European Journal of Operational Research 44 (1990) 289-299
North-Holland

Load planning for shipments of low density products

Robert W. HAESSLER and F. Brian TALBOT
School of Business Administration, The University of Michigan, Ann Arbor, MI 48109-1234, USA

Abstract: This paper presents a complex computer-based heuristic procedure for sizing customer orders and developing three dimensional load diagrams for rail and truck shipment of low density products. This heuristic procedure was developed for, and is in various phases of implementation at, a large multinational U.S.-based consumer products company. Products are shipped daily in high volume from inventory in corrugated containers of various sizes depending on the product package sizes and customer requirements. Vehicles used include railcars, truck trailers and tandem truck trailers, which also vary in size depending upon need and availability. In most cases, product volume or material handling considerations limit the amount of product loaded into vehicles before weight restrictions are met. Hence, the emphasis here is on low density products. The procedure developed has been demonstrated to significantly increase vehicle utilization, and improve customer service. It is fast and accurate enough to be used in real time during the order entry process. It has also been used successfully in a vehicle feasibility study of single versus tandem trailers.

Keywords: Vehicle load planning, heuristics, bin packing

1. Introduction

This paper describes a complex heuristic procedure for sizing customer orders and developing load patterns for truck and rail shipments of low density products. The density of the products is such that truck trailers as well as rail cars are loaded primarily based on volume rather than weight restrictions. The new computerized procedure is designed to run on-line at order entry time. It replaces an existing manual method for determining how large a customer order should be, and how it should be loaded into a vehicle. In order to understand how the new heuristic procedure operates, a brief description will first be given of the order entry system, and the old method of sizing and load planning.

Customer orders are taken over the phone by order entry personnel. Once the order is accepted, it is processed through the production planning and inventory control system, and in several hours to several days, it appears as a pick list for a clamp truck driver in the warehouse. Under the old system, the clamp truck driver would obtain the products on the pick list from the warehouse and would load them into the vehicle using his best judgment. The customer order would not be modified unless the driver could physically not fit some of the product onto the vehicle. In this case, the unloaded product would usually be shipped with some future order.

The order pricing system, and the nature of transportation costs, encourage the customer to order in full carload or truckload quantities. However, because the products are all packed in corrugated boxes of varying sizes, and each order typically contains multiple products, it is very difficult to know at order entry time if any order will completely utilize the cubic volume of the shipment vehicle. Based on past loading experience, the company developed written guidelines that would help a customer determine a 'good'

Received May 1988; Revised January 1989

order size. Since it wasn't known with certainty if the order would fit onto the vehicle until the loader physically tried to load it, and the company wanted to avoid shipping an order short, these guidelines were very conservative. A a result, cubic utilization was often significantly less than it could have been.

With the new procedure, information about the order and intended shipment vehicle are entered into the loading heuristic at order entry time. Order information includes product type and quantities, and how the order can be adjusted up or down. While the customer is still on the phone, the loading heuristic develops an actual load plan that specifies exactly (to a tenth of an inch) what should be loaded and how it should be loaded in the vehicle. The order is authorized by the customer, or alternative orders are examined for loading feasibility until an acceptable order is determined that trades off vehicle utilization and customer product needs. The load plan is then transmitted to the warehouse where it tells the driver exactly how to load the vehicle.

No explicit mathematical statement of the problem is developed and therefore no claim can be made for optimality. The development of an explicit model is made impossible by the nature of the non-physical constraints. The desire is to maximize the size of the order in cubic feet, while at the same time:

1. Avoiding the risk of product damage.
2. Limiting the time required to load the vehicle.
3. Keeping each product in the order as close together as possible to minimize the effort in stack building, and in unloading inventory and storing the product at the receiving location.

It is clear that if corrugated boxes containing the product ordered were individually hand loaded into the vehicles, significant improvements in cube utilization would be obtained. It is also clear that the increased labor cost and loading time of doing this would far outweigh any benefits provided by a reduction in transportation cost.

2. Literature review

There is a rich literature dealing with cutting stock and packing problems. The seminal work of Gilmore and Gomory (1961, 1963, 1965) demonstrated how linear programming could be used to maximize yield in one, two and three dimensional cutting stock or packing problems. More recently, Christofides and Whitlock (1977), and Beasley (1985) have developed sophisticated procedures for generating two dimensional patterns for rectangular shapes.

Although these developments provide useful insights and tools, they do not address directly the issue of loading products into fixed three dimensional vehicles based on volume with restrictions to avoid damage and to localize products within the vehicle. There is relatively little literature available on loading low density products. Roberts and Taylor (1972) developed a procedure for loading full width racks of stamped metal parts into railcars. Because the racks had a limited number of base dimensions and had to be stacked securely, they were able to reduce the packing problem down to solving two one-dimensional bin packing problems. The first step involved building the minimum number of stacks up to the height of the boxcar, and the second step was to minimize the number of railcars required by packing the stacks into railcars along the length of the car.

