Using Simulation to Assess the Effectiveness of Pallet Stacking Methods

Stephen Balakirsky[1], Fred Proctor[1], Tom Kramer[1], Pushkar Kolhe[2], and Henrik I. Christensen[2]

[1] National Institute of Standards and Technology, Gaithersburg, MD 20899-8230
{stephen.balakirsky,frederick.proctor,thomas.kramer}@nist.gov
[2] RIM@GT, CoC, Georgia Institute of Technology, Atlanta, GA 30332-0280
{pushkar,hic}@cc.gatech.edu

Abstract. Stacking objects onto pallets is the most widely used method of bulk shipping, accounting for over 60% of the volume of goods shipped worldwide. Due in part to the wide variation in size, shape and weight of objects to be palletized, successful stacking is more of an art than a science. Pallet stacking is an example of the three dimensional cutting stock problem, a variant of the combinatorial NP-hard knapsack problem. This makes it hard to find a common heuristic to satisfy requirements for major shippers and receivers. In this paper, we explore the role that simulation can play in comparing how well automated metrics can gauge the quality of a pallet. A physics enabled simulator allows us to study the interlocking among hidden internal boxes and makes it possible to study metrics that cannot be determined by mere statistics. Results from a competition held in May 2010 are used to illustrate the concepts.

Keywords: Palletizing, Simulation, Competition, Performance Metrics, USARSim.

1 Introduction

Consider the problem of distributing packed grocery items to various retailers. One possible solution is to have each class of goods (e.g. milk or cookie brand *x*) arranged by workers into individual pallets and then shipped on trucks. This operation takes a substantial amount of time and is subject to mistakes. The process becomes more efficient if more items are shipped per pallet. However, for some vendors and retailers a full pallet exceeds their total demand. This is very commonplace in manufacturing industries as the Just-In-Time and Efficient Consumer Response strategies have become popular. To solve this problem, various commercial logistics solutions allow products to be shipped in mixed pallet loads, where multiple classes of products are grouped onto a single pallet. Most of these solutions use heuristic approaches or formulate the problem as a mixed integer linear program to solve the manufacturer's bin packing problem. However the heuristics used in these problems are statistical, and there is no way to

N. Ando et al. (Eds.): SIMPAR 2010, LNAI 6472, pp. 336–349, 2010.

know if a pallet can be created at all. Since the solution is NP-hard it is hard to compare two different commercial solutions.

We employed a newly created pallet quality evaluation simulator known as Pallet Viewer and the Unified System for Automation and Robot Simulation (USARSim) framework to test the palletizing process scenario. The Pallet Viewer software allows us to evaluate various performance metrics of a pallet after each item is placed on the pallet stack. Since USARSim uses physics, we can use it to simulate the process of creating a pallet. This allows us to verify the correctness of the planner and the robot motion planning algorithms. This combination of software tools allows us to compare different pallet packing plans and motion solutions.

The remainder of this paper is organized as follows. Section 2 briefly discusses related work in this area. Section 3 decomposes and describes the entire palletizing process. Section 4 discusses our proposed metrics for evaluating the quality of a mixed pallet, while Section 5 describes the software that implements these metrics. Section 6 describes a recently run competition in which teams built both simulated and real mixed pallets that were judged by the metrics mentioned above. Finally, Section 7 presents conclusions and areas for future work.

2 Related Work

There are many commercially available palletizing solutions that offer software planners as well as complete robotic systems. Cape Pack from Cape Systems and TOPS Pro from Tops Engineering Corp[1]. are software packages that offer palletizing solutions. Almost all such solutions use statistical heuristics to determine the best pallet and present the user with these statistics to determine the pallet pattern to use. Only an expert user can make sense of such statistics. On the other hand, our system uses statistics and dynamics of a physics enabled simulation engine to guarantee the solution. Our system also uses an openly published interface which vendors can use to compare various pallet solutions.

