SOLVING THE PRODUCT MIX PROBLEM USING ACTIVITY-BASED COST INFORMATION - AN AGGREGATION/DISAGGREGATION APPROACH

Munoz, M.A.\textsuperscript{1}, Ruiz-Usano, R.\textsuperscript{1}, Framinan, J.M.\textsuperscript{1} and Leisten, R.\textsuperscript{2}.

\textsuperscript{1}University of Seville. Camino de los Descubrimientos s/n. 41092. Seville. e-mail: miguelangel@pluto.us.es Phone: +34-(9)5-4487220 Fax: +34-(9)5-4487210

\textsuperscript{2}University of Duisburg. Lotharstr. 65, D-47057 Duisburg. e-mail: leisten@uni-duisburg.de Phone: +49-(0)203-379-2624 Fax: +49-(0)203-379-2922

ABSTRACT

Activity-Based Cost information has been used to assist in a wide range of managerial decisions, including the product mix selection. Nevertheless, the resultant models are limited by the computational complexity to solve integer programming problems of large size generated in actual manufacturing environments. One solution methodology for large size problems is the aggregation and disaggregation technique. In this paper, we develop an aggregation/disaggregation solution approach for the product mix problem using activity-based cost information. The original programming problem is aggregated into a problem of moderate size. Feasible solutions for the detailed model are generated through a disaggregation procedure and the Theory of Constraints method is used as a heuristic in order to improve the solutions. An upper bound on the loss of accuracy is also calculated. Finally, a small test problem is solved in order to illustrate the idea.

Keywords: Aggregation/disaggregation, product mix, activity-based costing.

INTRODUCTION

One of the classical applications of mathematical programming is the product mix problem (Wagner, 1969, p.35). In this problem we are interested in determining which products should be manufactured in which quantities within a certain planning period, with the objective of determining the maximum profit, subject to constraints on the different resources. The product mix problem can be addressed in the aggregate planning module as a stage in the production planning and control hierarchy in situations where we know in advance that we will never carry inventory from one period to the next. In such situations we can assume a planning horizon of only one period (Hopp and Spearman, 1996, Chapter 16).

Increasing complexity of manufacturing environments and rising overhead expenses have led to the emergence of a new approach in cost accounting referred to as activity-based costing (ABC) (Innes et al., 1994). The ABC system has been used to assist in a range of managerial decisions, including the product mix selection. In this direction, some mixed integer linear programming models have been developed to determine optimal product mix
using ABC information (Malik, 1993; Malik and Sullivan, 1995; Kee, 1995; Yahya-Zadeh, 1998; Kee and Schmidt, 2000). These models permit the incorporation of different levels of detail with respect to costs that may be available in an ABC environment and there is no need to assume a unit cost of each product before solving the model.

Nevertheless, these models are limited by the computational requirements of integer programming on large scale problems generated in actual manufacturing environments. One solution methodology for large size problems is the aggregation and disaggregation technique (Rogers et al., 1991). Besides, it should be pointed out that the product mix problem is addressed within the aggregate planning module. As a result, certainty of the disaggregated data is questionable, and therefore, expensive computations are not worthwhile.

Aggregation and disaggregation (a/d) techniques have been shown to be valuable for mixed integer linear programming models, with the objective of reducing computational burden. Weintraub, Guitart and Kohn (1986) formulate a model for the forest industry with 600 constraints, 1500 variables and 30 binary variables. Through a/d a 75% reduction in computational effort is achieved with an error bound that is within 7.7% of the optimal objective value. Golovin (1975) considers a/d in a mixed integer programming model that has a concave objective function because of set-up costs. Aggregation procedures for the model and disaggregation procedures for reduced models are given. Hallefjord et al (1993) propose an a/d technique to solve large scale generalized assignment problems. Srinivasa and Wilhelm (1997) develop an optimization procedure that is based on an aggregation scheme and strong cutting plane methods for optimizing tactical response in oil spill clean up operations.

