Available online at: http://jurnal.untirta.ac.id/index.php/jiss

JOURNAL INDUSTRIAL SERVICESS

Industrial Engineering Advance Research & Application

eISSN 2461-0631

The 1-D bin packing problem optimisation using bees algorithm

Natalia Hartono[a,b*], Asrul Harun Ismail[c], Sultan Zeybek[d], Mario Caterino[e], Kaiwen Jiang[a], Murat Şahin[f], Christine Natalia[g]

[a] Department of Mechanical Engineering, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
[b] Department of Industrial Engineering, University of Pelita Harapan, MH Thamrin Boulevard, Lippo Karawaci 1100, Tangerang, 15811, Indonesia
[c] Department of Industrial Engineering, University of Pancasila, Raya Lenteng Agung 56-80, Jakarta Selatan, 12640, Indonesia
[d] Department of Computer Engineering, Fatih Sultan Mehmet Vakif University, Sutluce Neighb., Karaagac Road 12/A, Beyoglu, Istanbul, Turkey
[e] Department of Engineering, University of Campania Luigi Vanvitelli, Aversa, via Roma 29, Italy
[f] Department of Control System, Roketsan, 06780 Elmadag, Angkara, Turkey
[g] Department of Industrial Engineering, Atma Jaya Catholic University of Indonesia, Jend. Sudirman 51, Jakarta Selatan, 12930, Indonesia

ARTICLE INFO

Keywords:
Bees algorithm
Bin packing
Optimisation

ABSTRACT

The bin packing problem is a classic combinatorial optimisation problem that is widely used in various applications such as assembly line balancing, scheduling, and time-tabling. Metaheuristic algorithms can provide solutions to these problems faster than exact methods. Bees Algorithm, a metaheuristic algorithm inspired by the foraging activity of bees, is known for its performance in solving optimisation problems. To our best knowledge, this is the first use of the Bees Algorithm to solve a bin packing problem. In this paper, we use the basic Bees Algorithm to demonstrate near-optimal solutions and measure the accuracy of solutions to the one-dimensional bin packing problem. The algorithm procedure and parameter settings are set following the previous research. Benchmark datasets are used in the experiment, and accuracy is measured. The results indicate that the basic Bees Algorithm for bin packing problems and previous research on travelling salesman problems produce similar accuracy results.

1. Introduction

Bin Packing Problem (BPP) is one of basic combinatorial optimisation problem and widely studied [1, 2, 3]. It is well known that this classic problem can be applied to a variety of diverse practical problems, including scheduling, time tabling, facility location, allocating computer memory, assembly line balancing, and supply chain [1, 2, 3, 4, 5, 6].

The 1-D bin packing problem is defined as follows: Given an infinite supply of bins, each with a capacity of C, and a list L of items with sizes no greater than C, the problem is to pack the items into the fewest possible bins while keeping the sum of the sizes in each bin below C [3, 5, 7].

A NP-Hard problem like the BPP is difficult to solve using exact methods like branch and bound, because as the size of the problem increases, the solution grows exponentially [1, 3, 4, 8]. Many researchers use heuristic and metaheuristic approaches to find near-optimal solutions for BPP in faster computational time. The one-dimensional BPP has been subjected to a wide range of algorithms, including classical-based heuristic techniques as well as global metaheuristic algorithms. It is worth noting that metaheuristics are helpful in solving bin-packing problems since their algorithms can manage complex constraints and deliver high-quality solutions in less time [9].

The extended bin packing problem with variable size and capacity was introduced as a real-world problem in the logistics business and solved using three heuristics approaches by Liu *et al.* [10]. The proposed approach works well in tackling this problem and finds near-optimal solutions in a short amount of time. An evolutionary approach for bin packing problem is proposed in Luo *et al.* [11]. Another study that compared the effectiveness and efficiency of several metaheuristic optimisation techniques for solving the one-dimensional bin packing problem was presented by Munien *et al.* [12]. A comparison was made between the Genetic Algorithm, Firefly Algorithm, Cuckoo Search Algorithm, Artificial Bee Colony Algorithm, and some of their hybrids, and they demonstrate that when deciding which underlying heuristic is better, a trade-off between solution quality and processing time must be made. It is suggested that other cutting-edge techniques for computational intelligence, such as the Bees Algorithm, be investigated for solving the one-dimensional BPP.

