

Bin-Packing with Ant Colony Optimisation

650043500

28 November 2018

Abstract

The Bin-Packing Problem is an NP-hard combinatorial optimization problem. The objective is to equally distribute items of different weights into a finite set of bins. Due to its combinatorial complexity, it is impractical to solve for optimal solutions and hence computational approximations are better suited, especially for larger instances of the problem. This paper explores the growing field of nature-inspired solutions applied to Bin Packing. The paper then focuses on Ant Colony Optimisation as a heuristic approach to Bin Packing by transforming the problem to finding the best path through a weighted graph. It then highlights the limitations of this model before attempting to solve them with the *Max-Min* Ant Colony Optimisation algorithm.

1 Introduction

The Bin-Packing Problem (BPP) is a well-known NP-hard combinatorial optimization problem[1]. The objective is to equally distribute items of different weights into a finite set of bins, whilst minimising the difference in weight between the bins. Martello and Toth produced the MTP algorithm and Korf the Bin Completion algorithm as traditional, exact approaches to the problem[2, 3]. Despite how well these solutions work on small problems, the combinatorial nature of Bin Packing gives rise to the need for heuristic approaches to solve for large instances. Thus, nature-inspired approaches offer a more practical solution, which are discussed in the following section.

2 Nature-Inspired Approaches

Falkenauer devised the Hybrid Grouping Genetic Algorithm (HGGA) for bin packing[4]. HGGA is a GA modified to suit problems where the aim is to approximate the optimal partition of a set. In Falkenauer's implementation he attempted to minimise the number of bins used by minimising the following cost function: $f_{BPP} = \frac{\sum_{i=1}^N (\frac{F_i}{C})^k}{N}$ where N is the number of bins, F_i the weight of bin i , C the bin capacity and $k > 1$. The mechanism consisted of a crossover between two parents and

a mutation whereby a number of bins were selected uniformly at random, eliminated and their items inserted back into the remaining bins. Falkenauer found that his implementation performed better than the MTP procedure.

Liang *et. al.* proposed an Evolutionary Algorithm (EA) in response to Falkenauer's HGGA for the Cutting Stock Problem (CSP)[5]. The NP-hard BPP and CSP are both classical combinatorial optimisation problems, with BPP being a special case of CSP. Therefore, each BPP can be represented as an instance of CSP. The problem was represented by chromosomes, as a vectorised, ordered list of all items. A series of cut points split the vector into bins. Much like BPP the items cannot change and hence a 3-Point Swap (3PS) mutates the chromosome. The first swap point is selected uniformly at random, the next with probability $P = \frac{\sqrt{1/w_j}}{\sum_{j=1}^M \sqrt{1/w_j}}$ with w_j being the wastage at stock j . The fitness function has two terms; the first emphasises minimising the waste and the second emphasises minimising the number of stocks with wastage: $f = \frac{1}{M+1} (\sum_{j=1}^M \sqrt{\frac{w_j}{L_j}} + \sum_{j=1}^M \sqrt{\frac{v_j}{M}})$. From here the algorithm follows the same process as other EAs by generating an initial population, selecting the best portion of the population using the fitness function and generating new offspring with the 3PS. This process is repeated until termination.

3 Ant Colony Optimisation

Ant Colony Optimisation (ACO) is a multi-agent metaheuristic approach that can be applied to combinatorial problems by transforming them into finding the shortest path through a weighted graph. The initial Ant System was proposed by Dorigo *et. al.* in 1991, taking inspiration from the stigmergy shown by ants leaving pheromone trails to find the shortest path to food [6]. It was adapted and refined over the years into the Ant Colony System published in 1997 by Dorigo & Gambardella [7, 8, 9]. This version included a *pseudorandom proportional rule* that increased the likelihood of selecting a good path through the graph by favouring the exploitation of pheromone information. The pheromone

values are updated by each ant and allowed to evaporate at a specified rate, shown in Eq. 1. For this implementation $\Delta\tau_{ij}^k = 100/f_k$ where f_k is the fitness of ant k and $(1 - \rho) = e$, or the evaporation rate.

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^p \Delta\tau_{ij}^k \quad (1)$$

4 Results

4.1 Problem 1

The first BPP tackled with the ACO technique is packing $N = 200$ items into $b = 10$ bins where the weight of item i is a random permutation between 1 and 200. It is important to note that this technique has a verifiable exact solution, which is discussed in Section 5.