The manufacturer's bin packing problem has also been studied in research, and there has been significant work in finding good heuristics. It is well known that the problem is fundamentally NP-hard and as such the emphasis is on use of heuristics and formalization of the problem as discussed in [1,10]. Most of the recent work has concentrated on generating stable pallets by using pre-defined pallet patterns which are know to be theoretically stable as discussed in [2,3]. Our simulation software tries to evaluate pallets on heuristics not known by the palletizer software. These heuristics weigh stability and density of the pallet equally, and are therefore suitable for evaluation purposes. In our last venture of describing this software at a conference we also showed that we can confirm

[1] Certain commercial software tools and hardware are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the authors, nor does it imply that the software tools and hardware identified are necessarily the best available for the purpose.

the simulation results in the real world. We demonstrated the pallet stacking process on a real robot cell which closely resembles our simulation environment at ICRA 2010.

3 Modeling and Analysis

Consider a manufacturing plant such as a beverage company that produces products of various kinds. When the plant receives orders from vendors, the different products are packaged and shipped to the vendors in the most efficient way possible. This process can be subdivided into:

1. *Presorting*
2. *Packaging*
3. *Storing*
4. *Retrieving*

The logistics system in a plant makes the plan to process a given order. The planner outputs a package configuration which specifies the number of pallets to be created and the configuration of each product on each of those pallets. The presorting stage consists of a series of conveyors and stops that sort the packages. The packages are sorted such that each package goes to the correct palletizing system. A palletizing system consists of several robots picking up packages from the conveyor and arranging them onto one or many stacks. A given warehouse system can consist of many such palletizer systems. The packaging stage places all the products on their respective pallets. The storing and retrieving stage is usually an independent system. However, some plants also consider the storage and retrieval heuristics in their planner.

3.1 The Palletizing System

Most palletizing systems consist of a single robot being fed by a single conveyor. The arm takes the packages from the conveyor and arranges them on a pallet. If you consider an existing manufacturing plant, any improvement in the sorting stage would mean redesigning the plant structure. The palletizing system can be improved by improving the mixed palletizing algorithm the planner uses. Even a small improvement in the planner software can make the entire process faster and more efficient. This is the reason for our interest in modeling the simplest palletizing environment in USARSim simulation software and the Virtual Manufacturing Automation Competition (VMAC).

The Robot System. The palletizing environment also depends on the type of palletizing robot and gripper being used. In general, there are two types of systems - one in which the robot can pick up individual packages and the other in which the robot picks up packages in a layer. In both of these scenarios, a suction gripper is the most commonly used gripper. If the packages are flexible, however, other grippers are used.

Modeling in USARSim. At its current development stage, the simulator can simulate multiple robot arms stacking multiple pallets while picking packages from a single conveyor. At this stage, we only model a suction gripper on each robot arm that can lift one package at a time. For the presorting stage, we allow the user to sort the order and spawn packages in the order in which the planner is supposed to feed them into a palletizing system. This means that pallets cannot be constructed in parallel, but it achieves the objective of verifying the construction of every pallet in a dynamic simulation environment.

3.2 Planning and Analysis

The entire planning process can be thought of as a three-layered hierarchical planner.

1. The uppermost level processes the order from various vendors and develops a plan for each and every package. This plan is run only once for each order. From this plan, the system knows the final position of each package. This plan takes into account various heuristics such as time required to construct the pallet, the stability of the pallet, etc.
2. The middle level coordinates the higher level planning process and the lower level motion planners. The sorting of packages through the conveyors and coordination between multiple robots sharing the same workspace is done by this planner. In the case of multiple robots, it can schedule each robot's pick from the conveyor.
3. The lowest level is a motion planner for the robotic arms. This planner makes plans for every package. For every package, it finds a collision free path from the conveyor to the final position on the pallet.