Bees Algorithm (BA) is a well-known metaheuristic that mimics the foraging behaviours of honeybees, introduced by Pham *et al.* [13]. BA has been widely used to solve many optimisation problems since it can find near-optimal solutions within a reasonable computation time. Engineering, bioinformatics, business, and computer network are a few of the numerous fields where BA has been used to optimise a wide range of problems [14]. Zeybek *et al.* proposed BA as training algorithms for deep learning models [15]. Previous research has demonstrated that, despite the fact that continuous and combinatorial optimisation problems have very different solution strategies, BA was able to find feasible solutions in both continuous and combinatorial domain. The first combinatorial version of BA was proposed in 2007 for job scheduling [16]. BA has been proposed to solve fundamental combinatorial optimisation problems such as the travelling salesman problem (TSP) [17, 18, 19] and the vehicle route problem [20, 21]. Previous research on combinatorial problems demonstrates

* Corresponding author.
Email: natalia.hartono@uph.edu

Received: 5 March 2022; Revision: 22 April 2022;
Accepted: 25 April 2022; Available online: 25 April 2022
http://dx.doi.org/10.36055/jiss.v7i2.14387

that the Bees Algorithm outperforms other metaheuristic algorithms, including TSP [14], robotic disassembly [22], and many other applications [23].

BA, as previously mentioned, performs well in other combinatorial problems; however, because the no free-lunch theorem states that no algorithm can solve all optimisation problems, we are motivated to use the Bees Algorithm to find the near-optimal solution for the bin packing problem. To the best of our knowledge, no previous research has used BA to solve this classic problem. The purpose of this research is to use the basic Bees Algorithm to find a near-optimal solution to another fundamental optimisation problem, BPP.

2. Problem formulation

Illustration of BPP presented in the Figure 1. The illustration shows that each bin has a weight capacity of 20-unit weight. The weight of the items in the bin must not exceed the capacity. The first bin weighs a total of 19, while the second and third bins weigh a total of 18-unit weight. The bin packing problems' objective is to pack the a items in the lists (L) into the fewest possible bins while ensuring that the sum of the sizes in each bin does not exceed the capacity (C). A set of n items, L = (1, ..., n), with each item has positive weight (ω_i) and a set of bins, B = {1, ..., b} with each bin has same capacity. Equation 1 provides the objective for the bin packing problems and constraints presented in Equations 2–5 [24].

$$Objective\ f = min \sum_{j=1}^{b} y_j \quad (1)$$

subject to

$$\sum_{i=1}^{a} \omega_i x_{ij} \le C y_j \qquad \forall j \in B = \{1, \dots, b\} \quad (2)$$

$$\sum_{j=1}^{b} x_{ij} = 1 \qquad \forall i \in L \quad (3)$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in L, \forall j \in B \quad (4)$$

$$y_j \in \{0,1\} \qquad \forall j \in B \quad (5)$$

Equation 2 ensures that total item weights in bin j do not exceed capacity. Each item must be packed into exactly one bin, as specified by Equation 3. Equations 4 and 5 ensure that x_{ij} and y_j are both binary numbers.

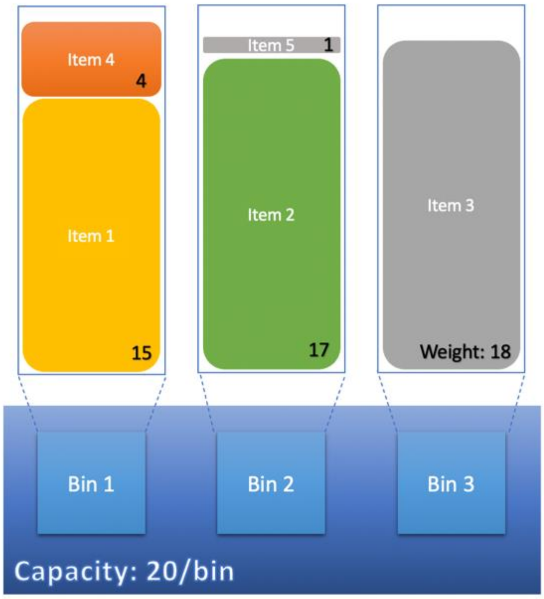

Figure 1. BPP illustration

3. Bees algorithm

The Bees Algorithm was inspired by the honey bees' foraging behaviour [13]. BA is composed of five parameters: the number of scout bees (n), the number of elite bees (nep), the number of best bees (nsp), the number of elite sites (e), and the number of best sites (m). This study uses the best parameter setting based on previous research on parameter tuning for combinatorial problem with a case study on TSP [25]. The BA parameter as follows: n = 40, m = 20, e = 8, nsp = 10, and nep = 40, iteration = 3000 [17, 25].