A/d techniques have been widely studied in the literature. Rogers et al. (1991) give an excellent survey over a/d for optimization problems. However, most of the a/d applications is in linear programming - see Leisten (1998) for a recent review of LP-aggregation -. Considering integrality constraints in linear programming result in much more complex models for which a/d considerations are significantly more complex. Since no theory of duality exists for integer variables, the optimal a/d weights cannot be approximated by means of classical iterative approaches. A/d considerations with respect to integer programming have been considered from a formal point of view by Hallefjord and Storøy (1990).

A/d techniques have been also applied to production planning - see Rogers et al. (1991), sections 2.6.3 and 3.2.2, for an overview on production planning and scheduling applications of a/d techniques -. Golovin (1975), Toczylowski (1986) and Pienkosz and Toczylowski (1993) are references of mixed integer production planning problems. Leisten (1998) gives an LP-aggregation view on aggregation in multi-level production planning.

A/d in product mix models is examined by Knolmayer (1983). Simulation studies of randomly generated problems are performed. Strategies for determining the best method of combination are suggested for this problem. Nevertheless, the model developed by Knolmayer is not a mixed integer linear programming model, but a simple linear programming one.

In this paper, we develop an a/d solution approach for product mix problem using activity-based cost information. In the next sections the product mix model using ABC information will be formulated, followed by the aggregated product mix problem. Then we develop a disaggregation procedure that gives a feasible solution of the original problem. One main part of the a/d techniques is to measure the degree of accuracy lost under the specific a/d approach. The bound on the loss of accuracy is also calculated. Finally, a small test problem is solved in order to illustrate the idea.
TERMINOLOGY

Consider the model proposed by Malik and Sullivan (1995):

(P1X) Maximize \[ \sum_{i=1}^{n} \left( s_i - m_i - l b_i \right) x_i - \sum_{j=1}^{r} c_{ji} \left\langle x_i / a_{ji} \right\rangle \]

Subject to \[ \sum_{i=1}^{n} m_i x_i \leq m \]
\[ \sum_{i=1}^{n} l b_i x_i \leq l \]
\[ \sum_{i=1}^{n} c_{ji} \left\langle x_i / a_{ji} \right\rangle \leq o_j, \quad j = 1, ..., r \]
\[ 0 \leq x_i \leq u_i, \quad i = 1, ..., n, \]

where:
\( x_i \) = amount of product \( i \) to be produced in a given time horizon.
\( n \) = total number of different products to be considered.
\( s_i \) = selling price per unit of product \( i \).
\( m_i \) = material cost per unit of product \( i \).
\( l b_i \) = labor cost per unit of product \( i \).
\( m \) = cash equivalent of maximum material resource available.
\( l \) = cash equivalent of maximum labor resource available.
\( u_i \) = upper bound on the amount of product \( i \).
\( r \) = total number of indirect resources.
\( a_{ji} \) = upper bound on the amount of product \( i \) that can be produced from the amount of resource \( j \) that costs \( c_{ji} \).
\( o_j \) = cash equivalent of maximum resource \( j \) available.
\( \left\langle x_i / a_{ji} \right\rangle \) denotes the smallest integer greater than or equal to \( x_i / a_{ji} \).

For the sake of simplicity we assume that there is no lower bound \( l_i \) on the amount \( x_i \).
(Otherwise, replace \( x'_i = x_i - l_i \) and \( u'_i = u_i - l_i \)).

Malik and Sullivan (1995) propose the transformation of variables \( y_{ji} = \left\langle x_i / a_{ji} \right\rangle \).
Therefore, an equivalent representation of P1X as a mixed integer programming problem is as follows:

(P1Y) Maximize \[ \sum_{i=1}^{n} \left( s_i - m_i - l b_i \right) x_i - \sum_{j=1}^{r} c_{ji} y_{ji} \]

Subject to \[ \sum_{i=1}^{n} m_i x_i \leq m \]
\[ \sum_{i=1}^{n} l b_i x_i \leq l \]
\[ \sum_{i=1}^{n} c_{ji} y_{ji} \leq o_j, \quad j = 1, ..., r \]
\[ y_{ji} \geq x_i / a_{ji}, \quad i = 1, ..., n; \quad j = 1, ..., r \]
\[ 0 \leq x_i \leq u_i, \quad i = 1, ..., n, \]
\[ y_{ji} \text{ integer,} \quad i = 1, ..., n; \quad j = 1, ..., r. \]
This model is very hard to solve when \( n \) and/or \( r \) are high, because of the integrality requirements on \( y_{ji} \).

