# HEURISTICS
# FOR
# GENERAL 3D PALLETIZATION PROBLEMS

Mourad El-Masry

A Thesis

in

The Department

of

Mechanical Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

May 1995

ISBN 0-612-05136-6

**ABSTRACT**

**HEURISTICS**
**FOR**
**GENERAL 3D PALLETIZATION PROBLEMS**

Mourad El-Masry

Interlocked stacking is a new procedure by which boxes of various dimensions are loaded on pallets. The focus of this work is on developing heuristics to solve general palletization problems. Two heuristic models were developed in this thesis. The first model is designed to comply with the palletization problem where boxes of different dimensions arrive to the pallet loading area in random order. The objectives of the heuristic are maximizing the volumetric pallet utilization (VPU), minimizing the work-in-process (WIP) as well as the robotic palletization time (RPT). New performance indices are established; Total Number of Sub Volumes, TNSV, Partitioned Remaining Volume Load Capacity, PRVLC, and Total Zero Count, TZC. Modifications were made on the first heuristic to develop a second heuristic that incorporates pallet load stability and box demand requirements. The two heuristics were applied using Turbo C++ programs to simulate the palletization process. Three simulations were conducted to test each of the heuristics. The results of the simulations were compared with those found in the literature. Interlocked stacking is also compared with the two conventional pallet loading techniques; layered and column stacked palletization used in previous studies.

Key Words: Palletization, Heuristics, Pallet Loading.

**ACKNOWLEDGEMENTS**

I would like to thank Dr. G. Abdou for his professional supervision, guidance, patience, and generosity. It was a pleasure working with him especially in the last few month where the real action took place. I would also like to thank Dr. S. V. Amiouny for his co-supervision, understanding, and valuable advice regarding the thesis. I certainly can not for get to thank Mr. M. Hanna for his help and expertise. I would also want to show my appreciation to my parents, to all the "Falcons", Dr. M. Salib, to Mark and Matthew, and to all those who remembered me in their prayers without which I would not have made it. Above all, I thank God First and Last for being with me from the beginning to the end which is yet to come.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

Page

FIGURE

## NOMENCLATURE

| | |
|---|---|
| **$Box_{C1}$** | The box at the pick up location at the end of the conveyor. |
| **$Box_{C2}$** | The box following $box_{C1}$ in the random sequence. |
| **$Box_{WIP}$** | Best Box in the Work-In-Process area. |
| **$h_{BTi}$** | Height of box type i. |
| **$H_{SVj}$** | Height of sub volume j. |
| **ISVLC** | Individual Box Type Sub Volume Load Capacity. |
| **$l_{BTi}$** | Length of box type i. |
| **$L_{SVj}$** | Length of sub volume j. |
| **MCBT** | Maximum Capacity of Box Type. |
| **N** | The number of sub volumes that can fit a particular box, $(N < M)$. |
| **NBOP** | Number of Boxes On the Pallet. |
| **NF** | Negative Flexibility. |
| **P** | Number of additional sub volume added after a box loading process. |
| **PF** | Positive Flexibility |
| **PRVLC** | Partitioned Remaining Volume Load Capacity. |
| **RPT** | Robotic Palletization Time. |
| **TNBT** | Total Number of Box Types. |
| **TNSV, M** | Total Number of Sub Volume. |
| **TZC** | Total Zero Count. |
| **VPU** | Volumetric Pallet Utilization. |
| **$w_{BTi}$** | Width of box type i. |
| **WIP** | Work-In-Process. |
| **$W_{SVj}$** | Width of sub volume j. |
| **ZCBT** | Zero Count of Box Type |

*To My Parents.*

*To The Abdou Family.*

*To St-Mary's Youth.*

*and*

*To All*

# CHAPTER 1.0 INTRODUCTION

Palletization is the final process in which items are either manually loaded on a pallet or through using material handling equipment. These items can vary in shape and size. They can be of rectangular shape such as boxes, corrugated board containers carrying any type of goods, and they can be irregularly shaped packages or products. In this research, only rectangularly shaped packages (referred to as boxes) are dealt with. The use of pallets to handle large quantities of properly stacked boxes has been recognized as one of the most important ideas in material handling (C. S. Chen *et al.*, 1991). Pallets are specially used for shipping purposes in trucks, railcars, and air crafts, and for storing purposes in warehouses and distribution centres. Traditionally, palletization was performed manually. That is a worker would lift the boxes one by one and put them somewhere on the pallet according to his or her judgement. When identical boxes were to be loaded on the pallet, the only inconvenience was the physical tiredness of the worker. In situations where boxes with different sizes are to be loaded on the same pallet, some concerns are given more emphasis: how will the worker decide which box and where to load the box so as the space of the pallet is well used? Is the total palletization time in an acceptable range? How to guarantee a stable final pallet configuration? Through practice, the human worker can master the process but as the number of box types increases or simply as shipment orders change, the palletization process performed manually becomes very confusing and its efficiency decreases. As technology advanced and industrial automation was introduced to the palletization activity, many industries have adopted a fixed automation type of palletizer: the pallet size and box dimensions are fixed leading to a fixed pallet pattern. That is the simplest and fastest form of automated palletization. It is well suited for mass production industries. The main disadvantage of fixed automation such as palletizers is its inflexibility to serve the process if modified. When the box sizes are variable, a flexible system possibly involving a robot is needed. Three-Dimensional (3D) palletization can define the situation where the boxes may have different lengths, widths, as well as heights. In such circumstances, robotic palletization is well suited for the task and is characterized by a medium speed of pallet loading and a high flexibility with respect to box types. Palletization robots are usually computer controlled by sequence programs. Normally, pattern identification systems such as cameras, laser beams, or bar coding systems, are required. As an important part of the manufacturing process and material handling, robotic palletization is attracting more and more interest from both industries and research institutions. Palletization algorithms are developed to satisfy the different industrial needs to use robots to perform the palletization task.

There are at least two major problems that can be identified as "pallet loading problems." The first problem could be called "The manufacturer's pallet loading problem" (Hodgson, 82). In this problem, the manufacturer produces a quantity of the same product which is packaged in identical boxes: the boxes may be packed in identical cartons; and the cartons are packed on identical pallets for shipping to buyers and distribution centres. The concern is to choose the package, carton, and pallet dimensions so that the volume of product packed on a pallet is maximized. To find a solution for most manufacturer's pallet loading problems, a one-time analysis is required. Such solution is called an off-line solution since it need not be changed during its application. Fixed automation palletizers are most suitable to implement the solution.

The second and more complex pallet loading problem, which is the focus of this research, could be called "The distributor's pallet loading problem" (Hodgson 82). In this problem, a customer may order a shipment of products packaged in boxes of varying dimensions. There may be some customer demands and/or conditions regarding the order. Such customer requests must be considered at the time of application of the palletization algorithm. That is input data from the user to the algorithm are needed to guide the solution of the problem so as to meet customer requirements. Some user interface query may consist of answers to questions regarding the following;

+ Box dimensions
+ Type of product inside each box to determine the possibility of box rotation
+ Stackability of the boxes to avoid any risk of product damage. In stacking boxes, it must be ensured that boxes in lower locations on the pallet are capable of withstanding the load due to boxes above them. This is termed '*stackability*' and requires knowledge of the box weights and box characteristics.
+ Sequence of box arrival to the pallet loading workcell
+ Loading/Unloading priorities
+ Overall pallet weight restrictions

When developing the palletization algorithm, it is very important to consider the above mentioned issues. This will help in modelling and fully describing the problem and at the same time not restricting it to limited applications. There are some common palletization objectives that generally need to be considered. Not only are they the concern of the clients but also are performance measures by which the overall effectiveness of the palletization models is evaluated. Some of these objectives that need to be fulfilled are:

maximization of the pallet volume utilization

reduction in the pallet loading and unloading times

facility in the loading and unloading processes

loading and unloading priorities with respect to customer's demand

maximization of the loading stability of the boxes on the pallet

reduction in the space used for intermediate pallet loading/unloading steps

As palletization models have objectives like those mentioned above, there are also some practical constraints that need to be considered when modelling the palletization system. These practical constraints are as much related to the user interface query list as they are related to the objective functions. Examples of such constraints are the following:

pallet constraints

newly created empty space (sub-volume) constraints

box availability constraints

box rotation constraints

stability constraints

work-in-process constraints

For the manufacturer's problem, present technology supports the loading of pallets using automated material handling systems. However, the distributor's problem, by its non repetitive nature, is more difficult. In order to automate the physical loading of a distributor's pallet, real-time algorithms are needed. The different variations of the distributor's problem give rise to different approaches to tackle the problem. Researchers have attempted to solve the distributor's problem through two main approaches: mathematical models and heuristic models.

The mathematical model is an approximate representation of a concept, an object, a system, or a process in mathematical terms (Wan 89). The model behaves, in some sense, like what it represents. The representation in the model is symbolic, in mathematical terms including variables, parameters, and relationships such as equalities and inequalities. Through mathematical tools, the model is formulated, studied, and solutions are developed. The mathematical results obtained from the analysis are then interpreted in an understandable form (Jacoby *et al.* 80). Because mathematical models are closely associated with analytical and mathematical approaches, they are constrained to abide with the rules of these approaches. This sometimes limits the use of mathematical models and prevents their utilization to serve some industrial needs. When dealing with the pallet loading problem, the mathematical approach is well suitable in situations where the dimensions of the pallet and of all box types are known and pre-specified, the availability of boxes of all types is certain. Under this category, falls the manufacturer's pallet loading problem as well as some of the many situations of the distributor's problem found in warehousing such as optimizing space utilization for order picking. Linear Programming (LP) is a common tool used to model and solve such problems.

As for heuristics, they are solution procedures that use different criteria, methods, or principles for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal (Pearl 84). However, this does prevent mathematics from playing their role in describing and building the models. Heuristics represent compromises between two requirements: the need to make such criteria simple and, at the same time, the desire to see them discern correctly between good and bad choices. It is the nature of good heuristics both that they provide a simple means of indicating which course of action is preferred, and that they are not necessarily guaranteed to identify the most effective course of action, but do so sufficiently often. Most complex problems require the evaluation of an immense number of possibilities to determine an exact solution. The time required to find an exact solution is often extremely enormous. Heuristics play an effective role in such problems by indicating a way to reduce the number of evaluations and to obtain near-optimum solutions within reasonable time constraints. With regard to using the heuristic approach to solve pallet loading problems, heuristics can find solutions to problems where little information is given or when variables are time dependent. For instance, when the availability of different box sizes is unknown and the boxes are coming on a conveyor line in a random order, they need to be loaded on a pallet in the most optimal way. In order to solve for such a solution, there is a need for on-line decision making. In other words, heuristics are powerful in making decisions for the future based on the past and the present.

As part of algorithm development, the way boxes are loaded on pallets plays a very important role in satisfying the different palletization objectives. There are three techniques by which boxes can be loaded on a pallet: the layer by layer method which produces a layered pallet load, the column stacking method which results in a column stacked pallet load, and thirdly, the interlocked stacking pallet load method. The layered pallet load is a pattern in which boxes are loaded a level (layer) at a time, each level made up of a particular box height (figure 1.1). This way there is a box selection process for each level and the loading algorithm is applied for each level. This process can be continued until the desired pallet height is achieved. The layering process can be viewed as a *two (2) directional* pallet loading technique since boxes are loaded adjacent to each other in the two directions along the horizontal axes.

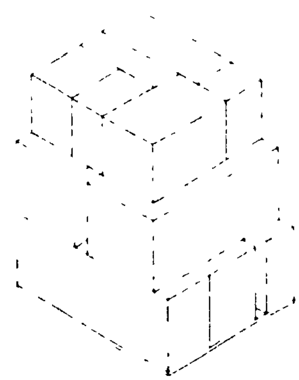


**Figure 1.1: Layered Pallet Load**

Column stacking which is the second loading approach loads the pallet with columns of boxes, each column is composed of a set of boxes stacked vertically and the column stacks are no higher than the maximum allowable pallet height (figure 1.2). Since column stacking builds the pallet loads only in the direction along the vertical axis, it is considered as a *one (1) directional* pallet loading technique. However, the selection of columns to load on the pallet is two (2) directional along the horizontal axes.

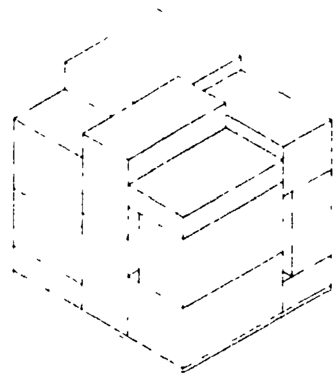


**Figure 1.2: Column Stacked Pallet Load**

The third style of pallet loading which is the interlocked stacking allows any box of any size to be loaded anywhere on the pallet - adjacent to or on top of other boxes, there is no particular box pattern on the pallet - (figure 1.3). Interlocked stacking is therefore a *three (3) directional* pallet loading technique because it builds the pallet load along the direction of the three cartesian axes. This last approach which will be used in this research, can better achieve the inter-locking of the boxes on the pallet since they are not restricted to layers or stacks. Thus by breaking the guillotine cuts among boxes, the pallet loading stability is increased.

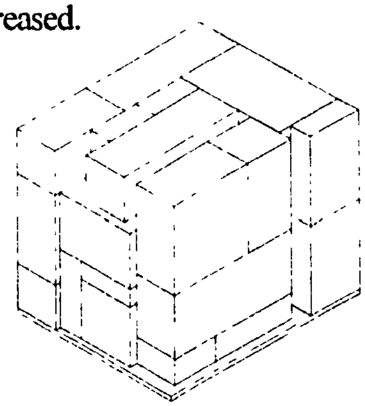


**Figure 1.3: Interlocking Stacked Pallet Load**

## 1.1 Literature Review

Interest in palletization research goes back to as early as the end of the 60's. Until that time, as simple as the palletization problem seemed to be, as much attention was given to it. But as quality standards increased, the palletization problem required extra attention and this can be observed in the many research studies that have tackled the problem in the last two decades. While they were mainly interested in maximizing the area covered by boxes on the pallet, the current trend is to focus on three-dimensional problems that are concerned with the volume of the pallet rather than its area coverage.

The present industrial needs demand such high standards due to the larger number of complexities related to three-dimensional palletization problem, that this research study focuses only on 3D problems as opposed to palletization problems with lower dimensions dealt with in the past. Thus only 3D palletization and related works are reviewed in the literature so as to continue the progress in developing new approaches serving today's needs.

Looking at the existing three dimensional palletization problems that were tackled in the literature, one can identify two distinct directions. One direction is to develop mathematical models to find solutions that can achieve maximum pallet volumetric utilization. The second direction is to develop heuristic model and solving for a problem that is more practical and more related to actual industrial needs especially when considering the random sequence of arrival of boxes to the system, or trying to satisfy different performance measures which so far were never addressed through mathematical models.

### 1.1.1 Literature on Mathematical Models

The main idea that inspired the development of the mathematical models is the following problem definition: given a list or order of products individually packaged in a variety of quantities, sizes, and weights, it is required to pack these packages on the minimum number of pallets of given uniform size.

As a leader in the area of palletization research, Hodgson's contribution (1982) was introducing two ways that boxes can be loaded on pallets. One way is to pack boxes a layer at a time, each layer being made up of boxes of common height. This way there is a box selection process for each layer and as one layer is completed, another one having a different height starts. This process is continued until the desired pallet height is reached. This is basically what was previously defined as the layered approach and was the most common approach used in palletization. Hodgson's second approach uses the stacking approach; that is, make up column stacks of boxes such that the stacks are no higher than the maximum allowable pallet height. Then the stacks can be used as input to a less complicated loading algorithm which is applied only once to optimize the base of the pallet.

Based on the Hodgson's first approach that is the layer-by-layer pallet loading, Yang (1993) developed an Integer Linear Programming (ILP) model to solve the palletization problem characterized by boxes of different base dimensions and different categories of heights. The problem solved has the following properties: the volumetric pallet dimensions as well as those of the boxes types and their quantity demand restriction are pre-defined. The objective of Yang's algorithm is to achieve maximum volumetric pallet utilization. The model uses a combinatorial approach to find solutions to the problem. Only the solutions with highest pallet utilization are candidates for optimality. At times when there are several possible layouts by which the boxes could be loaded on the pallet, a second step to Yang's solution is to consider the loading stability of the different layouts and choose the best layout with highest loading stability.

Arghavani (1993) developed a mathematical model to solve the 3D pallet utilization problem also through ILP and is considered the first to actually use the column stacking technique introduced by Hodgson in pallet loading. Arghavani's model can be seen as the implementation of Hodgson's second approach however in reverse order. That is to first develop a ILP procedure to 2D surface partitioning (2D cutting stock problem), resulting in some optimal subareas (small partitioned areas) on the pallet, then another ILP problem is formulated to determine the pallet optimum multi-layer stacking heights for each optimal subarea and thus, the stacking height can be achieved in all the columns. The results of the model consist of some optimal decision variables that determine the types and quantities of boxes whose combination can provide a maximum pallet utilization. Load stability was completely left out in Arghavani's model though it should have been considered to improve the column stacks stability since it is usually much less than that of layered loads.

Tsai *et al.* (1993) addressed the same pallet utilization problem and developed a 3D algorithm using a mixed 0-1 Integer Programming (IP) model which generates an exact optimal solution. As is the case of Yang and Arghavani, the solution of the mixed 0-1 model explicitly defines the desired number of boxes of each size but went further to define the x, y, z coordinates of each box's placement on the pallet. In terms of loading methodology of the boxes on the pallet, no specific method was reported and again, the model does not address the issue of loading stability. Though, the formulated mixed 0-1 model provides exact solutions to the pallet packing problem. However, the model requires extremely long computation times to reach the optimal solutions and the computation time tend to increase significantly as the number of boxes and box types increase. And quoting the authors' view of the model's practicality, "The computational time requirements of the developed model prevent its use in real-time palletization applications" (Tsai *et al.*, 1993).

Mathematical models may find solutions to a variety of problems and those solutions may well be successfully implemented. However, in some other cases, these methods are found impractical. As an example of very efficient models mathematically but not efficient if implemented, one might think of a mathematical model with objective function achieving 100% volumetric pallet utilization. Though, the objective function may be maximized, but it is found that if physically applied to the actual problem, there will be a need for an unrealistic box holding area or that the palletization time is increased to unacceptable values. Moreover, loading priority, loading stability, and randomization in box sequence, among other costumer needs and performance measures, are some of the issues current mathematical models are unable to satisfy. As noticed in the mathematical models in the literature, most of them do not report the practical applicability of the methods considered.

### 1.1.2 Literature on Heuristics

Heuristics are generally developed to satisfy special needs of manufacturers and warehousing industries. To state simply that the problem considered is that of pallet loading, is not sufficient to define the problem. This general description fits many situations with quite different characteristics. In fact, there exists a wide variety of different problems, each being constrained by its own problem definition, input requirements, and variations in the packing approaches adopted. A problem definition can vary from loading identical boxes on a pallet to loading a mixed combination of box types on the same pallet. Input requirement can vary from having all packaged goods lying on warehouse shelves to having boxes follow a path on conveyor belts in pre-defined or completely random order. As for the different packing approaches, boxes may sometimes require special attention such as being stacked with a certain face uppermost. In other cases certain goods are not allowed to be stacked in proximity of other goods and sometimes packages may contain fragile contents. Moreover, pallet fragility and material handling aspects are further considerations which play a dominant role. The list could be easily expanded, but these few examples are perhaps sufficient to illustrate that pallet loading can involve quit different - and often conflicting - objectives and constraints of which some may be very difficult to define in precise terms.

Han *et al.* (1989) proposed a heuristic to load packaged goods into vehicles or cargo containers. A dynamic programming approach was used to solve the 3D cargo-loading problem. A heuristic was designed to solve for different problems but the restriction in the algorithm is that the packages must be fixed in both size and shape in a particular problem. The loading procedure is based on the layer by layer style with no constraints with regard to rotation about any of the three coordinate axes of the boxes.

Haessler *et al.* (1990) came up with a computer-based heuristic procedure for sizing customer orders and developing 3D load diagrams for rail and truck shipment of low density products. The products are shipped from inventory in corrugated containers of various sizes depending on the product package sizes and customer requirements. The vehicles used also vary in size depending upon need and availability. In most cases, product volume and material handling considerations limit the amount of product loaded in vehicles before weight restrictions are met. Consequently, the heuristic is designed so as to accommodate low density products. The loading heuristic develops an actual load plan that specifies what and how boxes should be loaded in the vehicles. The objective of the model is to maximize the size of the order while at the same time avoid the risk of product damage, limit the time required to load the vehicle, and keep each product in the same 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. This last consideration would be relatively obsolete if the material handling were to be executed by mechanical equipment or robots. It is worth mentioning that since the customer could order any mix of products, it is very difficult to know at order entry time if any order will completely utilize the cubic volume of the shipment vehicle. Therefore, an extra input to the heuristic is information regarding how the order can be adjusted up and down if needed. Thus maximizing the volumetric utilization of the vehicle while still meeting customer product needs. Communication between the customer and the material handling and shipping departments plays a valuable role in optimizing the overall process and should not be ignored.

Gehring *et al.* (1990) designed a computer-based heuristic to pack rectangular boxes of different sizes in a shipping container of known dimensions. The problem is to determine positions for placing the boxes in the container so as to minimize the wasted space (which is sometimes inevitable). For this problem which can be considered a 3D cutting stock problem, various sub optimal solutions are generated using the proposed computer-based heuristic. The proposed solution procedure is based on sorting the boxes such that the boxes are entered in a linear list with decreasing volume of the elements from the beginning to the end of the list. The idea behind starting packing with high volume boxes is that it tends to result in good container volume utilization. Regarding the manner by which the boxes are stacked in the container, the boxes are stacked in vertical layers and no box is permitted to straddle neighbouring layers. The solution given by the heuristic enables the user to generate different loading patterns for a given problem. The user may then select the most appropriate pattern.

Lee (1990) briefly introduced the 3D palletization problem. He highlighted some ideas about how to solve such problem. Lee proposed to use the layer by layer loading approach and that once a box is loaded onto a new pallet layer, this layer must maintain the height of that box for the rest of the process. The way to achieve that was to restrict the particular layer to boxes of the considered height. This procedure may be efficient if all boxes are present and known. However, if boxes randomly arrive at the pallet loading station, this may lead to a very high number of boxes in the WIP which will automatically increase the palletization time. Based on Loschau's stability criterion model (1989), Lee discussed some box selection rules for the loading of the pallets but did not pursue on applying the rules to actual problems. Hence no results are reported.

Dowsland (1991) examined some of the solution approaches and strategies which may be relied on in the development/improvement of heuristics that provide packing which is volumetrically (3D) efficient. He discussed five strategies that might be adopted in the search for improvements of an already existing heuristic that is successfully developed, implemented, and tested rather than building a new heuristic from scratch. He emphasized the importance of an interactive approach to algorithm improvement which incorporates a two way dialogue between the algorithm developer and the end user while allowing both to make full use of the ideas of other researches in the area and the considerable number-crunching power available on today's technology.

Following Lee's work (1990), Yang (1993) tackled the problem with the same layer by layer approach. However, while restricting the considered layer to a specific box height, Yang proposed to define blocks consisting of at least one box such that the block height is equal to height of the layer being filled. Yang proposed two models to solve the problem. Both are characterized by the fact that boxes with similar heights are grouped together. The first model is as much mathematical as heuristic and is characterized by the pre-knowledge of the availability and quantity of each type of boxes. Furthermore, because the model is mathematically oriented, it was able to determine all possible layer combinations and pallet patterns and finally choosing the pattern with the best pallet utilization and relative loading stability according to the results obtained from a finite element software that simulated stresses, friction forces, and external forces to the different pallet loads. The second heuristic model added the random arrival of the boxes into the system. However load stability was not considered because utilization was the objective of the proposed model.

Arghavani (1993) found that most palletization problems found in the literature loaded pallets layer by layer. Therefore, he proposed to solve the problem with random arrival of boxes using the stacked column method to build the pallet. As in the case of the Yang's second heuristic model (1993), Arghavani did not consider any loading stability in his model. The optimal layout on the pallet (2D cutting stock) with respect to the boxes base area is first determined using LINDO. Then each optimal layout on the pallet is stacked up providing layers with different heights in each column stack. It seemed that

Arghavani followed Dowsland's advice (1991) regarding improving already existing research work and using tools found in the literature rather than building from scratch. In fact, though it is difficult when it comes to modifying computer programs, Arghavani was successful in using Yang's C program to solve for stacked instead of layered palletization. Again, the concept of building blocks is used and then each block is treated as a box and loaded as a whole. Though the model was restricted to loading boxes or blocks on a new column stack only after the previous stack is finished, its results showed better volumetric utilization of the pallet and more efficient palletization time when compared to those of Yang's layered palletization (1993) using the same box types and same random sequences. However, the load stacking stability is generally much lower than that of layered palletization.

### 1.1.3 Literature on Loading Pattern Stability

Very few studies have researched the loading stability of boxes. This may be caused by the difficulty to measure the stability of a load configuration. However, indicators of instability can be identified. It must be mentioned that the stability of a load depends on the characteristics of the individual boxes, in addition to the geometric composition of the pallet load. Most research primarily addresses the geometric composition issue only. There are two streams of research that studied the issue of pallet load stability as there were two styles by which boxes are loaded on pallets, namely layer and stack palletization. Furthermore, some researchers have focused on the static stability of pallet loads while others have studied the dynamic stability of pallet loads when the loads are in motion.

Carpenter *et al.* (1985) focused on the static stability of layered pallet loads. They identify some measures of load instability. One measure, which could be termed '*stack interlock*' or '*supportive criterion*' , suggests that boxes in layers other than that the lowest layer must be in contact with at least two boxes in the layer below; a lower limit of percentage contact area for each box is also specified. This measure, if satisfied, prevents the formation of columns that would tend to separate from the rest of the load during transportation. The second measure, which could be called '*support or base contact criterion*', sets a lower limit on the percentage area of a box in contact with boxes in the layer immediately below it. This measure, if satisfied, will help eliminate the situation where a box is not supported over most of its base. This would result in its failing to meet the stationary stability requirement and might also lead to potential box-crushing problems. In addition, considering one layer of a unit load, a straight line formed by box edges, other than on the periphery of the load, may result in a portion of the layer tending to separate during transportation. In their third measure, Carpenter *et al.* (1985) consider the problems associated with guillotine or near-guillotine sections cutting the pallet stack in vertical direction and propose that an upper limit be placed on the lengths of such straight edges as percentage of the length or width of the pallet. This measure could be called '*layer interlock*' or '*non-guillotine criterion*'. Carpenter and Dowsland emphasize that the degree of importance placed on each of the three criteria will depend on individual or application requirements.

The second direction in which research has studied pallet load stability is that of column stacks. Loschau (1989) investigated this issue through developing different methods that produced different dynamic stability indices. He defined a "rigid block" any box that is loaded on the base of the pallet. In the case where a box is loaded on top of other boxes, he modeled it as a "flexible bar" model since these type of boxes vibrate during transportation. The stability criterion for the rigid block model is the critical inclination angle which denotes the minimum angle of a box that causes it to slide off the pallet due to vibration during transportation. As for the flexible bar model, the stability criterion is the deviation from the centre of gravity. As the deviation from the centre of gravity decreases, so does the possibility of a box toppling off the stack.

Hanna (1993) using finite elements tested the static stability of several layouts of boxes. He considered stability as an index which measures the ability of the boxes to maintain their position when static forces are applied on them. This ability of a box maintaining its position is determined by how big is the minimum force needed to make the box unstable. Hanna defines two situations which are considered unstable for a box: sliding and toppling. In the finite elements simulation, Hanna defined the stability factor as the minimum force needed to either topple or slide a box and studied the factors that make a box topple and those that make it slide.

Concerning what makes a box slide, it is the external force that counteracts the friction force generated at the interface of the box and the surface underneath. The surface underneath could possibly be that of the pallet or some other boxes. The friction force is a function of the friction coefficient of the interface and the mass of the box itself or the total mass composed of the mass of the box and all the masses on top of the box. Therefore, the heavier the box load, (the weight of the box + the weight of the boxes on top of it), the larger the friction force, and the larger the force needed to make the box slide. This reflects on the greater sliding stability that the box will have.

In his study to identify the factors that influence the box' stability against toppling, Hanna observed that the only box properties that affect its stability are its density, width, and length. It is surprising to find that the height of the box does not contribute to the box stability, (assuming a uniform box density) as it might have been expected. The reason is that: on one hand, as the height of the box increases, the centre of gravity of the box moves upward. Thus the stability of the box is reduced. On the other hand, the increase in box height leads to an increase in box volume, resulting in a heavier box and a greater stability. When the two phenomena are combined together, their opposite effects cancel one another so that the influence of the box height on its stability is eliminated. It is important to keep in mind that this finding is valid only for not too large h/w ratios.

Among the most interesting layouts considered by Hanna was the case of a stack composed of two layers. The first layout consists of two small boxes on top of a large box. The second layout consists of one large box on top of two small boxes. The layouts were tested against different forces exerted on the top corner of the boxes to simulate the worst case for toppling. The layout with the smallest displacement would be the one with better stacking stability. The results of the simulation showed that the layout with the two small boxes on top of a large box displayed a larger displacement, and thus is the worst case of the two. Generally, if the differences among box dimensions are not too extreme, the smaller the number of boxes a layer has, the higher the stability will be. Thus, putting the stack layer with higher stability on top of the one with lower stability will achieve better stability for the overall stack.

Amiouny (1993) studied the dynamic stability of stacked pallet loads subject to acceleration whenever the transporting device (such as a forklift) changes speed or goes through a curve. Accordingly, he defines the stability of a column of boxes as the maximum acceleration that it can withstand without "failing". As Hanna (1993), Amiouny also defines the two ways in which a column of boxes can fail: by slipping of the boxes on their surfaces of contact or by tipping (toppling in Hanna's case) of the boxes on their ridges. Amiouny proposed a simple heuristic rule for stacking boxes of equal width - the base dimension in the direction of motion - in a column: sort the boxes in non-increasing order of their mass to height ratio, and stack them from the bottom up. This heuristic is referred to as Sort By Density. On average, Sort By Density outperforms the standard method of placing the heavier boxes on the bottom.

In another analysis for two boxes with equal mass to height ratios but different widths, Amiouny showed that the more stable configuration is the one in which the box with the larger width is at the bottom. This reveals that the width and the mass to height ratio are significant parameters in the dynamic stability of column stacks.

### 1.1.4 Limitations of Palletization Algorithms in the Literature

All the above studies have approached the 3D packing problem, especially the pallet loading problem through various objectives and constraints and obtained interesting results. However, they still have not covered all aspects of the problem. And with the rapidly changing demands of the manufacturing, production and other industries that need palletization, researchers must continue finding solutions to such demands and problems. The lacunae revealed as result of the literature review are as follows:

+ Most of the studies tried to solve the 3D problem through 2D approaches.

+ The pallet loading process was restricted to either layer by layer or stacking columns.

+ In layered type models, upper layers can not be loaded with boxes unless the lower layer is completely filled.

+ Similarly, models using the stacking columns approach considered sequential build-up. That is no simultaneous column stack built-up was considered.

+ Most have considered the pallet volumetric utilization as the only objective of their models.

+ Very few have given attention to the loading stability issue.

+ None have designed for boxes with dimensions that are non integral multiples of one another.

## 1.2 Objectives of the Research

The palletization problem is a research problem too wide to be solved through a general purpose algorithm because it involves many different situations from customer needs and requirements to physical constraints. Research is usually conducted to solve a sub palletization problem or in other words a constrained palletization problem. As the literature review reveals, the previous studies and researches have focused on deterministic mathematical models dealing with the 3D problem with pre-determined box quantity and availability. Furthermore very few have considered the pallet loading stability, and again it was not given enough attention though it is one of the most important performance criteria to affect the quality of the palletization process. Moreover, most of the previous work has either used the layered-pallet loading approach or the stacked-pallet .

This research takes into account the above mentioned limitations of previous work and attempts to a step further in developing heuristic solutions to the 3D palletization problems. The research analyses the palletization problem with a three-dimensional approach. The research develops interlocked stacking as a new pallet loading technique when dealing with boxes of different lengths, widths, and heights having a stochastic arrival pattern. There is no previous knowledge of the box availability or quantity and the interlocked stacking palletization approach is used. The heuristics also break the box dimensions barrier which haunted many in the past. Boxes with dimensions which are not harmonious among each others and which have slim chances of achieving good solutions are used to test the proposed heuristics. The goal of the proposed heuristics is to meet the most possible of the following performance criteria: maximization of the volumetric pallet utilization, minimization of the palletization time and the work-in-process area, as well as improving the pallet loading stability.

There are four main objectives of this research:

1) Develop Model 1 entitled: *"3D Loading Heuristic for Different Box Types."*

2) Develop Model 2 entitled: *"3D Loading Heuristic with Demand Requirements and Stability Measures."*

3) Implement and test both models using computer programming.