4 Metrics

4.1 Introduction

Roughly speaking, a metric for palletizing is a quantitative measure of some aspect of any of the following:

- one box that is part of a stack on a pallet
- the entire collection of boxes in a stack on a pallet
- a set of stacked pallets
- the process of building stack(s) of boxes on pallet(s).

Our work in developing metrics for palletizing has only just begun. The Pallet Viewer utility, which displays pallets stacked with boxes, calculates eleven metrics as follows. Two of these pertain to one box (the first type above). The rest pertain to a collection of boxes in a stack on a pallet (the second type above).

- Number of file errors
- Total weight
- Stack height
- Volume of boxes
- Pallet storage volume
- Volume density
- Package N overlap fraction
- Package N connections below
- Pallet average overlap fraction
- Pallet average connections below
- Current file error count

These are described further in Section 5.1. They are automatically updated and displayed on the screen each time a box is added to the stack or removed from it. The program gets its information from an order form, a plan for a pallet, and a description of the finished pallet. While the Pallet Viewer displays both the pallet as planned and the pallet as built, the metrics are calculated only for the pallet as planned.

We could readily calculate more metrics, but we would like to calculate only those that are important in industry. Towards this end, we have studied the web sites of major shippers and collaborated with a supplier, Kuka Logistics, of an automated palletizing system.

Further study and additional collaborations with commercial users and vendors of palletizing systems will be needed to ensure that our metrics are useful in practice. In addition, we will need to devise a weighting scheme for combining metrics. Different weighting schemes may be required for different operating environments.

4.2 Weight Support Mode

There are at least two different common modes for the way in which boxes are supported by other boxes. What constitutes a good stack is very different between the two.

In one mode, which we might call the *cardboard* mode, when anything (such as another box) is put on top of a box, it is the material from which the box is made that bears the weight. The *cardboard* mode occurs with boxes containing breakable items such as television sets and glassware. These boxes may have additional internal supports such as columns at the corners.

In the other, *contents* mode, it is the contents of the box that provide most of the support for any weight placed on top of the box. The *contents* mode occurs with boxes containing relatively robust items such as cans of soda or reams of paper. In the *contents* mode, the material from which the box is made may have very little resistance to compression. The contents have it, instead.

The goodness or badness of metrics such as how much boxes overlap varies widely between the two modes. In *cardboard* mode, overlap is usually a bad thing because the edges of the boxes (the load-bearing part) are not lined up vertically. One web site [6] says overlap (also called interlocking) “can destroy up to 50% of the compression strength”. In *contents* mode, overlap is generally good, since it tends to hold the stack together. The edges of the boxes are not load bearing in *contents* mode. All shippers expect pallets to be wrapped with plastic and possibly also with straps. The wrapping and strapping hold the stack together and fasten it to the pallet, so the benefits of overlap are usually not large.

The work we have done so far has used pallets of boxes of supermarket items. These are almost all *contents* mode boxes, so overlap has been looked upon favorably.

4.3 Box Robustness

Boxes of different items have different maximum loads. No box on a pallet should have its maximum load exceeded. The XML schema for boxes we have been using, Article.xsd [7], includes an optional Robustness element, which is intended to represent maximum load. Robustness has as sub-elements: MaxPressureOnTop, SourcePalletLayers, and RelativeRobustness. The last two of these are optional. The pallet metrics we have devised thus far do not include using Robustness.

MaxPressureOnTop is given in grams per square millimeter.

SourcePalletLayers, the number of layers on a pallet of identical boxes on which a product is received, is an empirical measure of robustness. If the manufacturer piled boxes N layers deep for shipment, and they arrived intact, it should be OK for the builder of a mixed pallet to do likewise.

RelativeRobustness is an enumeration of VeryWeak, Weak, Normal, Strong, and VeryStrong. This is in order of increasing robustness, but no quantitative meaning has been assigned.

4.4 Additional Metrics

Several additional metrics may be useful, as follows.

The four metrics listed above for connections and overlap are currently calculated only for the bottoms of boxes. It may be useful to calculate them for the tops as well.

