to the raw weight of the ingot (before scalping and trim losses). As a result, there might not be many β–effective combinations for an ingot. Hence, at this high level of threshold utilization, we might have to choose an ingot which will serve just a few high utilization combinations and many other relatively low utilization combinations. As a result, we might produce more scrap by choosing this ingot. On the other hand, when we reduce the threshold utilization level to 65 or 70%, we have the opportunity to consider more feasible combinations when selecting an ingot. And at this level, we are also considering the combinations that occupy more than 70% of an ingot.

Table 2. Problem Characteristics

Problem Number	Number of Products	Number of Candidate Ingot Sizes		Total Frequency
		Limited Set	Extended Set	
1	38	8	25	50
2	36	9	22	56
3	73	8	18	260
4	97	10	28	172

Table 3. Candidate Ingot sizes for the 4 problems

Problem Number	List of Candidate Ingot Sizes	
	Limited Set	Extended Set
1	42 - 8* 46 - 10, 12.5, 15, 20, 25, 30, 40	35 - 10, 20, 30, 40 39 - 10, 20, 30, 40 43 - 10, 20, 30, 40 46 - 10, 12.5, 15, 17.5, 20, 22.5, 25, 27.5, 30, 32.5, 35, 37.5, 40
2	46 - 10, 12.5, 40 60 - 15, 20, 25, 35, 40	46 - 10, 20, 30, 40 50 - 10, 15, 20, 25, 30, 35, 40 53 - 10, 15, 20, 25, 30, 35, 40 56 - 10, 20, 30, 40
3	60 - 10, 12.5, 15, 20, 25, 30, 35, 40	58 - 6, 8, 10, 12, 14, 16, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40
4	60 - 5, 20 62 - 40 72 - 7.5, 10, 15, 25, 30, 40	62 - 10, 15, 20, 25, 30, 35, 40 70 - 10, 15, 20, 25, 30, 35, 40 76 - 10, 15, 20, 25, 30, 35, 40 84 - 10, 15, 20, 25, 30, 35, 40

* The first number indicates the width of the candidate ingot in inches and the numbers following each width indicate the weight in thousands of pounds for the ingots.

Table 4. Performance of ingot utilization heuristic - Problem 3

Number of standard sizes allowed	Ingot Utilization Heuristic Utilization level				
	50%	65%	70%	80%	90%
3	3722000	434000	3472000	4792000	3952000
4	3722000	3400000	3438000	3930000	3884000
5	3722000	3390000	3398000	3902000	3392000
6	3722000	3386000	3392000	3402000	3386000

Table 5. Comparison of Heuristics - Problem 1, width differential = 2, limited set of candidate ingot sizes

Number of Standard Sizes Allowed	Ingot Utilization Heuristic Utilization Level			Order Based Heuristic
	65%	70%	80%	
3	910000 (0.34)*	890000 (0.28)	**885000** (0.33)	890000 (0.32)
4	890000 (0.29)	880000 (0.30)	**875000** (0.23)	880000 (0.28)
5	880000 (0.34)	877500 (0.30)	**875000** (0.32)	877500 (0.35)
6	877500 (0.33)	**872500** (0.31)	872500 (0.32)	877500 (0.36)

* Numbers in parenthesis indicate CPU time in seconds on an IBM 4381.

The highlighted number is the best upper bound.

Table 6. Comparison of Heuristics - Problem 2, width differential = 2, limited set of candidate ingot sizes

Number of Standard Sizes Allowed	Ingot Utilization Heuristic Utilization Level			Order Based Heuristic
	65%	70%	80%	
3	1245000 (0.57)*	1260000 (0.57)	1260000 (0.58)	**1240000** (0.40)
4	1245000 (0.58)	1245000 (0.58)	1245000 (0.58)	**1240000** (0.40)
5	**1235000** (0.60)	1245000 (0.58)	1245000 (0.59)	1240000 (0.40)
6	1235000 (0.60)	1245000 (0.58)	**1232500** (0.60)	1240000 (0.40)

* Numbers in parenthesis indicate CPU time in seconds on an IBM 4381

The highlighted number is the best upper bound.