4) Compare the results of the proposed models with those found in the literature.

# CHAPTER 2.0 PERFORMANCE CRITERIA

It is of great importance to develop a set of criteria meeting the requirements of the practical situation of the palletization problem and measure the efficiency of the palletization task. Traditionally, the pallet loading problem was tackled by attempting to maximize the volume of boxes that can be fitted on a single pallet or in other words, pallet volumetric utilization was the one criterion to which attention was given. Not all solutions of this type satisfy real-life problems. There are other practical considerations in the geometric composition of pallet loads and in the palletization task in general. Among such considerations are the maintaining of integrity of the load during transport that is the pallet load stability, load clampability, approval of different aspects as loading and unloading times, and WIP (Work-In-Process)..

## 2.1 Volumetric Pallet Utilization, VPU

When talking about pallet volume, what is meant is the product of the pallet's base area with the maximum stacking height allowed for the pallet. Volumetric Pallet Utilization, VPU is the percentage of the total pallet volume that is actually filled with boxes. Initially, the VPU is 0% and as boxes are loaded on the pallet, the VPU increases and can reach 100% in optimal conditions. The volumetric pallet utilization can be generally expressed as in equation 2.1.1.

$$VPU\ (\%) = \frac{\sum_{i=1}^{N} l_i \times w_i \times h_i}{L_p \times W_p \times H_p} \times 100 \qquad \textbf{(2.1.1)}$$

where;

| | |
|---|---|
| N | = the number of boxes on the pallet |
| $l_i$, $w_i$, $h_i$ | = the dimensions of box i |
| $L_p$, $W_p$, $H_p$ | = the dimensions of the pallet |

When the palletization process is carried out by layers, the VPU may be obtained from the individual layer utilizations. Each layer utilization is separately computed by the product of the area coverage of that layer by its height (Yang, 1993). On the other hand, when palletization is carried by columns, the VPU is based on the individual columns' utilizations.

When interlocked stacking is used to load boxes on the pallet, it is more likely that the VPU formulation will be similar to that of 'sequential' column stacking. However since interlocked stacking may cause to advantage the interlocking of boxes, the stacked columns themselves may merge at some stacking levels. In such case, it is no longer possible to discern individual column stacks but rather an interlocked load of boxes. Then, the VPU can only be calculated by the formulation of equation 2.1.1 above.

## 2.2 Load Stability

A pallet load will consist of a combination of several column stacks and/or layers of boxes and, in both warehouse storage or distribution, will be subjected to a variety of forces. Each box will need to maintain its position in the stack when subjected to '*static*' forces of other boxes, in the stationary stack. In addition, during transportation the load will be subjected to '*dynamic*' forces. These two aspects require that consideration be given to the support afforded to each individual box, and to the interaction between adjacent boxes. Such considerations and requirements are referred to as the *stability criterion.*

Based on the above definition and the stability studies discussed in the literature, especially that of Hanna (1993) on the finite element method, two practical stability rules are deduced and are used to incorporate stability considerations in this research. These two rules are:

Rule #1: If two boxes of same base dimensions and different heights are available, make a block combination of the two such that the box with greater height is placed on top of the box with smaller height. It is better in terms of static stability for the box in the bottom while it is of no harm to the box with larger height.

Rule #2: If a stack can be made of several boxes such that it is of two layers, set the layer with lesser boxes on top of the layer with more number of boxes.

## 2.3 Work-In-Process, WIP

Work-in-process is defined as the temporary storage space to which boxes are taken in between pallet loading operations. The need for such space is clearly understood when some of the boxes arriving at the end of the conveyor line to the pallet loading stage, are required by the palletization algorithm to lay aside for a while until the time is right for them to be loaded onto the pallet. This adds some flexibility to the decision making process of the algorithm to improve the quality of the palletization with respect to performance criteria. In some situations, these criteria may be the volumetric pallet utilization and/or load stability. It is clear that all decisions of this type are executed on-line during the actual palletization process. While the use of the WIP may help achieving better volumetric pallet utilization and pallet load stability, it is definitely a burden on the palletization time since it will require more number of pick-and-place operations by the robot arm. Palletization time is discussed in more details in the next section.

## 2.4 Robotic Palletization Time, RPT

Though there are many transportation mechanisms that can be used to move boxes in the palletization work cell, there is a common need to minimize the total palletization time. One common method found in industry uses a robot arm that is in charge of moving boxes from the conveyor to the pallet or to the WIP or from the WIP to the pallet. Consequently, one may define '*Robotic Palletization Time*, RPT.' as the total elapsed time taken by the robot to perform all the above mentioned motions until boxes are no longer loaded on the pallet. Assuming that before any action, the robot is always at its home position, the path describing the motion will include going from the home position to the "pick-up" location then, after arrival at "place" location or destination, the robot is to go back to its home position. Therefore one should add the times taken by the robot to go back and forth to and from its home position and the various system locations that are the conveyor, the pallet, and the WIP.

Therefore if the robot command is to pick up a box from the conveyor and place it on the pallet, the overall elapsed time for the command, $T_{C\text{-}P}$ is

$$T_{C-P} = t_{H-C} + t_{C-P} + t_{P-H} \tag{2.4.1}$$

where

$t_{H\text{-}C}$ = robot motion time from the home position to the conveyor,
$t_{C\text{-}P}$ = time taken by the robot to move a box from the conveyor to the pallet,
$t_{P\text{-}H}$ = robot motion time from the pallet to the home position,

Similarly, for a conveyor/WIP operation, $T_{C\text{-}WIP}$ is

$$T_{C-WIP} = t_{H-C} + t_{C-WIP} + t_{WIP-H} \tag{2.4.2}$$

and for a WIP/pallet operation, $T_{WIP\text{-}P}$ is

$$T_{WIP-P} = t_{H-WIP} + t_{WIP-P} + t_{P-H} \tag{2.4.3}$$

Furthermore, if

$T_{Idle}$ = total idling time in which the robot at its home position, waits for a motion command.
$N_{C\text{-}P}$ = the number of times the robot performs a conveyor/pallet operation,
$N_{C\text{-}WIP}$ = the number of times the robot performs a conveyor/WIP operation,
$N_{WIP\text{-}P}$ = the number of times the robot performs a WIP/pallet operation.

The robotic palletization Time, RPT is calculated by formula:

$$RPT = N_{C-P} \times T_{C-P} + N_{C-WIP} \times T_{C-WIP} + N_{WIP-P} \times T_{WIP-P} + T_{Idle} \quad \textbf{(2.4.4)}$$

The idling time of the robot is mainly affected by the irregular time intervals between the arrival of boxes on the conveyor at the pick-up location. There are some other physical factors that may increase the idling time of the robot arm which are the computation time required by the palletization algorithm, the response times of hardware components such the conveyor and the box identification mechanism. The presence of above factors is essential to the palletization problem and their effect on $T_{C-P}$, $T_{C-WIP}$, $T_{WIP-P}$, and $T_{Idle}$ is uncontrollable unless better equipment are used. However conceptually, lower RPT is a result of efficient palletization algorithms which work on minimizing $N_{C-WIP}$ that is the number of times a box needs to be taken from the conveyor to the WIP. Since for every conveyor/WIP motion, there is a WIP/pallet equivalent, one can visibly notice the relation between the WIP and the RPT. The less boxes are required to go to the WIP, the lower the overall palletization time and better. Thus, the performance of palletization algorithms is measured by such indices as WIP and palletization time. Furthermore, these indices are important when comparing palletization algorithms together. One way to combine all the different indices into one global index for algorithm evaluation is to give weight factors for each index according to its importance. However, the weight factor for each index may vary from application to application. Moreover, the use of such global index is restricted to comparing algorithms that solve identical problems in identical working conditions.

# CHAPTER 3.0 PROPOSED HEURISTIC MODELS

The proposed models can be visualized as one basic heuristic model and a modified version of that model that adds other considerations to its **"raison d'être"**. They are both of the heuristic procedure type. The basic 3D palletization problem solved by the two heuristics has the following main characteristics:

+ There are many box types different in length, width, and height.
+ The boxes arrive to the palletization workstation on a conveyor in a stochastic sequence.
+ The boxes are required to be loaded on a pallet with the use of robotic manipulators.
+ There is a temporary holding area, WIP to which boxes may be taken if necessary before loading onto the pallet.
+ Boxes are loaded on the pallet in interlocked stacks which may sometimes merge together.

The two proposed models differ mainly from each other in the performance criteria to which top priority is given. The first heuristic model entitled ***"3D Loading Heuristic for Different Box Types"*** is designed to solve the general 3D palletization problem with the main concern of maximizing the pallet volumetric utilization and as much as possible minimizing the WIP area as well as the total palletization time. Boxes are loaded onto the pallet as long as there is enough room for them. The second proposed heuristic model entitled ***"3D Loading Heuristic with Demand Requirements and Stability Measures"*** solves the same general 3D problem but with loading stability considerations going hand in hand with the other performance criteria. Moreover, it is assumed that there is a particular demand for each box type and when a box whose demand is zero enters the system, it is not loaded onto the pallet.

## 3.1 Basic Model Concepts

Before going further in discussing each of the heuristics, some terminologies and concepts are to be defined and clarified.

### 3.1.1 Partitioning empty space and sub volume creation

A *sub volume* is the three-dimensional equivalent of a subarea which is commonly found in the literature of two dimensional palletization algorithms. To introduce the term sub volume, one can look at how the remaining empty space of a pallet of length L, width W, and height H is partitioned when a box of length l, width w, and height h is loaded in it as shown in figures 3.1.

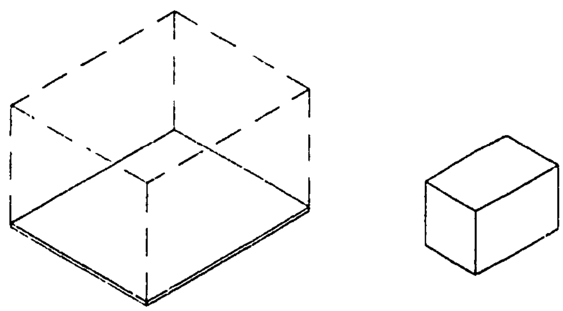


**Figure 3.1: Box being loaded on an Empty Pallet**

One can load the box within the boundaries of the pallet's base in three typical locations. It can be loaded at one of the four corners, it can be put such that it touches only one side of the base, and it can be loaded in the middle where it does not touch any of the sides of the pallet's base. Figure 5.1.2. represent these three cases.

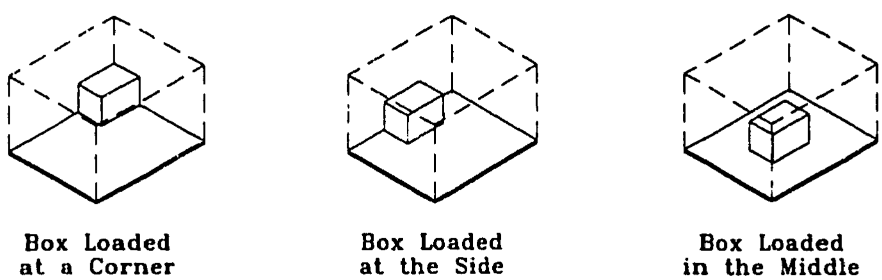



**Figure 3.2: Typical Box Locations on the Pallet**

The volumes of the remaining empty space in the three cases are equal. They are the volume of empty pallet minus the volume of the box. However, the way to partition them into small empty rectangular volumes is different. Moreover, the number of these small empty volumes is different in each situations. However, regardless of their number, there are different ways to define these empty volumes. Thus, how to partition the remaining empty volume is a question to be answered in a latter section. Figure 5.1.3 shows examples of partitioning the remaining volume for the three box locations discussed earlier.

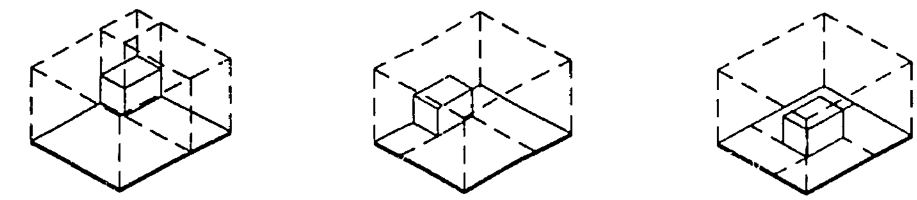



**Figure 3.3: Different Number of Sub Volumes for Different Box Locations**

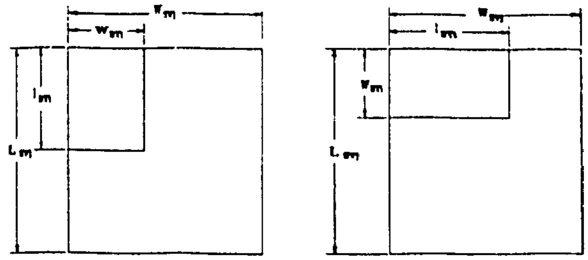
Each of these small empty rectangular volumes is called a *sub volume (SV)*. Therefore a sub volume can be defined as: *"a rectangularly shaped three dimensional empty space found anywhere at any given time on a pallet."* With the same total empty space volume, as the number of sub volumes increases, their individual volumes decreases. And the smaller the volume of a sub volume, the smaller boxes, the sub volume can accommodate. In other words, a relatively large box is not likely to fit into sub volumes of smaller sizes. In turn, this larger box will most probably be carried to the WIP where it will wait until the time is right for it to be loaded onto the pallet. Not only will the WIP be increased, but also time accounted for carrying the box from the conveyor to the WIP and later from the WIP to the pallet will increase the total palletization time. Therefore, as a general rule and as one of the heuristic's decision making criteria, partitioning of empty space is performed such as to minimize the Total Number of Sub Volumes, (TNSV) thus maximizing the individual volumes and capacities of sub volumes to accommodate small boxes as well as larger ones. Based on the above discussion, a recommended box loading location is at one of the four corners of the pallet or of sub volumes since this procedure will tend to give better results concerning pallet Volumetric utilization, WIP, as well as total palletization time. Furthermore, the closest sub volume corner to the pallet's origin, (0,0,0) is chosen as the corner at which the box is loaded. This ensures that there are no gaps between boxes when they are loaded adjacent to each other on the pallet.

### 3.1.2 Loading a Box into a Sub Volume

When loading a box in a sub volume, there are two important features that need to be considered in order to understand what is really going on. The first of the two issues is finding the optimal box orientation in the sub volume. The other feature is what happens to the sub volume as the box is loaded in it. Each of the two issues is discussed separately.

#### 3.1.2.1 Optimal Box Orientation in a Sub Volume

Generally, a box can fit into a sub volume in different orientations. Restricting the rotation of boxes to be only about the Z axis of the box which is vertical, the problem is simplified from 3D to a 2D surface orientation. A box type i of length $l_{BTi}$ and width $w_{BTi}$ can be placed in a sub volume j of length $L_{SVj}$ and width $W_{SVj}$ in two different orientations; *parallel orientation* where the largest box base dimension is orientated // to that of the sub volume, and *perpendicular orientation* where the box is rotated $90^{o}$ .



a) $l_{BTi}$ // $L_{SVj}$ and $w_{BTi}$ // $W_{SVj}$

b) $l_{BTi}$ // $W_{SVj}$ and $w_{BTi}$ // $L_{SVj}$

**Figure 3.4: Possible Box Orientations in a Sub Volume.**

The orientation of a box on the pallet in general and in the sub volume in particular is an important factor in maximizing the utilization of the pallet. Lee (1990) came up with a set of 2D formulae that compare the two orientations. Their respective capacities to fit as many as possible of the box are computed. The orientation loading to the larger loading capacity is chosen as the better of the two.

$$Max \left[ integer\left(\frac{L_{SV_j}}{l_{BT_i}}\right) \times integer\left(\frac{W_{SV_j}}{w_{BT_i}}\right) \; , \; integer\left(\frac{L_{SV_j}}{w_{BT_i}}\right) \times integer\left(\frac{W_{SV_j}}{l_{BT_i}}\right) \right] \qquad \textbf{(3.1.2.1)}$$

In order to avoid dealing with unrealistic portions of boxes, only the integer part of a division is used and the decimal part is discarded. The larger value means that more boxes of that type can be loaded onto the sub area (the base of the sub volume) with that box orientation. For instance, if the first set, int( $L_{SVj}/l_{BTi}$)*int($W_{SVj}/w_{BTi}$ ), gives a larger value, $box_i$ should be in the parallel orientation ( $l_{BTi}$ // $L_{SVj}$ and $w_{BTi}$ // $W_{SVj}$ ) and vice versa.

3.1.2.2 Result of Loading a Box into a Sub Volume

When a box is loaded onto the pallet, it must be loaded in one of the available M (TNSV) sub volumes. Initially M=1, that is the whole pallet is considered as that sub volume. Of the M sub volumes, M-1 sub volumes remain unchanged and one sub volume is filled either completely or partially. If the sub volume is completely filled (the 3 box dimensions match[1] the 3 dimensions of the sub volume) as in figure 3.5, the new TNSV = M-1+P with P = 0. Thus new TNSV = M-1 and the number of available sub volumes is reduced by one.

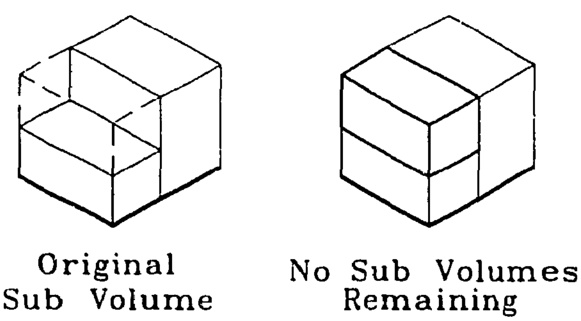



**Figure 3.5: Completely Filled Sub Volume.**

In the case of a partially filled sub volume there can be a creation of a maximum of three new sub volumes of which two are adjacent to the box and the third sub volume is on top of the box (P = 1, 2 or 3).

[1]In order to find the best orientation of a box in a sub volume, the box can only be rotated about its Z axis. (i.e. the Z dimension of the box is always parallel to the Z dimension of the sub volume whereas the X and Y dimensions can mix and match.)

When P = 1, two of the three box dimensions match with two of the sub volume dimensions. Refer to figure 3.6 for an illustration. The new TNSV = M-1+1 = M. Thus the number of available sub volumes remains unchanged.

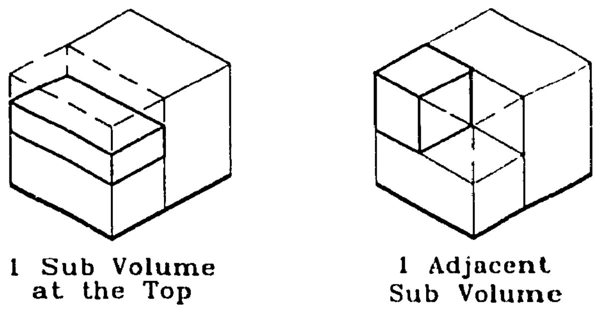



**Figure 3.6: Partially filled sub volume, Situation a) P = 1.**

When P = 2, only one of the three box dimensions match with one of sub volume's dimensions as in figure 3.7. In this case, the TNSV = M-1+2 = M+1. Thus the number of available sub volumes is increased by one.

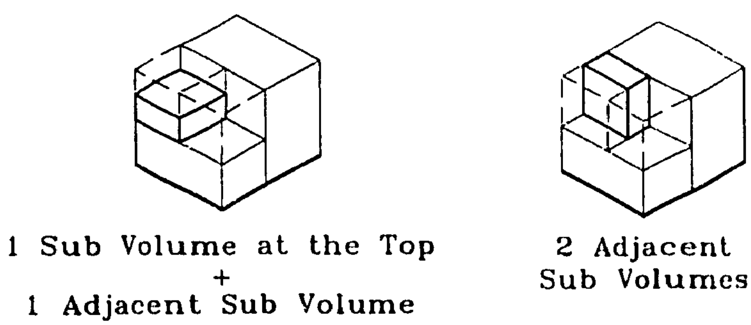



**Figure 3.7: Partially filled sub volume, Situation b) P = 2.**

Finally, when P = 3, none of the box dimensions match with any of the sub volume's dimensions as in figure 3.8. The TNSV = M-1+3 = M+2. Thus the number of available sub volumes is increased by two.

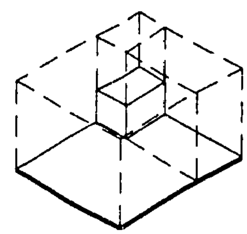



**Figure 3.8: Partially filled sub volume, Situation c) P = 3.**

### 3.1.3 Surface Levelling and Interlocking of boxes

When a box is loaded in a sub volume adjacent to boxes already on the pallet, it may sometimes occur that the remaining heights above them are equal. Thus, it is possible to level multiple interlocked column stacks and form a large column out of columns of smaller base dimensions. This phenomenon may occur when two sub volumes may be combined together after a box has recently been loaded on the pallet. Figure 3.9 shows the state of the column stacks on the pallet before the box is loaded in sub volume 1 belonging to column stack 2.

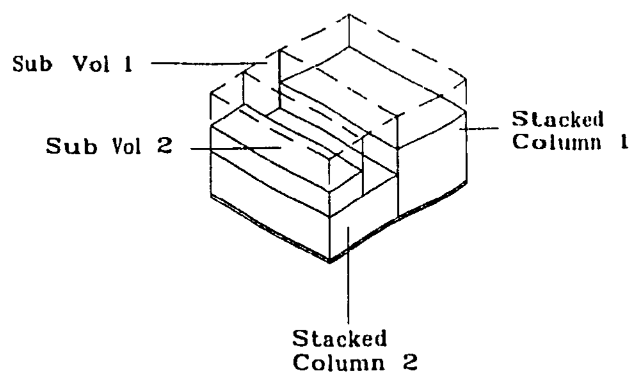



**Figure 3.9: Two Adjacent Stacked Columns before Surface Levelling**

Figure 3.10 illustrates how two sub volumes may be combined into one, thus the TNSV is decreased by one as opposed to being unchanged if no sub volume combining had taken place. Also levelling of stacked columns 1 and 2 is made possible allowing the possibility of a larger box to be loaded on top of both columns.

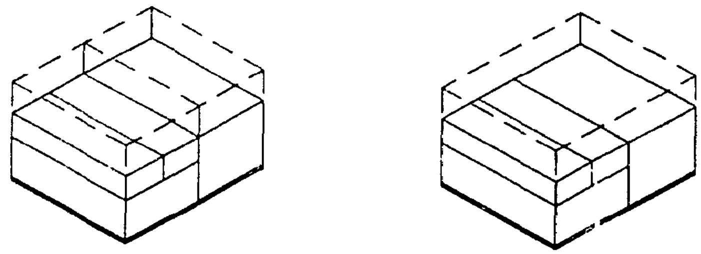


**Figure 3.10: Surface Levelling and Sub Volume Combining**

One benefit that is gained from considering combining sub volumes and the possibility of levelling stacked columns, is the enforcement of the interlocking of the columns stacks and so enhancing the loading stability of the pallet load since by loading a large box on top of both columns, figure 3.11, the guillotine cut is now interrupted.

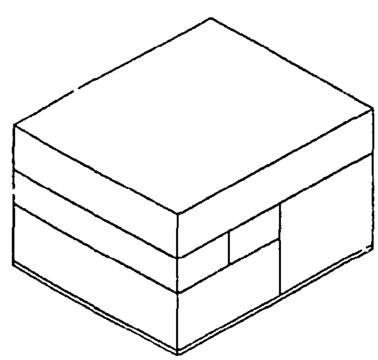


**Figure 3.11: Breaking the Guillotine Cut between Two Stacked Columns**

A final note on combining sub volumes is that although sub volumes combination may sometimes be feasible, not all cases are practical. The dilemma of choosing or not choosing to combine sub volumes together arise when there are more than one box as a candidate to go on the pallet. One candidate box, if loaded on the pallet may make sub volume combining and column stacking levelling possible. However, the resulting sub volume will not be as "efficient" as in the case of another candidate box when loaded, will increase the new TNSV. Which of the boxes to load to produce the more efficient pallet load is a decision made according to some other criteria discussed later.

## 3.2 Proposed Model 1: 3D Loading Heuristic for Different Box Types

### 3.2.1 Problem Definition and Characteristics

The palletization problem at hand was previously tackled in the literature. Specifically by **Yang (1993)** and **Arghavani (1993)** who have also proposed one and two directional 3D heuristics to solve it. Physically, the palletization workstation could be composed of a conveyor that transports the boxes to a robotic palletization cell. The cell could consist of a cartesian robotic manipulator that is in charge of loading the boxes on pallets on at a time. The robot picks up the boxes from the end of the conveyor - pick up place. - Somewhere on the conveyor, before the pick up place, a box identification mechanism could be installed to identify the box types going in the system before arriving to the pick up place. Sensors on the conveyor detect the presence of a box at the box identification location as well as at the pick up place. Whenever needed, the sensors cause the conveyor to stop. If a box arrives at the pick up place and there is no room for it on the pallet or loading priority is given to another box, the robot takes charge of transporting it to the WIP. As for the box types that are to be palletized, the two above mentioned heuristics from the literature have considered the following characteristics:

+ There is no restriction on the number of boxes available.

+ Boxes arrive in random sequences.

+ One pallet is loaded at a time.

+ Boxes can only be rotated about their Z axis.

+ Box dimensions are integer proportions of each other.

+ Boxes are grouped by similar heights.

+ Each group height has a holding area.

+ Maximum number of heights is three.

The proposed model solves for a more generalized form of the problem discussed above by eliminating the last four considerations.

The main conceptual difference between Model 1 and the heuristics in the literature is regarding the palletization technique by which boxes fill the pallet. Yang used the Layered-Palletization method where a new layer of boxes is started only when the layer underneath is completely filled. Arghavani used the Stacked-Palletization method where a column stack is started only when the previous stack is completely filled. Model 1 combined both techniques and developed the interlocked stacking loading methodology in which a box can be loaded whether on top or adjacent to any box on the pallet at any time as long as the constraints are satisfied and the heuristic rules are followed.

### 3.2.2 Objectives of the Proposed Model

In solving the palletization problem with the above characteristics, the heuristic attempts to achieve near optimum solutions with respect to maximum volumetric pallet utilization as a main goal. As secondary levels of optimization, the heuristic aims at reducing the maximum WIP as well as minimizing the total palletization time. Moreover, the heuristic rules indirectly build relatively stable pallet loads though load stability consideration is not intended in any way.

### 3.2.3 Model Inputs and Assumptions

As any other solution to the palletization problem, this heuristic requires some input information and assumptions.

#### 3.2.3.1 Pallet Dimensions

The pallet dimensions and the allowable pallet height must be predetermined and given as input to the heuristic.

#### 3.2.3.2 Stacking and Sub Volume Height

All sub volumes on the pallet have maximum possible height. That is the maximum allowable height of the pallet minus their respective individual stack height.

#### 3.2.3.3 Box Dimensions

The heuristic is flexible to accommodate any box types provided that their dimensions are given as input to the heuristic. If box types are added or removed from the existing set, an appropriate update input should be given to the heuristic.

#### 3.2.3.4 Box Availability

There is no upper or lower limit regarding the quantity of each box type involved in the palletization process. As boxes come into the system, they are tested for loading on the pallet.

3.2.3.5 Box Stackability

All boxes are assumed to have uniformly distributed masses and they are all stackable meaning that box crushing is assumed not possible.

3.2.3.6 Nature of Incoming Box Arrival

Boxes arrive into the system one at a time. The distance between two consecutive boxes is not taken into account. However it is assumed there are always two boxes on the conveyor, one at the pick up place and one at the box identification location.

3.2.4 Model Constraints

The model requires some physical constraints of which some are somehow trivial but important to consider in order to develop a practical solution.

3.2.4.1 Box Overhanging and Clampability

In some cases, boxes are not loaded directly off the production line or off the storage shelves onto pallets but are stacked and transported in this form to another area for storage or loading. The normal transportation method employed in these instances is that of a clamp truck -- which clamps the load on two opposite faces. To allow such an operation to be successfully carried out, the stack must possess at least one pair of perfectly flat opposite faces. This requirement is termed '*clampability*.' Carpenter *et al.* (1985) proposed a clampability criterion which demands that a least two opposite stack sides be flat, and set a lower limit of percentage of the length of all box edges parallel to the plane of the clamping (apart from those edges that form the perimeter of a layer) must be in contact with other box edges.

No box overhanging is allowed by the proposed heuristic. That is 100% of the base of any box must have support from boxes underneath. However, since smaller boxes may be loaded on top of larger ones, the clampability requirement is not guaranteed especially when small boxes are loaded near the perimeter of the pallet.

### 3.2.4.2 Box Rotation

Boxes are not allowed to rotate about the horizontal axes. Hence boxes can have one of two orientations on the pallet according to the rotation about the vertical axis.

### 3.2.4.3 Sub Volume Constraints

Since box rotation is permissible, the orientation of a box in a sub volume need not be length to length and width to width as some boxes may have their widths lying along the lengths and perpendicular to the widths of their respective sub volumes. The following equations represent the loading constraints of box i in sub volume j.

1. For perpendicular box orientation, the box must have an equal or smaller length than the length of the sub volume. The box width must also be less than or equal to the width of the sub volume

$$\begin{aligned} L_{SVj} - l_{BTi} &\geq 0 \\ W_{SVj} - w_{BTi} &\geq 0 \end{aligned} \qquad \textbf{(3.2.4.3.1)}$$

2. For perpendicular box orientation, the box that has been turned 90° must have an equal or smaller length than the width of the sub volume and have an equal or smaller width than the length of the sub volume.

$$\begin{aligned} L_{SVj} - w_{BTi} &\geq 0 \\ W_{SVj} - l_{BTi} &\geq 0 \end{aligned} \qquad \textbf{(3.2.4.3.2)}$$

3. In both cases, the height of box i, $h_{BTi}$ is always orientation in the direction parallel to the height of the sub volume j, $H_{SVj}$ and:

$$H_{SVj} - h_{BTi} \geq 0 \qquad \textbf{(3.2.4.3.3)}$$

### 3.2.4.4 WIP Constraints

To avoid over-using the WIP and thus increasing the RPT, a constraint is set to limit the number of boxes of the same type that are allowed to go to the WIP. Consequently, there can be a maximum of one (1) box of every box type in the WIP area.

### 3.2.5 Decision Making Theory

In order to better understand the heuristic, it is of major importance to look into special features of the heuristic such as how to decide about the loading priority of the boxes or how to choose the best partitioning pattern when the available empty space can be partitioned in more than one way. Also, how to determine the sub volume that best suits a particular box.

#### 3.2.5.1 Priority Levels

Generally, box selection is based on three main comparison levels. These three priority levels decide which of several boxes is given priority over the others. First, the boxes are compared at the first priority level. If one box is given priority in one level, it is immediately chosen as the best box and is loaded onto the pallet. However, if a tie occurs among several boxes when compared in a priority level, the decision making is postponed to the following level. Finally, if the tie reaches priority level 3 and a decision is not yet made, a box is randomly selected. The three priority levels are individually explained in the following sections.

Priority Level 1: **TNSV**

As discussed in section 3.1.1 earlier, among the many positions where one can load a box, boxes are always loaded at the closest corner to the pallet origin. It was also discussed that as the number of sub volumes decreases, their relative individual volumes increases and the larger their volume is, the bigger the chance of loading large boxes as well as smaller sized boxes. Moreover if a box can completely fill one of the available sub volumes, the number of remaining sub volumes is reduced by one. The application of the discussed rules increases significantly the volumetric pallet utilization. Therefore, the new **Total Number** of **Sub Volumes**. (TNSV) remaining if each of the boxes being tested were loaded is chosen as index for priority level 1. It is worth mentioning that sometimes, early in the test, two boxes may result in equal TNSVs. However, one box may also provide surface levelling and some sub volumes may merge together resulting in a lower new TNSV for that box which gives it advantage over its rival. Accordingly, priority level 1 is summarized in that the box with smallest index value (TNSV) is the box chosen to load on the pallet.

Priority Level 2: **PRVLC**

Priority level 2 is sought only when there is a tie between two boxes in level 1. Since the boxes come into the system in random order, optimization is used to solve for an unknown future. Thus, the procedure used in these circumstances can only be qualitative. Consequently, since each box considered for optimality results in a distinct partitioning pattern of the remaining volumed if this box were loaded, the total capacities of all sub volumes to accommodate larger varieties of box types, defines the second level of comparison. To better understand the meaning of this priority level, some parameters should be introduced.