The point of finding connections and overlaps is to get a measure of how well the stack holds together. It may be useful to calculate a more direct measure of holding together, such as the number of connected sub-areas on the top side of a layer (fewer being better). Intuitively, a sub-area is formed by a number of boxes in a layer being linked together by boxes in the layer above.

It may be useful to calculate the center of gravity (COG) of the stack of boxes. The main concern with the COG is that the pallet may be hard to handle if the COG is high or is off to one side of the pallet. Assuming that the COG of each box is at its center, the COG of the stack should be easy to calculate.

To unload a pallet efficiently, it should be possible to unload all boxes that have the same destination at once without having to move boxes that have a different destination. To support this idea, the schema Article.xsd defines Family as an element of Article. A metric could be calculated giving the number of Families that can be unloaded from a pallet without moving other boxes.

Overhang is a measure of the stack extending outside the pallet. It might be calculated as the sum over the four sides of the pallet of the maximum distance any box extends past the side. Many shippers, including the US Postal Service [8] and UPS [9], require that there be no overhang. The Great Little Box web site [6] says overhang is bad: “With as little as 1/2 inch hanging-over, as much as 30% of their strength is lost!". In some situations, however, overhang may be acceptable or desirable.

It will be useful to calculate the number of planning errors automatically. Planning errors include having boxes floating on air, planning to put a box on the stack before all the boxes it rests on are on the stack, having boxes intersect, failing to put all the boxes on the pallet that belong there, and putting boxes on the pallet that do not belong there. There may also be syntax errors in the XML file representing the plan; non-fatal errors could be counted.

4.5 Other Considerations of Metrics

We have not yet tackled metrics for multiple pallets. For example, if more than one pallet is required, is it going to be better to have roughly equal loads on all pallets than to load all but the last pallet full and put the remaining boxes (which may be few in number) on the last pallet?

For some metrics, such as determining automatically whether loading paths are collision-free, it may be necessary to incorporate a solid modeler.

The representation of a plan we are currently using allows placing a box in only two orientations – bottom down with sides parallel to the sides of the pallet. It would be practical to allow a wider range of orientations, but calculating metrics in that case will probably require using a solid modeler.

It is practical to treat differences of a millimeter or so in height as unimportant in building stacks of boxes of the size typically stacked on pallets. However, it is not clear how to model a stack with non-zero but negligible height differences or how to calculate metrics in this case.

A metric for spacing between the sides of boxes should be devised. Some automatic planning systems leave spaces between the sides of boxes and have the edges of every layer line up with the edges of the pallet. An example of this is shown in Figure 1(a). Other planning systems leave as little space as possible between the sides of boxes (see Figure 1(c)). Some boxes may be safely manipulated by suction grippers holding on to the top of the box. For other boxes, side and/or bottom gripping is necessary. In the latter case, it may be necessary to leave spaces between boxes for the gripper.

5 Evaluation of Metrics through Simulation

5.1 Pallet Viewer

The Pallet Viewer utility displays in a 3D color view a pallet and the as-planned stack of boxes on it. The figures in this paper are all images produced by Pallet Viewer. The Pallet Viewer executable is called with at least two arguments: the name of an order file and the name of a plan file. A third, optional, argument may also be given: the name of an as-built file. If the third argument is given, the Pallet Viewer shows the as-built stack in a color wire frame view off to the side of the as-planned stack. In addition, the Pallet Viewer calculates and displays eleven metrics for the as-planned stack. These are:

- Number of file errors - total number of overlap errors in the file.
- Total weight - total weight of boxes on the pallet.
- Stack height - from the top of the pallet to the highest point on the stack.
- Volume of boxes - total volume of all the boxes on the pallet.
- Pallet storage volume - area of the pallet times the stack height.
- Volume density - volume of boxes divided by the pallet storage volume.
- Package N overlap fraction - fraction of the area of the bottom of box N that sits directly on the pallet or the top of some other box.
- Package N connections below - number of boxes that box N is directly on top of.
- Pallet average overlap fraction - average of the overlap fraction of all boxes on the stack.
- Pallet average connections below - average over all boxes on the stack of the number of connections below. For boxes that are directly on the pallet, the number of connections is 1. For boxes that are on other boxes, the number of connections is the number of such other boxes.
- Current file error count - number of overlap errors up to and including the most recently placed box.