In many practical applications, the number of products is much larger than the number of resources and in the next section we will aggregate the products into product lines or groups and hence obtain an aggregated problem with the same number of resources, but with a smaller number of products. This corresponds to a column aggregation in the original problem. Resource aggregation would result into a combination of row aggregation and column aggregation, which will not be considered here.

THE AGGREGATE PROBLEM

According to Rogers et al. (1991) the a/d approach in optimization consists of three steps: aggregation analysis, disaggregation analysis and error analysis. In this paper, we assume that product groups are predetermined in the sense that a cluster procedure has previously determined which products are members of a cluster (e.g. a product group). Therefore, the only issue of aggregation analysis is the method of combination.

Let the products 1,...,\( n \) be partitioned into \( K \) groups. The set of indices in the \( k \)-th group is denoted by \( S_k, k =1,...,K \). A standard method of combination is to assign weights to the objects to be clustered (variables). We introduce variable weights \( g_i, i=1,...,n \) and \( h_{ji}, i=1,...,n \) and \( j=1,...,r \), which are normalized per cluster:

\[
\sum_{i \in S_k} g_i = 1, \quad k = 1,..., K
\]

\[
\sum_{i \in S_k} h_{ji} = 1, \quad j = 1,..., r, \quad k = 1,..., K
\]

Therefore, the following model is an aggregated version of P1Y:

\[
\text{(P2Y) Maximize } \sum_{k=1}^{K} \left( \sum_{i \in S_k} g_i \left( c_i - m_i - l b_i \right) \right) Y_{jk} - \sum_{j=1}^{r} \left( \sum_{i \in S_k} h_{ji} c_{ji} \right) Y_{jk} \]

subject to

\[
\sum_{k=1}^{K} \left( \sum_{i \in S_k} g_i m_i \right) X_k \leq m
\]

\[
\sum_{k=1}^{K} \left( \sum_{i \in S_k} g_i l b_i \right) X_k \leq l
\]

\[
\sum_{k=1}^{K} \left( \sum_{i \in S_k} h_{ji} c_{ji} \right) Y_{jk} \leq o_j, \quad j = 1,..., r
\]

\[
h_{ji} Y_{jk} \geq g_i X_k / a_{ji}, \quad j = 1,..., r; k = 1,..., K; \forall i \in S_k
\]

\[
0 \leq X_k \leq \min_{i \in S_k; g_i \neq 0} \left( u_i / g_i \right), \quad k = 1,..., K
\]

\[
Y_{jk} \text{ integers}, \quad j = 1,..., r; k = 1,..., K.
\]
In model P2Y, lowercase letters indicate original (i.e. detailed and disaggregated) variables and parameters and uppercase letters indicate aggregate variables and parameters.

Problem P2Y has a smaller number of variables than P1. Nevertheless, the P2Y model has two drawbacks. First, P2Y is also a mixed integer program and it can be very difficult to solve. Second, nothing can be said about the relation between the objective function value of P1Y and the objective function value of P2Y.

According to Hallefjord and Storøy (1990), we propose the problem P3Y. This problem is the same as problem P2Y, except that integrality requirements on \( Y_{jk} \) are not imposed. Problem P3Y is easier to solve than P1Y, since it is a pure linear programming model. Besides there is a relation between the objective function value of P1Y and the objective function value of P3Y - see Hallefjord and Storøy (1990) -. 

**THE DISAGGREGATION PROCEDURE**

After having solved problem P3Y, a detailed disaggregated solution for problem P1Y has to be determined. The most simple way to disaggregate is to use so-called fixed-weight disaggregation (Rogers et al, 1991). Nevertheless, in the integer programming case, fixed-weight disaggregation does not, in general, result in a feasible solution to the original problem P1Y, whichever of the aggregated problems P2Y and P3Y is solved (Hallefjord and Storøy, 1990). However, we can develop a specific procedure which can be used for computing a feasible solution to P1Y, given that P3Y has been solved.