Haessler (1979) reported a similar application involving loading large rolls of paper into railcars. This application was in one sense simpler than the one describe above because the rolls were loaded on end and the layout issue simply involved finding the number of circles of constant diameter that would fit into a rectangle. Individual stacks were constructed by solving a one dimensional trim problem. However, Haessler's procedure was more complicated in that it used freight rates by railcar size to determine the mix of cars to be used to minimize the total cost of shipping a multiple railcar order.

Two other related papers are those by George and Robinson (1980), and Chen and Liu (1987). George and Robinson developed a heuristic procedure for manually packing boxes in a container across one section at a time. They assumed a box could be oriented with any face up and could have any number of boxes on top of it. Chen and Liu used the first-fit-decreasing (FFD) heuristic to find an initial solution to the problem of locating rectangular components on a circuit board so as to minimize the size of the board required. Because FFD can give such poor results, and because the problem investigated here is much more complicated than the one studied by Chen and Liu, we made no attempt to use FFD as a solution method for the vehicle loading problem.

Literature searches carried out independently by us and the client organization did not uncover any research that directly deals with the problem discussed in this paper.

2. I. *Background*

The company has about one hundred different packaged products which it sells from inventory. The package sizes vary significantly, with the largest size being about 10 times the volume of the smallest. The packaged product is shipped in cases (corrugated boxes) that also vary by a factor of 10 from largest to smallest. The cases are stored in the warehouse in what are called unit loads. A unit load may contain as few as 10 to over 100 cases. The unit loads were designed so that each is as close as possible to being 40 inches wide, 48 inches long, and 60 inches high. One layer of a unit load is referred to as a tier. All the tiers in a unit load need not be the same heights or have the same number of cases. Because of variation in case sizes, the actual dimensions for a unit load may differ by three to six inches from the target values. Figure 1 shows a sample unit load with the accompanying variation by dimension. The unit loads were originally designed to ship in railcars two high and two across the width of a car.

Although customers are encouraged to order in full unit loads, they may order in both case and tier quantities. Prior to the development of this system, guidelines were developed as to how many units would fit in each size vehicle. These were disseminated to the customers through the sales force and full vehicle orders were entered according to these guidelines. Unless someone ‘knew’ that a larger number of units would fit, perhaps because all the units being ordered were on the low side of the target unit size, the order was entered and shipped consistent with the established guidelines.

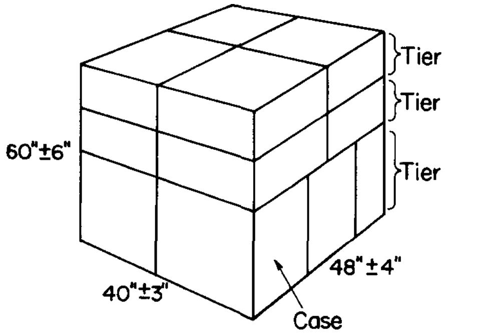

Figure 1. Illustration of a typical unit load configuration

Because a customer could order any mix of products, the guidelines had to be established in a conservative manner to insure that the order could be loaded as entered. This resulted in the cube utilization of most vehicles being lower than necessary. It should be noted that adding to the order after it was loaded when it was apparent that more product would fit, would not necessarily give as good a result as first increasing the order and then loading that entire larger order into the vehicle. The best way to handle the extra products may be to intermingle them with product on the original order, rather than load them by themselves in the gap that is left after the original order is loaded.

To summarize: The objective of this project was to develop a computer-based procedure to be run at order entry time, that will size the order and diagram the loading of that order so as to increase vehicle utilization and reduce freight cost.

The inputs to the computer system are:

- Initial customer order (including all product dimensions).
- Guidelines for changing the orders (which products should be added or dropped first).
- Dimensions of the shipment vehicle.

The outputs from the system are:

- Suggested order changes.
- A load diagram.

The guidelines for changing the order show the sequence in which product will be added to and dropped from the order. The original order can be either increased or decreased, but not both.

2.2. *Load diagram*

The easiest way to understand the loading problem is to view the top and side view of a load diagram for both railcars and trucks. A partial diagram for railcars is shown in Figure 2. In the top view, or footprint, there are two columns of stacks of product. Because no product has a base length greater than 52 inches, it is always possible to load two columns of stacks without concern for interference from side to side. Stacks 1, 4, and 6 represent one unit stacked on top of another unit. This is the most common situation in railcars. Stack 2 is also 2 units, but with a tier of some product sandwiched between to make use of the available height in the railcar. Stack 3 is a number of tiers stacked on top of a unit load. Stack 5 is handloaded cases of a single product which is inserted to fill the gap in the doorway. This is done if there is a gap large enough to fill with loose cases.

