Amazon Multi-Order Automation Project

Wenhao Zhang, Verina Dinata, Lucas See, Matt Lunde, Zach Lawless, Tianyu Ge

Background

A typical Amazon Fulfillment center processes over a million orders a week, moving items from picking, to packaging, and finally shipping to the customer. Of these processes, packaging is perhaps the most complex and labor intensive, requiring workers to determine item placement, place items, and close the box. While automation solutions exist for single order item packing, multi order packaging remains a manual process due to the difficulty of determining optimal item placement for any given order combination.

Amazon's current fulfillment center infrastructure is a set off malleable pieces and processes that is constantly being upgraded to reflect the latest and greatest technology. With the introduction of programmable industrial robots at competitive prices, Amazon has a new opportunity to make major changes to the way items reach their customers from several perspectives. Current proven technology and new technology can be combined to create highly autonomous systems that are scalable and consistent throughout a large fulfillment center network.

Problem Statement

Logic system

Develop a digitized logic system capable of guiding a robot to replicate the human item packing process.

Infrastructural Change

Investigate the infrastructure changes required to facilitate a robotic packing cell.

Summary of 215,000 Amazon Orders

AWS ORDERS DATASET

Here are the orders dataset from the AWS orders information at BFI4

Dataset Information
Orders ID, items dimension, recommended box dimension, units

215,093
Number of Multi-Orders

528,649
Number of Items within Orders

49.21%
The average usage rate of box space

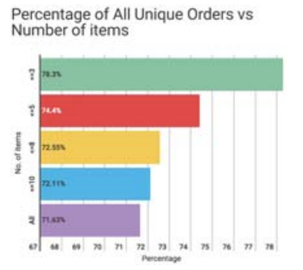
2.46
The average number of items within multi-orders

1.25%
The volume of recommended boxes is smaller than the sum of items volume (2702/215093)

50.73 %
The percentage of orders can be packed without stacking. (Run by LFFP algorithm, dimension of dataset=2004)

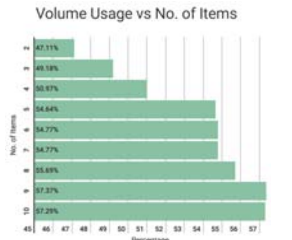
Percentage of Orders With All Unique Items

Percentage of All Unique Orders vs Number of items

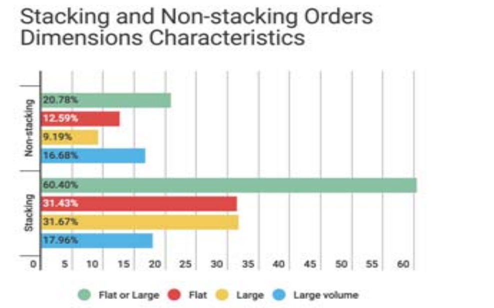


Number of Items Within Orders

Volume Usage vs No. of Items



Stacking and Non-stacking Orders Dimensions Characteristics



Interpretation: The nature of stacking and non-stacking orders is determined by LFFP algorithm. Large area items are items with bottom area larger than 60% of the box bottom area; flat items have length/height and width/height ratios larger than 7; Large volume have volume larger than 34% of the box total volume.

Findings:

- Multi-orders have low item count and high item variety
  - 76.82% of the Multi-orders contains only 3 or less units and 78.3% of the times they are all unique
- Multi-orders with small amount of units have less box space usage rate
  - Orders with 2 units has volume usage rate of 47% and orders with 10 units have volume usage of 57%
- Order stacking determined by surface area of items
  - 60.40% of the stacking orders are made up of large or flat items. Only 20% of non-stacking contains large or flat items
  - Compared with the face area, volume is not a significant factor influencing stacking. (16.68% vs 17.96%)

Observation of Human Packing Process

Data Collection and Analysis:

