Division of Research
Graduate School of Business Administration
The University of Michigan

February 1980

A 0-1 MODEL FOR SOLVING THE CORRUGATOR TRIM PROBLEM

Working Paper No. 205

Robert W. Haessler
and
F. Brian Talbot

The University of Michigan

FOR DISCUSSION PURPOSES ONLY

None of this material is to be quoted or reproduced without the express permission of the Division of Research.

ABSTRACT

This paper describes a new 0-1 formulation for solving the corrugator trim problem. This approach eliminates the spreading of orders over several stock sizes, a characteristic which has plagued the linear programming-based procedures that have been proposed for solving the problem. The improved material handling characteristics of the solution are obtained through the controlled generation of solution elements. The elements that provide a minimum cost solution are then selected using a 0-1 backtracking algorithm that has been modified for this application.

Introduction

Over the last 20 years, there have been a number of attempts to develop computer models for solving the trim problem in the corrugated container industry. The starting point for these efforts can be traced to the early papers by Paull and Walter [7] and Eisemann [1] which described linear programming formulations for the general trim problem in the paper industry. The improvements in the linear programming algorithms for solving these problems by Gilmore and Gomory [3,4] provided further impetus. However, there was an early recognition that the corrugator trim problem could not be completely modelled using linear programming [6] because of the nature of the production process. This in turn, led to the development of sequential heuristic procedures, the most notable of which was due to Van Wormer [8].. These heuristic procedures were unreliable, however, because of their inability to consider economic tradeoffs in a systematic manner. As a result, corrugator trim problems are still for the most part solved manually [5].

The purpose of this paper is to present a new approach to the corrugator trim problem that overcomes the major difficulties with previous attempts to solve this problem. The use of conveyorized material handling systems in plants makes it extremely important that orders are produced contiguously and not spread out over the entire production run as it typical in linear programming solutions to cutting stock problems. The elimination of order spreading is accomplished by the controlled generation of solution elements that completely satisfy one or more orders. An optimal solution is obtained by selecting that subset of elements with minimum cost that covers the orders to be produced. This is an improvement over heuristic procedures because it does permit an explicit economic tradeoff among corrugator utilization, trim loss, and slitter and stock size changes.

An overview of the operation of a corrugator plant is presented in the following sections along with a detailed discussion of the economic factors that must be taken into consideration when combining orders across the width of the corrugator. The remaining sections of the paper are devoted to a discussion of a 0-1 model and the solution algorithm that can be used to solve the problem.

Corrugator Plant

A corrugator processes stock rolls of liner and medium to form corrugated material from which shipping containers are made. These input rolls are generally stocked in a variety of widths to limit the amount of scrap generated. Because corrugated material is bulky and rigid, it must be cut into rectangular blanks of the sizes needed to fill customer orders while still on the corrugator. The solution to the corrugator trim problem is simply a specification of what must be done to obtain the needed quantities of rectangular blanks of the sizes ordered. This involves specifying the stock roll sizes to be used, the feet of corrugated material to be produced from each stock size and the cutting instructions for each stock size to produce the required order sizes and quantities.

A key determinant of the capacity of a corrugator is the maximum width of the stock rolls it can process. A typical corrugator is able to process rolls up to 87 inches in width. To obtain a high degree of utilization of the corrugator, orders for various sizes should be combined across the width so that the total width of the orders is close to but does not exceed the usable width of the output of the corrugator. This combining problem is more difficult than solving a one-dimensional trim problem because the manner in which the blanks are cut on the corrugator and handled in the plant severely restrict the ways orders can be combined across the width of the corrugator. The cutting is done by slitting the corrugated material into strips of the appropriate width and then chopping the strips into pieces of the ordered lengths. On most corrugators, the strips can be chopped to at most two different lengths, thereby restricting the number of ordered sizes that can be combined across the width of the corrugator. Once the rectangular blanks have been cut, they may be shipped directly to the customer or processed through various operations to obtain finished shipping containers. In modern plants, material is moved to the primary finishing operations on conveyors with limited room for in-process storage. This makes it important that each order is produced contiguously so that it can be kept together on the conveyors.

