ISSN: 0975-766X
CODEN: IJPTFI
Available Online through
www.ijptonline.com
Research Article

BIN PACKING PROBLEMS: COMPARATIVE ANALYSIS OF HEURISTIC TECHNIQUES FOR DIFFERENT DIMENSIONS

BoominathanPerumal [*1], RishinHaldar[*2], S. Rajkumar[*3]
[*1,2,3]School of Computer Science and Engineering, VIT University, Vellore, India.
Email:bominathan@yahoo.com

Received on 13-05-2016 *Accepted on 12-06-2016*

Abstract:

Bin packing is a NP-hard combinatorial problem which provides a certain placement of non-overlapping objects in a container in such a way that one or more objectives are optimised. This packing arrangement has many variants in dimension, namely single dimensional, two dimensional or multi-dimensional. The problem objectives can be as simple as minimizing the number of bins to a more complex one like considering a trade-off between the number of bins. Problems can also be categorized into optimization of a single objective or multi objective. Bi objective optimization is considered as a multi objective one in most of the studies, presented in this review. Solution to these problems can be categorised in many ways. In addition to the limited number of exact methods, the extensive literature on bin packing categorises the heuristic solution approaches into three categories, namely heuristics and approximation algorithms, meta heuristics and hyper heuristics. This article mainly focuses on surveying the effectiveness and applicability of these three categories of heuristic approaches on 2D and 3D packing with both single and multi-objective optimization.

Keywords: Bin packing problems survey; variants of bin packing; heuristic approaches; approximation algorithms; meta heuristic approaches; hyper heuristic approaches

I. Introduction

Packing problems have been widely applicable in practical situations of day to day lives as well as in industries, where an optimal packing arrangement can lead to a significant cost reduction. This is the reason why extended mathematical analysis is being done to improve the solutions to packing problems. According to Wascher[1], packing problems can be loosely defined by having two sets of objects, one set for large objects and the other one for small items. The objective is to select some or all small items within one or more large objects, subject to the conditions that the small items, in its entirety, will fit within the large object(s) and the small items cannot overlap.. Obviously, the challenge is to maximize the number of small items in the minimum number and/or size of the large object(s).

The fundamental nature of the packing problem can be applied to many applications and hence this has led to different variants which are specifically suited to a particular domain or application. For example, packing problems and cutting problems are basically same [2]. Finding the optimal way to cut a large item into smaller ones is conceptually same as optimally arranging small pieces within a large container. A simple one dimensional cutting stock problem (1DCSP) is where one has large rolls of paper, metal or cloth and the customers would want to purchase smaller different sized rolls of the material, and the challenge is to satisfy all the customers' demand with the minimum length of large rolls.

The one dimensional bin packing problem (1DBPP) is closely related to 1DCSP [2]. In this 1DBPP scenario, a set of i items, each having individual size w_i, are to be packed in bins of size W, and the objective is to find the minimum number of bins that can accommodate all the i items.

The same problems can be applied in a two dimensional perspective. For the two dimensional knapsack packing problem (2DKPP), we take into account both the weight and the height, and the objective is to fit a subset of items in a knapsack, so that the total profit of the items is maximized, subject to the constraint that the items can be included under the total weight W and total height H of the knapsack. [3] 2DKPP is again, very similarto the two dimensional cutting stock problem (2DCSP), where the large roll, with one dimension fixed, and the other one variable, is replaced by a two dimensional piece of material. Manufacturers face a variant of this two dimensional problem, called Pallet Loading Problem (PLP). The idea is to find out an optimal loading pattern for identical sized pieces/boxes, to be loaded on a pallet. Coming to more practical problems in the manufacturing/transportation industries, [2] Container Loading Problem (CLP) is a three dimensional problem, where a set of items has to be packed in a container, such that the minimum volume is wasted in the container. Needless to say, there is a three dimensional variety of the knapsack packing problem (3DKPP) also. Extrapolating further, there is a multi-container loading problem (MCLP) and the multi pallet loading problem (MPLP), where the problem is to minimize the number of containers or pallets, instead of just restricting the packing to only one container or pallet. It is also obvious that in a single container problem (CLP), there may be some items which couldn't be accommodated in the container, and the objective was to minimize the un-accommodated items, while in the MCLP, all items are accommodated and the objective is to minimize the number of containers. [2] Putting these variants of the bin packing from another perspective, one can categorize the problems into three categories. One is where the objective is to minimize one dimension of the container, commonly known as strip packing. Area minimization is the second category, where minimizing the area or volume is the primary objective. The third category is called circle packing, where the challenge is to minimize the radius of a circular container. Strip packing originated from the textile industry, where from a roll of fabric, different components of clothing items need to be cut, and the objective is to place and cut the items such a way that the length of the strip is minimized, provided that he width of the strip is fixed. The 1DCSP problem is a popular example of strip packing.

