J Math Model Algor
DOI 10.1007/s10852-006-9039-0

An Optimization Model to Determine Master Designs and Runs for Advertisement Printing

S. R. Mohan · S. K. Neogy · A. Seth ·
N. K. Garg · S. Mittal

Abstract In this paper we consider a common optimization problem faced by a printing company while designing masters for advertisement material. A printing company may receive from various customers, advertisements for their products and services and their demand is for a specified number of copies to be printed. In a particular case, the printer receives these orders to be delivered next week from the customers, until the Thursday of a week. By Monday the printed copies have to be delivered to the customers. These advertisement items of the various customers are to be printed on large sheets of papers of specified standard sizes. The size is called a *k*-up if *k* items can be printed on one sheet. It is a given constraint that only items of the same size can be loaded on a master. This constraint results in a decomposition of the original problem of designing masters into many sub-problems, one for each size. The objective is to minimize the number of masters required while meeting the requirements of the customers. We formulate this optimization problem mathematically, discuss the computational issues and present some heuristic approaches for solving the problem.

Key words advertisement printing · heuristic approach · G-MIM heuristic · G-RBA heuristic · adjacent vertex heuristic.

1. Introduction to Advertisement Printing Problem

A printing company has orders for printing advertisement items of various types. Depending on the sizes (which are standardized in the industry) the item types are

---

S. R. Mohan · S. K. Neogy (✉) · A. Seth
Indian Statistical Institute, 7, S.J.S. Sansanwal Marg, New Delhi, India
e-mail: skn@isid.ac.in

N. K. Garg · S. Mittal
R. Systems International Ltd., C-40, Sector 59, Noida- 201307, India

described as 4-up, 6-up, 12-up etc. A 4-up item is of a standard size such that four of these items can be loaded on a single sheet. Similarly, a 6-up type is such that six items can be accommodated on a single sheet. It is also a practice in the industry that items of different types are not designed to be loaded on a single sheet. So, a master can have only items of the same type designed on a single sheet. But the items of the same type can be different advertisements either from the same customer or from different customers. The problem is to design the masters and run them a number of times in an optimal manner so that the demand made by various customers for specified items to be produced for the next week is met. The design and the production runs should satisfy the following constraints:

1. A master has to be run at least 5,000 times and not more than 50,000 times.
2. A master can carry only items of the same type.

The criteria for optimization are basically two. The design and number of production runs should be such that only the minimal number of master sheets is used to meet all of the demand. This is to ensure that the setup cost is minimum. The other objective is the minimization of total wastage. Suppose we need to design and run the masters for N items of advertisement with demand for the j^{th} item as d_j, $j = 1, 2, \ldots, N$. Suppose we have a solution that actually produces $m_1, m_2, \ldots, m_N$ of the items $1, 2, \ldots, N$. For the solution to be feasible, we have $m_j - d_j \geq 0$. The excess production for the j^{th} item is $s_j = m_j - d_j$. We want that $\sum_{j=1}^{N} s_j$ is also minimized. Thus the problem is a multiobjective minimization problem, with the two objectives of minimizing the number of masters used and also minimizing the total wastage.

1.1. A Simplification

The fact that only one type can be loaded on a master sheet, decomposes the original problem encompassing all the types of demand into a number of problems, one for each type. Thus by solving the problem associated with the demand for items of each type separately and then by putting together these solutions we can obtain a solution for the whole problem. In fact further decomposition results from the consideration of quality of paper. So a master can be prepared only with items of the same type and quality of paper. The larger problem therefore decomposes into solving one problem for each type and quality. The two objectives of the problem we stated, namely minimizing the number of masters used and also minimizing the wastage, however can be conflicting. To see this we consider the following two simple examples.

