Use of ACO for solving the BPP

670008197

ABSTRACT

In this paper, we investigate different methods for solving the bin-packing problem, with particular focus on ant colony optimization.

1. INTRODUCTION

The bin packing problem is a problem where items of different volumes must be packed into a finite number of bins or containers of a fixed given volume in a way that uses the least bins possible. Many algorithms have been given for the traditional formulation above, so we'll be exploring a modified version of the problem where the aim is to have the weights as evenly distributed as possible.

2. LITERATURE REVIEW

2.1 Introduction

This review will briefly explore the different options available for solving the BPP that take inspiration from nature, as such algorithms such as First-Fit and Best-Fit shall not be considered.

2.2 Ant Colony Optimisation

ACO is a nature-inspired algorithm introduced by Dorigo in 1992 [1]. It takes inspiration from the behavior of ants. When foraging ants move randomly from their nest to a food source, during the travel ants leave pheromones. This pheromone path leads other members of the colony to the food source, causing them to leave more pheromones and creating a positive feedback [1,2,3]. Pheromone evaporates meaning that on longer paths the pheromone strength ends up reduced and the ants all begin to take the shortest path.

The algorithm consists of generating a construction graph that turns this into a path finding algorithm, generating paths for the ants, updating the pheromones and having the pheromones evaporate. We will test this heuristic in detail later.

ACO is a very popular nature-inspired heuristic and has been modified for a variety of problems, including ship trajectory planning and e-learning.

2.3 Cuckoo Search Algorithm

Cuckoo search was developed by Yang and Deb in 2009 [4]. It takes inspiration from the life of a cuckoo bird, in particular the obligate brood parasitism of the species. Cuckoos lay their eggs in the nests of other host birds, some of which don't take kindly to intruders. Some hosts will throw the eggs away, some will abandon the nest. Cuckoo eggs tend to hatch earlier than most and the chicks will throw other eggs overboard. This algorithm idealized such behavior. There are 3 idealized rules:

- Each cuckoo lays a single egg at a time and puts its eggs in a randomly chosen nest
- Best nests with high quality eggs will carry over to the next generation
- Number of existing host eggs is fixed, a host can find a cuckoo egg with probability P. In this case, the host can either throw the egg away or abandon the nest to build one somewhere else. [4,5]

2.4 Genetic Algorithm

GA are search algorithms based on natural selection and genetics. It was developed by Holland in 1960 [6,7]. There are many variations of GAs but the simplest involves:

- Selection
- Crossover
- Mutation

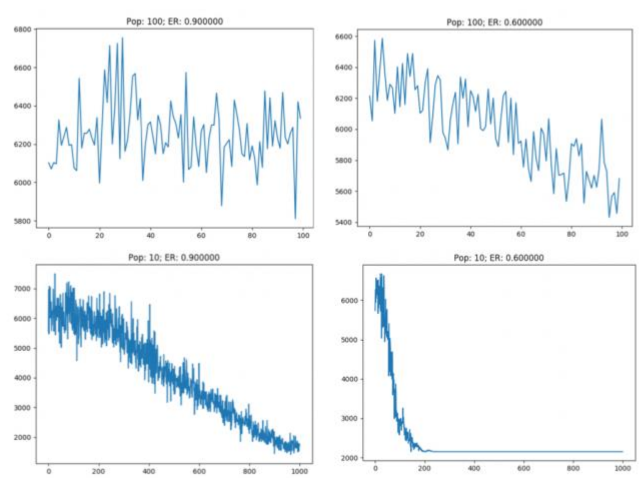
Creating a population consists of taking pairs from the initial population and using a crossover operator on them and then probabilistically mutating them as to prevent convergence to a local optimum. The new generation is then selected from the offspring so as to prevent early convergence but slowly tend towards the optimum [8,9].

Genetic Algorithms have been utilized in many different ways from generating game playing AI to designing antennae.

3. RESULTS

After implementing the Ant Colony Optimisation that was required, I ran a variety of tests, to check the progression of the generations (figures 1 and 2) and the average fitness found by each set of parameters (figure 3).