Area minimization encompasses all the variants of PLP and CLP, where the challenge is to minimize the area or volume of the container, which is large enough to pack a set of items. This problem can also be related to Very Large Scale Integration (VLSI) placement problem where optimal positioning of rectangular modules within a rectangular plate is desired.

Circle packing is also a minimization class of problem, where one must find the minimum radius sized circular container, which can fit in a set of circles. This kind of optimization is relevant to cable or oil pipeline industries.

Exact Approaches

Bin packing problem, and its variants, is an NP-Hard problem. However, there has been substantial effort in trying to find exact approaches to solve these problems, obviously with smaller values of parameters. Downsland [4] initially proposed an exact algorithm for the PLP problem. Later Bhattacharya and Bhattacharya [5] proposed a depth first approach for the same problem. Tarnowski et al.[6] presented a polynomial algorithm for the PLP with the restriction that the placements are required to be in guillotine-cuttable fashion.

Martello et al.[7],[10] proposed an enumerative branch and bound approach to solve the two dimensional strip packing (2SP) problem. After sorting the items in decreasing height, a reduction procedure determines the optimal arrangement for a subset of items, consequently reducing the instance size.

Martello and Vigo [8],[10] also proposed an enumerative method to tackle the two dimensional Bin packing (2BP) problem. The method adopts a two level branching scheme, the outer branch decision tree and the inner one, where the items are loosely assigned to the bins in the outer level and the inner branch decision tree enumerates possible sequences to finalise the positions of the items in the corresponding bin.

Fekete and Schepers [9] modelled the 2 dimensional bin packing problem with Interval Graph (IG) approach, where the items are represented by nodes in the graph, and two nodes are adjacent if their dimensions overlap. By creating two graphs, one for the x coordinates and the other for the y coordinates, if there is a feasible packing arrangement of the objects in a bin, the graphs would exhibit salient properties. This method of representation can be scaled to higher dimensions also.

In Section II, we discuss some of the popular heuristics which are used to solve the bin packing problem or its variants, for both single and multidimensional perspectives. Section III talks about the different meta-heuristics techniques in this field, while some hyper-heuristics approaches for this problem are discussed in Section IV. Finally, we conclude in Section V.

II. Heuristic Approaches

[2][10] The computationally hard nature of bin packing problems has given rise to numerous heuristic approaches to obtain acceptable results in a feasible amount of time. In fact, the literature on heuristics (or variants, like meta-heuristics or hyper-heuristics) approaches far outnumber the ones which use exact approaches for the bin packing problem. The heuristics approaches can be broadly classified into construction based and local search based approaches. The construction based heuristics tries to accommodate one object at a time in the bin till all objects are placed. Theseheuristics range from very simple and naïve methods to moderately developed ones which are used as approximation techniques with guaranteed bounds in terms of optimality. Local search based heuristics, on the other hand, iteratively try to achieve a better packing as a whole. While there are some local search heuristic approaches, they are mainly used alongside meta-heuristic approaches.

Construction Based Approaches

The literature is filled with numerous approximation techniques which make a valiant effort to solve this NP-Hard problem of bin packing. These approaches may not get the optimal answer but it guarantees an upper bound of how much it is close to an optimal solution.