Example 1.1. Consider $k = 4$-up size and a specified quality of paper to be used. Let there be four items with demands $d_1 = 30500$, $d_2 = 31200$, $d_3 = 31600$, and $d_4 = 31800$. If we consider a solution in four masters, allotting item 1 to all the four slots in the first master, allotting item 2 to all the four slots in the second master and so on, we need to run the first master 7,625 times, the second master 7,800 times, the third master 7,900 times and the fourth master 7,950 times. Then the demand for each of the item will be met exactly, the wastage will be 0, but the number of masters will be 4. This solution is feasible and optimal for the wastage. However, if we consider a solution with one master in which each slot is allotted to a distinct item and run this master is run 31,800 times, then this solution produces a wastage of 1,300 for the first item, 600 for the second item, 200 for the third item and no wastage for the fourth item. The total wastage produced is 2,100, a mere 1.7%. The management will clearly prefer this solution, as it uses only one master, even though it is not optimal for the wastage. Note that both the solutions meet all the other constraints and hence are feasible.

Example 1.2. Consider now three items requiring the same quality of paper with size $k = 4$, whose demands are $d_1 = 7287$, $d_2 = 20891$ and $d_3 = 50722$. We can consider the following solution that uses one master in which one slot is assigned to item 1, one slot is assigned to item 2 and two slots are assigned to item 3. This master is run 25,361 times to meet the demand on item 3 exactly, but producing a wastage of 18,074 on item 1 and 4,470 on item 2. The total wastage is 22,544, 28.6% of the total. Now let us consider another solution that uses two masters. On master 1, one slot is allotted to each of items 1 and 2, and two slots are allotted to item 3, but this master is run only 8,940 times, producing a wastage of 1,653 on item 1, and not meeting the demands for the other items fully. Then a second master is used in which no slot is allotted to item 1, one slot is allotted to item 2 and three slots are allotted to item 3. This master is run 11,951 times, to meet the demand on item 2 exactly, but producing a wastage of 3,011 on item 3. The total wastage is therefore 4,664, 5.91% of the total. The management will clearly prefer the second solution although it uses two masters, as the first solution gives a high wastage.

Thus we see that the two objectives we have formulated can be conflicting. The management is not willing to give an estimate of the wastage cost or setup cost. However it is required that the number of masters must be as few as possible (which is an objective of higher priority) and at the same time the wastage is as little as possible. The management will be satisfied to examine a few solutions with minimum wastage, which do not use too many masters, and then choose one of them.

This paper is organized as follows. In Section 2, a mathematical formulation of the problem is presented. In Section 3, an enumeration scheme based on branch and bound ideas is presented. A number of heuristics are proposed in Section 4. We also provide a sufficient condition under which the problem has an optimal solution. Finally in Section 5, as a further refinement of the solution a limited local search algorithm is proposed.

2. The Mathematical Model

In view of the above decomposition, we now formulate the problem of printer optimization for the same type (i.e., size) of items of advertisement that use the same quality of paper. Let there be L items of the same size and quality of paper. Also let $d_1, d_2, \ldots, d_L$ denote the number demanded of these items. The design of loading items on a master sheet can be specified with the help of a pattern vector. A pattern vector p is a column vector of length L, whose coordinates are $x_1, x_2, \ldots, x_L$ satisfying the following conditions.

$$x_j \geq 0, \ \sum_{j=1}^{L} x_j = k, \ x_j \text{ integer}$$

Note that x_j denotes the number of slots allotted to item j on the master sheet with the allocation given by the pattern vector p. We can identify a master sheet having a specified allotment of slots to items with a pattern vector. Thus the four master sheets in the first solution of Example 1.1 can be identified with the pattern vectors,

$$p_1 = [4, 0, 0, 0]^t, \ p_2 = [0, 4, 0, 0]^t, \ p_3 = [0, 0, 4, 0]^t \text{ and } p_4 = [0, 0, 0, 4]^t.$$

The master in the second solution can be identified with the pattern vector $p_5 = [1, 1, 1, 1]^t$. Similarly the master in the first solution of the problem in Example 1.2 can be identified with the pattern vector $p_1 = [1, 1, 2]^t$, and the other master in this example can be identified with the pattern vector $p_2 = [0, 1, 2]^t$. Given L and k the following formula gives the total number of feasible pattern vectors, or the number of possible master designs: $M = \binom{L+k-1}{k} = \binom{L+k-1}{L-1}$. See [3, p. 36].

So we can number the allocation pattern vectors as, $1, 2, \ldots, M$, denoting them as $p_1, p_2, \ldots, p_M$ where p_{ij} will denote the j^{th} coordinate of p_i.