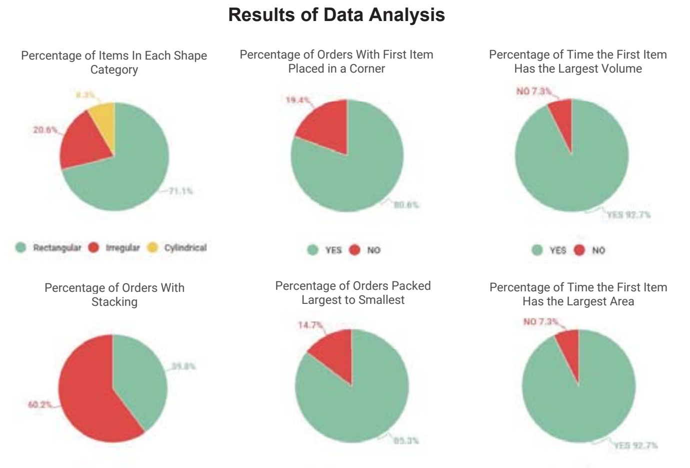
In order to understand the human packing process, data was collected from 191 observed human packaged orders. This data was then analyzed for trends defining the human packing procedure which then served to guide our algorithm development.

Data Collection Sheet

1-Largest | S-Smallest | R-Rectagular | C-Cylindrical | I-Irregular
*largest item based on the SA of the side placed at the bottom
**all Ns under the First item placed in corner involve using an self-built box

Order Number	# of Items in order	First item placed in corner	Items packed largest to smallest	Stacking	Size compared to input order	1st item	2nd item	3rd item
1	3	Y	Y	N	Area	1	1	1
					Volume	1	1	1
					Type of items	R	R	R
2	2	Y	Y	N	Area	1	2	
					Volume	1	2	
					Type of items	I	I	
3	3	Y	Y	Y	Area	1	1	2
					Volume	1	1	2
					Type of items	R	R	R
4	2	N	Y	Y	Area	1	1	
					Volume	1	1	
					Type of items	R	R	

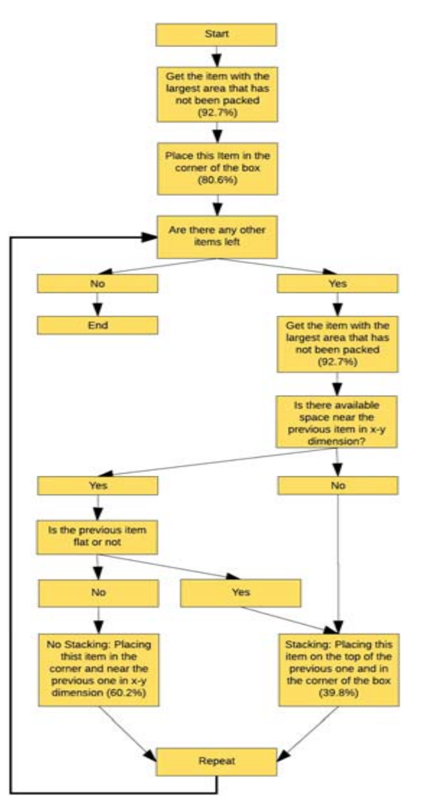
Results of Data Analysis



Findings:

- 71.1% of the items in an order are rectangular
  - ➔ Items are treated as rectangular
- 80.6% of the time, the first item is placed in the corner of the box
  - ➔ The first item is always placed in the corner of the box
- 85.3% of the time, the largest item is packed first followed by smaller ones
  - ➔ Items are packed from largest to smallest
- 39.8% of the time, stacking is used.
  - ➔ Stacking is used to save box space

Human Packing Flowchart



Exceptions to Heuristics

- Some packers have "muscle memory" leading them to pack in certain patterns
- Packing rate correlates with the experience that a packer has, and certainly has a quantifiable effect on the pack itself
- When two or more items have the same or similar size, packers are more likely to pack the rectangular one first instead of cylindrical or irregular items
- If the volume utilization of a pack is very low, the packer does not neccessarily adhere to the flow chart