The stack of boxes shown in Pallet Viewer may be built or unbuilt by using keyboard keys. Each time the g key is pressed, the next box is added to the stack (until there is no next box). Each time the f key is pressed, the last box is removed from the stack (until no boxes are left). The eleven metrics are recalculated each time a box is added or removed.

The Pallet Viewer is a C++ program using OpenGL graphics. Manipulating the view is done entirely with the mouse, except that the h key returns the view to its default position. The stack may be rotated and translated. The view may be zoomed in and out.

The Pallet Viewer program is useful for analyzing the process of creating a pallet. Since metrics are calculated for placement of every box on the stack, the software can assess whether the stack will remain stable when it is being constructed. Another important aspect of palletizing is the order in which boxes are put on the stack. By observing the Pallet Viewer display an expert user can to determine if the planner is making motion planning hard or impossible.

5.2 USARSim

The Pallet Viewer application is used to evaluate the quality of a pallet plan based on the geometry of the objects and their placement. Its metrics apply to the geometry of a planned pallet and answer the question, “will this be a good pallet when built?” These metrics don’t answer the questions, “how well does this pallet lend itself to being built?” or “when the pallet was actually built, how good was it?” It is a static evaluation that does not evaluate dynamic effects such as objects sliding, tipping or being crushed. It also does not evaluate the pallet building process; there may be many ways to stack objects onto a pallet to satisfy the plan, and some are more efficient than others.

An intermediate step toward dynamic simulation is to use Pallet Viewer to apply static quality metrics after each object has been stacked, so that problems with the intermediate condition of a pallet can be detected. However, since Pallet Viewer does only rudimentary geometric analysis, problems such as objects sliding will not be detected.

We use simulation to supplement Pallet Viewer's static evaluation with dynamic evaluation. There are two aspects of dynamic evaluation: qualitative visualization by an expert, and evaluation of the final as-built pallet that may be different from the planned pallet due to object slipping, tipping, crushing or misplacement.

A shortcoming of dynamic evaluation is that the source of problems can't be easily isolated to the pallet building process or the pallet plan itself. For example, if an object slips or tips, it could be due to an improper stacking order as chosen by the robot controller, or it could be due to the plan itself, and no stacking order would fix the problem.

We use the Unified System for Automation and Robot Simulation (USARSim) [5] to evaluate the pallet build process and resulting built pallet. USARSim runs in real time. As a consequence, the evaluation takes as long as the pallet build process, typically on the order of tens of minutes. This makes it unsuitable as a way to compare many different object stacking plans to quickly select the best plan for execution.

The physics simulation within USARSim is plausible, and includes effects such as friction and gravity that will make sliding and tipping problems evident. However, the fidelity of the simulation does not equal the real world, and lacks the capability to represent continuum mechanics effects like compression or buckling of objects on the pallet, or bending of the robot arm as it moves throughout the work volume. It should be considered as a tool to get figures of merit on the object stacking sequence and intermediate steps toward the complete pallet.

In addition to quantitative metrics that can be applied to physically plausible behavior in simulation, the USARSim visualization can be watched by an expert person who can make subjective observations. Such observations include statements like, "the pallet looks good, but when you try to build it, it falls over before you get finished," or "with the boxes arranged like this, the robot can't reach the far side without knocking over these boxes." Subjective statements like these are unsuitable for metrics, but they can be used as a check that the scores from quantitative metrics match up with an expert's opinion. If, for example, a metric scores a pallet highly when an expert disagrees, the source of the discrepancy can be tracked down and the metric can be revised.

