Bin Packing Problem Report

Thomas Meehan – 20425946

2.0 – Bin Packing problem

Description of aspects of the solution

Representation: Each individual in the solution is represented as a list of integers, each integer corresponds to the bin index where an item is assigned, [0,0,2,2] means items 1 and 2 are in bin 1 and items 3 and 4 are in bin 3.

Fitness Function: The fitness function judges the quality of a solution by counting the number of bins used and penalizing bins that exceed the bin capacity, it counts the number of valid Bins where the total weight of the items assigned to each bin does not exceed the bins capacity.

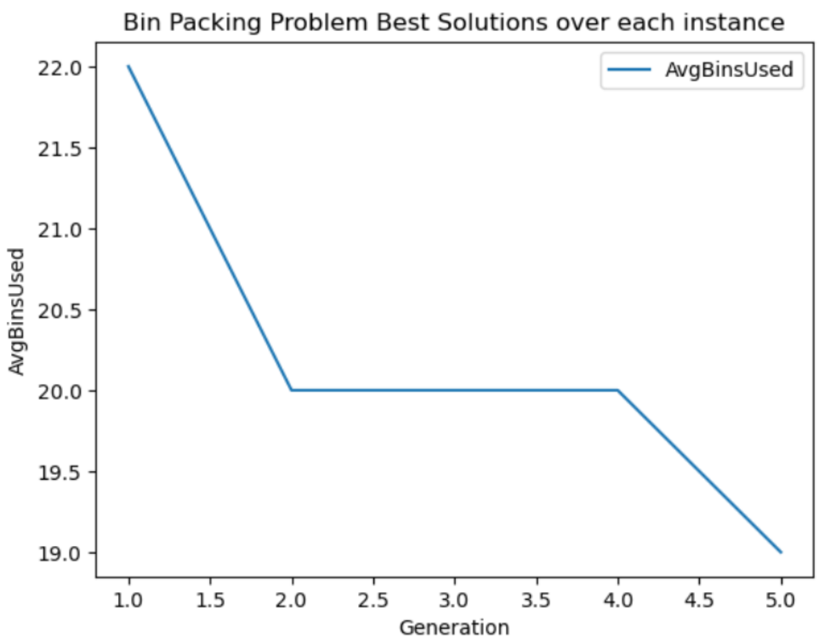
Mutation: The mutation function randomly selects bins for mutation based on probability using a random number and checking if it is lower than the mutation rate to decide if that bin is mutated and replaced with a random bin index, my mutation rate was set at 0.02

Crossover: Chooses parents for crossover based on their fitness scores, with higher fitness individuals more likely to be chosen for crossover. A random crossover point is then chosen and crossover is performed, creating two children with different parts of genetic material from the two parents.

Elitism: Elitism is used in this solution with parents being chosen based on their fitness scores, leading to more efficient solutions and higher fitness offsprings and populations

Plotting best solution for each of the 5 instances

```
[22, 20, 20, 20, 19]
```



Discussion of results

There is variability in the results across different instances with bins used by each instances best solution ranging from 22 to 19. The Algorithm shows that it effectively reduces the number of bins needed within each instance as can be seen below.

```
Next Instance
Best Solution uses -  30
Best Solution uses -  26
Best Solution uses -  26
Best Solution uses -  25
Best Solution uses -  24
Best Solution uses -  24
Best Solution uses -  24
Best Solution uses -  23
Best Solution uses -  22
Best Solution uses -  22
Next Instance
Best Solution uses -  30
Best Solution uses -  28
Best Solution uses -  26
Best Solution uses -  23
Best Solution uses -  24
Best Solution uses -  21
Best Solution uses -  23
Best Solution uses -  23
Best Solution uses -  22
Best Solution uses -  20
Next Instance
```

As can be seen here the algorithm effectively reduces the number of bins used for each instance as it moves through the 10 generations, using the new population consisting of the fittest parents of the previous generations offspring to improve the algorithms performance on the same instance over multiple generations.

We can also see the solution is relatively stable with little to no fluctuations across the best solution results of the different instances, we can also see that within each instance the algorithm consistently improves across the generations but also shows stability as there is no major jumps between one generation and the next, it tends to gradually but consistently reduce the bins needed as it iterates over the generations.

Problem Landscape Insights for the BPP

A problem landscape insight is a deeper understanding of the characteristics and the structure of the landscape of a particular optimization problem, in this case the bin packing problem.

Binary Solution Characteristics

Due to the binary nature of bins this problem shows both discrete and nonlinear features, with each changing allocation of an item to a bin potentially altering the fitness of a solution greatly, this is due to the allocation of items to bins not being linear meaning an improvement in one aspect of the solution does not constitute the overall fitness of the solution also being improved.

Exponential Solution Space:

This problem is combinatorial, meaning that it becomes more and more complex depending on the number of items added, this is due to there being a solution space that is made up of all the possible combinations of item allocations to bins. Due to this the problems complexity grows as the number of items increases.

Local Optima and Plateaus:

The problem landscape contains multiple local optima where the small changes made to a solution may not improve the overall solutions efficiency and fitness. These local optima make it challenging to escape suboptimal or subpar solutions. Plateaus which are regions of the problem landscape where the fitness remains constant for periods hinder the progress of the algorithm by delaying the arrival at better solutions. I.e..10 generations with a plateau across 3 of them means the algorithm arrives and a worse solution than one that does not plateau by the end of the 10 generations.

Consistent plateaus   Less Plateaus

```
                                    Next Instance
Best Solution uses -  30            Best Solution uses -  30
Best Solution uses -  26            Best Solution uses -  26
Best Solution uses -  26            Best Solution uses -  25
Best Solution uses -  25            Best Solution uses -  24
Best Solution uses -  24            Best Solution uses -  23
Best Solution uses -  24            Best Solution uses -  24
Best Solution uses -  24            Best Solution uses -  22
Best Solution uses -  23            Best Solution uses -  22
Best Solution uses -  22            Best Solution uses -  20
Best Solution uses -  22            Best Solution uses -  19
```

We can see here that the instance with less plateaus arrives at a better solution by the end of the 10 generations than the one that plateaus 3 times.

Diverse Solution Landscape

Solutions vary depending on the number of bins used, the distribution across these bins and also the amount of unused space within bins, this poses challenges for both exploration and exploitation attempts, as the algorithm needs to navigate diverse regions of the solution space to find the optimal solution

References

- https://universityofgalway.instructure.com/courses/5391/files/1573828?module_item_id=488677
- https://en.wikipedia.org/wiki/Evolutionary_algorithm
- https://en.wikipedia.org/wiki/Combinatorial_optimization
- https://www3.cs.stonybrook.edu/~algorith/implement/syslo/implement.shtml
- https://www.linkedin.com/pulse/optimization-deep-dive-world-problem-solving-david-cain/