Table 7. Comparison of Heuristics - Problem 3, width differential = 2, limited set of candidate ingot sizes

Number of Standard Sizes Allowed	**Ingot Utilization Heuristic Utilization Level**			**Order Based Heuristic**
	65%	**70%**	**80%**	
3	3625000 (40.53)*	3625000 (40.20)	**3610000** (33.01)	Infeasible
4	**3525000** (43.56)	3610000 (37.42)	3605000 (40.31)	Infeasible
5	3522500 (54.97)	**3515000** (28.10)	3515000 (41.13)	Infeasible
6	3522500 (34.25)	**3515000** (31.06)	3515000 (31.52)	3515000 (31.76)

* Numbers in parenthesis indicate CPU time in seconds on an IBM 4381.

The highlighted number is the best upper bound.

Table 8. Comparison of Heuristics - Problem 4, width differential = 2, limited set of candidate ingot sizes

Number of Standard Sizes Allowed	Ingot Utilization Heuristic Utilization Level			Order Based Heuristic
	65%	70%	80%	
3	**2930000** (4.33)*	3115000 (3.8)	2930000 (4.31)	3095000 (4.11)
4	**2910000** (4.09)	2910000 (4.06)	2910000 (4.05)	2910000 (4.17)
5	**2865000** (3.29)	2865000 (3.27)	2910000 (4.06)	2865000 (3.46)
6	2865000 (3.42)	2865000 (3.3)	2865000 (3.36)	**2855000** (3.91)

* Numbers in parenthesis indicate CPU time in seconds on an IBM 4381.

The highlighted number is the best upper bound.

Hence, the selection we make at this utilization level might produce less scrap when compared to the selection made at 90% utilization level. One interesting observation is that the order based heuristic failed to produce a feasible solution (the standard sizes chosen based on the orders with the maximum ordered weights could not satisfy demand for all orders) for p = 3, 4, and 5 for problem 3. The order based heuristic chooses standard ingots to satisfy demand for high volume combinations with minimum scrap. At each stage, since the choice of ingots is based on only a one combination, the final set of standard sizes could be infeasible.

We also varied the number of standard sizes allowed. As the number of sizes allowed increases, the solution values either remain the same or improve, since we have more ingots to assign the feasible combinations to. We also varied the maximum width differential from 2 inches to 4 and 6 inches. As the width differential decreases, the number of feasible combinations reduces. Hence, we do not have as much flexibility in combining orders. Thus, for a given value of p, as the width differential decreases, the solution value increases. Tables 8-11 show the number of feasible combinations for the different width differentials for the four problems. This table also gives the best upper bound from the two heuristics for all the problems, and the percentage of orders dedicated to an ingot in the best solution. In most of the cases, the percentage of dedicated orders decreases as the width differential increases. This is due to the fact that we have more feasible combinations, and hence more opportunity for order combination. The behavior of the two heuristics exhibit the same characteristics as we change the width differential.

The time taken by the two heuristics are comparable. The reported CPU times include both initialization and computations. When the number of feasible combinations in a problem increases, the matching problem contains more arcs and hence, takes longer to solve. Problem 3 which has 4119 feasible combinations with the first set of ingots takes

Table 9. Comparison of bounds - Problem 1, limited set of candidate ingot sizes

# of Standard Sizes	Width differential	# of feasible combinations	% of orders dedicated in the best solution	Best Upper Bound	Lower Bound	% Gap	Average Gap
3	2	562	32	885000	868500	1.9	2.7
	4	613	20	870000	860000	1.2	
	6	766	20	850000	810000	4.9	
4	2	562	16	875000	868500	0.7	1.5
	4	613	16	860000	860000	0.0	
	6	766	8	840000	810000	3.7	
5	2	562	32	875000	868500	0.7	1.5
	4	613	16	860000	860000	0.0	
	6	766	8	840000	810000	3.7	
6	2	562	32	872500	868500	0.5	1.4
	4	613	16	860000	860000	0.0	
	6	766	8	840000	810000	3.7	

