





M.I.T. LIBRARIES - DEWEY



HD28  
.M414  
10  
3734-  
94

13

Selecting Ingot Sizes for Joint Order Processing  
in Sheet Manufacturing

Anantaram Balakrishnan  
and  
Srimathy Gopalakrishnan

WP# 3734-94-MSA

October 1994

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

DEC 01 1994

LIBRARIES

## Abstract

This paper addresses a tactical planning problem of selecting standard ingot sizes to facilitate joint processing of orders in a make-to-order aluminum sheet manufacturing operation. The facility has traditionally dedicated ingots to individual orders, but is now considering producing multiple orders from each ingot to reduce surplus and exploit economies of scale in processing costs. The effectiveness of this joint processing strategy depends on the available standard ingot sizes. We develop an integer programming model to decide ingot sizes given the distribution of demand for various products and the order combination constraints. The model simultaneously addresses the issue of selecting a prespecified number of standard sizes while deciding which product combinations to use. The objective is to minimize the total weight of ingots processed to satisfy demand for all products. We describe a dual-ascent method to generate lower bounds for the problem and heuristic procedures to identify good solutions. Extensive computational results using actual data from a leading aluminum sheet manufacturer confirm the effectiveness of the algorithm: on average the gaps between the upper and lower bounds are about 4%. We also illustrate how the model can be used to perform sensitivity analyses for tactical planning.

# 1. Introduction

This paper addresses a tactical planning problem in a leading aluminum sheet manufacturing facility that produces sheet and plate products to customer specifications. The problem, motivated by technological and market trends in the industry, deals with selecting standard ingot sizes to produce multiple orders from a single ingot. The facility produces ingots to stock in a few standard sizes. The downstream rolling mills use these standard ingots to produce finished coils to order. To exploit the considerable economies of scale in ingot casting and rolling costs, the facility has continuously upgraded its equipment to produce and process larger ingots. At the same time, customers are placing smaller orders more frequently to reduce their raw material inventories and move towards just-in-time manufacturing. Consequently, the plant's traditional policy of dedicating an ingot for each order is not economically viable. For instance, producing a 4,000 pound order using a 10,000 pound ingot requires either storing the remaining 6,000 pounds of finished product in stock or recycling this metal; furthermore, processing the 10,000 pound ingot uses up scarce rolling mill capacity, possibly delaying other orders. To overcome these problems, the plant is exploring the option of combining orders, i.e., jointly producing multiple orders that require the same alloy and have similar specifications using a single ingot. For a detailed discussion of the costs and benefits of order combination, see Ventola [1991] and Balakrishnan and Brown [1992].

The choice of standard ingot sizes (widths and weights) determines the number of available opportunities to combine orders, and thus greatly influences the cost effectiveness of the order combination strategy. Reviewing and revising ingot sizes is a medium-term (one to two years) decision since adding new standard ingot sizes requires considerable lead time and investment to acquire and test new tooling. This paper develops an optimization model to support this tactical *ingot sizing* decision. The model selects a prespecified number of standard ingot sizes to minimize the total weight of ingots used to manufacture a representative mix of products for each alloy. Minimizing the total ingot weight reduces processing time and costs as well as planned scrap and "surplus" (i.e., excess production that is not shipped). We formulate the problem as an integer program, and develop an optimization-based solution procedure that combines dual ascent and heuristics. Computational results using actual data from an aluminum sheet manufacturing facility demonstrates that the solution procedure is quite effective, producing solutions that are within 4% of optimality on average even without local improvement of the heuristic solutions. For one product family that we analyzed, the new ingot sizes chosen by this



model provide savings of hundreds of thousands of dollars compared to using current ingot sizes. We also use the model to study the sensitivity of ingot sizing decisions to various technological and operational parameters.

This paper is organized as follows. Section 2 presents a brief description of the sheet manufacturing process, defines the ingot sizing problem, and describes the integer programming formulation. In Section 3, we develop a dual ascent procedure to compute lower bounds and generate heuristic solutions. We also describe two stand-alone heuristics. Section 4 provides details of our implementation and test problems, and presents computational results and sensitivity analyses.

## **2. Problem Context and Formulation**

### **2.1 Overview of sheet manufacturing process**

Aluminum sheets have a variety of uses ranging from aerospace and automotive applications to construction and packaging. The sheet manufacturing facility that we studied produces a wide range of sheet and plate products to customer specifications. These specifications include alloy, width, gauge (thickness), and total weight (or total length). The facility maintains little or no uncommitted finished goods inventory. The main steps in the sheet manufacturing process are ingot casting, hot rolling, cold rolling, heat treatment, and finishing (e.g., slitting) operations.

Ingot casting consists of melting pure and scrap aluminum with the required alloying elements, and pouring and cooling the molten metal to produce solid rectangular aluminum ingots of the desired dimensions. This process requires separate tooling for each size, and changing over production from one alloy and size to another entails long setup times and wastage of metal (e.g., to flush the furnaces). Therefore, the ingot plant produces only a limited set of standard sizes, primarily to stock, in large lots using a cyclic schedule.

Hot rolling, performed on reversing or multi-stand rolling mills, reduces the ingot's thickness (and elongates it) at elevated temperatures. Cold rolling further reduces the thickness to the desired finished gauge(s). A minimum amount of cold rolling is often necessary to ensure that the product meets tight gauge and finish tolerances, and material specifications (e.g., temper). Several cold rolling passes might be necessary if the finished gauge is small relative to the sheet's thickness when it exits the hot mills. The width of the sheet does not change during hot and cold rolling operations.

Although the number of ingot sizes is small, the facility is able to produce hundreds of different finished specifications (width, gauge, and weight combinations) because the hot and cold rolling operations are quite flexible, i.e., they can produce a wide range of finished gauges from the same ingot, subject to width and weight restrictions. Furthermore, the cold rolling process permits changing the gauge (during the final cold rolling pass) while processing a coil, enabling the plant to jointly produce multiple orders with comparable (but not necessarily identical) widths and gauges in a single coil (i.e., using a single ingot). Since the coil has uniform gauge until the final pass, and since the amount of reduction in each pass is limited, the range of possible gauges in the finished coil is limited. To account for this restriction, the plant has provided a “gauge combination table” that specifies various gauge “windows” (or ranges) for each width. Two products can be jointly produced from a single ingot only if both their gauges belong to one of these windows.

One of the factors contributing to the scale economies in sheet production is the “planned” scrap generated at various stages. For instance, ingots are “scalped” (a fixed depth of material is removed from the top and bottom surfaces) prior to hot rolling. Similarly, fixed lengths of the coil’s leading and trailing ends (called head and tail scrap) are discarded after rolling, and the sides of each coil need to be trimmed prior to shipping. The proportion of metal lost due to “planned” scrap—scalping loss, head and tail scrap, and side trim—decreases as the weight of the ingot increases.

Order combination, i.e., producing multiple orders using a single ingot, is advantageous because it permits the facility to exploit economies of scale in processing large ingots and also reduces the amount of surplus, i.e., excess quantity produced but not shipped to customers. The surplus might be kept in inventory to satisfy some future order for the same specification. To avoid accumulating uncommitted finished goods inventory, the plant currently melts and reuses the surplus, incurring handling and reprocessing costs (including melt loss). The plant's decision to explore order combination was prompted by the increasing quantities of uncommitted metal (i.e., surplus) being recycled due to the steady decrease in the quantity per customer order.

Combining up to three orders per ingot is apparently technically feasible (although not actually tested in practice). However, the plant prefers not to combine more than two orders for operational reasons, especially since order combination is a new and untested

strategy. We say an order pair is *feasible* if the two orders can be jointly produced using an available ingot without violating any of the processing restrictions. In particular, the following conditions must be met:

1. *Alloy feasibility*: Both orders require the same alloy.
2. *Weight feasibility*: The sum of the weights of the two orders must be less than or equal to the net weight (after subtracting the planned scrap allowances) of an available ingot for that alloy;
3. *Width feasibility*: The width of each order must be less than or equal to the width of the ingot minus the required side trim;
4. *Gauge differential*: The gauges of the two products must lie within one of the corresponding gauge windows.
5. *Width differential*: The absolute difference in widths for the two orders must not exceed a prespecified limit which we call the *maximum width differential*  $\omega$ . This operational restriction limits the amount of scrap (since the width of the ingot and coil must exceed the larger of the two orders' widths). In our computational experiments, we perform sensitivity analysis with respect to  $\omega$ .

We refer to these conditions collectively as the *order combination rules*. The weight and width feasibility conditions imply that the feasibility of order pairs depends on the available standard ingot sizes. Two orders might require the same alloy and have compatible gauges and widths, and yet might not be feasible because none of the available ingots for that alloy can accommodate both orders. Alternatively, an order pair might be feasible, but combining them might require using a very large ingot (relative to the total ordered weight); dedicating two smaller ingots to each order might then be more economical. For *dedicated* ingots, i.e., producing a single order using an ingot, the net weight and width of that ingot must equal or exceed the order weight and width.

In the short-run, given the set of standard ingot sizes, the scheduler must decide how to combine the orders on hand and assign them to appropriate ingots. Ventola [1991] and Gopalan [1992] discussed this decision problem, and observed that choosing good standard ingot sizes can considerably improve short-run performance. This paper focuses on the medium-term ingot sizing decision, namely, finding an effective set of standard ingot sizes to facilitate short-run order combination. Since we cannot combine orders requiring different alloys, the ingot sizing problem decomposes by alloy.