If the total number of box types is **TNBT**, **I**ndividual Box Type Sub Volume **L**oad **C**apacity, **ISVLC** is the maximum number of boxes of a given type that fits in each sub volume in one orientation. For instance, the ISVLC of sub volume, $SV_j$ for box type, $BT_i$ is:

$$ISVLC_{ij} = Max\ [\ integ(\frac{L_{SV_j}}{l_{BT_i}}) \times integ(\frac{W_{SV_j}}{w_{BT_i}})\ ,\ integ(\frac{L_{SV_j}}{w_{BT_i}}) \times integ(\frac{W_{SV_j}}{l_{BT_i}})\ ] \times integ(\frac{H_{SV_j}}{h_{BT_i}}) \qquad (3.2.5.1.1)$$

$$\forall\ i = 1, 2, \ldots\ TNSV$$
$$\forall\ j = 1, 2, \ldots\ TNBT$$

The **M**aximum **C**apacity of each **B**ox **T**ype for all sub volumes, **MCBT** is the sum of all ISVLCs for each box type. For instance, the MCBT of $BT_i$ for all sub volumes is:

$$MCBT_i = \sum_{j=1}^{TNSV} ISVLC_{ij} \qquad (3.2.5.1.2)$$

As a quantitative index by which comparison is made at level 2, the **P**artitioned **R**emaining Volume **L**oad **C**apacity, **PRVLC** defines the maximum filling of all sub volumes with all box types. It is expressed as the sum of the MCBTs for all box types that can be expected in the system. For instance. for a box among others being tested, its PRVLC is:

$$PRVLC_{Test\ Box} = \sum_{i=1}^{TNBT} MCBT_i \qquad (3.2.5.1.3)$$

Based on the above discussion, the larger the value of the PRVLC of a box, the more likely is its corresponding pallet layout to accept a larger variety or a larger number of box types. Therefore, the box is given better chances of being chosen to load on the pallet. When two boxes result in equal values of PRVLCs, the tie break is postponed to the third comparison level to decide on their loading priority.

Regarding the relation between the MCBTs and their PRVLC, It is of great value to discover that it is not only a matter of summation but also a matter of finding the proportion of each box type to the PRVLC. Positive Flexibility of box type i, $PF_i$ is defined as the flexibility of the partitioned pattern to accommodate box type i. It is determined as follows:

$$PF_i = \frac{MCBT_i}{PRVLC} \qquad \textbf{(3.2.5.1.4)}$$

Since the MCBTs are independent and mutually exclusive meaning that only one situation may occur at any state analysis of the PRVLC calculation, the probability of all situations can be formulated as the sum of all $PF_i$'s:

$$PF_1 + PF_2 + \ldots + PF_{TNBT} = 1 \qquad \textbf{(3.2.5.1.5)}$$

When a particular $PF_i = 0$, the probability of the partitioned pattern to accommodate box type i is nil meaning that if a box of type i comes into the system while this pattern is the actual layout of the pallet, none of the available sub volumes on the pallet would be large enough for the box to fit into. At this stage, the only feasible solution would be to take the box to the WIP.

Priority Level 3: **Total Zero Count**

As a final level in which loading priority of boxes is decided, priority level 3 takes into account the handicap of each sub volume of the pattern in terms of not being able to accommodate particular box types. This situation is manifested in a value of 0 for the ISVLC of each sub volume for each box type. As opposed to a non zero positive value for ISVLC, **Zero-Count**, $ZC_{ij} = 1$ whenever $ISVLC_{ij} = 0$ meaning that box type i does not fit in sub volume j. When the ZCs of all sub volumes for a particular box type are added together, a new parameter called **Zero-Count** for a **Box Type, ZCBT** is produced.

$$ZCBT_i = \sum_{j=1}^{TNSV} ZC_{ij} \qquad \forall\ i = 1, 2, \ldots TNBT \qquad \textbf{(3.2.5.1.6)}$$

ZCBT can reach a maximum value equal to the TNSV. In such case, none of the TNSVs can accommodate the box type. Furthermore, if the demand for this box type is greater or equal to 1, then the tested box leading to such outcome is eliminated from the race to the pallet.

On another hand, if all the ZCBTs for all box types are added together, the result is the index used at level 3 for breaking the tie between boxes tied at level 2. This index is called the Total **Zero-Count, TZC** for the partitioned pattern of the among others being tested for loading on the pallet.

$$TZC_{TestBox} = \sum_{i=1}^{TNBT} ZCBT_i \qquad \textbf{(3.2.5.1.7)}$$

It is clear that the smaller the TZC, the better performance will the pattern have in accommodating more box types in a larger number of its sub volumes. Therefore, the box which results in a smaller total zero-count is selected for loading on the pallet. To complete the quantitative analysis of the TZC, it is important to compare it to the maximum value it can possibly take, $TZC_{max}$. $TZC_{max}$ is reached only when the number of remaining sub volumes on the pallet is $\neq 0$, i.e. the pallet is not 100% full and all of the remaining sub volumes are too small to fit any of the box types. For each of the boxes tested for loading priority, $TZC_{max}$ is expressed as:

$$TZC_{\max,\ Box} = TNBT \times TNSV_{Box} \qquad \textbf{(3.2.5.1.8)}$$

Since level 3 is only used to compare boxes that share a common TNSV and a common value for their PRVLCs, there can only be one value of $TZC_{max}$ . The proportion of each TZC for each box considered to $TZC_{max}$ is named **Negative Flexibility, NF** of every box. It is expressed as follows:

$$NF_{Test\ Box} = \frac{TZC_{Test\ Box}}{TZC_{\max}} = \frac{TZC_{Test\ Box}}{TNBT \times TNSV} \qquad \textbf{(3.2.5.1.9)}$$

The concept of negative flexibility is another way of looking at the total zero-count in which one is able to study the effect of the box loading process on the flexibility of the pallet layout to accommodate box types. One is able to recognize the progressive diminishing of flexibility as the volumetric pallet utilization increases. This is due to the decreasing volume of the remaining empty space on the pallet as boxes are loaded on it.

3.2.5.2 Choosing Between Partitioning Patterns

When a box is loaded onto the pallet, the remaining empty volume can be partitioned in more than one way. In order to choose the best (optimal) partitioning pattern, the PRVLC of each pattern is computed and they are all compared at priority level 2. The pattern with higher value is the one chosen. However, if there is a tie among partitioning patterns, the tie break is make at level 3 as discussed above.

3.2.5.3 Choosing between Sub Volumes

When there are many sub volumes that can fit a box, there is a need to find which one of them should be chosen to load the box into in order to obtain best results. Since the empty space on the pallet has been previously partitioned into an old pattern made of M available old sub volumes, the following is the procedure to consider for finding the best loading position for $box_i$.

- Among the M old sub volumes, there are N sub volumes that can accommodate $box_i$, $N \leq M$.
- Consider loading $box_i$ in each one of the N sub volumes at a time.
- Starting at n = 1
- The best box orientation in the sub volume is found.
- The remaining empty space of the sub volume used can be partitioned in 2 ways, thus 2 sub patterns.
- Compare the 2 sub patterns at priority levels 2 and 3 if needed.
- The chosen new sub pattern is composed of P new sub volumes, ($P_{max} = 3$).
- The M-1 old sub volumes that are intact, when added to the P new sub volumes, they form one new pattern composed of a TNSV = M - 1 + P sub volumes

- If one or some of the P new sub volumes can be combined with any of the M-1 old sub volumes thus forming K additional new patterns each having a TNSV ≤ M - 1 + P

- The best of the K+1 patterns is selected according to priority levels 2 and 3.

- Repeat the above procedure for all the N sub volumes that can accommodate $box_i$, thus $box_i$ will have N PRVLCs and TZC corresponding to the N sub volumes.

- Select the optimal sub volume to accommodate $Box_i$ according to priority level 2 and 3.

### 3.2.6 Heuristic for Model 1

Overview of the Heuristic

As a new box enters the system, it is given the title of $Box_{c2}$ when it arrives at the box identification mechanism. If no box is present at the end of the conveyor in the pick up place. $box_{c2}$ moves to the pick up place and its name changes to $Box_{c1}$. Afterward, a new box enters the system and becomes the new $Box_{c2}$.

Assuming that there are many boxes in the WIP, they are all compared among each other and the best box in the WIP, $Box_{WIP}$ is selected according to priority levels 1, 2, and then 3. If $Box_{c1}$ and $Box_{WIP}$ are of the same type, $Box_{c1}$ is loaded right away on the pallet since there can only be a maximum of one of each box type in the WIP. Not only will this decision contribute to maximizing the VPU, but will also minimize the RPT since it avoids the possibility of loading $Box_{WIP}$ first then being obliged to carry $Box_{c1}$ to the WIP afterwards in case $Box_{c2}$ has priority over it. However, if $Box_{WIP}$ and $Box_{c1}$ are not of the same type, they are compared using priority levels 1, 2, and then 3. If $Box_{WIP}$ is found to have priority over $Box_{c1}$, it is loaded right away on the pallet thus working on reducing the WIP. Then the cycle loops back to find a new $Box_{WIP}$, among the remaining boxes in the WIP, if any. On the other hand, if $Box_{c1}$ has priority over $Box_{WIP}$ or if there is no boxes left in the WIP, $Box_{c1}$ is compared with $Box_{c2}$ using priority level 1 only. If the index (new TNSV) for $Box_{c1}$ is smaller or equal to that of $Box_{c2}$, $Box_{c1}$ is loaded on the pallet. Otherwise, $box_{c1}$ is carried to the WIP. $Box_{c2}$ advances on the conveyor and becomes the new $Box_{c1}$ and new box becomes $Box_{c2}$. The reason why $Box_{c1}$ is only compared with $Box_{c2}$ at priority level 1 only is to minimize the number of times $Box_{c1}$ must go to the WIP which will have a negative impact upon the RPT as well as the WIP performance measures.

The cycle continues until one of three situations occurs; the pallet is 100% filled, the remaining sub volumes on the pallet are too small for any of the boxes in the system, or boxes stopped entering the system.

The following steps are to determine the heuristic solutions to the 3D interlocked stacking palletization:

**STEP 1: Start.**

Initialize the system.

TNSV = 1.

$l_{SV1} = L_{Pallet}$, $w_{SV1} = W_{Pallet}$, $h_{SV1} = H_{Pallet}$.

**STEP 2: Check the available empty volume on the pallet.**

If TNSV = 0 ☞ Volumetric Pallet Utilization, VPU = 100%
Terminate.

Else, if TZC = TNBT x TNSV ☞ VPU = 100% is impossible
Terminate

**STEP 3:**

If there is no boxes in the WIP ☞ STEP 6.

Otherwise ☞ Find the best box in the WIP, $Box_{WIP}$ according to priority levels 1, 2 then 3.

**STEP 4: Compare $Box_{C1}$ to $Box_{WIP}$.**

If $Box_{WIP}$ is of the same type as $Box_{C1}$ ☞ Load $Box_{C1}$ ☞ Perform update ☞ STEP 8

Otherwise ☞ the best box is chosen according to priority levels 1, 2 then 3.

**STEP 5: Confirm Priorities.**

If $Box_{WIP}$ has priority over $Box_{C1}$ ☞ Load $Box_{WIP}$ ☞ Perform update ☞ Return to STEP 2.

**STEP 6: Compare $Box_{C2}$ to $Box_{C1}$.**

If $Box_{C1}$ is of same base dimensions as those of $Box_{C2}$ ☞ priority is to $Box_{C1}$.

Otherwise ☞ The best of the 2 boxes is chosen according to priority level 1 only.

**STEP 7: Confirm Priorities.**

If $Box_{C1}$ has priority over $Box_{C2}$ ☞ Load $Box_{C1}$ ☞ Perform update

Otherwise ☞ Move $Box_{C1}$ to WIP ☞ Perform update.

**STEP 8:**

$Box_{C2}$ ☞ $Box_{C1}$
New Box ☞ $Box_{C2}$
Return to STEP 2

For simpler and more efficient visualization of the heuristic, a flow chart containing all the steps involved is shown in figure 3.12.

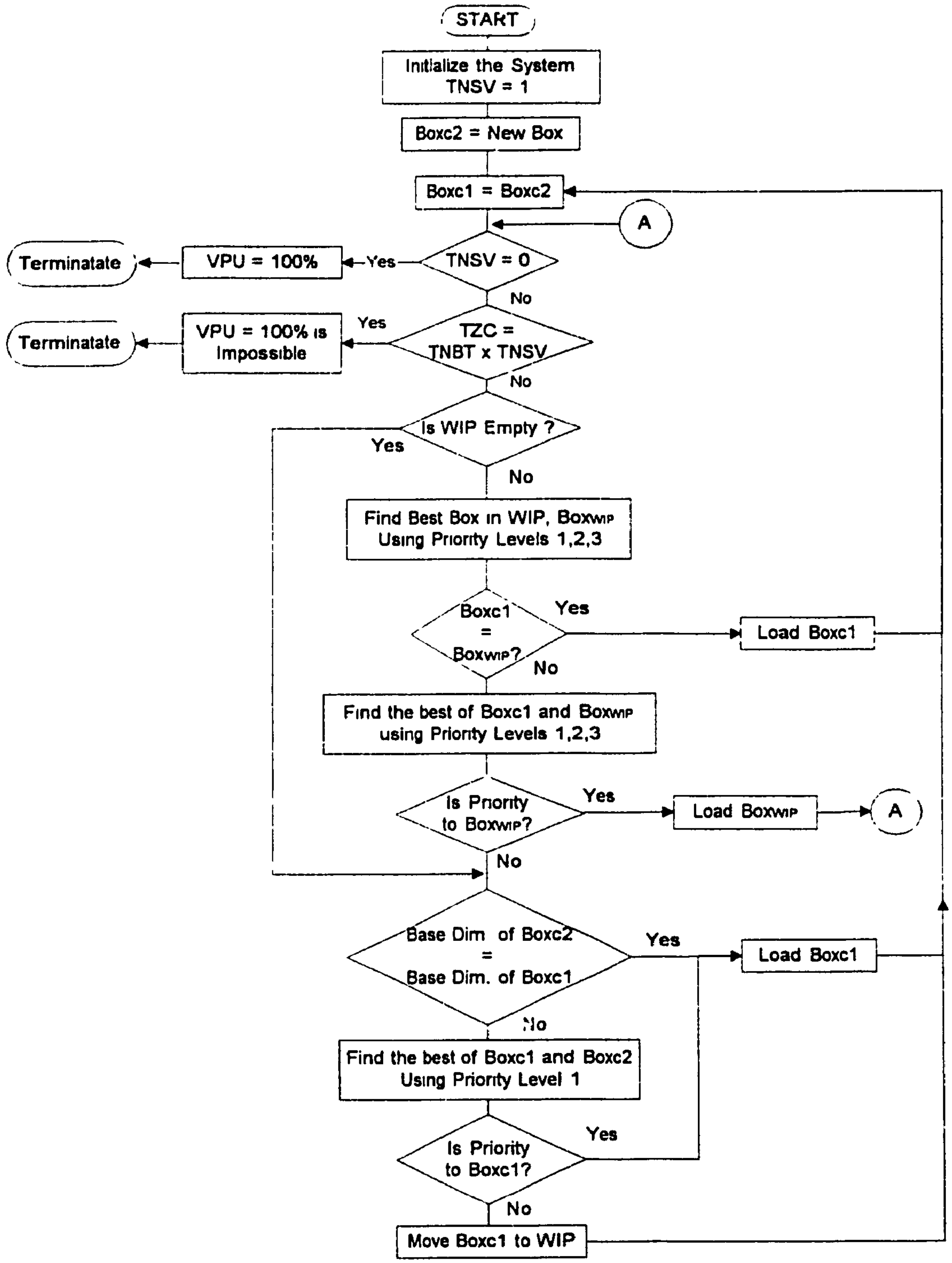



**Figure 3.12: Flow Chart for the 3D Loading Heuristic (Model 1)**

### 3.2.7 Note on Stability Considerations in Model 1

Although model 1 does not consider pallet load stability, it sometimes works indirectly towards achieving better pallet loads. This happens when two boxes are tied at priority level 1 - the TNSV - the box with largest PRVLC is chosen over the other. This may mean that the box chosen is smaller in size since it results a set of larger sub volumes whose PRVLC is larger. Thus, in such circumstances, when smaller boxes are loaded before larger ones, more stable pallet loads are built.

## 3.3 Proposed Model 2: 3D Loading Heuristic with Demand Requirements and Stability Measures

### 3.3.1 Problem Definition and Characteristics

Proposed model 2 is developed to solve a more complicated 3D palletization problem than the one that model 1 solved for. The main difference between the two problems is the fact that the basic problem is concerned with loading all the boxes coming on the conveyor as long as they keep coming to the system and the pallet can still accommodate them. However, palletization problem 2 loads on the pallet selected box types and quantities according to a demand requirement for each type of box. Boxes which have zero demands or whose demands are already satisfied are transported to the WIP permanently in addition to boxes being moved temporarily to WIP found in the case of palletization problem 1 solved for in the first model.

The main characteristics of palletization problem 2 are the following:

+ One pallet is loaded at a time.

+ Box types arrive in random sequences.

+ Boxes can only be rotated about their Z axis.

+ There is no restriction on the number of boxes available.

+ There is a restriction on the quantity of each box type required to load on the pallet.

+ Interlocked Stacking is used to load Boxes on the Pallet.

### 3.3.2 Objectives of the Proposed Model

While model 1 concentrated on achieving near optimal solutions for maximizing the VPU, minimizing the WIP and the RPT, the proposed model 2 gives priority to the application of some developed stability rules. Though, by doing so, the VPU, WIP, and RPT may be effected, the proposed model still aims at optimizing these performance criteria.

### 3.3.3 Changes in Model Inputs and Assumptions

Similarly to model 1, there are some inputs and assumptions that need to be taken care of in model 2. Only ones modified because of stability of demand requirements are discussed.

#### 3.3.3.1 Box Availability

There is no upper or lower limit regarding the quantity of each box type coming into the system. However, their random arrival may interfere with the available box types at every palletization stage.

#### 3.3.3.2 Input Demand for each Box type

As additional input to the model, the quantity of each box type that must be loaded on the pallet must be pre specified.

### 3.3.4 Changes in Model Constraints

Only the modified model constraints are discussed in this section.

#### 3.3.4.1 Box Quantity Constraints

The number of boxes of each type that need to be loaded on the pallet is fixed. Quantities larger than required are taken to the WIP. In other words, the maximum number of boxes on the pallet of each box type must not exceed its demand constraints.

$$NBOP_{BTi} \leq demand_{BTi} \qquad \forall\ i = 1, 2, \ldots\ TNBT \qquad \textbf{(3.3.4.1.1)}$$

Also the maximum number of boxes on the pallet, $NBOP_{Max}$ must not exceed the added demands of all box types.

$$NBOP_{Max} \leq \sum_{i=1}^{TNBT} demand_{BTi} \qquad \textbf{(3.3.4.1.2)}$$

#### 3.3.4.2 Block Constraints

Since blocks made of several boxes may be formed, there are some constraints regarding their loading in the available sub volumes. The following equations represent the loading constraints of block i in sub volume j.

1. For parallel block orientation, the block must have an equal or smaller length than the length of the sub volume. The block width must also be less than or equal to the width of the sub volume

$$\begin{aligned} L_{SVj} - l_{Block\ i} &\geq 0 \\ W_{SVj} - w_{Block\ i} &\geq 0 \end{aligned} \qquad \textbf{(3.3.4.2.1)}$$

2. For perpendicular block orientation, the block being turned 90° must have an equal or smaller length than the width of the sub volume and have an equal or smaller width than the length of the sub volume.

$$L_{SVj} - w_{Block\ i} \geq 0$$
$$W_{SVj} - l_{Block\ i} \geq 0 \qquad \textbf{(3.3.4.2.2)}$$

3. In both cases, the height of block i, $h_{Block\ i}$ must always be oriented in the direction parallel to the height of the sub volume j, $H_{SVj}$ and:

$$H_{SVj} - h_{Block\ i} \geq 0 \qquad \textbf{(3.3.4.2.3)}$$

3.2.4.3 WIP Constraints

There is no restricting constraints regarding the maximum quantity of each box type in the WIP. This is due to the possibility of moving unlimited number of boxes from the conveyor to the WIP because they have zero quantity demand or that their demands are already satisfied.

### 3.3.5 New Decision Making Theory

Based on the two stability rules discussed in chapter 2, section 2 regarding pallet load stability, a new term is introduced. It is the term "block". A block is a combination of several boxes forming a larger three dimensional rectangular. A block can be made of boxes present in the WIP and on the conveyor provided that their demand constraints are not violated. The main idea behind the formation of blocks is giving the opportunity to implement the two stability rules within the block. Therefore, when the pallet is composed of several blocks of boxes, provided that the stability of the individual blocks are enhanced, it is more likely that the overall stability of the pallet load is improved. Figure 3.13 shows how more stable blocks can be formed through the application of stability rule 1 which says:

Rule #1: If two boxes of same base dimensions and different heights are found, make a block combination of the two such that the box with greater height is placed on top of the box with smaller height.

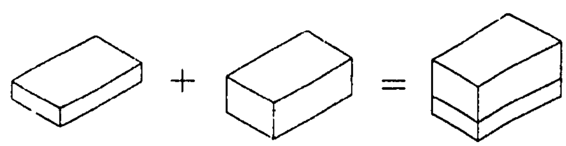


**Figure 3.13: Using Stability Rule 1 to Build Stable Blocks**

Figure 3.14 below shows how stability rule 2 can also improve the stability of blocks.

Rule #2) If a block can be made of several boxes in two layers, set the layer with smaller number of boxes on top of the layer with more number of boxes.

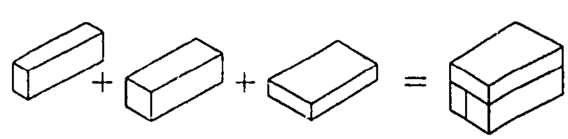


**Figure 3.14: Using Stability Rule 2 to Build Stable Blocks**

The possibility of block formation is checked at every stage of the palletization process. Whenever a new box with non zero demand enters the system, if a block can be formed, according to the specifications of stability rule 2 first and then stability rule 1, the block is treated as a box and tested for loading on the pallet in one of the available sub volumes. The reason why stability rule 2 has priority over rule 1 is because the block pattern made by rule 2 involves three boxes which is more than that of rule 1. Therefore, stability rule 2 would build bigger blocks with higher load stability. Moreover, if it occurs that a particular box is part of more than one block, the block with bigger volume is chosen for loading.

However, there is a price that sometimes has to be paid as a result of the block building process. If the new feasible block can fit somewhere on the pallet, some special arrangements must be made to physically load the block on the pallet in the desired pattern. This may sometimes require moving boxes from the conveyor to the WIP temporarily, thus increasing the maximum WIP at any one time as well as increasing the robotic palletization time.

The following section describes the heuristic of model 2 which was designed so as to keep the WIP and the RPT to a minimum as much as possible.

### 3.3.6 Heuristic for Model 2

The heuristic of model 2 is divided into two main sections. The first part of the heuristic is concerned with building stable blocks out of the boxes in the WIP and those on the conveyor, $Box_{C1}$ and $Box_{C2}$. Again, it must be reminded that this may be possible only when their demands are non zero. The second part of the heuristic is a modified version of the heuristic of model 1 but with box demand considerations.

**STEP 1:** Start.

Initialize the system.
TNSV = 1.
$l_{SV1} = L_{Pallet}$, $w_{SV1} = W_{Pallet}$, $h_{SV1} = H_{Pallet}$.

**STEP 2:** Check the available empty volume on the pallet.

If TNSV = 0 ☞ Volumetric Pallet Utilization, VPU = 100%
☞ Terminate.

Else, if TZC = TNBT x TNSV ☞ VPU = 100% is impossible
☞ Terminate

**STEP 3:** Check Updated Demand for $Box_{C1}$

if Updated $Demand_{Boxc1}$ = 0 ☞ Move $Box_{C1}$ permanently to the WIP
☞ $Box_{C2}$ becomes $Box_{C1}$, New Box ☞ $Box_{C2}$
☞ Repeat STEP 3

**STEP 4:** Study Stability Rule 2

If block possible using stability rule 2 ☞ Load the boxes forming the block as follows

if any or all the boxes of bottom layer of block are in the WIP

☞ Load them
☞ Decrease the Demand of each box loaded by 1

**Step 4.1:** Is Bottom Layer Complete?

If bottom layer complete

if $Box_{Top\ Layer}$ in WIP ☞ Load it ☞ Decrease $Demand_{Box\ loaded}$ by 1
☞ Block complete ☞ STEP 2

else if $Box_{C1} = Box_{Top\ Layer}$ ☞ Load it ☞ Block complete ☞ STEP 2

else if $Box_{C2} = Box_{Top\ Layer}$

if $Box_{C1}$ present ☞ Move $Box_{C1}$ to WIP

Boxc2 becomes $Box_{C1}$ ☞ Load it ☞ Block is complete ☞ STEP 2

Else (either $Box_{C1}$ or $Box_{C2}$ or both go in bottom layer)

if $Box_{C1} = Box_{Bottom\ Layer}$ ☞ Load it

if bottom layer complete ☞ $Box_{C1}$ is empty ☞ Step 4.1

else if BoxC1 empty ☞ $Box_{C2}$ becomes $Box_{C1}$ ☞ Load it
☞ Bottom Layer complete ☞ Step 4.1

else (BoxC1 not part block) ☞ Move $Box_{C1}$ to WIP ☞ $Box_{C2}$ becomes $Box_{C1}$
☞ Load it ☞ Bott. layer complete ☞ Step 4.1

<u>**STEP 5:** Study Stability Rule 1</u>

If block possible using stability rule 1 ☞ Load the boxes forming the block as follows

if $Box_{Bottom\ Layer}$ in WIP ☞ load it ☞ Decrease $Demand_{Box\ loaded}$ by 1

<u>**Step 5.1:** Is Bottom Layer Complete?</u>

If Bottom Layer Complete

if $Box_{Top\ Layer}$ in WIP ☞ Load it ☞ Decrease $Demand_{Box\ loaded}$ by 1
☞ Block complete ☞ STEP 2

else if $Box_{C1} = Box_{Top\ Layer}$ ☞ Load it ☞ Block complete ☞ STEP 2

else if $Box_{C2} = Box_{Top\ Layer}$
if $Box_{C1}$ present ☞ Move $Box_{C1}$ to WIP
$Box_{C2}$ becomes $Box_{C1}$ ☞ Load it ☞ Block complete ☞ STEP 2

Else (either $Box_{C1}$ or $Box_{C2}$ goes in bottom layer)

if $Box_{C1} = Box_{Bott.Layer}$ ☞ Load it ☞ Decrease $Demand_{Box\ loaded}$ by 1 ☞ Step 5.1

else ($Box_{C1}$ is not part of block) ☞ Move $Box_{C1}$ to WIP ☞ Boxc2 becomes $Box_{C1}$
☞ Load it ☞ Bott. layer complete ☞ Step 5.1

<u>**STEP 6:**</u>

If there are no boxes in the WIP ☞ STEP 9.

Otherwise ☞ Find the best box in the WIP, $Box_{WIP}$ from among those whose demand > 0 according to priority levels 1, 2 then 3.

STEP 7: Compare $Box_{C1}$ to $Box_{WIP}$.

If $Box_{WIP}$ is of the same type as $Box_{C1}$ ☞ Load $Box_{C1}$
☞ Decrease $Demand_{Box}$ by 1
☞ STEP :1

Otherwise ☞ the best box is chosen according to priority levels 1, 2 then 3.

STEP 8: Confirm Priorities.

If $Box_{WIP}$ has priority over $Box_{C1}$ ☞ Load $Box_{WIP}$
☞ Decrease $Demand_{Box}$ by 1
☞ Return to STEP 2.

STEP 9: Compare $Box_{C2}$ to $Box_{C1}$.

If $Box_{C1}$ is of same base dimensions as those of $Box_{C2}$ ☞ priority is to $Box_{C1}$.

Otherwise

if $Demand_{Boxc2} = 0$ ☞ Priority is to $Box_{C1}$

else, the best of the 2 boxes is chosen according to priority level 1 only.

STEP 10: Confirm Priorities.

If $Box_{C1}$ has priority over $Box_{C2}$ ☞ Load $Box_{C1}$ ☞ Decrease $Demand_{Box}$ by 1

Otherwise ☞ Move $Box_{C1}$ to WIP.

STEP 11:

$Box_{C2}$ ☞ $Box_{C1}$
New Box ☞ $Box_{C2}$
Return to STEP 2

For better visualization of the heuristic rules, a flow chart containing all the steps involved is shown in figure 3.15.

