Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/4/20230489

A variable neighborhood search framework combining improved simulated annealing search to solve the one-dimensional crating problem

Gaoyang Qiao

University of Nottingham Ningbo, Zhejiang, China, 315154

scygq1@nottingham.edu.cn

Abstract. The N-P problem, similar to bin packing, has a great impact on us in real life. Similar to the classical combinatorial problem, with the expansion of the problem size, usually there is no way to solve it by enumeration. The common solution is to solve by heuristic methods, in addition to the improved local search method by forbidden search or simulated annealing to solve. However, due to the multivariate nature of the bin packing problem, it is difficult to solve most of the bin packing problems by the above single method. In this paper, a variable neighborhood search combined with an improved local search is used to solve the one-dimensional bin packing problem. The method in this paper is generalizable and can easily improve the existing method to suit new problems, and the results are acceptable.

Keywords: Variable Neighborhood Search, Bin Packing Problem, Hyper Heuristic, N-P Problem, Simulated Annealing, Local Search.

1. Introduction

The bin packing problem is a classical combinatorial optimization problem.The bin packing problem has a wide range of applications, and the study of it has profoundly affected many industries. For instance, the supply chain profit maximization problem and the port container allocation problem. bin packing problem refers to minimizing the use of a bin of finite capacity so that it can hold all items. Bin packing problems have been proven to be NP-hard problems, and as the size of the problem increases, the bin packing problem will become increasingly difficult to solve. For similar problems, it is difficult or even impossible to solve them using traditional enumeration search methods, so intelligent search methods have become a popular solution to such problems. There are many variants of the bin packing problem, such as taking into account the shape, density, price, etc. But in this paper, the author will focus only on the one-dimensional bin packing problem. In this paper, it will apply VNS also known as variable neighborhood search) to solve the bin packing problem.

2. Related work

Bin packing problems with larger problem sizes are mainly solved using heuristics. In near-optimal bin packing algorithms, Johnson used the offline First Fit Decreasing algorithm[1]. Offline in FFD means to decrement all items by in FFD means sorting all the items in decreasing capacity size and then sorting them using First Fit, where First Fit is loading the items in order into the first unfilled box knowing that it cannot be filled, thus opening a second box to repeat the process. In addition, there is Best Fit, and Best Fit Decreasing algorithms. Best Fit means prioritizing the largest volume of items that can fit into the current box so that it fills the current box as much as possible. The addition of the offline algorithm increases the speed of the search to varying degrees. Maiza reviewed the MBS, also known as Minimum Bin Slack, and proposed the Adapted MBS heuristic, proposed by Gupta, which intends to find the optimal subset that can fill the current box[2]. The MBS algorithm traverses the objects to be boxed, and each step aims to find the set of objects that best fits the current box capacity. The algorithm terminates when the box is completely filled or when there are no more objects that can fit into the given box[3]. Before each assignment in the heuristic AMBS, we test the compatibility of the currently rented items with the already taken items, then execute the MBS procedure by first arranging the items into a list I′ in decreasing order of size until the list I′ contains no items. Use the MBS blanket search procedure on I′ to determine the best subset by testing the compatibility of the current items with the considered items. Remove the newly created subset from set I. This procedure performs well in real-world problems with a large number of items, but it suffers greatly in larger problems[4]. Its heuristic selection mechanism will dynamically adjust the priority of different heuristics, which is made during the search process. Whether any given heuristic is superior to other heuristics and performs better than others is unknown to the heuristic selection mechanism. Based on this situation, it is a random program that treats all lower level heuristics equally and makes heuristic selection decisions. The heuristic selection mechanism will adapt to different low-level heuristic algorithms by learning some of their historical performance. This process is carried out in the search process, and starts applying priorities between them. In this case, it is more likely to choose a heuristic method that performs well. The other is to meet the simulated annealing acceptance criteria, which is also necessary for the successful application of the selected heuristic. Then, the information about the acceptance decision of the acceptance criteria will be fed back to the heuristic selection mechanism to make better decisions in the future. Finally, we need to use short-term memory. Their role is that each low-level heuristic shows different performance levels in different regions of the search space or at different times of the simulated annealing process. There may also be a situation where some areas of the search space are effective but others are not performing well. If the heuristic algorithm can improve the current solution in the initial stage of the search, it will also be effective in the middle or final stage of the annealing process, which is not certain.