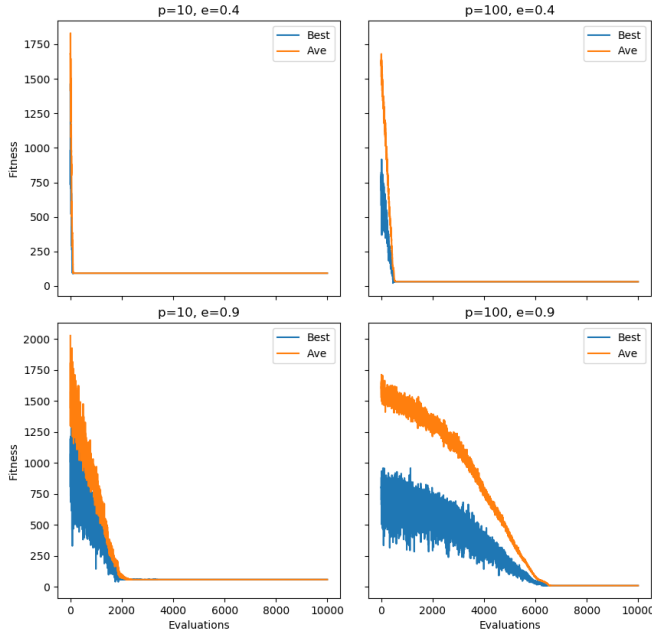


Figure 1: BPP1 Graphs

<i>Trial</i>	<i>p</i> =10, <i>e</i> =0.4	<i>p</i> =100, <i>e</i> =0.4	<i>p</i> =10, <i>e</i> =0.9	<i>p</i> =100, <i>e</i> =0.9
1	253	44	67	42
2	93	65	129	25
3	176	75	59	20
4	341	102	102	26
5	461	31	108	10
<i>Ave</i>	264.8	63.4	93	24.6

Figure 2: BPP1 Results

On average, for the BPP1 instance of the problem the values $p = 100$ and $e = 0.9$ converge to a more exact solution, displayed in Fig. 1 and Fig. 2.

4.2 Problem 2

For the second BPP, the ACO method packs $N = 200$ items into $b = 50$ bins where the weight of item i is $\frac{ij}{2}$, with j being a random permutation between 1 and 200.

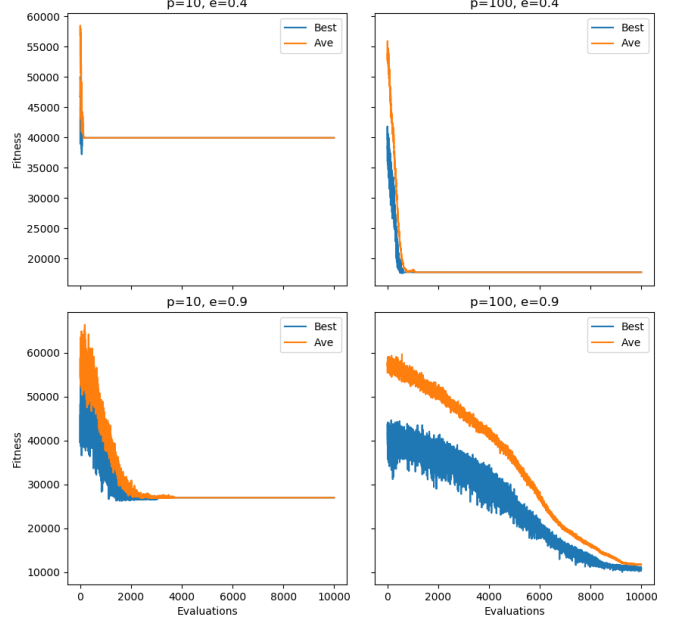


Figure 3: BPP2 Graphs

<i>Trial</i>	<i>p</i> =10, <i>e</i> =0.4	<i>p</i> =100, <i>e</i> =0.4	<i>p</i> =10, <i>e</i> =0.9	<i>p</i> =100, <i>e</i> =0.9
1	39945	17765	29769	13135.5
2	47197	23491.5	33571.5	11448.5
3	48398	34678.5	27028	11024.5
4	51308.5	22124.5	27572.5	12918
5	40614.5	22171	26995.5	14109
<i>Ave</i>	45492.6	24046.1	28987.3	12527.1

Figure 4: BPP2 Results

Again, for the BPP2 instance of the problem the values $p = 100$ and $e = 0.9$ converge to a more exact solution, displayed in Fig. 3 and Fig. 4.

