ECM3412 Nature Inspired Computing: Ant Colony Optimisation

Student Number: 710019712

1. LITERATURE REVIEW

The traditional bin-packing problem (BPP) is a combinatorial optimisation problem that involves arranging a finite set of items with weights into a set of fixed-size bins, where the goal is to find the minimum number of bins that can hold all items. As the BPP is an NP-hard problem, it is nondeterministic and cannot be found in polynomial time [1]. Our research addresses a variation of this, where we want to assign items into a set number of bins and the total weight of each bin should be as close to one another as possible. Although many of the algorithms explored in this literature have been developed for the traditional BPP, the underlying complexity and solution approaches remain closely aligned with ours.

Approximate algorithms were the first approach to solve the traditional 1D-BPP. These algorithms can be categorised into two classes: online and offline, where the former has no knowledge of succeeding items and the latter preprocesses the items, typically sorting them in increasing or decreasing order before packing [2]. Examples of online heuristics are Next Fit (NF), First Fit (FF) and Best Fit (BF) [3], whereas offline algorithms, such as NF-Decreasing and BF-Decreasing, perform better than their online counterparts. Hybrids of these such as the Better-Fit heuristic algorithm [4] have further contributed to better solutions, however they cannot guarantee to find the optimal solution.

Meta-heuristic algorithms are problem independent and guide the search space to explore solutions more effectively. In recent years, nature-inspired algorithms have emerged as a promising class of meta-heuristics for tackling the BPP. These algorithms are inspired by patterns of behaviour in nature, and their mechanisms can be leveraged to guide the search for optimal or near-optimal solutions.

The Whale Optimisation Algorithm (WOA) [5] is a swarm intelligence-based meta-heuristic inspired by the bubble-net hunting strategy of humpback whales. The improved Levy-based Whale Optimisation Algorithm (ILWOA) [6] enhances the algorithm's performance by incorporating Levy flights to improve exploration. In addition, a mutation phase is added to accelerate convergence, and a discretisation technique (LOV) is used to convert continuous values into a combinatorial solution suitable for the BPP.

Another nature-inspired approach is the Fitness-Dependent Optimiser (FDO) [7], which is based on the foraging behaviour of scout bees in a bee colony [8]. The adapted version, Adaptive Fitness-Dependent Optimiser (AFDO) [9], builds upon FDO by introducing a weight factor to control the balance between exploration and exploitation.

The Cuckoo Search via Levy Flights (CS) [10] algorithm is inspired by the behaviour of cuckoo birds, where cuckoos lay eggs in the nests of other species. The Adaptive Cuckoo Search (ACS) [11] was designed to solve the BPP. As the original CS algorithm was designed to solve continuous optimisation problems, integer representations are used to connect the continuous outputs with the combinatorial nature of the BPP, similarly to ILWOA.

The Squirrel Search Algorithm (SSA) [12] is another nature-inspired approach, using the mechanisms of the gliding behaviour exhibited by flying squirrels. Once again, this algorithm has been adapted in the Modified Squirrel Search Algorithm (MSSA) [13], which introduces several enhancements to the original algorithm. These include a fitness evaluation based on assigned trees, as well as swap operators to swap items between bins.

The final nature-inspired algorithm that was investigated in the research for this project was the Genetic Algorithm [14]. This algorithm is primarily inspired by exon shuffling [15], a process where genetic segments (exons) are recombined to create new protein variations. The Grouping Genetic Algorithm with Controlled Gene Transmission (GGA-CGT) [16] employs a modified First Fit (FF) heuristic to create a high-quality initial population. This algorithm ensures that fitter solutions have better chances for reproduction, while the genetic processes of crossover and mutation find a good balance between exploration and exploitation.

A comparative analysis of these algorithms [8] found that they would often outperform traditional heuristic approaches, especially on more challenging problem definitions. Overall, the ACS and ILWOA algorithms seemed to perform best for hard instances of the BPP, however, the effectiveness of these algorithms is highly dependent on the appropriate selection and tuning of their parameters. The researchers emphasised the need for more extensive experiments to determine the optimal parameter combinations to yield more accurate results.

This paper studies the Ant Colony Optimisation (ACO) algorithm, using two alternative problems and a comparative analysis of different model parameters to test the effectiveness of ACO on the BPP.