Table 10. Comparison of bounds - Problem 2, limited set of candidate ingot sizes

# of Standard Sizes	Width differential	# of feasible combinations	% of orders dedicated in the best solution	Best Upper Bound	Lower Bound	% Gap	Average Gap
3	2 4 6	391 497 530	17.9 17.9 17.9	1240000 1240000 1240000	1200000 1170000 1200000	3.3 6.0 3.3	4.2
4	2 4 6	391 497 530	17.9 21.4 17.9	1240000 1230000 1230000	1200000 1170000 1200000	3.3 5.1 2.5	3.6
5	2 4 6	391 497 530	17.9 21.4 17.9	1235000 1230000 1230000	1200000 1170000 1200000	2.9 5.1 2.5	3.5
6	2 4 6	391 497 530	21.4 21.4 17.9	1232500 1230000 1230000	1200000 1170000 1200000	2.7 5.1 2.5	3.4

Table 11. Comparison of bounds - Problem 3, limited set of candidate ingot sizes

# of Standard Sizes	# of feasible combinations	% of orders dedicated in the best solution	Best Upper Bound	Lower Bound	% Gap	Average Gap
3	4119	8.5	3610000	3367500	7.2	6.1
4	4119	8.5	3525000	3340000	5.5	
5	4119	7.7	3515000	3325416	5.7	
6	4119	7.7	3515000	3316666	6.0	

Table 12. Comparison of bounds - Problem 4, limited set of candidate ingot sizes

# of Standard Sizes	Width differential	# of feasible combinations	% of orders dedicated in the best solution	Best Upper Bound	Lower Bound	% Gap	Average Gap
3	2	1952	22.1	2930000	2751250	6.5	5.6
	4	2335	16.3	2860000	2731250	4.7	
	6	3068	14.0	2840000	2689687	5.6	
4	2	1952	22.1	2910000	2741250	6.2	5.4
	4	2335	16.3	2850000	2731250	4.3	
	6	3068	14.0	2825000	2676250	5.6	
5	2	1952	31.4	2865000	2738750	4.6	4.0
	4	2335	24.4	2815000	2731250	3.1	
	6	3068	23.3	2790000	2676250	4.3	
6	2	1952	34.9	2855000	2736250	4.3	3.8
	4	2335	25.6	2810000	2731250	2.9	
	6	3068	26.7	2785000	2676250	4.1	

40-50 seconds to solve, while problems 1 and 2 which have a few hundred feasible combinations solve within one second. We measure of the quality of the heuristic solutions by comparing them with the lower bound from the dual ascent procedure. The next section presents the results and discussion of the comparison of the bounds.

4.4 Comparison of Lower and Upper Bounds

We compare the best upper bound from the two heuristics with the lower bound generated by the dual ascent procedure. The dual ascent procedure also generates heuristic solutions. For problems 1 and 2, the product frequencies are low when compared to the value of p and hence, the dual ascent procedure stops with the initial solution. So, the heuristic solution generated by the dual ascent bound is equal to the best upper bound from the two heuristics. However for problems 3 and 4, the dual ascent procedure improves the initial bound and obtains a heuristic solution. In all cases, the scrap produced by this selection of ingots is greater than or equal to the scrap produced by the best heuristic solution.