5 Discussion

It is clear from both problems that a high e value, and hence a lower rate of evaporation, produces better results. This is due to the smoother search landscape that results in a slower rate of convergence. At each decision point, the pheromone values of the next node are similar, giving an approximately equal weight to the selection of each. Thus, the ants traverse more of the graph and, although it may converge slower, it is more likely to approximate the optimal solution. This is represented best in Fig. 1 and Fig. 3 where $e = 0.9$ as we can

see a smoother convergence than the sharp alternative presented where $e = 0.4$.

For both problems, there are b^N possible paths through the graph. For BPP1 this is 10^{200} and for BPP2 this is 50^{200} . Even in 10000 iterations it is simply impossible for all paths to be traversed with the given number of paths. However, with an increase in the number of paths it is statistically more likely for a larger search space to be covered. It does however produce a trade off between time complexity and accuracy as the time complexity is contingent on the number of evaluations E and the number of paths p . Therefore, for a higher p the optimisation will take longer to converge but it is more likely that an ant will traverse the graph with closer proximity to the optimal solution. This is displayed especially well in Fig. 3 where $p = 100$ converges slower but with greater precision than $p = 10$.

With BPP1 there are $N = 200$ items, ranging from 1 to 200. Thus, the optimum weight per bin is the total weight of all the items $W = \frac{N(N-1)}{2} = 20100$ divided by the total number of bins $b = 10$, making $\frac{W}{b} = 2010$. It is theoretically possible for all the items in BPP1 to be perfectly packed into all the bins. The optimum weight per bin can be achieved by iteratively placing the maximum and minimum weighted items in each bin such that for the first iteration: $b_1 = \{200, 1\}, b_2 = \{199, 2\}, \dots, b_M = \{191, 10\}$. Repeating this process results in all bins having a balanced weight of 2010 making the difference between the maximum and minimum bins 0. This proves that all the solutions in BPP1 have converged prematurely because they stagnate before reaching the exact solution, which prompts the research in the following section. Stagnation describes the situation in which all ants follow the same path, displayed on Fig. 1 by the average line meeting the optimal line, followed by a lack of further improvement in fitness. This is explained mathematically by all but one pheromone value at each decision point having evaporated completely, resulting in each ant choosing the exact same path with a probability close to 1 as they are highly unlikely to choose a node that has already evaporated.

The difference in weights of individual items is higher across the range of items in BPP2. The potential for big differences between maximum and minimum bins is reflected in the differences between Figures 1 and 2 when compared to Figures 3 and 4 as the fitness values are several orders of magnitude higher in BPP2. Thus, upon updating the pheromone values with Eq. 1 the increments from the second term cause miniscule difference to the pheromone graph. This results in a slower rate of convergence. Similarly, due to the increased complexity in BPP2 over BPP1 we can assume that BPP2 converges prematurely.

Falkenauer found close to optimal solutions to the BPP by enhancing the GGA with a local optimisation inspired by the Dominance Criterion of Martello and Toth.

However, Liang *et. al.* concluded that, in their tested scenarios, EAs performed better than GAs. Hence, the EA of Liang *et. al.* would likely perform better than this rudimentary ACO implementation. However, the following section implements the *Max-Min* ACO technique, which drastically improves performance.

6 Further Work

Stützle and Hoos proposed an ACO algorithm called *Max-Min* as an approximation of the Travelling Salesman Problem [10]. The differences in their implementation are as follows:

1. Only a single ant adds to the pheromone graph, with debate over whether to use the iteration or global best.
2. Stagnation is avoided by limiting pheromone values between τ_{max} and τ_{min} .
3. Pheromones are initialised to τ_{max} over the uniform distribution between 0 and 1.