[11] The First Fit (FF) approach tries to place the current object in the very first bin that can accommodate the object. [11] The Next Fit (NF) approach first tries to fit the object in the current bin, if unsuccessful, it assigns a new bin for the object to be placed. The [11] Best Fit (BF) technique tries to fit the current item in the bin which will have the minimum remaining capacity, if the object is placed in that bin. Conversely, the Worst Fit (WF) approach tries to fit the object in the bin with the maximum remaining capacity. Interestingly, Almost Worst Fit (AWF) approach goes for the bin with the second maximum remaining capacity. Literature shows that FF approach fared the worst compared to these methods. All these approaches however, were limited in their success in the one dimensional

scenario only. [12][17]The Decreasing FF algorithm first sorted the objects in non-increasing order with respect to their dimensions and then applied the basic FF approach. This gave better results than its primitive version and was effective in scenarios having more than one dimension also. Consequently similar variants, like Decreasing NF (NFD), Decreasing BF (BFD) and Decreasing WF (WFD) were proposed which also gave better results for both one dimensional and more than one dimensional scenario.

The First Fit Lookup (FFL) [13] is a variant of FF, where a table is used to keep track of the bin where the last object was fit, thus when the next object of similar size is needed to be placed, the algorithm looks up the table to find the bin. Best Fit Lookup (BFL) was also designed on similar lines. They both provided marginally better results than their primitive counterparts.

The study of these basic heuristics brought out the fact that BFD came out to be the best in efficient packing and NFD came out to be the best in speed of packing. It also brought out this contradiction that more efficient packing reduces the speed of packing, and vice versa. In this context, Maxi Min (MM) bin packing algorithm, proposed by Dingju Zhu [14], can match the best of both scenarios with some restrictions on the size repetition rate of packed objects.

Coffman et al. [12] proposed an approximation approach for the 2SP. After the items are sorted in non-increasing order of height, they are packed in levels by means of two algorithms, Next-Fit Decreasing Height (NFDH) and First-Fit Decreasing Height (FFDH). NFDH tries to place the next item on the current level. If it doesn't fit, it tries to place it in the new current level created by a horizontal line on top of the tallest item placed on the old current level. FFDH tries to place the next item on the first level where it can fit. These algorithms can run in O(nlogn). The Best-Fit Decreasing Height (BFDH) algorithm is also used to solve 2SP. BFDH tries to fit the next item, in the same level, so that after placement, there is minimal residual horizontal space. The Bottom-Left (BL) approach [15] attempts to allocate the item in the lowest possible position. In all of these approximation algorithms, placing of items is always left justified.

Chung et al. [16] approached the 2BP with a Hybrid-First Fit (HFF) algorithm, which uses FFDH in the first phase and then uses one dimensional First-Fit Decreasing (FFD) [17] algorithm in the second phase. The Hybrid-Next Fit (HNF) algorithm is very similar to NFDH, the difference being that, even if the preference is to place the item in the next level in the same bin, it can also choose the next bin for placement of the item. Berkey and Wang [18] proposed a Finite First-Fit (FFF) algorithm which modified HFF, such that the item is placed on the lowest level of the first bin where it can be fitted, otherwise new level is created in the first suitable bin, finally, even if that fails, a new bin is initialized.

Local Search Based Approaches

Significant studies are also done on developing local search based heuristics to effectively tackle the bin packing problems. The underlying characteristic of this approach is that even if the local search is synonymous with a greedy approach resulting in a possible suboptimal solution, the results are obtained quickly.

Bengtsson (19) used a local search technique on 2DBPP. After a subset of the objects is placed in the bin to start with, the algorithm iteratively places unallocated objects in the bin having maximum unused space. The objective was to make the number of unallocated (inactive) bins to zero.

Voudouris and Tsang (20) proposed a guided local search technique where information is accumulated from various sources and is utilized to guide the local search to discard non- promising parts of search space, thus making the search faster. Faroe et al (21) used this guided local search approach on 3DBPP. Starting with a defined upper bound on the number of obtained bins, the algorithm iteratively tries to reduce the number of bins, provided the boxes are feasibly packed in the bins.