Figure 2 represents the pseudocode of the basic Bees Algorithm for BPP (see Figure 2). Local search operator in this study uses the basic BA [17]: swap, insert and reverse operator. The stopping criteria uses maximum number of iterations. The procedure begins by randomly generating starting solutions with n scoutbees and then sorting the population. The nep bees search on elite sites (e), whereas the nsp bees search on selected sites (m). Both are local search and employ the swap, insert, and reverse operators. Global bees ($n - m$) search in the remaining search space at random. The findings are sorted according to their fitness value, and the best bees are selected until the stopping criteria is met.

4. Experiments result and discussion

Twenty 1-D BPP datasets from OR-library use to evaluate the basic BA in BPP [26]. The datasets listed in Table 1. The experiment for each dataset run 10 times. The BA programmed in MATLAB 2020b. The appendix provides a link to the MATLAB code for the 1D bin packing problem using the basic bees algorithm, which the user can use and test with other datasets.

Algorithm 1: The pseudo-code of the basic Bees Algorithm (BA) for combinatorial domains.

```
Input: n: ScoutBee, m: SelectedBee, e: EliteBee,
nsp: SelectedSitesBee, nep:SelectedEliteSitesBee
Function BA(nScout,m,e,nsp,nep):
  population ← InitialSolutions(ScoutBee)
  while stopping criterion not met do
    Evaluate fitness of the population
    Sort population according to fitness function
    Select m best solution for local search
    // Generate local solutions with combination of
    search operators
    for each Bee ∈ e do
      for each Bee ∈ nep do
        localBee ← LocalSearch(Bee)
        localBee ← Fitness(localBee)
        if localBee better than Fitness(Bee) then
          // Update Bee
          Bee = localBee
        end
      end
    end
    for each Bee ∈ m − e do
      for each Bee ∈ nsp do
        localBee ← LocalSearch(Bee)
        localBee ← Fitness(localBee)
        if localBee better than Fitness(Bee) then
          // Update Bee
          Bee = localBee
        end
      end
    end
    // Assign remaining bees for global search)
    for each Bee ∈ n − m do
      globalSolutions ← GenerateRandomSolutions(Bee)
    end
    Evaluate fitness of the new population
    Select Best Bee from the new population
  end
  return BestBee
```

Figure 2. BA pseudo-code [22]

Table 1.
Datasets information

Datasets	Bin capacity	Number of items
U120_00	150	120
U120_01	150	120
U120_02	150	120
U120_03	150	120
U120_04	150	120
U120_05	150	120
U120_06	150	120
U250_00	150	250
U250_01	150	250
U250_02	150	250
U250_03	150	250
U250_04	150	250
U250_05	150	250
U500_00	150	500
U500_01	150	500
U500_02	150	500
U500_03	150	500
U1000_00	150	1000
U1000_01	150	1000
U1000_02	150	1000

Table 2.
Experiment results

Datasets	Number of Bins - Best Known Solutions (BKS)	Proposed BA				
		Average run time (seconds)	Average number of bins	Best number of bins	Average accuracy	Best accuracy
U120_00	48	505.4187	50.5	50	5.2%	4.2%
U120_01	49	347.3972	51.8	51	5.7%	4.1%
U120_02	46	348.4024	48.7	48	5.9%	4.3%
U120_03	49	409.5740	52.1	51	6.3%	4.1%
U120_04	50	416.4470	51.9	51	3.8%	2.0%
U120_05	48	374.9484	51.1	50	6.5%	4.2%
U120_06	48	366.0720	51	51	6.3%	6.3%
U250_00	99	852.4341	107.2	106	8.3%	7.1%
U250_01	100	837.4328	107.5	106	7.5%	6.0%
U250_02	102	979.4737	109.9	109	7.7%	6.9%
U250_03	100	811.1937	108.5	108	8.5%	8.0%
U250_04	101	845.1696	108.8	107	7.7%	5.9%
U250_05	101	823.4550	109.3	107	8.2%	5.9%
U500_00	198	2082.2044	218.8	217	10.5%	9.6%
U500_01	201	1693.3054	221.7	219	10.3%	9.0%
U500_02	202	1781.9435	222	220	9.9%	8.9%
U500_03	204	1738.4321	224.6	223	10.1%	9.3%
U1000_00	399	5340.8642	448.2	445	12.3%	11.5%
U1000_01	406	4042.4125	455	452	12.1%	11.3%
U1000_02	411	4036.4744	457.8	454	11.4%	10.5%