A typical trailer load diagram is shown in Figure 3. In the top view or footprint, there are several common configurations across the width of the truck. Section A shows 2 units with their long side along the width of the truck. This is possible in a 95″ wide trailer if the unit base length is less than the target value of 48″ and is generally the best utilization of space. Section B shows 4 units in a box-like arrangement that is also quite efficient in terms of space utilization. Section C is sometimes the only way to layout the units with larger base dimensions and is generally the least efficient utilization of space. In the side view, Stack 1 shows one unit on top of the other. This can occur when the unit height of some product is at the very low end of the height range. Stack 2 shows a common arrangement in which one half of a unit is placed on top of a full unit. This generally does not completely utilize the height of the vehicle, but is very efficient from a material handling standpoint. Stack 3 is the same as Stack 2 except a tier of some product has been sandwiçhed between a unit and one half unit to better utilize the available trailer height. Stacks 4 and 5 show tiers of different products on top of a full unit. This also may lead to better vehicle height utilization. Finally, Stack 6 is made up of tiers of a variety of products. This also tends to yield good vehicle height utilization. Loose cases may also be hand loaded to fill any large gaps at the rear of the truck.

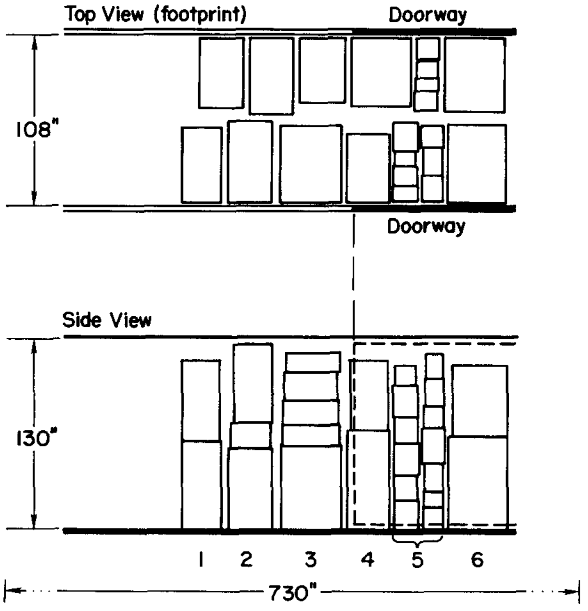

Figure 2. Section of typical railcar load diagram

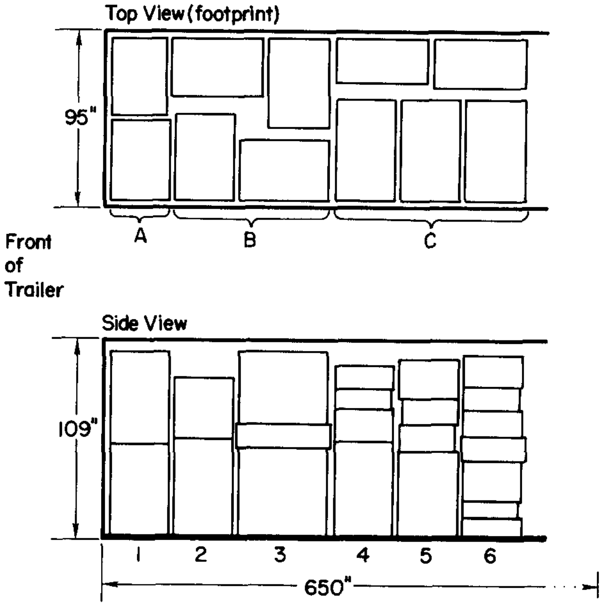

Figure 3. Section of typical trailer load diagram

3. Overall solution procedure

The solution procedure begins by estimating the number of stacks that will fit into the vehicle. The order is then adjusted by adding or deleting units so that the adjusted order requirements will generate the estimated number of stacks. The next step is to determine if a footprint can be found that locates the base dimensions of the stacks in the rectangular area of the vehicle floor. If a footprint can be found, then the number of stacks is increased by one and the process is repeated. If no footprint is found, the number of is reduced by 1 and the process repeated. The procedure stops when the footprint containing the largest number of stacks is found. A detailed description of this procedure follows:

Step 1. Estimate the number of stacks, NS.
For railcars,

$$\mathrm{NS} = \left[\frac{2 * \mathrm{VL}}{\mathrm{AUW}}\right],$$

where

VL : vehicle length,
AUW: average width of all units on order,
$[a]$: largest integer less than or equal to a.

For trailers,

$$\mathrm{NS} = \left[\frac{0.95 * \mathrm{TFA}}{\mathrm{AUBA}}\right],$$

where

TFA : trailer floor area,
AUBA: average unit base area.

Step 2. Calculate the number of equivalent units, NEU, that can be included in NS.

For railcars,

$$\mathrm{NEU} = 2 * \mathrm{NS}.$$

Tier quantity orders are grouped into unit size blocks for comparing units on order, UOO, to NEU. Maximum tier block height is defined to be one half of the vehicle height.