Practical experience with USARSim has brought to light several issues that static analysis of ideal pallets cannot find. These include:

- objects being placed outside the robot's work volume, or at locations or orientations that are awkward for the robot to reach
- objects being knocked off by collisions with an object to be placed or the robot arm itself

– excessive motion resulting from conservative approach- and retract-points that could be optimized
– temporary sensitivity of objects to perturbations before they are stabilized by objects placed atop them

Through experience garnered by visual simulations, we hope to improve the pallet metrics so that they can be easily applied and yield results consistent with expert opinions.

6 ICRA Virtual Manufacturing Competition

The first real trial of the metrics being developed for this effort occurred during the 2010 Virtual Manufacturing Automation Competition (VMAC) [4] that was part of the IEEE International Conference on Robotics and Automation (ICRA) robot challenge. During this event, three different palletizing approaches were evaluated through the use of our metrics. These approaches included a university created neural network learning-based approach, a university created deterministic planning approach, and a commercial product that is commonly used by industry.

Participants were required to accept an eXtensible Markup Language (XML) based order file and to generate an XML based packing list. The order file contained information on the pallet (e.g. pallet size, maximum load, maximum

Table 1. Results of the 2010 Virtual Manufacturing Automation Competition. Both the Neural and Deterministic teams experienced errors in their plans for round 1, pallet 1. These errors were corrected by hand and the scores of our metrics were recomputed in round 1, pallet 1*. The Neural team was unable to produce any pallets for round 2 while the Deterministic team was unable to produce more than one pallet per order file and did not place all of the packages in round 2. Round 1's order file contained 40 packages and round 2's contained 79 packages. Entries containing a '-' indicate that the algorithm was unable to find a solution.

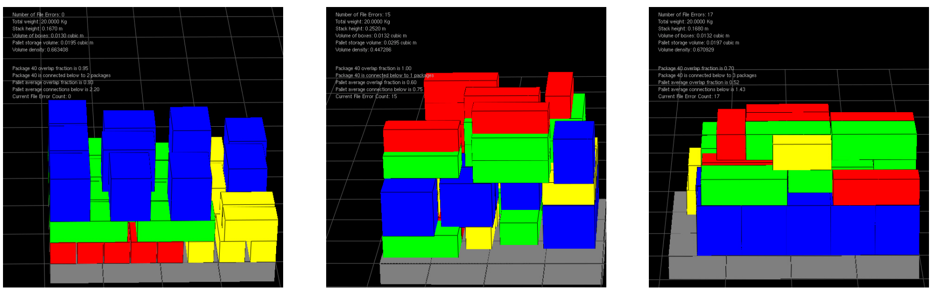
VMAC 2010 Results								
Round	Pallet	Team	Errors	Time (s)	Height (m)	Density	Overlap	Connections
1	1	Neural	15	2054	0.252	0.447	0.60	0.75
		Deterministic	17	58	0.168	0.671	0.52	1.43
		Commercial	0	10	0.167	0.663	0.93	2.20
	1*	Neural	0	2054	0.252	0.447	0.93	1.32
		Deterministic	0	58	0.168	0.671	0.92	2.07
		Commercial	0	10	0.167	0.663	0.93	2.20
2	1	Neural	–	–	–	–	–	–
		Deterministic	19	1500	0.403	0.68	0.64	1.00
		Commercial	1	28	0.375	0.79	0.90	2.26
	2	Neural	–	–	–	–	–	–
		Deterministic	–	–	–	–	–	–
		Commercial	0	NA	0.083	0.23	0.99	1.00

stacking height) as well as various package classes (e.g. package size, weight, fragility, family) and the quantity of each class of package that was required. The team-generated packlist file was required to specify the sequence in which the packages should be sent to the palletizing system as well as each package's final 4 degree-of-freedom (4-DOF) resting location (x, y, z, and yaw) and a series of 4-DOF approach points to guide the robot arm in package placement. The teams were also notified of the metrics that would be used to judge the competition.