Let \( X^*, Y^* \) be the optimal solution to P3Y. Then, for every variable cluster \( S_k \) \( (k = 1, \ldots, K) \) we define the subproblem

\[
(P4X_k) \quad \text{Maximize} \quad \sum_{i \in S_k} \left( s_i - m_i - lb_i \right) x_i - \sum_{j=1}^{r} c_{ji} \langle x_i, a_{ji} \rangle \tag{1}
\]

Subject to

\[
\sum_{i \in S_k} m_i x_i \leq \left( \sum_{i \in S_k} g_i m_i \right) X_k^*,
\]

\[
\sum_{i \in S_k} lb_i x_i \leq \left( \sum_{i \in S_k} g_i lb_i \right) X_k^*,
\]

\[
\sum_{i \in S_k} c_{ji} \langle x_i, a_{ji} \rangle \leq \left( \sum_{i \in S_k} h_{ji} c_{ji} \right) Y_{jk}^*, \quad j = 1, \ldots, r
\]

\[
0 \leq x_i \leq u_i, \quad \forall i \in S_k.
\]

Note that \( P4X_k \) has always a feasible solution, since the solution \( x_i = 0, \forall i \in S_k \), is feasible. Also, we may state the following proposition:

**Proposition** If \( X^*, Y^* \) is the optimal solution to P3Y and \( x' \) is the solution for the original problem obtained by the subprograms \( P4X_k \), then \( x' \) is feasible for P1X.

**Proof.** We only need to demonstrate that \( x' \) satisfies the constraints of P1X:
Therefore, we have developed a disaggregation procedure that gives a feasible solution to the original problem.

In order to solve $P4X_k$, we may formulate the following mixed integer programs for $k = 1, \ldots, K$:

\[
\begin{align*}
\text{Maximize} & \quad \sum_{i \in S_k} \left\{ s_i - m_i - lb_i \right\} x_i - \sum_{j=1}^{r} c_{ji} w_{ji} \\
\text{Subject to} & \quad \sum_{i \in S_k} m_i x_i \leq \left( \sum_{i \in S_k} g_i m_i \right) x_k^* \\
& \quad \sum_{i \in S_k} lb_i x_i \leq \left( \sum_{i \in S_k} g_i lb_i \right) x_k^* \\
& \quad \sum_{i \in S_k} c_{ji} w_{ji} \leq \left( \sum_{i \in S_k} h_{ji} c_{ji} \right) y_{jk}^* \\
& \quad w_{ji} \geq x_i / a_{ji}, \quad j = 1, \ldots, r; \forall i \in S_k \\
& \quad 0 \leq x_i \leq g_i x_k^*, \quad \forall i \in S_k \\
& \quad w_{ji} \text{ integer,} \quad j = 1, \ldots, r; \forall i \in S_k.
\end{align*}
\]

Although $P4W$ is a collection of mixed integer programs, each problem may contain a small number of integer variables.

**BOUND ON THE LOSS IN ACCURACY**

One main part of every a/d approach is to measure the degree of accuracy loss under the specific aggregation. In the integer programming case, the optimal objective function value of the relaxed-original (detailed) problem, $z^*_{P1YLP}$, is an upper bound for the optimal objective function value of the original integer programming model:

\[
z^*_{P1Y} = z^*_{P1Y} \leq z^*_{P1YLP}
\]

However, if $P1Y$ (and also $P1YLP$) is a large scale problem, $z^*_{P1YLP}$ cannot be available. Therefore, the estimation of the a/d error is required by means of the standard aggregation theory.
Zipkin (1980) gave a bound on the loss in the objective function value that results from column aggregation. Hallefjord and Storøy (1990) remark that this bound is also valid in the integer programming case.

On the other hand, we note that upper bound constraints of the form \( x_i \leq u_i \) exist. Note also that

\[
y_{ji} = \langle x_j / a_{ji} \rangle \leq \langle u_j / a_{ji} \rangle, i = 1, \ldots, n; j = 1, \ldots, r.
\]

Therefore, we may consider each cluster as consisting of one element each – see, for example, Rogers et al. (1991), p.564-.