For trailers,

$$\mathrm{NEU} = 1.5 \times \mathrm{NS} + 0.5 * \left[\tfrac{1}{2}\mathrm{NSU}\right],$$

where NSU is the number of short units where short is defined to be less than one half of trailer height. Tier quantity orders are conceptually grouped into either full unit, half unit, or full vehicle height stacks for comparing UOO, to NEU.

Step 3. Adjust the order up or down based on the difference between UOO and NEU. If NEU is greater than UOO, units or half units are added to the order. If UOO is greater than NEU, units are deleted from the order. Adding or deleting is done based on a predetermined list that controls the sequence in which changes are made. This list is a function of the particular customer's usage and ordering patterns.

Step 4. Build NS stacks for the ordered items as adjusted in Step 3. There are two primary considerations in building stacks in addition to the constraint that the stack height not exceed the vehicle height:

- Very heavy products must be on the bottom of the stack.
- All the products in the stack should have identical or at least similar base dimensions. The ideal situation is to have only one product in each stack. If this is not possible, then the difference should be small and the larger dimension along the length of the vehicle should be on top. This will minimize damage due to shifting of product along the length of the vehicle as the vehicle accelerates and stops.

Step 5. Determine if a footprint can be found for the NS stacks generated in Step 4.

For railcars, finding a footprint reduces to solving a knapsack problem. Let SW_j be the width of stack j and $T = \Sigma_j \mathrm{SW}_j$. Clearly to be feasible,

$$T \leqslant 2 * \mathrm{VL},$$

where VL is the usable length of the vehicle. NS stacks will fit in the railcar if values of $X_j = 0, 1$, can be found such that

$$T - \mathrm{VL} \leqslant \sum_j \mathrm{SW}_j X_j \leqslant \mathrm{VL}.$$

This can be illustrated with a simple numerical example. Suppose that $\mathrm{VL} = 600$ and $\Sigma_j \mathrm{SW}_j = 1190$. Any subset of stacks with total width between 590 and 600 will be a feasible solution. If the width used on one side is at least 590, the remaining stocks will have total width not exceeding 600 and therefore will fit on the other side of the railcar.

Once again there are soft constraints that should be satisfied if possible. These are,

- Localize all of each product in some section of the vehicle to facilitate checking the order as it is loaded and unloaded.
- Minimize the height difference between adjacent stacks so that the top tiers are effectively blocked in and cannot move very much if they become loose.
- Save short, lightweight stacks for the doorway. This is the most difficult area to load because of limited maneuverability of the clamp trucks.

In order to meet these constraints, an attempt is made to solve the knapsack problem heuristically. If this attempt is unsuccessful, then a mathematical solution to the problem is found, if one exists, using a lexicographic search algorithm.

For trailers, finding a footprint is more complex. This is discussed in detail in the next two sections of the paper.

Step 6. If a footprint is found for NS stacks, increase NS by 1 and go to Step 2.

If no footprint is found, reduce NS by 1 and go to Step 2 if NS − 1 has not been previously checked. If a footprint has been found for NS − 1, go to Step 7.

Step 7. Fill in major gaps in the vehicle with hand loaded cases. Such gaps smaller than the base dimensions of a unit can occur in any load configuration.

In railcars there may be a gap on either or both sides that is significant but less than the width of a stack. This space is intentionally positioned in the doorway in the loading diagram and filled in by hand loaded cases. Alternatively, air bags may be used to fill this space.

In trucks, the gap is forced to occur at the back of the trailer and may be on one or both sides. Again, cases of a single product are hand loaded to utilize as much of that gap as possible.

3.1. Trailer footprint problem definition

The trailer footprint procedure for Step 5 is more complex. This is due to a combination of factors including greater variety of possible loading patterns, inventory handling and staging concerns, product damage concerns, axle weight limitations, multiple customers and tandem trailers. First, as with rail, at most two stacks can fit side by side across the width of a trailer. However, unlike rail, each stack can be oriented with either its long side or short side parallel to the long side of the trailer. The former is referred to as a rotated orientation. As indicated in Figure 3, this leads to a number of possible loading patterns. Inventory and staging considerations are generally more complicated with trucks than with rail, due to the fact that the loads contain a greater variety of product types, which translates into more complex stack building and material handling. In order to reduce staging complexity, the footprint procedures give high priority to positioning stacks containing identical products close together. A secondary but important benefit of this is a reduction in clerical counting errors at both shipment and receiving.

Concerns about minimizing product damage are operationalized in several ways. One already mentioned is keeping identical products close together, which has the effect of reducing loading dock congestion and consequential handling damage. Other, often conflicting, mechanisms are:

(1) to place the highest stacks in the front of the vehicle to avoid having tall stacks tip over shorter ones during sudden stops;

(2) to follow a prespecified stack orientation preference (rotated or non-rotated) which is defined prior to loading as being the only acceptable way to avoid in-transit damage to a particular stack (this could be due to the unevenness of a vertical face of a stack, or due to the stack load bearing geometry resulting from how cases are built into tiers, and thence, into unit loads);

(3) to place stacks such that the front/back vertical faces remain parallel during sudden stops (there is a need to avoid having a following stack rub against just the corner of the predecessor stack on the opposite side of the trailer);

(4) to tightly load stacks to minimize side to side movement. This is most important at the back of a trailer load, where loose products could potentially move sideways or backwards into the empty gap that may exist.