The desired approach was to run all three algorithms on increasingly difficult order files. After each round, the teams would be allowed to build their pallets in simulation using the USARSim framework. Teams that were successful in simulation would then be allowed to construct their pallets on our 1/3 scale real palletizing cell. The results from the two rounds of the competition are shown in Table 1. As may be seen from this table, only the commercial software was able to place the full complement of boxes on pallets for both rounds. The Neural software never converged to a solution for the larger package order, and the Deterministic software was unable to plan for a second pallet that would be required to accommodate all of the packages. Due to only having one successful algorithm in round 2, order files of higher complexity were not attempted.

6.1 Round 1 Details

All of the teams successfully created packing plans for the round 1 order. Details of the results of applying the metrics are shown in Table 1 and images of the solutions are shown in Figure 1. As shown in Table 1, the Neural and Deterministic teams suffered from multiple errors that significantly affected their overall performance. These errors were due to the placement of packages in either a manner that would cause collisions in a real implementation of the plan or left the package suspended in mid-air. During the determination of the scores of our



(a) (b) (c)

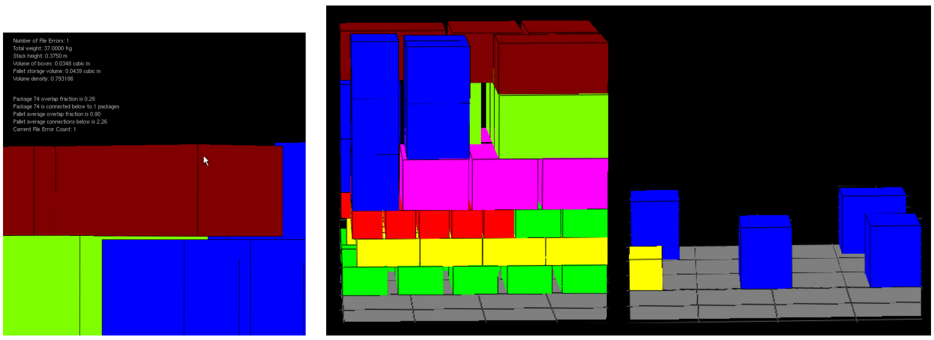
Fig. 1. Results from "round 1" of the competition which consisted of 40 packages to stack. Image (a) depicts the pallet produced by the commercial software package. Image (b) depicts the pallet produced by the neural network learning algorithm, while image (c) depicts the pallet produced by the deterministic approach.

metrics, packages which caused errors were given reduced credit (based on the portion of the box in error) for the computations. Table 1 Round 1, pallet 1* shows the improvement in scores that is possible if the package placements are adjusted slightly to avoid collisions. The remainder of this section will examine the corrected plans and the resulting higher scores.

From an execution time perspective, the commercial solution is a clear winner as it is close to six times faster than its nearest competitor. However, if time is discounted as a major scoring criterion, the results become much less clear. The scores for density, overlap, and package connections for both the commercial package and the deterministic algorithm are very similar. However, this similarity of scores using our metrics does not appear to indicate that the constructed pallets are similar. Figure 1(a) shows that the commercial system has distributed the top layer of packages over the entire surface of the layer while Figure 1(c) shows that the deterministic algorithm has concentrated the top layer towards the center of the pallet. This difference in the distribution of weight over the pallet will affect pallet stability and is not captured by the currently implemented set of metrics. The discrepancy between the difference in the pallet appearance and the scores on the metrics show that more work is required to develop a set of metrics that captures all aspects of the constructed pallet. In fact, a palletizing expert will need to be consulted to advise which pallet should indeed be deemed superior. Through the use of the pallet viewer software, we are able to easily view and share these results without the need for costly construction and evaluation of actual pallets.