Scheduling Environment

Because of the bulky nature of corrugated shipping containers and the absolute necessity of having them available when needed, good customer service is a key success factor in this business. Every effort is made to supply the material to the customer on time. To insure on time delivery, the production scheduler typically maintains an order file of items not yet scheduled that is sorted by grade of corrugated material and due date. The additional information that must be maintained to schedule the order is:

- blank size (width and length)
- required quantity
- finishing operations required

The production scheduler will typically review the entire order file each day and generate a corrugator schedule one day at a time. It is possible that virtually every grade a plant makes will be run every day. This is unlike the paper industry where a given grade may be run no more frequently than every two weeks. The service requirements in the box business do not permit large scale batching of orders for infrequent runs to attain production economies. Fortunately, the changeover costs from one grade to another are not very large. Therefore the production economies foregone by frequent runs of a grade relate primarily to combining options rather than changeover costs.

The set of factors that the scheduler must consider in determining which orders to make and how to combine them is given below.

1) Order date - This is used to determine what must be made, what can be made if it improves corrugator utilization, and what should not be made yet

because the shipping date is too far off. The determination of which non-critical orders get made on any day is based on the economics of corrugator utilization in conjunction with the supply-demand situation. The corrugator is normally scheduled to run 24 hours a day, 5 days a week. Except during times of very high demand, orders are accepted based on the customers required date without checking schedule feasibility. When demand is low, future orders are pulled forward to keep the corrugator running as long as possible. When demand is high, only the orders absolutely needed are scheduled. Weekend overtime is used to try to keep up during peak periods. During periods of slack demand, the corrugator may be shut down during the week for one or more shifts.

2) Finishing equipment backlogs - It is desirable to keep the backlog on the primary pieces of finishing equipment balanced. As a result, this consideration may influence the orders to be scheduled and the way in which orders are combined.

3) Corrugator width utilization - Productivity of the corrugator is partially controlled by the width of usable material produced. By utilizing more of the width capacity, the time required to produce a given set of orders can be reduced. It should be noted that the value of corrugator time will vary depending on the supply-demand situation. During periods of slack demand, the incremental value of an hour of corrugator may be low, reflecting only the out-of-pocket operating costs. Whereas when demand is high, the value of corrugator time should reflect the opportunity cost of incremental business and margin.

4) Side trim - To limit scrap losses, rolls of liner and medium are generally stocked in more than one width. For example, if three 25 inch width blanks are to be run on an 87 inch corrugator, they might be cut from 77 inch stock rolls with 2 inch side trim instead of the 12 inches that would result

if only 87 inch width rolls were available. However, it is impossible to completely eliminate trim losses. In general a loss of about .375 inches per side is required in order to insure that blanks meet dimensional tolerance criteria. So even if the paper were available in 75 inch rolls, this size could not be used for the three 25 inch blanks.

5) Cutting pattern changes - Because the blanks are cut on the corrugator itself, there is a loss of both time and materials when the blank sizes are changed. Therefore, the scheduler attempts to minimize the cutting pattern changes. Changing cutting patterns involves rotating a mechanism which holds three sets of slitters and adjusting the length cutoff. The setting of the slitters for two cutting patterns can be done while the corrugator is running. However, care must be taken so that pattern changes do not occur so frequently that there is insufficient time to set the slitters before the corrugator stops for the next pattern change.

6) Upgrading - In certain situations, because of excessive side trim or low corrugator utilization, it may be more economical to provide the customer with a higher quality box than was ordered. The customer pays the regular price for the box ordered. The incremental value of the material "given away" is recovered by combining the order in question with orders for different sizes of the higher quality grades and thereby reducing side trim or increasing corrugator utilization.

7) Partial orders - Processing and handling costs after the corrugater may be increased substantially if each order is not produced contiguously during a run. Failure to accumulate all the blanks needed for an order before starting the finishing operations may result in multiple setups on these machines. To avoid multiple setups for orders which are produced in partial lots, the material must be staged after the corrugator. This can be difficult and expensive in a plant that uses conveyors for movement and storage.

8) Stock width changes - There is a cost when a stock width change is made in addition to the cost of the accompanying knife change. For corrugators that have automatic splicers and process control this may be nothing more than the cost of moving stock rolls in and out of inventory. Without automatic splicers there is also some lost time and material associated with a stock width change.

9) Stock availablity - At times there may be limited amounts of any roll stock size in inventory. The availability of roll stock must be considered in solving the corrugator trim problem.