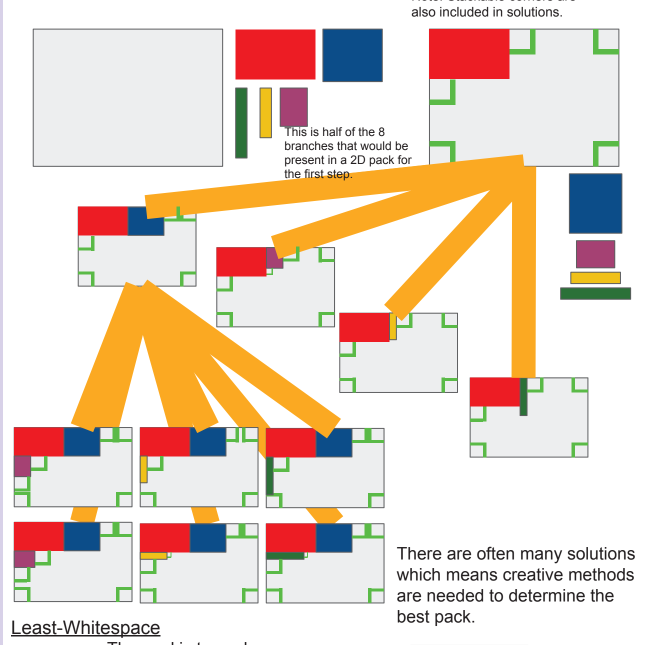
Digitized Logic Solutions

Several approaches were decided upon to approach the problem of translating the heuristic method observed by a human packer to a robotic system. These solutions take into account information gathered from our observation. Ideas like: placing the biggest item first and the smallest last, always looking to place an item in a corner, and trying to leave one larges space for dunnage were kept in mind. Our solutions were coded in Java and Matlab, and represent a significant part of our project as we have created solutions that can be used by Amazon as baselines going forward.

Tree-Pack and Whitespace

(1) Place the first item in the grid
(2) Locate each packable corner as well as corners that can be stacked on(not pictured)
(3) Make a node for each item being placed in each corner for each possible orientation
(4) Repeat steps 2 and 3 until you have created an exhaustive tree of all possible ways to pack the box with the items
(5) Repeat for "n" number of boxes. Run each finished node through custom processing to determine which pack you want

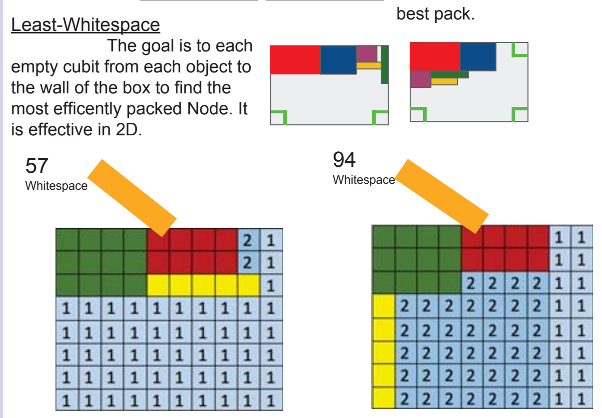
Note: Stackable corners are also included in solutions.



There are often many solutions which means creative methods are needed to determine the best pack.

Least-Whitespace

The goal is to each empty cubit from each object to the wall of the box to find the most efficently packed Node. It is effective in 2D.

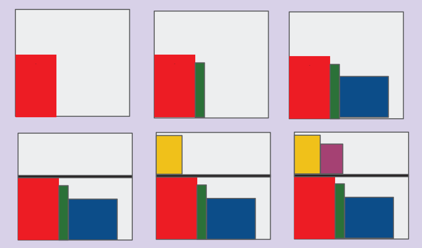


Column Packing Example

(1)Permute item orientations to generate 3^(number of items) item sets
(2) Sort each permuted set from largest to smallest Y-Dimension
(3) Run each order permutation through each of Amazons 21 boxes, recording fit/no-fit
(4) Compare the possible solutions based upon box size
(5) Output box size, item centerpoint, and orientation of best pack solution

Permuted Item Set 1

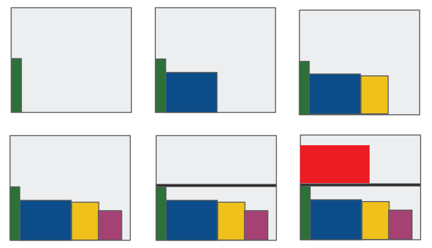
Solution process, Permuted Item Set 1, Box 1



