Nature Inspired Computing Coursework

December 2021

1 Description of Experiments

Due to time limitations and code execution speed all test are run using 100 fitness evaluations.

1.1 BPP1

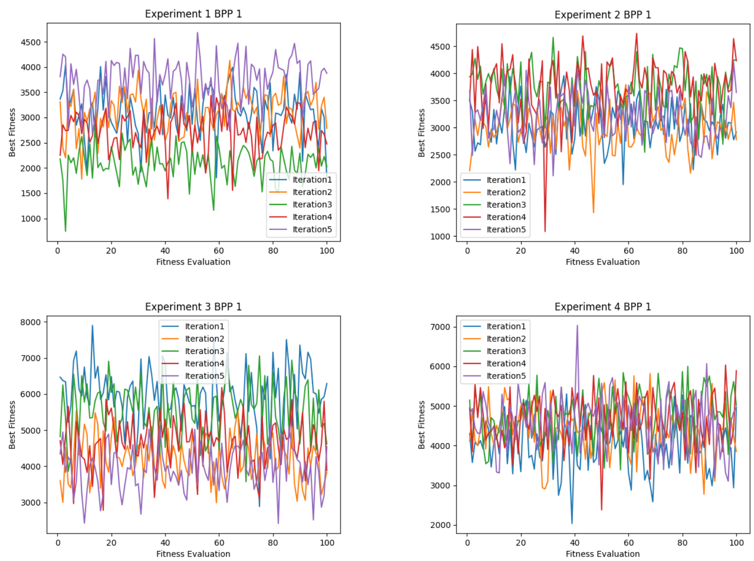
All BPP1 experiments are run using 500 items and 10 bins.

The first experiment consists of using 100 ants and a pheromone evaporation rate (rho) of 0.1. This produces a graph, BPP1 Experiment 1, that produces some good fitness scores across all five trial runs. In this trial some noticeable outliers can be identified. Most prominently the excellent sub 1000 score produced by iteration 3 almost immediately, showing that these fitness scores are possible, however on the other hand iteration 5 produced some of the worst scores with some outliers being over 4500. Furthermore iteration 4 produced some good results compared to the rest of it's scores, these outlier are the signs of the randomness inherent within the algorithm and are sure to show in further trials.

Experiment 2 changes the evaporation rate from 0.1 to 0.5, this will potentially decrease the accuracy of the model, but greatly improves its adaptability as if a new, improved path is found the ants will quickly switch, whereas at an evaporation rate of 0.1 old paths do not evaporate quickly and will therefore still be used by many ants. We see a perfect example of this in iteration 4 where a large anomaly can be seen where a new path is found the reduces the fitness score to almost 1000, the best by far, however the fitness quickly returns to normal as the combined randomness and quick evaporation of this path causes a sharp return. Equally on average the fitness scores of this experiment are higher than experiment 1, potentially showing that having a higher evaporation rate is less effective at finding an optimal solution.

Experiment 3 changes the number of ants that are placed onto the graph from 100 to 10, this should further reduce the fitness scores produced by the algorithm as fewer paths are being explored per fitness evaluation and less pheromone is being deposited causing it to evaporate faster. As can be seen in BPP1 Experiment 3, with fewer ants the ceiling for fitness scores has greatly increased with iteration 1 producing a score of close to 8000, however iteration 5 has relitivly low scores, with many being below 3000, this shows that the semi-random path finding can still produce some good results, even when fewer paths are explored.

Finally for BPP1 experiment four follows on from experiment 2 and changes the evaporation rate form 0.1 to 0.5, with potentially similar effects of larger fitness scores between iterations. This adaptability is show through the difference in scores, ranging from 7000 to 2000 and large jumps between fitness evaluations.



1.2 BPP 2

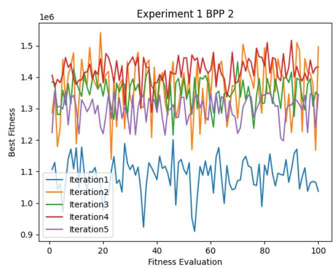
BPP 2 changes the problem by increasing the number of bins to 50 and all of the weights are squared from what there were in BPP1.

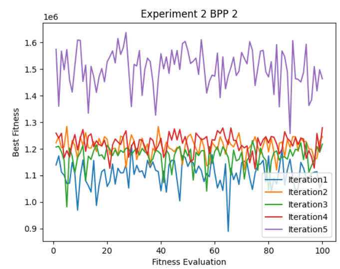
Upon running experiment 1 there is a drastic difference between BPP1 and BPP2 where the fitness range is now in the millions. The combination of a much wider range of items and many more bins increases both the execution time and the fitness. However iteration 1 produces results that are much lower than the rest of this iterations, this demonstrates that the effectiveness of the algorithm can be determined by the fitness of its starting ants, with iteration 1 starting much lower and producing much lower results overall.