2. EXPERIMENTS

2.1 Problem Definitions

The two bin-packing problems can be defined as such:

BPP1 (Linear Weights)

- Number of bins: 10
- Number of items: 500
- Weight distribution: Linear scale from 1 to 500

BPP2 (Quadratic Weights)

- Number of bins: 50
- Number of items: 500
- Weight distribution: Values from 1 to 500, squared then halved

2.2 Parameter Configuration

This study investigated three key parameters through an exploration of their combinations:

- bpp: The bin-packing problem, either bpp1 or bpp2
- p: The number of ants, either 10 or 100
- e: The evaporation rate, either 0.6 or 0.9

2.3 Algorithm Design

The problem was modelled using a construction graph comprising of k layers (no. of items) and b nodes within each layer (no. of bins). This graph was represented as a matrix. Upon initialisation, random amounts of pheromone (between 0 and 1) were assigned to each edge. When the algorithm starts, ants will traverse the network and will have a higher probability of visiting nodes with more pheromone.

At the end of each ant traversal iteration, the fitness of the paths is calculated by the difference in the maximum and minimum weighted bins. The pheromone matrix is updated accordingly, and then evaporated. The algorithm halts once the total number of evaluations reaches 10000, and at this point the best fitness score is tracked.

2.4 Results

Each experiment was conducted with five trials, where each trial used a random seed that varied between one another. However, the seed for the same trial across experiments was constant, yielding comparable results between them.

The experimental results reveal significant variations in performance across different parameter configurations, as seen in **Table 1**.

Fig. 1 shows the experimental findings in a visual representation. The choice of box plots was used to track both the mean and variance of the best fitness scores achieved by the trials for each experiment.

Table 1 Summary of the results of each experiment, where the trial with the best fitness and the average of the best fitness among trials is calculated

	Params	p=10, e=0.6	p=10, e=0.9	p=100, e=0.6	p=100, e=0.9
BPP1	Best trial fitness	410	974	961	1299
	Average trial fitness	472.2	1073	1519.2	1542.4
BPP2	Best trial fitness	409699	345084.5	421861.5	433145.0
	Average trial fitness	443957.8	390322.6	459062.6	445772.3

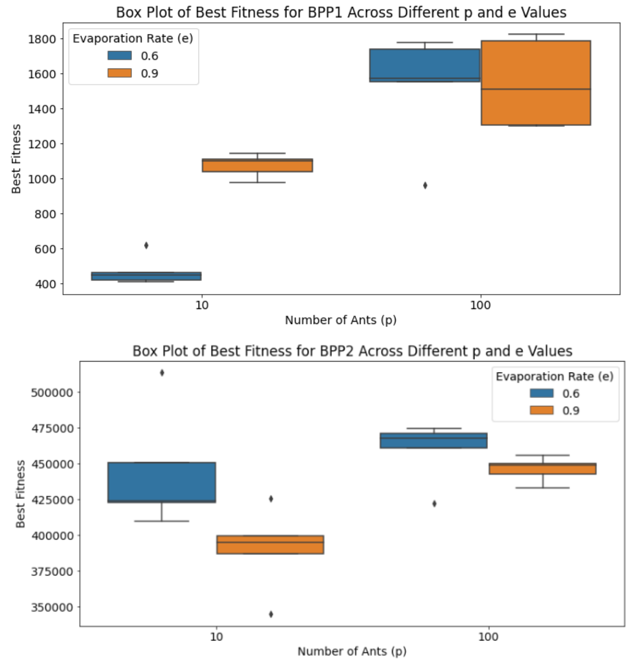


Fig. 1 Box plot distribution of best fitness scores across different configurations of ant population size (p) and evaporation rate (e) for BPP1 and BPP2

3. QUESTIONS

3.1 Which combination of parameters produces the best results?

For BPP1, the combination of parameters that produced the best results was with p = 10 and e = 0.60. This achieved an average best fitness score of 472.2 after 10000 evaluations.

For BPP2, the best combination of parameters was with p = 10 and e = 0.90. This achieved an average best fitness score of 390322.6 after 10000 evaluations.

3.2 What do you think is the reason for your findings in Question 1?

One contributing factor that could have led to the improvement of findings was the evaporation rate. A lower evaporation rate (e=0.60) means pheromone trails fade more quickly since the pheromone matrix is multiple by the evaporation rate. This evaporation rate in BPP1 helps prevent the algorithm from becoming too focused on suboptimal paths early in the search process, and instead focuses on exploration as there are less bins to organise items into. A higher evaporation actually retains pheromones longer, promoting a focus on promising paths. This is applicable to BPP2, where the best configuration was with e=0.9. A reason for this could be due to larger number of bins and much looser weight distribution in BPP2, there are many more solutions that can be created compared to BPP1. Therefore, it is suitable this algorithm prioritises exploitation over exploration.

The smaller ant population (p=10) consistently yielded better results across the two problems. With fewer ants, less ant traversals are considered after each iteration and therefore exploration is limited, and stronger paths will be quickly reinforced, albeit potentially falling into local maxima.