We then have the following mathematical formulation of the printer optimization problem for a specific size k-up and the same quality of paper on which L advertisements have to be printed, the j^{th} advertisement d_j times. Here $n_1, n_2, \ldots, n_M$ are *decision variables* and $y_1, y_2, \ldots, y_M$ are *indicator variables.*

The optimization model for advertisement printing may be stated as follows.

$$\text{Objective 1: Minimize} \sum_{i=1}^{M} y_i$$

$$\text{Objective 2: Minimize} \sum_{j=1}^{L} s_j$$

$$\text{Subject to} \sum_{i=1}^{M} n_i p_{ij} - s_j = d_j, \ j = 1, 2, \ldots, L \tag{2.1}$$

$$n_i \leq 50000, \ i = 1, 2, \ldots, M \tag{2.2}$$

$$y_i(n_i - 5000) \geq 0, \ i = 1, 2, \ldots, M \tag{2.3}$$

$$n_i(1 - y_i) = 0, \ i = 1, 2, \ldots, M \tag{2.4}$$

$$y_i = 0 \text{ or } 1, \ i = 1, 2, \ldots, M \tag{2.5}$$

$$n_i \geq 0, \ i = 1, 2, \ldots, M \text{ integer and } s_j \geq 0, \ j = 1, 2, \ldots, L \tag{2.6}$$

In the above mathematical model, $\sum_{i=1}^{M} y_i$ will give the number of masters corresponding to a feasible solution. The nonlinear constraint $y_i(n_i - 5000) \geq 0$ ensures that if the i^{th} master is run, it is run for at least 5,000 times. The constraint $n_i \leq 50000$ ensures that a master is not used for more than 50,000 times. The first objective (Objective 1) is very important. The second objective is also important, as we would like to ensure the total wastage does not exceed 8 or 10% of the total production. We note that the problem has two objective functions, $2M + L$ variables, M of them 0, 1, $M + L$ of them integer, L linear equality constraints, M linear inequality constraints, and $2M$ nonlinear constraints. For moderate values of L, M is large. Hence this is a difficult combinatorial optimization problem.

2.1. Some Simplifications to the Model

Note that the above is a mixed integer nonlinear programming problem if we treat s_j's as nonnegative real variables. Note that the equations that determine the s_j's ensure that they have actually integer values given that the data d_j's, p_{ij}'s and the decision variables n_i's, are integers. Thus there is no need to explicitly impose the constraints that the s_j's, are also integers. This means that we can solve the problem as mixed integer programming problem. The non-linearity is due to the sets of constraints (2.3) and (2.4). Now the *manufacturer's view* is that (2.3) is a serious constraint. The set (2.4) of constraints can be taken care of while implementing the algorithm ensuring that only if a y_i is fixed as 1, the corresponding n_i will have a positive value.

3. The Computational Strategy

With the simplifications made in the previous paragraph, we see that the problem can be solved as a mixed integer linear programming problem. The slack variables s_j's, are the real variables and the 0–1 variables are the y_i's,. Of course we have two objectives, but this can be taken care of as follows.

We develop an enumeration scheme based on branch and bound ideas, but begin with the least number of y_i's, that have to be at level 1 for a feasible solution to the problem. This number and such a set with the least cardinality can be easily determined. Then in the course of the algorithm when a new y_i is increased to level 1, we examine if this can replace a already positive y_j; if not then the cardinality of the set of y_i's, increase by 1. We develop a branch and bound algorithm similar to the implicit enumeration algorithm of Balas [1]. See also [2, 4] and [5].

Suppose at node v_j, w_j denotes the set of fixed variables, with $w_j^+ = \{i \mid y_i = 1\}$ and cardinality of w_j^+ as r. Let F_j denote the set of free variables. The upper bound on objective 2, $z(v_j)$ is the optimal value of the linear programming problem that results from the set of fixed variables w_j that corresponds to this node. This node can be fathomed if the following conditions hold.