## 2.2 Problem definition

For medium-term planning, we discretize the range of widths, gauges, and weights of orders requiring a particular alloy, and consider all orders within each cell as a single product group or **product**. Using past data on actual orders, we estimate the frequency of orders or *demand* (number of orders) for each product. Ingots are specified by their alloy, width, weight, and thickness (or length). We will focus on the ingot's width and weight since only these two dimensions are relevant to test if an ingot can produce a particular product pair (assuming this pair satisfies the gauge and width differential constraints). Equipment capacities impose upper and lower limits on the ingot sizes. We assume that the plant has identified a discrete set of candidate ingot sizes from this range. The ingot plant also specifies the **maximum number  $p$  of standard ingot sizes** to select. Limiting the number of standard ingot sizes has several advantages including reducing the investment in tooling, limiting the number of changeovers and ensuring reasonably large lot sizes, and reducing ingot safety stocks (and hence reducing both storage space requirements and holding costs) due to the risk pooling benefits of having greater commonality (see, for example, Baker, Magazine, and Nuttle [1986]). Our model does not explicitly incorporate these factors; instead, by varying  $p$ , we can study the sensitivity of the overall performance to this parameter.

Given the demand for each product, the parameters for the order combination rules, the candidate ingot sizes, and the maximum number  $p$  of standard ingot sizes, the **ingot sizing problem (ISP)** requires selecting  $p$  or fewer standard ingot sizes, and deciding the number of ingots of each size to use for feasible product pairs (for the selected ingot sizes) or single orders so as to meet the demand for all products. The objective is to minimize the total planned scrap and surplus, or equivalently (since demand is fixed) to minimize the total weight of ingots used.

The ISP combines two features—selecting  $p$  standard sizes, and “packing” orders in ingots—that have been addressed separately in other contexts. Several papers have addressed the problem of selecting standard lengths to stock and the related assortment problem (e.g., Wolfson [1965], Pentico [1974], Beasley [1985], Pentico [1988]). This problem deals with selecting standard sizes (e.g., lengths) for semi-finished items from a given set of candidate sizes to meet demand for various finished sizes. The objective is to minimize scrap and/or inventory holding and substitution costs. This class of models does not consider order combination, i.e., it assumes that only one order can be produced from each semi-finished item. For the one-dimension case, say, selecting standard lengths of

steel rods, if we assume that a finished product with length  $l$  can be produced from any semi-finished item with length greater than or equal to  $l$ , the problem can be solved efficiently using dynamic programming (e.g., Pentico [1974]). Vasko, Wolf, and Stott [1987] developed and applied a set covering approach to select optimal ingot sizes for the steel industry. Again, their model did not incorporate the option of jointly producing multiple orders using a single ingot. The literature on cutting stock problems considers the issue of how to optimally “pack” multiple orders in available semi-finished sizes (e.g., Gilmore and Gomory [1961], [1963]). For instance, given rolls of certain widths and lengths, the cutting stock problem selects an optimal set of patterns to produce all orders. Chambers and Dyson [1976] and Beasley [1985] present heuristic algorithms for the two-dimensional cutting stock problem with stock size selection.

### 2.3 Problem formulation

Let  $I = \{1, \dots, |I|\}$  and  $K = \{1, \dots, |K|\}$  respectively denote the index sets of products and candidate ingot sizes. For all  $i \in I$ , let  $d_i$  and  $q_i$  denote the demand and weight per order for product  $i$ . For all  $k \in K$ , let  $w_k$  denote the total weight of ingot type  $k$ . For every pair of products  $i, j \in I$  with  $i \leq j$  and every ingot type  $k \in K$ , we apply the order combination rules to determine if the pair  $(i,j)$  can be produced using ingot type  $k$ . Let  $IJK$  denote the set of all *feasible assignments*  $(i,j,k)$  such that the pair of products  $(i,j)$  can be jointly produced using ingot type  $k$ . This set might include triplets such as  $(i,i,k)$  denoting the assignment of two orders of product  $i$  to one type  $k$  ingot. We also include the option of dedicating ingots: if a type  $k$  ingot can produce a single order of product  $i$ , we denote this triplet as  $(0,i,k)$ . (The index 0 represents a dummy product that can be combined with any single order, and has zero demand.) Notice that if, for some  $j \in I$ , the pair  $(i,j)$  is feasible using ingot type  $k$ , then the pair  $(0,i)$  must also be feasible using ingot type  $k$ . Let  $J(i)$  denote the set of all “**i-compatible**” products, i.e.,  $J(i) = \{j: (i,j,k) \text{ or } (j,i,k) \in IJK \text{ for some } k \in K\}$ .

To formulate the ingot sizing problem, we define the following two sets of integer variables:

$$z_k = \begin{cases} 1 & \text{if ingot type } k \text{ is chosen as a standard size, and} \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } k \in K; \text{ and,}$$

$$y_{ijk} = \text{number of type } k \text{ ingots used to produce product pair } (i,j), \text{ for all } (i,j,k) \in IJK.$$

For all  $i, j \in I$ , define  $\lambda_{ij} = \min \{d_i, d_j\}$ ; let  $\lambda_{0i} = d_i$  for all products  $i$ . The ingot sizing problem has the following integer programming formulation.

$$[\text{ISP}] \quad Z^* = \min \sum_{(i,j,k) \in \text{IJK}} w_k y_{ijk} \quad (2.1)$$

subject to:

*Demand constraints:*

$$2 \sum_{(i,i,k) \in \text{IJK}} y_{iik} + \sum_{j \neq i} \sum_{(i,j,k) \in \text{IJK}} y_{ijk} \geq d_i \quad \text{for all } i \in \text{I} \quad (2.2)$$

*Forcing constraints:*

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

*Maximum # of standard sizes:*

$$\sum_{k \in \text{K}} z_k \leq p, \text{ and} \quad (2.4)$$

*Integrality/nonnegativity:*

$$z_k = 0 \text{ or } 1 \quad \text{for all } k \in \text{K} \quad (2.5a)$$

$$y_{ijk} \geq 0 \text{ and integer} \quad \text{for all } (i,j,k) \in \text{IJK} \quad (2.5b)$$

The objective function (2.1) minimizes the total weight of ingots used to satisfy the demand for all products. Constraints (2.2) ensure that the demand for every product is satisfied. The first term in the left-hand side of this equation represents the number of ingots used to process the product-pair (i,i) if this pair is feasible for any ingot; each of these ingots generates satisfies two units of demand for product i. The second term in the left-hand side corresponds to the number of ingots containing one unit each of product i (either dedicated or joint with some product  $j \neq i$ ). The forcing constraint specifies that two products i and j, with  $(i,j,k) \in \text{IJK}$ , can be jointly produced using ingot type k only if this ingot type is chosen as a standard size. In this constraint, limiting the number of type k ingots used to produce the pair (i,j) to at most  $\lambda_{ij}$  ( $= \min \{d_i, d_j\}$ ) is valid since  $(i,j,k) \in \text{IJK}$  implies that  $(0,i,k)$  and  $(0,j,k)$  also belong to the set IJK. Therefore, if a solution assigns  $\lambda_{ij} + \delta$  ingots to the pair, (i,j) we can replace  $\delta$  of these pairs with single orders of product i or product j depending on whether  $d_i \geq d_j$ . This change retains feasibility of the solution, and does not increase the objective function value. Constraint (2.4) imposes the upper limit of p on the number of standard sizes we select. We assume that [ISP] is feasible (for instance, we can add a “large” ingot size to the candidate list such that dedicating this ingot is feasible for every product).

The ISP is NP-hard since, as we note in the next section, the p-median problem (see, for instance, Francis and Mirchandani [1990]) is a special case. Formulation [ISP] requires

prior enumeration of all feasible product pairs  $(i,j)$  for all candidate ingot sizes. Consequently, the number of variables  $y_{ijk}$  can be quite large. Instead, we might consider directly encoding the product-pair feasibility rules as constraints in the problem formulation. For instance, we can represent the problem as one of assigning products to individual ingots, and checking the feasibility of assignments via constraints corresponding to weight feasibility, width feasibility, gauge differential, and width differential. Our solution procedure exploits the special structure of [ISP].

## 2.4 Special cases

Formulation [ISP] has two interesting special cases. First, suppose we do not permit joint production, or the feasibility constraints are such that the pair  $(0,i)$  is the only feasible "combination" for every product  $i$ . In this case, we can treat the products as "customers" and candidate ingot sizes as "facilities" in a  $p$ -median problem. An ingot type  $k$  can serve product  $i$  only if  $(0,i,k) \in IJK$ . Note that, for a given choice of standard ingot sizes, the problem has an optimal solution that assigns all  $d_i$  units of product  $i$  to the same ingot type  $k$ . The total surplus and planned scrap generated by this assignment,  $d_i(w_k - q_i)$ , is the "cost" of assigning product  $i$  to ingot type  $k$ . The  $p$ -median model selects at most  $p$  ingots, and assigns products to ingots to minimize the total assignment cost.

The second special case, which we refer to as the *order assignment* problem, corresponds to the situation when the standard ingot sizes are prespecified, or when  $|K| \leq p$  (so [ISP] has an optimal solution with  $z_k = 1$  for all  $k \in K$ ). We can find the optimal  $y$ -values for this special case by solving a non-bipartite matching problem defined over the following network  $G$ ; we later exploit this property in our heuristic solution procedure. Figure 1 shows the network  $G$  for a 3 product, 3 ingot example. The network contains  $d_i$  nodes corresponding to each product  $i$ . We denote these nodes as  $(i,l)$  for  $l = 1, \dots, d_i$ . If the pair  $(i,i)$  is feasible using one of the available ingots, and if producing this pair is more economical than dedicating two ingots to produce individual orders of product  $i$ , then we add a single "dummy" node  $(i,d_i+1)$ . Otherwise, we say the pair  $(i,i)$  is dominated; in this case, we add  $d_i$  dummy nodes  $(i,l)$  for  $l = d_i+1, \dots, 2d_i$ . For every feasible pair  $(i,j)$  (using the available ingots) with  $i \neq 0$ , we connect every node  $(i,l)$  for  $l = 1, \dots, d_i$  to every node  $(j,l_2)$  for  $l_2 = 1, \dots, d_j$ . The cost of this edge is the weight of the best ingot for the pair  $(i,j)$ , i.e., the ingot  $k$  with the lowest weight such that  $(i,j,k) \in IJK$ ; let  $k_{ij}$  denote the index of this ingot type. We also connect each node  $(i,l)$  for  $l = 1, \dots, d_i$  to every dummy node  $(i,l_2)$  corresponding to product  $i$ ; the cost of each of these edges is the weight of the lowest

weight ingot  $k_{0i}$  that can produce a single order of product  $i$ . Finally, for each  $i \in I$ , all dummy nodes corresponding to product  $i$  are connected to each other with zero-cost edges.