Tables 9-12 present the results of the comparison between the bounds for the four problems. We compare the bounds for all the three width differentials and the four standard sizes. For problem 3, we do not have any width differentials, since this problem deals with only one width. We calculate the gap between the bounds as:

$$\text{percentage gap} = \frac{(\text{upper bound} \ - \ \text{lower bound})}{\text{lower bound}} * 100\%$$

For all our test problems, the gap between the bounds decreases with an increase in the number of standard sizes. When the dual ascent stops with the initial solution for some value of p = p', the lower bound remains the same for all values of p greater than p'. However, we noticed that if the dual ascent improves the initial solution, the lower bound decreases with an increase in the number of standard sizes. The average gaps for the four problems are 1.8%, 3.7%, 6.1%, and 4.7% respectively for the first set of ingot sizes. When we increase the number of candidate ingot sizes, the total scrap for the best solution from the two heuristics decreases and so does the lower bound. The average gap between the upper and lower bounds for the four problems in this case are 2.7%, 3.0%, 4.9%, and

5.9%. Tables 13-16 present the comparison between the lower bound and the best upper bound for the four problems, when using the second set of candidate ingot sizes. The procedure obtains solutions within 4% of optimality on an average.

4.5 Comparison with Current Practice

The manufacturing facility currently uses 6 standard sizes. Using these 6 sizes as the standard ingot sizes, we use the matching procedure to determine the order combinations and the total scrap for each problem. We assume that the facility combines orders rather than just dedicating ingots to orders. We compare the best heuristic solution with current practice and evaluate the percentage reduction in scrap for the proposed set of standard sizes.

Table 17 presents the results of the comparison with current practice for the first set of candidate ingot sizes. Once again, for problem 3, we do not have 3 width differentials, since this problem has only one width. The standard sizes currently used by the facility are of only three widths - 60, 72, and 84. In our proposed solutions, at least for the first problem which has orders of width 30 to 42 inches, we allow ingots of width 42-54 inches. This is clearly reflected by the 18.3% reduction in the proposed solution's total scrap over current practice. Problem 2 has a reduction of 21.5%, while problem 3 has only a 0.9% reduction.

We used 8 candidate ingot sizes for problem 3. In the limited set of candidate ingot sizes, we used ingot weights from 10,000 to 40,000 pounds, with an ingot for every 500 pounds. We also used an additional 7,500 pound ingot. All ingots are of width 58 inches. We wanted to test if increasing the number of candidate ingots could reduce the scrap. So, in the extended set of candidate ingots, we used 18 ingot weights from 6,000 to 40,000 pounds, with an ingot for every 2,000 pounds.

We compared the reduction in scrap for the proposed solution when compared to current practice in both cases. The 0.9% reduction in scrap when using the limited set of ingots increased to 5.2% when we used the extended set of ingots. This highlights the importance of using good candidate ingot sizes. Even with the extended set of ingots sizes, the reduction in scrap for problem 3 is lower than the reduction for problems 1 and 2. This is due to the fact that the current set of standard sizes at the facility consist of three 60 inch width ingots, two 72 inch ingots, and a 84 inch ingot. The 60 inch ingot caters to the products in problem 3 (of width 54 inches) with minimum scrap. On the other hand, our proposed solution for problems 1 and 2 introduces new widths which reduce the scrap significantly. The results for the fourth problem are similar to that of the third problem. We present the comparison of the proposed solution with current practice for the extended set of candidate ingots in Table 18. The proposed set of standard sizes reduce scrap when compared with current practice. By reducing scrap, the facility can save on the ingot casting and reprocessing costs for the scrap. For the alloy (all four problems together) that we studied, the proposed solution could potentially reduce total scrap by an average of 9.5%, and could result in total savings of up to $100,000 annually.