3. Problem description

The one-dimensional boxing problem requires that a series of items whose volume does not exceed the bin be packed into the bin in a particular arrangement and that the number of bins be as small as possible. It can be expressed in mathematical notation in the following form:

$$minimize\ B=\sum_{i=1}^{n} y_i$$

$$subject\ to\ B \ge 1,$$

$$\sum _{j=1}^na_jx_{ij} \le Vy_i,\forall i\in \{1,...,n\}$$

$$\sum_{i=1}^nx_{ij}=1,\forall j \in \{1,...,n\}$$

$$y_i \in \{0,1\}, \forall i \in \{1,...,n\}$$

$$x_{ij}\in \{0,1 \}, \forall i \in \{1,...,n\},\forall j \in \{1,...,n\}$$

In this paper, it is assumed that the size of items is determined by their volume only, and that all items are smaller than the volume of the bin.

4. Method

4.1. System composition

Encoding. The dictionary type and lambda expression in Python are used by the article to represent the number of items in the instance, the serial number of the item, the maximum capacity of each bin, and the initial serial number of the bin. The serial numbers of the items are encoded together with the volumes to track their movement later.

DOI: 10.54254/2755-2721/4/20230489

Initialization. This paper, firstly, uses an item to occupy a bin to initialize the bin list. Then it traverses all the items, uses the items volume accounted for by the largest items loaded exchange into the first box (known as seed) and then gradually finds the second largest can fill the container items and so on using Best Fit Decreasing to exclude the first answer, for the later VNS algorithm The VNS algorithm provides a primitive solution.

Neighborhood definition. In order to create a VNS heuristic for this optimization problem, it is important to first define a neighborhood. In discrete optimization problems containing binary variables, the neighborhood of the solution can be obtained by making some simple modifications, for example, by supplementing one of its components, or by supplementing it with two complementary components(i.e., setting one component from 1 to 0 and the other from 0 to 1)[5]. In contrast, in the boxing problem, we define the neighborhood as the change in the volume of the box associated with the number of boxes.

For the selection of heuristic algorithms, in the first place, Nest Fit and Best Fit, Next Fit, and Worst Fit were tried, but the results were not satisfactory. In this thesis, the algorithm with better results of sorting before searching, which is Best Fit Descend, was finally chosen to generate the initial solution. However, the difficulty is how to obtain the domain solution. It is unrealistic to exchange each item repeatedly in large-scale problems. In that case, the time will be too sensitive, and the algorithm will degenerate into an enumeration method. Moreover, these methods cannot propose a good shaking operator due to the lack of conceptual appreciation in designing the shaking.

The three different neighborhood structures that we use to yield domain solutions in this paper are.

-Swap: Two items are randomly selected from different bins that already have items but are not yet completely filled to swap them and evaluate them for a total of several times (less than eight times) Each swap is completed using BFD to rearrange the unfilled bins again.

Reversal *Order:* items are processed in sequence from large batch to small batch, then to small batch and then to large batch. The start of the reversal To determine the length of the reversal, the input parameters are selected randomly. Next, use BFD to rearrange the reverse order further.

-Exchanging the largest bin. The items of two unfilled bins are exchanged, and the spatial variance before and after the exchange is calculated using the evaluation function so that the volume of one of the bins is smaller than before the exchange.

Better solutions are immediately accepted and repeated as further expansions on them to obtain new neighborhood solutions knowing that the stopping condition is reached.

Search method. Since the problem size is so large that the results cannot be obtained directly by enumerative search methods, this paper uses an improved local search to obtain the approximate optimal solution[6]. The simulated annealing algorithm uses a probabilistic mechanism to control the acceptance process of the solution. Good solutions are accepted unconditionally, and poor solutions are accepted with a certain probability. This allows the simulated annealing algorithm to have more chances to jump out of the local optimum[7].