III. Meta-Heuristic Approaches

A meta-heuristic generates a heuristic for an optimization problem, with imperfect or incomplete information to begin with. There is some degree of randomness and also assumption in the process, but one can recognize a good solution when it is available. Most of the heuristics approaches described in the previous section are construction based and are essentially greedy approaches, which has limitations on global optima. In this context, meta- heuristics approaches are much preferred choice as these can search much larger search space in a feasible manner than simple heuristics and converges to a solution much faster. This enables us to find out an optimal or near optimal solution in a reasonable time. Meta-heuristic approaches usually apply genetic algorithms, ant colony optimization, tabu search, simulated annealing and their variants. In addition, it also combines these approaches with local search heuristics for better results.

Dowsland [22] used a simulated annealing approach on 2SP, to explore both feasible solutions and a neighbourhood area in which items overlap in pairwise manner. Shifting (vertical or horizontal) of the neighbourhood results in reduction of overlap, and improves the solution and consequently upper bound is updated. In the three dimension scenario, Zhang et al (23) proposed a combination of a bin loading heuristic and simulated annealing to solve the

3DBPP. Pisinger (24) used a novel box placement representation called sequence triple and dealt with both 2D and 3D knapsack packing problem using simulated annealing. K.H. Loh et al (25) proposed Weighted Annealing (WA) by using a modified greedy heuristic to assign varying weights to different parts of solution space.

Lodi et al (26) proposed a tabu search algorithm for 2DBPP, where the tabu search controls the movement of objects in between the bins by using two neighbourhood functions. The first one tries to put the objects in one bin or the other, while the second one tries to combine two bins to consolidate all the objects. Lodi (27) later proposed a TSPack, a tabu search unified framework, which can be applied to two dimensional as well as three dimensional scenarios. Crainic et al (28) proposed a two level tabu search, TS2Pack, on 3DBPP. The first level aims at reduction in the number of bins, while the second level tries to optimize the packing inside the bins. It also puts forward a method to increase the size of the neighbourhood without increasing the overall time complexity, this, in turn, improves the quality of the search.

Jakobs [29] approached the 2SP with a genetic algorithm, where the individual is created by a permutation of the order in which the items are packed, with the assumption that the objects would be allocated in Bottom Left (BL) strategy. This technique made the crossover and mutation process simpler. Kroger (30) used an implicit representation in his genetic algorithm for the 2SP, but constrained to guillotine packing with right angle rotations. It used a complex slicing tree structure and consequently a complex crossover process, however, this resulted in linear time packing, and neighbourhood information was utilized for a local optimization in the mutation phase. The results were quite encouraging, compared to simpler simulated annealing approaches. Falkenauer (31) used Grouping Genetic Algorithm (GGA) along with a dominance based local optimization to solve bin packing problems effectively. The GGA is quite a deviation from the classic Holland style GA, and is customised to represent the structure of grouping problems. Gehring et al (32) provided a parallel genetic algorithm for the 3DBPP with heterogeneous cartons/bins. They also showed that their hybrid genetic algorithm (2002) fared better on heterogeneous bins compared to a tabu search. Egeblad et al (24) used a layer building heuristics for a Biased Random Key based Genetic Algorithm (BRKGA) for both 2DBPP and 3DBPP. This method keeps all the bins open for allocation and then utilizes the free space available in the bins to come up with an effective placement process.

Many of the simulated annealing, tabu search and genetic algorithm approaches stress on neighbourhood search to obtain their results. Pisinger et al (33) proposed an Adaptive Large Neighbourhood Search (ALNS) in which destruct and repair sequences were used to generate solutions. Hansen et al (34) summarized Variable Neighbourhood Search

(VNS) algorithms which progressively change the neighbourhood both by local search leading to local minima and also escape from the valleys. Alvarez-Valdes et al (35) presented a Greedy Randomized Adaptive Search Procedure (GRASP) for SP. This search procedure had both a constructive phase and an improvement phase. The constructive phase had greedy approaches to select several objects for the bin at hand, however, the selection was randomized, not deterministic, among the chosen ones. Finally, the improvement phase did the fine tuning. Alvarez-Valdes et al (2010) then utilized both GRASP and Variable Neighbourhood Descend (VND) search for 2D and 3DBPP where the construction phase is guided by a maximum space heuristic and the refinement phase uses a VND structure.