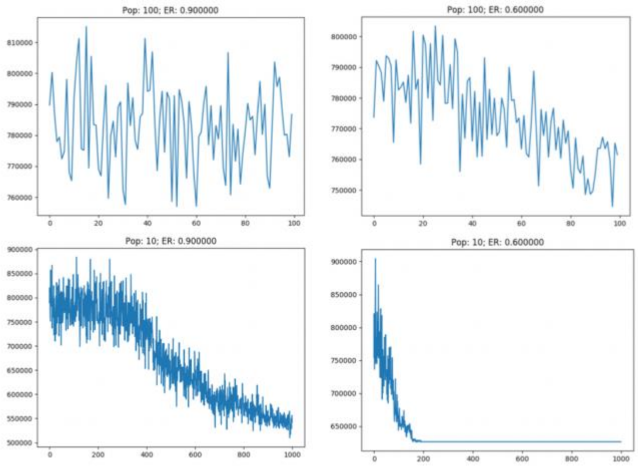
Figure 1



Generation Progression for the first BPP.

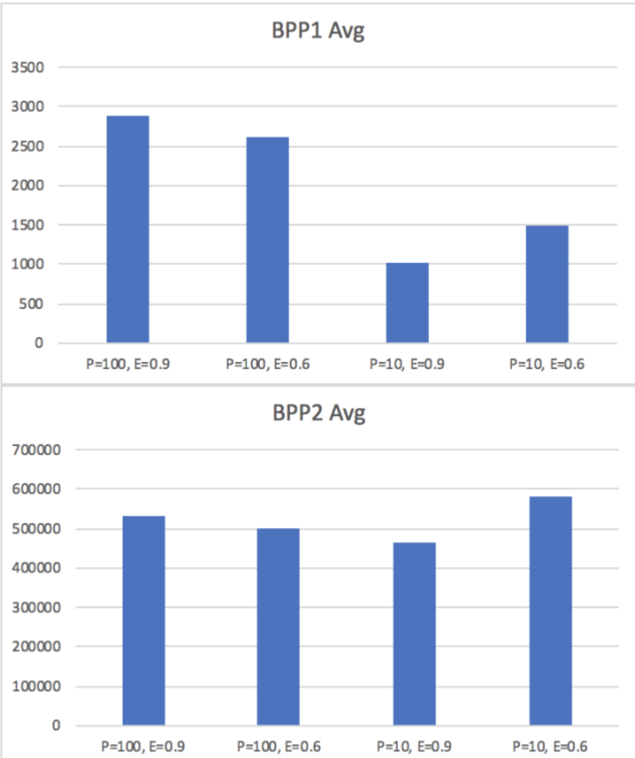
It can be seen that the second problem lead to much higher fitness values meaning we were supposedly further from an optimal solution, however this was expected due to the larger weights of the items. An optimal solution was never found although the trends in the fitness values for certain parameter sets show a lot of promise given a higher evaluation limit.

Figure 2



Generation Progression for the second BPP.

Figure 3



Average final fitness given by each parameter set for each BPP.

The results behaved more or less as expected, runtime results were also gathered in order to see the effects on efficiency. The parameters had no real effect as one would expect, but each trial in BPP1 took approximately 15 seconds and each in BPP2 took approximately 30 seconds – which was faster than I expected.

4. FURTHER WORK AND DISCUSSION

4.1 Which Combination Produces the Best Results and Why?

As can be seen in Figure 3, the combination of P=10 and E=0.9 produces the lowest average fitness value. This was unusual as a lower value for E produced better results when used with a higher population. I initially thought the evaporation rate been so high kept variation to a point of almost being random, but given more generations to continue, it seems to begin converging towards what I assume is the global optimum.

The higher evaporation rate leads to less evaporation of pheromone, which means more paths will have a chance of being explored, in earlier generations this leads to an almost random path being taken as can be seen in the generation progression for P=100, E=0.9 but it prevents early convergence to a local optimum and continues to show a trend towards better solutions as generations progress.

I’m not sure the population effects too much other than adding more conflicting paths early on, it’s more difficult to discern trends as we have less generations due to use of an evaluation limit as opposed to a generation limit. Given more time, I’d like to test with higher limits and perhaps with an equal number of generations.

4.2 Effects of Each Parameter

4.2.1 Population

Higher population seems to produce worse results for the same number of evaluations, though this is likely, at least in part, due to a lower number of generations. A higher population does seem to be more consistent regardless of evaporation rate, likely due to the amount of pheromone being dropped in each generation.