Table 13. Comparison of bounds - Problem 1, extended set of candidate ingot sizes

# of Standard Sizes	Width differential	# of feasible combinations	% of orders dedicated in the best solution	Best Upper Bound	Lower Bound	% Gap	Average Gap
3	2	2656	16	780000	740000	5.4	3.7
	4	2955	0	762500	740000	3.0	
	6	3819	0	755000	735000	2.7	
4	2	2656	4	777500	740000	5.1	3.4
	4	2955	8	757500	740000	2.4	
	6	3819	0	755000	735000	2.7	
5	2	2656	8	760000	740000	2.7	1.8
	4	2955	4	747500	740000	1.0	
	6	3819	4	747500	735000	1.7	
6	2	2656	8	760000	740000	2.7	1.8
	4	2955	4	747500	740000	1.0	
	6	3819	4	747500	735000	1.7	

Table 14. Comparison of bounds - Problem 2, extended set of candidate ingot sizes

# of Standard Sizes	Width differential	# of feasible combinations	% of orders combined in the best solution	Best Upper Bound	Lower Bound	% Gap	Average Gap
3	2	1387	17.9	1200000	1150000	4.3	4.5
	4	1781	17.9	1200000	1147500	4.6	
	6	1892	17.9	1200000	1147500	4.6	
4	2	1387	17.9	1190000	1150000	3.5	3.0
	4	1781	21.4	1180000	1147500	2.8	
	6	1892	21.4	1180000	1147500	2.8	
5	2	1387	17.9	1180000	1150000	2.6	2.2
	4	1781	17.9	1170000	1147500	2.0	
	6	1892	17.9	1170000	1147500	2.0	
6	2	1387	21.4	1180000	1150000	2.6	2.2
	4	1781	17.9	1170000	1147500	2.0	
	6	1892	17.9	1170000	1147500	2.0	

Table 15. Comparison of bounds - Problem 3, extended set of candidate ingot sizes

# of Standard Sizes	# of feasible combinations	% of orders combined in the best solution	Best Upper Bound	Lower Bound	% Gap	Average Gap
3	9838	6.9	3434000	3251999	5.6	4.9
4	9838	6.9	3400000	3244999	4.8	
5	9838	7.7	3390000	3242999	4.5	
6	9838	6.7	3386000	3231000	4.8	

Table 16. Comparison of bounds - Problem 4, extended set of candidate ingot sizes

# of Standard Sizes	Width differential	# of feasible combinations	% of orders combined in the best solution	Best Upper Bound	Lower Bound	% Gap	Average Gap
3	2	7107	33.7	3115000	2680000	16.2	9.7
	4	8457	16.3	2850000	2672500	6.6	
	6	11068	14.0	2830000	2662500	6.3	
4	2	7107	19.8	2855000	2680000	6.5	5.6
	4	8457	15.1	2815000	2672500	5.3	
	6	11068	14.0	2795000	2662500	5.0	
5	2	7107	19.8	2840000	2680000	6.0	5.0
	4	8457	15.1	2795000	2672500	4.6	
	6	11068	14.0	2780000	2662500	4.4	
6	2	7107	30.2	2780000	2680000	3.7	3.3
	4	8457	24.4	2755000	2672500	3.1	
	6	11068	24.4	2745000	2662500	3.1	

Table 17. Comparison of Proposed Solution with Current Practice - limited set of candidate ingots

Problem Number	# of Standard Sizes	Width Differential	Current Practice	Proposed Solution	% Reduction	Average
1	6	2 4 6	1071792 1050408 1025784	872500 860000 840000	18.6 18.1 18.1	18.3
2	6	2 4 6	1571376 1570080 1559064	1232500 1230000 1230000	21.6 21.7 21.1	21.5
3	6	-	3546504	3515000	0.9	0.9
4	6	2 4 6	2799352 2773432 2745568	2855000 2810000 2785000	-2.0 -1.3 -1.4	-1.6

Table 18. Comparison of Proposed Solution with Current Practice - extended set of candidate ingots

Problem Number	# of Standard Sizes	Width Differential	Current Practice	Proposed Solution	% Reduction	Average
1	6	2 4 6	1071792 1050408 1025784	760000 747500 747500	29.1 29.0 27.1	28.4
2	6	2 4 6	1571376 1570080 1559064	1180000 1170000 1170000	24.9 25.5 25.0	25.1
3	6	-	3546504	3364000	5.2	5.2
4	6	2 4 6	2799352 2773432 2745568	2780000 2755000 2745000	0.70 0.70 0.02	0.5