Ant Colony Optimization (ACO) and its variants, in recent times, have also been applied for this class of problems. Levine J. et al (2004) used ACO for both 1DSP and 1DBPP. Silveira et al (38) used ACO for 3DBPP with guillotine cut constraint, with variable sized bins. D.S. Liu et al (39) proposed a Multi-Objective (bi-objective) Particle Swarm Optimization (MOEPSO) algorithm for 2DBPP to handle the twin conflicting objectives of minimizing the number of bins as well as balancing individual load of the bins. This approach incorporated the Pareto optimal set for its answers. The literature also shows the effectiveness of combining more than one meta-heuristic approach. C Blum et al (40) effectively used an evolutionary algorithm framework on an Improved Lowest Gap Fill approach to propose an EA-LGFi algorithm on 2DBPP. The original Lowest Gap Fill approach (LGFi), proposed by Wong et al (41), is a randomized constructive one pass heuristic. In the pre-processing stage, objects, with respect to the area, are sorted in non-increasing order. The packing stage follows the bottom leftmost approach, with an objective to fill up the horizontal or vertical gap completely.

Meta-heuristic approaches and its variants, so far, occupy a very significant share of the literature available on bin packing problems. This section mentions only few of them as token examples. In fact, the recent findings have indicated that using a combination of meta-heuristic approaches or a combination of meta-heuristic and approximation/heuristic approaches generate much better results than individual approaches. It can be stated that applying a standalone meta-heuristic is sometimes effective only in lower dimensions (1D, maximum to 2D), however a combination of approaches gives near optimal results for higher dimensions (for eg., 3D). Hybrid meta-heuristic approaches are also more adept in providing better results for multi-objective constraints.

IV. Hyper Heutistic Approaches

Hyper heuristics essentially aims to automatically arrange the meta-heuristics or heuristics needed to tackle complex search problems. They fundamentally differ from meta-heuristics by the fact that in case of meta-heuristics, the search space consists of the different solutions to the specific problem or application, while in case of hyper heuristics, the search space has heuristic techniques. Furthermore in terms of generality, we have mentioned in the last section that a combination of heuristics/meta-heuristics approach generates much better solution than individual ones. However, the solution is geared towards a specific instance of a problem, for eg., 3DBPP or 3DSP. Hyper heuristics, ideally, is geared towards a higher, more general approach, where the order of heuristics chosen by the hyper heuristics approach should solve more than one problem in a domain, for eg., both 3DKnapsack Problem and 3DBPP.

Burke et al (43) brought out, in detail, the idea of hyper heuristics and also stressed on the salient objective of being able to elevate the generality of the approach for optimization problems. Later, Burke at al (44) proposed a genetic programming framework which automatically generated quality heuristic which tackled both knapsack problem and bin packing problems with rectangular pieces for all one, two and three dimensions. In their 2013 paper, Burke et al categorised hyper heuristics into two classes: selection hyper heuristics and generation hyper heuristics. Selection hyper heuristics selects or chooses existing heuristics and applies them on the problem at hand. Generation hyper heuristics creates new heuristics from components of existing heuristics. It is relevant to point out that in both the cases, a genetic algorithm framework is used.

The initial population consists of individuals who have popular/existing heuristics, the selection hyper heuristics approach chooses good individuals from the existing population by a GA selection mechanism and thus generates a good solution for the problem. The generation hyper heuristics approach performs crossover and mutation on the selected individuals, thus new heuristics are created which have mixed characteristics of heuristics from their parents, and the GA mechanism outputs a good individual which generates a good solution for the problem.

Lopez-Camacho et al (45) presented a unified selection based hyper heuristic framework for 1D and 2D bin packing problems. Basic heuristics like FFD, BFD, Djang and Finch (DJD) were initially used but it generated a fast, deterministic algorithm which bettered the results from the best problem specific heuristics. Jaya Thomas et al (46) also used a genetic algorithm based hyper heuristics approach to effectively handle 2D packing problems, and can also be used for cutting problems.

The hyper heuristics were created from a limited set of low level heuristics like Rightmost placement, MaxDiff (place the rectangle which has the maximum difference of height between itself and the last rectangle already placed in the bin) placement and MinDiff (vice versa) placement.