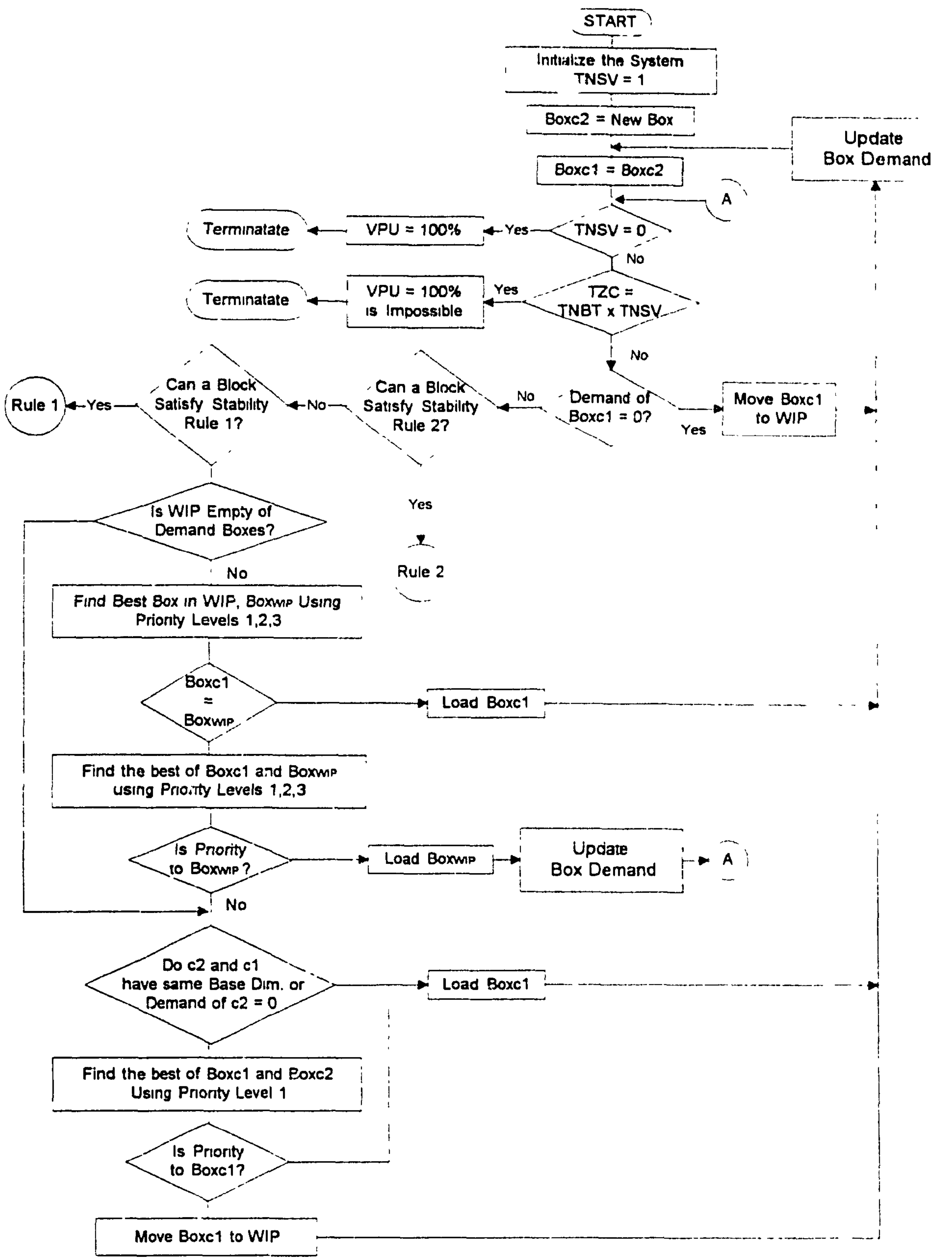



**Figure 3.15: Flow Chart for the 3D Loading Heuristic with Demand Requirements and Stability Measures (Model 2)**

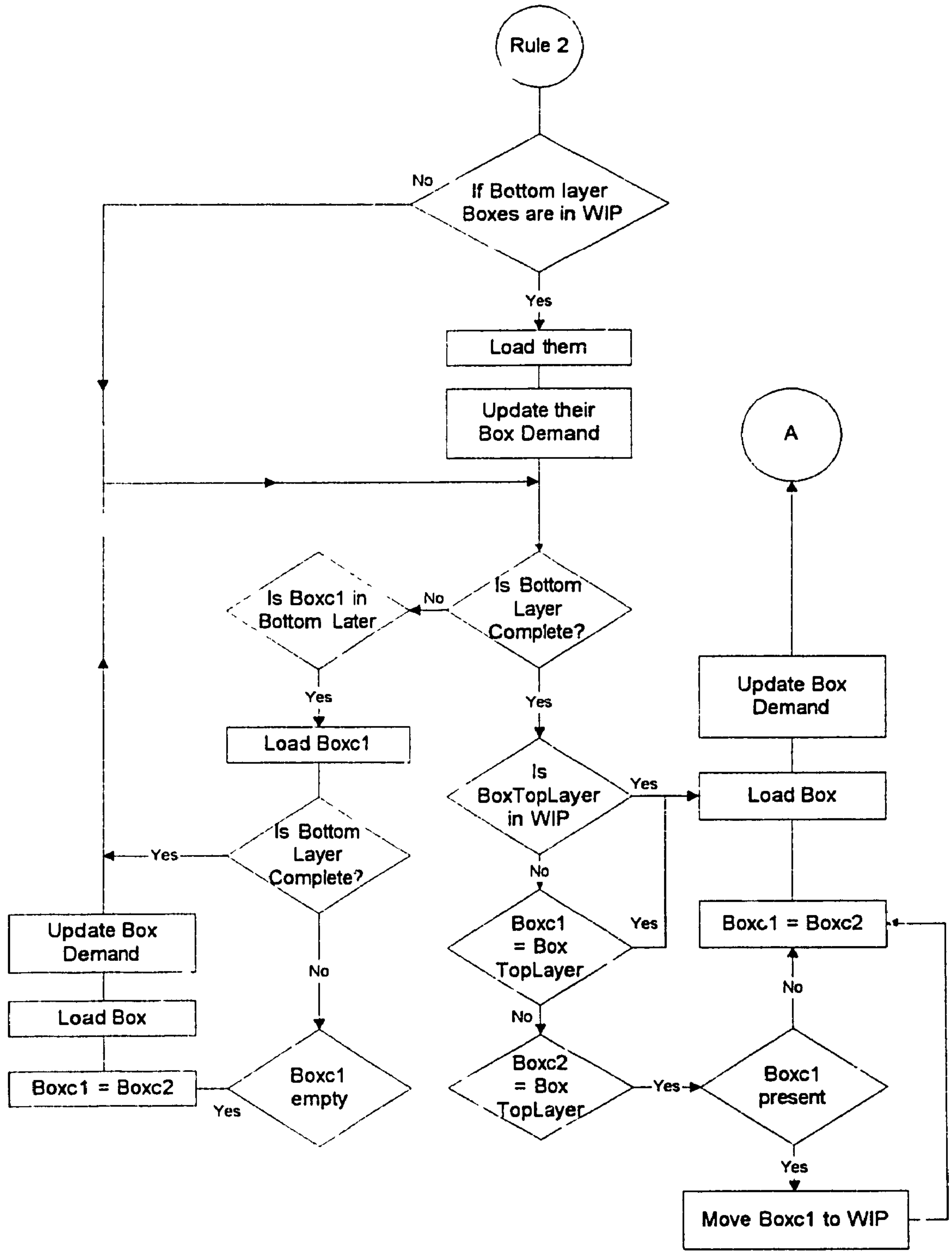



**Figure 3.15 Cont'd: Stability Rule 2**

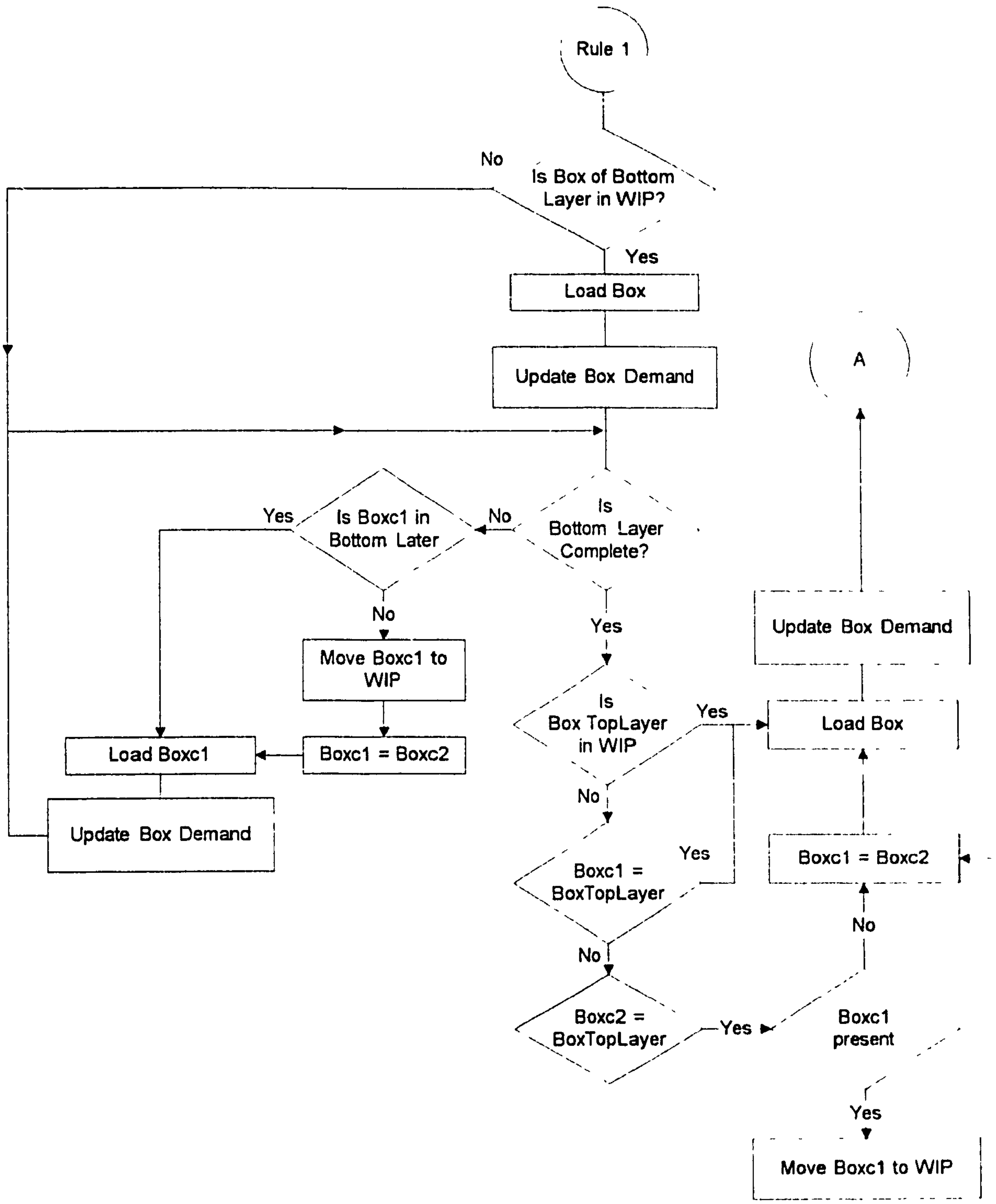



**Figure 3.15 Cont'd: Stability Rule 1**

# 4.0 COMPUTER PROGRAMS DESCRIPTION

This chapter describes the computer programs written in the Turbo C++ programming language to implement both models. All the programs' input and output data as well as all the modules - functions - involved in the main programs are examined and explained.

## 4.1 Turbo C++ Program for Model 1

### 4.1.1 Input Data and Variables:

The following are the input data entered by the user to set up the program before the system is started and the program executed.

**TNBT** integer variable is used to store the number of different box types that may be encountered in the system.

**l_boxtype, w_boxtype,** and **h_boxtype** are 3 one-dimensional integer arrays used to store integer numbers that represent the lengths, widths, and the heights of all the box types respectively.

**pallet_size** is a one-dimensional integer array used to store the length, width, and the maximum allowable pallet height in the model.

4.1.2 Output Data and Variables:

As boxes are loaded on the pallet, the program prints an update report of the palletization process. Eventually, when the pallet is filled to its maximum, - not necessary 100% - or when boxes stop coming to the system, the program prints out a final palletization report containing all characteristics related to the palletization process. More precisely, all information regarding the pallet, the boxes on the pallet, the WIP, the conveyor, and all performance evaluation values.

**vol_pal_util** is a float variable (non integer) that represents volumetric pallet utilization, VPU in percent.

**boxesonpallet** is an integer variable which keeps record of the number of boxes that are on the pallet at any given time.

**L_vector_of_box_on_pallet, W_vector_of_box_on_pallet,** and **H_vector_of_box_on_pallet,** are 3 one-dimensional integer arrays representing the position vector (L,W,H) of the closest corner of every box on the pallet with respect to point (0,0,0) of the pallet.

**orientation_box_on_pallet** is a character array that keeps record of the orientation with which each box was loaded on the pallet. Whether it is " l//L and w//W" or "l//W and w//L" where lower case letters are the length and width of the box, and the uppercase letters are those of the pallet.

**TNSV** is an integer variable denoting the number of empty sub volumes remaining on the pallet, if any.

**L_subv, W_subv,** and **H_subv** are 3 one-dimensional arrays that store the length, width, and height of each of the empty sub volumes.

**L_vector, W_vector,** and **H_vector** are 3 one-dimensional arrays that store the position vector (L,W,H) of the closest corner of every empty sub volume on the pallet with respect to point (0,0,0) of the pallet.

**boxesinWIP** is an integer variable containing the number of boxes in the WIP at the present time.

**box_in_WIP** is an integer array that can have a binary value of either 0 or 1 for each box type. A value of 1 for a particular type means its presence in the WIP.

**actual_WIP_surface** is an integer variable representing the actual WIP surface coverage by the boxes in WIP at that instant.

**max_WIP_surface** is an integer variable representing the maximum WIP surface area coverage by the boxes in WIP at any one time during the palletization process.

**RPT** is a integer variable representing the total palletization time starting from the time the first box arrives at the pick up place to the instant the last box is gone to the pallet.

### 4.1.3 Program Modules:

**main( )**

This main module of function entitled main( ) is more or less a representation of the heuristic in a programming structure. It manages and distributes the work to all the other program functions either directly or indirectly through intermediate function calls.

Interactions with Other Functions:

Called from:

nowhere

Calls:

initialize( )
get_new_box( )
generate_boxes_c1_and_c2( )
can_box_fit_in_subv( ... )
if_box_were_loaded( )
find_best_box_in_WIP( ... )
find_best_box_in_c1_and_WIP( ... )
find_best_box_in_c1_and_c2( ... )
load_best_box_in_WIP( ... )
update_TNSV_because_WIP( ... )
load_boxc1( ... )
update_TNSV_because_c1( ... )
move_boxc1_to_WIP( )
print_final_results( )

**initialize( )**

As the name may deduce, this function is called from main( ) as soon as the program is executed. Its duty is to give appropriate initializing values to the variables and arrays that are globally declared in the program - i.e. variables and arrays available to any function at any time. This function is called only once in the beginning of the program.

Interactions with Other Functions:

Called from:

main( )

Calls:

none

**get_new_box( )**

This function is called whenever a new box is required to enter the system. If a random arrival of boxes is the case, a random number generator is used. On the other hand, if a specific sequence of box types is fed to the system, the function advances in the sequence to get the following box type value which is returned to the calling function.

Interactions with Other Functions:

Called from:

main( )

Calls:

none

**generate_boxes_c1_and_c2( )**

The need for this function comes when the previous boxc1 is gone from the pick up place. The functions simulates boxc2 going from the box identification location to become boxc1 at the pick up place. It also calls function get_new_box() asking for a new box, boxc2 for identification.

Interactions with Other Functions:

Called from:

main( )

Calls:

get_new_box( )

**find_best_box_in_WIP( ... )**

When there is a box or more in the WIP, this function finds the best box in the WIP to be loaded on the pallet, the sub volume where the box would go if it were loaded, the box orientation, its rank, PRVLC, zero-count, and the new TNSV.

Interactions with Other Functions:

Called from:

main( )

Calls:

can_box_fit_in_subv( ... )
if_box_were_loaded( )

**find_best_box_in_c1_and_WIP( ... )**

According to the heuristic rules, the function compares the best box in the WIP with the box at the pick up place, boxc1 for loading priority.

Interactions with Other Functions:
Called from:
main( )
Calls:
none

**find_best_box_in_c1_and_c2( ... )**

Again, this function compares the qualifications of the boxc2, with those of boxc1 which is at the pick up place for loading priority.

Interactions with Other Functions:
Called from:
main( )
Calls:
none

**load_boxc1( ... )**

This function updates the pallet with the newly loaded box, boxc1, its position, and orientation. The number of boxes on the pallet is increased by one and the volume of the box is added to the volumetric pallet utilization.

Interactions with Other Functions:
Called from:
main( )
Calls:
none

**move_boxc1_to_WIP( )**

When boxc1 is to go in the WIP, this function increases the number of boxes in the WIP by one. The number of boxes of boxc1 type is also increased by one. The WIP surface area coverage is updated with the boxc1 base area and if needed, the maximum WIP surface coverage is also updated.

Interactions with Other Functions:

Called from:

main( )

Calls:

print_status( )

**load_best_box_in_WIP( ... )**

When the best box in the WIP is loaded on to the pallet, this function updates the pallet with the type, position, and orientation of the box. The number of boxes on the pallet is increased by one and the volume of the box is added to the volumetric pallet utilization. On the other hand, the number of boxes in the WIP is reduced by one. The WIP surface area coverage is updated accordingly.

Interactions with Other Functions:

Called from:

main( )

Calls:

none

**print_final_results( )**

This function is in charge of printing the "FINAL PALLETIZATION REPORT" as soon as the palletization process is finished. Concerning the pallet, the printed results are: the volumetric pallet utilization, the number of boxes on the pallet, their types, positions, and orientations. As for boxes in the WIP if any, their number is printed as well as the WIP surface coverage and the maximum WIP surface coverage at any one time during the whole process. Furthermore, because palletization time is an important performance criterion, it is also printed with the final results. This function is only called once and it is at the end of the program.

Interactions with Other Functions:

Called from:

main( )

Calls:

none

**print_status( )**

Whenever a box is loaded onto the pallet, whether from the conveyor or from the WIP, or whenever a box on the conveyor is taken to the WIP, this function is called to print the status of the pallet, the WIP, and the conveyor under the title of "PALLETIZATION UPDATE REPORT". The report includes the volumetric pallet utilization, the number of boxes on the pallet, their types, positions, and orientations, the number of boxes present in the WIP and their types and the present WIP surface area coverage. The robotic palletization time is also printed.

Interactions with Other Functions:

Called from:

update_TNSV_because_WIP( ... )
update_TNSV_because_c1( ... )
move_boxc1_to_WIP( )

Calls:

none

**if_box_were_loaded( )**

Given a particular box, this function finds which among the empty sub volumes is best for the box to load into if the box were to load on the pallet.

Interactions with Other Functions:

Called from:

main( )

find_best_box_in_WIP( ... )

Calls:

if_the_box_were_loaded_in_subv( ... )

find_PRVLC( ... )

**if_the_box_were_loaded_in_subv( ... )**

This function is given one of the empty sub volumes to be tested to fit a box. If the box does not fit in the sub volume, the function terminates. Otherwise, it calls best_box_orientation(...) to find the best orientation of the box in the sub volume. Afterwards, it calls making_the_two_subpatterns(...) to partition the remaining volume in the only two possible ways. It then compares them to choose the best of the two. The function then updates the TNSV by adding the newly created sub volumes to the M-1 (TNSV-1) old ones through update_the_first_M_1_subvolumes(...) and add_the_P_new_subvolumes(...) respectively. As a final step, the function calls combine_if_you_can(...) to investigate whether any sub volumes can be combined together or if the partitioning pattern can be redesigned for better performance.

Interactions with Other Functions:

Called from:

if_box_were_loaded( )

Calls:

can_box_fit_in_subv( ... )

best_box_orientation( ... )
making_the_two_subpatterns( ... )
update_the_first_M_1_subvolumes( ... )
add_the_P_new_subvolumes( ... )
find_PRVLC( ... )
combine_if_you_can( ... )

**find_PRVLC( ... )**

This function considers all the empty sub volumes of a partitioning pattern and computes its PRVLC. The zero-count for that partitioning pattern is also computed by this function.

Interactions with Other Functions:

Called from:

if_box_were_loaded( )
if_the_box_were_loaded_in_subv( ... )

Calls:

none

**combine_if_you_can( ... )**

When considering loading a box in a particular sub volume on the pallet, as a result of partitioning the remaining empty space of the sub volume, there is a possibility of finding a new sub volume adjacent to one of the old sub volumes such that the two can be united together thus forming one bigger sub volume which will improve the flexibility of the partitioned space to accommodate more box types. This function takes care of this task and updates the TNSV accordingly. Another aspect of this function is to look into redesigning - repartition - any two sub volumes that are adjacent to each other and are forming an "L shaped" empty space. The function compares the flexibility of the original L shape volume to that of the redesigned one and accordingly decides which of the two patterns to keep.

Interactions with Other Functions:

Called from:

if_the_box_were_loaded_in_subv( ... )

Calls:

find_PRVLC( ... )

**can_box_fit_in_subv( ... )**

This function tests whether a specific box fits into a specific sub volume or not. An integer value of 1 is returned to the calling function if the box fits in the sub volume and a value of 0 if not.

Interactions with Other Functions:

Called from:

main( )
find_best_box_in_WIP( ... )
if_the_box_were_loaded_in_subv( ... )

Calls:

none

**best_box_orientation( ... )**

This function compares the two possible box orientations in a sub volume. The first orientation is that of the box' length // length of the sub volume and its width // to that of the sub volume. The other box orientation is when the box is rotated 90° in the sub volume. The function returns a different value to the calling function depending on which orientation is the best for the box being tested in the sub volume.

Interactions with Other Functions:

Called from:

if_the_box_were_loaded_in_subv( ... )

calls:

none

**making_the_two_subpatterns( ... )**

This function uses the results of best_box_orientation(...) to partition the remaining empty space of the sub volume in two different patterns. Both patterns have the same number of new sub volumes which could be a maximum of three.

Interactions with Other Functions:

Called from:

if_the_box_were_loaded_in_subv( ... )

calls:

none

**update_the_first_M_1_subvolumes( ... )**

Since a box can only be loaded in one sub volume at a time, the remaining TNSV-1 sub volumes remain as they were before the box loading operation. This updating task is performed by the current function. The results of this function are associated with the box being tested. These results are stored in memory for future use in case the box is the one chosen to go on the pallet.

Interactions with Other Functions:

Called from:

if_the_box_were_loaded_in_subv( ... )

calls:

none

**add_the_P_new_subvolumes( ... )**

The new sub volumes created as a result of loading the chosen box into the chosen sub volume are added to the TNSV-1 to update the pallet's configuration and make it ready for the next box loading process. As in the case of the update_the_first_M_1_subvolumes(...) function, the results of this function are also stored in memory in case the box being tested is the one chosen to go on the pallet.

Interactions with Other Functions:

Called from:

if_the_box_were_loaded_in_subv( ... )

calls:

none

**update_TNSV_because_WIP( ... )**

If the chosen box to be loaded on the pallet is the best box in the WIP, this functions extracts from memory the pallet's configuration associated with the best box in the WIP and makes the official update of the TNSV and their new sizes and locations. The function is called by main( ) right after calling load_best_box_in_WIP( ... ).

Interactions with Other Functions:

Called from:

main( )

calls:

print_status( )

**update_TNSV_because_c1( ... )**

As opposed to update_TNSV_because_WIP( ... ), this function extracts from memory the pallet's configuration associated with boxc1 and makes the official update of the TNSV and their new sizes and locations. The function is called by main( ) right after calling load_boxc1( ... ).

Interactions with Other Functions:

Called from:

main( )

calls:

print_status( )

## 4.2 Modified T· ' ○ C++ Program for Model 2

The source code of the computer program of model 2 takes most of the modules of program 1 and modifies them to suit the demand requirements. Several new functions were developed to take care of the building blocks of boxes as well as implementing the two stability rules.

### 4.2.1 Input Data and Variables

The only addition to the set of input variables is an array named **demand_org**.

**demand_org** is a one dimensional integer array that stores the number of boxes of each type that is required to be loaded on the pallet.

### 4.2.2 Output Data and Variables

The only addition to the set of output variables are two integer variables; **stability_rule1** **stability_rule2**.

**stability_rule1** is a counter that counts how many times stability rule 1 was implemented in the palletization process.

**stability_rule2** is another counter that counts how many times stability rule 2 was implemented in the palletization process.

4.2.3 Additional Modules

Because the computer program used to implement the second heuristic is a modified version of that of model 1, some of the existing modules are modified and some new ones are added. These modules called functions are explained below.

**move_boxc1_to_WIP_no_demand( )**

This function replaces the old move_boxc1_to_WIP( ) function that was used in program 1. It is used whenever the demand for boxc1 is zero and it updates the WIP accordingly.

Interactions with Other Functions:

Called from:

main( )

Calls:

print_status( )

**move_boxc1_to_WIP_demand( )**

This function updates the status of the WIP and all demand information for the box type going to the WIP, and increases the WIP surface coverage.

Interactions with Other Functions:

Called from:

main( )

Calls:

print_status( )

**try_making_stable_block_rule2( )**

This function is called to look into the possibility of forming a block based on the requirements of stability rule 2.

Interactions with Other Functions:

Called from:

main( )

Calls:

can_block_fit_anywhere( ... )

load_block_rule2( ... )

**load_block_rule2( ... )**

Whenever the above function succeeds in forming a stable block, this function's role is to identify the loading priority among the boxes making the block. It also updates all information regarding the boxes going on the pallet, the boxes going and leaving the WIP in order to load the complete block in the proper order.

Interactions with Other Functions:

Called from:

main( )

Calls:

generate_boxes_c1_and_c2()

**try_making_stable_block_rule1( )**

This function is similar to try_making_stable_block_rule2( ) but looks into building blocks using stability rule 1

Interactions with Other Functions:

Called from:

main( )

Calls:

can_block_fit_anywhere( ... )

load_block_rule1( ... )

**load_block_rule1( ... )**

Again, this function is similar to load_block_rule2( ... ) but the updates are based on stability rule 1.

Interactions with Other Functions:

Called from:

main( )

Calls:

generate_boxes_c1_and_c2( )

**can_block_fit_anywhere( ... )**

This function tests whether a block fits on the pallet or not. An integer value of 1 is returned to either try_making_stable_block_rule1( ) or try_making_stable_block_rule2() if the box fits and a value of 0 if not.

Interactions with Other Functions:

Called from:

try_making_stable_block_rule1( )

try_making_stable_block_rule2( )

Calls:

none

**update_TNSV_because_block( ... )**

This function updates the remaining sub volumes on the pallet whenever a block is loaded.

Interactions with Other Functions:

Called from:

main( )

Calls:

print_status()

## CHAPTER 5.0 CASE STUDY AND APPLICATIONS

In order to validate the proposed models, numerical examples have been worked out. The pallet used in all the examples is a standard 48" x 40" pallet. Two sets of box types and dimensions are used in the numerical analysis of the proposed models. The maximum allowable height of the pallet is taken as 40". The first set is made of 8 types of boxes and is displayed in table 5.1.

TABLE 5.1: First Set of Box Types and Dimensions

| Box Type | Length (inches) | Width (inches) | Height (inches) | Base Area ( $inches^2$ ) |
|---|---|---|---|---|
| 1 | 12 | 5 | 10 | 60 |
| 2 | 12 | 5 | 20 | 60 |
| 3 | 18 | 10 | 10 | 180 |
| 4 | 18 | 10 | 30 | 180 |
| 5 | 30 | 10 | 10 | 300 |
| 6 | 30 | 10 | 20 | 300 |
| 7 | 25 | 15 | 20 | 375 |
| 8 | 22 | 18 | 30 | 396 |
| Pallet | 48 | 40 | 40 | 1920 |

It is based on actual dimensions of boxes that are commonly used for packaging. The variety of box dimensions was selected knowing that 100% volumetric pallet utilization is not likely to be achieved using the standard pallet dimensions. This is due to the dimensions of the box types not being integral multiples of each other as well as the dimensions of the pallet not being integral multiples of the boxes. The reason why such dimensions were chosen is to demonstrate how the models can be used in real world palletization problems which involve many boxes with divers dimensions. The second set of box dimensions is taken from the literature in order to compare the results of the proposed heuristic models with those found in the literature. This second set is characterized by boxes of integral multiples dimensions.

TABLE 5.2: Second Set of Box Types and Dimensions

| Box Type | Length (inches) | Width (inches) | Height (inches) | Base Area ( $inches^2$ ) |
|---|---|---|---|---|
| 1 | 40 | 24 | 10 | 960 |
| 2 | 40 | 24 | 20 | 960 |
| 3 | 24 | 20 | 10 | 480 |
| 4 | 24 | 20 | 20 | 480 |
| 5 | 24 | 20 | 30 | 480 |
| 6 | 20 | 12 | 20 | 240 |
| 7 | 20 | 12 | 30 | 240 |
| Pallet | 48 | 40 | 40 | 1920 |

These box dimensions were used in the literature so that there would be a high likelihood of good solutions that achieve 100% pallet utilization.

There are two main sections of this chapter. One section is devoted for each of the two models. In either case, whenever VPU, WIP, and the RPT are computed, the following formulae are used.

Volumetric Pallet Utilization is calculated by formula 2.1.1.

$$VPU\ (\%) = \frac{\sum_{i=1}^{N} l_i \times w_i \times h_i}{L_p \times W_p \times H_p} \times 100 \qquad \textbf{(2.1.1)}$$

where;

N = the number of boxes on the pallet

$l_i$, $w_i$, $h_i$ = the dimensions of box i

$L_p$, $W_p$, $H_p$ = the dimensions of the pallet

The WIP index is taken as the maximum surface area covered in the WIP at any point in time during the sequence. As for the robotic palletization time, RPT which is ideally calculated by formula 2.4.4.

$$RPT = N_{C\text{-}P} \times T_{C\text{-}P} + N_{C\text{-}WIP} \times T_{C\text{-}WIP} + N_{WIP\text{-}P} \times T_{WIP\text{-}P} + T_{Idle} \qquad \textbf{(2.4.4)}$$

When actual palletization is carried out, all components of the above formula can be recorded. However, since the current study is only a simulation, $T_{idle}$ is practically impossible to find unless it is assumed. Therefore, a very basic version of equation 2.4.4 which was used in the literature (Yang), (Arghavani) will be used here for comparison with simulation results by both Yang and Arghavani. The assumptions made are the following:

$T_{idle}$ is dropped form the equation.

$T_{C\text{-}P} = T_{C\text{-}WIP} = T_{WIP\text{-}P} = 40$ s.

Therefore, the robotic palletization time is simplified to

$$RPT = 40 \times (N_{C\text{-}P} + N_{C\text{-}WIP} + N_{WIP\text{-}P}) \qquad \textbf{(5.1)}$$

where;

$N_{C\text{-}P}$ = the number of times the robot performs a conveyor/pallet operation,
$N_{C\text{-}WIP}$ = the number of times the robot performs a conveyor/WIP operation,
$N_{WIP\text{-}P}$ = the number of times the robot performs a WIP/pallet operation.

**Model 1 Application:**

Three simulations were conducted to validate the heuristic of mode 1. Each of the three simulations consists of ten different random sequences of boxes.

The first simulations uses boxes from the first set of box types found in table 5.1 above. The types of boxes in the sequences and their order are decided by an equal probability random number generator that is incorporated in the get_new_box( ) function. Simulations 2 and 3 use boxes from the second set of box types found in table 5.2. which was used in the literature by Yang as well as Arghavani. The same sequence inputs used in the literature are also used for simulations 2 and 3. In the following chapter, the results of the two simulations will be compared to those of Yang and Arghavani.

The first part of this chapter is a case study that demonstrates the performance of model 1 through a complete palletization cycle of the first of the ten sequences of simulation 1. Boxes keep coming into the system in random fashion and are loaded on the pallet until the pallet is filled to its maximum.

The box type pattern of the random sequence of boxes coming into the system is the following:

**4 - 5 - 7 - 2 - 6 - 4 - 2 - 8 - 6 - 5 - 1 - 3 - 7 - 6 - 1 - 5**

As the first box enters the system, it is given the title of $Box_{C2}$. The box is identified by the box identification mechanism and is found to be of type 4. Since there is not any box at the pick up place on the conveyor, the conveyor advances transports box type 4 from the Boxc2 position to the Boxc1 position at the end of the conveyor. Meanwhile, a new box enters the system and becomes Boxc2. It is of type 5. Until this moment, there are no boxes in the WIP area, therefore the decision that will be made is based on the comparison between Boxc1 and Boxc2 only.

Each of the two boxes is tested for loading on the pallet. At this stage, the pallet is considered the one and only sub volume available to accommodate any of the two boxes. As a first step in considering the candidature of Boxc1, its ISVLC is found using equation 3.2.5.1.1 for the two possible orientations in the sub volume.

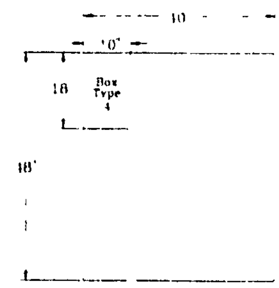


Figure 5.1: Orientation 1 of Box Type 4 (Top View).

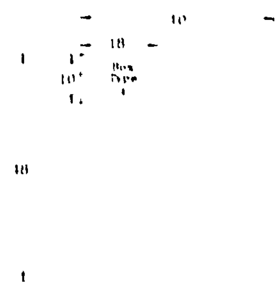


Figure 5.2: Orientation 2 of Box Type 4 (Top View) .

$$ISVLC_{Orient1,\,41} = integ(\frac{L_{SV_1}}{l_{BT_4}}) \times integ(\frac{W_{SV_1}}{w_{BT_4}}) \times integ(\frac{H_{SV_1}}{h_{BT_4}})$$

$$= integ(\frac{48}{18}) \times integ(\frac{40}{10}) \times integ(\frac{40}{30})$$

$$= 2 \times 4 \times 1$$

$$= 8 \text{ boxes of type } 4$$

$$ISVLC_{Orient2,\,41} = integ(\frac{L_{SV_1}}{w_{BT_4}}) \times integ(\frac{W_{SV_1}}{l_{BT_4}}) \times integ(\frac{H_{SV_1}}{h_{BT_4}})$$

$$= integ(\frac{48}{10}) \times integ(\frac{40}{18}) \times integ(\frac{40}{30})$$

$$= 4 \times 2 \times 1$$

$$= 8 \text{ boxes of type } 4$$

Ideally, the orientation leading to the largest ISVLC is chosen as the best orientation for the box being tested. However, in cases such as the one at hand where the two box orientations result in equal values of ISVLCs, the heuristic gives priority to the orientation where the length of the box is parallel to that of the sub volume (in this case, the pallet). The step that follows consists of finding the best way to partition the remaining empty space of the sub volume (or pallet). Since there are two possible partitioning solutions for the sub volume, they are compared according to the three priority levels in the following order: 2, 3, then 1.

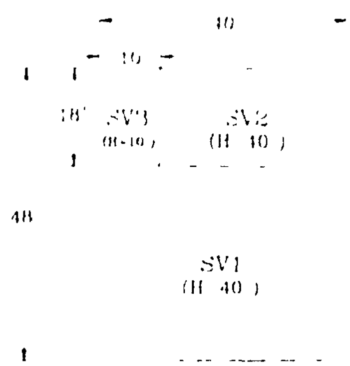



Figure 5.3: Partitioning Patt.1 of Box Type 4 (Top View)

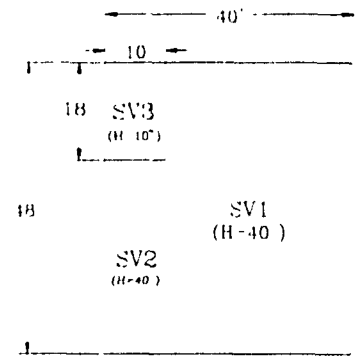



Figure 5.4: Partitioning Patt.2 of Box Type 4 (Top View)

The first comparison between the two partitioning patterns is between their respective PRVLCs. Based on the theory discussed in earlier chapters, tables 5.3 and 5.4 summarize the process of calculating the PRVLC of patterns 1 and 2 respectively. It is clear that partitioning pattern 2 has better performance since the value of its PRVLC is larger than that of partitioning pattern 1.

