Foundational Heuristics for the One-Dimensional Bin Packing Problem: Mechanics, Bounds, and Comparative Analysis

Team Alan Turing

Abstract

This section establishes the theoretical baseline for the project by analyzing the four fundamental approximation algorithms for the One-Dimensional Bin Packing Problem (1DBPP): First-Fit (FF), Best-Fit (BF), First-Fit Decreasing (FFD), and Best-Fit Decreasing (BFD). We formally define their mechanics, analyze their time complexities, and present their asymptotic performance guarantees. Furthermore, we provide a qualitative analysis of the space fragmentation phenomena that differentiate First-Fit from Best-Fit strategies.

1 Introduction and Classification

The One-Dimensional Bin Packing Problem (1DBPP) is a combinatorial optimization problem wherein a set of items $I = \{s_1, s_2, \dots, s_n\}$ with sizes $s_i \in (0, 1]$ must be packed into the minimum number of unit-capacity bins. As established in the project's complexity analysis, this problem is NP-hard, necessitating the use of approximation algorithms.

We categorize the foundational heuristics based on information availability:

- **Online Algorithms (FF, BF):** The algorithm must pack item s_i immediately without knowledge of subsequent items $s_{i+1}, \dots, s_n$.
- **Offline Algorithms (FFD, BFD):** The algorithm possesses global knowledge of the input set I and may sort or manipulate the sequence prior to packing.

2 Online Heuristics

2.1 First-Fit (FF)

2.1.1 Mechanics and Complexity

The First-Fit algorithm processes items in the order they arrive. For each item s_i, it scans the existing bins $B_1, B_2, \dots, B_k$ sequentially and places the item in the first bin B_j such that the residual capacity $r(B_j) \geq s_i$. If no such bin exists, a new bin B_{k+1} is opened.

While a naive implementation requires $O(n^2)$ time (scanning all previous bins for every item), the algorithm can be optimized to $O(n \log n)$ using a segment tree or a similar data structure to query the first valid bin efficiently.

2.1.2 Theoretical Bounds

The performance of First-Fit is bounded by an asymptotic approximation ratio of 1.7.

Theorem 1 (Approximation Ratio of FF). *For any list of items L, the number of bins used by First-Fit, $FF(L)$, satisfies:*

$$FF(L) \leq 1.7 \cdot OPT(L) + 2$$

where $OPT(L)$ is the optimal number of bins.

Proof Sketch of Lower Bound. Consider an input instance designed to exploit the "greedy" nature of FF. Let the input list L consist of:

- $6M$ items of size $1/2 + \epsilon$
- $6M$ items of size $1/2 + \epsilon$ (Wait, simpler construction below)

A canonical worst-case example involves items of size $\frac{1}{2} + \epsilon$ followed by items of size $\frac{1}{2}$. FF will place each $\frac{1}{2} + \epsilon$ item into a new bin. The subsequent $\frac{1}{2}$ items cannot fit into those bins, forcing the opening of new bins. This behavior leads to significant resource wastage compared to an optimal packing that might pair them differently. □

2.2 Best-Fit (BF)

2.2.1 Mechanics and Complexity

The Best-Fit algorithm attempts to minimize immediate space wastage. For an item s_i, it searches for a bin B_j that minimizes the residual capacity $r(B_j) - s_i$, subject to $r(B_j) \geq s_i$. Ties are typically broken by choosing the lowest index.

Like FF, a naive implementation is $O(n^2)$. However, by maintaining the bins in a balanced Binary Search Tree (BST) ordered by remaining capacity, the best-fitting bin can be identified and updated in $O(\log n)$ time, resulting in an overall complexity of $O(n \log n)$.

2.2.2 Theoretical Bounds

Despite the strategic difference, Best-Fit shares the same worst-case asymptotic ratio as First-Fit.

Theorem 2 (Approximation Ratio of BF). *For any list of items L:*

$$BF(L) \leq 1.7 \cdot OPT(L) + C$$

Although the worst-case bounds are identical, empirical average-case performance often differs due to the fragmentation effects discussed in Section 4.

3 Offline Heuristics

3.1 Decreasing Variants (FFD and BFD)

The offline heuristics differ from their online counterparts solely by the inclusion of a preprocessing step: the items are sorted in non-increasing order of size such that $s_1 \geq s_2 \geq \cdots \geq s_n$.

This strategy, analogous to placing "big rocks" into a jar before "sand," ensures that large items—which are hardest to pack—are processed when the maximum number of bins have maximum capacity available.

3.1.1 Theoretical Bounds

Sorting the items significantly improves the approximation guarantee.

Theorem 3 (Johnson's Theorem, 1973). *For the First-Fit Decreasing (FFD) algorithm:*

$$FFD(L) \leq \frac{11}{9} OPT(L) + 4$$

Theorem 4 (Tight Bound, Dósa 2007). *The bound was later tightened to:*

$$FFD(L) \leq \frac{11}{9} OPT(L) + \frac{6}{9}$$

This implies an asymptotic approximation ratio of approximately 1.22, a substantial improvement over the 1.7 ratio of the online variants. Best-Fit Decreasing (BFD) achieves the same asymptotic bound of 11/9.

4 Qualitative Comparison: The Fragmentation Trade-off

While FF and BF share identical worst-case asymptotic bounds (1.7), their internal packing dynamics differ fundamentally regarding space fragmentation.

4.1 The Best-Fit "Sand" Problem

Best-Fit is designed to minimize the residual space in the chosen bin. While locally optimal, this strategy often creates bins that are nearly full but contain tiny, unusable gaps (often referred to as "sand" or "splinters").

- **Consequence:** These small gaps are often too small to accommodate even the smallest future items, effectively rendering that capacity wasted.

4.2 The First-Fit "Gap" Advantage

First-Fit is oblivious to the "tightness" of the fit; it simply selects the first valid option.

- **Consequence:** This often leaves larger, contiguous chunks of free space in the earlier bins. These larger gaps are statistically more likely to accommodate future items than the fragmented slivers created by Best-Fit.

5 Conclusion

The foundational heuristics present a clear hierarchy of efficiency versus complexity. The move from Online (FF/BF) to Offline (FFD/BFD) yields a quantifiable improvement in the approximation ratio (from 1.7 to ≈ 1.22) at the cost of the $O(n \log n)$ sorting requirement. These algorithms serve as the baseline against which advanced structured heuristics, such as Harmonic-k, and metaheuristics, such as the Grouping Genetic Algorithm, must be measured.