Gomez et al (47) presented a bi-objective optimization on irregular 2D cutting stock problems by creating a set of Pareto optimal Multi-objective Hyper Heuristics (MOHH). The MOHHs were created by using a Multi-objective Evolutionary Algorithm (MOEA) called NSGA-II (Non-dominated Sorting Genetic Algorithm, version II). The dual objectives were to minimize both the number of sheets needed to fit a fixed number of objects, as well as the time required to perform the fitting.

Kevin Sim et al (48) presented the idea of generating cooperative hyper heuristics which collectively reduce the number of required bins for the 1DBPP using Single Node Genetic Programming (SNGP) island model. Kevin Sim et al (49) then proposed a lifelong learning hyper heuristic method which fuses SNGP methods and generates novel heuristics, which is self-sustaining within a network of available problems and heuristics, and can adapt over time as and when the environment changes or new knowledge is introduced.

Beyaz et al (50) proposed robust hyper heuristic algorithms for 2DBPP by using Memetic Algorithms (MA) with novel crossover and mutation operators for heuristics selection. Memetic Algorithms is a Hybrid Evolutionary Algorithm (HEA) which uses both population based representation and also uses local search techniques. This paper also used state of the art heuristics like Finite First Fit, Finite Next Fit, BFH, Unified Tabu Search, LGFi.

Hyper Heuristic approaches can also be classified in terms of online and offline approaches. Online approaches are more dynamic in nature, where the hyper heuristic readjusts its choice and order of heuristics when a new instance of a problem arrives. Offline approaches create the hyper heuristics based on the complete set of heuristics and problem instances available at the beginning.

Needless to say, the offline approach is much faster and gives better results in terms of both time consumed and optimality. The online approach, on the other hand, needs to modify its hyper heuristic for the changing input scenario, and hence is slower and might not match the level of optimality provided by the offline variety. However, the online approach is more geared to the ideal objective of generality in hyper heuristic approaches, where it can solve more than one problem in a domain, or even across domains in optimization problems.

V. Conclusion

This paper conducted a brief survey of the approaches used to tackle the bin packing problems and its variants, with an eye towards their applicability in one or more dimensions. Even though some exact approaches for the problem were mentioned in this review, the emphasis was on the categorization of the approaches into three main types, the heuristic approaches, the meta-heuristic approaches and the hyper heuristic approaches. We have tried to point out the distinctive features for each of these categories, emphasizing on the importance of each category and why research in this field has evolved from one category to the other.

This survey points out that the shifts from heuristic to meta-heuristic to hyper heuristic approaches show that there is a continuous evolution in the paradigm of problem solving in this bin packing problem domain of combinatorial optimization. On a different note, we also wish to point out that the complex workings of the evolving hyper heuristic approaches, in turn, need basic heuristic approaches only to provide a near optimal or optimal solution to the problem. Therefore, a novel and innovative change in the basic heuristic approaches can have a cascading effect in both the meta-heuristic and hyper heuristic solutions.

The literature shows the magnanimity of the studies done on bin packing problems and its variants, however this survey looks at only few of them as representations for the categories it has classified into. Therefore, this survey, although wide-ranging, is not comprehensive. We hope that this effort has brought out the evolving trends in tackling bin packing problems and would be helpful in finding more novel and innovative techniques.

References

1. G. Wascher, H. Haussner, and H. Schumann. An improved typology of cutting and packing problems.European Journal of Operational Research, 183:1109–1130, 2007.
2. J. Egeblad. Heuristics for Multidimensional Packing Problems.PhD thesis, University of Copenhagen, Department of Computer Science, 2008.
3. M.A. Boschetti, E. Hadjiconstantinou, and A. Mingozzi.New upper bounds for the two-dimensionalorthogonal cutting stock problem. IMA Journal of Management Mathematics, 13:95–119, 2002.
4. K. A. Dowsland. An exact algorithm for the pallet loading problem.European Journal of OperationalResearch, 31(1):78–84, July 1987.
5. S. Bhattacharya and R. Bhattacharya. An exact depth-first algorithm for the pallet loading problem.European Journal of Operational Research, 110(3):610–625, 1998.
6. A. Tarnowski, J. Terno, and G. Scheithauer.A polynomial time algorithm for the guillotine pallet loadingproblem. INFOR, 32:275–287, 1994.
7. S. Martello, M. Monaci, D. Vigo, An exact approachtothe strip packing problem, Technical paper OR/00/18,Dipartimento di Elettronica, Informatica e Sistemistica,Universita di Bologna, 2000.