The minimum-cost matching over  $G$  provides an optimal solution to the order assignment problem. If a node  $(i, l)$  is connected to a node  $(j, l')$  in the optimal matching, then we assign one ingot of type  $k_{ij}$  to the pair  $(i, j)$ . If node  $(i, l)$  is connected to one of product  $i$ 's dummy nodes in the optimal matching, then we assign an ingot of type  $k_{0i}$  to produce a single order of product  $i$ . The total cost of the optimal matching equals the optimal value of the order assignment problem. Finally, we can reduce the number of edges in  $G$  by noting that for any subset of  $i$ -compatible products  $J' \subseteq J(i) \setminus \{0, i\}$ , the number of ingots assigned to produce the pairs  $(i, j)$  for all  $j \in J'$  must not exceed the sum of the demands for all products in  $J'$ . Suppose we index the products in  $J(i) \setminus \{0, i\}$  in increasing order of demand; then, for any  $j \in J(i)$ , we need to connect only the nodes  $(i, l)$  for  $l = 1, \dots, \min \{d_i, \sum_{j' \leq j} d_{j'}\}$  to all the nodes  $(j, l')$ , for  $l' = 1, \dots, d_j$ . This observation reduces considerably the number of edges in  $G$  if a few products have high demands while others have relatively low demands.

### 3. Solving the Ingot Sizing Problem

Since the ISP model is NP-hard, we focus on developing effective optimization-based heuristics and lower bounds. This section describes a dual ascent method to generate lower bounds on the optimal value, and heuristic methods (including a dual-based method) to identify feasible solutions. These methods exploit the special covering and  $p$ -median features of formulation [ISP]. The gap between the lower bound and the best heuristic solution value provides an assessment of the quality of the heuristic solution.

#### 3.1 Dual ascent procedure

Dual ascent refers to a broad class of heuristic strategies to generate lower bounds as well as feasible solutions by approximately solving the linear programming dual. This technique has been successfully applied to several difficult discrete optimization problems including the plant location problem (Erlenkotter [1978]), the Steiner tree problem (Wong [1984]), set packing problems (Fisher and Kedia [1986]), and the uncapacitated network design problem (Balakrishnan, Magnanti, and Wong [1989]).



For the ISP, the value of any feasible LP dual solution provides a lower bound on the optimal value  $Z^*$ . Starting with a specific set of initial dual values, the dual ascent procedure iteratively modifies these values in a myopic fashion to monotonically increase the dual objective function value. The procedure terminates when no further myopic improvement is possible. The final dual solution not only provides a lower bound, but also suggests a partial solution to the original problem.

Consider the linear programming relaxation of formulation [ISP] obtained by replacing the integrality restrictions (2.5a) and (2.5b) on the  $z$  and  $y$  variables with nonnegativity constraints. Let  $u_i$  for all  $i \in I$ ,  $v_{ijk}$  for all  $(i,j,k) \in IJK$ , and  $\alpha$  be the dual variables corresponding to the demand constraints (2.2), the forcing constraints (2.3), and the maximum standard sizes constraint (2.4). Let  $IJ(k) = \{(i,j): (i,j,k) \in IJK\}$  denote the set of all order pairs  $(i,j)$  that ingot  $k$  can produce. The linear programming dual [DISP] of formulation [ISP] has the following form:

$$\text{[DISP]} \quad Z_D = \max \sum_{i \in I} d_i u_i - p \alpha \quad (3.1)$$

subject to:

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

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

$$u_i \geq 0 \quad \text{for all } i \in I, \quad (3.4a)$$

$$v_{ijk} \geq 0 \quad \text{for all } (i,j,k) \in IJK, \quad (3.4b)$$

$$\alpha \geq 0. \quad (3.4c)$$

Note that constraints (3.2) corresponding to triplets  $(i,i,k)$  have the form:

$$2 u_i - v_{iik} \leq w_k.$$

To simplify the notation, we assume that formulation [DISP] also contains a dual variable  $u_0$  corresponding to the dummy product 0, but we fix the value of  $u_0$  at zero.

Our dual ascent method exploits the following observations. Suppose we are given nonnegative values for all the  $v$ -variables, say,  $\hat{v}_{ijk}$  for all  $(i,j,k) \in IJK$ . We wish to “complete” this partial solution  $\hat{V}$ , i.e., determine values for  $\alpha$  and the  $u$ -variables that are feasible in [DISP] for the specified  $v$ -values, and maximize the dual objective function

value. Since  $p > 0$  and since the objective function coefficient of  $\alpha$  is  $-p$ , the “best” value of  $\alpha$  is the smallest value necessary to ensure feasibility of constraints (3.3), i.e.,

$$\hat{\alpha} = \text{Max}_{k \in K} \sum_{(i,j) \in U(k)} \lambda_{ij} \hat{v}_{ijk}. \quad (3.5)$$

Determining the best values of the  $u$ -variables is not as easy; however, we can identify and exploit some properties that these values must satisfy. Let  $\mathbf{K}(i,j)$  denote the subset of ingots that can produce the pair  $(i,j)$ . Since the  $u$ -values have positive coefficients in the maximization objective function, the best values of the  $u$ -variables must be locally maximum, i.e., we cannot increase one  $u$ -value without decreasing another. In particular, for every product  $i \in I$ , at least one constraint (3.2) containing the variable  $u_i$  must be “tight”, i.e., if  $\hat{U} = (\hat{u}_i)$  is an optimal completion of the partial solution  $(\hat{V}, \hat{\alpha})$  then:

$$\hat{u}_i + \hat{u}_j = w_k + \hat{v}_{ijk} \quad \text{for at least one } j \in J(i) \text{ and } k \in K(i,j). \quad (3.6)$$

Otherwise, we can increase some  $u$ -value without violating constraints (3.2); the new solution has a higher dual objective function value. For all  $i \in I$  and  $j \in J(i)$ , let

$$\delta_{ij} = \min_{k \in K(i,j)} (w_k + \hat{v}_{ijk}). \quad (3.7)$$

Condition (3.6) implies that

$$(\hat{u}_i + \hat{u}_j) = \delta_{ij} \quad \text{for all } i \in I \text{ and } j \in J(i). \quad (3.8)$$

For convenience, if the pair  $(i,i)$  is not feasible using any candidate ingot size, we will assume  $\delta_{ii} = \infty$ . Our dual ascent procedure iteratively changes the  $v$ -values, and heuristically selects  $u$ -values to satisfy condition (3.8). We first describe how to select initial values of the dual variables before discussing the multiplier adjustment methods and the dual-based heuristic solution.

### Initial dual solution

To construct an initial dual feasible solution, we select  $\hat{v}_{ijk} = 0$  for all  $(i,j,k) \in IJK$ . Consequently,  $\hat{\alpha} = 0$ . Assume, for convenience, that the products are indexed in order of decreasing demand. To heuristically maximize the dual objective function value (for the given  $\hat{v}$ -values), we consider each product  $i$  in this decreasing-demand sequence, and set  $\hat{u}_i$  to its highest possible value given the prior choice of values  $\hat{u}_j$  for  $j = 1, \dots, i-1$ . In particular, we set:

$$\hat{u}_i = \min \left\{ \min_{j < i \text{ with } j \in J(i)} (\delta_{ij} - \hat{u}_j); \frac{\delta_{ii}}{2}; \min_{j > i \text{ with } j \in J(i)} \delta_{ij} \right\}. \quad (3.9)$$

This procedure ensures that the dual solution is feasible and satisfies condition (3.8).

The following procedure summarizes this dual initialization method.

### Dual Initialization Method:

Step 1: Index the products from 1 to III in non-increasing order of demand  $d_i$ .

$$\begin{aligned} \text{Set } \hat{v}_{ijk} &\leftarrow 0 \text{ for all } (i,j,k) \in IJK, \\ \hat{\alpha} &\leftarrow 0, \text{ and} \\ \hat{Z}_D &\leftarrow 0. \end{aligned}$$

Step 2: For  $i = 1, \dots, III$ , let

$$\mu_1 = \min_{j < i, j \in J(i)} \left\{ \min_{k \in K(i,j)} [w_k] - \hat{u}_j \right\};$$

$$\mu_2 = \frac{\min_{k \in K(i,i)} [w_k]}{2} \quad \text{if } i \in J(i), \text{ and } \infty \text{ otherwise};$$

$$\mu_3 = \min_{j > i, j \in J(i)} \left\{ \min_{k \in K(i,j)} [w_k] \right\};$$

$$\text{Set } \hat{u}_i \leftarrow \min \{ \mu_1, \mu_2, \mu_3 \}, \text{ and}$$

$$\hat{Z}_D \leftarrow \hat{Z}_D + d_i \hat{u}_i;$$

Next  $i$ ;

When this procedure terminates, the final value of  $\hat{Z}_D$  is our *initial lower bound* on the optimal value  $Z^*$  of problem [ISP]. Next, we discuss how to improve this lower bound by changing  $v$ -values, first one at a time and subsequently two values simultaneously.

### Ascent procedure Phase 1: Changing $v$ -values one at a time

Increasing certain  $v$ -values might permit us to selectively increase  $u$ -values, and hence the dual objective value. However, this increase might be offset by objective value reductions due to required changes in other variables. For instance, increasing a particular  $v$ -value might also require increasing  $\alpha$  to maintain feasibility in constraints (3.3).

Likewise, when we increase one  $u$ -value, we might have to decrease other  $u$ -values to maintain feasibility in constraints (3.2). Therefore, we must judiciously select the values to increase so as to produce a net increase in the objective function value. The first phase in our dual ascent procedure increases just one  $v$ -value at each iteration. We next motivate and describe this procedure.