This means that Eq. 1 is adapted to only update from the best ant as shown below, where like before $\Delta\tau_{ij}^{best} = 100/f_{best}$ such that:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij}^{best} \quad (2)$$

The theoretical maximum amount of pheromone added after each iteration is $100/f_{opt}$ where f_{opt} is the optimal fitness. Hence, we try to approximate this value by using the global best fitness f_{gbest} . This refines our concept of convergence by showing that τ_{max} from Eq. 3 is asymptotically bounded.

$$\tau_{max} = \frac{1}{1 - e} \frac{100}{f_{gbest}} \quad (3)$$

To determine τ_{min} Stützle and Hoos assumed better solutions are found just before stagnation occurs and that improved traversals through the graph are often close to the current best. With this implementation, the main influence on the rate of convergence is the relative difference between τ_{max} and τ_{min} rather than heuristic information like the evaporation rate e and number of paths p . The best solution is found with probability P_{best} after making N decisions. Therefore, assuming constant probability for each decision point, $P_{best} = P_{dec}^N$ which is used to construct the following equation:

$$\tau_{min} = \frac{\tau_{max}(1 - \sqrt[N]{P_{best}})}{\frac{N}{2} \sqrt[N]{P_{best}}} \quad (4)$$

We replace $\sqrt[N]{P_{best}}$ from their equation with P_{dec} as they are equivalent. Although a range of solutions were suitable, we settled on $\frac{1}{b}$ because, assuming the graph is unweighted, at each decision point there is a probability of $\frac{1}{b}$ of going to any of the subsequent nodes.

If P_{dec} is too small the convergence becomes chaotic as $\tau_{min} > \tau_{max}$. We set $\tau_{min} = \tau_{max}$ in this instance, which corresponds to performing the method of Section 4. If $P_{dec} = 1$ then $\tau_{min} = 0$ which results in a greedy search through the graph. For the experiment in Fig. 5 we use the best heuristic information of $e = 0.9$ and $p = 100$ from Section 4, making P_{dec} the independent variable. The graphs clearly show that $\frac{1}{b}$ is a suitable approximation of P_{dec} .

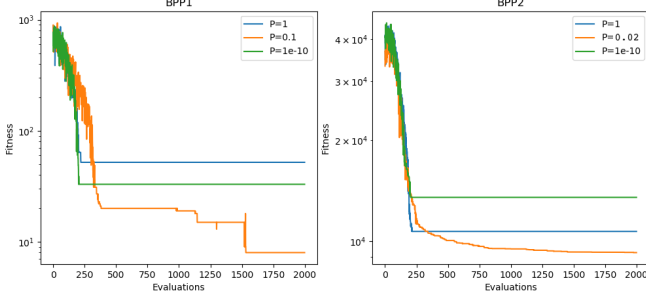


Figure 5: P_{dec}

The best e value of 0.9 from Section 4 was preserved to keep the search space smooth, although the evaporation rate's effect is somewhat limited by τ_{min} . The best path number $p = 100$ from Section 4 was used as an unbiased comparison between the best solutions from BPP1 and BPP2 with the new implementation. Fig. 6 shows the results (Note the logarithmic scale).

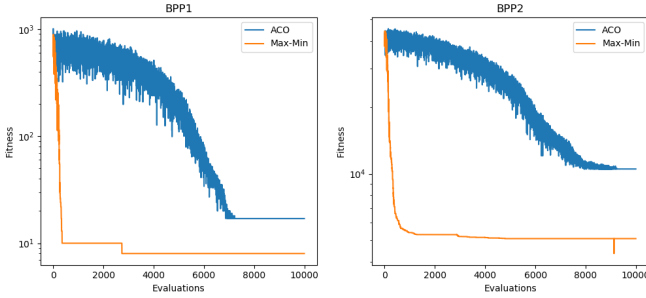


Figure 6: ACO vs Max-Min

Levine and Ducatelle implemented *Max-Min* for the BPP [11]. They used the same calculation of fitness as Falkenauer using $f_{g_{best}}$ multiplied by the number of times item i and j , from Eq. 2, are in the same bin. It is clear from Fig. 6 that *Max-Min* is a far superior implementation. There is less variance and a faster rate of convergence to a more exact solution. This is because τ_{min} ensures that there is always a probability greater than 0 of a node being picked, favouring exploration of the search space over exploitation. On the other hand, using only the global best ant for updating makes for a more aggressive search, favouring exploitation of the pheromone graph over exploration. The complimentary combination of these two methods creates an auspicious balance between exploration and exploitation

that demonstrates significant improvements in performance.