Repeat algorithm on additional 20 boxes

Permuted item set 2

Solution process, permuted item set 2



Repeat algorithm on additional 20 boxes

Repeat process on all permutated solutions

2D LFFP Example

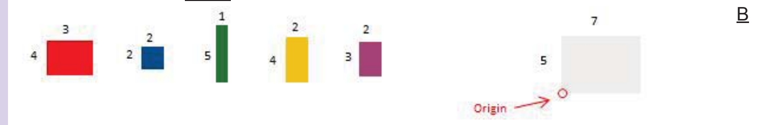
2-Dimensional Less Flexibility First Principle Example:

- For 2D a corner-occupying placement move (COPM) will be defined as:

(longer item dimension, shorter item dimension, orientation, x location, y location)

- The orientation variable holds a 0 if the longer item dimension is horizontal and holds a 1 if the longer item dimension is vertical
- Break ties with largest area utilization

Items



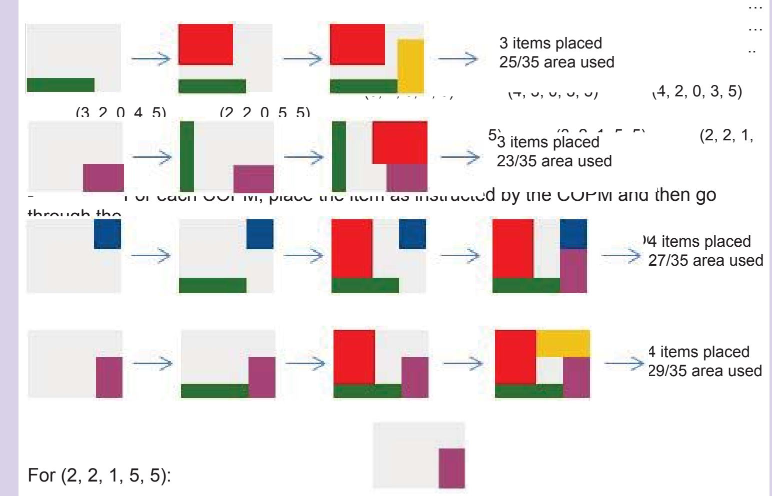
Iteration 1:

Corners: (0, 0) (7, 0) (0, 5) (7, 5)

COPMs: (5, 1, 0, 0, 1) (4, 3, 0, 0, 3) (4, 2, 0, 0, 2) (3, 2, 0, 0, 2) (2, 2, 0, 0, 2)

n=40 (5, 1, 1, 0, 5) (4, 3, 1, 0, 4) (4, 2, 1, 0, 4) (3, 2, 1, 0, 3) (2, 2, 1, 0, 2)

…



For each COPM, place the item as instructed by the COPM and then go through the …

For (2, 2, 1, 5, 5):

Best COPM ends up being (3, 2, 1, 5, 3):

Thus the first item is placed as such:

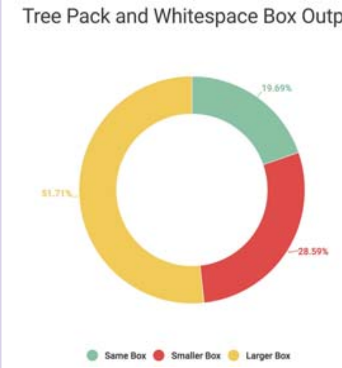
Algorithm Comparison: Performance Analysis

To test the performance of these algorithms, each was run on a dataset containing the box that was recommended by the current system. Data was collected on the box size used by the created algorithms as compared to Amazon's recommenc

$$Volume\ Difference = Volume\ of\ Box\ Used - Volume$$

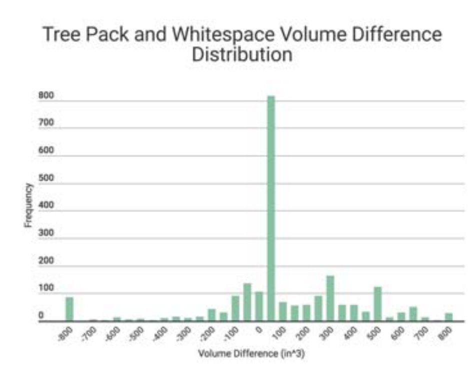
Thus a negative value for volume difference means that volume was saved.