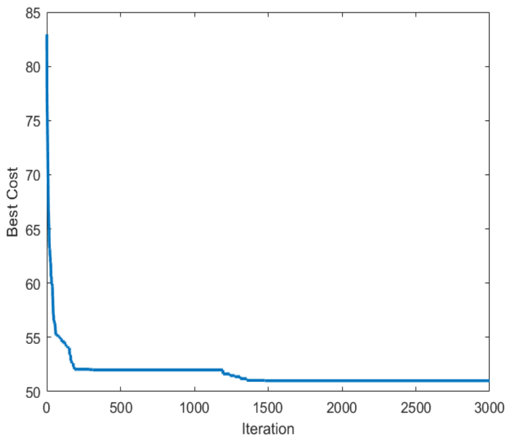

Figure 3. Graph results (dataset U120_00)

The experiment results presented in the Table 2. Figure 3 show the graph results of dataset U120_00. The accuracy measure how close the result from Best Known Solution (BKS). The average accuracy is calculated by subtracting the average from BKS and dividing the results by BKS. The same calculation to find the best accuracy is performed. The results show that for all datasets, basic BA for BPP have average accuracy starts from 3.8% to 12.3% and best accuracy starts from 2.0% to 11.5%. As the number of items increases, the accuracy decreases, which is true for all NP problems due to the exponential growth of solution spaces as the number of problems increases. With the same parameter settings, the basic BA for the Traveling Salesman Problem (TSP) demonstrates that the accuracy measurement also produces similar results. Average accuracy ranges from 1.94% to 3.33% for datasets with 100 cities, and from 5.08% to 10.47% for datasets with 150-200 cities [16].

5. Conclusion

In this paper, we introduce a 1-D bin packing problem solved using metaheuristic approach. We have shown that the Bees Algorithm can be applied to solve this problem. Experimental studies were performed for 20 different BPPs selected from the OR-library. As previously stated, the optimal parameter setting with balanced proportions of scout bees produces the best results. While it is true that this study utilised a balanced proportion, the study could be expanded by utilising a different number with a balanced proportion and statistically comparing the results. Future research will focus on developing an enhanced version of BA for the 1-D BPP. Additionally, the BA can be used to test for 2-D and 3-D BPP.

Acknowledgement

The first author would like to thank the Indonesian Endowment Fund for Education (LPDP) for their support of her study. The authors thank the reviewers for their valuable feedback that help to improve this paper.

Appendix

The code in this work can be found in https://github.com/NataliaHartonoFung/Binpacking-Bees-Algorithm.git.

References

[1] A. Ekici, "Bin packing problem with conflicts and item fragmentation," *Computers & Operations Research*, vol. 126, p. 105113, 2021, doi:10.1016/j.cor.2020.105113.

[2] J. Pereira, "Procedures for the bin packing problem with precedence constraints," *European Journal of Operational Research*, vol. 250, no. 3, pp. 794–806, 2016, doi:10.1016/j.ejor.2015.10.048.

[3] M. Abdel-Basset, G. Manogaran, L. Abdel-Fatah, and S. Mirjalili, "An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems," *Personal and Ubiquitous Computing*, vol. 22, no. 5, pp. 1117–1132, 2018, doi:10.1007/s00779-018-1132-7.

[4] X. Schepler, A. Rossi, E. Gurevsky, and A. Dolgui, "Solving robust bin-packing problems with a branch-and-price approach," *European Journal of Operational Research*, vol. 297, no. 3, pp. 831–843, 2022, doi:10.1016/j.ejor.2021.05.041.

[5] P. Shaw, "A constraint for bin packing," in *International conference on principles and practice of constraint programming*, 2004, pp. 648–662, doi:10.1007/978-3-540-30201-8_47.

[6] R. Kramer, M. Dell'Amico, and M. Iori, "A batching-move iterated local search algorithm for the bin packing problem with generalised precedence constraints," *International Journal of Production Research*, vol. 55, no. 21, pp. 6288–6304, 2017, doi:10.1080/00207543.2017.1341065.

[7] E. G. Coffman, J. Csirik, G. Galambos, S. Martello, and D. Vigo, "Bin packing approximation algorithms: Survey and classification," in *Handbook of Combinatorial Optimization*, vol. 1–5, 2013, pp. 455–531, doi: 10.1007/978-1-4419-7997-1_35.

[8] E. Falkenauer, "A hybrid grouping genetic algorithm for bin packing," *Journal of heuristics*, vol. 2, no. 1, pp. 5–30, 1996, doi: 10.1007/BF00226291.

[9] A. Stawowy, "Evolutionary based heuristic for bin packing problem," *Computers & Industrial Engineering*, vol. 55, no. 2, pp. 465–474, Sep. 2008, doi: 10.1016/j.cie.2008.01.007.

[10] Q. Liu et al., "Algorithms for the variable-sized bin packing problem with time windows," *Computers & Industrial Engineering*, vol. 155, p. 107175, May 2021, doi: 10.1016/j.cie.2021.107175.