Unlike rail shipments where volume constraints are always met before weight constraints, truck shipments can exceed axle weight limits. Hence, an attempt is made to develop load patterns within vehicle axle weight limits. Unlike rail shipments, it is possible with trucks to specify the order in which product groupings for a given customer are to be dropped off, and it is possible to ship several customer orders in a given trailer. In these situations, products within a group or for a given customer must be loaded contiguously to avoid clerical errors and extra material handling at the drop-off points. Finally, truck tractors can pull either a single trailer or two trailers in tandem. The latter is more complicated than simply taking a loading pattern for a larger trailer and splitting it in two. All of the above concerns, flush faces, height, orientation, axle weight, etc., come into play for *each* trailer. In addition, if like product (or product group, or customer group) cannot totally fit in the back of the leading trailer, the carryover should go in the *back* of the tandem so as to facilitate material handling at the drop off point. (The two trailers are loaded and unloaded simultaneously.)

A very simplified 0–1 integer formulation of the single trailer footprint problem follows.

Let

W = Floor width of trailer in inches.

T = Floor length of trailer in inches ($t = 1, \dots, T$, where $t = 1$ indicates front of trailer).

w_{jm} = Footprint width in inches of stack j in orientation m ($m = 1$, 2, indicating rotated or not; $j = 1, \dots, N$, where N is a unique dummy terminal stack with zero footprint).

$$x_{jmt} = \begin{cases} 1 & \text{if the following edge of stack } j \text{ in orientation } m \text{ is positioned at location } t, \\ 0 & \text{otherwise.} \end{cases}$$

Then the problem of minimizing the usable floor space is

$$\text{min.} \quad \sum_{t=1}^{T} t x_{N1t} \tag{1}$$

$$\text{s.t.} \quad \sum_{m=1}^{2} \sum_{t=1}^{T} x_{jmt} = 1 \quad \text{for } j = 1, \dots, N, \tag{2}$$

$$\sum_{j=1}^{N} \sum_{m=1}^{2} \sum_{q=t}^{t+w_{jm}-1} w_{jm} x_{jmq} \leqslant W \quad \text{for } t = 1, \dots, T, \tag{3}$$

$$\sum_{t=1}^{T} t x_{N1t} - \sum_{t=1}^{T} t x_{jmt} \geqslant 0 \quad \text{for } j = 1, \dots, N-1, \ m = 1, 2. \tag{4}$$

The objective (1) is met positioning the following edge of the unique (dummy) terminal stack as close to the front of the vehicle as possible. Constraints (2) insure that each stack is loaded only once. Constraints (3) insure that the vehicle width is not violated, and (4) permits the objective function to force loading towards the front of the trailer. It should be pointed out that (3) is valid in this loading environment because no more than two stacks can fit side by side across the width. Constraints (3) would have to be modified if more than two stacks could fit side by side across the trailer width in order to maintain contiguity of the space available for loading a stack.

Early in this research process two 0–1 integer implicit enumeration algorithms were developed and tested for solving this problem. For the following reasons, however, these algorithms were dropped in favor of the heuristic procedures to be described. First, it was quickly discovered that optimality was impossible to verify within an acceptable period of time, and the quality of non-optimal solutions found during partial enumeration was difficult to control. As mentioned earlier, the client organization required that the loading diagram be prepared while their customer was still on the phone with their order entry person. This was translated into a contractual specification of generating the diagram within two seconds of CPU time on their order entry computer. Testing indicated that two seconds was not sufficient time for our optimization codes to be of much value, especially given all the data manipulation and stack building that had to also occur within those two seconds. We developed the second optimization code, because, as the formulation indicates, this 0–1 approach is sensitive to the metric used. When the client decided that solution accuracy had to be increased from inches to *tenths* of an inch, the size of the formulation grew significantly. Hence, in the second code, data structures and logic were developed to make the solution procedure computationally metric free. This significantly reduced solution times, but not enough to make the optimal seeking approach practical in this setting.

Other major reasons for selecting a heuristic are:

(1) The difficulty of explicitly modeling the client's notion of 'closeness' of like product (recall that each stack can contain multiple product types).

(2) The objective function and constraints are not firm.

In reality, the problem is multicriteria. For example, some of the prespecified stack orientation preferences in a load might be relaxed in order to achieve a tight back end on the trailer. Another example is that when customers do not want a full load, then it becomes important to spread the product across the trailer floor to minimize product damage, rather than minimize floor space as is implied by (1)–(4).

