Applied Soft Computing 36 (2015) 236–245

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# Robust hyper-heuristic algorithms for the offline oriented/non-oriented 2D bin packing problems

Muhammed Beyaz [a,*], Tansel Dokeroglu [b], Ahmet Cosar [b]

[a] SAP Development Center, Pendik, 34906 Istanbul, Turkey
[b] Computer Engineering, Middle East Technical University, 06800 Ankara, Turkey

ARTICLE INFO

*Article history:*
Received 2 April 2015
Received in revised form 3 June 2015
Accepted 22 June 2015
Available online 29 July 2015

*Keywords:*
2D bin packing
Hyper-heuristic
Evolutionary
Genetic
Memetic

ABSTRACT

The offline 2D bin packing problem (2DBPP) is an NP-hard combinatorial optimization problem in which objects with various width and length sizes are packed into minimized number of 2D bins. Various versions of this well-known industrial engineering problem can be faced frequently. Several heuristics have been proposed for the solution of 2DBPP but it has not been possible to find the exact solutions for large problem instances. Next fit, first fit, best fit, unified tabu search, genetic and memetic algorithms are some of the state-of-the-art methods successfully applied to this important problem. In this study, we propose a set of novel hyper-heuristic algorithms that select/combine the state-of-the-art heuristics and local search techniques for minimizing the number of 2D bins. The proposed algorithms introduce new crossover and mutation operators for the selection of the heuristics. Through the results of exhaustive experiments on a set of offline 2DBPP benchmark problem instances, we conclude that the proposed algorithms are robust with their ability to obtain high percentage of the optimal solutions.



## 1. Introduction

Given a set of rectangular items and 2D bins of fixed width and variable length, the 2D bin packing problem (2DBPP) consists of orthogonally placing all the pieces within the bin, without overlapping, such that the overall number of the bins is minimized [1–6]. The 2DBPP is an intractable combinatorial optimization problem and widely faced during the industrial manufacturing processes. For the Offline bin packing (OffBP) version of the problem, the objects to be inserted are known before they are packed so that they can be reordered according to some insertion heuristics. Plane load planning is a typical OffBP problem. On the other hand, with Online bin packing (OnBP) version of the problem objects arrive one by one and there is no way to know complete input sequence, so it must be inserted into a bin immediately without waiting other objects. Hard drive partitioning for online storage systems usually deals with these types of problems. Orientation (whether the objects can be rotated 90 degrees or not) is a key aspect of the 2DBPP. Rotating objects in packing creates better results but objects may not be rotatable in every problem definition. For example textile industry can change the orientation of single color shirts by rotating while the process is in cutting phase, because there is no difference between two processes but shipping industry must consider the orientation of fragile items.

Hyper-heuristic optimization methods aim for the process of selecting, combining, generating or adapting various simpler heuristics to solve computational search problems efficiently [7,8]. There may be several heuristics that the algorithm can choose for solving an NP-Hard problem. Given a problem, hyper-heuristic methods select which heuristic should be applied, depending on the search stage. In this study, we propose a set of novel hyper-heuristic solution operators (crossover and mutation) and computation methods inspired from Evolutionary Algorithms (EA) for the solution of the 2D OffBP problem. Reproduction, mutation, recombination and selection are the key mechanisms of EA that are used to solve NP-Hard optimization problems. BPP, Traveling Salesman and Quadratic Assignment Problem [5,9] are some of the well-known challenging NP-Hard problems modelled and solved with EAs. Genetic Algorithm (GA) and Memetic Algorithm (MA) are the most efficient approaches of EAs. GA mimics the natural evolution process and has the ability to find (near-) optimal solutions in a large search space within very short time periods. Survival of the fittest individual is a rule allowing the best solution in each iteration to converge to a (near-) optimal solution. In GA, the fittest of individuals is selected as the solution of optimization problem.

* Corresponding author. Tel.: +90 3124414475.
*E-mail addresses:* muhammed.beyaz@sap.com (M. Beyaz), tansel@cengmetu.edu.tr (T. Dokeroglu), cosar@ceng.metu.edu.tr (A. Cosar).

http://dx.doi.org/10.1016/j.asoc.2015.06.063
1568-4946/

Parents mate and produce offspring and the best individuals are selected to survive to the next generation. MA is another growing area of EA. It mimics natural evolution process but it may differ from GA by performing individual learning which is also known as meme(s).

In this study, we propose two different algorithms for solving the oriented and non-oriented versions of offline 2DBPP. The algorithms implement novel crossover and mutation operators and make use of well known heuristics. The first algorithm, Hyper-heuristic Algorithm for the Oriented 2D BPP (HHA-O), uses heuristics Finite Next Fit (FNF) [10], Finite First Fit (FFF) [10], Best Fit Decreasing Height (BFDH), Unified Tabu Search (UTS) [13] and oriented Improved Left Gap Fill (LGFof) [14] and optimizes the oriented 2DBPP. The second algorithm, Hyper-heuristic Algorithm for the Non-oriented 2D BPP (HHA-NO), uses heuristics FNF, FFF, BFDH, UTS, LGFof and Improved Left Gap Fill (LGFi) [14] in order to minimize the number of bins for the non-oriented 2D offline BPP. Through exhaustive experiments performed with 500 benchmark problem instances, we report the performance of our novel robust hyper-heuristic algorithms in comparison with the other state-of-the-art 2DBPP methods.