It should be clear from the proceeding list that there are a number of tradeoffs that must be considered in combining orders for production on a corrugator. A simple example can be used to demonstrate the nature of the tradeoffs and the type of information required to make them. Suppose an order for 48,000 lineal feet of 16½ inch blanks is to be produced. The stock sizes available range from 67-87 inches in 2-inch increments. Assume that there are no other orders available that can be combined with this one. The order can be run either 4-out from 67 inch stock rolls or 5-out from 85 inch stock rolls. The question is which is the most economical way to run the order? The key issue in this case is the tradeoff between the value of paper and time on the corrugator. Running the order 5-out requires 9,600 feet of corrugator production and results in 2.5 inches of side trim, whereas running 4-out requires 12,000 feet of corrugator production and results in 1 inch of side trim. The total scrap generated by running 5-out is 2,000 square feet and 4-out is 1,000 square feet. The determination of which way to run the order rests on a comparison of the value of 1,000 square feet of scrap and the value of the corrugator time required to produce 2,400 lineal feet.

Although the above tradeoff is straightforward and easy to make if the required data is available, it should be clear that in a multi-order problem the tradeoffs can become much more difficult to identify and evaluate. In a multi-order problem there may be a large number of possible ways to combine sizes on the corrugator. These alternative solutions may involve differences in pattern and stock width changes as well as differences in scrap generated and corrugator time required. What is needed to identify and evaluate the tradeoffs in a systematic way is a mathematical statement of the problem that is computationally feasible. It is possible to formulate the problem as an extension of the one-dimensional cutting stock problem (see 5). But, the resulting model can only be solved heuristically because of the combinatorial nature of the problem. The difficulty with heuristic procedures for generating solutions is that they have either attempted to generate patterns sequentially and as a result have had trouble with trim loss at the end of the procedure, or have used linear programming to minimize scrap losses and have performed poorly with regard to pattern changes and order contiguity.

The next section presents an alternative formulation of the problem which is based on observed scheduling practices in modern corrugator plants where the primary objective is to avoid making only part of an order from a given stock size. This approach overcomes the drawbacks mentioned above.

0-1 Formulation

Instead of formulating the problem in terms of cutting pattern variables, a much more manageable formulation can be developed by focusing on solution elements. A solution element is a specification of the way in which one or more orders can be completed from a single stock size. The four types of solution elements currently being considered are listed below.

1. Producing one order by itself. For example cutting three 25 inch width blanks from a 77 inch stock size.

2. Producing two orders from a single cutting pattern. For example, cutting two 20 inch blanks and two 18 inch blanks from a 77 inch stock size. This can be done if and only if the quantity requirements are such that both orders are simultaneously completed within the allowable quantity tolerances. Typical industry practice is to produce an amount that is in an interval from the order quantity to the order quantity plus 10%

3. Producing two orders from two cutting patterns from a single stock size. For example, cutting one 25 inch blank and one 51 inch blank from a 77 inch stock roll until the order for 51 inch blanks is filled and then finishing the order for 25 inch blanks by cutting them three across.

4. Producing three orders from two cutting patterns from a single stock size. For example, cutting two 25 inch blanks and one 26 inch blank from a 77 inch stock size until the 26 inch order is completed, and then cutting two 25 inch blanks and two 12 inch blanks until these two orders are completed. This again requires a specific relationship among the quantities required.

In both the third and fourth elements an order is spread over 2 pattern setups from the same stock size. This, however, is not considered partialling the order because the patterns are run sequentially.

The corrugator trim problem can be formulated as a 0-1 model by using elements of the type listed above as follows:

$$\text{Min } \Sigma_j c_j x_j + \Sigma_k s_k y_k$$

$$\Sigma_j a_{ij} x_j = 1 \text{ for } i = 1,\ldots, m$$

$$\Sigma_j u_{jk} x_j \leq A_k y_k \text{ for } k = 1,\ldots, k$$

$$x_j = 0 \text{ or } 1 \quad y_k = 0 \text{ or } 1$$

Where

x_j is 1 if element j is used and 0 otherwise ($j = 1,\ldots,n$)

y_k is 1 if stock size k is used and 0 otherwise

a_{ij} is 1 if order i is completed in element j and 0 otherwise

u_{jk} is the feet of stock size k required by element j

A_k is the feet of stock size k available in inventory

c_j is the total cost of using element j exclusive of the cost of changing to the required roll stock size. It includes the cost of corrugator time and paper used plus the cost of pattern changes. This value is adjusted if any order is not produced at the maximum quantity to reflect the cost of producing the whole order.

s_k is the cost of loading stock size k on the machine. It is assumed that if two or more elements use the same stock size, they will be run sequentially so there will only be one setup for each stock size.

The upgrading of orders and the use of optional sizes can be easily handled in this formulation. An order can be included in several elements involving different stock sizes and grades of paper. The only requirement is that an order not be included in an element using lower quality paper than specified on the order. Optional items can be handled by reducing the cost of the elements in which that item is produced by itself. If that element is in the solution, it is not scheduled. If the optional order is produced in some other element in the solution it is because there is sufficient cost saving on some other order to overcome the reduced cost available from producing the order by itself.