According to Leisten (1997), we may formulate the following upper bound for the objective function value of \( P_1X \)

\[
z_{P1X}^* = z_{P1Y}^* \leq vb + \sum_{i=1}^{n+n^r} c_i - vA^T u_p
\]

where:

\( v \) is an “appropriate” non-negative dual solution of \( P_3Y \). This vector may be represented by \( v = (v_1, v_2, v_1, \ldots, v_r, v_1^x, \ldots, v_n^x) \). Note that each dual variable corresponds to a constraint of \( P1Y \).

\( b \) is the vector of constraints: \( b = (m, l, o_1, \ldots, o_r, 0, \ldots, 0, u_1, \ldots, u_n) \).

\( c_l \) and \( u_p \) correspond to the coefficient of column \( l \) in the objective function of \( P1Y \), the \( l \)-th column of the technological matrix of \( P1Y \) and the upper bound on the variable of column \( l \), respectively.

\(|y|^+\) indicates max \((y, 0)\).

Taking into account the structure of the rows and columns of \( A \) (see table 1 for \( n = r = 2 \)), this bound may be stated for our problem as follows:

\[
z_{P1X}^* = z_{P1Y}^* \leq \left(m v_m + l v_l + \sum_{j=1}^r o_j v_j + \sum_{i=1}^n u_i v_i^x\right) + \]

\[
+ \sum_{j=1}^r \left(s_j - m_j - lb_j\right) - m_j v_m - lb_j v_l - \sum_{j=1}^r v_j - v_i^x \right)^T u_i + \]

\[
+ \sum_{j=1}^r \sum_{i=1}^n - c_{ji} - c_{ji} v_j + a_{ji} v_{ji} \left\langle u_j / a_{ji} \right\rangle
\]

<table>
<thead>
<tr>
<th>Columns</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( y_{11} )</th>
<th>( y_{12} )</th>
<th>( y_{21} )</th>
<th>( y_{22} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 - m_1 - lb_1 )</td>
<td>( s_2 - m_2 - lb_2 )</td>
<td>- ( c_{11} )</td>
<td>- ( c_{12} )</td>
<td>- ( c_{21} )</td>
<td>- ( c_{22} )</td>
<td></td>
</tr>
<tr>
<td>( m_1 )</td>
<td>( m_2 )</td>
<td>( m )</td>
<td>( l )</td>
<td>( o_1 )</td>
<td>( o_2 )</td>
<td></td>
</tr>
<tr>
<td>( lb_1 )</td>
<td>( lb_2 )</td>
<td>( c_{11} )</td>
<td>( c_{12} )</td>
<td>( c_{21} )</td>
<td>( c_{22} )</td>
<td></td>
</tr>
<tr>
<td>( 1 )</td>
<td>- ( a_{11} )</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 1 )</td>
<td>- ( a_{21} )</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 1 )</td>
<td>- ( a_{12} )</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 1 )</td>
<td>- ( a_{22} )</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 1 )</td>
<td>( u_1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 1 )</td>
<td>( u_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Structure of the rows and columns of \( A \)
If we choose the optimal dual solution of the aggregate problem, then we get the Zipkin bound mentioned above.

AN ILLUSTRATIVE EXAMPLE

In this section we will illustrate the procedure by solving a small test example adopted from Kee (1995):

\[
\begin{align*}
\text{Maximize} & \quad 54x_1 - 36 \langle x_1/l \rangle - 800 \langle x_1/1000 \rangle - 1000 \langle x_1/4000 \rangle - 100000 \langle x_1/100000 \rangle + \\
& \quad + 61x_2 - 36 \langle x_2/l \rangle - 800 \langle x_2/1000 \rangle - 1600 \langle x_2/4000 \rangle - 100000 \langle x_2/100000 \rangle + \\
& \quad + 176x_3 - 96 \langle x_3/l \rangle - 1600 \langle x_3/500 \rangle - 2400 \langle x_3/1000 \rangle - 300000 \langle x_3/30000 \rangle + \\
& \quad + 416x_4 - 216 \langle x_4/l \rangle - 2000 \langle x_4/200 \rangle - 3000 \langle x_4/500 \rangle - 500000 \langle x_4/20000 \rangle
\end{align*}
\]