6.2 Round 2 Details

The round 2 pallet order contained too many packages for a single pallet. The commercial software again computed a solution while the two university



(a) (b)

Fig. 2. (a) Error or no error? The metrics software designates an error for the top right package since it does not sit on both of its support packages. In reality, this box would probably lean to be supported by both of its supports. (b) The commercial software constructed these two pallets to solve the round 2 problem.

algorithms had more difficulty. The neural network learning algorithm was unable to converge to a solution and the deterministic approach was only able to complete the first of the two pallets. However, this round of the competition also raised some interesting problems for our metrics. Figure 2 shows the package that caused an error to be generated for the commercial system. In Figure 2(a) a slight gap can be observed between the top package and the supporting package on its right. Under real conditions, this package would simply "settle" onto its supporting package and have a slight tilt. The question that needs to be answered is if this arrangement will be stable or will cause the box to slide off of the pallet. Construction of the pallet by USARSim and the evaluation of the as-built pallet can answer this question.

Figure 2 shows the two pallets that were constructed for this order by the commercial software. The first pallet's height was maximized, while the second pallet was left mostly empty. In addition, the first pallet's construction contained numerous "towers" of packages where each package was only supported by a single package (no interlocking). The currently defined metrics were unable to judge the joint quality of this solution and instead were forced to judge the pallets individually.

7 Conclusions

We have adopted a model of palletizing, used it to define metrics, and held a palletizing competition. A Pallet Viewer utility has been built that calculates the metrics, displays them, and displays a naive simulation of executing a palletizing plan. The USARSim system has been used to simulate execution of a palletizing plan more realistically and produce an as-built description of a stack of boxes on a pallet. We have collaborated with industrial partners to ensure that our work is practical.

While we have accomplished much, more remains to be done, and we plan to continue our work. We are aware of quite a few additional metrics that might be calculated. We need to consult with more commercial firms to determine what metrics are important in what environments.

The palletizing model we are using is limited in the following ways:

- It does not support loading more than one pallet at a time.
- It supports placing a box in only two orientations.
- It mixes the design of a finished pallet with the plan for producing the design, which allows little freedom in the way in which the plan is realized.
- It does not provide for using any buffer space where boxes might be stored temporarily during palletizing.
- It does not allow for using more than one robot to load a pallet.
- It does not allow for expressing a constraint on the minimum space between boxes (which may be necessary for automatic equipment).

We need to improve our palletizing model so that it addresses these limitations.

References

1. Baltacioglu, E., Moore, J.T., Hill Jr, R.R.: The distributor's three-dimensional pallet-packing problem: a human intelligence-based heuristic approach. International Journal of Operational Research 1(3), 249–266 (2006)
2. Bischoff, E.E.: Stability aspects of pallet loading. OR Spectrum 13(4), 189–197 (1991)
3. Carpenter, H., Dowsland, W.B.: Practical considerations of the pallet-loading problem. Journal of the Operational Research Society 36(6), 489–497 (1985)
4. VMAC Executive Committee: Virtual manufacturing automation competition homepage (June 2010), http://www.vma-competition.com
5. USARSim Community. Usarsim wiki homepage (June 2010), http://usarsim.sourceforge.net/wiki/index.php/Main_Page
6. Great Little Box Company. Palletization (June 2010), http://www.greatlittlebox.com/packaging/storing
7. Kuka SysTec GmbH. Xml schema for article (June 2010), http://moast.cvs.sourceforge.net/viewvc/moast/devel/src/manufact/sample%Data/Schema/Article.xsd?view=markup
8. US Postal Service: Special standards - pallets, pallet boxes, and trays on pallets (June 2010), http://pe.usps.com/text/QSG300/Q705b.htm
9. UPS. Ups: Preparing your freight shipment (June 2010), http://www.ups.com/content/us/en/resources/ship/packaging/prepare/freig%ht.html
10. Yaman, H., Sen, A.: Manufacturer's mixed pallet design problem. European Journal of Operational Research (January 2008)