Table 5.3: PRVLC Sample Calculation for Partitioning Pattern 1 (Box Type 4)

| Box Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | PRVLC ($\sum$MCBT) |
|---|---|---|---|---|---|---|---|---|---|
| SV1 | 72 | 36 | 24 | 6 | 16 | 8 | 4 | 2 | |
| SV2 | 24 | 12 | 12 | 3 | 4 | 2 | 2 | 1 | |
| SV3 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | |
| MCBT ($\sum$ ISVLC) | 98 | 48 | 37 | 9 | 20 | 10 | 6 | 3 | 231 |

Table 5.4: PRVLC Sample Calculation for Partitioning Pattern 2 (Box Type 4)

| Box Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | PRVLC ($\sum$MCBT) |
|---|---|---|---|---|---|---|---|---|---|
| SV1 | 96 | 48 | 24 | 6 | 16 | 8 | 6 | 2 | |
| SV2 | 16 | 8 | 4 | 1 | 4 | 2 | 0 | 0 | |
| SV3 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | |
| MCBT ($\sum$ ISVLC) | 114 | 56 | 29 | 7 | 20 | 10 | 6 | 2 | 244 |

Consequently, the way the remaining empty space of the pallet is partitioned according to the pattern found in figure 5.4. Furthermore, there is no need to compare the Total Zero Counts, TZC of the two partitioning patterns since the decision was possible at the level before. However, let's assume that $PRVLC_{\text{Partitioned Pattern 1}} = PRVLC_{\text{Partitioned Pattern 2}}$, then, the TZC of the two patterns would need to be compared. Tables 5.5 and 5.6 below summarize the calculations of their TZCs respectively. Whenever any of the 3 sub volumes (SV1, SV2, or SV3) can not accommodate a particular box type, a ZC value of 1 is written in the corresponding position in the table. Otherwise, a dash " - " is put instead. Looking at the two TZC values, (6) for pattern 1 and (8) for pattern 2, priority would have been given to pattern 1 for box type 4 contrary to the decision based on priority level 2.

Table 5.5: Zero Count Sample Calculation for Partitioning Pattern 1 (Box Type 4)

| **Box Type** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | |
|---|---|---|---|---|---|---|---|---|---|
| **SV1** | - | - | - | - | - | - | - | - | **TZC ($\sum$ZCBT)** |
| **SV2** | - | - | - | - | - | - | - | - | |
| **SV3** | - | 1 | - | 1 | 1 | 1 | 1 | 1 | |
| **ZCBT=$\sum$ZC** | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 6 |

Table 5.6: Zero Count Sample Calculation for Partitioning Pattern 2 (Box Type 4)

| **Box Type** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | |
|---|---|---|---|---|---|---|---|---|---|
| **SV1** | - | - | - | - | - | - | - | - | **TZC ($\sum$ZCBT)** |
| **SV2** | - | - | - | - | - | - | 1 | 1 | |
| **SV3** | - | 1 | - | 1 | 1 | 1 | 1 | 1 | |
| **ZCBT=$\sum$ZC** | 0 | 1 | 0 | 1 | 1 | 1 | 2 | 2 | 8 |

Now that all analysis related to box type 4 (actual Boxc1) is finished, the same is repeated for box type 5 (actual Boxc2). To decide which is better than the other, according to the heuristic rules for comparing two boxes on the conveyor, priority level is used. Since $TNSV_{Boxc1}$ (3) $\leq TNSV_{Boxc2}$ (3), it is decided that Boxc1 goes on the pallet and not to the WIP.

Now that Boxc1 is gone to the pallet, Boxc2 (type 5) proceeds to the end of the conveyor and becomes Boxc1. A new box enters the system. It is of type 7. Priority level 1 reveals that it is profitable to load Boxc1 since the new TNSV (=3) will not be increased as opposed to loading Boxc2 (TNSV=5). Type 7 box becomes Boxc1 and the new box of type 2 is Boxc2. Now TNSV of Boxc2 is 5 which is equal to that of Boxc1, therefore, Boxc1 of type 7 is loaded onto the pallet. Now comparing box type 2 to the next box in the sequence, box type 6. Priority is given to box type 6 because its TNSV = 4 is less than that of box type 2 which is 6. Therefore type 2 box with base area = 60 $inches^2$ is moved to the WIP and box type 6 advances to become the new Boxc1. In the meantime, new box of type 4 becomes Boxc2. Boxc1 (type 6) still remains better than the box in the WIP (type 2) and is also better than Boxc2 (type 4) through priority level 1. Thus it is carried to the pallet.

Fortunately for the box of type 2 in the WIP, its TNSV being 3 gives it priority over the box on the conveyor of type 4 whose TNSV = 6. Accordingly, box type 2 leaves the WIP which becomes empty again. Since a box from the WIP was loaded on the pallet, no new boxes are allowed to enter the system because both Boxc1 and Boxc2 are still in position on the conveyor. The WIP does not stay empty for long since Boxc1 (type 4) is taken there because Boxc2 (type 2) gives smaller TNSV (= 4) as opposed to a value of 7 for Boxc1.

The WIP so far was used twice and the maximum WIP area at any one time is equals to 180 inches$^2$ corresponding to the base area of box type 4 which is currently in the WIP. Boxc2 (type 8) as well as type 4 box in the WIP increase the TNSV from 5 to 7 while Boxc1 (type 2) reduces it from 5 to 4. Thus Boxc1 exceeds the other two boxes by far in terms of performance and is loaded onto the pallet. At the next stage, for the first time, the second priority level is used to break the tie between Boxc1 (type 8) and box type 4 in the WIP at the first comparison level. Each of the two if loaded, will result in a TNSV = 6 However, box type 4 gives a PRVLC = 119 which is larger than the value of 91 resulting from loading box type 8. Therefore, box type 4 is moved from the WIP to the pallet.

So far, nine boxes have entered the system in the following order: 4 - 5 - 7 - 2 - 6 - 4 - 2 - 8 - 6. Seven of these boxes are presently on the pallet producing a volumetric pallet utilization of 38.67%, a robotic palletization time of 360 seconds, a maximum WIP of 180 inches$^2$ and is presently empty. The palletization process continues in the same fashion until the remaining sub volumes on the pallet totalling 7 can not accommodate any of the 8 predefined box types. At that stage, there are 16 boxes on the pallet which is 91.25% filled in a palletization time of 760 seconds. For better visualization of the pallet loading process, A step by step pallet build up with the 16 boxes is shown in figure 5.5.

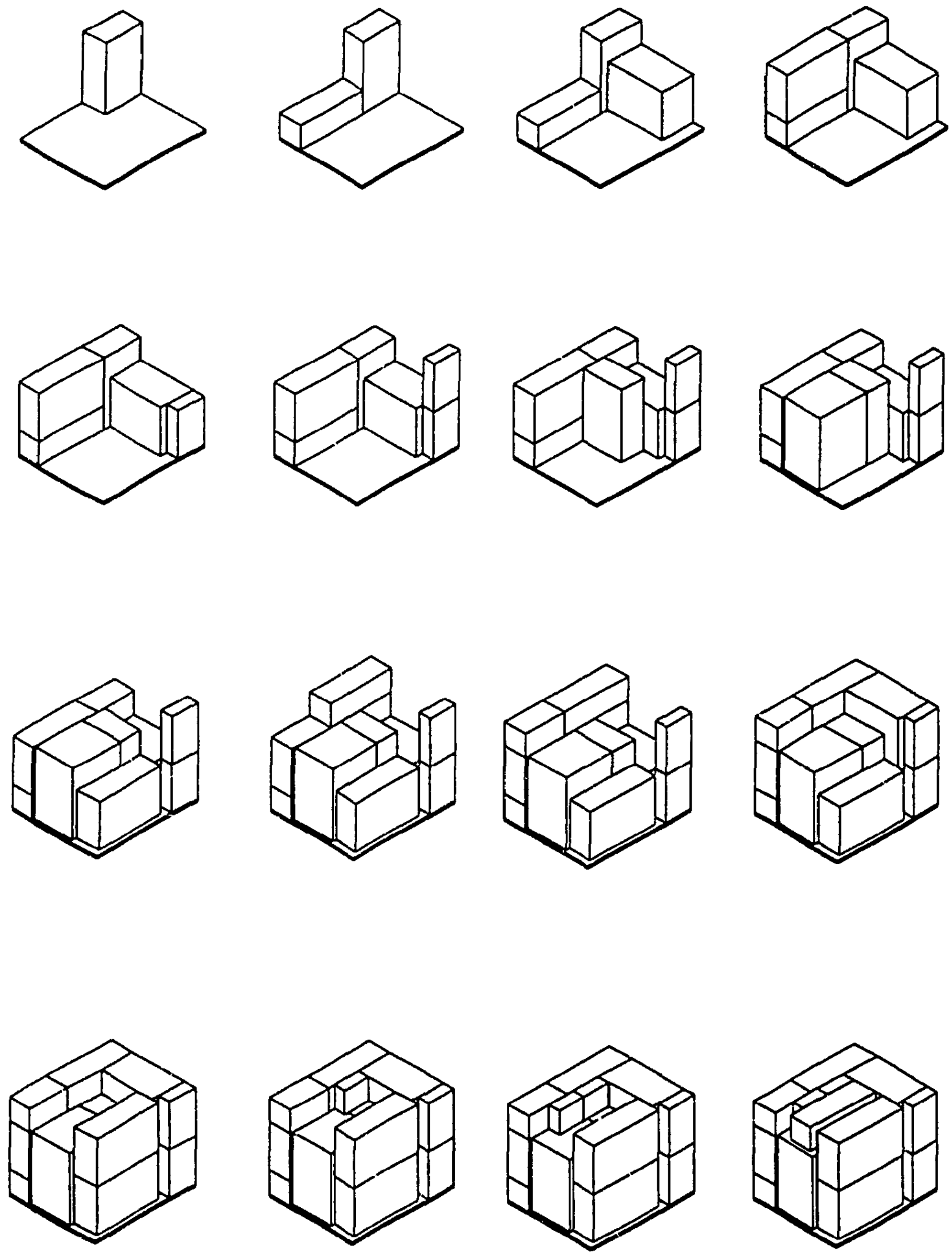


Figure 5.5: Step by Step Pallet Build Up (Seq. 1, Sim. 1, Model 1)

The complete palletization process including moving boxes to and from the conveyor as well as the WIP is summarized in table 5.7.

**Table 5.7: Summary of Stage by Stage Palletization Process (Seq. 1, Sim.1, Model 1)**

| Stage | Box c2 | Box c1 | Box WIP | Decision | Comparison Index | VPU (%) | WIP (in$^2$) | RPT (s) |
|---|---|---|---|---|---|---|---|---|
| **1** | 4 | - | - | Boxc2 ☞ Boxc1 | - | - | - | - |
| **2** | 5 | 4 | - | Boxc1 ☞ Pallet | TNSVc1(3) = TNSVc2(3) | 7.03 | 0 | 40 |
| **3** | 7 | 5 | - | " " | TNSVc1(3) < TNSVc2(5) | 10.94 | 0 | 80 |
| **4** | 2 | 7 | - | " " | TNSVc1(5) = TNSVc2(5) | 20.70 | 0 | 120 |
| **5** | 6 | 2 | - | Boxc1 ☞ WIP | TNSVc1(4) > TNSVc2(5) | " " | 60 | 160 |
| **6** | 4 | 6 | 2 | Boxc1 ☞ Pallet | TNSVc1(4) < TNSVc2(5) | 28.52 | 60 | 200 |
| **7** | 2 | 4 | 2 | $Box_{WIP}$ ☞ Pallet | TNSVc1(6) > $TNSV_{WIP}$(5) | 30.08 | 0 | 240 |
| **8** | " | " | - | Boxc1 ☞ WIP | TNSVc1(7) > TNSVc2(4) | " " | 180 | 280 |
| **9** | 8 | 2 | 4 | Boxc1 ☞ Pallet | TNSVc1(4) < TNSVc2(7) | 31.64 | 180 | 320 |
| **10** | 6 | 8 | 4 | $Box_{WIP}$ ☞ Pallet | $PRVLC_{c1}$(91)<$PRVLC_{WIP}$(119) | 38.67 | 0 | 360 |
| **11** | " | " | - | Boxc1 ☞ Pallet | TNSVc1(6) < TNSVc2(7) | 54.14 | 0 | 400 |
| **12** | 5 | 6 | - | " " | Base $Dim._{c1}$ = Base $Dim._{c2}$ | 61.95 | 0 | 440 |
| **13** | 1 | 5 | - | " " | TNSVc1(8) < TNSVc2(9) | 65.86 | 0 | 480 |
| **14** | 3 | 1 | - | Boxc1 ☞ WIP | TNSVc1(9) > TNSVc2(7) | " | 0 | 520 |
| **15** | 7 | 3 | 1 | Boxc1 ☞ Pallet | TNSVc1(7) = TNSVc2(7) | 68.20 | 60 | 560 |
| **16** | 6 | 7 | 1 | " " | TNSVc1(6) = TNSVc2(6) | 77.97 | " | 600 |
| **17** | 1 | 6 | 1 | " " | TNSVc1(5) < TNSVc2(7) | 85.78 | " | 640 |
| **18** | 5 | 1 | 1 | " " | Boxc1 = Box WIP | 86.59 | " | 680 |
| **19** | - | 5 | 1 | $Box_{WIP}$ ☞ Pallet | TNSVc1(7) > $TNSV_{WIP}$(6) | 87.34 | 0 | 720 |
| **20** | - | 5 | - | Boxc1 ☞ Pallet | **TNSV=7, PRVLC=0, $TZC_{max}$ is reached (=7x8)** | **91.25** | 0 | 760 |

Since table 5.7 in the previous page shows only the index based on which the decision is made at every stage, table 5.8 puts together all indices of each stage. These indices are grouped into two main categories, decision making indices and physical performance indices. Each group comprises three indices. The indices of the first group are the total number of sub volumes remaining at the end of each stage, TNSV, the load capacity of the partitioned remaining volume at the end of each stage, PRVLC, and the total zero count of the pattern of each stage, TZC. The indices of the second group are the volumetric pallet utilization in percentage, VPU, the work-in-process surface coverage, WIP, and the robotic palletization time, RPT at the end of each of the 16 pallet loading stages of the palletization process.

**Table 5.8: Performance Criteria and Indices (Seq. 1, Sim. 1, Table 1)**
**Sequence : 4 - 5 - 7 - 2 - 6 - 4 - 2 - 8 - 6 - 5 - 1 - 3 - 7 - 6 - 1 - 5**

| | Pallet Loading Stages | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **TNSV** | 3 | 3 | 5 | 4 | 5 | 4 | 6 | 6 | 8 | 8 | 7 | 6 | 5 | 6 | 6 | 7 |
| **PRVLC** | 244 | 233 | 176 | 160 | 157 | 154 | 119 | 89 | 59 | 51 | 48 | 27 | 10 | 9 | 8 | 0 |
| **TZC** | 8 | 8 | 18 | 15 | 21 | 15 | 29 | 32 | 49 | 50 | 44 | 40 | 37 | 44 | 45 | 56 |
| **VPU** | 7.03 | 10.94 | 20.7 | 28.52 | 30.08 | 31.64 | 38.67 | 54.14 | 61.95 | 65.86 | 68.2 | 77.97 | 85.78 | 86.56 | 87.34 | 91.25 |
| **WIP** | 0 | 0 | 0 | 60 | 0 | 180 | 0 | 0 | 0 | 0 | 60 | 60 | 60 | 60 | 0 | 0 |
| **RPT** | 40 | 80 | 120 | 200 | 240 | 320 | 360 | 400 | 440 | 480 | 560 | 600 | 640 | 680 | 720 | 760 |

The numbers found in table 5.8 are used in the analysis of results chapter to graphically represent the behaviour of and relationships among the indices. The actual Turbo C++ output file for the sequence containing all values of table 5.8 is found in appendix A.

The results of the ten sequences of the first simulation of model 1 are tabulated in table 5.9 while those of the second and third simulations are found in table 5.10 and table 5.11 respectively.

**Table 5.9: Simulation 1: Performance criteria Output (Model 1)**

| Seq # | Sequence | VPU (%) | TNSV Remain'g | WIP Remain'g | Max.WIP ( inches$^2$ ) | RPT (s) |
|---|---|---|---|---|---|---|
| 1 | 4-5-7-2-6-4-2-8-6-5-1-3-7-6-1-5 | 91.25 | 7 | 0 | 180 | 760 |
| 2 | 6-4-5-8-2-8-4-6-4-1-3-5-1-5-3-2-5-2-2 | 87.92 | 7 | 300 | 600 | 880 |
| 3 | 3-2-6-5-4-2-7-4-3-7-2-1-5-3-6-2-6-3-2-3-1-2-2 | 89.06 | 4 | 0 | 360 | 1080 |
| 4 | 5-1-6-7-3-1-5-6-7-3-2-4-6-2-7-2-3-3-5-1-4-2-2-1 | 94.92 | 4 | 300 | 375 | 1120 |
| 5 | 6-2-8-4-6-4-8-1-3-5-7-6-1-4-2-3-7-5-2-3-3 | 92.03 | 5 | 876 | 876 | 960 |
| 6 | 4-5-7-6-7-2-2-4-8-6-1-5-3-1-6-5 | 91.25 | 7 | 0 | 375 | 760 |
| 7 | 8-8-7-4-4-1-2-5-2-2-2-3-2-7-3-1-5-1 | 87.19 | 8 | 0 | 60 | 760 |
| 8 | 4-5-7-2-2-1-7-7-2-2-1-3-6-5-1-7-2-7-7-2-1 | 96.09 | 1 | 0 | 375 | 920 |
| 9 | 2-2-1-8-8-6-1-2-6-3-3-4-3-8-5-3-3-1 | 91.72 | 4 | 0 | 300 | 760 |
| 10 | 4-3-7-7-2-1-6-2-4-1-1-2-4-6-5-2-3-2-1-3-2-5-6 | 91.41 | 4 | 0 | 300 | 1000 |

**Table 5.10: Simulation 2: Performance criteria Output (Model 1)**

| Seq # | Sequence* | VPU (%) | TNSV Remain'g | WIP Remain'g | Max.WIP ( $inches^2$ ) | RPT (s) |
|---|---|---|---|---|---|---|
| 1 | 2-2-1-4-1-4 | 100 | 0 | 0 | 480 | 280 |
| 2 | 2-2-4-4-1-1 | 100 | 0 | 0 | 0 | 240 |
| 3 | 2-1-4-2-1-4 | 100 | 0 | 0 | 480 | 280 |
| 4 | 1-2-1-4-4-2 | 100 | 0 | 0 | 0 | 240 |
| 5 | 1-4-4-1-2-2 | 100 | 0 | 0 | 0 | 240 |
| 6 | 1-4-2-2-4-1 | 100 | 0 | 0 | 480 | 280 |
| 7 | 4-1-1-4-2-2 | 100 | 0 | 0 | 480 | 280 |
| 8 | 4-2-2-1-4-1 | 100 | 0 | 0 | 480 | 280 |
| 9 | 2-1-4-4-1-2 | 100 | 0 | 0 | 0 | 240 |
| 10 | 4-1-1-2-2-4 | 100 | 0 | 0 | 0 | 240 |

* The 10 sequences of the simulation are those used by Arghavani.

**Table 5.11: Simulation 3: Performance criteria Output (Model 1)**

| Seq # | Sequence** | VPU (%) | TNSV Remain'g | WIP Remain'g | Max.WIP ( $inches^2$ ) | RPT (s) |
|---|---|---|---|---|---|---|
| 1 | 1-3-2-3-5-1-5 | 100 | 0 | 0 | 480 | 360 |
| 2 | 5-5-3-3-2-1-1 | 100 | 0 | 0 | 0 | 280 |
| 3 | 5-5-1-3-1-2-3 | 100 | 0 | 0 | 480 | 320 |
| 4 | 5-5-1-3-3-1-2 | 100 | 0 | 0 | 0 | 280 |
| 5 | 1-3-1-2-5-5-3 | 100 | 0 | 0 | 480 | 320 |
| 6 | 1-5-1-2-5-3-3 | 100 | 0 | 0 | 960 | 320 |
| 7 | 2-3-5-5-1-3-1 | 100 | 0 | 0 | 0 | 280 |
| 8 | 3-3-2-5-1-5-1 | 100 | 0 | 0 | 960 | 320 |
| 9 | 3-3-2-1-1-5-5 | 100 | 0 | 0 | 0 | 280 |
| 10 | 3-3-5-1-1-2-5 | 100 | 0 | 0 | 0 | 280 |

** The 1( :quences of the simulation are those used by Yang.

**Model 2 Application:**

As in the case of model 1, three simulations were conducted to validate the heuristic of model 2. Each of the three simulations consists of ten different random sequences of boxes. The sequences contain larger quantities of boxes so as to cover the demand of the box types and show how boxes with zero demand are moved permanently to the WIP. The results of the three simulations are tabulated in their respective tables showing the number of times each of the two stability rules was applied during each sequence.
Table 5.12 contains the demand of each box type for the first simulation. The demands having equal probabilities are randomly generated while considering the maximum pallet volume.

**Table 5.12: Box Demand Requirements (Simulation 1, Model 2)**

| Seq # | Box Type | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 2 | 2 | 1 | 2 | 5 | 2 | 2 | 1 |
| 2 | 4 | 1 | 2 | 3 | 4 | 2 | 0 | 2 |
| 3 | 2 | 7 | 5 | 2 | 2 | 3 | 2 | 0 |
| 4 | 4 | 5 | 4 | 2 | 2 | 3 | 3 | 0 |
| 5 | 2 | 4 | 3 | 3 | 0 | 3 | 1 | 1 |
| 6 | 2 | 2 | 3 | 1 | 4 | 2 | 2 | 1 |
| 7 | 3 | 5 | 2 | 2 | 2 | 0 | 2 | 2 |
| 8 | 4 | 6 | 1 | 1 | 2 | 1 | 6 | 0 |
| 9 | 3 | 3 | 6 | 2 | 0 | 2 | 0 | 2 |
| 10 | 4 | 6 | 4 | 3 | 2 | 3 | 2 | 0 |

Table 5.13: Simulation 1: Performance criteria Output (Model 2)

| # | Sequence | Stability | | Demand Satisfac'n (%) | VPU (%) | TNSV | WIP Remain'g ($inch^2$) | Max. WIP ($inch^2$) | RPT (s) |
|---|---|---|---|---|---|---|---|---|---|
| | | Rule 1 | Rule 2 | | | | | | |
| 1 | 4-5-7-2-6-4-2-8-6-5-1-3-7-8-2-5-1-5-3-5 | 1 | 0 | 100 | 91.25 | 7 | 636 | 696 | 1000 |
| 2 | 6-7-4-5-8-2-8-8-4-6-4-1-4-3-5-1-4-5-3-4-2-5-1-1 | 0 | 0 | 96.5 | 84.84 | 7 | 1671 | 1671 | 960 |
| 3 | 3-2-6-5-4-2-7-4-3-7-2-1-5-3-6-2-6-3-2-3-1-2-2 | 4 | 0 | 98.25 | 87.5 | 9 | 60 | 360 | 1120 |
| 4 | 5-8-1-6-7-3-1-5-6-8-7-3-2-4-6-8-2-7-2-3-7-3-5-1-4-2-2-1 | 2 | 0 | 92.59 | 87.89 | 7 | 2043 | 2043 | 1320 |
| 5 | 6-2-8-4-6-4-7-3-1-3-6-4-1-2-3-1-2-5-2 | 1 | 0 | 90.77 | 76.8 | 8 | 840 | 840 | 920 |
| 6 | 8-5-4-5-6-5-7-5-6-7-3-2-2-4-8-6-3-1-5-3-1 | 2 | 0 | 93.15 | 85.0 | 6 | 1356 | 1356 | 960 |
| 7 | 8-8-7-4-6-4-4-1-2-5-2-2-8-2-3-2-6-7-8-3-1-5-1 | 1 | 0 | 100 | 87.19 | 8 | 1572 | 1572 | 960 |
| 8 | 4-5-7-2-4-2-1-7-8-7-2-2-1-3-3-6-5-1-3-7-2-7-7-2-1-5 | 3 | 0 | 100 | 96.09 | 1 | 936 | 1431 | 1160 |
| 9 | 7-2-2-1-8-5-8-6-1-2-6-3-3-4-3-6-8-3-3-1-4-3 | 4 | 0 | 89.1 | 81.72 | 6 | 1371 | 1551 | 1000 |
| 10 | 4-3-7-7-7-2-1-6-2-4-1-8-1-2-4-4-6-3-5-2-3-2-1-3-2-5-6 | 4 | 0 | 89.74 | 82.03 | 7 | 1551 | 1551 | 1240 |

**Table 5.14: Simulation 2: Performance criteria Output (Model 2)**

| Seq # | Sequence | Stability | | VPU (%) | TNSV | WIP Remain'g (inch²) | Max. WIP (inch²) | RPT (s) |
|---|---|---|---|---|---|---|---|---|
| | | Rule 1 | Rule 2 | | | | | |
| 1 | 7-2-2-1-5-4-1-4 | 1 | 0 | 100 | 0 | 720 | 1200 | 400 |
| 2 | 2-2-4-4-7-1-3-1 | 0 | 0 | 100 | 0 | 720 | 720 | 320 |
| 3 | 2-1-6-4-5-2-1-4 | 1 | 0 | 100 | 0 | 720 | 1680 | 400 |
| 4 | 1-2-1-1-3-4-4-2 | 1 | 0 | 100 | 0 | 1440 | 1440 | 320 |
| 5 | 1-4-4-1-3-6-2-2 | 0 | 0 | 100 | 0 | 720 | 720 | 320 |
| 6 | 1-1-4-2-7-2-4-1 | 0 | 1 | 100 | 0 | 1200 | 120 | 400 |
| 7 | 4-1-3-5-1-4-2-2 | 0 | 1 | 100 | 0 | 960 | 1920 | 400 |
| 8 | 4-6-6-2-2-1-4-1 | 0 | 0 | 100 | 0 | 480 | 1440 | 360 |
| 9 | 7-2-1-1-4-4-1-2 | 1 | 0 | 100 | 0 | 1200 | 1200 | 360 |
| 10 | 4-1-6-7-1-2-2-4 | 1 | 1 | 100 | 0 | 480 | 1440 | 400 |

**Table 5.15: Simulation 3: Performance criteria Output (Model 2)**

| Seq # | Sequence | Stability | | VPU (%) | TNSV | WIP Remain'g (inch²) | Max. WIP (inch²) | RPT (s) |
|---|---|---|---|---|---|---|---|---|
| | | Rule 1 | Rule 2 | | | | | |
| 1 | 1-2-3-2-3-7-5-1-5 | 1 | 1 | 100 | 0 | 1200 | 2160 | 440 |
| 2 | 5-5-5-3-3-2-1-1-1 | 1 | 0 | 100 | 0 | 480 | 1440 | 360 |
| 3 | 5-3-5-1-3-6-1-2-3 | 2 | 0 | 100 | 0 | 720 | 1680 | 440 |
| 4 | 4-5-5-1-3-3-1-2-6 | 1 | 0 | 100 | 0 | 720 | 720 | 360 |
| 5 | 1-5-3-1-2-4-5-5-3 | 2 | 0 | 100 | 0 | 960 | 960 | 400 |
| 6 | 1-5-1-2-5-5-3-3-3 | 2 | 0 | 100 | 0 | 480 | 960 | 360 |
| 7 | 2-3-4-5-5-1-4-3-1 | 1 | 0 | 100 | 0 | 960 | 960 | 360 |
| 8 | 7-3-3-2-7-5-1-5-1 | 0 | 1 | 100 | 0 | 480 | 1440 | 440 |
| 9 | 3-3-3-2-1-1-1-5-5 | 1 | 0 | 100 | 0 | 1440 | 1440 | 400 |
| 10 | 3-7-3-5-1-4-1-2-5 | 2 | 0 | 100 | 0 | 720 | 720 | 360 |

# CHAPTER 6.0 ANALYSIS OF RESULTS

This chapter comprises two main parts: the first part discusses and compares the results of the two proposed models. The second part of the chapter compares the two models with models of the literature.

## 6.1 The Results of the 2 Models

The three simulations of the proposed model 1 show that model 1 can achieve near optimal solutions to the 3D palletization problems. They show the effect of random box arrival on the palletization performance. More specifically on the four main performance criteria; VPU, WIP, RPT, and pallet stability. None of the ten sequences of simulation 1 produced 100% VPU. Table 5.9 shows that the VPU ranges from 87.19% to 96.06%. This is due to using box dimensions (table 5.1) with which it is very unlikely to obtain a 100% perfect cube. Figures 6.1 and 6.2 are graphical representations of the data found in table 5.8 which tabulates the change in all the decision making as well as the performance indices when the first box sequence of the simulation was run step by step in the case study.

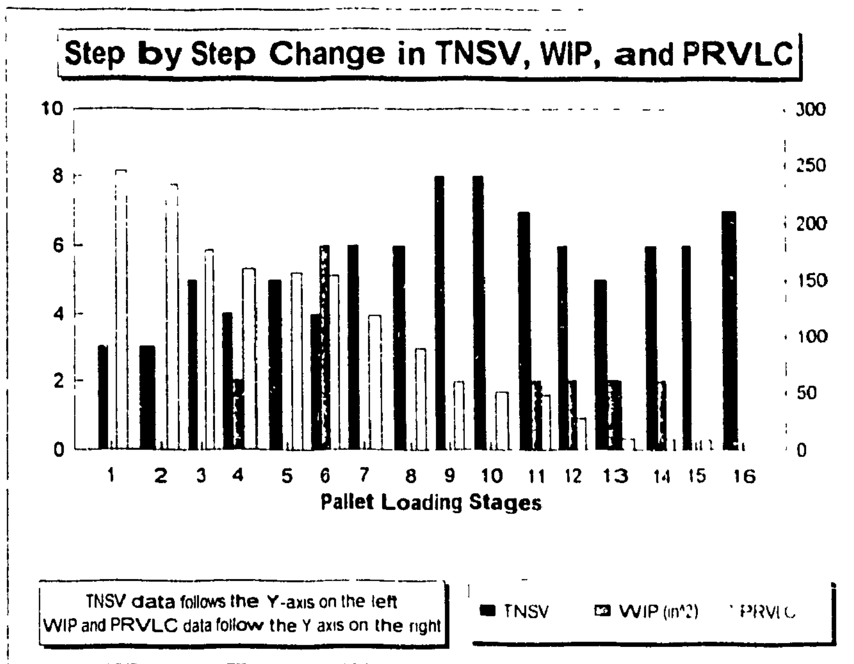



**Figure 6.1: Graphical Representation of TNSV, WIP, and PRVLC**
**(Seq. 1, Sim. 1, Model 1)**

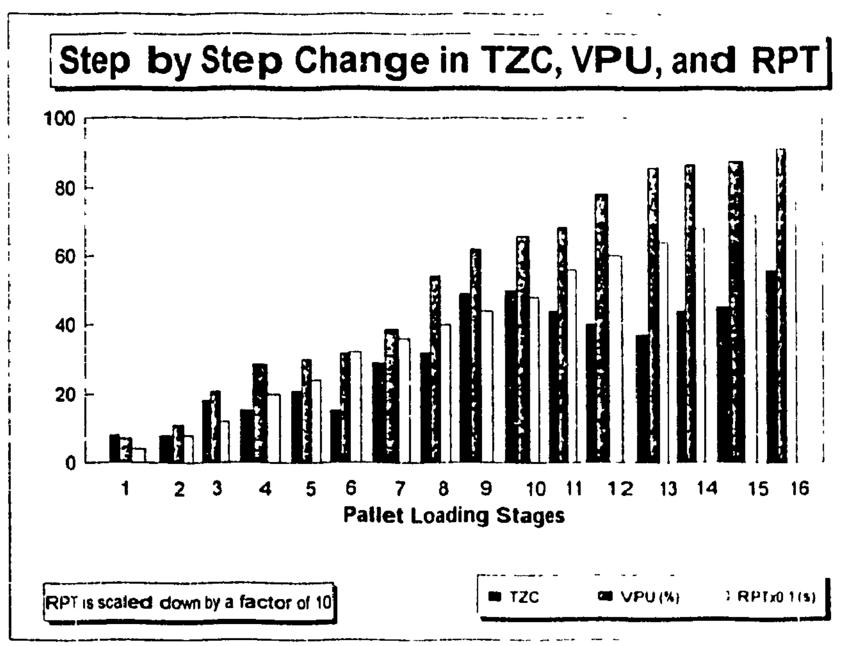



**Figure 6.2: Graphical Representation of TZC, VPU, and RPT**
**(Seq. 1, Sim. 1, Model 1)**

Because 100% VPU is not achieved. some empty spaces remain here and there on the pallet. The TNSV fluctuates up and down as boxes are loaded on the pallet until it settles at the value of 7. Then all remaining sub volumes are too small for any of the box types. Regarding the PRVLC, its value starts at a maximum value when the pallet is empty and as the pallet gets filled, the relative sizes of the remaining sub volumes become smaller and thus the PRVLC decreases to reach zero at the end of the palletization stage.

Looking at figure 6.2. it is noticed that the TZC behaves oppositely to the PRVLC. It starts low and increases to its maximum. This is also explained by the fact that the more empty is the pallet. its sub volumes can accommodate more variety of box types, which means a lower TZC.