Solution Procedure

The solution procedure for the 0-1 model is an adaption of the set-partitioning algorithm introduced by Garfinkel and Nemhauser [2]. Although there exist a number of approaches for solving 0-1 integer problems, several factors motivated the use of this particular procedure. Typically, corrugator plants have available only small, relatively slow computer systems which can be used for scheduling purposes. These limitations mitigate against the adoption of optimizing codes which require either large amounts of computer core, or if they are to be cost effective, much computer time. Garfinkel and Nemhauser's approach is a list processing, implicit enumeration technique that has very low core requirements and has been shown [2] to be a relatively fast method for optimally solving the set-partitioning problem, which is a subproblem of (1)-(3). In addition, this approach, is attractive because of the ease with which inventory constraints and changeover costs can be included without significantly increasing core requirements. Finally, this 0-1 backtracking method yields good feasible solutions very quickly. This characteristic permits the scheduler to terminate program execution prior to reaching optimality with good heuristic solutions in-hand, if program running time does become excessive.

The basic approach is to solve the set-partitioning subproblem (1) and (2) using Garfinkel and Nemhauser's six-step algorithm but with additional tests included to insure that stock availability (3) is not exceeded and that changeover costs in (1) are considered when appropriate. The adaptation details which follow use the notation in [4] for consistency.

During the initialization phase the rows in the a_{ij} matrix are dynamically sorted such that $i \leq e$ implies $\Sigma_j a_{ij} \leq \Sigma_j a_{ej}$. Dynamically sorted in this case means that the sorting criterion is recalculated m-1 times, where the above sums are taken each time over only those j where $a_{ij} = 0$ in all rows i that have been renumbered. This procedure appears to be superior to the single pass sort suggested in [2], although in either case the objective is the same: number the most restrictive constraints so that they will be evaluated first by the implicit enumeration algorithm.

Also during the initialization phase the variables x_j are assigned to m lists as in [2] such that x_j is in list i if and only if $a_{ij} = 1$ and $a_{ej} = 0$ for all $e < i$. Rather than sorting the variables within each list in order of nondecreasing cost c_j as is suggested in [2], however, they are sorted in nondecreasing order of a surrogate cost priority index. This sort is much more effective for the corrugator problem because of the nature of the solution elements. The priority index is set at -1 for all type 1 elements (elements involving only an order). For all other element types involving more than one order the priority index is calculated as the potential saving by using this element instead of the set of type 1 elements that satisfy the same orders.

Both stock changeover costs and stock availability constraints are evaluated during step III (building the partial solution). A_k and y_k are used, respectively, to indicate the lineal feet of each stock width currently available and whether a given stock width is in the current partial solution. Variable x_j is only brought into solution if $u_{jk} \leq A_k$, and the sum of the current partial solution costs plus all the costs of bringing j into solution do not exceed the cost of the best (incumbent) heuristic solution generated. Of course, during the backtracking step, A_k is increased by u_{jk} when x_j comes out of solution. Similarly, all cost elements are updated.

Example Problem

An example problem with 15 orders is presented in Table 1. The width and length refer to the size of the blank required to fill the customer's order. The quantity is the number requested by the customer. An order is within tolerance if the number scheduled does not exceed the order quantity by more than 10%. Given a choice, the quantity scheduled will be the upper limit.

An optimal solution to the example problem is given in Table 2. The solution is presented in terms of the patterns and stock sizes to be used and the lineal feet of corrugated material to be produced for each pattern. This solution was obtained and verified as being optimal in .696 seconds of CPU time on an AMDAHL V/7 computer using the standard FORTRAN G compiler. The problem solved had 62 solution elements.

Because the practical usefulness of this technique is closely related to the computer costs of running the program, it is worth discussing several computational issues. In order to further reduce CPU core and time requirements, the a_{ij} matrix is replaced with a matrix of column and row indices indicating the location of unit coefficients. The a_{ij} matrix is sparse (density is typically under 15%) with no more than three unit coefficients in each column. This permits the mxn a_{ij} matrix to be replaced by a 3xn matrix. Core requirements are reduced somewhat by this procedure, but more important is the 300% improvement in CPU time this approach affords because it reduces tha amount of list processing required. Even though the resulting algorithm is fast enough to be run to optimality for most problems, there are occasions when the cost of obtaining the optimal solution simply outweighs the benefit of having it. To accomodate this situation, the algorithm can easily be programmed with stopping rules based on running time, or number of iterations. One method which seems to have merit is to terminate program execution when an improved solution is not found within three times the cumulative CPU time needed to obtain the previous solution. We have found that this rule almost always permits the