Tree Pack and Whitespace Box Output

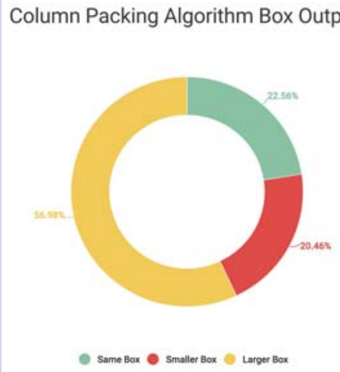


- Approximately 51.7% of orders required larger box size
- Pending the acquisition of more processing power, we can run this complex algorithm on more boxes and more possible solution branches and get closer to the realistic expectations for this algorithm

Tree Pack and Whitespace Volume Difference Distribution

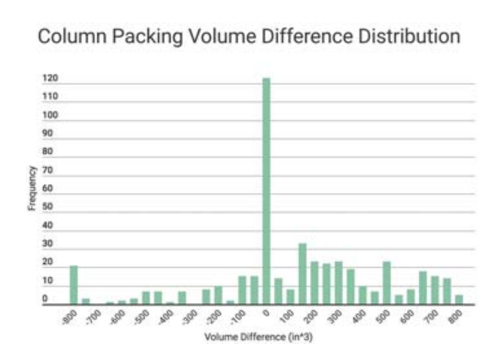


Column Packing Algorithm Box Output

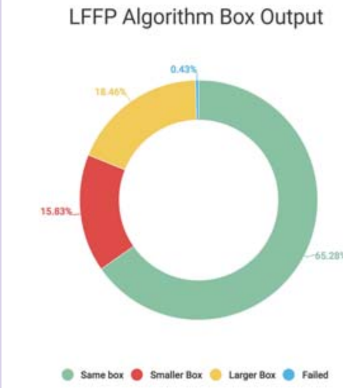


- Approximately 57% of orders required larger box size
- Likely due to primary weakness of algorithm: not packing multiple items in same row

Column Packing Volume Difference Distribution

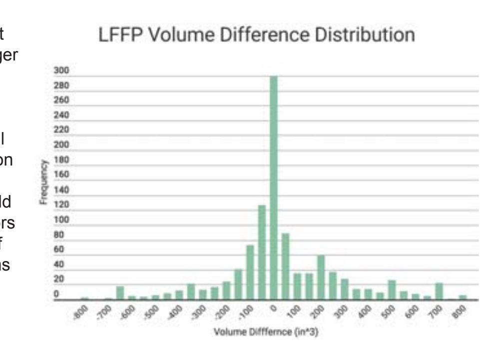


LFFP Algorithm Box Output

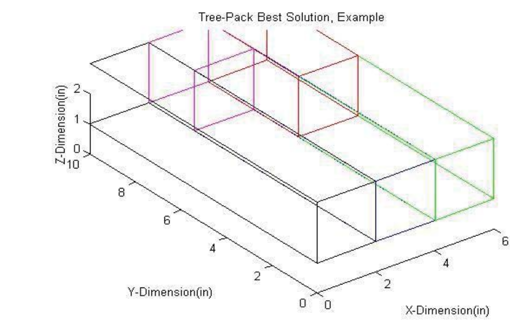


- Based on a smaller dataset about 50% of the orders that used a larger box and all of the instances of failure were due to a poorly recommended box meaning that there was no possible way to fit all of the items in the box that Amazon recommended
- Another 48% of larger boxes could be attibuted to cubit rounding errors
- The last 2% were due to a lack of the algorithms ability to stack items where overhang occurs