4.6 Impact of Order Combination

In order to understand the effect of order combination on the total scrap, we solved two special cases of the ingot sizing problem for problem 3. In the first case (product dedication), orders can either be dedicated to an ingot or two orders of the same product type can be combined on an ingot. In the second case (order dedication), orders can only be dedicated to ingots. Table 19 presents the results of this comparison. The number of feasible combinations increases dramatically when we allow order combination. We have 897 possible assignments of orders to ingots for the order dedication model, 1441 possible assignments of products to ingots for the product dedication model, and 9838 combinations for the order combination model. In each case, we use our solution methodology to determine a good set of standard sizes. Order combination allows us to utilize the available ingots better, and reduces total scrap. The results indicate that order combination reduces scrap by an average of 8.3% when compared to the product dedication model, and by an average of 26% when compared to the order dedication model. These figures illustrate the benefits of order combination.

4.7 Summary

This chapter presents the computational results of our study. We have implemented the dual ascent procedure and the heuristics and tested them with data on actual orders received at an aluminum sheet manufacturing facility. The overall results indicate that the methodology is efficient. In order to illustrate the results of the various sensitivity analyses that we performed, we summarize the results for problem 3. This problem accounts for more than 50% of the total volume. The average gap between the upper and the lower bounds is 5.5%. The ingot utilization heuristic performed better than the order based heuristic in most cases. The impact of the utilization threshold β on the total scrap was

Table 19. Order combination versus dedication - Problem 3

Number of Standard Sizes	Problem Type	Number of Feasible Combinations	Solution	% of Dedicated Orders	% Increase in Scrap over ISP
3	ISP Product Dedication* Order Dedication **	9838 1441 897	3434000 3978000 4928000	6.9 24.6 100.0	- 13.7 30.3
4	ISP Product Dedication* Order Dedication **	4838 1441 897	3400000 3704000 4648000	6.9 25.4 100.0	- 8.2 26.9
5	ISP Product Dedication* Order Dedication **	9838 1441 897	3390000 3602000 4528000	7.7 25.4 100.0	- 5.9 25.1
6	ISP Product Dedication* Order Dedication **	9838 1441 897	3364000 3558000 4268000	6.2 25.4 100.0	- 5.5 21.2

* Product Dedication: Orders can either be dedicated on an ingot, or two orders of the same product can be combined on an ingot.

** Order Dedication: Orders can be dedicated on an ingot quite interesting. We obtained the least scrap for 65 or 70% threshold utilization. The scrap increased when we either decreased or increased β beyond these levels, in most cases.

As we increased the number of allowed standard sizes from 3 to 6, the total scrap reduced and for the problems that we solved, the gap between the bounds decreased too. We also illustrated the impact of candidate ingot sizes. When we increased the number of candidate ingot sizes from 8 to 18, the total scrap reduced by 4.1%. In comparison with current practice, the proposed set of ingots for this problem reduces scrap by 5.2%. This problem contained products of a single width. For the other three problems, we studied the effect of increasing the maximum width differential - the maximum allowed difference in widths of combined orders. As we increase the width differential, the number of feasible product combinations increases, and the total scrap decreases. For instance, the scrap reduces by 2.4% on an average when we increase the width differential from 2 to 6 for problem 4.

For the problems that we solved the method obtains solutions within 4% of optimality on an average. Both the heuristics and the dual ascent procedure are relatively quick. We solve relatively large problems within one minute of CPU time. Finally, we highlighted the benefits of order combination by comparing the scrap for the order combination solution to scrap for the order dedication and product dedication models. Order combination reduces scrap by an average of 8.3% when compared to the product dedication model, and by an average of 26% when compared to the order dedication model. Finally, for the alloy that we studied, the proposed set of candidate ingots reduces scrap by an average of 9.5%, and the total reduction in scrap could result in savings of up to $100,000 annually.