Let us first understand the implications of increasing the current value of a specific  $v$ -variable, say,  $v_{ijk}$ . To facilitate our subsequent discussions, we define:

**A** = Set of all triplets  $(i,j,k)$  for which constraints (3.2) are tight.

**B** = Set of ingot types  $k$  for which constraints (3.3) are tight.

First, note that if the constraint (3.2) corresponding to a triplet  $(i,j,k)$  is not currently tight, then the current  $u_i$  and  $u_j$  values must be restricted by some other constraint(s) (3.2) (assuming that these current values satisfy condition (3.8)). Therefore, increasing  $v_{ijk}$  does not create new opportunities to increase either  $u_i$  or  $u_j$ . So, we consider only triplets  $(i,j,k)$  in the set **A**.

Suppose we increase  $v_{ijk}$  for one such triplet by  $\Delta$  units. If  $k \in \mathbf{B}$ , then  $\alpha$  must correspondingly increase by  $\lambda_{ij}\Delta$  units to preserve feasibility of constraints (3.3), in which case the dual objective function value decreases by  $p\lambda_{ij}\Delta$ . Otherwise (i.e.,  $k \notin \mathbf{B}$ ), if  $s_k$  denotes the current slack in constraint (3.3) corresponding to ingot type  $k$ ,  $\alpha$  is unchanged if we set  $\Delta$  to be no more than  $s_k/\lambda_{ij}$ .

Increasing  $v_{ijk}$  by  $\Delta$  introduces a slack of  $\Delta$  units in the corresponding constraint (3.2) which we can use to increase either  $u_i$  or  $u_j$ . (If  $i = j$ , we can increase  $u_i$  by only  $\Delta/2$  units.) Suppose we select  $u_i$  as the candidate variable to increase. Increasing its value by  $\Theta$  ( $= \Delta$  if  $i \neq j$ , and  $\Delta/2$  if  $i = j$ ) increases the dual objective function value by  $d_i\Theta$  units. But, we must also ensure that all other constraints (3.2) remain feasible. Consider any triplet  $(i,j',k') \in \mathbf{A} \setminus (i,j,k)$ . Since the corresponding constraint (3.2) contains  $u_i$  in its left-hand side and is tight, increasing  $u_i$  by  $\Theta$  requires decreasing  $u_{j'}$  or increasing  $v_{ij'k'}$  by  $\Theta$ . Since Phase 1 increases only one  $v$ -value at a time, we ignore the latter option. Also, if  $j' = j$ , i.e., if two or more constraints (3.2) corresponding to the product pair  $(i,j)$  are tight, then increasing  $u_i$  requires decreasing  $u_j$  which implies that the original increase in  $v_{ijk}$  is unnecessary. So, we focus on product pairs  $(i,j)$  for which exactly one constraint (3.2) is tight. Since the  $u$ -values must all be nonnegative, we cannot reduce  $u_{j'}$  by more than its current value  $\hat{u}_{j'}$  i.e.,  $\Theta$  must be less than or equal to  $\hat{u}_{j'}$  for all  $j'$  such that  $(i,j',k') \in \mathbf{A} \setminus (i,j,k)$  for some  $k' \in \mathbf{K}$ .

For each such product  $j'$ , reducing the value of  $u_j$  decreases the objective function value by  $d_j \cdot \Theta$ . Finally, for triplets  $(i, j', k') \notin A \setminus (i, j, k)$ , increasing  $u_i$  does not require changes in other  $u$ -values as long as  $\Theta$  is less than the current slacks in constraints (3.2) corresponding to these triplets.

To summarize, we consider increasing one  $v_{ijk}$  value, and if necessary (i.e., if  $k \in B$ ) correspondingly increasing  $\alpha$  to preserve feasibility of constraints (3.3). We increase  $u_i$  (or  $u_j$ ), and identify other variables  $u_j$  belonging to tight constraints shared with  $u_i$  (or  $u_j$ ) that must be reduced to preserve feasibility of constraints (3.2). Increasing  $u_i$  (or  $u_j$ ) increases the dual objective function value, but compensating changes in the other variables reduce this value. Furthermore, various factors limit the maximum possible change including the slacks in constraints (3.2) that involve  $u_i$  (or  $u_j$ ) but are not tight, the slack in constraint (3.3) corresponding to ingot type  $k$ , and the current values of the  $u$ -variables that must be decreased. We evaluate the net change in the dual objective value for each triplet  $(i, j, k) \in A$ , and select the dual adjustment that produces the maximum increase. If none of the adjustments produces a net increase in the dual objective function value, we terminate the first phase. A formal description of the Phase 1 procedure follows:

### Ascent procedure Phase 1:

#### Step 1:

$A$  = Set of triplets  $(i, j, k)$  for which constraints (3.2) are tight;

$B$  = Set of ingot types  $k$  for which constraints (3.3) are tight;

$s_k$  = slack in the constraint (3.3) corresponding to ingot  $k$ , for all  $k \in K$ .

For every  $(i, j, k) \in A$  with  $(i, j, k') \notin A$  for any other  $k' \in K(i, j)$ ,

For  $l = i, j$ ,

let  $M = \{m : (l, m, k') \text{ or } (l, m, k) \in A \setminus (i, j, k)\}$ .

**Calculate maximum allowable value of  $\Theta$ :**

$$\Theta_1 = \min_{m \in M} \hat{u}_m;$$

$$\Theta_2 = \min_{(i', j', k') \notin A, i' \text{ or } j' = l} (\hat{w}_k + \hat{v}_{ij'k'} - \hat{u}_{i'} - \hat{u}_{j'}); \text{ and}$$

$$\Delta_3 = \frac{s_k}{\lambda_{ij}} \text{ if } k \notin B, \text{ and } \infty \text{ otherwise.}$$

Set  $\Theta \leftarrow \min \{ \Theta_1, \Theta_2, \Delta_3 \}$  if  $i \neq j$ , or

$$\Theta \leftarrow \min \left\{ \Theta_1, \Theta_2, \frac{\Delta_3}{2} \right\} \quad \text{if } i = j.$$

Calculate net change in objective function value:

$$\sigma_{ijk}(l) = \begin{cases} \left\{ d_l - \sum_{m \in M} d_m \right\} \Theta - \lambda_{ij} p \Theta & \text{if } k \in B \text{ and } i \neq j, \\ \left\{ d_l - \sum_{m \in M} d_m \right\} \Theta - 2 \lambda_{ij} p \Theta & \text{if } k \in B \text{ and } i = j, \text{ and} \\ \left\{ d_l - \sum_{m \in M} d_m \right\} \Theta & \text{if } k \notin B. \end{cases}$$

next  $l$ ;

next  $(i,j,k)$ ;

**Step 2:**

Find the triplet  $(i,j,k)$  and  $l = i$  or  $j$  with maximum increase  $\sigma_{ijk}(l)$  in the dual objective function value.

If this maximum increase is not positive, STOP.

Else, update

$$\begin{aligned} \hat{u}_l &\leftarrow \hat{u}_l + \Theta; \\ \hat{u}_m &\leftarrow \hat{u}_m - \Theta \quad \text{for all } m \in M; \\ \hat{v}_{ijk} &\leftarrow \hat{v}_{ijk} + \Theta; \\ \text{if } k \in B \text{ and } i \neq j, &\hat{\alpha} \leftarrow \hat{\alpha} + \lambda_{ij} \Theta; \\ \text{if } k \in B \text{ and } i = j, &\hat{\alpha} \leftarrow \hat{\alpha} + 2\lambda_{ij} \Theta; \text{ and,} \\ \hat{Z}_D &\leftarrow \hat{Z}_D + \sigma_{ijk}(l). \end{aligned}$$

Return to Step 1.

### Ascent Procedure Phase 2: Increasing two v-values simultaneously

In Phase 2, we evaluate ascent opportunities by simultaneously increasing two v-values,  $v_{ijk}$  and  $v_{ijk'}$ , for the same product pair  $(i,j)$  but different ingot types  $k$  and  $k'$ . As before, we consider only triplets  $(i,j,k)$  and  $(i,j,k')$  that both belong to the set A of triplets for which constraints (3.2) are tight. The procedure for increasing the two v-values is very similar to the Phase 1 procedure, except that now we might have to decrease fewer  $u_m$  values (since increasing the two v-values by  $\Theta$  creates a slack of  $\Theta$  units in two tight constraints). If constraint (3.3) is tight for either  $k$  or  $k'$ , then we must increase  $\alpha$ .

Otherwise, the slack in these two constraints determines the maximum allowable increase  $\Theta$  in the v-values. Other considerations that limit this allowable increase are the current

values of the variables  $u_m$  that need to be reduced (to preserve feasibility of tight constraints involving  $u_l$ ), and the current slack in constraints that are not tight but contain  $u_l$ . We give below a formal description of this Phase 2 procedure.

### Ascent procedure Phase 2:

#### Step 1:

$A$  = Set of triplets  $(i,j,k)$  for which constraints (3.2) are tight;

$B$  = Set of ingot types  $k$  for which constraints (3.3) are tight;

$s_k$  = slack in the constraint (3.3) corresponding to ingot  $k$ , for all  $k \in K$ .

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

For  $l = i, j$ ,

let  $M = \{ m: (l, m, k'') \text{ or } (m, l, k'') \in A \setminus \{(i,j,k), (i,j,k')\} \}$

**Calculate maximum allowable value of  $\Theta$ :**

$$\Theta_1 = \min_{m \in M} \hat{u}_m;$$

$$\Theta_2 = \min_{(i',j',k') \in A, i' \text{ or } j' = l} (\hat{w}_{k'} + \hat{v}_{ij'k'} - \hat{u}_{i'} - \hat{u}_{j'}); \text{ and}$$

$$\Delta_3 = \begin{cases} \min(s_k, s_{k'})/\lambda_{ij} & \text{if } k \text{ and } k' \notin B, \\ s_k/\lambda_{ij} \text{ (or } s_{k'}/\lambda_{ij}) & \text{if } k \text{ (or } k') \notin B \text{ and } k' \text{ (or } k) \in B, \text{ and} \\ \infty & \text{otherwise.} \end{cases}$$

Set  $\Theta \leftarrow \min \{ \Theta_1, \Theta_2, \Delta_3 \}$  if  $i \neq j$ , or

$\Theta \leftarrow \min \{ \Theta_1, \Theta_2, \frac{\Delta_3}{2} \}$  if  $i = j$ .