We examine each of the possible successor nodes obtained by replacing a variable from w_j^+ by another free variable and calculate the upper bound for each of these successor nodes. If none of the upper bounds of these successor nodes exceed $z(v_j)$ and if $z(v_j)$ is low, (within the limit preset for the second objective) we fathom this node. Otherwise we branch to that successor node that has the least upper bound. These steps are then repeated as in any branch and bound method. If none of the nodes corresponding to subsets of cardinality r and live, then we branch to a node corresponding to a set of cardinality $r+1$, in case the preset limit on the second objective is not met using the usual branching rule.

As we noted earlier, the above computational strategy is a modification of Balas' implicit enumeration algorithm. It looks for a solution to the mixed integer linear programming problem that has the least value for the first objective and among the solutions with this least value, one that minimizes or meets the requirement on the second objective and is feasible. However the number of successor nodes to be considered from an initial node of a given number of fixed variables can be very large. Further if we require an exact optimal solution, for the second objective (instead of meeting a preset limit) for a fixed value of the first objective, the enumeration tree may turn out to contain a very large number of nodes. So one need to look for other heuristic algorithms. This is developed in the next section.

4. Heuristic Algorithm for the Advertisement Printing Optimization Problem

In this section we propose three heuristics namely, G-MIM, GRBA and AVB to solve the printing optimization problem.

4.1. G-MIM Heuristics (Greedy Heuristics by Maximum Number of Items per Master)

In this heuristic, we obtain an initial solution by using the principle of allotting as many item as possible in each master minimizing instances of duplications. We note that if there are L items for the same type of paper and size k-up, then the minimum number of masters required is $\lceil \frac{L}{k} \rceil$. We can't make fewer than these many masters. We should try to obtain a solution with just $\lceil \frac{L}{k} \rceil$ masters and look for improvement if the wastage has to be reduced further. The steps of the heuristic are given below.

Steps of the Heuristic:

- Step 1 Arrange the L items in the increasing order of demand and renumber the items accordingly.
- Step 2 Consider the first k items from the rearranged list for allotment to the first master using the pattern vector $[1, 1, 1, \ldots, 1]^t$. Determine the set of serial numbers m, $1 \leq m \leq k$, so that when the master is run d_m times, the unsatisfied demand for item $m+1$ is more than 10,000 (or 5,000 as specified by the producer).
- Step 3 For each serial number m obtained in Step 2, calculate the percentage wastage for the master, assuming that the master is run d_m times. Choose the d_m that corresponds to the minimum percentage wastage. Decide to run the first master d_m times.
- Step 4 Obtain a reduced problem eliminating items whose demand has been met by the first master and by considering the remaining demand for items whose demand has been partially met.
- Step 5 Repeat Step 1. If the number of items is less than k, then allot more slots for items whose demand is high in the decreasing order of the demand. Repeat Step 5 until the demand for all the items are met. Calculate the overall percentage. If this is not satisfactory, then for a reduced problem obtained in a previous stage, choose a different serial number m from the set of serial

numbers obtained in Step 2 of that reduced problem and repeat. Calculate the percentage wastage. If all the serial numbers obtained in Step 2 have been considered, then terminate the heuristic with the present solution as the initial solution.

Improved G-MIM: We apply the above procedure to items arranged in the decreasing order of demand and take the best resulting solution

4.2. G-RBA (Greedy Ratio Based Allotment) Heuristics

In this heuristics we allot slots to items on the basis of ratios. The steps of this heuristics are as follows:

Steps of the heuristic:

Step 1 Start with the item with the least demand and a list arranged in the increasing order of demand.

Step 2 Consider the first item in the list. This is the item with the minimum demand. Allot one slot to it in the current master.

Step 3 As long as the ratio of demands $\frac{d_m}{d_1}$ remains less than 1.25, consider allotting one slot to item m on the first master. If the ratio is two times or more but less than 2.50 then allot two slots to item m. Similarly if the ratio is three times or more but less than 3.50 then allot three slots to it. If the ratio is more than 1.25, but less than 2 or more than 2.5, but less than 3, then do not allot it to the current master.

Step 4 Continue allotments in the current master until all the k slots are filled. If there are r free slots in the master and we have an item for which we have to allot q slots ($r < q$), then allot on the first master r slots to this item with a reduced demand of $\lceil r(d/q) \rceil$ and leave the remaining demand for this item to be allotted to the next master.