Fitness function. This paper will use two different evaluation methods

Total bin counts

Volume variance of bins from before to after exchange, which is max\ f(x) =\sum_{\alpha=1}^m=(l(\alpha))^2

The first method is the criterion for the final solution we want, however, sometimes this is inaccurate, as the exchange between two bins often does not change the total number of bins used, however, the state of their internal space cannot be measured, and sometimes one of the bins is emptier after the exchange so that it can accommodate other items, which is what we need, which is why the second evaluation method is used.

Stopping criteria. Time was used as a stopping criterion when the preset 30 seconds was not exceeded as a limit. The minimum time for the local search was replaced with multiple random walks. If there is no optimal solution in all of the neighbors, the search is similarly stopped.

Shaking operator. The distribution is used to obtain random solutions from different neighborhood structures in the shaking step. Uniform distributions in different neighborhoods are a manifestation of the existence of better solutions. However, there is a possibility that other distributions may bring better performance in some problems[8]. Implementing more aggressive random walks with the shuffle function in Python. Use the shaking function to randomly generate a solution for the kth neighborhood.

Solving process. First, the specific problem will be encoded as a classical one-dimensional bin packing problem, then a primitive solution will be obtained using the heuristics presented in this paper, followed by setting the parameters of the VNS and setting the end condition, which will be a 10-minute running time. Three different domain structures were created. The second domain structure uses the second evaluation function above. The purpose of this is that other switching operators can find potential storage units that can be reduced. This can further reduce the use of storage units and the shaking function is used to further change the domain structure after each search pass[9]. This is shown in the figure below.

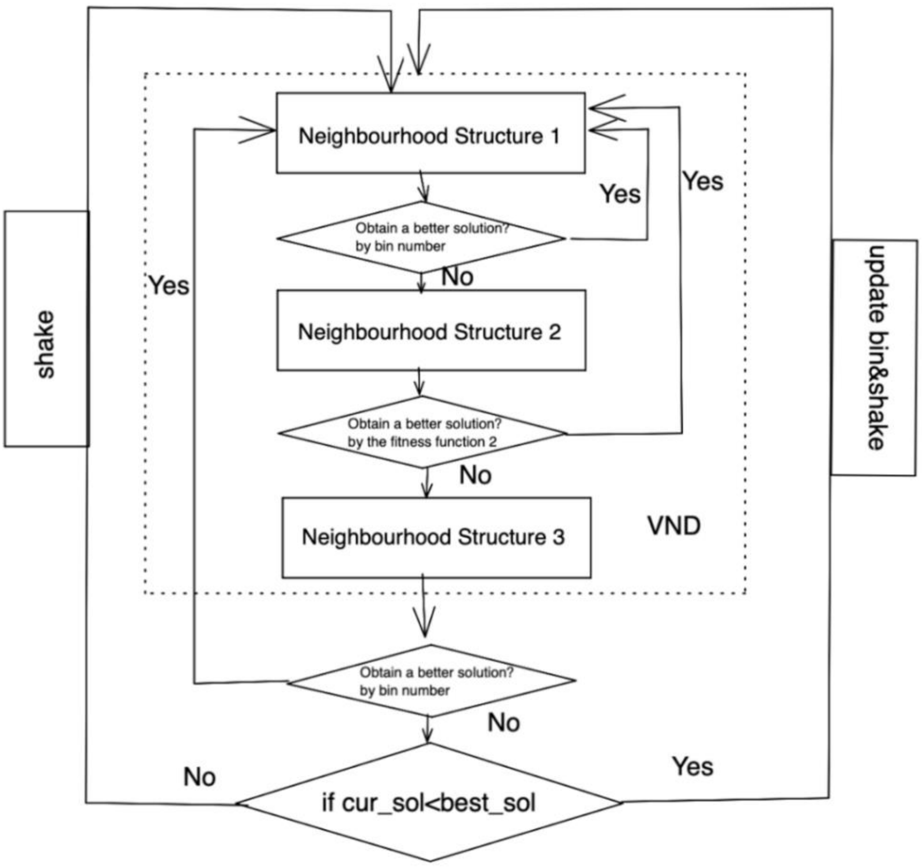

Figure 1. VNS workflow.

4.2. Data

This paper uses the bin packing dataset from a website called euro-online for 120 problems with items of 150 volume, 500 problems with items of 150 volume, and 200 problems with items of 10, 0000 volume respectively experiments[10].