Chapter 5

Conclusions and Recommendations

In this thesis, we addressed the tactical production planning problem of ingot sizing in metal sheet manufacturing. Ingots used for sheet manufacturing are made-to-stock, and used when necessary, to satisfy customer demands. Orders can be combined on a single ingot in order to minimize total scrap. In this scenario, given the set of products, their dimensions, weights and frequencies, the set of available ingots, and the rules for order combination, we address the question of selecting standard ingot sizes to maintain in stock. The objective is to minimize total processing and scrap reprocesisng cost.

5.1 Summary and Conclusions

We have formulated the ingot sizing problem as an integer program, and developed an efficient solution procedure. The solution procedure consists of dual ascent to obtain lower bounds, and two heuristics to provide good feasible solutions to the ingot sizing problem. We have implemented the dual ascent procedure and the heuristics, and tested them with data on actual orders received at a leading aluminum sheet manufacturing facility. The computational results indicate that the solutions obtained by the dual ascent and heuristic solution procedure are within 4% of optimality on an average.

A comparison of the proposed solution with the current set of standard ingots suggests that the proposed set of standard sizes could reduce the total scrap for the alloy that we studied by an average of 9.5%. Since approximately 9% of the orders received are of width less than 40 inches, the proposed solution suggests carrying 48 inch width ingots also as standard size, as opposed to just 54, 66, and 78 inch ingots. This reduces the scrap generated when satisfying demand for the orders with width less than 45 inches. We also demonstrated that order combination reduces total scrap by an average of 26% when compared to the scrap generated when we dedicate an ingot to every order. The time taken by the solution procedure is under one minute for all problems. The results suggest that the proposed methodology is effective and could result in significant savings for the company.

5.2 Future Work

The ingot sizing problem determines the set of standard ingot sizes to stock, and the optimal order combinations and the assignment of these combinations to the standard ingots. Our model considers the forecast demand for all products over a long planning horizon. For a particular demand realization, two products that have to combined might not occur simultaneously. In this case, we have to choose an alternate combination, and hence the actual amount of scrap generated might be different from the amount produced by our heuristic solution. It will be interesting to determine the deviation of the actual scrap generated from the total scrap generated by the ingot sizing model solution. We can simulate the order arrival process, and schedule the orders as they arrive on the proposed set of ingots to measure the actual amount of scrap generated.

When a sheet manufacturing company has more than one plant where it can make ingots and final products, it must decide how to allocate ingot and sheet production to various plants to utilize capacities effectively while meeting customer requirements at minimum total production and transportation cost. Therefore, at the long term planning stage, we would decide which plants would produce what size of ingots, given the production costs and capacities at the various plants, the forecasted customer demands, and transportation costs between plants, and between customers and plants. The goal is to minimize total production and distribution costs, and the decision serves as an input to the medium term planning problem.

References

Balakrishnan, A., T. L. Magnanti, and R. T. Wong, "A Dual Ascent Procedure for Large scale Uncapacitated Network Design", *Operations Research*, Vol. 37, pp. 716-740 (1989).

Balakrishnan, A., and S. B. Brown, "Process Planning for Metal-forming Operations: An Integrated Engineering-Operations Perspective", Working Paper # 3432-92-MSA, Sloan School of Management, MIT, Cambridge, Massachusetts (1992).

Balakrishnan, A., "Packing Orders in Ingots: A Production Planning Model for Aluminum Sheet Manufacturing", Working Paper, Sloan School of Management, MIT, Cambridge, Massachusetts (1993).

Beasley, J. E., "An Algorithm for the Two-dimensional Assortment Problem", *European Journal of Operational Research*, Vol. 19, pp. 253-261 (1985).