For BPP2, the combination of a high evaporation rate with fewer ants forces the algorithm to focus on building up already strong paths. Due to the increased search space, this enhances exploitation and prevents the model from failing to converge. For BPP1, the fewer number of bins and tighter item weight distributions lead to a more exploratory approach.

3.3 How do each of the parameter settings influence the performance of the algorithm?

For each of these parameter settings, their performance can be determined by the balance of exploration vs exploitation, convergence speed and the quality of solutions.

The number of ants had a large influence on the performance of the algorithm. A larger population of ants will lead to an increased exploration of the solution space since more ants are generating paths simultaneously. However, many ant paths generated at once will make it difficult for the dominant solutions to stand out due to a higher saturation of pheromone, therefore increasing the chance of building upon sub-optimal solutions and getting stuck in local maxima. With more ants, we would expect the convergence speed to be faster. However, as the number of evaluations are limited to 10000 for this experiment, the larger population of ants would mean each path would not be explored as deeply. A smaller ant population can lead to increased exploitation of known good solutions, although this depends largely on if the initial/early solutions are strong. Convergence might be slower because fewer pheromones are laid down, which may make it harder for the algorithm to reinforce good solutions. However, due to the limitation of 10000 evaluations, this favours fewer ants as the exploitation of ant paths can go much deeper.

The evaporation rate is also a key parameter that has great influence over the performance of the algorithm. A lower evaporation rate (more evaporation) will encourage ants to explore new paths rather than relying too heavily on previously visited paths. This can prevent the algorithm getting stuck in local optima by encouraging exploration, however, if the evaporation rate is too low (even more evaporation), this could slow down convergence to optimal paths. An optimal evaporation rate is necessary to balance exploration and exploitation. On the other hand, a high evaporation rate (little to no evaporation) will retain pheromone strength for longer, but sub-optimal solutions will also persist for longer, leading to the possibility of ants exploring these suboptimal routes.

The characteristics of each BPP significantly affect the performance. In BPP1, there are only 10 bins and a linear item weight distribution from 1 to 500, so the search space is significantly smaller. Due to this, optimal solutions are easier to reinforce through a lower evaporation rate and fewer ants. With more bins in BPP2, the update of the pheromone trails can become more diluted, as the pheromones are spread across more edges, and this can lead to difficulties in convergence. The distribution of the item weights in BPP2 in combination with more bins will demand exploitation over exploration.

3.4 Do you think that one of the algorithms in your literature review might have provided better results? Explain your answer.

The algorithm that might produce better results than Ant Colony Optimisation (ACO) would be the Adaptive Fitness-Dependent Optimiser (AFDO). Although other algorithms performed better in the comparative analysis study summarised in the literature review, many of these algorithms are tailored towards the traditional BPP where the number of bins is not fixed. For instance, the Improved Levy-based Whale Optimisation Algorithm is optimised for minimising the number of bins, and the LOV discretisation technique is less relevant when there is a fixed number of bins.

On the other hand, AFDO uses a fitness function that is focused on bin occupancy rather than the number of bins used. This aligns better with our BPP, and therefore we can expect solutions to perform well. A weight-factor is introduced in AFDO compared to the regular FDO, and this parameter is responsible for controlling the exploration-exploitation trade-off. Compared to other nature-inspired algorithms, AFDO has less parameter to tune, and therefore the implementation of the algorithm is simpler. Although other algorithms such as Adaptive Cuckoo Search may perform better for hard problems, AFDO performs very well for easy and medium difficulty problems, and still performs well for difficult problems. As our BPP can be claimed to be simpler than the traditional BPP, I have confidence that this algorithm will perform well and may provide better results than ACO.

4. FURTHER WORK

To further explore the behaviour of this algorithm, I analysed the progression of the best fitness score as the number of evaluations increased. This helped determine whether the algorithm was converging, and whether the solutions could be improved if we used a higher number of evaluations. This was particularly interesting for BPP2, as due to the increased number of bins and item weights compared to BPP1 suggested that the solutions may not have fully converged and could benefit with more evaluations.

As seen in **Fig. 2**, all configurations in BPP1 show rapid initial improvement within the first 2000 evaluations. The best performing configuration (p=10, e=0.6) reaches its optimal fitness early on (around the 3000-evaluation mark) whereas the other configurations (with the exception of p=100, e=0.9) continue to improve, even up to the 10000-evaluation mark. Trials with 10 ants performed better, whilst the value of the evaporation rate was not as influential on weaker solutions. Unlike BPP1, the best solution in BPP2 only changes around the 9000-evaluation mark, where the configuration of p=10, e=0.9 takes the lead. There is much more movement in fitness scores near the end of the evaluation period, suggesting that extending the evaluations beyond 10000 could lead to better convergence and improved solutions.