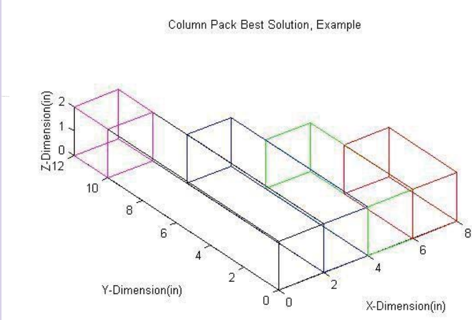
LFFP Volume Difference Distribution



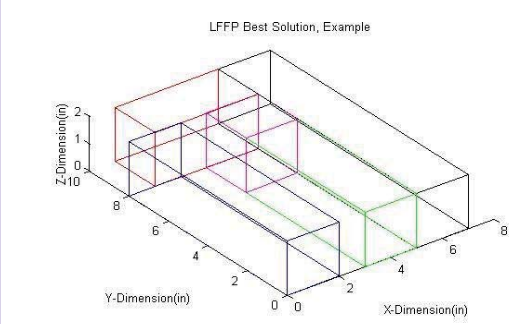
Algorithm Comparison: Test Case



Comments
- Generated pack minimizes the empty space between items
- Results in the tightest possible pack
- Greatest difficulty comes from selecting the 'best' solution from the set of all possible solutions



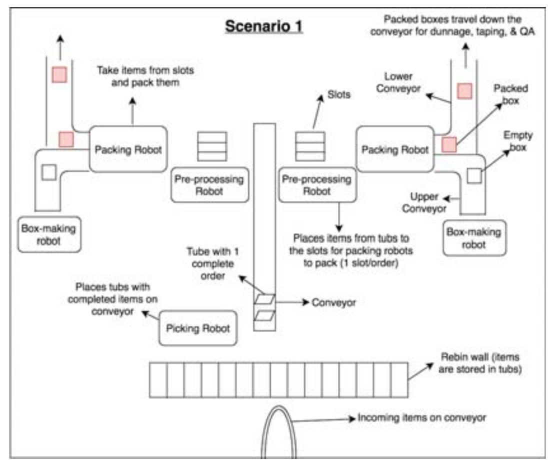
Comments
- Volume utilization lower for orders composed of unique items
- Does not place multiple items in Y direction in single row
- Performance increases for orders containing repeat items

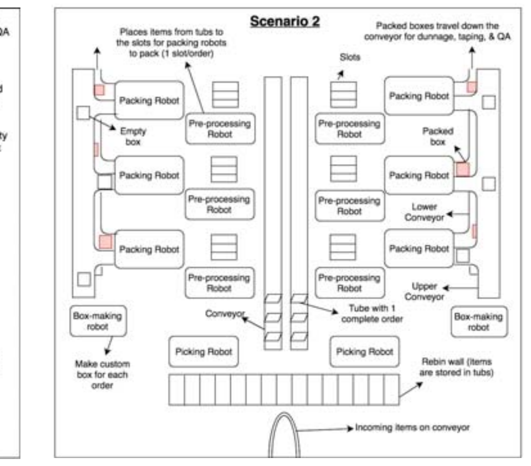


Comments
- Places items by examining each possible cornerpoint placement
- Gap in pack due to multiple possible solutions with same box size
- Solution selected based on the fit of items in box rather than the placement inside box

Potential Infrastructure Changes

To successfully implement the packing robots at the fulfillment center, some infrastructure changes are needed to accommodate their abilities. Two ideal scenarios (shown below) are generated. Both scenarios start from the incoming items on conveyor to the departure of packed boxes for dunnage, QA, & sealing. All the processes in between them are expected to be autonomous since we try to minimize the number of touches from humans.





Economic Feasibility

The purpose of performing an ROI analysis is to measure the rates of return of the money invested and also to determine the number of human packers that can be replaced when we implement the suggested scenarios. It is also used as an indicator to compare the 2 scenarios. Sensitivity analysis is performed to the ROI analysis to determine how much the ROI and number of packers eliminated/2 shifts change when the human packing rate and robot packing rate vary. We allow an increase and decrease of 10% to both the human and robot packing rate and compute the new results for each case.