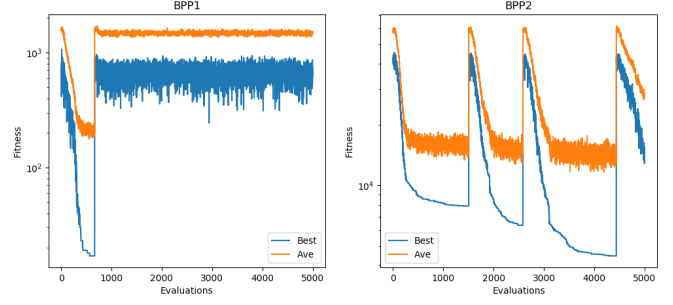


Figure 7: Standard Reinitialisation

Despite the obvious improvements, we know from looking at BPP1 in Fig. 6 that we have still converged prematurely. Stützle and Hoos proposed avoiding stagnation by reinitialising the pheromone graph to τ_{max} . When applied to the BPP, Fig. 7 highlights the lack of overall improvement of this method because reinitialising to τ_{max} essentially restarts the algorithm in hope that the stochastic nature of ACO will allow for a better solution to arise. Even though, given enough time, this solution may offer improvements, it would be better to reinitialise in such a way that we always converge to a better solution. Here follows, a novel exploratory method of reinitialisation in which τ_{max} is preserved while increasing the probability that an ant will choose a path that was previously a τ_{min} . This was the biggest limitation of standard reinitialisation as the importance of τ_{max} was lost. With T_t being the current pheromone graph and $\tau_{max} \geq T_{t+1} \geq \tau_{min}$.

$$T_{t+1} = T_t * \frac{1}{\omega \tau_{min}} \quad (5)$$

Upon stagnation, only one bin per item is τ_{max} with the others being τ_{min} . Therefore, all the ants are highly likely to follow the exact same path. By applying Eq. 5 and imposing the upper and lower bounds we preserve the value of τ_{max} while bringing $\tau_{min} = \frac{1}{\omega}$. Fig. 8 shows the improvement. Further research could go into choosing an appropriate value for ω although this paper used $f_{g_{best}}$ which showed improvement over Fig. 7.

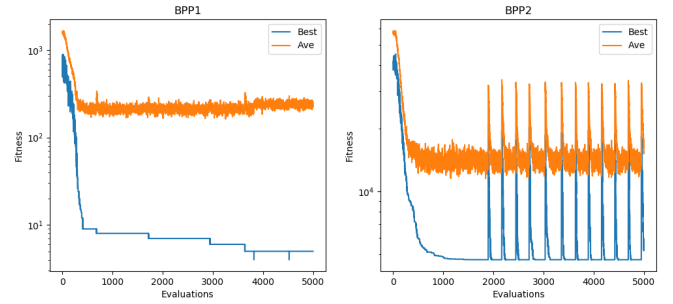


Figure 8: Novel Reinitialisation

References

- [1] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [2] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.
- [3] Richard E. Korf. A new algorithm for optimal bin packing. In *Eighteenth National Conference on Artificial Intelligence*, pages 731–736, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.
- [4] Emanuel” ”Falkenauer. A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2(1):5–30, Jun 1996.
- [5] Ko-Hsin Liang, Xin Yao, Charles Newton, and David Hoffman. A new evolutionary approach to cutting stock problems with and without contiguity. *Computers & Operations Research*, 29(12):1641 – 1659, 2002.
- [6] Alberto Coloni, Marco Dorigo, and Vittorio Maniezzo. Distributed optimization by ant colonies. *Proceedings of the First European Conference on Artificial Life*, 01 1991.
- [7] Marco Dorigo. Optimization, learning and natural algorithms. *Ph.D. Thesis, Politecnico di Milano, Italy*, 1992.
- [8] M. Dorigo, V. Maniezzo, and A. Coloni. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, Feb 1996.
- [9] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, April 1997.
- [10] Thomas Stützle and Holger H. Hoos. Max-min ant system. *Future Gener. Comput. Syst.*, 16(9):889–914, 2000.
- [11] J. Levine and F. Ducatelle. Ant colony optimization and local search for bin packing and cutting stock problems. *Journal of the Operational Research Society*, 55(7):705–716, Jul 2004.