4.3. Statistical result

In the first two problems with smaller problem sizes, essentially all problems are different from the known optimal solutions, which is between 0 and 1. The experimental results were within 1 gap of the known optimal solution. The optimal solution was found in 18 of the 20 experiments conducted, with

DOI: 10.54254/2755-2721/4/20230489

an optimal efficiency of 90%. Most instances can achieve abs<=1. Good results are positively correlated with the number of iterations, which is the result of the last large-scale problem. The maximum gap between the experimental results and the known optimal solution is 3. There are 12 times that the gap is less than or equal to 1 in the 20 trials conducted, and the efficiency of gap <= 1 is 60%.

5. Discussion

Since there is no better heuristic algorithm, the program runs progressively slower and the performance decreases as the problem scales up. The convergence of the solution is poor due to the use of similar heuristics between the generated solutions of different domains, which also leads to poor convergence. The construction of shaking structure needs to be further enhanced, too many single perturbation operators will affect the performance of this algorithm. It is necessary to add diversity into the design of vibration operators in order to generate more abundant neighbor structures. This is also for further VND search. Both MBS and MBS algorithms can be used to obtain improved default solutions. This can reduce the time required for subsequent algorithms, and also improve the convergence of the domain structure, which uses variance as the fitness equation. The maximum variance within a box is used as the evaluation function of the disabled search algorithm. This operation further reduces the time required for searching using simulated annealing.

6. Conclusion

In conclusion, the bin packing problem is solved using a modified local search, which is simulated annealing for bin packing combined with a variable neighborhood search. The method in this paper is simple to implement, highly refinable, does not easily fall into local optimum, and gives acceptable results for problems of large size. The key to using variable neighborhood search lies in the construction of the neighborhood structure, and heuristics are the key to neighborhood construction. In this paper, the lack of excellent heuristics leads to poor search efficiency in the face of problems that partially contain items with too large a volume items. The shortcomings of this aspect can be improved by applying more effective heuristics. Compared with traditional heuristic algorithms and local search algorithms written for a specific problem, the method in this paper is more compatible and can adopt traditional heuristics internally. It is also more adaptable in the face of new problems, and the system is highly robust due to the clear division of labor and has high practical value in bin packing problems.

Acknowledgment

First of all, I will express my gratitude to my Professor. Without whose teachings I could not have completed this assignment. I would also like to thank Guo, my close friend in the same department, for his advice and encouragement to let me keep trying new approaches , which has been a great inspiration for my research. Finally, I want to thank my family for being able to support me in my studies all the time.

References

[1] Ullman, J. D. (1971). The performance of a memory allocation algorithm. Technical Report 100 Princeton Univ, 59-61.
[2] Mohamed Maiza. (2011). Heuristics for Solving the Bin-Packing Problem with Conflicts, Applied Mathematical Sciences, Vol. 5, 2011, no. 35, 1741-1743.
[3] J.N.D. Gupta and J.C. Ho. (1999). A new heuristic algorithm for the one- dimensional bin-packing problem, Production Planning&Control, 10(6), 598-602.
[4] Bai, R., Blazewicz, J., Burke, E.K. et al. (2012). A simulated annealing hyper-heuristic methodology for flexible decision support. 4OR-Q J Oper Res 10, 43–66.
[5] Sonia Cafieri, Pierre Hansen, Nenad Mladenovic. (2014). Edge-Ratio Network Clustering by Variable Neigh-borhood Search. European Physical Journal B: Condensed Matter and

DOI: 10.54254/2755-2721/4/20230489

Complex Systems, Springer-Verlag, 87 (5), pp 116.
[6] VijayV. Vaziran. (2001). Approximation Algorithms, Springer. ISBN: 978-3-540-65367-7, 73-74.
[7] Pincus, Martin (Nov–Dec 1970). A Monte-Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems, Operations Research. 18(6):1225-1228.
[8] Michel Gendreau. (2010). Handbook of Metaheuristics, Springer, 64-69.
[9] Nenad Mladenović; Pierre Hansen. (1997). Variable neighborhood search. Computers and Operations Research, 6-8.
[10] Association of European Operational Research Societies, 2022, data sets esicup, Retrieved from https://www.euro-online.org/websites/esicup/data-sets/.