Step 5 The current master is to be run n times where $n = \max\{i \mid \frac{d_i}{r_i},\ i \in C_i\}$ where C_i denotes the set of items whose demand is completely taken care of by the current master; i.e. items for which the determined number of slots are available in the current master.

Step 6 Repeat Steps 1–5, for the remaining items and items for which there is positive unsatisfied demand.

Step 7 Repeat Steps 1–6 until all the demands are satisfied. (In case we are unable to make allotments to some of the items, then we relax the parameters such as 1.25, 2.50 and so on.)

We apply each heuristic and choose one that is better as a starting solution.

4.3. The Adjacent Vertex Based Heuristics

The adjacent vertex based quick optimization method consists of two routines. Routine one, called the Steepest Descent (SD) routine starts with a solution in the minimum number of masters and seeks to minimize wastage. Routine 2, called the Flattest Ascent (FA) assumes a solution in which the wastage is small and proceeds to construct a solution in which the wastage may be a little higher, but the number of masters is reduced.

This is based on the following easily proved result.

THEOREM 4.1. *Suppose we consider a k-up problem with L items. Suppose $L = Nk$ (an exact multiple of k). Then for the problem of minimizing the wastage in the minimum number of masters, the following is an optimal solution.*

Arrange the items in the increasing order of demand. Allot the first k items in this list to master 1. Allot the items with serial numbers $(m-1)k+1$ to mk to master m, for $m = 1, 2, \ldots, N$. Run master m, d_{mk} times, for $m = 1, 2, \ldots, N$.

Proof. It is clear that the minimum number of masters required in this problem is N. We have to verify that if we are allowed to use only N masters, then the solution specified above is the best. It is easy to note that each master has to have k distinct items and has to be run as many times as the maximum of the demands of the items allotted to a master. If the items are not consecutive on a master then the wastage will go up on such masters and the total wastage will also be more. It follows from here that the solution specified above is an optimal one, in case we are allowed to use only the minimum number, namely, ☐

In case $L = Nk + r$ where $0 < r < k$, clearly the minimum number of masters required is $N + 1$, but can we obtain an optimal solution in this case? It is clear that we have to allot more than one slot in one or more masters to some items, so that all the slots in all the $N + 1$ masters are used. Heuristically it appears that the best candidates for allotment of more than one slot will be the items towards the end of the list. We may allot all these items in the last master, namely master $N + 1$. If r is $k - 1$, then this will yield minimum wastage. In general allotting more slots to the last few items so that the maximum–minimum (i.e.; the range of quantity allotted per slot) is the least possible, should give minimum wastage for $(N + 1)$ masters.

Routine 1 (SD):

Steps of the Adjacent Vertex based Routine SD:

Step 1 We begin with an optimal solution in the minimum number of masters.

Step 2 If the optimal wastage for the minimum number of masters is satisfactory, we may stop here. Else we proceed to Step 3.

Step 3 Work out the inverse of the basis matrix B consisting of the allocation vectors (columns) of the chosen masters and the columns $-I_{.j}$ corresponding to items j that have a positive wastage. Let $y = B^{-1}d$ be the solution vector.

Step 4 Construct the row vector c by taking its j^{th} coordinate to be 1 if the j^{th} column of B is a column of the negative of identity matrix and 0 otherwise. Calculate the weight vector (row vector) w by taking $w = B^{-1}x$.

Step 5 First we check that the current basis is optimal for the given set of masters. For this we should have $w_j \geq -1$, (or $-w_j - 1 \leq 0$) for a coordinate j corresponding to a column of B that is an allocation vector. If this condition is satisfied, then go to Step 6. Otherwise, for j such that $-I_{.j}$ is not a column of B and w_j has the least value that is less than -1, calculate $\theta = \min\{\frac{y_j}{-(B^{-1})_{tj}} \mid -(B^{-1})_{tj} > 0, 1 \leq t \leq L\}$. Let this minimum be attained for $t = v$. Column v of B is replaced by column $-I_{.j}$. Go to Step 9 with $r = v$ and $p = -I_{.j}$.