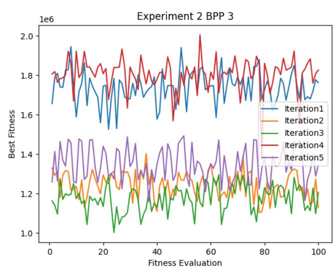
Experiment 2 increases the evaporation rate of the algorithm back up to 0.5. As can be seen on average this experiment produced slightly lower fitness scores than experiment 1, and can be seen to drop equally as low in iteration 1. Some data can also be seen to vary much more in some iterations than others, this may be a side effect of the increased evaporation rate allowing the ants to find new paths much easier than previously. Iteration 5 can be classed as a outlier due to its very high starting fitness, hindering it from reducing its fitness.

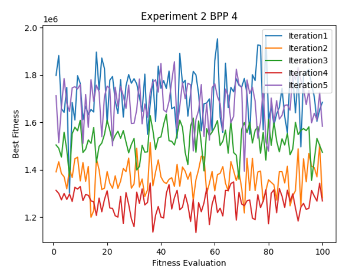
Experiment 3 shows an interesting difference that is not recorded in any of the other experiments, where two of the iterations are very close together throughout yet far away from the rest. Equally this experiment produces the highest fitness scores so far, further solidifying the evidence from BB1 experiment 3 that had a similar pattern of vary large fitness, alongside some normal values. The constant and large difference in iteration 5 should be noted, and can once again be contributed to semi-randomness causing some ant generations to preform much better within an iteration.

Finally experiment 4 that shows a wide range of starting fitness, producing a strata of each iteration, once again showing the very high scores, and on average higher scores across all iterations than experiment 3.









2 Question 1

The best results were produced when using a 100 ants as opposed to 10, and usually with an evaporation rate of 0.5 as opposed to 0.9, which is a result I was not expecting.

3 Question 2

The improved performance in reference to the number of ants can be seen when comparing BBP1 experiment 1 and 2 with BPP1 experiment 3 and 4, where experiment 1 and 2 produced much lower fitness scores overall showing that more ants per fitness evaluation improves the algorithm.

The larger evaporation rate produces both a lower average fitness score and more outliers achieving very low scores, due to the enhanced ability for ants to stray from the very pheromone heavy paths allowing them to locate better routes and not getting stuck following the same paths. This however is not the case between BPP2 Experiment 1 and 2 where experiment 1 produced lower scores. This finding may be further proof of the semi-random starting conditions of the algorithm greatly impacting the overall fitness of the iteration.

4 Question 3

The change in pheromone evaporation rate does not change the performance of the algorithm at all, however the number of ants can greatly increase the time required to run.

Experiment	Average Rounded time for one iteration (s)
BPP1 Experiment 1	132
BPP1 Experiment 2	125
BPP1 Experiment 3	13
BPP1 Experiment 4	13
BPP2 Experiment 1	263
BPP2 Experiment 2	275
BPP2 Experiment 3	27
BPP2 Experiment 4	28

5 Question 4

Ant Colony Optimization is an effective algorithm, however some different algorithms provide some significant improvements for this problem.

The random nature of ACO[1] and its generation of new solutions at each iteration a model that can retain the data between iterations to further improve the solution may provide better results. This is the concept of Evolutionary Algorithms (EA)[2], where various pressures can be added to a population to remove weak scores[3], such as tournament selection, where a proportion of the population is randomly selected, then only the set fittest are returned. This allows for favourable paths to be maintained throughout iterations.

Furthermore algorithms such as Convolution Neural Networks (CNN) could be an effective algorithm for this problem as they already deal in weights between node and are able to find connections between these nodes that might not be visible to us [4]. This improvement is more likely to find an optimal solution by 'solving' the problem as opposed to the more brute force method of ACO.

Overall ACO is an effective solution to the Bin-Packing problem, however other algorithms may provide a competitive advantage over ACO for both time and accuracy.

References

[1] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[2] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary computation*, vol. 1, no. 1, pp. 1–23, 1993.

[3] T. Blickle and L. Thiele, "A comparison of selection schemes used in evolutionary algorithms," *Evolutionary Computation*, vol. 4, no. 4, pp. 361–394, 1996.

[4] J. Yang and J. Li, "Application of deep convolution neural network," in *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2017, pp. 229–232.