Table 1

EXAMPLE PROBLEM

ORDER	WIDTH INCHES	LENGTH INCHES	QUANTITY ORDERED
A	24.6250	62.250	4000
B	22.5000	66.625	5000
C	19.4375	54.500	2000
D	19.1875	57.250	3500
E	19.9375	63.250	5000
F	18.0625	47.500	5000
G	19.9375	47.500	5000
H	20.5625	54.500	3600
I	21.6250	48.250	2400
J	16.8125	83.750	3000
K	21.6875	55.750	3000
L	21.5625	63.000	2000
M	16.5625	49.250	5000
N	42.0000	37.875	750
O	43.3125	59.625	5000

Stock Sizes Available Range From 67 to 87 Inches in 2 Inch Increments

Economic parameters for this problem

- corrugator value $100/hour
- corrugator speed 300 feet per minute
- slitter charge $10
- stock charge $5
- paper cost $15/thousand square feet
- paper weight 125 pounds/thousand square feet
- minimum edge trim .375 inches/side

algorithm to find the optimal solution, but without consuming an inordinate amount of time verifying optimality. Had this rule been applied in the example problem, total CPU time would have been .219 seconds (it took .073 seconds to find the best solution) instead of .696 seconds. Further justification for using this rule is based on the observation that for all the problems solved thus far, proving optimality consumes from three to ten times as much CPU time as does simply finding the optimal solution.

Table 2

EXAMPLE PROBLEM SOLUTION

STOCK SIZE	PATTERN*	PRODUCTION IN FEET
87	4L	2,890
87	2A and 2F	10,890
85	2N	1,300
85	5J	4,610
85	5M	4,520
83	2I and 2C	5,000
83	2D and 2K	7,670
	2D and 1P	1,370
83	4H	4,540
81	2E and 2G	10,890
69	3B	10,180

* A pattern designated as 4L means that 4 blanks for order L, 21.5625 inches wide, are combined across the width of the corrugator.

Optimal Value of the Objective Function Is $7068.

Summary

The corrugator trim problem involves some very difficult tradeoffs between linear and fixed costs. As such it appears doubtful that the problem can be formulated and solved in the format of a cutting stock problem. The 0-1 approach presented in this paper leads to a compact formulation which can be solved optimally in a reasonable amount of time. The solution elements used are comparable in complexity to those used by schedulers. One major advantage of this approach over manual solution procedures is the ability to generate multiple elements that contain the same order and to then select the one that gives the best overall result with all economic factors and order requirements considered.

In addition to the obvious use in day-to-day scheduling of corrugator, the approach defined above can be used to carry out studies that would otherwise be impossible. One of the most important of these would be an analysis of the impact that the number and choice of stock sizes has on total corrugator operating costs. It is well known that raw material inventory is directly related to the number of sizes stocked. By using this model to solve the same problems with various numbers and combinations of stock sizes, it will be possible to determine the expected increase in corrugator operating costs associated with a reduction in the number of sizes stocked. This can then be compared to the expected reduction in carrying costs to come to a determination of a rational stocking size program for any box plant.

References

1. E. Eisemann, "The Trim Problem," Management Science 3, 279-284 (1957).

2. R. S. Garfinkel and G. L. Nemhauser, "The Set-Partitioning Problem: Set Covering with Equality Constraints," Operations Research 17, 848-856.

3. P. C. Gilmore and R. E. Gomory, "A Linear Programming Approach to the Cutting-Stock Problem," Operations Research 9, 848-859 (1961).

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

5. R. W. Haessler, "The Corrugator Trim Problem," Paper presented at the 1977 Joint National ORSA/TIMS Meeting in Atlanta.

6. J. L. Marley and D. Mahoney, "Can Cutting Trim Waste Cut Into Your Profits?" Paperboard Packaging, (Oct. 1963).

7. A. E. Paull and J. R. Walter, "The Trim Problem - An Application of Linear Programming to the Manufacture of Newsprint," Paper presented at the Annual Econometric Society Meeting, Montreal (Sept. 1954).

8. T. Van Wormer, "The Trimmer: A Heuristic Solution to the Trim Problem in the Corrugated Container Industry," Unpublished doctoral dissertation, Carnegie Institute of Technology.(1963).