Step 6 We need to find the set of all allocation vectors p for which $\sum_{i=1}^{L} w_i x_i$ is positive. Since the set of all such allocation vectors is large, we relax it

to finding a set of $2L$ allocations vectors for which $\sum_{i=1}^{L} w_i x_i$ is as high as possible, the maximum, the next maximum and so on. We can proceed as follows:
Let $S = \{j \mid w_j > 0,\ w_j \text{ maximum}\}$. We can allocate all the L slots among the items p in S. This will actually maximize $\sum_{i=1}^{L} w_i x_i$. Let $w_{i_1} \geq w_{i_2} \ldots \geq w_{i_p}$ denote the positive coordinates of w arranged in the decreasing order. We allocate slots to the items $i_1, i_2, \ldots, i_p$ and generate about $2L$ allocation vectors in a systematic manner. Let S_1 denote the set of vectors generated in this manner. Go to Step 7.

Step 7 For each vector p in S_1, compute $B^{-1}p$ and calculate the minimum ratio $p(\theta) = \min\{\frac{y_j}{(B^{-1}p)_j} \mid (B^{-1}p)_j \text{ is positive}\}$. Go to Step 8.

Step 8 Choose p in S_1 such that $p(\theta)$ is maximum. Suppose that the minimum ratio for this p is attained for $j = r$. Go to Step 9.

Step 9 Replace column r of B by p and update the inverse and the solution. If this step has been reached from Step 5, then go to Step 4. Otherwise go to Step 10.

Step 10 If further reduction in the wastage is desired go to Step 4. Else terminate. Or terminate if the number of masters is more than or equal to the number of masters already obtained by the other heuristics with satisfactory wastage.

Routine 2 (FA):
Suppose we have a solution in $m + 1$ masters (may be more than what we believe is required) but with a small wastage. Suppose we want to see if the number of masters can be reduced at the cost of a little more of wastage. We can then apply the following routine. This may be one of the solutions obtained by some other heuristics or by applying Routine 1 of the LP based heuristics. Typically the number of masters $m + 1$ is small compared to the number of products L.

Step 1 We have the basis matrix B of order L corresponding to the given solution. Compute the inverse matrix B^{-1}. Let y denote the solution vector $B^{-1}d$. Go to Step 2.

Step 2 Let the weight vector w be calculated as in step 4 of the routine SD. Let S be the set of indices t such that $-w_t - 1$ is negative. Let $t(\theta) = \min\{\frac{y_j}{(B^{-1})_{jt}} \mid (B^{-1})_{jt} > 0\}$. This minimum is likely to be attained for a variable of type n_r. Choose v in S such that $v(\theta)$ is the minimum among all the $t(\theta) > 0$, for t in S. Let this minimum be attained for some v and row $j(v)$. Then $-I_{.v}$ replaces column $j(v)$ of B. Go to Step 3.

Step 3 Update the inverse and the solution. If the starting solution in $(m + 1)$ masters, of this routine has been obtained by the routine SD from a solution in m masters and the solution in step 2 is the same as this solution, then it proves the local optimality of the m master solution in step 2. We can make no local improvements and we terminate the solution. If the solution in step 2 is still a solution in $(m + 1)$ masters, then we go back to Step 2.

As an example consider a four up problem with six items and the demand of the six items in the increasing order are 20,900, 21,000, 23,700, 25,600, 31,800, 32,300. So, the demand vector $d = [20900, 21000, 23700, 25600, 31800, 32300]'$.

We start with the initial solution given by the increasing order of items.

Master 1	20,900, 21,000, 23,700, 25,600; run 25,600 times.
Master 2	allot two slots each to items 5 and 6 and run the master 16,150 times.

The wastage for the initial solution is given by: $s_1 = 4700,\ s_2 = 4600,\ s_3 = 1900,\ s_4 = 0,\ s_5 = 500,\ s_6 = 0$; the percentage wastage is: 7.53%.

We start the adjacent vertex SD routine with the initial basis matrix, say

$$A_1 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

The first three columns of A_1 correspond to s_1, s_2 and s_3, column four is master 1 allocation, column 5 corresponds to s_5 and column 6 is master 2 allocation vector.