Data used for ROI analysis:

- Capital cost for picking robot: $11k/robot
- Capital cost for pre-processing robot: $20k/robot
- Capital cost for box-making robot: $800k/robot
- Capital cost for packing robot: $150k/robot
- Standard wage rate: $22.23/hr
- Human packing rate: 214 units/hr
- Robot packing rate: 381 units/hr

Assumptions:

- Cost of conveyor = $0
- Engineers' pay, maintenance cost, technology update = $0
- 10 hr/shift, 2 shifts/day, 6.5 days/week, 52 weeks/year
- Packing robots & human packers will be the bottleneck

Summary of ROI Analysis

		Scenario 1	Scenario 2
Number of Picking Robot		1	2
Number of Pre-processing Robot		2	6
Number of Box-making Robot		2	2
Number of Packing Robot		2	6
Total Cost		$1,951,000.00	$2,642,000.00
Current Packing Rate (units/hr/packer)		214	214
Robot Packing Rate (units/hr)		762	2286
Annual wage/packer		$75,137.40	$75,137.40
Without Sensitivity Analysis	Numbers of packers eliminated/2 shifts	7.12	21.36
	Annual savings per site	$535,090.64	$1,605,271.93
	ROI	4.42	1.80
+10% Human Packing Rate	Numbers of packers eliminated/2 shifts	6.47	19.42
	Annual savings per site	$486,446.04	$1,459,338.12
	ROI	4.90	1.99
-10% Human Packing Rate	Numbers of packers eliminated/2 shifts	7.91	23.74
	Annual savings per site	$594,545.16	$1,783,635.48
	ROI	3.93	1.61
+10% Robot Packing Rate	Numbers of packers eliminated/2 shifts	7.83	23.5
	Annual savings per site	$588,599.71	$1,765,799.12
	ROI	3.98	1.63
-10% Robot Packing Rate	Numbers of packers eliminated/2 shifts	6.41	19.23
	Annual savings per site	$481,581.58	$1,444,744.74
	ROI	4.95	2.01

4.42 Years
The Return On Investment of Scenario 1

1.8 Years
The Return On Investment of Scenario 2

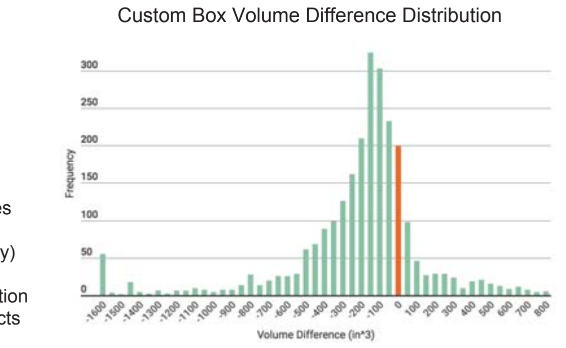
Findings

- Multi-orders is largely made up of 3 or less units and the box volume usage of these orders is low (47.11%)
- The human packaging process is one that can be broken down into a set of logical steps to form the basis of robot packaging algorithms
- Performance of Less Flexibility First and White Space algorithms is comparable to the existing system
  - Column packing algorithm tended to perform worse with 57% requiring a larger box
- A robotic packing cell rivaling the current throughput can pay for itself in as little as two years
  - Scenario 1 ROI of 4.42 years
  - Scenario 2 ROI of 1.80 years

Recommendations

- Custom box making machine would greatly increase pack tightness for 2-3 item orders
  - Average volume utilization of 2-3 item orders is 47% and 49% respectively
- Begin implementation of robotic packing cell for strictly cubic items
  - Scenario 2 has the shorter ROI and would be recommended in the long term
- Implement Tree algorithm for usage in robotic cell
- Investigate approaches to dealing with cylindrical and irregularly shaped items
  - Cylindrical items will roll if placed on their side
  - Irregular items can only be stacked on certain faces
- Create data collection program to categorize additional item characteristics (stackability, placeable faces, fragility)
  - Could be extra step added at picking station
  - Use scanning tunnel to collect all required information
- Conduct a sensitivity analysis on how the cubit size affects algorithm box outputs

Custom Box Volume Difference Distribution



38
Semi Trucks of shipping saved per week

252
Cubic Inches of space saved per order

Impacts

- Data analysis on multi-orders characteristics
- Data based documentation of human packing process
- Development of three packing algorithms to replicate human process
  - Amazon will be working this summer on developing these algorithms to bring them closer to implementation
- Designed two possible robotic cell layouts to accommodate an automated infrastructure change
  - Evaluated these possible layouts to determine their economic feasibility
- Shipping cost reduction from reduced box sizes
- Reduced labor costs from elimination of the labor intensive multi-order pack process