3.2. *Trailer footprint heuristic (Step 5 in solution procedure for trailers)*

The trailer footprint heuristic has two stages, and can involve more than one pass of *Stage* 2.

The first stage develops a list of all stacks that specifies the order in which stacks should be considered for loading. No effort is made to actually load in *Stage* 1. Stacks are ordered on this list into contiguous groups. The hierarchy of grouping from most to least inclusive is:

(1) by customer; or

(2) by drop off sequence; and,

(3) by likeness of product (two levels used).

Within likeness groups, stacks are ordered from tallest to shortest.

More formally, in *Stage* 1, consider an ordered set of stack numbers, $A = \{1, 2, 3, \dots, i, \dots, \mathrm{NS}\}$. The objective is to reorder these stack numbers and place them in set A', in groups, such that when the stacks are considered sequentially for loading from A', none of the above constraints are violated.

If unusual height variation occurs within a common product group, then the shorter stacks are split off and treated as separate groups. To deal with axle weight constraints, stacks are sequenced by alternating the heaviest and lightest stacks within the group without violating firm drop off ordering or height restrictions. The effect of this is to approximate an even weight distribution.

The Ordering Procedure used is as follows:

OP1. Sort the stacks in A according to decreasing drop off sequence number. (This can be defined across customer groups or within a customer group depending on the situation. The lower the drop off sequence number, the closer the stack will be placed to the rear of the vehicle.) Let A_k be one of these groups.

Do OP2–OP5

for each drop-off group A_k defined in OP1:

OP2. Sort A_k according to decreasing stack height.

OP3. Assign the first stack (i.e., the tallest) to position $j = 1$ of A'_k. Call this the last stack assigned, LS_k.

OP4. Scan for stacks in A'_k from first to last. Calculate a 'closeness value' between LS_k and each non-assigned stack $i \in A'_k$ such that i is no shorter than the tallest unassigned stack plus a user specified constant DIFF. (This insures that overly short stacks are not loaded in front of tall stacks.) The closeness value measures the degree to which it would be desirable to place i next to LS_k in the vehicle, and is on a scale from 0 (no similar products) to 21 (both LS_k and i comprise identical product mixes).

OP5. Select stack i with the highest closeness rating, and assign it to position $j+1$ in A'_k. Set $\mathrm{LS}_k = i$ and go to OP4. If there is no stack with acceptable height and non-zero closeness rating, assign the tallest unassigned stack from A_k to position $j+1$ in A'_k, call this stack LS_k, and go to OP4. When all stacks in A_k have been assigned to A'_k, go to OP2.

OP6. Set $A' = \cup A'_k$, for all k, maintaining stack ordering across k's. Calculate estimated weight distribution over axles by assuming $\frac{1}{2}$NS stacks are over each axle ($\frac{1}{4}$NS if a tandem vehicle). If distribution is within axle weight limits, go to *Stage* 2. If drop-off sequence is firm, go to *Stage* 2. Resequence drop-off groups within A' alternating heaviest and lightest groups, without violating the height difference parameter DIFF. For single drop-off problems, alternate heavy and light likeness groups without violating DIFF. Go to *Stage* 2.

Stage 2 sequentially 'loads' the stacks from the ordered list in A' much the same way as the stacks are physically placed in the trailer, starting at the front and filling in towards the rear. In Stage 2 actual dimensions, orientation preferences, and other product damage concerns are taken into account. Each stack is taken from the list A' and is positioned either on the right or left side of the trailer immediately behind the existing loaded left or right stack. Hence, as each stack is considered for loading, two decisions are made: stack orientation (rotate or not) and whether it should be placed on the left or right side of the trailer. These decisions are driven by a desire to obtain specific results. From the most desirable to the least, the following results are preferred:

(a) two non-rotated stacks side by side;

(b) boxing of four stacks;

(c) one non-rotated and one rotated side by side; and finally,

(d) two non rotated side by side. (See illustrations in Figure 3.) The major factors that might mitigate against reaching the ideal utilization pattern in Stage 2 include concerns of prespecified stack orientation preference, product closeness, insufficient vertical faces between adjacent stacks, a loose back end, and, of course, stack footprint dimensions. (Preprocessing in Stage One largely takes care of height and axle weight concerns, and simplifies the process of keeping like product, or customer groups, together in Stage 2.)

As each stack is loaded, an analysis is made of the current partial solution, (for example, have more stacks been loaded on one side than the other), and characteristics of the stack to be loaded are noted along with unusual characteristics of unloaded stacks. (An example of the latter might be that all remaining unloaded stacks must be rotated due to their sizes.) An effort is made to orient and position the stack such that the most likely outcome is (a), then (b), (c) or (d), respectively. As the procedure begins to fill the vehicle, increasingly more attention is paid to the tightness of the back end, and outcomes (a)–(d) are viewed as being equivalently desirable.