The overall effect seems to be rather unclear based on the results gathered. But it seems to indicate that a lower population converges quicker.

4.2.2 Evaporation Rate

As previously stated, a higher evaporation rate leads to a random looking pattern in earlier generations as many different paths are chosen, but it also prevents early convergence. It seems that given enough time, a higher evaporation rate seems to give better results.

A lower evaporation rate leads to a faster decrease in value but seems very likely to converge to a local optimum, as can be seen in the experiments with P=10 and E=0.6.

Which is better is down to the situation, in particular whether speed or solution optimality is the primary objective.

4.2.3 Scale

Scale was a variable I introduced to determine the method for generating the items for the problem. The second bin packing problem has much higher fitness values but this is to be expected as the item weights are much greater.

4.3 Other Algorithms

Other nature-inspired algorithms were considered earlier in this paper, without testing, it is hard to tell whether any of the other algorithms would have served any better than Ant Colony Optimisation.

Having worked with genetic algorithms before, I would see a problem arising due to reaching a local optimum and perhaps reach very similar issues. There would also be a question of how to do crossover effectively, taking bins from each solution could work but there's a lot of chance of reusing the same item that would need to be accounted for and checked, thus causing longer runtimes.

Cuckoo search is more interesting, and can likely be optimized. I feel that it'd also struggle on runtime though. The throwing out of certain eggs will keep variety up but I feel the chance of loss would be too high causing longer runtime.

From further reading, the most effective forms of these heuristics use hybrid measures to further optimize the algorithms. These could work better than the naïve ACO explored here.

4.4 Further Experiments

In order to answer the questions more completely, I'd like to run experiments at more evaporation rates and population sizes, perhaps run for a number of generations as opposed to evaluations. That way the populations can stand on a more even footing in terms of their fitness.

I'd also be interested in seeing how optimal final results would be, I would achieve this by editing the program to terminate when the fitness doesn't improve for a certain number of generations.

5. CONCLUSION

We have demonstrated the use of Ant Colony Optimisation on the modified version of the bin packing problem, the results seem positive and show this heuristic can definitely work. The runtime on each problem is pretty good considering the number of evaluations.

We also explored the effects of different parameters, finding the low evaporation leads to fast convergence to a local optimum but reaches a pretty good solution really fast and a lower population has a similar effect.

6. REFERENCES

[1] Vittorio Maniezzo & Matteo Roffilli 2008. Very Strongly Constrained Problems: An Ant Colony Optimization Approach. *Cybernetics and Systems.* 39(4): 395-424.

[2] Bijaya Kumar Nanda & Gyanesh Das 2011. Ant Colony Optimization. A Computational Intelligence Technique. *International Journal of Computer & Communication Technology (IJCCT).* Volume-2, Issue-VI.

[3] Marco Dorigo, Mauro Birattari, & Thomas Stützle 2005. Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique. *IRIDIA – Technical Report Series Technical Report No. R/IRIDIA/2006-023.*

[4] Abdesslem Layeb & Seriel Rayene Boussalia 2012. A Novel Quantum Inspired Cuckoo Search Algorithm for Bin Packing Problem. *I.J. Information Technology and Computer Science.* 5, 58-67.

[5] Milan TUBA, Milos SUBOTIC & Nadezda STANAREVIC 2011. Modified cuckoo search Algorithm for unconstrained optimization problems. *Proceedings of the European Computing Conference.* ISBN: 978-960-474-297-4.

[6] E. Falkenauer, & A. Delchambre 1992. A Genetic Algorithm for Bin Packing and Line Balancing. *Proceedings of the IEEE International Conference on Robotics and Automation.*

[7] N. Mohamadi. Application of Genetic Algorithm for the Bin Packing Problem with a New Representation Scheme. *Mathematical Sciences.* Vol. 4, No. 3, 253-266, 2010.

[8] Carlos Manuel Mira da Fonseca 1995. Multi objective Genetic Algorithms with Application to Control Engineering Problems. September.

[9] Mitchell Melanie 1999. An Introduction to Genetic Algorithms. *A Bradford Book the MIT Press.* Fifth printing.