**Calculate net change in objective function value:**

$$\sigma_{ijkk'}(l) = \begin{cases} \left\{ d_l - \sum_{m \in M} d_m \right\} \Theta - \lambda_{ij} p \Theta & \text{if } k \text{ or } k' \in B \text{ and } i \neq j, \\ \left\{ d_l - \sum_{m \in M} d_m \right\} \Theta - 2 \lambda_{ij} p \Theta & \text{if } k \text{ or } k' \in B \text{ and } i = j, \text{ and} \\ \left\{ d_l - \sum_{m \in M} d_m \right\} \Theta & \text{if } k \text{ and } k' \notin B. \end{cases}$$

next  $l$ ;

next  $(i,j,k)$  and  $(i,j,k')$ ;

**Step 2:**

Find the triplets  $(i,j,k)$ ,  $(i,j,k')$  and  $l = i$  or  $j$  with maximum increase  $\sigma_{ijkk'}(l)$  in the dual objective function value.

If this maximum increase is not positive, STOP.

Else, update

$$\hat{u}_l \leftarrow \hat{u}_l + \Theta;$$

$$\hat{u}_m \leftarrow \hat{u}_m - \Theta \quad \text{for all } m \text{ such that } (l, m, k') \in A \setminus (i,j,k);$$

$$\hat{v}_{ijk} \leftarrow \hat{v}_{ijk} + \Theta;$$

$$\hat{v}_{ijk'} \leftarrow \hat{v}_{ijk'} + \Theta;$$

$$\text{if } k \text{ or } k' \in B \text{ and } i \neq j, \hat{\alpha} \leftarrow \hat{\alpha} + \lambda_{ij} \Theta; \text{ and,}$$

$$\text{if } k \text{ or } k' \in B \text{ and } i = j, \hat{\alpha} \leftarrow \hat{\alpha} + 2\lambda_{ij} \Theta; \text{ and,}$$

$$\hat{Z}_D \leftarrow \hat{Z}_D + \sigma_{ijkk'}(l).$$

Return to Step 1.

Note that the same principle of increasing two  $v$ -values applies even to “unrelated” triplets  $(i,j,k)$  and  $(i',j',k')$  belonging to the set  $A$ , possibly with  $i \neq i'$  and  $j \neq j'$ . In this case, when we increase  $v_{ijk}$  and  $v_{i'j'k'}$ , we must also decide whether to increase  $u_i$  or  $u_j$  and  $u_{i'}$  or  $u_{j'}$ . In our implementation, we consider only the special case with  $i = i'$  and  $j = j'$ .

If Phase 2 changes any of the dual values, then there might be opportunities to reapply Phase 1 and further improve the dual objective function value. We continue alternating between the Phase 1 and Phase 2 procedures until no further improvement is possible. The final dual objective function value when the algorithm terminates is a lower bound on the optimal value  $Z^*$  of [ISP]. As we explain in the next section, we also use the final dual solution to construct a feasible solution to the ingot sizing problem.

## 3.2 Heuristic procedures

This section describes three methods to select the standard ingot sizes—a dual-based method, and two greedy heuristics. For each choice of standard ingot sizes, we solve the order assignment problem (using a non-bipartite matching algorithm as described in Section 2), and select the best solution as the ISP heuristic solution.

### 3.2.1 Dual-based heuristic solution

When the dual ascent procedure terminates, we try to construct a primal feasible solution using complementary slackness conditions. If, in the final dual solution, the constraint (3.3) corresponding to ingot type  $k$  is tight, then by complementary slackness



ingot type  $k$  is a candidate for being included in the set of standard sizes. Hence, if the dual ascent procedure ends with  $p$  or fewer constraints in the set (3.3) being tight, we choose the ingots corresponding to these tight constraints as the standard ingots. We then determine the optimal combinations and the actual solution value by solving the non-bipartite matching subproblem as described in the Section 2. If more than  $p$  constraints in (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 greedy heuristics that we describe next to select a subset of standard ingot sizes from this restricted set of candidate ingots.

Given the set of candidate ingot sizes, the following two heuristics—the *ingot utilization* heuristic and the *greedy assignment* heuristic—select standard ingot sizes one at a time to heuristically minimize the total weight of ingots used. The heuristics differ in the order in which they consider the candidate sizes.

### 3.2.2 Ingot Utilization heuristic

This heuristic attempts to select standard ingot sizes that can satisfy a large proportion (in terms of weight) of the demand at high *utilization* levels. If we use an ingot type  $k$  to produce the pair  $(i,j)$ , the *utilization* of the ingot is the total weight  $(q_i+q_j)$  of the two orders divided by the gross weight  $w_k$  of ingot  $k$ . Ideally, the chosen standard sizes and combinations should be such that every ingot is nearly fully utilized. For this purpose, we specify a *threshold utilization level*  $\beta\%$  ( $0 \leq \beta \leq 100$ ) representing the minimum acceptable ingot utilization. By parametrically varying  $\beta$ , we can generate multiple solutions.

The algorithm iteratively evaluates the total demand (in pounds) that every remaining candidate ingot size can satisfy subject to the threshold utilization requirement, and selects the size that satisfies the maximum demand. At any iteration, let  $K^*$  and  $\hat{K}$  denote the current set of chosen and remaining candidate ingot sizes. The algorithm progressively assigns demand for various products to the chosen sizes. Let  $\hat{d}_i$  denote the residual demand (number of orders) for each product  $i \in I$ . If  $\hat{\lambda}_{ij} = \min \{\hat{d}_i, \hat{d}_j\}$ , we define the *volume* of a product pair  $(i,j)$  as  $M(i,j) = \hat{\lambda}_{ij} (q_i+q_j)$ . Let  $IJ(k)$  be the set of all product pairs that utilize a type  $k$  ingot  $\beta\%$  or more, i.e.,  $IJ(k) = \{(i,j): (i,j,k) \in IJK \text{ and } (q_i+q_j) \geq \beta w_k/100\}$ . Let  $W(k)$  be the sum of volumes for all pairs  $(i,j) \in IJ(k)$ . The ingot utilization heuristic selects the ingot  $k$  with maximum  $W(k)$  as the next candidate size, assigns all the product pairs  $(i,j) \in IJ(k)$  to this ingot, updates the residual demands, and repeats the procedure until  $p$  candidate sizes are chosen. If at any stage,  $W(k)$  is zero for every remaining candidate size

$k \in \hat{K}$  and if the demands are not completely satisfied, then we reduce the threshold value  $\beta$  by a fixed percentage (10% in our computations) and continue the algorithm.

### Ingot utilization heuristic:

**Step 0:** Initialize  $\hat{K} \leftarrow$  set of candidate ingot sizes

$$K^* \leftarrow \phi$$

$$\hat{d}_i = d_i \text{ for all products } i \in I.$$

$$q_i = \text{weight per order of product } i.$$

**Step 1:** For every  $k \in \hat{K}$ ,

$$IJ(k, \beta) = \text{Set of all feasible pairs } (i, j) \in IJ(k) \text{ with } (q_i + q_j) \geq \frac{\beta}{100} w_k.$$

$$\hat{\lambda}_{ij} = \min \{ \hat{d}_i, \hat{d}_j \} \quad \text{for all } (i, j) \in IJ(k, \beta).$$

$$\text{Compute } W(k) = \sum_{(i, j) \in IJ(k, \beta)} \hat{\lambda}_{ij} (q_i + q_j)$$

next  $k$ ;

Let  $k' = \text{argmax} \{ W(k) : k \in \hat{K} \}$ .

If  $W(k') = 0$ , update  $\beta \leftarrow 0.90 * \beta$ , and repeat Step 1.

Else, select ingot type  $k'$  as a standard ingot, i.e., update

$$K^* \leftarrow K^* \cup \{k'\};$$

$$\hat{K} \leftarrow \hat{K} \setminus \{k'\};$$

$$\hat{d}_i \leftarrow \max \{ \hat{d}_i - \hat{\lambda}_{ij}, 0 \} \text{ and } \hat{d}_j \leftarrow \max \{ \hat{d}_j - \hat{\lambda}_{ij}, 0 \} \text{ for all } (i, j) \in IJ(k, \beta).$$

If  $|K^*| = p$  or if  $\hat{d}_i = 0$  for all  $i \in I$ , STOP. Else, repeat Step 1.

**Step 2:** For the chosen standard ingot sizes  $K^*$ , solve the order assignment problem to determine the optimal combinations.

### 3.2.3 Greedy Assignment heuristic

This heuristic attempts to first select good ingots for high-volume combinations. It considers product pairs  $(i, j)$  in decreasing order of total weight, and selects the best ingot type for each pair in sequence. As before, let  $\hat{d}_i$  denote the residual demand for product  $i$  at any intermediate iteration. Let  $\hat{U}$  denote the set of feasible product pairs  $(i, j)$  such that the residual demand for both  $i$  and  $j$  are positive. Define  $\hat{\lambda}_{ij} = \min \{ \hat{d}_i, \hat{d}_j \}$  and volume  $M(i, j) = \hat{\lambda}_{ij} (q_i + q_j)$  for every pair  $(i, j)$ . Let  $\hat{K}$  denote the current set of available candidate ingot sizes. We select the pair  $(i', j') \in \hat{U}$  with maximum volume  $M(i', j')$ , and determine the best ingot  $k' \in \hat{K}$  for this combination, i.e., the ingot for which the scrap and surplus  $(w_{k'} - q_i -$

$q_{j'})$  is minimum among all ingot types  $k \in \hat{K}$ . We add ingot type  $k'$  to the current set of standard ingots, reduce the demands for products  $i'$  and  $j'$ , and repeat the procedure until  $p$  ingots have been selected or the residual demand for every product is zero. This heuristic strategy might be useful when the demand distribution is skewed, i.e., the demand for a few products far exceeds other demands.