Chambers, M. L and R. G. Dyson, "The Cutting Stock Problem in Flat Glass Industry - Selection of Stock Sizes", *Operational Research Quarterly*, Vol. 27, pp. 949-957 (1976).

Christofides, N. and C. Whitlock, "An Algorithm for 2-dimensional Cutting Problems", *Operations Research*, Vol. 25, pp. 30-44 (1977).

Derigs, U., "Solving Non-bipartite Matching Problems via Shortest Path Techniques", *Annals of Operations Research*, ed. Peter L. Hammer, Vol. 13, pp. 225-264 (1988).

Erlenkotter, D., "A Dual Based Procedure for Uncapacitated Facility Location", *Operations Research*, Vol. 26, pp. 992-1009 (1978).

Farley, A. A., "Selection of Stockplate Characteristics and Cutting Style for two Dimensional Cutting stock Situations", *European Journal of Operational Research*, Vol. 44, pp. 239-246 (1990).

Fisher, M. L., and P. Kedia, "Optimal Solution of Set Covering/Partitioning Problems using Dual Heuristics", *Management Science*, Vol. 25, pp. 955-966 (1986).

Garey, M. R. and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco (1979).

Gilmore, P. C., and R. E. Gomory, "A Linear Programming Approach to the Cutting Stock Problem", *Operations Research*, Vol. 9, pp. 849-859 (1961).

Gilmore, P. C., and R. E. Gomory, "A Linear Programming Approach to the Cutting Stock Problem, Part II", *Operations Research*, Vol. 11, pp. 863-888 (1963).

Gilmore, P. C., and R. E. Gomory, "Multistage Cutting Stock Problems of two or more Dimensions", *Operations Research*, Vol. 13, pp. 94-120 (1965).

Gopalan, R., "Exploiting Process Flexibility in Metal Forming Operations", Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (1992).

Goulimis, C., "Optimal Solutions for the Cutting Stock Problem", *European Journal of Operational Research*, Vol. 44, pp. 197-208 (1990).

Mirchandani, P. B. and R. L. Francis, *Discrete Location Theory*, John Wiley & Sons, Inc., New York, 1990.

Papdimitriou, C. H. and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey (1982).

Pentico, D. W., "The Assortment Problem with Probabilistic Demands", *Management Science*, Vol. 21, pp. 286-290 (1974).

Pentico, D. W., "The Discrete Two-dimensional Assortment Problem", *Operations Research*, Vol. 36, pp. 324-332 (1988).

Stadtler, H., "A One-dimensional Cutting Stock Problem in the Aluminum Industry and its Solution", *European Journal of Operational Research*, Vol. 44, pp. 209-223 (1990).

Vasko, F. J., F. E. Wolf and K. L. Scott, "Optimal Selection of Ingot Sizes via Set Covering", *Operations Research*, Vol. 35, pp. 346-353 (1987).

Vasko, F. J., F. E. Wolf and K. L. Scott, "A Set Covering Approach to Metallurgical Grade Assignment", *European Journal of Operational Research*, Vol. 38, pp. 27-34 (1987).

Vasko, F. J., F. E. Wolf, K. L. Scott, and J. W. Scheirer, "Selecting Optimal Ingot Sizes for Bethlehem Steel", *Interfaces*, Vol. 19, pp. 68-84 (1989).

Ventola, D. P., "Order Combination Methodology for Short-term Lot Planning at an Aluminum Rolling Facility", Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (1991).

Wang, P. Y., "Two Algorithms for Constrained Two-Dimensional Cutting Problems", *Operations Research*, Vol. 31, pp. 573-586 (1986).

Wolfson, M. L., "Selecting the Best Lengths to Stock", *Operations Research*, Vol. 13, pp. 570-585 (1965).

Wong, R. T., "Dual Ascent Approach for Steiner Tree Problems on a Directed Graph", *Mathematical Programming*, Vol. 28, pp. 271-287 (1984).