[11] F. Luo, I. D. Scherson, and J. Fuentes, "A novel genetic algorithm for bin packing problem in jmetal," in 2017 IEEE International Conference on Cognitive Computing (ICCC), 2017, pp. 17–23.

[12] C. Munien, S. Mahabeer, E. Dzitiro, S. Singh, S. Zungu, and A. E.-S. Ezugwu, "Metaheuristic Approaches for One-Dimensional Bin Packing Problem: A Comparative Performance Study," *IEEE Access*, vol. 8, pp. 227438–227465, 2020, doi: 10.1109/ACCESS.2020.3046185.

[13] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, "- The Bees Algorithm — A Novel Tool for Complex Optimisation Problems," , Ed.D. T. Pham, E. E. Eldukhri, and A. J. Soroka Oxford: Elsevier Science Ltd, 2006, pp. 454–459.

[14] A.H. Ismail, "Enhancing the Bees Algorithm using the traplining metaphor," Ph.D. dissertation, Dept. Mech. Eng., Univ. of Birmingham, Birmingham, WM, 2021.

[15] S. Zeybek, D. T. Pham, E. Koç, and A. Seçer, "An Improved Bees Algorithm for Training Deep Recurrent Networks for Sentiment Classification," *Symmetry*, vol. 13, no. 8, p. 1347, Aug. 2021, doi: 10.3390/sym13081347.

[16] D.T. Pham, E. Koc, J. Lee, and J. Phrueksanant, "Using the bees algorithm to schedule jobs for a machine," in Proceedings Eighth International Conference on Laser Metrology, CMM and Machine Tool Performance, LAMDAMAP, Euspen, UK, Cardiff, 2007, pp. 430–439.

[17] A. H. Ismail, N. Hartono, S. Zeybek, and D. T. Pham, "Using the Bees Algorithm to solve combinatorial optimisation problems for TSPLIB," *IOP Conference Series: Materials Science and Engineering*, vol. 847, no. 1, p. 12027, Apr. 2020, doi: 10.1088/1757-899X/847/1/012027.

[18] S. Zeybek, A. H. Ismail, N. Hartono, M. Caterino, and K. Jiang, "An Improved Vantage Point Bees Algorithm to Solve Combinatorial Optimization Problems from TSPLIB," *Macromolecular Symposia*, vol. 396, no. 1, p. 2000299, 2021, doi: https://doi.org/10.1002/masy.202000299.

[19] M. ŞAHİN, "Improvement of the Bees Algorithm for Solving the Traveling Salesman Problems," Bilişim Teknolojileri Dergisi, vol. 15, no. 1, pp. 65–74.

[20] A. H. Ismail, N. Hartono, S. Zeybek, M. Caterino, and K. Jiang, "Combinatorial Bees Algorithm for Vehicle Routing Problem," *Macromolecular Symposia*, vol. 396, no. 1, p. 2000284, 2021, doi: https://doi.org/10.1002/masy.202000284.

[21] C. Natalia, V. Triyanti, G. Setiawan, and M. Haryanto, "Completion of Capacitated Vehicle Routing Problem (CVRP) and Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) Using Bee Algorithm Approach to Optimise Waste Picking Transportation Problem," Journal of Modern Manufacturing Systems and Technology, vol. 5, no. 2, pp. 69–77, 2021.

[22] J. Liu, Z. Zhou, D. T. Pham, W. Xu, C. Ji, and Q. Liu, "Robotic disassembly sequence planning using enhanced discrete bees algorithm in remanufacturing," *International Journal of Production Research*, vol. 56, no. 9, pp. 3134–3151, May 2018, doi: 10.1080/00207543.2017.1412527.

[23] B. Yuce, M. S. Packianather, E. Mastrocinque, D. T. Pham, and A. Lambiase, "Honey Bees Inspired Optimization Method: The Bees Algorithm," *Insects*, vol. 4, no. 4, pp. 646–662, Nov. 2013, doi: 10.3390/insects4040646.

[24] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer implementations*. John Wiley and Sons Ltd: Chichester, UK, 1990.

[25] N. Hartono, A. H. Ismail, S. Zeybek, M. Caterino, K. Jiang, and M. Sahin, "Parameter Tuning for Combinatorial Bees Algorithm in Travelling Salesman Problems," presented at the 13th ISIEM, Bandung, West Java, Indonesia, July 28th, 2021, Paper 22.

[26] J. E. Beasley, June 1990 to February 2018, "OR-Library," [Online]. Available: http://people.brunel.ac.uk/~mastjjb/jeb/info.html.