### Greedy Assignment heuristic:

**Step 0:** Initialize  $\hat{K} \leftarrow$  set of candidate ingot sizes

$$K^* \leftarrow \phi$$

$$\hat{d}_i = d_i \text{ for all products } i \in I$$

Let  $q_i$  denote the weight per order of product  $i$ .

**Step 1:**  $\hat{U} =$  set of all feasible pairs  $(i,j)$  with  $\hat{d}_i$  and  $\hat{d}_j > 0$ .

$$\hat{\lambda}_{ij} = \min \{ \hat{d}_i, \hat{d}_j \} \text{ for all } (i,j) \in \hat{U}.$$

$$M(i,j) = \hat{\lambda}_{ij} \{q_i + q_j\} \text{ for all } (i,j) \in \hat{U}.$$

$$(i',j') = \operatorname{argmax} \{ M(i,j) : i,j \in \hat{U} \}.$$

$$k' = \operatorname{argmin} \{ (w_k - q_{i'} - q_{j'}) : k \in \hat{K} \}.$$

Select ingot type  $k'$  as a standard ingot, i.e., update,

$$K^* \leftarrow K^* \cup \{k'\};$$

$$\hat{K} \leftarrow \hat{K} \setminus \{k'\};$$

$$\hat{d}_{i'} \leftarrow \hat{d}_{i'} - \hat{\lambda}_{i'j'}, \text{ and } \hat{d}_{j'} \leftarrow \hat{d}_{j'} - \hat{\lambda}_{i'j'}.$$

If  $|K^*| = p$  or if  $\hat{d}_i = 0$  for all  $i \in I$ , STOP. Else, repeat Step 1.

**Step 2:** For the standard ingot sizes  $K^*$ , solve the order assignment problem to determine the optimal combinations.

Several variants of these heuristics are possible. For instance, in the ingot utilization heuristic, we might dynamically update the residual demand as we compute the total weight  $W(k)$  of each candidate ingot. Likewise, in the greedy assignment method, for each candidate ingot  $k$  we might consider the total weight of all combinations for which ingot  $k$  is the best available ingot. We might also apply local improvement procedures to reduce the total cost of each of these three heuristic solutions—the dual-ascent, ingot utilization, or demand-based solutions. For instance, one procedure consists of considering each current standard size in turn and evaluating the potential savings by replacing this size with a neighboring (either larger or smaller) size. If none of these “swaps” produces any

savings, the procedure terminates; otherwise, we perform the swap with maximum savings, and repeat the procedure. To evaluate the savings for each swap, we solve the matching problem to compute the total cost with the new proposed set of standard sizes (the total cost is infinite if the new set of ingots cannot produce all products), and compare this cost with the current cost.

In our computational experiments, we found that the best solution produced by the three heuristics was itself quite close to optimality. So, we did not implement any local improvement procedure.

## 4. Computational Results

We implemented and tested the dual ascent procedure and heuristics using data on actual orders from a leading aluminum sheet manufacturer. We classified the orders into 4 distinct product groups, and separately solved the ingot sizing problem for each group. We analyzed various scenarios obtained by varying the number of candidate ingot sizes, the number of standard sizes required, and the maximum width differential. We also compared the results with current practice and with restricted order combination options. This section describes some implementation details and summarizes our computational results.

### 4.1 Implementation details and test problems

We implemented the dual ascent procedure and heuristics in FORTRAN on an IBM 4381 computer. To solve the order assignment subproblem, we used Derigs' [1988] non-bipartite matching algorithm.

We developed four test problems from a database of actual orders for a popular product family (sheet products with gauges in a specified range and requiring a particular alloy) over a 12-month period. The database contained over 500 "small" orders (i.e., orders for which dedicating an ingot was not considered economical). We first classified these orders into 4 natural product classes based on their widths. For one of these classes (Problem 3), all products have the same width  $w$ ; approximately 50% of the orders had this width. We divided the range of widths below  $w$  into two classes such that each class has approximately equal number of products, and treat the range of widths above  $w$  as the fourth class. Since products with large width differences (greater than the maximum width differential parameter  $\omega$ ) cannot be combined, partitioning the data set into these four

product classes does not eliminate many feasible pairs. Within each product class, we selected appropriate intervals of widths, gauges, and weights to group the orders into products. The width and weight intervals were defined by rounding up the actual order widths and weights to the nearest inch and thousand pounds, respectively. The gauge intervals corresponded to non-overlapping ranges from the table of feasible gauge combinations (see Gopalakrishnan [1994] for additional details regarding the test problems).

The facility currently produces ingots in 3 widths and a total of 6 standard sizes (width and weight combinations). For each problem, we generated two sets of candidate ingot sizes—a *limited* set and an *extended* set. The limited set has both fewer candidate widths and fewer candidate weights for every width compared to the extended set. The number of candidate weights for a particular width was chosen to reflect the relative demand for that width. For instance, if many products have a width close to a candidate ingot width  $w_1$ , then for this width we might select candidate weights at 2,500 pound intervals in the range of ingot weights that the ingot plant can produce; on the other hand, less popular widths might have candidate weights at 10,000 pound intervals. Table 1 shows the number of products and the number of candidate ingot sizes for both the limited and extended sets for each of the four problems.

For the order combination constraints, we used the actual gauge combination table supplied by the manufacturer. Our “standard” problems assume a maximum width differential  $\omega$  of 2 inches, i.e., we cannot combine two orders that differ in width by more than 2 inches. We subsequently performed sensitivity analysis with respect to  $\omega$ . To compute the planned scrap allowances, we used the actual values of scalping loss, head and tail scrap, and trim loss used by the planners. For each problem, we first generate all the feasible combinations for every candidate ingot size based on the order combination constraints, i.e., subject to the total weight, width, width differential, and gauge differential restrictions. The demand for each product is the number of orders belonging to the corresponding product group in our data set. Given the product demands, candidate ingot sizes, feasible combinations, we obtain lower and upper bounds for various values of  $p$  by applying the dual ascent procedure and heuristics. The integer programming formulation for our largest test problem has 11096 variables and 11166 constraints.

## 4.2 Framework for experimentation

Our computational results seek to address two issues: (i) how effective are the dual ascent and heuristic procedures in generating bounds and solutions; and, (ii) what is the economic impact of changing the problem parameters. In all, we applied the dual ascent and heuristic methods to over 80 problem instances.

We use the % gap between the upper and lower bounds, defined as  $(\text{upper bound} - \text{lower bound}) * 100 / \text{lower bound}$ , to assess algorithmic effectiveness. Low values of % gap indicate that both the dual ascent lower bound is tight, and the heuristic procedures perform well. For each problem instance, we select the best among the upper bounds obtained by the heuristic procedures. Recall that the ingot utilization heuristic requires a user-specified threshold utilization value  $\beta$ . We experimented with various values of  $\beta$  for our standard test problems to identify effective settings for subsequent experimentation. Section 4.3 reports these results on algorithmic effectiveness and compares the performance of the heuristic solution procedures.

Section 4.4 illustrates how the ISP model and solution methodology can be used to perform various types of sensitivity analyses. In particular, we explore the sensitivity of the results to three problem features—the set of candidate ingot sizes, the number of standard ingot sizes, and the maximum width differential parameter. We also compare the total scrap and surplus when we permit order combinations with the results for two restrictive scenarios: when we only permit dedicated ingots (i.e., one order per ingot), and if we limit the combinations to identical orders. Finally, we compare the objective value for the ISP solution with the corresponding value for the current set of standard ingot sizes (but with order combination permitted).

## 4.3 Algorithmic effectiveness

We first compare the performance of various heuristics and determine an effective range for the parameter  $\beta$  in the ingot utilization heuristic before assessing the quality of the lower bounds.

### 4.3.1 Heuristic performance

To evaluate heuristic performance, we consider only problem instances with the “standard” width differential of 2 inches and the limited set of candidate ingot sizes, but several values for  $p$ . For the ingot utilization heuristic, we initially tried several threshold utilization levels  $\beta$  ranging from 50% to 90%. Let us first understand how this parameter

might affect the quality of the solution. For high values of  $\beta$ , relatively few pairs can utilize an ingot  $\beta\%$  or higher. Consequently, in the initial stages, the method might select ingots that satisfy only a small proportion of the total demand. Subsequently, when this threshold is no longer attainable for any remaining candidate ingot, the algorithm lowers the threshold thus permitting less effective ingot sizes. At the other extreme, for low values of  $\beta$ , many pairs will achieve or exceed this threshold for every candidate ingot. However, assigning these pairs reduces the overall average ingot utilization for the final set of standard sizes. Therefore, for either extreme, we expect that the heuristic solution will deteriorate. Our initial experiments confirmed this conjecture. For all our initial test problems, the heuristic solution with  $\beta = 65, 70$  or  $80\%$  was always better than the solutions with  $\beta = 50$  or  $90\%$ . So, we report results only for the former three values.

Table 2 shows the objective values for all four test problems for both the ingot utilization heuristic, with  $\beta = 65, 70$ , and  $80\%$ , and for the greedy assignment heuristic. Computation times (not reported here) for the heuristics were modest. Problems 1, 2, and 4 with a few hundred feasible assignments required 0.3 to 4.5 CPU seconds on an IBM 4381, whereas Problem 3 with several thousand feasible combinations required 30 to 50 CPU seconds. The results in Table 2 indicate that none of the methods is consistently superior. Out of the 16 problem instances, the ingot utilization heuristic with  $\beta = 65\%$  and  $\beta = 80\%$  produced the best solution in 5 instances each; the greedy assignment heuristic and ingot utilization heuristic with  $\beta = 70\%$  produced the best solution in 3 problem instances each. In general, the ingot utilization heuristic performs better than the greedy assignment method. The greedy method selects ingot sizes based on the residual volume of a single product pair, whereas the ingot utilization heuristic considers multiple product pairs that the ingot can produce. Indeed, the greedy assignment heuristic might sometimes select small ingot sizes initially, making certain low-demand but "heavy-weight" products infeasible at the end (after  $p$  standard sizes are selected). This phenomenon is apparent for Problem 3, where the greedy heuristic failed to produce a feasible set of standard sizes for values of  $p$  less than 6. In contrast, the ingot utilization heuristic has an adaptive feature (reducing the threshold utilization level) to avoid infeasible solutions. Finally, in all of our computational results, we observed that the dual-based heuristic solution often coincided with the ingot utilization or greedy assignment solutions, but in no case was the dual-based solution superior. We, therefore, do not separately report the results of this heuristic method.

