**4- Methodology**

In this section we will develop a solution for the bin packing problem using a hyper heuristic approach.

In previous solutions we have tried multiple approaches by using Heuristics (specific for the problem) and more general Algorithms (Meta Heuristics)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIGURE OF THE General Solution\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This time we will combine them in a hyper heuristic solution where the lower level contains the Following Heuristics

**4.1 Low Level Heuristics**

In the complexity calculation, n is the number of items and m is the number of bins used.

#### **First Fit Algorithm**

The First Fit (FF) algorithm places each item into the first available bin that has sufficient space. If no such bin exists, a new bin is opened.

* **Optimality Bound:** 1.7×OPT
* **Computational Complexity:** O(n⋅m).

**Best Fit Algorithm**

The Best Fit (BF) algorithm places each item into the bin that will leave the least remaining space after the item is added. If no suitable bin is found, a new bin is opened.

* **Optimality Bound:** 1.7×OPT
* **Computational Complexity:** O(n⋅m)

#### **Next Fit Algorithm**

The Next Fit (NF) algorithm places each item into the current bin. If the item does not fit, a new bin is opened, and the item is placed there.

* **Optimality Bound:** 2×OPT
* **Computational Complexity:** O(n)

#### **Worst Fit Algorithm**

The Worst Fit (WF) algorithm places each item into the bin that will leave the most remaining space after the item is added. If no suitable bin is found, a new bin is opened.

* **Optimality Bound:** 2×OPT−2
* **Computational Complexity:** O(n⋅m)

### **4.2 Heuristics Selection Criteria**

#### **Performance vs. Execution Time**

* **Best Fit (BF) and First Fit (FF):** These heuristics are efficient in terms of the number of bins used, with moderately good execution times.
* **Next Fit (NF) and Worst Fit (WF):** These heuristics are slightly less efficient in bin usage but offer superior execution times.

#### **Space Utilization**

* **Best Fit (BF):** This heuristic is optimized to maximize bin utilization by minimizing unused space.

#### **Placement Strategy Diversity**

* **Worst Fit (WF):** This approach aims to distribute items in a way that leaves room for larger items in future bins, thus introducing a strategic diversity in item placement.

and the Higher level is managed by the following Metaheuristic

**High Level: Genetic Algorithm (GA)**

#### **Initial Population**

The initial population in the genetic algorithm consists of randomly generated chromosomes. Each chromosome represents a unique combination of heuristics. For example, a chromosome might be represented as "NNFBWF," where each letter corresponds to a specific heuristic. The chromosome length is variable but does not exceed the number of items in the dataset.

#### **Objective Function**

The objective function evaluates each chromosome by balancing the optimality and complexity of the chosen heuristics. It is defined as a weighted combination:

**Fitness=a×Optimality+b×Complexity**

where a and b are adjustable coefficients. This balance facilitates the prioritization of efficient solutions while maintaining manageable complexity levels.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIGURE OF THE AG\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### **Solution Inputs:**

**A,B:** Fitness Coefficients

**Items list:** List of items to be treated

**Bin Capacity:** Capacity of a single bin

**Chromosome Length:** Length of a single chromosome (less than or equal to the items list length) **Initial Population:** The starting number of chromosomes in the genetic algorithm.