8. S. Martello, D. Vigo, Exact solution of the two-dimensionalfinite bin packing problem, Management Science 44(1998) 388–399.
9. S.P. Fekete, J. Schepers, On more-dimensional packing III:Exact algorithms, Technical paper ZPR97-290, MathematischesInstitut, Universit€atzuK€oln, 1997.
10. A. Lodi, S. Martello, M. Monaci, Two-dimensional packing problems: A survey, European Journal of Operational Research, 141 (2002), pp. 241–252.
11. E. E. Bischoff and M. D. Marriott.A comparative evaluation of heuristics for container loading. EuropeanJournal of Operational Research, 44:267–276, 1990.
12. E.G. Coffman Jr., M.R. Garey, D.S. Johnson, R.E. Tarjan,Performance bounds for level-oriented two-dimensionalpacking algorithms, SIAM Journal on Computing 9 (1980), 801–826.
13. Francis, Robert, Jonathan Rose, and ZvonkoVranesic. Chortle-crf: Fast technology mapping for lookup table-based FPGAs. Proceedings of the 28th ACM/IEEE Design Automation Conference.ACM, 1991.
14. Dingju Zhu, Max–min Bin Packing Algorithm and its application in nano-particles filling, Chaos, Solitons and Fractals, 2015 (in press).
15. B.S. Baker, E.G. Coffman Jr., R.L. Rivest, Orthogonalpacking in two dimensions, SIAM Journal on Computing9 (1980) 846–855.
16. F.K.R. Chung, M.R. Garey, D.S. Johnson, On packingtwo-dimensional bins, SIAM Journal of Algebraic andDiscrete Methods 3 (1982) 66–76.
17. D.S. Johnson, Near-optimal bin packing algorithms, Ph.D.Thesis, MIT, Cambridge, MA, 1973.
18. J.O. Berkey, P.Y. Wang, Two dimensional finite binpacking algorithms, Journal of the Operational ResearchSociety 38 (1987) 423–429.
19. Bengtsson B. E. Packing rectangular pieces—A heuristic approach. Comput.J. (1982) 25:353–357.
20. C. Voudouris, E. Tsang, Guided local search and its application to the travelling salesman problem, European Journal of Operational Research113 (1999) 469–499.
21. O. Faroe, D. Pisinger, and M. Zachariasen.Guided local search for the three-dimensional bin packing problem. INFORMS Journal on Computing, 15(3):267–283, 2003.
22. K. Dowsland, Some experiments with simulated annealing techniques for packing problems, European Journal of Operational Research 68 (1993) 389–399.