To better understand the impact of parameter values on the ACO's performance, I conducted an exploration study to track the best fitness scores achieved with more parameter combinations for BPP1 and BPP2. Upon studying **Fig. 3**, it is evident that for BPP1, fewer ants led to better solutions. The influence of evaporation rate was less consistent across different ant populations, although higher evaporation rates tended to produce weaker solutions with fewer ants. BPP2 followed similar trends, except for p=10, where a higher evaporation rate produced significantly better fitness scores.

The inconsistency in the effect of the evaporation rate prompted a further study, where I examined the fitness scores against a wider range of evaporation rates. In these experiments, the ant population was fixed at p=10, as this was the best performing configuration across both bin-packing problems. I tested evaporation rates from 0.1 to 1.0, in increments of 0.1. Note, an evaporation rate of 0 (complete evaporation) was excluded as this would entirely erase pheromone trails from previous exploration. The same seed was used across experiments to minimise random variations.

As visualised in **Fig. 4**, the configuration of p=10 and e=0.8 for BPP1 produced the highest fitness score. Extreme evaporation rates drastically affected the performance. For BPP2, the fitness scores followed a more consistent trend, generally improving as the evaporation increased, other than at e=1. The best configuration for BPP2 was p=10 and e=0.9. Therefore, across both problems, a higher evaporation rate is favoured, which leads to more pheromone remaining on the trail after each iteration. This therefore favours exploitation over exploration, which is understandable for BPP2, and although slightly contradicts our claims for BPP1, the evaporation rate for BPP1 is still lower than that of BPP2.

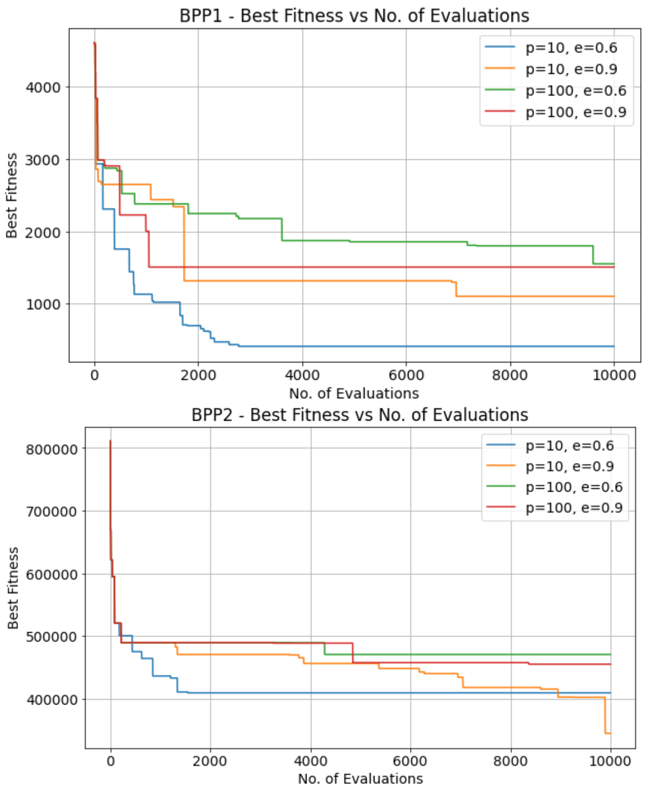
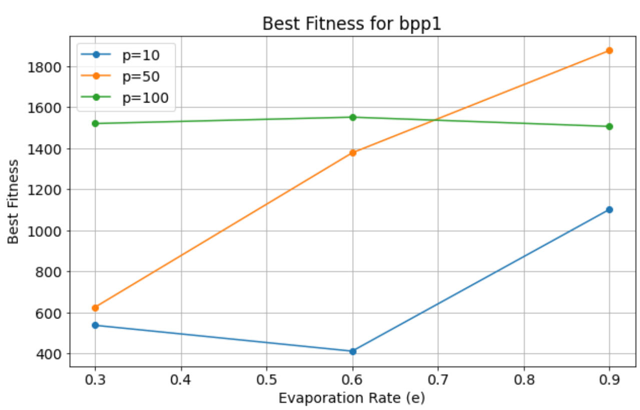


Fig. 2 Plots comparing the best fitness values versus the number of evaluations for BPP1 and BPP2 with different parameter configurations