Simulations 2 and 3 both use box dimensions from table 5.2. All the ten sequences of each of the two simulations obtain 100% VPU. Thus the final TNSV is always 0 as opposed to values ranging from 1 to 8 sub volumes in the case of simulation 1. Therefore the change in TNSV depends greatly on the set of box dimensions that is used in the palletization process. Figures 6.3 and 6.4 show two typical progression of the TNSV when the two sets of box dimensions are used.

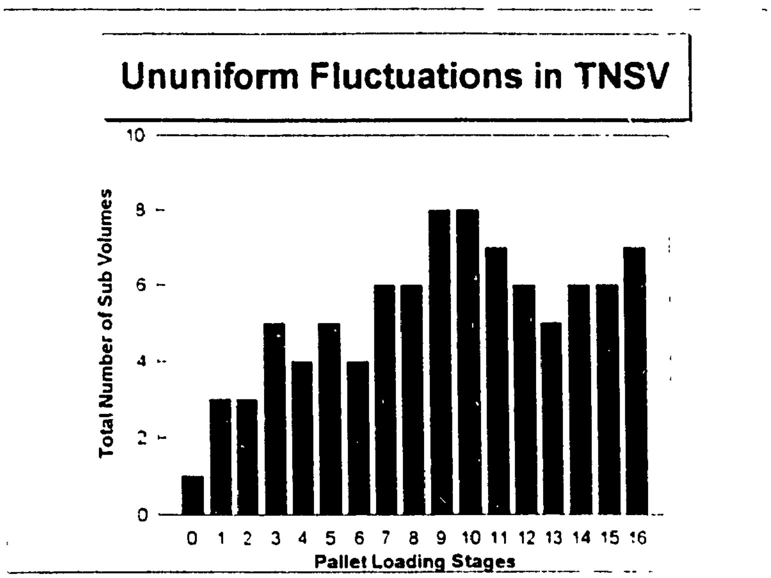



**Figure 6.3: TNSV Life Cycle**
(Seq. 1, Sim. 1, Model 1)

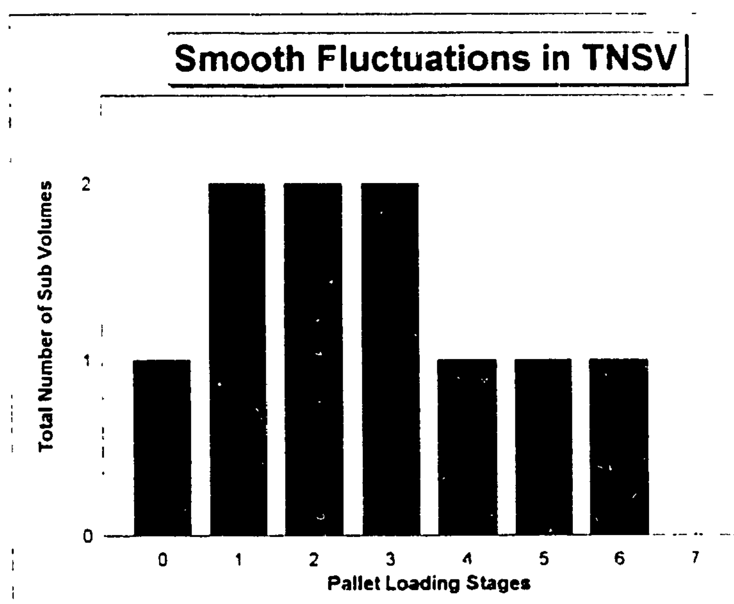



**Figure 6.4: TNSV Life Cycle**
(Seq. 6, Sim. 3, Model 1)

Figure 6.4 shows how its TNSV starts off at 1, increases then decreases until it reaches zero when VPU = 100%. On the other hand, figure 6.3 shows that TNSV increases, fluctuates, and finally remains at a non zero number that is 7.

Although model 1 did not address the issue of pallet load stability, looking at the final pallet layout of sequence 6 from simulation 3 in figure 6.5, one can see that stability rule 1 was indirectly applied when the box of bigger height was loaded on top of the box with same base dimensions but with smaller height.

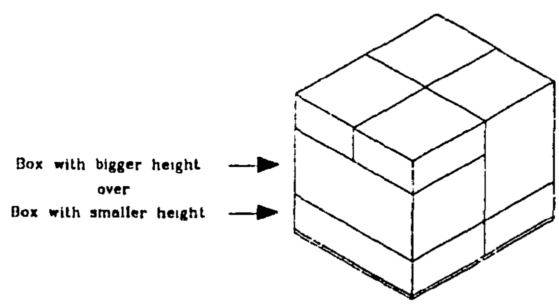



**Figure 6.5: Naturally implemented Stability (Seq. 6, Sim. 3, Model 1)**

Model 2 was developed to tackle the loading stability problem as well as considering demand requirements of the boxes that should be loaded on the pallet. When looking at the results of the 3 simulations of model 2 (tables 5.13, 5.14, and 5.15), the first thing that grabs one's attention is the higher values of RPT and maximum WIP. The WIP remaining after the pallet is filled to its maximum is mainly the boxes that arrived at the system and were not needed. In other words, they are boxes whose demand requirements were zero at the time of their arrival into the system. The results of simulation 1 show that stability rule 1 was implemented in nine of the ten sequences once to four times in some sequences. It is not surprising to find out that stability rule 2 was never used in simulation 1. This is because none of the box dimensions in the set used in the simulation match with each other to enable the formation of blocks. On the other hand, the set of harmonious box dimensions enabled simulations 2 and 3 to used the full strength of both stability rules. However, stability rule 1 was implemented more often than rule 2. This is because in order to implement rule 1, only two boxes are required whereas rule 2 needs three boxes to construct a block. Furthermore, rule 2 can only be applied whenever there is at least one box in the WIP as there can be a maximum of two boxes on the conveyor, Boxc1 and Boxc2.

The demand requirements of only three out of the ten sequences of simulation 1 were fully satisfied while the box demands of the remaining seven sequences were satisfied in the range of 89.1% to 98.25%. There are two possible causes for such results. The first possibility is unavoidable. It is the random arrival of the boxes. Since the loading decision is based on Boxc1, Boxc2, and the boxes in the WIP only, and boxes enter the system in random order, the only way to deal with the unknown future is to use the little knowledge about the past and the present in making the decision as much effective as possible. This is the basic idea behind the decision making indices and priority levels discussed in previous chapters. The second possible cause of unsatisfied box demands is the application of the stability rules which may have side effects of limiting the partitioning of the empty space as a result of loading blocks onto the pallet. The problem of unfulfilled box demands does not appear in simulations 2 and 3 as all of their sequences achieve 100% demand satisfaction. Again, it is a result of using box sizes that have higher flexibility to load on the pallet.

To visualize the effect of implementing the stability rules in model 2, the pallet load layout of sequence 6, of simulation 3 in figure 6.6 can be compared with its corresponding layout from model 1 in figure 6.5 above.

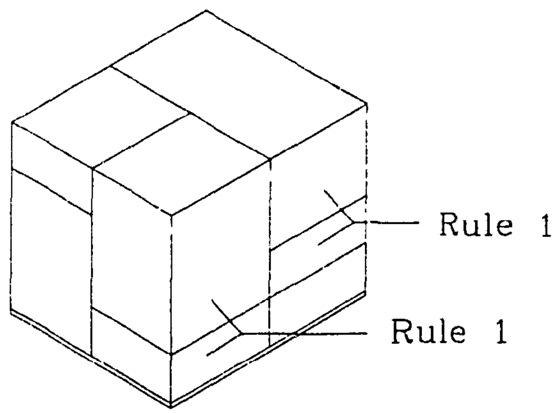



**Figure 6.6: Implementation of Stability Rules (Seq. 6, Sim. 3, Model 2)**

To show how the application of the stability rules affect the performance of the palletization process, the results of sequence #3 from simulation 1 of the two models are compared in table 6.1. The quantity of each box type in the pallet load resulting from model 1 is taken as the demand requirements for these boxes in model 2. The sequence is as follows:

3-2-6-5-4-2-7-4-3-7-2-1-5-3-6-2-6-3-2-3-1-2-2

**Table 6.1: Effect of Stability implementation on Palletization Performance**

| | Stability | | Demand Satisfac'n (%) | VPU (%) | TNSV | WIP Remain'g ($inch^2$) | Max. WIP ($inch^2$) | RPT (s) |
|---|---|---|---|---|---|---|---|---|
| | Rule 1 | Rule 2 | | | | | | |
| Model 1 | - | - | 100 | 89.03 | 4 | 0 | 360 | 1080 |
| Model 2 | 4 | 0 | 98.25 | 87.5 | 9 | 60 | 360 | 1120 |

It is worth mentioning that the 89.03% VPU is the maximum utilization of the pallet in the case of model 1 and that the remaining 4 sub volumes can no longer accommodate any of the available 8 types of boxes used in the simulation. The results of model 2 found that the box type whose demand was not fully satisfied is type 2. Thus the VPU is reduced to 87.5%. Since there were no change in the box order coming into the system, the reduction in VPU must be because stability rule 1 was successfully implemented 4 times in model 2. TNSV increased from 4 to 9 sub volumes. However, when the dimensions of the 9 sub volumes of model 2 are carefully looked at, it is found that they can accommodate 2 more boxes of type 1 and 1 more box of type 3. Thus in terms of demand requirements, stabilitv application did not allow full satisfaction of demand of box type 2. However, it allows a potential increase in VPU to a value of 91.41% with a reduction in TNSV from 9 to a value of 6.

## 6.2 Comparing Results with the Literature

This part of the chapter compares the results of the proposed models with those of previous works found in the literature. Since all literature found on 3D palletization heuristics solve the basic palletization problem without considering box demand requirements or stability of pallet loads, only the results of model 1 are compared with the literature. Furthermore, the comparison is narrowed to the second and third simulations of model 1 because they have considered the set of box sizes of table 5.2 composed of geometrically harmonious box dimensions as opposed to the set of box dimensions used in simulation 1. Comparison is made between the results of simulations 2 and 3 of model 1 and those of Yang and Arghavani since their contributions are the closest among literature to the research at hand. The indices that are used for comparison are the volumetric pallet utilization, VPU, the work-in-process, WIP, and the robotic palletization time, RPT.

Arghavani developed his model to deal with the random pattern of incoming boxes of the different dimensions shown previously in table 5.2. Arghavani uses the column stacking procedure to load the boxes on the pallet. The stacking procedure in his model is based on loading boxes onto one column at a time and the next column stack starts loading only when the previous one is completely filled. The different aspects of the comparison between Arghavani's column stacked palletization results and the interlocking stacked loads of model 1 are shown in Table 6.2. The random sequences used by Arghavani are composed of two boxes of type 1, two boxes of type 2, and two boxes of type 4. This combination of box types and quantities was selected by Arghavani because it provides 100% pallet utilization.

**Table 6.2: Comparison of Performance criteria Outputs with Literature (Sim. 2, Model 1)**

| # | Sequence | RPT (s) | Max. WIP Area | VPU |
|---|---|---|---|---|
| 1 | 2-2-1-4-1-4 | 280 ≤ [280] | 480 ≤ [480] | 100% = 100% |
| 2 | 2-2-4-4-1-1 | 240 ≤ [240] | 0 ≤ [ 0 ] | 100% = 100% |
| 3 | 2-1-4-2-1-4 | 280 < [320] | 480 < [1440] | 100% = 100% |
| 4 | 1-2-1-4-4-2 | 240 ≤ [240] | 0 ≤ [ 0 ] | 100% = 100% |
| 5 | 1-4-4-1-2-2 | 240 ≤ [240] | 0 < [ 0 ] | 100% = 100% |
| 6 | 1-4-2-2-4-1 | 280 < [320] | 480 < [1920] | 100% = 100% |
| 7 | 4-1-1-4-2-2 | 280 < [320] | 480 < [960] | 100% = 100% |
| 8 | 4-2-2-1-4-1 | 280 < [440] | 480 < [2400] | 100% = 100% |
| 9 | 2-1-4-4-1-2 | 240 < [320] | 0 < [ 960 ] | 100% = 100% |
| 10 | 4-1-1-2-2-4 | 240 < [400] | 0 < [1440] | 100% = 100% |

The results in the square brackets [ ] are obtained by Arghavani.

Both, the results of Arghavani's model and model 1 show 100% volumetric pallet utilization. However, the results of model 1 concerning both the WIP and the RPT are far better than those of Arghavani. Five out of the ten sequences did not use the WIP at all thus resulting in the minimum possible value of the RPT which is the number of trips from the conveyor to the pallet. While the first two sequence show no difference in performance. the WIP of sequence 3 is reduced by 200% of Arghavani's WIP. The most achieving sequence is #10 where the maximum WIP used is completely eliminated and

66.67% of the RPT is saved. This is clearly seen in table 6.3. where the simulated robot movement frequency and robotic palletization time are summarized.

**Table 6.3: Comparison of Robot Movement Frequency with Literature (Sim. 2, Model 1)**

| Seq # | C ☞ P | C ☞ WIP | WIP ☞ P | P ☞ WIP | Time (s) |
|---|---|---|---|---|---|
| 1 | 5 [5] | 1 [1] | 1 [1] | 0 [0] | 280 ≤ [280] |
| 2 | 6 [6] | 0 [0] | 0 [0] | 0 [0] | 240 ≤ [240] |
| 3 | 5 [4] | 1 [2] | 1 [2] | 0 [0] | 280 < [320] |
| 4 | 6 [6] | 0 [0] | 0 [0] | 0 [0] | 240 ≤ [240] |
| 5 | 6 [6] | 0 [0] | 0 [0] | 0 [0] | 240 ≤ [240] |
| 6 | 5 [4] | 1 [2] | 1 [2] | 0 [0] | 280 < [320] |
| 7 | 5 [4] | 1 [2] | 1 [2] | 0 [0] | 280 < [320] |
| 8 | 5 [3] | 1 [3] | 1 [4] | 0 [1] | 280 < [440] |
| 9 | 6 [4] | 0 [2] | 0 [2] | 0 [0] | 240 < [320] |
| 10 | 6 [4] | 0 [2] | 0 [3] | 0 [1] | 240 < [400] |

The results in the square brackets [ ] are obtained by Arghavani.

The table contains the number of times the robot loaded boxes on the pallet straight from the conveyor denoted by (C ☞ P), the frequency of the robot going to the WIP with a box from the conveyor (C ☞ WIP). Also tabulated is the frequency of the robot picking boxes from the WIP and loading them on the pallet, (WIP ☞ P). The column before last contains the number of times it was decided to unload boxes from the pallet to the WIP (P ☞ WIP). This later column reveals that it was twice needed to unloaded a box from Arghavani's pallet whereas this resort was never sought by model 1. Moreover, the table shows that the number of times boxes were brought and taken from the WIP lies in the range of 0 to 4 (two way trips) in Arghavani's 10 sequences while it is only between 0 and 1 in the case of model 1. It is not surprising that interlocked stacking achieves better and faster solutions than column stacking palletization since it offers greater flexibility in loading boxes at more locations on the pallet. Thus minimizing WIP utilization as well as the robotic palletization time.

Simulation 3 enables not only the comparison between the interlocked stacking and the column stacking approach, but also includes the results of the layer by layer palletization procedure developed by Yang. The ten sequences of simulation 3 were once used by Yang, then Arghavani to validate their models. They are composed of boxes from the same set of box dimensions used in simulation 2 proposed also by Yang, (table 5.2). Each sequence is made of two boxes of type 1, one box of type 2, two boxes of type 3 as well as of type 5. The results of the simulation are presented in table 6.4 below along with the results of Arghavani and Yang. Again, the VPU is always 100% except for two of Yang's sequences. The table shows how column stacking produces improved results over the layered palletization approach. In turn, the interlocked stacking of boxes reveals to be the most achieving in reducing the RPT as well as the usage of the WIP.

**Table 6.4: Comparison of Performance criteria Outputs with Literature (Sim. 3, Model 1)**

| # | Sequence | RPT(s) | Max. WIP Area | VPU |
|---|---|---|---|---|
| 1 | 1-3-2-3-5-1-5 | 360 ≤ [360] < (440) | 480 < [960] < (2400) | 100% = 100% =100% |
| 2 | 5-5-3-3-2-1-1 | 280 ≤ [280] < (360) | 0 ≤ [ 0 ] < (1440) | 100% = 100% =100% |
| 3 | 5-5-1-3-1-2-3 | 320 < [360] < (400) | 480 < [1920] < (2400) | 100% =100% =100% |
| 4 | 5-5-1-3-3-1-2 | 280 ≤ [280] < (400) | 0 ≤ [ 0 ] < (1920) | 100% = 100% > 75% |
| 5 | 1-3-1-2-5-5-3 | 320 ≤ [320] < (400) | 480 ≤ [480] < (1920) | 100% = 100% =100% |
| 6 | 1-5-1-2-5-3-3 | 320 < [360] ≤ (360) | 960 < [1920] ≤ (1920) | 100% = 100% =100% |
| 7 | 2-3-5-5-1-3-1 | 280 < [440] > (400) | 0 < [1440] < (1920) | 100% = 100% =100% |
| 8 | 3-3-2-5-1-5-1 | 320 ≤ [320] < (440) | 960 > [480] < (1920) | 100% = 100% =100% |
| 9 | 3-3-2-1-1-5-5 | 280 ≤ [280] < (400) | 0 ≤ [ 0 ] < (1920) | 100% = 100% =100% |
| 10 | 3-3-5-1-1-2-5 | 280 < [440] > (400) | 0 ≤ [1920] ≤ (1920) | 100% = 100% > 75% |

The results in the square brackets [ ] are obtained by Arghavani for column stacking.
The results in the parentheses ( ) are obtained by Yang for layer palletization.

Table 6.5 contains the compared frequency of motions of the robot to and from the various stations of the system. Unlike Arghavani, and similar to model 1, Yang's model seems not using the unloading of boxes from the pallet. The reason is because it has side effects of increasing the RPT as well as the WIP usage.

**Table 6.5: Comparison of Robot Movement Frequency with Literature (Sim. 3, Model 1)**

| Seq # | C ☞ P | C ☞ WIP | WIP ☞ P | P ☞ WIP | Time (s) |
|---|---|---|---|---|---|
| 1 | 5 [5] (3) | 2 [2] (4) | 2 [2] (4) | 0 [0] (0) | 360 [360] (440) |
| 2 | 7 [7] (5) | 0 [0] (2) | 0 [0] (2) | 0 [0] (0) | 280 [280] (360) |
| 3 | 6 [5] (4) | 1 [2] (3) | 1 [2] (3) | 0 [0] (0) | 320 [360] (400) |
| 4 | 7 [7] (3) | 0 [0] (4) | 0 [0] (3) | 0 [0] (0) | 280 [280] (400) |
| 5 | 6 [6] (3) | 1 [1] (4) | 1 [1] (3) | 0 [0] (0) | 320 [320] (400) |
| 6 | 6 [5] (5) | 1 [2] (2) | 1 [2] (2) | 0 [0] (0) | 320 [360] (360) |
| 7 | 7 [5] (4) | 0 [2] (3) | 0 [3] (3) | 0 [1] (0) | 280 [440] (400) |
| 8 | 6 [6] (3) | 1 [1] (4) | 1 [1] (4) | 0 [0] (0) | 320 [320] (440) |
| 9 | 7 [7] (4) | 0 [0] (3) | 0 [0] (3) | 0 [0] (0) | 280 [280] (400) |
| 10 | 7 [5] (3) | 0 [2] (4) | 0 [3] (3) | 0 [1] (0) | 280 [440] (400) |

The results in the square brackets [ ] are obtained by Arghavani.
The results in the parentheses ( ) are obtained by Yang.

# CHAPTER 7.0 CONCLUSIONS AND RECOMMENDATIONS

This study provides insight into three-dimensional palletization problems. Two heuristic models are developed to find near optimum solutions to the general palletization problem where boxes of different sizes and quantities arrive at the pallet loading workstation in random order. The research's main feature is the method by which the heuristics load the boxes on to the pallet. Interlocked stacking is a new palletization technique. It is a mix of the layered and the column stacked palletization procedures. This new approach proved to be far more efficient than the two conventional ones since it offers higher flexibility to load more box types at more locations on the pallet. It also achieves the inter-locking of the boxes on the pallet thus indirectly increasing the pallet load stability.

In trying to achieve maximum volumetric pallet utilization (VPU), minimum WIP surface coverage, and minimum robotic palletization time(RPT), Model 1 introduced new decision making indices. These indices are part of the decision making mind of the heuristic. Model 2 developed two practical stability rules that improved the overall stability of the pallet load when they were applied to blocks made of two or three boxes available on the conveyor and the WIP. Besides designing for stability, model 2 also considered box demand requirements. Two sets of box dimensions were used to test the models. The first set is based on actual dimensions of boxes used in the day to day activities. The second set of box sizes is taken from the literature. This later set is composed of boxes with proportional dimensions among each other so a to increase the probability of achieving good solutions. Three simulations, each comprising of 10 different box sequences, were conducted to test the two heuristics. It is no surprising to find out that the results of the model using the first set of box dimensions, do not produce 100% VPU. In fact, it was known before hand that VPU=100% was not possible due to the boxes' non harmonious dimensions. It was found that when there was a demand for boxes belonging to the first set of box sizes, building more stable pallet loads would often interfere with satisfying the box demands. In the other hand, when the demand was for boxes belonging to the second set of box dimensions (that of the literature), 100% demand satisfaction was always achieved besides having much more stable loads.

As interlocked stacking was the loading procedure used in the research, the results of the second simulation of model 1 were compared to Arghavani's column stacking results. Furthermore, the results of the third simulation of model 1 were compared with Yang's layer palletization results as well as those of Arghavani. The comparison shows that the interlocked stacking achieves higher or equal VPU and greatly reduces the WIP usage and the RPT. Even though the pallet load stability is extensively explored in model 2, the results of model 1 share some of its glory.

This work has only opened new doors to palletization research. Especially to solve practical problems that often use the heuristic approach. The inspiration from real-life palletization scenarios whether in warehousing or transportation, can only bring more ideas that need to be pursued and developed to improve and go beyond what has already been achieved.

Future study may include the following:

1- Physically implement the proposed models.

2- Improve and generalize the current models for more robust industrial applications.

3- Study extensively the newly established performance measures and indices.

4- Develop empirical and analytical relationships between the various palletization parameters.

5- Incorporate CAD simulations to existing or new models. This will have a great impact on the visualization part of the research as well as enhance new design concepts.

## REFERENCES


**1993**

M. Yang (1993) **Multi-layer Palletization of Multi-Size Boxes for 2½D and 3D Problems.** Master Thesis, Concordia University.

J. Arghavani (1993) **3D Stacking Palletization of Multi-Size Boxes.** Master Thesis, Concordia University.

R. D. Tsai *et al.* (1993) **Three Dimensional Palletization of Mixed Box Sizes.** *IIE Trans.*, vol. 25, no. 4. pp. 64-75.

M. Hanna. (1993) **Stability Criteria of Interlocked Boxes on a Pallet.** Research Report, Concordia University.

S. V. Amiouny (1993) **Combinational Mechanics.** Ph.D. Thesis, Georgia Institute of Technology.

**1991**

W. B. Dowsland (1991) **Three-Dimensional Packing – Solution Approach and Heuristic Development.** *Int. J. Prod. Res.* vol. 29, no. 8, pp. 1673-1685.

C. S. Chen *et. al.* (1991) **The pallet packing problem for non-uniform box sizes.** *Int. J. Prod. Res.* vol. 29, no.10, pp.1963-1968.

**1990**

R. W. Haessler *et al.* (1990) **Load Planning for Shipments of Low Density Products.** *Eur. J. Opr. Res.* vol. 44, pp.289-299.

H. Gehring *et al.* (1990) **A Computer-Based Heuristic for Packing Pooled Shipment Containers.** *Eur. J. Opr. Res.* vol. 44, pp. 277-288.

E. Lee (1990) **Automated Palletization of Multiple Box Sizes.** Master Thesis, University of Windsor.

**1989**

Wan, Frederic Y. M.(1989) <u>Mathematical Models & their Analysis</u>, New York: Harper & Row, pp. 1-7.

C. P. Han *et al.* (1989) **A Heuristic Approach to the Three-Dimensional Cargo-Loading Problem.** *Int. J. Prod. Res.* vol. 27, no. 5 pp. 757-774

G. Löschau (1989) **Stability Criterion for Column Stacks.** *Packaging Technology and Science*, 2, pp. 155-163.

**1985**
H. Carpenter *et al.* (1985) **Practical Considerations of the Pallet-Loading Problem.** *J. Opr. Res. Soc.* vol. 36, pp. 489-497.

J. Pearl, (1985)Heuristics: Intelligent search strategies for computer problem solving, Addison-Wesley Pub. Co. pp.3-5.

**1982**
T. J. Hodgson (1982) **A Combined Approach to the Pallet Loading Problem.** *IIE Trans.* vol. 14, no. 3, pp. 175-182.

**1980**
Jacoby, Samuel L. S. et al (1980) Mathematical Modelling with Computers, N.J. Prentice Hall, pp. ix-5.

# BIBLIOGRAPHY


**1994**

B. B. Mohanty *et al.* (1994) **Value considerations in three-dimensional packing - A heuristic procedure using the fractional knapsack problem.** *Eur. J. Opr. Res.* vol. 74, no.1, pp. 143-151.

**1993**

Yeong-Dae Kim *et. al.* (1993) **Heuristic Approaches for Loading Problems in Flexible Manufacturing Systems.** *IIE Trans.*, vol. 25, no. 1. pp. 26-39.

M. Yang (1993) **Multi-layer Palletization of Multi-Size Boxes for 2½D and 3D Problems.** Master Thesis, Concordia University.

J. Arghavani (1993) **3D Stacking Palletization of Multi-Size Boxes.** Master Thesis, Concordia University.

R. D. Tsai *et al.* (1993) **Three Dimensional Palletization of Mixed Box Sizes.** *IIE Trans.*, vol. 25, no. 4. pp. 64-75.

M. Hanna (1993) **Stability Criteria of Interlocked Boxes on a Pallet.** Research Report, Concordia University.

S. V. Amiouny (1993) **Combinational Mechanics.** Ph.D. Thesis, Georgia Institute of Technology.

Lin, Jin-ling *et.al.* (1993) **A Smile for Packing and Pallet Loading in Three Dimensions,** Hybrid Genetic Algorithm for Container Packing in Three Dimensions. *Proc. Conf. Artif. Intell. Appl.* pp. 763-768.

J. Cahill (1993) **Mixed load order picking: The next frontier.** *Beverage World*, vol. 112, no. 1535, pp. 26-29.

**1992**

R. Balasubramanian (1992)**The pallet loading problem: A survey.** *Int. J. Prod. Econ.* vol 28, no.2, pp. 217-225.

B. Chandra (1992) **Does randomization help in on-line bin packing?** *Information Processing Letters*, vol 43, no.1, pp.15-19.

**1991**
W. B. Dowsland (1991) **Three-Dimensional Packing – Solution Approach and Heuristic Development.** *Int. J. Prod. Res.* vol. 29, no. 8, pp. 1673-1685.

C. S. Chen *et. al.* (1991) **The pallet packing problem for non-uniform box sizes.** *Int. J. Prod. Res.* vol. 29, no.10, pp.1963-1968.

**1990**
R. W. Haessler *et al.* (1990) **Load Planning for Shipments of Low Density Products.** *Eur. J. Opr. Res.* vol. 44, pp.289-299.

H. Gehring *et al.* (1990) **A Computer-Based Heuristic for Packing Pooled Shipment Containers.** *Eur. J. Opr. Res.* vol. 44, pp. 277-288.

E. Lee (1990) **Automated Palletization of Multiple Box Sizes.**
Master Thesis, University of Windsor.

**1989**
Wan, Frederic Y. M.(1989) Mathematical Models & their Analysis, New York: Harper & Row, pp. 1-7.

C. P. Han *et al.* (1989) **A Heuristic Approach to the Three-Dimensional Cargo-Loading Problem.** *Int. J. Prod. Res.* vol. 27, no. 5 pp. 757-774

G. Löschau (1989) **Stability Criterion for Column Stacks.** *Packaging Technology and Science*, 2, pp. 155-163.

**1988**
R. D. Tsai *et. al.* **A Two-Dimensional Palletization procedure for Warehouse Loading Operations.** *IIE Trans.*, vol. 20, no. 4. pp. 418-425.

R. A. Penington et. al. **Robotic palletization of multiple box sizes.** *Int. J. Opr. Res.* vol. 26, no.1, pp. 95-105.

**1986**
F. M. Puls *et. al.* **Robotic implementation of pallet loading patterns.** *Int. J. Prod. Res.* vol. 24, no.3, pp.635-645.

**1985**

H. Carpenter *et al.* (1985) **Practical Considerations of the Pallet-Loading Problem.** *J. Opr. Res. Soc.* vol. 36, pp. 489–497.

Y. Ohtake et. al. (1985) **A branch-and-bound Algorithm for 0-1 Parametric Mixed Integer Programming.** *Opr. Res. Letters*, vol.4, no.1, pp.41–45.

J. Pearl, (1985) Heuristics: Intelligent search strategies for computer problem solving, Addison-Wesley Pub. Co. pp.3-5.

J. E. Beasley (1985) **An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure.** *Operations Research*, vol.33, no.1, pp.49-64.

J. E. Beasley (1985) **Bounds for Two-Dimensional Cutting.** *J. Opr. Res. Soc.* vol.36, no.1, pp.71-74.

R. M. O'Keefe (1985) **Expert Systems and Operational Research - Mutual Benefits.** *J. Opr. Res. Soc.* vol.36, no.2, pp.125-129.

J. E. Beasley (1985) **Algorithms for Unconstrained Two-Dimensional Guillotine Cutting.** *J. Opr. Res. Soc.* vol.36, no.4, pp.297-306.

R. G. Jeroslow and J. K. Lowe (1985) **Experimental Results on the New Techniques for Integer Programming Formulations.** *J. Opr. Res. Soc.* vol.36, no.5, pp.393–403.

**1984**

S. A. Roberts (1984) **Application of Heuristic Techniques to the Cutting-Stock Problem for Worktops.** *J. Opr. Res. Soc.* vol.35, no.5, pp.369-377.

Kathryn A. Dowsland (1984) **The Three-Dimensional Pallet Chart: An Analysis of the Factors Affecting the Set of Feasible Layouts for a Class of Two-Dimensional Packing Problems.** *J. Opr. Res. Soc.* vol.35, no.10, pp.895-905.

**1983**

P. Y. Wang (1983) **Two Algorithms for Constrained Two-Dimensional Cutting Stock Problems.** *Operations Research*, vol.31, no.3, pp.573-586.

**1982**

T. J. Hodgson (1982) **A Combined Approach to the Pallet Loading Problem.** IIE Trans. vol. 14, no. 3, pp. 175-182.

Bengt-Erik Bengtsson (1982) **Packing Rectangular Pieces - A Heuristic Approach.** *The Computer Journal*, vol. 25, no. 3, pp. 353-357.

**1981**

Igal Golan (1981) **Performance Bounds for Orthogonal Oriented Two-Dimensional Packing Algorithms.** *Siam J. Comput.* vol.10. no.3, pp.571-582.

**1980**

Jacoby, Samuel L. S. et al (1980) Mathematical Modeling with Computers. N.J. Prentice Hall, pp. ix-5.

Donna J. Brown (1980) **An Improved BL Lower Bound.** *Information Processing Letters*, vol.11, no.1, pp. 37-39.

J. A. George and D. F. Robinson (1980) **A Heuristic for Packing Boxes into a Container.** *Comput. & Ops Res.* vol. 7, pp. 147-156.

**1979**

Harold J. Steudel (1979) **Generating Pallet Loading Patterns: A Special Case of the Two-Dimensional Cutting Stock Problem.** *Management Science*, vol.25, no.10, pp. 997-1004.

**1977**

Richard M.H. Cheng and Aron W. Pila (1977) **Maximized Utilization of Surface Areas with Defects by the Dynamic Programming Approach.** *ISA Trans.* vol.17, no.4, pp. 61-69.

**1965**

P. C. Gilmore and R. E. Gomory (1964) **Multistage Cutting Stock Problems of Two and more Dimensions.** Opns. Res. vol.13, pp. 94-120.

## APPENDIX A

Sample Output File of Model 1 Program
Sequence 1 of Simulation 1.

