Using Deep Learning to Select Optimal 2-D Bin Packing Heuristics

Teya Bergamaschi (tbergam1@jhu.edu), Akaash Sanyal (asanyal5@jhu.edu), Lori Baxter (lbaxter6@jhu.edu)
Johns Hopkins University

Background

The Bin Packing Problem (BPP) is a classic NP-hard problem in combinatorial optimization and can take many forms. For this project, we focused on an offline variety of BPP that involves fitting an arrangement of rigid rectangular items inside larger rectangular bins with a single fixed capacity while minimizing the number of bins used. Specifically, we considered the case of a single fixed bin size and a known set of boxes whose sizes were less than the size of the bin and which could be ordered in any fashion prior to packing. Additionally, we allowed for free rotation of the boxes during the packing process.

Motivation and Approach

This BPP has many practical applications in common domains; for example, domains such as logistics transportation, scheduling, manufacturing, and network optimization all utilize forms of Bin Packing.

BPP Challenges:

1. NP-hard nature makes finding an exact solution for every instance impractical.
2. Different approximation heuristics, which are much more efficient than finding an exact solution, work better in different scenarios: there is no universal best heuristic.

To overcome these challenges, we propose an efficient method to approximate the optimal solution of a BPP instance can be created by training a deep neural network to choose the best heuristic.

Data and Features

Our dataset consisted of 40,000 pseudo-randomly generated instances. Each instance consisted of a bin between 40x40 and 1x1 in size and between 2 and 1000 boxes, each smaller than the bin but at least 1x1. All sizes were integers.

We extracted features for each instance based on properties of boxes, properties of bins, and the interaction of the two. In total, we extracted around 60 features, which are listed below:

Feature Type	Features
Item Features	Number of items, item area*, perimeter*, length of short side*, length of long side*, length of a side*, ratio of short to long side*
Bin Features	Bin area, bin perimeter, length of short side, length of long side, ratio of short to long side
Cross Features	Item:bin area ratio*, item:bin perimeter ratio*, item:bin length of short side ratio*, item:bin length of long side ratio*

*Six individual features were extracted from each of these: the sum, minimum, maximum, average, standard deviation, and variance.

Heuristics

We considered three bin algorithms and three packing algorithms:

Bin Algorithms	Packing Algorithms
1. Bin Next Fit 2. Bin First Fit 3. Bin Best Fit	1. Maximal Rectangles Best Short Side Fit (BSSF) 2. Maximal Rectangles Best Area Fit (BAF) 3. Maximal Rectangles Best Long Side Fit (BLSF)

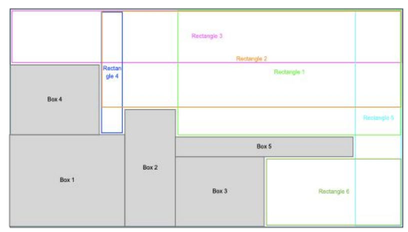

Fig. 1: This figure shows a potential bin during packing. In this scenario, boxes have been placed in the bottom left corner and any of the rectangles highlighted may be utilized depending on the packing algorithm employed.

In total, we considered the nine heuristics below through the combination of the bin and packing algorithms:

Bin Algo\Packing Algo	BSSF	BAF	BLSF
Bin Next Fit	0	1	2
Bin First Fit	3	4	5
Bin Best Fit	6	7	8

To create labels for our model, we ran each heuristic on every instance and one-hot encoded the most efficient heuristic. In the event of a tie, a random heuristic would be chosen and labeled as the correct one.

Neural Network

In order to select the best neural network, we tuned the model with varying architectures and hyperparameters. Given the one-hot, mutually exclusive labels, we used the categorical cross-entropy loss function. In order to counteract internal covariance shift and overfitting, we used batch normalization and dropout at each layer. We fit the model using the Adam optimizer and found the best learning rate to be 0.01 through tuning.

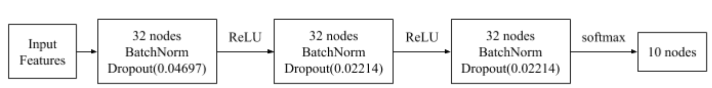

Fig. 2: This figure shows the final architecture of the neural network. The model was trained for 50 epochs with a 75% training split and a further 20% validation split.

Computational Results

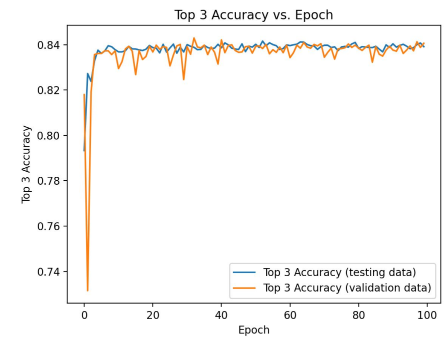

Fig. 3: This figure shows the top 3 categorical accuracy of the training and validation sets over the course of 100 epochs of training. This metric computes how often the correct label is in the top 3 predictions of the neural network.

To evaluate performance, we calculated the mean number of bins for each heuristic and our neural net and compared this to number of bins needed when selecting the best heuristic for each instance. We present the results for the top 4 heuristics.

Solving Choice	Mean # of Bins	Proportion to Best
Heuristic 4	179.3072	1.0176
Heuristic 7	178.6351	1.0138
Heuristic 3	177.0074	1.0045
Heuristic 6	176.5397	1.0019
Neural Net	176.5068	1.0017
Best Choice	**176.2076**	**1.000**

Limitations

One major limitation of our approach is that it is bounded by the heuristic approximations. While this approach can outperform many single heuristics on a large set of problems, it makes no further guarantee than any heuristic as to its performance compared to the true minimum.

Future Work

Further work on this approximation method would likely employ a multi-label approach. This approach would use a binary cross-entropy loss function for each label and a final thresholded sigmoid activation layer to allow for multiple labels.

To make the results meaningful, the difference between the best heuristic and best choice would need to be larger. Currently, there is less than a one bin difference, meaning there is no room for meaningful improvement by our neural net. Since this is just a small subset of possible heuristics, this method would probably be more meaningful if a wider selection of heuristics was considered.

Additionally, our method could easily be used to study other BPPs through minor changes and the substitution of the heuristics considered.