The inverse of A_1 is given by

$$B_1 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$

Note that $B_1 d$ gives the solution vector. The wastage corresponding to this is about 7%

The next step of the routine SD is to calculate the row vector c, which is given by: $c = [1, 1, 1, 0, 1, 0]$; 1 for the surplus (wastage variables in the basis, and 0 for master allocation vectors in the basis matrix A_1, in the order in which these columns appear in A_1. The next step is to calculate $w = cB_1$. This row vector is obtained as: $w = [-1, -1, -1, 3, -1, 1]$.

We next try to generate as many solutions as possible to:

$$\sum_{j=1}^{L} w_j x_j > 0,\ x_j \geq 0,\ \text{integer} \tag{4.1}$$

$$\sum_{j=1}^{L} x_j = k \tag{4.2}$$

We can generate as many solutions to the above as possible. The algorithm suggests that we generate $2L$ (This can be modified).

We note that the maximum coordinate in w is coordinate 4, which is 3, the next maximum is at coordinate 6, which is 1. The other coordinates are -1. We should generate the x vectors that have different values for the weighted sum.

We generated the following solutions.
$t_1 = [0, 0, 0, 4, 0, 0]'$; $B_1 t_1 = d_1 = [4, 4, 4, 4, 0, 0]'$,
Min ratio: $\min (S_1)_j/(d_1)_j \mid (d_1)j > 0$.
We present these solutions in the following table:

Generated allocation vector t (col. vector)	$B_1 t = d$ (col. vector)	Minimum ratio $(S_1)_j/d_j$, for $d_j > 0$
$t_1 = [0, 0, 0, 4, 0, 0]$	$d_1 = [4, 4, 4, 4, 0, 0]$	1900/4
$t_2 = [0, 0, 0, 2, 0, 2]$	$d_2 = [2, 2, 2, 2, 2, 1]$	500/2
$t_3 = [0, 0, 0, 3, 0, 1]$	$d_3 = [3, 3, 3, 3, 1, 0.5]$	500/1
$t_4 = [0, 0, 0, 1, 0, 3]$	$d_4 = [1, 1, 1, 1, 3, 1.5]$	500/3
$t_5 = [1, 0, 0, 1, 0, 2]$	$d_5 = [0, 1, 1, 1, 2, 1]$	500/2
$t_6 = [0, 1, 1, 1, 0, 1]$	$d_6 = [1, 0, 0, 1, 1, 0.5]$	500/1
$t_7 = [0, 0, 0, 2, 1, 1]$	$d_7 = [2, 2, 2, 2, 0, 0.5]$	1900/2
$t_8 = [1, 0, 0, 2, 0, 1]$	$d_8 = [1, 2, 2, 2, 1, 0.5]$	500/1
$t_9 = [0, 0, 0, 0, 0, 4]$	$d_9 = [0, 0, 0, 0, 4, 2]$	500/4
$t_{10} = [0, 0, 1, 1, 1, 1]$	$d_{10} = [1, 1, 0, 1, 0, 0.5]$	4600/1

We find that maximum of the minimum ratio occurs for the allocation vector $[0, 0, 1, 1, 1, 1]'$. We note that when an allocation vector is introduced into the basis, the minimum ratio gives actually the number of times the master with this allocation vector will be run. So none of the allocation vectors listed above (including t_{10}) can be actually considered for use, as they are considered infeasible. However the allocation vector t_{10} is worth trying as the minimum ratio for this is 4600. We can check how much improvement is obtained by using this allocation vector. If even when this is run 5000 times instead of 4600 times (which may increase the wastage) if the resulting wastage is not much, the solution would be worth considering.

In the course of solving this problem, we can learn certain rules for the generation of good allocation vectors that satisfy (4.1) and (4.2). We need to set x_5 and x_3 corresponding to s_5 and s_3 where the surplus is not much to 1. On the other hand, a coordinate for which the corresponding surplus is high should be set equal to 0. So $x_1 = x_2 = 0$, $x_3 = x_4 = x_5 = x_6 = 1$. (Such a vector must be a candidate vector. However as there is no proof that in general this will be a good allocation vector, we need try out others also).