If the above process is successful in creating a feasible load, and that is all that is required, then the procedure goes to Step 6. If the quality of the solution is unsatisfactory, or if the objective is to maximize the number of stacks that will fit in a vehicle, then there could be three more passes through Stage 2. During the second pass, orientation preferences are relaxed for those stacks in the last half of the load that could likely be oriented in either direction without incurring in-transit damage. In pass three, orientation preferences for all such stacks are relaxed. In pass four, an attempt is made to load all stacks in a rotated orientation on one side and a non-rotated orientation on the other, without violating orientation preferences. The procedure selects the best of these four solutions, and returns it to the stack building part of the procedure (Step 6). Axle weights are calculated for the best solution found. If these exceed limits or if the total weight limit is exceeded, a warning is issued that the order is not acceptable from a loading perspective.

3.3. Coding and computational characteristics

The order entry and supporting data management programs that call the loading heuristic code were developed by the sponsoring company and are written primarily in Cobol. The loading heuristic is written in Fortran and comprises about 5000 lines of code. On an IBM 3090 computer using the IBM Fortran 77 Compiler without optimization (OPT(0)), the loading procedure typically takes from 0.15 to 0.30 seconds to solve a rail or truck problem. (The numerical example given in Section 4.0 required 0.206 seconds, which is total CPU time including input and output.) The loading code has also been run on a stand alone basis for test purposes on an IBM-XT personal computer, where it usually takes 20–40 seconds to develop a load diagram.

4. Numerical example

To further clarify the procedure, a load diagram is generated for a trailer shipment using real order data except for the fact that the product code has been disguised. All sizes are in inches. The trailer dimensions are 504 × 94 × 100 (length × width × height). The initial order requirements are shown in Table 1. All the order items are in unit quantities. Because any two units have a total height that exceeds the vehicle height, some units will have to be split to make stacks containing a unit on the bottom and a half unit on the top. The allowed split points are indicated in Table 1 under "Vertical split point", which measures in inches from the bottom of the unit load to where it can be divided. This is a single customer order, and there are no stack orientation preferences.

Step 1. Estimate the number of stacks,

$$\mathrm{NS} = \left[\frac{0.95 \times \mathrm{TFA}}{\mathrm{AUBA}}\right] = \left[\frac{0.95 \times 47376}{1999}\right] = 22.$$

Step 2. Calculate the number of units needed for 22 stacks,

$$\mathrm{NEU} = 1.5 \times 22 = 33.$$

Step 3. Adjust the order to provide the correct number of stacks. The number of units ordered is 33. Since NEU = 33, no adjusting is necessary.

Step 4. Build NS stacks. Table 2 shows the 22 stacks constructed from the 33 units required. The stack generation sequence is controlled by the following priorities.

- Limit each stack to one product.
- Avoid splitting some designated units.
- Minimize differences in width and length of products in a stack.

Table 1
Order data for numerical example

Product code	Quantity	Width	Length	Height	Weight	Vertical split point
P1	2	41.0	49.2	62.0	824	31.0
P2	1	40.5	49.9	58.0	1000	34.8
P3	2	40.8	48.6	60.6	518	30.3
P4	3	41.0	48.9	58.0	540	29.0
P5	5	41.0	48.9	58.0	763	29.0
P6	3	41.0	49.2	62.0	520	31.0
P7	5	41.2	51.6	59.2	515	29.6
P8	1	40.8	48.6	60.6	1015	30.3
P9	2	40.8	48.6	57.9	802	19.3
P10	2	40.8	48.6	60.6	858	30.3
P11	1	40.5	49.9	58.0	864	34.8
P12	2	41.5	49.5	62.4	1029	31.2
P13	1	38.7	50.7	61.4	576	30.7
P14	1	42.3	51.0	55.8	603	27.9
P15	1	40.8	48.6	60.6	750	30.3
P16	1	40.8	48.6	60.6	896	30.3

– If a stack has more than one product, satisfy all requirements in the same kind of stacks.

Step 5. Find a footprint for 22 stacks. Initially, $A = \{1, 2,\dots,22\}$. After processing OP1–OP6,

$$A' = \{3, 4, 15, 16, 17, 18, 21, 22, 19, 20, 7, 8, 1, 2, 14, 9, 10, 11, 12, 13, 5, 6\}.$$

Table 2
22 stacks generated for numerical example

Stack number	Bottom unit	Top ½ unit	Width	Length	Height
1	P7	P7	41.2	51.6	88.8
2	P7	P7	41.2	51.6	88.8
3	P6	P6	41.0	49.2	93.0
4	P6	P6	41.0	49.2	93.0
5	P4	P4	41.0	48.9	87.0
6	P4	P4	41.0	48.9	87.0
7	P5	P5	41.0	48.9	87.0
8	P5	P5	41.0	48.9	87.0
9	P14	P7	42.3	51.6	85.4
10	P2	P8	40.8	51.6	88.3
11	P11	P8	40.8	49.9	88.3
12	P9	P15	40.8	48.6	88.2
13	P9	P15	40.8	48.6	88.2
14	P7	P7	41.2	51.6	88.8
15	P12	P16	41.5	49.5	92.7
16	P12	P16	41.5	49.5	92.7
17	P1	P13	41.0	50.7	92.7
18	P1	P13	41.0	50.7	92.7
19	P5	P3	41.0	48.9	88.3
20	P5	P3	41.0	48.9	88.3
21	P10	P3	40.8	48.9	90.0
22	P12	P3	40.8	48.6	90.9