For our subsequent experiments, we applied the ingot utilization heuristic, for all three values of  $\beta$ , as well as the greedy assignment heuristic, and selected the best heuristic solution value as the upper bound.

### 4.3.2 Dual ascent performance

Tables 3 through 6 present the results for Problems 1 through 4, respectively, for various  $p$  values and width differentials (except Problem 3 in which all products have the same width) with both the limited and extended set of candidate ingot sizes. The tables provide information on the respective problem sizes (# of feasible assignments) as well as the % of orders processed using dedicated ingots in the best heuristic solution. The low % gaps in these tables show that the dual ascent lower bounds are tight, and the heuristic solutions are near-optimal. In almost all problem instances, the % gap is below 7%. With the limited set of candidate sizes, the average gap between the upper and lower bounds for the four problems are 1.8%, 3.7%, 6.1%, and 4.7%, respectively. For the extended set of candidate sizes, the corresponding averages are 2.7%, 3.0%, 4.9%, and 5.9%. Overall, the procedure generates solutions that are (provably) within 4% on average.

In general, for a given width differential, the % gap tends to decrease as the number of standard sizes  $p$  increases (except for Problem 3). However, increasing the width differential or increasing the number of candidate ingot sizes (both of which increase the number of feasible combinations) does not produce a similar consistent effect, e.g., in Table 3 with the limited set of candidate ingots, for each value of  $p$ , the gap first decreases and then increases as the width differential increases.

## 4.4 Sensitivity analyses

We now illustrate how to use the ISP model and methodology for planning and decision support. In particular, we examine the sensitivity of the objective function value to three problem features:

- (i) the number of standard ingot sizes: Permitting more standard sizes enables the method to choose sizes that are closer to the order combinations; consequently, the objective function value should decrease;
- (ii) the set of candidate ingot sizes: With more candidate sizes, the model has greater choice in selecting a set of standard sizes, and hence we expect the total scrap and surplus to decrease;



(iii) the width differential: Permitting greater differences in widths for combined orders increases the number of feasible pairs, thus potentially reducing total pounds of metal processed.

Tables 2 through 6 confirm these effects. For instance, Table 2 shows that as the number  $p$  of standard ingot sizes increases from 3 to 6, the objective value decreases by up to 2.7% (for Problem 3), assuming the standard width differential (2 inches) and for the limited set of candidate ingot sizes. To assess the impact of increasing the set of candidate ingot sizes consider, for instance, Problem 3 with  $p = 4$ . The objective function decreases by 3.7% when we extend the set of candidate ingot sizes. Finally, increasing the width differential appears to provide considerable benefits for Problems 1 and 4, but not so much improvement for Problem 2 (for Problem 3, width differential is not relevant since all products in this class have the same width). For instance, for Problem 4 with  $p = 3$ , the objective value decreases by 10% when we increase the maximum width differential from 2 inches to 6 inches.

The tables also show the effect of increasing the maximum width differential parameter  $\omega$  and extending the set of candidate ingot sizes on the size of the problem (measured by the number of feasible assignments). Both of these changes increase the number of feasible combinations; however, increasing the number of candidate sizes has much greater impact on problem size. Also notice that, generally, as we extend the set of candidate ingot sizes the % of orders with dedicated ingots in the best solution tends to decrease (since more combinations are feasible); likewise, this % decreases as the width differential increases.

#### **4.5 Benefits of ingot sizing and order combination**

To assess the benefits of order combination, we solved the ingot sizing problem for the following two special cases : (i) no combinations permitted, i.e., all orders use dedicated ingots, and, (ii) limited combinations, i.e., only dedicated ingots or combining two orders for the same product are permitted. 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.

First, we compare the effectiveness of the ISP solution with the current set of standard ingot sizes. The facility's current 6 sizes, although not tailored for order combination, do permit joint processing of orders. Using the matching procedure, we determined the

optimal assignment of feasible pairs to the current set of standard ingot sizes to meet total demand. Table 7 compares the ISP values with the total weight of ingots processed for the current ingot sizes (we compute the percentage deviation relative to the total weight for the current sizes). For Problems 1 and 2, the ISP solution is considerably superior (over 18% savings), whereas for Problems 3 and 4 the improvement is only marginal (indeed, for Problem 4 with the restricted set of candidate ingots, the ISP solution is worse than current practice; we attribute part of this deterioration to the poor quality of our heuristic solution). Since all products in Problem 3 have the same width, solving the ISP does not produce dramatic savings. When we extend the set of candidate ingot sizes, the savings are uniformly higher, highlighting the importance of using a good set of candidate ingot sizes. The total savings for the product family that we studied was of the order of hundreds of thousands of dollars annually, and represented almost 10% reduction in the total weight of metal processed.

To quantify the benefit of combining orders, we applied the ISP solution procedure to two variants of Problem 3—one without any combinations (i.e., only dedicated ingots), and one with limited combinations (within each product group only). Table 8 compares the standard solution to these two special cases. The number of assignment variables increases dramatically when we allow order combination, from 897 with no combinations to 1441 for limited combinations to 9838 for the standard 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 surplus and scrap. The results indicate that order combination reduces total ingot usage by an average of 8.3% when compared to the limited-combinations model, and by an average of 26% when compared to the no-combinations model.

## **5. Concluding Remarks**

This paper has described the tactical planning problem of deciding standard ingot sizes in a make-to-order aluminum sheet manufacturing facility. We modeled this problem as an integer program, and developed a dual ascent lower bounding procedure and heuristics. Our computational results indicate that the method is very effective, producing solutions that are guaranteed to be within 4% of optimality. We also illustrated how to use the model and solution method to assess the sensitivity of overall performance to problem parameters such as the number of standard ingot sizes, the maximum width differential,

and the set of candidate ingot sizes. These results demonstrate that order combination is an effective means to reduce total production costs.

Future work might address related algorithmic and practical issues, including improving the two heuristic methods and exploring more sophisticated ascent methods to increase several  $v$ -values simultaneously. In this paper we limited the number of orders that can be combined to two, but combining more orders might become operationally feasible in the future. The underlying dual ascent principles apply to this more complex problem as well; however, the order assignment problem (for a given choice of standard ingot sizes) is not as easy. Another issue to explore is the linkage between the medium-term ingot sizing decisions and the short-term order combination decisions, perhaps, using detailed simulations.

## References

- A. BALAKRISHNAN, T. L. MAGNANTI, and R. T. WONG, 1989. A Dual Ascent Procedure for Large scale Uncapacitated Network Design, *Operations Research* 37, 716-740.
- A. BALAKRISHNAN, and S. B. BROWN, 1992. Process Planning for Metal-forming Operations: An Integrated Engineering-Operations Perspective", Working Paper # 3432-92-MSA, Sloan School of Management, MIT, Cambridge, MA.
- K. R. BAKER, M. J. MAGAZINE, and H. L. W. NUTTLE, 1986. The Effect of Commonality on Safety Stock in a Simple Inventory Model, *Management Science* 32, 982-988.
- J. E. BEASLEY, 1985. An Algorithm for the Two-dimensional Assortment Problem, *European Journal of Operational Research* 19, 253-261.
- M. L. CHAMBERS and R. G. DYSON, 1976. The Cutting Stock Problem in Flat Glass Industry - Selection of Stock Sizes, *Operational Research Quarterly* 27, 949-957.
- U. DERIGS, 1988. Solving Non-bipartite Matching Problems via Shortest Path Techniques, *Annals of Operations Research* 13, 225-264.
- D. ERLINKOTTER, 1978. A Dual Based Procedure for Uncapacitated Facility Location, *Operations Research* 26, 992-1009.
- M. L. FISHER and P. KEDIA, 1986. Optimal Solution of Set Covering/Partitioning Problems using Dual Heuristics, *Management Science* 25, 955-966.
- P. C. GILMORE and R. E. GOMORY, 1961. A Linear Programming Approach to the Cutting Stock Problem, *Operations Research* 9, 849-859.
- P. C. GILMORE and R. E. GOMORY, 1963. A Linear Programming Approach to the Cutting Stock Problem, Part II, *Operations Research* 11, 863-888.
- S. GOPALAKRISHNAN, 1994. Optimization Models for Production Planning in Metal Sheet Manufacturing, Unpublished thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.
- R. GOPALAN, 1992. Exploiting Process Flexibility in Metal Forming Operations, Unpublished thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.
- R. L. FRANCIS and P. B. MIRCHANDANI, 1990. *Discrete Location Theory*, John Wiley & Sons, Inc., New York.
- D. W. PENTICO, 1974. The Assortment Problem with Probabilistic Demands, *Management Science* 21, 286-290.
- D. W. PENTICO, 1988. The Discrete Two-dimensional Assortment Problem, *Operations Research* 36, 324-332.
- F. J. VASKO, F. E. WOLF and K. L. SCOTT, 1987. Optimal Selection of Ingot Sizes via Set Covering, *Operations Research* 35, 346-353.

- D. P. VENTOLA, 1991. Order Combination Methodology for Short-term Lot Planning at an Aluminum Rolling Facility, Unpublished thesis, Massachusetts Institute of Technology, Cambridge, MA.
- M. L. WOLFSON, 1965. Selecting the Best Lengths to Stock, *Operations Research* 13, 570-585.
- R. T. WONG, 1984. Dual Ascent Approach for Steiner Tree Problems on a Directed Graph, *Mathematical Programming* 28, 271-287.