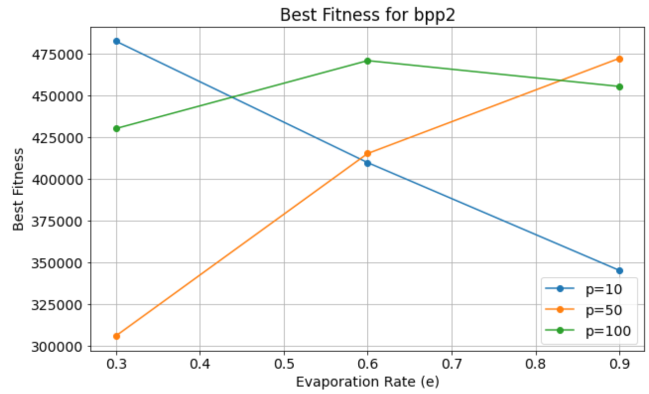


Fig. 3 Plots comparing the relationship between exploration rate (e) and best fitness values for BPP1 and BPP2 under different ant population sizes (p)

Fig. 4 Plots illustrating the impact of evaporation rate (e) on best fitness for BPP1 and BPP2 with a fixed ant population size of p=10

5. Conclusion

This research investigated the effectiveness of Ant Colony Optimisation (ACO) for solving two distinct bin packing problems. My findings demonstrate that ACO's performance is highly dependent on both the problem characteristics and parameter settings, with different configurations yielding better results for different problem types. While ACO demonstrated some capability in solving both bin packing problems, there is room for further parameter tuning and exploration of other nature-inspired algorithms such as the Adaptive Fitness-Dependent Optimiser.

6. REFERENCES

[1] Olivares, R., Medina, S. and Godoy, J. (2024) '0/1 1-D Bin Packing Problem Solved by a Recent Nature-Inspired Optimizer', Machine Learning Methods in Systems (CSOC 2024), pp. 437-445.

[2] Sgall J. Online bin packing: Old algorithms and new results. In Conference on Computability in Europe. Cham: Springer; 2014. p. 362–72.

[3] Johnson DS, Demers AJ, Ullman JD, Garey MR, Graham RL. Worst-case performance bounds for simple one-dimensional packing algorithms. SIAM J Comput. 1974;3(4):299–325.

[4] Bhatia AK, Hazra M, Basu SK. Better-fit heuristic for 1D BPP. In 2009 IEEE International Advance Computing Conference; 2009. p. 193–6.

[5] Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

[6] Abdel-Basset, M., Manogaran, G., Abdel-Fatah, L. and Mirjalili, S. (2018) 'An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems', Personal and Ubiquitous Computing, 22, pp. 1117-1132.

[7] Abdullah JM, Ahmed T. Fitness dependent optimizer: Inspired by the bee swarming reproductive process. IEEE Access. 2019;7:43473–86.

[8] Munien, C. and Ezugwu, A.E. (2021) 'Metaheuristic algorithms for one-dimensional bin-packing problems: A survey of recent advances and applications', Journal of Intelligent Systems, De Gruyter. doi: 10.1515/jisys-2020-0117.

[9] Abdul-Minaam DS, Al-Mutairi WMES, Awad MA, El-Ashmawi WH. An adaptive fitness-dependent optimizer for the one-dimensional bin packing problem. IEEE Access. 2020;8:97959–74.

[10] Yang X-S, S Deb. Cuckoo search via levy flights In 2009 World congress on nature & biologically inspired computing (NaBIC). IEEE; 2009. p. 210–14.

[11] Zendaoui Z, Layeb A. Adaptive cuckoo search algorithm for the bin packing problem. In: Chikhi S, Amine A, Chaoui A, Kholladi M, Saidouni D. editors. Modelling and implementation of complex systems. Lecture notes in networks and systems, vol. 1. Cham: Springer; 2016.

[12] Jain M, Singh V, Rani A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm Evol Comput. 2019;44:148–75.

[13] El-Ashmawi WH, Abd Elminaam DS. A modified squirrel search algorithm based on improved best fit heuristic and operator strategy for bin packing problem. Appl Soft Comput. 2019;82:105565.

[14] Rohlfshagen, P. and Bullinaria, J.A. (2010) 'Nature inspired genetic algorithms for hard packing problems', Annals of Operations Research, 179, pp. 393-419.

[15] Roy, S. W. (2003). Recent evidence for the exon theory of genes. Genetica, 118(2–3), 251–266

[16] Quiroz M, Reyes LC, Torres-Jimenez J, Santillán C, Fraire-Huacuja H, Alvim A. A grouping genetic algorithm with controlled gene transmission for the bin packing problem. Comput Operat Res. 2014 Oct;55:52–64.