Subject to

(i) \(4x_1 + 7x_2 + 15x_3 + 28x_4 \leq 1500000\)
(ii) \(12x_1 + 12x_2 + 32x_3 + 72x_4 \leq 3040000\)
(iii) \(36 \langle x_1/l \rangle + 36 \langle x_2/l \rangle + 96 \langle x_3/l \rangle + 216 \langle x_4/l \rangle \leq 9000000\)
(iv) \(800 \langle x_1/1000 \rangle + 800 \langle x_2/1000 \rangle + 1600 \langle x_3/500 \rangle + 2000 \langle x_4/200 \rangle \leq 200000\)
(v) \(100000 \langle x_1/100000 \rangle + 1600 \langle x_2/4000 \rangle + 2400 \langle x_3/1000 \rangle + 3000 \langle x_4/500 \rangle \leq 160000\)
(vi) \(100000 \langle x_1/100000 \rangle + 100000 \langle x_2/100000 \rangle +
+ 300000 \langle x_3/30000 \rangle + 500000 \langle x_4/20000 \rangle \leq 1000000\)
(vii) \(0 \leq x_1 \leq 100000\)
(viii) \(0 \leq x_2 \leq 100000\)
(ix) \(0 \leq x_3 \leq 30000\)
(x) \(0 \leq x_4 \leq 20000\)

where

(i) is the material constraint.
(ii) is the labor constraint.
(iii) is the unit-level overhead constraint.
(iv) is the set-up constraint.
(v) is the purchasing constraint.
(vi) is the engineering constraint.
(vii) - (x) are the upper bound constraints.

The optimal solution of P1X is

\[
\begin{align*}
x_1^* &= 30000; \quad x_2^* = 100000; \quad x_3^* = 30000; \quad x_4^* = 0 \\
z^*_{\text{P1X}} &= z^*_{\text{P1Y}} = 4620000.
\end{align*}
\]

The optimal objective function value to the continuous relaxation is \(z^*_{\text{P1YLP}} = 4690500\).

Suppose the problem is aggregated as follows. Let \(K = 2, S_1 = \{1,2\}\) and \(S_2 = \{3,4\}\). We choose the weighting \(g_1 = g_2 = g_3 = g_4 = 0.5\). So, the relaxed aggregated problem is
The optimal primal solution of P3Y is

\[ X_1^* = 200000 \quad X_2^* = 4444.4 \]
\[ Y_{11}^* = 200000 \quad Y_{12}^* = 4444.4 \]
\[ Y_{21}^* = 200 \quad Y_{22}^* = 22.2 \]
\[ Y_{31}^* = 50 \quad Y_{32}^* = 8.9 \]
\[ Y_{41}^* = 2 \quad Y_{42}^* = 0.2 \]
\[ z^*_{P3Y} = 4344333.3 \]

and the optimal dual solution is

\[ \nu_{m}^* = 0 \quad \nu_{j}^* = 0 \]
\[ \nu_{1}^* = 0 \quad \nu_{2}^* = 11.73 \quad \nu_{3}^* = 0 \quad \nu_{4}^* = 0 \]
\[ \nu_{11}^* = 0 \quad \nu_{21}^* = 0 \quad \nu_{31}^* = 0 \quad \nu_{41}^* = 0 \]
\[ \nu_{12}^* = 72 \quad \nu_{22}^* = 20.37 \quad \nu_{32}^* = 0.65 \quad \nu_{42}^* = 2 \]
\[ \nu_{13}^* = 0 \quad \nu_{23}^* = 0 \quad \nu_{33}^* = 0 \quad \nu_{43}^* = 0 \]
\[ \nu_{14}^* = 312 \quad \nu_{24}^* = 229.2 \quad \nu_{34}^* = 10.8 \quad \nu_{44}^* = 40 \]
\[ \nu_{1}^{x*} = 0 \quad \nu_{2}^{x*} = 19.98 \quad \nu_{3}^{x*} = 0 \quad \nu_{4}^{x*} = 0 \]
Then, the Zipkin bound can be calculated as we mentioned above. This gives us:

\[ z_{p_{1x}}^* = z_{p_{1y}}^* \leq 24121300 \]