Figure 1: Example Network for Order Assignment Problem

**Products:** 1 2 3  
**Demand:** 3 2 1

**Feasible combinations (i,j,k)** (excl. dominated combinations):  
 $\{(1,1,1), (1,2,2), (1,3,2), (2,3,3), (0,1,1), (0,2,2), (0,3,3)\}$

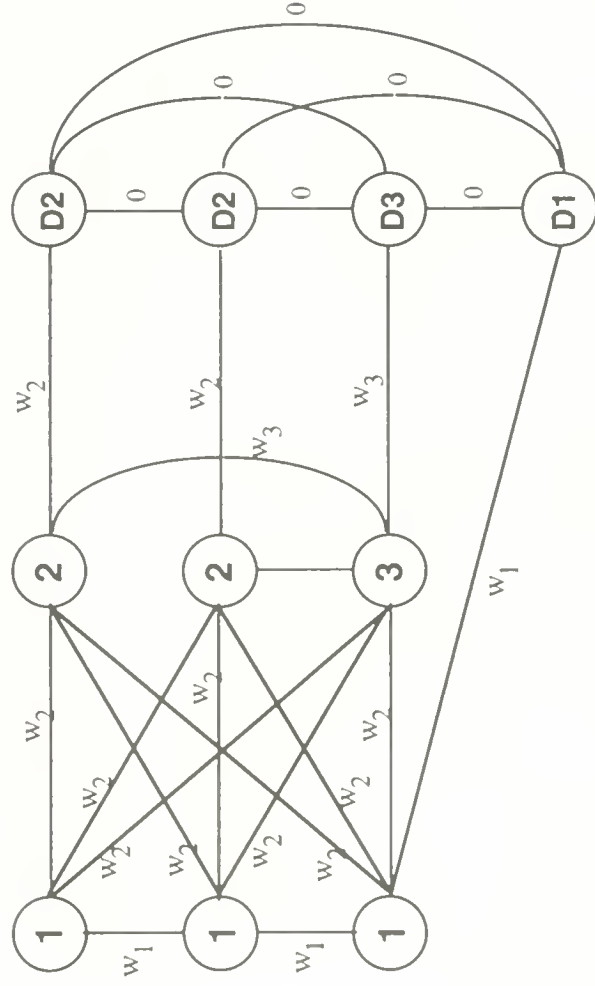


Table 1. Problem Sizes

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

Table 2. Heuristic Performance: Standard width differential, limited set of candidate ingot sizes<sup>†</sup>

Number of Std sizes p	Problem Number	Ingot Utilization Heuristic			Greedy Assignment Heuristic
		$\beta = 65\%$	$\beta = 70\%$	$\beta = 80\%$	
3	1	910	890	<b>885</b>	890
	2	1245	1260	1260	<b>1240</b>
	3	3625	3625	<b>3610</b>	Infeasible
	4	<b>2930</b>	3115	2930	3095
4	1	890	880	<b>875</b>	880
	2	1245	1245	1245	<b>1240</b>
	3	<b>3525</b>	3610	3605	Infeasible
	4	<b>2910</b>	2910	2910	2910
5	1	880	877.5	<b>875</b>	877.5
	2	<b>1235</b>	1245	1245	1240
	3	3522.5	<b>3515</b>	3515	Infeasible
	4	<b>2865</b>	2865	2910	2865
6	1	877.5	<b>872.5</b>	872.5	877.5
	2	1235	1245	<b>1232.5</b>	1240
	3	3522.5	<b>3515</b>	3515	3515
	4	2865	2865	2865	<b>2855</b>

Table 3. Bounds<sup>†</sup> and Sensitivity Analyses: Problem 1

# of Std Sizes	Width differential	Limited set of candidate ingots				Extended set of candidate ingots			
		# of feasible assignments	% orders dedicated	% Gap*	Upper Bound	# of feasible assignments	% orders dedicated	% Gap*	Upper Bound
3	2	562	32	1.9	885	2656	16	5.4	780
	4	613	20	1.2	870	2955	0	3.0	762.5
	6	766	20	4.9	850	3819	0	2.7	755
4	2	562	16	0.7	875	2656	4	5.1	777.5
	4	613	16	0.0	860	2955	8	2.4	757.5
	6	766	8	3.7	840	3819	0	2.7	755
5	2	562	32	0.7	875	2656	8	2.7	760
	4	613	16	0.0	860	2955	4	1.0	747.5
	6	766	8	3.7	840	3819	4	1.7	747.5
6	2	562	32	0.5	872.5	2656	8	2.7	760
	4	613	16	0.0	860	2955	4	1.0	747.5
	6	766	8	3.7	840	3819	4	1.7	747.5

\* % Gap =  $\{( \text{Best upper bound} - \text{Lower Bound} ) + \text{Lower Bound} \} * 100$

† All upper bounds in '000s of pounds



**Table 4. Bounds<sup>†</sup> and Sensitivity Analyses: Problem 2**

# of Std Sizes	Width differential	Limited set of candidate ingots				Extended set of candidate ingots			
		# of feasible assignments	% orders dedicated	% Gap*	Upper Bound	# of feasible assignments	% orders dedicated	% Gap*	Upper Bound
3	2	391	17.9	3.3	1240	1387	17.9	4.3	1200
	4	497	17.9	6.0	1240	1781	17.9	4.6	1200
	6	530	17.9	3.3	1240	1892	17.9	4.6	1200
4	2	391	17.9	3.3	1240	1387	17.9	3.5	1190
	4	497	21.4	5.1	1230	1781	21.4	2.8	1180
	6	530	17.9	2.5	1230	1892	21.4	2.8	1180
5	2	391	17.9	2.9	1235	1387	17.9	2.6	1180
	4	497	21.4	5.1	1230	1781	17.9	2.0	1170
	6	530	17.9	2.5	1230	1892	17.9	2.0	1170
6	2	391	21.4	2.7	1232.5	1387	21.4	2.6	1180
	4	497	21.4	5.1	1230	1781	17.9	2.0	1170
	6	530	17.9	2.5	1230	1892	17.9	2.0	1170

\* % Gap =  $\{(\text{Best upper bound} - \text{Lower Bound}) + \text{Lower Bound}\} * 100$

† All upper bounds in '000s of pounds

Table 5. Bounds<sup>†</sup> and Sensitivity Analyses: Problem 3

# of Std Sizes	Limited set of candidate ingots				Extended set of candidate ingots			
	# of feasible assignments	% orders dedicated	% Gap*	Upper Bound	# of feasible assignments	% orders dedicated	% Gap*	Upper Bound
3	4119	8.5	7.2	3610	9838	6.9	5.6	3434
4	4119	8.5	5.5	3525	9838	6.9	4.8	3400
5	4119	7.7	5.7	3515	9838	7.7	4.5	3390
6	4119	7.7	6.0	3515	9838	6.7	4.8	3386

\* % Gap =  $\{(\text{Best upper bound} - \text{Lower Bound}) \div \text{Lower Bound}\} * 100$

† All upper bounds in '000s of pounds

Table 6. Bounds<sup>†</sup> and Sensitivity Analyses: Problem 4

# of Std Sizes	Width differential	Limited set of candidate ingots				Extended set of candidate ingots			
		# of feasible assignments	% orders dedicated	% Gap*	Upper Bound	# of feasible assignments	% orders dedicated	% Gap*	Upper Bound
3	2	1952	22.1	6.5	2930	7107	33.7	16.2	3115 <sup>§</sup>
	4	2335	16.3	4.7	2860	8457	16.3	6.6	2850
	6	3068	14.0	5.6	2840	11068	14.0	6.3	2830
4	2	1952	22.1	6.3	2910	7107	19.8	6.5	2855
	4	2335	16.3	4.3	2850	8457	15.1	5.3	2815
	6	3068	14.0	5.6	2825	11068	14.0	5.0	2795
5	2	1952	31.4	4.6	2865	7107	19.8	6.0	2840
	4	2335	24.4	3.1	2815	8457	15.1	4.6	2795
	6	3068	23.3	4.3	2790	11068	14.0	4.4	2780
6	2	1952	34.9	4.3	2855	7107	30.2	3.7	2780
	4	2335	25.6	2.9	2810	8457	24.4	3.1	2755
	6	3068	26.7	4.1	2785	11068	24.4	3.1	2745

\* % Gap = {(Best upper bound - Lower Bound) + Lower Bound} \* 100

† All upper bounds in '000s of pounds

§ Poor heuristic solution; limited candidate set case provides better upper bound

**Table 7. ISP Solution versus Current Standard Sizes (with combination)  
(Number of standard sizes = 6)**

Problem Number	Width Differential	Limited set of candidate ingots		Extended set of candidate ingots	
		% Reduction in obj value for ISP solution	Average % reduction	% Reduction in obj value for ISP solution	Average % reduction
1	2	18.6	18.3	29.1 29.0 27.1	28.4
	4	18.1			
	6	18.1			
2	2	21.6	21.5	24.9 25.5 25.0	25.1
	4	21.7			
	6	21.1			
3	-	0.9	0.9	5.2	5.2
4	2	-2.0	-1.6	0.70 0.70 0.02	0.5
	4	-1.3			
	6	-1.4			



3 9080 00846197 9

**Table 8. Benefits of Order combination: Problem 3**

Number of Standard Sizes	Combination strategy	Number of Feasible Assignments	% orders dedicated	% Increase in obj value over ISP solution
3	ISP	9838	6.9	-
	Limited combination*	1441	24.6	13.7
	No combination**	897	100.0	30.3
4	ISP	4838	6.9	-
	Limited combination*	1441	25.4	8.2
	No combination**	897	100.0	26.9
5	ISP	9838	7.7	-
	Limited combination*	1441	25.4	5.9
	No combination**	897	100.0	25.1
6	ISP	9838	6.2	-
	Limited Combination*	1441	25.4	5.5
	No combination**	897	100.0	21.2

"Limited combination" ≡ Orders can either have dedicated ingots, or combined with another order in same product group.

\*\* "No combinations" ≡ All orders must use only dedicated ingots





Date Due

SEP. 6 1995

Lib-26-67