The first pass of Stage 2 obtained the solution given in Figure 4. Boxing is the best possible outcome since the stacks are all too large to be placed adjacent to each other across the vehicle width in non-rotated orientations. The solution keeps like products together, and all stacks towards the front of the vehicle. The solution con-

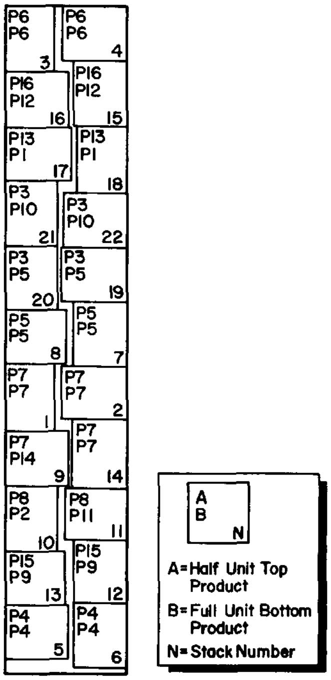

Figure 4. Load diagram for numerical example

tains no forward stacks that are shorter than following stacks by more than DIFF = 10.0 inches.

Step 6. Increase NS. The number of stacks to be tried is increased to 23. One and one half more units are added to make a total of 34.5, and 23 stacks are created. No feasible solution could be found. Therefore, the procedure terminates with the solution of 22 stacks shown in Figure 4.

Step 7. Fill gaps.

In this example, the space utilization of the 22 stacks is quite good and there are no open spaces large enough at the back of the truck to be filled with loose cases.

5. Conclusion

Industrial packing problems are complex, and have numerous soft constraints. As such, they do not have concise mathematical formulations and corresponding optimization solution procedures. Heuristic procedures, on the other hand, can readily incorporate preferences and soft constraints as the search is carried out and selections made. Heuristics are also generally much faster than optimization algorithms. This is extremely important in this application because of the large number of orders processed each day and the requirements for immediate feedback during the order entry process.

The methods presented in this paper have been used by the company sponsoring this research in loading railcars since late 1987, and parallel testing of the truck portion has been ongoing since early 1988, both with very favorable results. The major benefits recognized are significantly higher vehicle utilizations, and improved customer service. Included in the latter are all the benefits associated with increasing the certainty that orders will arrive as originally specified at order entry time. In addition, the methods presented are able to be used at loading time to quickly and accurately adjust to last minute changes, such as having to replace a vehicle with a different size trailer, or to change load patterns because of customer dictated needs. Finally, these procedures have been used outside their intended operational setting to perform a detailed feasibility study of single versus tandem trailers for another manufacturing/distribution region for the company sponsoring this research. Without these procedures, it would have been virtually impossible for this organization to accurately carry out such a study in a timely and economical manner.

Computerizing the process of preparing loading diagrams is a major hurdle that has been overcome as this company continues its drive towards automating all material handling.

Acknowledgements

The authors appreciate the constructive comments made on an earlier draft of this paper by the editor and two anonymous referees.

References

Beasley, J.E. (1985), "An exact two-dimensional non-guillotine cutting tree search procedure", *Operations Research* 33, 49–64.

Chen, C., and Liu, M. (1987), "An application of bin-packing to building block placement", *Proceedings of the 1987 IEEE International Symposium on Circuits and Systems*, Philadelphia, Vol. 2, 576–579.

Christofides, N., and Whitlock, C. (1977), "An algorithm for two dimensional cutting problems", *Operations Research* 25, 30–35.

George, J.A., and Robinson, D.F. (1980), "A heuristic for packing boxes into a container", *Computers and Operations Research* 7, 147–156.

Gilmore, P.C., and Gomory, R.E. (1963), "Linear programming approach in the cutting-stock problem—Part II", *Operations Research* 11, 863–888.

Gilmore, P.C., and Gomory, R.E. (1963), "Linear programming approach in the cutting-stock problem—Part II", *Operations Research* 11, 863–888.

Gilmore, P.C., and Gomory, R.E. (1965), "Multistage cutting stock problems of two or more dimensions", *Operations Research* 13, 64–120.

Haessler, R.W. (1979), "Solving a boxcar loading problem", presented at the joint ORSA/TIMS Conference, May.

Roberts, D., and Taylor, D. (1972), "A distribution and railcar loading system", Paper presented at the Joint ORSA/TIMS Conference, November.