SEQUENCE #1

newbox = 4
newbox = 5
Now boxc1 = 4
Now boxc2 = 5
There is no boxes in the WIP.
The box, 4 fits in sub volume 1
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 1
PRVLC = 244
zerocount = 8
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
boxc1 is of type 4
    subvolume_boxc1 = 1
    orientation_boxc1 = 13
    TNSV_boxc1 = 3
    case_boxc1 = 4
    PRVLC_boxc1 = 244
    zerocount_boxc1 = 8
The box, 5 fits in sub volume 1
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 1
PRVLC = 247
zerocount = 6
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
boxc2 is of type 5
    subvolume_boxc2 = 1
    orientation_boxc2 = 13
    TNSV_boxc2 = 3
    case_boxc2 = 4
    PRVLC_boxc2 = 247
    zerocount_boxc2 = 6
boxc1 of type 4 is gone to pallet
PALLETIZATION UPDATE REPORT

PALLET STATUS
    Volumetric Pallet Utilization, VPU = 7.03%
    Number of Boxes on the Pallet: 1
        1) Box Type 4
            Position Vector, (L,W,H) = 0, 0, 0
            Orientation: l//L and w//W
    Number of Empty Sub Volumes on the Pallet: 3

        Sub Volume #1
            Dimensions:
                Length = 48
                Width = 30
                Height = 40
            Position Vector:

Length = 0
Width = 10
Height = 0
Sub Volume #2
Dimensions:
Length = 30
Width = 10
Height = 40
Position Vector:
Length = 18
Width = 0
Height = 0
Sub Volume #3
Dimensions:
Length = 18
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 40 seconds.

newbox = 7
Now boxc1 = 5
Now boxc2 = 7
There is no boxes in the WIP.
The box, 5 fits in sub volume 1
the box orientation is l//W and w//L
PRVLC_spat 1 = PRVLC_spat 2 = 186
zerocount 1 <= zerocount 2
subvolume = 1
PRVLC = 224
zerocount = 10
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 5 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 24
zerocount 1 <= zerocount 2
subvolume = 2
PRVLC = 233
zerocount = 8
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 224
The box, 5 does not fit in sub volume 3
boxc1 is of type 5

subvolume_boxc1 = 2
orientation_boxc1 = 13
TNSV_boxc1 = 3
case_boxc1 = 2
PRVLC_boxc1 = 233
zerocount_boxc1 = 8
The box, 7 fits in sub volume 1
the box orientation is l//W and w//L
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 1
PRVLC = 187
zerocount = 18
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 7 does not fit in sub volume 2
The box, 7 does not fit in sub volume 3
boxc2 is of type 7
subvolume_boxc2 = 1
orientation_boxc2 = 23
TNSV_boxc2 = 5
case_boxc2 = 4
PRVLC_boxc2 = 187
zerocount_boxc2 = 18
boxc1 of type 5 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS
Volumetric Pallet Utilization, VPU = 10.94%
Number of Boxes on the Pallet: 2
1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
Number of Empty Sub Volumes on the Pallet: 3

Sub Volume #1
Dimensions:
Length = 48
Width = 30
Height = 40
Position Vector:
Length = 0
Width = 10
Height = 0
Sub Volume #2
Dimensions:
Length = 18
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30

Sub Volume #3
Dimensions:
Length = 30
Width = 10
Height = 30
Position Vector:
Length = 18
Width = 0
Height = 10

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 80 seconds.

newbox = 2
Now boxc1 = 7
Now boxc2 = 2
There is no boxes in the WIP.
The box, 7 fits in sub volume 1
the box orientation is l//W and w//L
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 1
PRVLC = 176
zerocount = 18
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 7 does not fit in sub volume 2
The box, 7 does not fit in sub volume 3
boxc1 is of type 7
subvolume_boxc1 = 1
orientation_boxc1 = 23
TNSV_boxc1 = 5
case_boxc1 = 4
PRVLC_boxc1 = 176
zerocount_boxc1 = 18
The box, 2 fits in sub volume 1
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 1
PRVLC = 227
zerocount = 18
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 2 does not fit in sub volume 2
The box, 2 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 3
PRVLC = 226
zerocount = 23

zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 227
boxc2 is of type 2
subvolume_boxc2 = 1
orientation_boxc2 = 13
TNSV_boxc2 = 5
case_boxc2 = 4
PRVLC_boxc2 = 227
zerocount_boxc2 = 18
boxc1 of type 7 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS

Volumetric Pallet Utilization, VPU = 20.7%
Number of Boxes on the Pallet: 3

1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W

2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W

3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L

Number of Empty Sub Volumes on the Pallet: 5

Sub Volume #1
Dimensions:
Length = 18
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30

Sub Volume #2
Dimensions:
Length = 30
Width = 10
Height = 30
Position Vector:
Length = 18
Width = 0
Height = 10

Sub Volume #3
Dimensions:
Length = 48
Width = 5
Height = 40
Position Vector:
Length = 0
Width = 35
Height = 0

Sub Volume #4
Dimensions:

Length = 33
Width = 25
Height = 40
Position Vector:
Length = 15
Width = 10
Height = 0
Sub Volume #5
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 120 seconds.

newbox = 6
Now boxc1 = 2
Now boxc2 = 6
There is no boxes in the WIP.
The box, 2 does not fit in sub volume 1
The box, 2 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 2
PRVLC = 169
zerocount = 33
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 2 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 21
zerocount 1 <= zerocount 2
subvolume = 3
PRVLC = 173
zerocount = 24
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 169
The box, 2 fits in sub volume 4
the box orientation is l//W and w//L
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 4
PRVLC = 157
zerocount = 30
zero_TCBT = 1

zero_TCBT_min = 0
PRVLC_max = 173
The box, 2 fits in sub volume 5
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 5
PRVLC = 172
zerocount = 25
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 173
boxc1 is of type 2
    subvolume_boxc1 = 3
    orientation_boxc1 = 13
    TNSV_boxc1 = 6
    case_boxc1 = 3
    PRVLC_boxc1 = 173
    zerocount_boxc1 = 24
The box, 6 does not fit in sub volume 1
The box, 6 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 6
zerocount 1 <= zerocount 2
subvolume = 2
PRVLC = 160
zerocount = 15
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 6 does not fit in sub volume 3
The box, 6 fits in sub volume 4
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 66
zerocount 1 <= zerocount 2
subvolume = 4
PRVLC = 138
zerocount = 30
zero_TCBT = 1
zero_TCBT_min = 0
PRVLC_max = 160
The box, 6 does not fit in sub volume 5
boxc2 is of type 6
    subvolume_boxc2 = 2
    orientation_boxc2 = 13
    TNSV_boxc2 = 4
    case_boxc2 = 1
    PRVLC_boxc2 = 160
    zerocount_boxc2 = 15
boxc1 of type 2 is gone to WIP

PALLETIZATION UPDATE REPORT

PALLET STATUS
    Volumetric Pallet Utilization, VPU = 20.7%
    Number of Boxes on the Pallet: 3
      1) Box Type 4
        Position Vector, (L,W,H) = 0, 0, 0

Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
Number of Empty Sub Volumes on the Pallet: 5

Sub Volume #1
Dimensions:
Length = 18
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30
Sub Volume #2
Dimensions:
Length = 30
Width = 10
Height = 30
Position Vector:
Length = 18
Width = 0
Height = 10
Sub Volume #3
Dimensions:
Length = 48
Width = 5
Height = 40
Position Vector:
Length = 0
Width = 35
Height = 0
Sub Volume #4
Dimensions:
Length = 33
Width = 25
Height = 40
Position Vector:
Length = 15
Width = 10
Height = 0
Sub Volume #5
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20

WIP STATUS

Number of Boxes in the WIP: 1
It's type is 2
Actual WIP surface area = 60 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 160 seconds.

newbox = 4
Now boxc1 = 6
Now boxc2 = 4
The box, 2 does not fit in sub volume 1
The box, 2 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 2
PRVLC = 169
zerocount = 33
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 2 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 21
zerocount 1 <= zerocount 2
subvolume = 3
PRVLC = 173
zerocount = 24
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 169
The box, 2 fits in sub volume 4
the box orientation is l//W and w//L
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 4
PRVLC = 157
zerocount = 30
zero_TCBT = 1
zero_TCBT_min = 0
PRVLC_max = 173
The box, 2 fits in sub volume 5
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 5
PRVLC = 172
zerocount = 25
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 173
bestboxinWIP is of type 2
orientation_bestboxinWIP = 13
subvolume_bestboxinWIP = 3
TNSV_bestboxinWIP = 6
case_bestboxinWIP = 3
PRVLC_bestboxinWIP = 173
zerocount_bestboxinWIP = 24

The box, 6 does not fit in sub volume 1
The box, 6 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 6
zerocount 1 <= zerocount 2
subvolume = 2
PRVLC = 160
zerocount = 15
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 6 does not fit in sub volume 3
The box, 6 fits in sub volume 4
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 66
zerocount 1 <= zerocount 2
subvolume = 4
PRVLC = 138
zerocount = 30
zero_TCBT = 1
zero_TCBT_min = 0
PRVLC_max = 160
The box, 6 does not fit in sub volume 5
boxc1 is of type 6
    subvolume_boxc1 = 2
    orientation_boxc1 = 13
    TNSV_boxc1 = 4
    case_boxc1 = 1
    PRVLC_boxc1 = 160
    zerocount_boxc1 = 15
The box, 4 does not fit in sub volume 1
The box, 4 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 8
zerocount 1 <= zerocount 2
subvolume = 2
PRVLC = 160
zerocount = 22
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 4 does not fit in sub volume 3
The box, 4 fits in sub volume 4
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 4
PRVLC = 141
zerocount = 32
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 160
The box, 4 does not fit in sub volume 5
boxc2 is of type 4
    subvolume_boxc2 = 2
    orientation_boxc2 = 13
    TNSV_boxc2 = 5

case_boxc2 = 2
PRVLC_boxc2 = 160
zerocount_boxc2 = 22
boxc1 of type 6 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS
Volumetric Pallet Utilization, VPU = 28.52%
Number of Boxes on the Pallet: 4
1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
Number of Empty Sub Volumes on the Pallet: 4

Sub Volume #1
Dimensions:
Length = 48
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30
Sub Volume #2
Dimensions:
Length = 48
Width = 5
Height = 40
Position Vector:
Length = 0
Width = 35
Height = 0
Sub Volume #3
Dimensions:
Length = 33
Width = 25
Height = 40
Position Vector:
Length = 15
Width = 10
Height = 0
Sub Volume #4
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:

Length = 0
Width = 10
Height = 20

WIP STATUS
Number of Boxes in the WIP: 1
It's type is 2
Actual WIP surface area = 60 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 200 seconds.

newbox = 2
Now boxc1 = 4
Now boxc2 = 2
The box, 2 does not fit in sub volume 1
The box, 2 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 21
zerocount 1 <= zerocount 2
subvolume = 2
PRVLC = 157
zerocount = 21
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 2 fits in sub volume 3
the box orientation is l//W and w//L
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 3
PRVLC = 141
zerocount = 27
zero_TCBT = 1
zero_TCBT_min = 0
PRVLC_max = 157
The box, 2 fits in sub volume 4
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 4
PRVLC = 156
zerocount = 22
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 157
bestboxinWIP is of type 2
orientation_bestboxinWIP = 13
subvolume_bestboxinWIP = 2
TNSV_bestboxinWIP = 5
case_bestboxinWIP = 3
PRVLC_bestboxinWIP = 157
zerocount_bestboxinWIP = 21
The box, 4 does not fit in sub volume 1
The box, 4 does not fit in sub volume 2
The box, 4 fits in sub volume 3
the box orientation is l//W and w//L

PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 3
PRVLC = 125
zerocount = 29
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 4 does not fit in sub volume 4
boxc1 is of type 4
    subvolume_boxc1 = 3
    orientation_boxc1 = 23
    TNSV_boxc1 = 6
    case_boxc1 = 4
    PRVLC_boxc1 = 125
    zerocount_boxc1 = 29
best box in WIP, type 2 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS
    Volumetric Pallet Utilization, VPU = 30.08%
    Number of Boxes on the Pallet: 5
    1) Box Type 4
        Position Vector, (L,W,H) = 0, 0, 0
        Orientation: l//L and w//W
    2) Box Type 5
        Position Vector, (L,W,H) = 18, 0, 0
        Orientation: l//L and w//W
    3) Box Type 7
        Position Vector, (L,W,H) = 0, 10, 0
        Orientation: l//W and w//L
    4) Box Type 6
        Position Vector, (L,W,H) = 18, 0, 10
        Orientation: l//L and w//W
    5) Box Type 2
        Position Vector, (L,W,H) = 0, 35, 0
        Orientation: l//L and w//W
    Number of Empty Sub Volumes on the Pallet: 5

    Sub Volume #1
        Dimensions:
            Length = 48
            Width = 10
            Height = 10
        Position Vector:
            Length = 0
            Width = 0
            Height = 30
    Sub Volume #2
        Dimensions:
            Length = 33
            Width = 25
            Height = 40
        Position Vector:
            Length = 15
            Width = 10
            Height = 0

Sub Volume #3
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20
Sub Volume #4
Dimensions:
Length = 36
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #5
Dimensions:
Length = 12
Width = 5
Height = 20
Position Vector:
Length = 0
Width = 35
Height = 20

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 240 seconds.

There is no boxes in the WIP.
The box, 4 does not fit in sub volume 1
The box, 4 fits in sub volume 2
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 2
PRVLC = 117
zerocount = 37
zero_TCBT = 1
zero_TCBT_min = 8
PRVLC_max = 0
The box, 4 does not fit in sub volume 3
The box, 4 does not fit in sub volume 4
The box, 4 does not fit in sub volume 5
boxc1 is of type 4
subvolume_boxc1 = 2
orientation_boxc1 = 23
TNSV_boxc1 = 7
case_boxc1 = 4
PRVLC_boxc1 = 117

zerocount_boxc1 = 37
The box, 2 does not fit in sub volume 1
The box, 2 fits in sub volume 2
the box orientation is l//W and w//L
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 2
PRVLC = 138
zerocount = 33
zero_TCBT = 1
zero_TCBT_min = 8
PRVLC_max = 0
The box, 2 fits in sub volume 3
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 3
PRVLC = 153
zerocount = 28
zero_TCBT = 0
zero_TCBT_min = 1
PRVLC_max = 138
The box, 2 fits in sub volume 4
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 15
zerocount 1 <= zerocount 2
subvolume = 4
PRVLC = 154
zerocount = 21
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 153
The box, 2 fits in sub volume 5
the box orientation is l//L and w//W
The sub volume is completely filled
subvolume = 5
PRVLC = 154
zerocount = 15
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 154
boxc2 is of type 2
subvolume_boxc2 = 5
orientation_boxc2 = 13
TNSV_boxc2 = 4
case_boxc2 = 1
PRVLC_boxc2 = 154
zerocount_boxc2 = 15
boxc1 of type 4 is gone to WIP

PALLETIZATION UPDATE REPORT

PALLET STATUS
Volumetric Pallet Utilization, VPU = 30.08%
Number of Boxes on the Pallet: 5
1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W
2) Box Type 5

Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W

Number of Empty Sub Volumes on the Pallet: 5

Sub Volume #1
Dimensions:
Length = 48
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30

Sub Volume #2
Dimensions:
Length = 33
Width = 25
Height = 40
Position Vector:
Length = 15
Width = 10
Height = 0

Sub Volume #3
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20

Sub Volume #4
Dimensions:
Length = 36
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0

Sub Volume #5
Dimensions:
Length = 12
Width = 5
Height = 20
Position Vector:
Length = 0

Width = 35
Height = 20

WIP STATUS
Number of Boxes in the WIP: 1
It's type is 4
Actual WIP surface area = 180 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 280 seconds.

newbox = 8
Now boxc1 = 2
Now boxc2 = 8
The box, 4 does not fit in sub volume 1
The box, 4 fits in sub volume 2
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 2
PRVLC = 117
zerocount = 37
zero_TCBT = 1
zero_TCBT_min = 8
PRVLC_max = 0
The box, 4 does not fit in sub volume 3
The box, 4 does not fit in sub volume 4
The box, 4 does not fit in sub volume 5
bestboxinWIP is of type 4
orientation_bestboxinWIP = 23
subvolume_bestboxinWIP = 2
TNSV_bestboxinWIP = 7
case_bestboxinWIP = 4
PRVLC_bestboxinWIP = 117
zerocount_bestboxinWIP = 37
The box, 2 does not fit in sub volume 1
The box, 2 fits in sub volume 2
the box orientation is l//W and w//L
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 2
PRVLC = 138
zerocount = 33
zero_TCBT = 1
zero_TCBT_min = 8
PRVLC_max = 0
The box, 2 fits in sub volume 3
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 3
PRVLC = 153
zerocount = 28
zero_TCBT = 0
zero_TCBT_min = 1
PRVLC_max = 138
The box, 2 fits in sub volume 4
the box orientation is l//L and w//W

PRVLC_spat 1 = PRVLC_spat 2 = 15
zerocount 1 <= zerocount 2
subvolume = 4
PRVLC = 154
zerocount = 21
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 153
The box, 2 fits in sub volume 5
the box orientation is l//L and w//W
The sub volume is completely filled
subvolume = 5
PRVLC = 154
zerocount = 15
zero_TCBT = 0
zero_TCBT_min = 0
PRVLC_max = 154
boxc1 is of type 2
    subvolume_boxc1 = 5
    orientation_boxc1 = 13
    TNSV_boxc1 = 4
    case_boxc1 = 1
    PRVLC_boxc1 = 154
    zerocount_boxc1 = 15
The box, 8 does not fit in sub volume 1
The box, 8 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 2
PRVLC = 94
zerocount = 37
zero_TCBT = 2
zero_TCBT_min = 8
PRVLC_max = 0
The box, 8 does not fit in sub volume 3
The box, 8 does not fit in sub volume 4
The box, 8 does not fit in sub volume 5
boxc2 is of type 8
    subvolume_boxc2 = 2
    orientation_boxc2 = 13
    TNSV_boxc2 = 7
    case_boxc2 = 4
    PRVLC_boxc2 = 94
    zerocount_boxc2 = 37
boxc1 of type 2 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS

Volumetric Pallet Utilization, VPU = 31.64%
Number of Boxes on the Pallet: 6

1) Box Type 4
   Position Vector, (L,W,H) = 0, 0, 0
   Orientation: l//L and w//W
2) Box Type 5
   Position Vector, (L,W,H) = 18, 0, 0
   Orientation: l//L and w//W

3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W

Number of Empty Sub Volumes on the Pallet: 4

Sub Volume #1
Dimensions:
Length = 48
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30

Sub Volume #2
Dimensions:
Length = 33
Width = 25
Height = 40
Position Vector:
Length = 15
Width = 10
Height = 0

Sub Volume #3
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20

Sub Volume #4
Dimensions:
Length = 36
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0

WIP STATUS

Number of Boxes in the WIP: 1
It's type is 4
Actual WIP surface area = 180 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 320 seconds.

newbox = 6
Now boxc1 = 8
Now boxc2 = 6
The box, 4 does not fit in sub volume 1
The box, 4 fits in sub volume 2
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 2
PRVLC = 119
zerocount = 29
zero_TCBT = 0
zero_TCBT_min = 8
PRVLC_max = 0
The box, 4 does not fit in sub volume 3
The box, 4 does not fit in sub volume 4
bestboxinWIP is of type 4
    orientation_bestboxinWIP = 23
    subvolume_bestboxinWIP = 2
    TNSV_bestboxinWIP = 6
    case_bestboxinWIP = 4
    PRVLC_bestboxinWIP = 119
    zerocount_bestboxinWIP = 29
The box, 8 does not fit in sub volume 1
The box, 8 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 2
PRVLC = 91
zerocount = 29
zero_TCBT = 1
zero_TCBT_min = 8
PRVLC_max = 0
The box, 8 does not fit in sub volume 3
The box, 8 does not fit in sub volume 4
boxc1 is of type 8
    subvolume_boxc1 = 2
    orientation_boxc1 = 13
    TNSV_boxc1 = 6
    case_boxc1 = 4
    PRVLC_boxc1 = 91
    zerocount_boxc1 = 29
best box in WIP, type 4 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS

Volumetric Pallet Utilization, VPU = 38.67%
Number of Boxes on the Pallet: 7
1) Box Type 4
    Position Vector, (L,W,H) = 0, 0, 0
    Orientation: l//L and w//W
2) Box Type 5
    Position Vector, (L,W,H) = 18, 0, 0

Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L

Number of Empty Sub Volumes on the Pallet: 6

Sub Volume #1
Dimensions:
Length = 48
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30

Sub Volume #2
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20

Sub Volume #3
Dimensions:
Length = 23
Width = 30
Height = 40
Position Vector:
Length = 25
Width = 10
Height = 0

Sub Volume #4
Dimensions:
Length = 13
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0

Sub Volume #5
Dimensions:

Length = 10
Width = 7
Height = 40
Position Vector:
Length = 15
Width = 28
Height = 0
Sub Volume #6
Dimensions:
Length = 10
Width = 18
Height = 10
Position Vector:
Length = 15
Width = 10
Height = 30

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 360 seconds.

There is no boxes in the WIP.
The box, 8 does not fit in sub volume 1
The box, 8 does not fit in sub volume 2
The box, 8 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 35
zerocount 1 <= zerocount 2
subvolume = 3
PRVLC = 89
zerocount = 32
zero_TCBT = 1
zero_TCBT_min = 8
PRVLC_max = 0
The box, 8 does not fit in sub volume 4
The box, 8 does not fit in sub volume 5
The box, 8 does not fit in sub volume 6
boxc1 is of type 8
subvolume_boxc1 = 3
orientation_boxc1 = 13
TNSV_boxc1 = 6
case_boxc1 = 2
PRVLC_boxc1 = 89
zerocount_boxc1 = 32
The box, 6 does not fit in sub volume 1
The box, 6 does not fit in sub volume 2
The box, 6 fits in sub volume 3
the box orientation is l//W and w//L
PRVLC_spat 1 = PRVLC_spat 2 = 64
zerocount 1 <= zerocount 2
subvolume = 3
PRVLC = 111

zerocount = 34
zero_TCBT = 1
zero_TCBT_min = 8
PRVLC_max = 0
The box, 6 does not fit in sub volume 4
The box, 6 does not fit in sub volume 5
The box, 6 does not fit in sub volume 6
boxc2 is of type 6
    subvolume_boxc2 = 3
    orientation_boxc2 = 23
    TNSV_boxc2 = 7
    case_boxc2 = 3
    PRVLC_boxc2 = 111
    zerocount_boxc2 = 34
boxc1 of type 8 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS

Volumetric Pallet Utilization, VPU = 54.14%
Number of Boxes on the Pallet: 8

1) Box Type 4
   Position Vector, (L,W,H) = 0, 0, 0
   Orientation: l//L and w//W
2) Box Type 5
   Position Vector, (L,W,H) = 18, 0, 0
   Orientation: l//L and w//W
3) Box Type 7
   Position Vector, (L,W,H) = 0, 10, 0
   Orientation: l//W and w//L
4) Box Type 6
   Position Vector, (L,W,H) = 18, 0, 10
   Orientation: l//L and w//W
5) Box Type 2
   Position Vector, (L,W,H) = 0, 35, 0
   Orientation: l//L and w//W
6) Box Type 2
   Position Vector, (L,W,H) = 0, 35, 20
   Orientation: l//L and w//W
7) Box Type 4
   Position Vector, (L,W,H) = 15, 10, 0
   Orientation: l//W and w//L
8) Box Type 8
   Position Vector, (L,W,H) = 25, 10, 0
   Orientation: l//L and w//W

Number of Empty Sub Volumes on the Pallet: 6

Sub Volume #1
  Dimensions:
    Length = 48
    Width = 10
    Height = 10
  Position Vector:
    Length = 0
    Width = 0
    Height = 30
Sub Volume #2

Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20
Sub Volume #3
Dimensions:
Length = 32
Width = 12
Height = 40
Position Vector:
Length = 15
Width = 28
Height = 0
Sub Volume #4
Dimensions:
Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #5
Dimensions:
Length = 32
Width = 18
Height = 10
Position Vector:
Length = 15
Width = 10
Height = 30
Sub Volume #6
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 400 seconds.

newbox = 5
Now boxc1 = 6
Now boxc2 = 5

There is no boxes in the WIP.
The box, 6 does not fit in sub volume 1
The box, 6 does not fit in sub volume 2
The box, 6 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 17
zerocount 1 <= zerocount 2
subvolume = 3
PRVLC = 59
zerocount = 49
zero_TCBT = 2
zero_TCBT_min = 8
PRVLC_max = 0
The box, 6 does not fit in sub volume 4
The box, 6 does not fit in sub volume 5
The box, 6 does not fit in sub volume 6
boxc1 is of type 6
    subvolume_boxc1 = 3
    orientation_boxc1 = 13
    TNSV_boxc1 = 8
    case_boxc1 = 4
    PRVLC_boxc1 = 59
    zerocount_boxc1 = 49
The box, 5 fits in sub volume 1
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 3
zerocount 1 <= zerocount 2
subvolume = 1
PRVLC = 81
zerocount = 33
zero_TCBT = 1
zero_TCBT_min = 8
PRVLC_max = 0
The box, 5 does not fit in sub volume 2
The box, 5 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 24
zerocount 1 <= zerocount 2
subvolume = 3
PRVLC = 66
zerocount = 48
zero_TCBT = 1
zero_TCBT_min = 1
PRVLC_max = 81
The box, 5 does not fit in sub volume 4
The box, 5 fits in sub volume 5
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 2
zerocount 1 <= zerocount 2
subvolume = 5
PRVLC = 81
zerocount = 42
zero_TCBT = 1
zero_TCBT_min = 1
PRVLC_max = 81
The box, 5 does not fit in sub volume 6

boxc2 is of type 5
subvolume_boxc2 = 1
orientation_boxc2 = 13
TNSV_boxc2 = 6
case_boxc2 = 2
PPVLC_boxc2 = 81
zerocount_boxc2 = 33
boxc1 of type 6 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS
Volumetric Pallet Utilization, VPU = 61.95%
Number of Boxes on the Pallet: 9
1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L
8) Box Type 8
Position Vector, (L,W,H) 25, 10, 0
Orientation: l//L and w//W
9) Box Type 6
Position Vector, (L,W,H) = 15, 28, 0
Orientation: l//L and w//W
Number of Empty Sub Volumes on the Pallet: 8

Sub Volume #1
Dimensions:
Length = 48
Width = 10
Height = 10
Position Vector:
Length = 0
Width = 0
Height = 30
Sub Volume #2
Dimensions:
Length = 15
Width = 25
Height = 20

Position Vector:
Length = 0
Width = 10
Height = 20
Sub Volume #3
Dimensions:
Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #4
Dimensions:
Length = 32
Width = 18
Height = 10
Position Vector:
Length = 15
Width = 10
Height = 30
Sub Volume #5
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0
Sub Volume #6
Dimensions:
Length = 2
Width = 12
Height = 40
Position Vector:
Length = 45
Width = 28
Height = 0
Sub Volume #7
Dimensions:
Length = 30
Width = 2
Height = 40
Position Vector:
Length = 15
Width = 38
Height = 0
Sub Volume #8
Dimensions:
Length = 30
Width = 10
Height = 20
Position Vector:
Length = 15

Width = 28
Height = 20

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 440 seconds.

newbox = 1
Now boxc1 = 5
Now boxc2 = 1
There is no boxes in the WIP.
The box, 5 fits in sub volume 1
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 3
zerocount 1 <= zerocount 2
subvolume = 1
PRVLC = 51
zerocount = 50
zero_TCBT = 2
zero_TCBT_min = 8
PRVLC_max = 0
The box, 5 does not fit in sub volume 2
The box, 5 does not fit in sub volume 3
The box, 5 fits in sub volume 4
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 2
zerocount 1 <= zerocount 2
subvolume = 4
PRVLC = 51
zerocount = 59
zero_TCBT = 2
zero_TCBT_min = 2
PRVLC_max = 51
The box, 5 does not fit in sub volume 5
The box, 5 does not fit in sub volume 6
The box, 5 does not fit in sub volume 7
The box, 5 fits in sub volume 8
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 6
zerocount 1 <= zerocount 2
subvolume = 8
PRVLC = 48
zerocount = 51
zero_TCBT = 3
zero_TCBT_min = 2
PRVLC_max = 51
boxc1 is of type 5
subvolume_boxc1 = 1
orientation_boxc1 = 13
TNSV_boxc1 = 8
case_boxc1 = 2
PRVLC_boxc1 = 51

zerocount_boxc1 = 50
The box, 1 fits in sub volume 1
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 1
PRVLC = 58
zerocount = 56
zero_TCBT = 2
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 fits in sub volume 2
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 2
PRVLC = 56
zerocount = 63
zero_TCBT = 3
zero_TCBT_min = 2
PRVLC_max = 58
The box, 1 does not fit in sub volume 3
The box, 1 fits in sub volume 4
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 4
PRVLC = 58
zerocount = 56
zero_TCBT = 2
zero_TCBT_min = 2
PRVLC_max = 58
The box, 1 does not fit in sub volume 5
The box, 1 does not fit in sub volume 6
The box, 1 does not fit in sub volume 7
The box, 1 fits in sub volume 8
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 8
PRVLC = 54
zerocount = 64
zero_TCBT = 3
zero_TCBT_min = 2
PRVLC_max = 58
boxc2 is of type 1
subvolume_boxc2 = 4
orientation_boxc2 = 13
TNSV_boxc2 = 9
case_boxc2 = 3
PRVLC_boxc2 = 58
zerocount_boxc2 = 56
boxc1 of type 5 is gone to pallet
PALLETIZATION UPDATE REPORT

PALLET STATUS
Volumetric Pallet Utilization, VPU = 65.86%
Number of Boxes on the Pallet: 10
1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0

Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L
8) Box Type 8
Position Vector, (L,W,H) = 25, 10, 0
Orientation: l//L and w//W
9) Box Type 6
Position Vector, (L,W,H) = 15, 28, 0
Orientation: l//L and w//W
10) Box Type 5
Position Vector, (L,W,H) = 0, 0, 30
Orientation: l//L and w//W
Number of Empty Sub Volumes on the Pallet: 8

Sub Volume #1
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20
Sub Volume #2
Dimensions:
Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #3
Dimensions:
Length = 32
Width = 18
Height = 10
Position Vector:
Length = 15
Width = 10

Height = 30
Sub Volume #4
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0
Sub Volume #5
Dimensions:
Length = 2
Width = 12
Height = 40
Position Vector:
Length = 45
Width = 28
Height = 0
Sub Volume #6
Dimensions:
Length = 30
Width = 2
Height = 40
Position Vector:
Length = 15
Width = 38
Height = 0
Sub Volume #7
Dimensions:
Length = 30
Width = 10
Height = 20
Position Vector:
Length = 15
Width = 28
Height = 20
Sub Volume #8
Dimensions:
Length = 18
Width = 10
Height = 10
Position Vector:
Length = 30
Width = 0
Height = 30

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 480 seconds.

newbox = 3

Now boxc1 = 1
Now boxc2 = 3
There is no boxes in the WIP.
The box, 1 fits in sub volume 1
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 1
PRVLC = 48
zerocount = 64
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 does not fit in sub volume 2
The box, 1 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 3
PRVLC = 50
zerocount = 57
zero_TCBT = 2
zero_TCBT_min = 3
PRVLC_max = 48
The box, 1 does not fit in sub volume 4
The box, 1 does not fit in sub volume 5
The box, 1 does not fit in sub volume 6
The box, 1 fits in sub volume 7
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 7
PRVLC = 46
zerocount = 65
zero_TCBT = 3
zero_TCBT_min = 2
PRVLC_max = 50
The box, 1 fits in sub volume 8
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 1
zerocount 1 <= zerocount 2
subvolume = 8
PRVLC = 49
zerocount = 59
zero_TCBT = 2
zero_TCBT_min = 2
PRVLC_max = 50
boxc1 is of type 1
    subvolume_boxc1 = 3
    orientation_boxc1 = 13
    TNSV_boxc1 = 9
    case_boxc1 = 3
    PRVLC_boxc1 = 50
    zerocount_boxc1 = 57
The box, 3 fits in sub volume 1
the box orientation is l//W and w//L
PRVLC_spat 1 = PRVLC_spat 2 = 9
zerocount 1 <= zerocount 2
subvolume = 1

```
PRVLC = 39
zerocount = 64
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
The box, 3 does not fit in sub volume 2
The box, 3 fits in sub volume 3
the box orientation is l//W and w//L
PRVLC_spat 1 = PRVLC_spat 2 = 6
zerocount 1 <= zerocount 2
subvolume = 3
PRVLC = 47
zerocount = 51
zero_TCBT = 2
zero_TCBT_min = 3
PRVLC_max = 39
The box, 3 does not fit in sub volume 4
The box, 3 does not fit in sub volume 5
The box, 3 does not fit in sub volume 6
The box, 3 fits in sub volume 7
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 9
zerocount 1 <= zerocount 2
subvolume = 7
PRVLC = 43
zerocount = 59
zero_TCBT = 3
zero_TCBT_min = 2
PRVLC_max = 47
The box, 3 fits in sub volume 8
the box orientation is l//L and w//W
The sub volume is completely filled
subvolume = 8
PRVLC = 48
zerocount = 44
zero_TCBT = 2
zero_TCBT_min = 2
PRVLC_max = 47
boxc2 is of type 3
    subvolume_boxc2 = 8
    orientation_boxc2 = 13
    TNSV_boxc2 = 7
    case_boxc2 = 1
    PRVLC_boxc2 = 48
    zerocount_boxc2 = 44
boxc1 of type 1 is gone to WIP
                PALLETIZATION UPDATE REPORT

    PALLET STATUS
        Volumetric Pallet Utilization, VPU = 65.86%
        Number of Boxes on the Pallet: 10
          1) Box Type 4
            Position Vector, (L,W,H) = 0, 0, 0
            Orientation: l//L and w//W
          2) Box Type 5
            Position Vector, (L,W,H) = 18, 0, 0
```

Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L
8) Box Type 8
Position Vector, (L,W,H) = 25, 10, 0
Orientation: l//L and w//W
9) Box Type 6
Position Vector, (L,W,H) = 15, 28, 0
Orientation: l//L and w//W
10) Box Type 5
Position Vector, (L,W,H) = 0, 0, 30
Orientation: l//L and w//W

Number of Empty Sub Volumes on the Pallet: 8

Sub Volume #1
Dimensions:
Length = 15
Width = 25
Height = 20
Position Vector:
Length = 0
Width = 10
Height = 20
Sub Volume #2
Dimensions:
Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #3
Dimensions:
Length = 32
Width = 18
Height = 10
Position Vector:
Length = 15
Width = 10
Height = 30
Sub Volume #4
Dimensions:

```
            Length = 1
            Width = 30
            Height = 40
         Position Vector:
            Length = 47
            Width = 10
            Height = 0
      Sub Volume #5
         Dimensions:
            Length = 2
            Width = 12
            Height = 40
         Position Vector:
            Length = 45
            Width = 28
            Height = 0
      Sub Volume #6
         Dimensions:
            Length = 30
            Width = 2
            Height = 40
         Position Vector:
            Length = 15
            Width = 38
            Height = 0
      Sub Volume #7
         Dimensions:
            Length = 30
            Width = 10
            Height = 20
         Position Vector:
            Length = 15
            Width = 28
            Height = 20
      Sub Volume #8
         Dimensions:
            Length = 18
            Width = 10
            Height = 10
         Position Vector:
            Length = 30
            Width = 0
            Height = 30

  WIP STATUS
     Number of Boxes in the WIP: 1
     It's type is 1
     Actual WIP surface area  = 60  inches^2

  CONVEYOR STATUS

  TOTAL PALLETIZATION TIME = 520 seconds.

newbox = 7
Now boxc1 = 3
Now boxc2 = 7
```

The box, 1 fits in sub volume 1
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 1
PRVLC = 48
zerocount = 64
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 does not fit in sub volume 2
The box, 1 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 3
PRVLC = 50
zerocount = 57
zero_TCBT = 2
zero_TCBT_min = 3
PRVLC_max = 48
The box, 1 does not fit in sub volume 4
The box, 1 does not fit in sub volume 5
The box, 1 does not fit in sub volume 6
The box, 1 fits in sub volume 7
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 7
PRVLC = 46
zerocount = 65
zero_TCBT = 3
zero_TCBT_min = 2
PRVLC_max = 50
The box, 1 fits in sub volume 8
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 1
zerocount 1 <= zerocount 2
subvolume = 8
PRVLC = 49
zerocount = 59
zero_TCBT = 2
zero_TCBT_min = 2
PRVLC_max = 50
bestboxinWIP is of type 1
    orientation_bestboxinWIP = 13
    subvolume_bestboxinWIP = 3
    TNSV_bestboxinWIP = 9
    case_bestboxinWIP = 3
    PRVLC_bestboxinWIP = 50
    zerocount_bestboxinWIP = 57
The box, 3 fits in sub volume 1
the box orientation is l//W and w//L
PRVLC_spat 1 = PRVLC_spat 2 = 9
zerocount 1 <= zerocount 2
subvolume = 1
PRVLC = 39
zerocount = 64
zero_TCBT = 3

zero_TCBT_min = 8
PRVLC_max = 0
The box, 3 does not fit in sub volume 2
The box, 3 fits in sub volume 3
the box orientation is l//W and w//L
PRVLC_spat 1 = PRVLC_spat 2 = 6
zerocount 1 <= zerocount 2
subvolume = 3
PRVLC = 47
zerocount = 51
zero_TCBT = 2
zero_TCBT_min = 3
PRVLC_max = 39
The box, 3 does not fit in sub volume 4
The box, 3 does not fit in sub volume 5
The box, 3 does not fit in sub volume 6
The box, 3 fits in sub volume 7
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 9
zerocount 1 <= zerocount 2
subvolume = 7
PRVLC = 43
zerocount = 59
zero_TCBT = 3
zero_TCBT_min = 2
PRVLC_max = 47
The box, 3 fits in sub volume 8
the box orientation is l//L and w//W
The sub volume is completely filled
subvolume = 8
PRVLC = 48
zerocount = 44
zero_TCBT = 2
zero_TCBT_min = 2
PRVLC_max = 47
boxc1 is of type 3
    subvolume_boxc1 = 8
    orientation_boxc1 = 13
    TNSV_boxc1 = 7
    case_boxc1 = 1
    PRVLC_boxc1 = 48
    zerocount_boxc1 = 44
The box, 7 fits in sub volume 1
the box orientation is l//W and w//L
The sub volume is completely filled
subvolume = 1
PRVLC = 30
zerocount = 46
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
The box, 7 does not fit in sub volume 2
The box, 7 does not fit in sub volume 3
The box, 7 does not fit in sub volume 4
The box, 7 does not fit in sub volume 5
The box, 7 does not fit in sub volume 6

The box, 7 does not fit in sub volume 7
The box, 7 does not fit in sub volume 8
boxc2 is of type 7
    subvolume_boxc2 = 1
    orientation_boxc2 = 23
    TNSV_boxc2 = 7
    case_boxc2 = 1
    PRVLC_boxc2 = 30
    zerocount_boxc2 = 46
boxc1 of type 3 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS
    Volumetric Pallet Utilization, VPU = 68.2%
    Number of Boxes on the Pallet: 11
        1) Box Type 4
            Position Vector, (L,W,H) = 0, 0, 0
            Orientation: l//L and w//W
        2) Box Type 5
            Position Vector, (L,W,H) = 18, 0, 0
            Orientation: l//L and w//W
        3) Box Type 7
            Position Vector, (L,W,H) = 0, 10, 0
            Orientation: l//W and w//L
        4) Box Type 6
            Position Vector, (L,W,H) = 18, 0, 10
            Orientation: l//L and w//W
        5) Box Type 2
            Position Vector, (L,W,H) = 0, 35, 0
            Orientation: l//L and w//W
        6) Box Type 2
            Position Vector, (L,W,H) = 0, 35, 20
            Orientation: l//L and w//W
        7) Box Type 4
            Position Vector, (L,W,H) = 15, 10, 0
            Orientation: l//W and w//L
        8) Box Type 8
            Position Vector, (L,W,H) = 25, 10, 0
            Orientation: l//L and w//W
        9) Box Type 6
            Position Vector, (L,W,H) = 15, 28, 0
            Orientation: l//L and w//W
        10) Box Type 5
            Position Vector, (L,W,H) = 0, 0, 30
            Orientation: l//L and w//W
        11) Box Type 3
            Position Vector, (L,W,H) = 30, 0, 30
            Orientation: l//L and w//W
    Number of Empty Sub Volumes on the Pallet: 7

        Sub Volume #1
            Dimensions:
                Length = 15
                Width = 25
                Height = 20
            Position Vector:

Length = 0
Width = 10
Height = 20
Sub Volume #2
Dimensions:
Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #3
Dimensions:
Length = 32
Width = 18
Height = 10
Position Vector:
Length = 15
Width = 10
Height = 30
Sub Volume #4
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0
Sub Volume #5
Dimensions:
Length = 2
Width = 12
Height = 40
Position Vector:
Length = 45
Width = 28
Height = 0
Sub Volume #6
Dimensions:
Length = 30
Width = 2
Height = 40
Position Vector:
Length = 15
Width = 38
Height = 0
Sub Volume #7
Dimensions:
Length = 30
Width = 10
Height = 20
Position Vector:
Length = 15
Width = 28

Height = 20

WIP STATUS
Number of Boxes in the WIP: 1
It's type is 1
Actual WIP surface area = 60 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 560 seconds.

newbox = 6
Now boxc1 = 7
Now boxc2 = 6
The box, 1 fits in sub volume 1
the box orientation is l//W and w//L
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 1
PRVLC = 45
zerocount = 58
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 does not fit in sub volume 2
The box, 1 fits in sub volume 3
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 3
PRVLC = 47
zerocount = 51
zero_TCBT = 2
zero_TCBT_min = 3
PRVLC_max = 45
The box, 1 does not fit in sub volume 4
The box, 1 does not fit in sub volume 5
The box, 1 does not fit in sub volume 6
The box, 1 fits in sub volume 7
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 7
PRVLC = 43
zerocount = 59
zero_TCBT = 3
zero_TCBT_min = 2
PRVLC_max = 47
bestboxinWIP is of type 1
orientation_bestboxinWIP = 13
subvolume_bestboxinWIP = 3
TNSV_bestboxinWIP = 8
case_bestboxinWIP = 3
PRVLC_bestboxinWIP = 47
zerocount_bestboxinWIP = 51
The box, 7 fits in sub volume 1
the box orientation is l//W and w//L
The sub volume is completely filled
subvolume = 1

PRVLC = 27
zerocount = 40
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
The box, 7 does not fit in sub volume 2
The box, 7 does not fit in sub volume 3
The box, 7 does not fit in sub volume 4
The box, 7 does not fit in sub volume 5
The box, 7 does not fit in sub volume 6
The box, 7 does not fit in sub volume 7
boxc1 is of type 7
    subvolume_boxc1 = 1
    orientation_boxc1 = 23
    TNSV_boxc1 = 6
    case_boxc1 = 1
    PRVLC_boxc1 = 27
    zerocount_boxc1 = 40
The box, 6 does not fit in sub volume 1
The box, 6 does not fit in sub volume 2
The box, 6 does not fit in sub volume 3
The box, 6 does not fit in sub volume 4
The box, 6 does not fit in sub volume 5
The box, 6 does not fit in sub volume 6
The box, 6 fits in sub volume 7
the box orientation is l//L and w//W
The sub volume is completely filled
subvolume = 7
PRVLC = 31
zerocount = 41
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
boxc2 is of type 6
    subvolume_boxc2 = 7
    orientation_boxc2 = 13
    TNSV_boxc2 = 6
    case_boxc2 = 1
    PRVLC_boxc2 = 31
    zerocount_boxc2 = 41
boxc1 of type 7 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS
    Volumetric Pallet Utilization, VPU = 77.97%
    Number of Boxes on the Pallet: 12
        1) Box Type 4
            Position Vector, (L,W,H) = 0, 0, 0
            Orientation: l//L and w//W
        2) Box Type 5
            Position Vector, (L,W,H) = 18, 0, 0
            Orientation: l//L and w//W
        3) Box Type 7
            Position Vector, (L,W,H) = 0, 10, 0
            Orientation: l//W and w//L
        4) Box Type 6

Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L
8) Box Type 8
Position Vector, (L,W,H) = 25, 10, 0
Orientation: l//L and w//W
9) Box Type 6
Position Vector, (L,W,H) = 15, 28, 0
Orientation: l//L and w//W
10) Box Type 5
Position Vector, (L,W,H) = 0, 0, 30
Orientation: l//L and w//W
11) Box Type 3
Position Vector, (L,W,H) = 30, 0, 30
Orientation: l//L and w//W
12) Box Type 7
Position Vector, (L,W,H) = 0, 10, 20
$Orientation: l//W and w//L

Number of Empty Sub Volumes on the Pallet: 6

Sub Volume #1
Dimensions:
Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0

Sub Volume #2
Dimensions:
Length = 32
Width = 18
Height = 10
Position Vector:
Length = 15
Width = 10
Height = 30

Sub Volume #3
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0

Sub Volume #4

```
                Dimensions:
                    Length = 2
                    Width = 12
                    Height = 40
                Position Vector:
                    Length = 45
                    Width = 28
                    Height = 0
            Sub Volume #5
                Dimensions:
                    Length = 30
                    Width = 2
                    Height = 40
                Position Vector:
                    Length = 15
                    Width = 38
                    Height = 0
            Sub Volume #6
                Dimensions:
                    Length = 30
                    Width = 10
                    Height = 20
                Position Vector:
                    Length = 15
                    Width = 28
                    Height = 20

    WIP STATUS
        Number of Boxes in the WIP: 1
        It's type is 1
        Actual WIP surface area  = 60  inches^2

    CONVEYOR STATUS

    TOTAL PALLETIZATION TIME = 600 seconds.

newbox = 1
Now boxc1 = 6
Now boxc2 = 1
The box, 1 does not fit in sub volume 1
The box, 1 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 2
PRVLC = 26
zerocount = 47
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 does not fit in sub volume 3
The box, 1 does not fit in sub volume 4
The box, 1 does not fit in sub volume 5
The box, 1 fits in sub volume 6
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 6
```

PRVLC = 22
zerocount = 55
zero_TCBT = 4
zero_TCBT_min = 3
PRVLC_max = 26
bestboxinWIP is of type 1
    orientation_bestboxinWIP = 13
    subvolume_bestboxinWIP = 2
    TNSV_bestboxinWIP = 7
    case_bestboxinWIP = 3
    PRVLC_bestboxinWIP = 26
    zerocount_bestboxinWIP = 47
The box, 6 does not fit in sub volume 1
The box, 6 does not fit in sub volume 2
The box, 6 does not fit in sub volume 3
The box, 6 does not fit in sub volume 4
The box, 6 does not fit in sub volume 5
The box, 6 fits in sub volume 6
the box orientation is l//L and w//W
The sub volume is completely filled
subvolume = 6
PRVLC = 10
zerocount = 37
zero_TCBT = 5
zero_TCBT_min = 8
PRVLC_max = 0
boxc1 is of type 6
    subvolume_boxc1 = 6
    orientation_boxc1 = 13
    TNSV_boxc1 = 5
    case_boxc1 = 1
    PRVLC_boxc1 = 10
    zerocount_boxc1 = 37
The box, 1 does not fit in sub volume 1
The box, 1 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 2
PRVLC = 26
zerocount = 47
zero_TCBT = 3
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 does not fit in sub volume 3
The box, 1 does not fit in sub volume 4
The box, 1 does not fit in sub volume 5
The box, 1 fits in sub volume 6
the box orientation is l//L and w//W
PRVLC_spat 1 > PRVLC_spat 2, no need for zerocount
subvolume = 6
PRVLC = 22
zerocount = 55
zero_TCBT = 4
zero_TCBT_min = 3
PRVLC_max = 26
boxc2 is of type 1

subvolume_boxc2 = 2
orientation_boxc2 = 13
TNSV_boxc2 = 7
case_boxc2 = 3
PRVLC_boxc2 = 26
zerocount_boxc2 = 47
boxc1 of type 6 is gone to pallet
PALLETIZATION UPDATE REPORT

PALLET STATUS
Volumetric Pallet Utilization, VPU = 85.78%
Number of Boxes on the Pallet: 13
1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L
8) Box Type 8
Position Vector, (L,W,H) = 25, 10, 0
Orientation: l//L and w//W
9) Box Type 6
Position Vector, (L,W,H) = 15, 28, 0
Orientation: l//L and w//W
10) Box Type 5
Position Vector, (L,W,H) = 0, 0, 30
Orientation: l//L and w//W
11) Box Type 3
Position Vector, (L,W,H) = 30, 0, 30
Orientation: l//L and w//W
12) Box Type 7
Position Vector, (L,W,H) = 0, 10, 20
Orientation: l//W and w//L
13) Box Type 6
Position Vector, (L,W,H) = 15, 28, 20
Orientation: l//L and w//W
Number of Empty Sub Volumes on the Pallet: 5

Sub Volume #1
Dimensions:
Length = 3

Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #2
Dimensions:
Length = 32
Width = 18
Height = 10
Position Vector:
Length = 15
Width = 10
Height = 30
Sub Volume #3
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0
Sub Volume #4
Dimensions:
Length = 2
Width = 12
Height = 40
Position Vector:
Length = 45
Width = 28
Height = 0
Sub Volume #5
Dimensions:
Length = 30
Width = 2
Height = 40
Position Vector:
Length = 15
Width = 38
Height = 0

WIP STATUS
Number of Boxes in the WIP: 1
It's type is 1
Actual WIP surface area = 60 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 640 seconds.

newbox = 5
Now boxc1 = 1
Now boxc2 = 5
The box, 1 does not fit in sub volume 1

The box, 1 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 2
PRVLC = 9
zerocount = 44
zero_TCBT = 5
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 does not fit in sub volume 3
The box, 1 does not fit in sub volume 4
The box, 1 does not fit in sub volume 5
bestboxinWIP is of type 1
    orientation_bestboxinWIP = 13
    subvolume_bestboxinWIP = 2
    TNSV_bestboxinWIP = 6
    case_bestboxinWIP = 3
    PRVLC_bestboxinWIP = 9
    zerocount_bestboxinWIP = 44
The box, 1 does not fit in sub volume 1
The box, 1 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 2 > PRVLC_spat 1, no need for zerocount
subvolume = 2
PRVLC = 9
zerocount = 44
zero_TCBT = 5
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 does not fit in sub volume 3
The box, 1 does not fit in sub volume 4
The box, 1 does not fit in sub volume 5
boxc1 is of type 1
    subvolume_boxc1 = 2
    orientation_boxc1 = 13
    TNSV_boxc1 = 6
    case_boxc1 = 3
    PRVLC_boxc1 = 9
    zerocount_boxc1 = 44
The box, 5 does not fit in sub volume 1
The box, 5 fits in sub volume 2
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 2
zerocount 1 <= zerocount 2
subvolume = 2
PRVLC = 2
zerocount = 47
zero_TCBT = 7
zero_TCBT_min = 8
PRVLC_max = 0
The box, 5 does not fit in sub volume 3
The box, 5 does not fit in sub volume 4
The box, 5 does not fit in sub volume 5
boxc2 is of type 5
    subvolume_boxc2 = 2
    orientation_boxc2 = 13

TNSV_boxc2 = 6
case_boxc2 = 3
PRVLC_boxc2 = 2
zerocount_boxc2 = 47
boxc1 of type 1 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS

Volumetric Pallet Utilization, VPU = 86.56%
Number of Boxes on the Pallet: 14

1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L
8) Box Type 8
Position Vector, (L,W,H) = 25, 10, 0
Orientation: l//L and w//W
9) Box Type 6
Position Vector, (L,W,H) = 15, 28, 0
Orientation: l//L and w//W
10) Box Type 5
Position Vector, (L,W,H) = 0, 0, 30
Orientation. l//L and w//W
11) Box Type 3
Position Vector, (L,W,H) = 30, 0, 30
Orientation: l//L and w//W
12) Box Type 7
Position Vector, (L,W,H) = 0, 10, 20
Orientation: l//W and w//L
13) Box Type 6
Position Vector, (L,W,H) = 15, 28, 20
Orientation: l//L and w//W
14) Box Type 1
Position Vector, (L,W,H) = 15, 10, 30
Orientation: l//L and w//W

Number of Empty Sub Volumes on the Pallet: 6

Sub Volume #1
Dimensions:

Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #2
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0
Sub Volume #3
Dimensions:
Length = 2
Width = 12
Height = 40
Position Vector:
Length = 45
Width = 28
Height = 0
Sub Volume #4
Dimensions:
Length = 30
Width = 2
Height = 40
Position Vector:
Length = 15
Width = 38
Height = 0
Sub Volume #5
Dimensions:
Length = 32
Width = 13
Height = 10
Position Vector:
Length = 15
Width = 15
Height = 30
Sub Volume #6
Dimensions:
Length = 20
Width = 5
Height = 10
Position Vector:
Length = 27
Width = 10
Height = 30

WIP STATUS
Number of Boxes in the WIP: 1
It's type is 1

Actual WIP surface area = 60 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 680 seconds.

newbox = 999
Now boxc1 = 5
Now boxc2 = 999
The box, 1 does not fit in sub volume 1
The box, 1 does not fit in sub volume 2
The box, 1 does not fit in sub volume 3
The box, 1 does not fit in sub volume 4
The box, 1 fits in sub volume 5
the box orientation is l//W and w//L
PRVLC_spat 1 = PRVLC_spat 2 = 6
zerocount 1 <= zerocount 2
subvolume = 5
PRVLC = 7
zerocount = 53
zero_TCBT = 6
zero_TCBT_min = 8
PRVLC_max = 0
The box, 1 fits in sub volume 6
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 0
zerocount 1 <= zerocount 2
subvolume = 6
PRVLC = 8
zerocount = 45
zero_TCBT = 5
zero_TCBT_min = 6
PRVLC_max = 7
bestboxinWIP is of type 1
    orientation_bestboxinWIP = 13
    subvolume_bestboxinWIP = 6
    TNSV_bestboxinWIP = 6
    case_bestboxinWIP = 2
    PRVLC_bestboxinWIP = 8
    zerocount_bestboxinWIP = 45
The box, 5 does not fit in sub volume 1
The box, 5 does not fit in sub volume 2
The box, 5 does not fit in sub volume 3
The box, 5 does not fit in sub volume 4
The box, 5 fits in sub volume 5
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 0
zerocount 1 <= zerocount 2
subvolume = 5
PRVLC = 1
zerocount = 55
zero_TCBT = 7
zero_TCBT_min = 8
PRVLC_max = 0
The box, 5 does not fit in sub volume 6
boxc1 is of type 5

subvolume_boxc1 = 5
orientation_boxc1 = 13
TNSV_boxc1 = 7
case_boxc1 = 3
PRVLC_boxc1 = 1
zerocount_boxc1 = 55
best box in WIP, type 1 is gone to pallet
PALLETIZATION UPDATE REPORT

PALLET STATUS
Volumetric Pallet Utilization, VPU = 87.34%
Number of Boxes on the Pallet: 15
1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20
Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L
8) Box Type 8
Position Vector, (L,W,H) = 25, 10, 0
Orientation: l//L and w//W
9) Box Type 6
Position Vector, (L,W,H) = 15, 28, 0
Orientation: l//L and w//W
10) Box Type 5
Position Vector, (L,W,H) = 0, 0, 30
Orientation: l//L and w//W
11) Box Type 3
Position Vector, (L,W,H) = 30, 0, 30
Orientation: l//L and w//W
12) Box Type 7
Position Vector, (L,W,H) = 0, 10, 20
Orientation: l//W and w//L
13) Box Type 6
Position Vector, (L,W,H) = 15, 28, 20
Orientation: l//L and w//W
14) Box Type 1
Position Vector, (L,W,H) = 15, 10, 30
Orientation: l//L and w//W
15) Box Type 1
Position Vector, (L,W,H) = 27, 10, 30

Orientation: l//L and w//W
Number of Empty Sub Volumes on the Pallet: 6

Sub Volume #1
Dimensions:
Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0
Sub Volume #2
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0
Sub Volume #3
Dimensions:
Length = 2
Width = 12
Height = 40
Position Vector:
Length = 45
Width = 28
Height = 0
Sub Volume #4
Dimensions:
Length = 30
Width = 2
Height = 40
Position Vector:
Length = 15
Width = 38
Height = 0
Sub Volume #5
Dimensions:
Length = 32
Width = 13
Height = 10
Position Vector:
Length = 15
Width = 15
Height = 30
Sub Volume #6
Dimensions:
Length = 8
Width = 5
Height = 10
Position Vector:
Length = 39
Width = 10

Height = 30

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 720 seconds.

There is no boxes in the WIP.
The box, 5 does not fit in sub volume 1
The box, 5 does not fit in sub volume 2
The box, 5 does not fit in sub volume 3
The box, 5 does not fit in sub volume 4
The box, 5 fits in sub volume 5
the box orientation is l//L and w//W
PRVLC_spat 1 = PRVLC_spat 2 = 0
zerocount 1 <= zerocount 2
subvolume = 5
PRVLC = 0
zerocount = 56
zero_TCBT = 8
zero_TCBT_min = 8
PRVLC_max = 0
The box, 5 does not fit in sub volume 6
boxc1 is of type 5
subvolume_boxc1 = 5
orientation_boxc1 = 12
TNSV_boxc1 = 7
case_boxc1 = 2
PRVLC_boxc1 = 0
zerocount_boxc1 = 56
boxc1 of type 5 is gone to pallet

PALLETIZATION UPDATE REPORT

PALLET STATUS

Volumetric Pallet Utilization, VPU = 91.25%
Number of Boxes on the Pallet: 16

1) Box Type 4
Position Vector, (L,W,H) = 0, 0, 0
Orientation: l//L and w//W
2) Box Type 5
Position Vector, (L,W,H) = 18, 0, 0
Orientation: l//L and w//W
3) Box Type 7
Position Vector, (L,W,H) = 0, 10, 0
Orientation: l//W and w//L
4) Box Type 6
Position Vector, (L,W,H) = 18, 0, 10
Orientation: l//L and w//W
5) Box Type 2
Position Vector, (L,W,H) = 0, 35, 0
Orientation: l//L and w//W
6) Box Type 2
Position Vector, (L,W,H) = 0, 35, 20

Orientation: l//L and w//W
7) Box Type 4
Position Vector, (L,W,H) = 15, 10, 0
Orientation: l//W and w//L
8) Box Type 8
Position Vector, (L,W,H) = 25, 10, 0
Orientation: l//L and w//W
9) Box Type 6
Position Vector, (L,W,H) = 15, 28, 0
Orientation: l//L and w//W
10) Box Type 5
Position Vector, (L,W,H) = 0, 0, 30
Orientation: l//L and w//W
11) Box Type 3
Position Vector, (L,W,H) = 30, 0, 30
Orientation: l//L and w//W
12) Box Type 7
Position Vector, (L,W,H) = 0, 10, 20
Orientation: l//W and w//L
13) Box Type 6
Position Vector, (L,W,H) = 15, 28, 20
Orientation: l//L and w//W
14) Box Type 1
Position Vector, (L,W,H) = 15, 10, 30
Orientation: l//L and w//W
15) Box Type 1
Position Vector, (L,W,H) = 27, 10, 30
Orientation: l//L and w//W
16) Box Type 5
Position Vector, (L,W,H) = 15, 15, 30
Orientation: l//L and w//W

Number of Empty Sub Volumes on the Pallet: 7

Sub Volume #1
Dimensions:
Length = 3
Width = 5
Height = 40
Position Vector:
Length = 12
Width = 35
Height = 0

Sub Volume #2
Dimensions:
Length = 1
Width = 30
Height = 40
Position Vector:
Length = 47
Width = 10
Height = 0

Sub Volume #3
Dimensions:
Length = 2
Width = 12
Height = 40

Position Vector:
Length = 45
Width = 28
Height = 0
Sub Volume #4
Dimensions:
Length = 30
Width = 2
Height = 40
Position Vector:
Length = 15
Width = 38
Height = 0
Sub Volume #5
Dimensions:
Length = 8
Width = 5
Height = 10
Position Vector:
Length = 39
Width = 10
Height = 30
Sub Volume #6
Dimensions:
Length = 2
Width = 10
Height = 10
Position Vector:
Length = 45
Width = 15
Height = 30
Sub Volume #7
Dimensions:
Length = 30
Width = 3
Height = 10
Position Vector:
Length = 15
Width = 25
Height = 30

WIP STATUS

Actual WIP surface area = 0 inches^2

CONVEYOR STATUS

TOTAL PALLETIZATION TIME = 760 seconds.

All remaining sub volumes are too small for any of the box types.
100% Pallet Utilization is not possible.•

FINAL PALLETIZATION REPORT

PALLET STATUS
Volumetric Pallet Utilization, VPU = 91.25%

Number of Boxes on the Pallet: 16

1) Box Type 4
   Position Vector, (L,W,H) = 0, 0, 0
   Orientation: l//L and w//W

2) Box Type 5
   Position Vector, (L,W,H) = 18, 0, 0
   Orientation: l//L and w//W

3) Box Type 7
   Position Vec'or, (L,W,H) = 0, 10, 0
   Orientation: l//W and w//L

4) Box Type 6
   Position Vector, (L,W,H) = 18, 0, 10
   Orientation: l//L and w//W

5) Box Type 2
   Position Vector, (L,W,H) = 0, 35, 0
   Orientation: l//L and w//W

6) Box Type 2
   Position Vector, (L,W,H) = 0, 35, 20
   Orientation: l//L and w//W

7) Box Type 4
   Position Vector, (L,W,H) = 15, 10, 0
   Orientation: l//W and w//L

8) Box Type 8
   Position Vector, (L,W,H) = 25, 10, 0
   Orientation: l//L and w//W

9) Box Type 6
   Position Vector, (L,W,H) = 15, 28, 0
   Orientation: l//L and w//W

10) Box Type 5
   Position Vector, (L,W,H) = 0, 0, 30
   Orientation: l//L and w//W

11) Box Type 3
   Position Vector, (L,W,H) = 30, 0, 30
   Orientation: l//L and w//W

12) Box Type 7
   Position Vector, (L,W,H) = 0, 10, 20
   Orientation: l//W and w//L

13) Box Type 6
   Position Vector, (L,W,H) = 15, 28, 20
   Orientation: l//L and w//W

14) Box Type 1
   Position Vector, (L,W,H) = 15, 10, 30
   Orientation: l//L and w//W

15) Box Type 1
   Position Vector, (L,W,H) = 27, 10, 30
   Orientation: l//L and w//W

16) Box Type 5
   Position Vector, (L,W,H) = 15, 15, 30
   Orientation: l//L and w//W

WIP STATUS

Number of Boxes in the WIP: 0

Actual WIP surface area = 0 inches^2

Maximum WIP surface area at any one Time = 180 inches^2

ROBOTIC PALLETIZATION TIME = 760 seconds.

# GLOSSARY

| | |
|---|---|
| Block | A combination of several boxes forming a larger three dimensional rectangular |
| Box Type Demand | The number of boxes of that type that are required to go on the pallet |
| Boxes in the System | All the boxes that can be found on the conveyor, in the WIP, and on the pallet. |
| Box Orientation | It is the way the length and width of the box are directed with respect to the length and width of the pallet. |
| Column Stacking | A pallet loading technique by which boxes are loaded column by column. |
| Guillotine Cut | Separation between two box surface adjacent to each other. |
| Heuristic | Different criteria, methods, or principles for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal. |
| Individual Box Type Sub Volume Load Capacity, ISVLC: | The maximum integer number of that box type that the sub volume can accommodate. |
| Layer Palletization | A pallet loading technique by which boxes are loaded layer by layer. |
| Mathematical Model | Approximate representation of a concept, an object, a system, or a process in mathematical terms |
| Maximum Capacity of Box Type, MCBT: | The sum of all the ISVLCs that relate to the particular box type. |

| | |
|---|---|
| Negative Flexibility | The proportion of the TZC of a particular box type to the maximum TZC |
| New Pattern | The overall configuration of the empty volume after the box is loaded onto one sub volume on the pallet. Sub volume combining is considered to produce new patterns. |
| New Sub Volume | A sub volume that still exists or is newly created right after the box is loaded onto the pallet. |
| Old Pattern | The way the empty volume is partitioned before the box is loaded onto the pallet. |
| Old Sub Volume | A sub volume that existed right before the box was loaded onto the pallet. |
| Pallet Stability | State of the pallet in which the load is supported |
| Partitioned Remaining Volume Load Capacity, PRVLC: | The sum of all the MCBTs that relate to the particular pattern. |
| Pick up Place | The location at the end of the conveyor from which the robot picks up the boxes. |
| Positive Flexibility | The flexibility of a partitioned pattern to accommodate a particular box type. |
| Priority Level | A comparison stage where boxes, sub patterns, partitioning patterns are compared. |
| Interlocked Stacking | Pallet loading technique by which boxes can be loaded anywhere on the pallet upon decision making. |
| Sub Pattern | One of the two ways that the empty volume is partitioned into sub volumes after a box is loaded onto the pallet. It is related to the two possible box orientation in its sub volume. |
| Sub Volume | Rectangularly shaped three dimensional empty space found anywhere at any given time on a pallet. |

| | |
|---|---|
| Surface Levelling | Loading boxes on adjacent columns so that they have a common height. |
| System | The palletization environment composed of the conveyor, the WIP, and the pallet. |
| Total Number of Sub Volumes, TNSV | The number of old sub volumes. (Before the box is loaded onto the pallet) |
| Work-In-Process | The holding area in which boxes that are not immediately loaded onto the pallet are temporarily stored. |
| Z-Axis of a Box | The vertical axis of a box. |
| Zero Count | A computed parameter by which ties of patterns are broken. |