23. De-Fu ZHANG, Sheng-Da CHEN, Yan-Juan LIU,An Improved Heuristic Recursive Strategy Based on Genetic Algorithm for the Strip Rectangular Packing Problem. ActaAutomaticaSinicaVolume 33 (9), (2007), 911–916.
24. Egeblad, Jens, and David Pisinger. Heuristic approaches for the two-and three-dimensional knapsack packing problem. Computers & Operations Research 36.4 (2009): 1026-1049.
25. Loh, Kok-Hua, Bruce Golden, and Edward Wasil. A Weight Annealing Algorithm for Solving Two-dimensional Bin Packing Problems, Operations Research and Cyber-Infrastructure. Springer US, 2009.121-146.
26. Lodi, Andrea, Silvano Martello, and Daniele Vigo. Approximation algorithms for the oriented two-dimensional bin packing problem. European Journal of Operational Research 112.1 (1999): 158-166.
27. Lodi, Andrea, Silvano Martello, and Daniele Vigo.TSpack: a unified tabu search code for multi-dimensional bin packing problems. Annals of Operations Research 131.1-4 (2004): 203-213.
28. Crainic, Teodor Gabriel, Guido Perboli, and Roberto Tadei.TS 2 PACK: A two-level tabu search for the three-dimensional bin packing problem. European Journal of Operational Research 195.3 (2009): 744-760.
29. S. Jacobs, On genetic algorithms for the packing of polygons, European Journal of Operational Research 88 (1996) 165–181.
30. Kröger, Berthold. Guillotineable bin packing: A genetic approach. European Journal of Operational Research 84.3 (1995): 645-661.
31. Falkenauer, Emanuel. A hybrid grouping genetic algorithm for bin packing.Journal of Heuristics 2.1 (1996): 5-30.
32. Gehring, Hermann, and Andreas Bortfeldt. A parallel genetic algorithm for solving the container loading problem. International Transactions in Operational Research 9.4 (2002): 497-511.
33. Ropke, Stefan, and David Pisinger.An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science 40.4 (2006): 455-472.
34. Hansen, Pierre, NenadMladenović, and José A. Moreno Pérez.Variable neighbourhood search: methods and applications. Annals of Operations Research 175.1 (2010): 367-407.
35. Alvarez-Valdés, Ramón, Francisco Parreño, and José Manuel Tamarit.Reactive GRASP for the strip-packing problem. Computers & Operations Research 35.4 (2008): 1065-1083.
36. Parreño, Francisco, et al. A hybrid GRASP/VND algorithm for two-and three-dimensional bin packing. Annals of Operations Research 179.1 (2010): 203-220.

37. Levine, John, and Frederick Ducatelle. Ant colony optimization and local search for bin packing and cutting stock problems. Journal of the Operational Research Society (2004): 705-716.
38. Silveira, Miguel Espinheira, Susana Margarida Vieira, and Joao Miguel Da Costa Sousa. An ACO algorithm for the 3D bin packing problem in the steel industry. Recent Trends in Applied Artificial Intelligence.Springer Berlin Heidelberg, 2013.535-544.
39. Liu, D. S., et al. On solving multiobjective bin packing problems using evolutionary particle swarm optimization. European Journal of Operational Research 190.2 (2008): 357-382.
40. Blum, Christian, VerenaSchmid, and Lukas Baumgartner. On Solving the Oriented Two-Dimensional Bin Packing Problem under Free Guillotine Cutting: Exploiting the Power of Probabilistic Solution Construction. arXiv preprint arXiv:1209.0880 (2012).
41. Wong, Lily. Heuristic Placement Routines For Two-Dimensional Rectangular Bin Packing Problems. Diss. Universiti Putra Malaysia, 2009.
42. Burke, Edmund, et al. "Hyper-heuristics: An emerging direction in modern search technology." International series in operations research and management science (2003): 457-474.
43. Burke, Edmund K., et al. "Automating the packing heuristic design process with genetic programming." Evolutionary Computation 20.1 (2012): 63-89.
44. Burke, Edmund K., et al. Hyper-heuristics: A survey of the state of the art, Journal of the Operational Research Society 64.12 (2013): 1695-1724.
45. López-Camacho, Eunice, et al. A unified hyper-heuristic framework for solving bin packing problems. Expert Systems with Applications 41.15 (2014): 6876-6889.
46. Thomas, Jaya, and Narendra S. Chaudhari. Design of efficient packing system using genetic algorithm based on hyper heuristic approach. Advances in Engineering Software 73 (2014): 45-52.
47. Gómez, Juan Carlos, and Hugo Terashima-Marín. Building general hyper-heuristics for multi-objective cutting stock problems. Computación y Sistemas 16.3 (2012): 321-334.
48. Sim, Kevin, and Emma Hart. Generating single and multiple cooperative heuristics for the one dimensional bin packing problem using a single node genetic programming island model. Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation.ACM, 2013.

49. Sim, Kevin, Emma Hart, and Ben Paechter. A lifelong learning hyper-heuristic method for bin packing. Evolutionary Computation 23.1 (2015): 37-67.

50. Beyaz, Muhammed, TanselDokeroglu, and AhmetCosar. Robust hyper-heuristic algorithms for the offline oriented/non-oriented 2D bin packing problems. Applied Soft Computing 36 (2015): 236-245.

Corresponding Author:

BoominathanPerumal*,

Email: *bominathan@yahoo.com*