So the basis matrix now becomes A_2 where

$$A_2 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 2 \\ 0 & 1 & 0 & 0 & 0 & 2 \end{bmatrix}$$

and its inverse is given by

$$B_2 = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0.5 & 0 & -0.5 & 0 & 0.5 \end{bmatrix}.$$

The solution vector corresponding to this is given by $[100, 4600, 1900, 21000, 500, 13850]'$, which means that the wastage is 100 + 1,900 + 500 = 2,500; the third master is run 4,600 times, the first master is run 21,000 times and the second master is run 13,850 times. The wastage is 1.6%. If we run the first master another 400 times the wastage increases to 2,500 + 1,600 = 4,100. The wastage is 2.64%.

So one can choose between three masters and wastage of 2.64% or two masters and 7.53% wastage.

The SD routine can be terminated at this point as we have a reasonable solution. But can we get a solution in two masters which is better than the two master solution we have? To answer this we need to apply the FA routine to the present solution. We note that now the row vector $c_2 = [-1, 0, -1, 0 - 1, 0]$ and $c_2 B_2 = w_2 = [-1, 1, -1, 1, -1, 1]$. The missing surplus variables are s_2, s_4 and s_6. Now $w(-I_{.2}) - 1 = -2$ which is also equal to $w(-I_{.4}) = w(-I_{.6})$. So if we introduce any of these variables the wastage will increase. Now we check that $B_2(-I_{.2})$ takes us back to the two master solution we started with, as the minimum ratio with this column eliminates the third master introduced. $B_2(-I_{.6})$ has only negative entries and hence in this context not acceptable. That leaves $B_2(-I_{.4})$ and this makes only the last entry of the solution vector eligible for the minimum ratio test. This ratio is 13,850/0.5 = 27,500. This is the wastage s_4 that would be produced if the second master is eliminated from the basis by the entering s_4. In addition there would be increased wastage for s_3. This is also clearly unacceptable solution. So Routine FA shows that there is no solution other than the initial solution that is adjacent to the three master solution we obtained. In the neighborhood therefore there is no better two master solution, than the initial solution. With this conclusion we terminate the SD and FA routines.

5. Further Refinement

As an initial solution, we can use each of the above heuristics with different values for the parameters and generate a set of good initial solutions. From among these we may choose the best one. If we use routines SD and FA, one after the other, then the final solution obtained will be locally optimal. To further optimize we suggest the following limited local search algorithm. We maintain a list of allocation vectors generated in the earlier steps or over the previous occasions for this product and pattern of demand. Suppose this list is T. Let m denote the number of masters in which we are interested in getting a solution. Let β be the cardinality of T. We then consider all possible combinations of β choose m and try out the solutions by enumeration to obtain a better solution.

Steps of the refinement:

Step 1	Make a candidate list T of allocation vectors. Let β be the number of such candidate vectors. Go to Step 2.
Step 2	Suppose we have a solution in m masters from T and B is an optimal basis corresponding to this solution. Calculate B^{-1}. Let $y = B^{-1}d$ be the corresponding solution vector. Let M denote the set of the allocation vectors $\{p_1, p_2, \ldots, p_m\}$. Go to Step 2.
Step 3	Choose any vector p from T not included in M. Go to Step 3.
Step 4	Do the feasibility test on each set $M_r = M\backslash\{p_r\} \cup \{p\}$. This essentially consists in checking if for each item j there is at least one vector in M_r with 1 or more in coordinate j. Go to Step 5.
Step 5	For each feasible M_r, replace p_r from B by p. Update the inverse and the solution. Calculate the corresponding wastage. Go to Step 6.
Step 6	Repeat Steps 3–5 for each vector p from T, not included in M. Go to Step 7.
Step 7	Choose the solution from among generated in Step 5, for which the wastage is minimum.
Output	A solution in m masters which has the least wastage among the vertices adjacent to the starting solution, from the set T.

References

1. Balas, E.: An additive algorithm for solving linear programs with zero–one variables. Oper. Res. **13**, 517–546 (1965)
2. Geoffrion, A.M.: Integer programming by implicit enumeration and balas method. SIAM Rev. **9**, 178–190 (1967)
3. Feller, W.: An Introduction to Probability Theory and Its Applications. Wiley, New York (1967)
4. Garfinkel, R., Nemhauser, G.L.: Integer Programming. Wiley, New York (1972)
5. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley, New York (1988)