We note that the optimal objective function value of the relaxed original (detailed) problem, \( z_{p_{1y}lp}^* = 4690500 \), is clearly better than the Zipkin bound. This situation is very common in a/d approaches, and, consequently, several extensions and modifications of the Zipkin bound have been proposed in order to improve this bound (Leisten, 1997). This topic will be addressed in future work.

If the disaggregation procedure is applied, two disaggregation models (P4W1 and P4W2) will be solved. The solution to these subprograms leads to

\[ x_1' = 100000; \ x_2' = 100000; \ x_3' = 0; \ x_4' = 0 \]
\[ z' = 3875000 \text{ (percentage error} = 16.1\% \text{).} \]

It may be possible to improve this feasible solution by means of an heuristic. A simple one is just to use the Theory Of Constraints (TOC) method described in Fox (1987).

First of all, we identify the system constraint when \( x_i = u_i, \ \forall i \). Computing the slack variables, we get

Material slack = -610000 (141%).
Labor slack = -1760000 (158%).
Unit-level overhead constraint = -5400000 (160%).
Set-up slack = -2560000 (228%).
Purchasing slack = -97000 (161%).
Engineering slack = 0 (100%).

So, set-up is the critically constrained resource. We then compute the profit per $ of set-up for every product \( i \) \( (x_i \neq u_i) \):

\[
{\text{ratio}_i = \frac{s_i - m_i - l_b_i - \sum_{j=1}^{L} c_{ij} \ a_{ij}}{c_{2i} \ a_{2i}}} \Rightarrow \text{ratio}_3 = 20.1\$/$; \text{ratio}_4 = 15.9\$/$
\]

Since product 3 offers the highest ratio, we will produce as much as possible of 3 and then we use the remaining available capacity to produce as much as possible of product 4. So, we get:

\[ x_1'' = 100000; \ x_2'' = 100000; \ x_3'' = 12500; \ x_4'' = 0 \]
\[ z'' = 4503800 \text{ (percentage error} = 2.5\% \text{).} \]

Note also that the method proposed by Fox (1987) may be applied to select a "good" product mix without a/d considerations. However, the TOC method has been shown to be heuristic in nature (Balakrishnan, 1999) and this procedure may lead to very inefficient solutions when there are multiple constrained resources (Plenert, 1993).
CONCLUSIONS AND POSSIBLE EXTENSIONS

In this paper, an aggregation/disaggregation approach has been applied to the product mix problem using Activity Based Cost information. The optimal aggregated solution is disaggregated in order to obtain a feasible solution to the original problem. An upper bound is also calculated in order to measure the deterioration of the objective function value under the specific aggregation. This bound could be improved according to Leisten (1997), but this topic will be addressed in future work.

The aggregation/disaggregation approach considered here is a noniterative procedure, in the sense that the given problem is aggregated, solved, and the solution is disaggregated. Another further development would be to design an iterative technique for successive improvement of the disaggregated solution programming – see, for example, Jornsten et al. (1999) -. This is a difficult task, since the optimal weights in mixed integer programming problems can neither be approximated by straight-forward subgradient procedures or reclustering strategies (Leisten,1998).

The aggregation/disaggregation approach developed here can be extended to aggregate resources. However, simultaneous aggregation of variables and constraints in mixed integer programming problems has seldom been discussed in the literature, since simultaneous column and row aggregation causes problems in feasible disaggregation and in generating bounds.

Another topic of future work would be to obtain some computational results from larger problems, which show the performance of the proposed approach.

Finally, it would also be very interesting to discuss the approach within a managerial context, i.e., how can an a/d approach be integrated into an activity-based controlling system and how can it contribute to the solution of cost accounting problems, apart from solely technical aspects.

REFERENCES