In Section 2, we briefly review the relevant and the best performing state-of-the-art algorithms for the solution of 2DBPP. Section 3 gives information about the formal definition of the 2DBPP. Section 4 defines the details of the novel crossover and mutation operators and the proposed algorithms. Experimental comparisons of the proposed algorithms are presented in Section 5. Finally, conclusions and further research directions are given in Section 6.

## 2. Related work

In this section, we give brief information about the well-known heuristics/metaheuristics Next Fit (NF), First Fit (FF), Best Fit (BF), Unified Tabu Search (UTS), Left Gap Fill (LGF), Genetic Algorithm (GA) and Memetic Algorithms (MA) which are used in our hyper-heuristic algorithms to solve the 2DBPP in a more efficient way. Next Fit Decreasing (NFD) [38], First Fit Decreasing (FFD), and Best Fit Decreasing (BFD) are level-oriented packing heuristic algorithms. NFD packs the largest square cube into the left most corner of the hyberbox, and picks the second largest cube. It checks the remaining part of ground level of the hypercube. If the remaining part is enough then it inserts the second largest square cube into the position which it can touch the first largest cube. If the second largest cube cannot fit then it creates a new level with height of first largest cube and tries to insert the second largest cube at that level. NFD Height (NFDH) [35] and Finite Next Fit (FNF) [10] are other versions of NFD. Items are packed into the first bin which has available space for item is the heuristic known as FF [45]. First Fit Decreasing Height (FFDH) applies non-increasing height sorting to FF Decreasing for 2DBPP [35]. Finite First Fit (FFF) algorithm packs the item into finite bin set in one phase [10]. BF algorithm is similar to FF algorithm but items are inserted by remaining space in the bin [45]. Best Fit Decreasing (BFD) algorithm is an extended version of BF. First, it sorts the items and then items are inserted. Finite Best Strip Packing (FBSP) is a two level algorithm [10]. At first level, items are inserted according to BF. At second level, prepared strip result is divided into levels and levels are merged into bins by using remaining horizontal space. Best Fit Decreasing Height (BFDH) sorts the items according to decreasing height and inserts them by using BFD algorithm [29].

Tabu Search (TS) is developed by Glover and consists of a stopping condition, keeping tabu list, neighborhood search, and aspiration condition [25–27]. Stopping condition checks the end-point of the search. Tabu list keeps the track of last known moves in order not to stuck into a local optima. Neighborhood search investigates if small changes (e.g. swapping two items) made on the current solution can produce any improvements. Aspiration condition or fitness function calculates the value of current search. Unified Tabu Search (UTS) algorithm is developed by Lodi [17,13].

LGF uses dynamic selection of the best fit strategy in order to choose the next rectangle [36]. It consists of two stages for processing of rectangles. First stage is preprocessing which rotates items whose height is greater than width and then sorts the item by decreasing width. At the second stage it inserts items by using left-most, tallest neighbor or shortest neighbor heuristics. Burke's strip packing strategy is enhanced and the same preprocessing strategy and packing best fit rectangle into bin by using waste space insertion is used in Best Fit Bin (BFB) [15]. Lowest Gap Fill (LGF) algorithm is an extension of BFB by using left most position search instead of waste space insertion [37]. LGF is improved into Improved Lowest Gap Fill (LGFi) by changing the shortest edge as remaining gap and also developed a new version of LGFi in order to pack oriented bin packing known as Oriented Improved Lowest Gap Fill (LGFof) [14].

GA is offered by Bremermann as a mathematical model for solving problems by using genes, mutation, and crossover [31,33]. Holland extended GA and developed a framework [32]. Goldberg introduced GA terminology and application areas [22]. Smith tried to find the maximum number of rectangles that can be packed into a bin by using a GA [20]. He used permutation of rectangles in order to encode the chromosomes. Hopper and Turton found a solution to the BPP of rotatable objects by use of BL and BLF heuristics [23]. He used order-based (permutation) encoding schema. Kroger et al. create a new vision of GA encoding in [19].

MA is also known as Hybrid Evolutionary Algorithms (HEA) that is a hybrid population-based and local search technique. Dawkins unfolded a new idea of gene-centered view of evolution with a new phrase *meme*. Moscato developed a term Memetic Algorithm (MA) [42,43]. MA uses two different learning methods such as Lamarckian Learning, Baldwinian Learning [21,51]. Development of MA can be divided into three generations. First generation describes the basis of MA. It introduces MA as marriage between a global search technique with local learning. Hyper-heuristic [30], Meta-Lamarckian MA [48] and Multi-meme [39] are classified as second generation of MA. In Hyper-heuristic and Meta-Lamarckian MA, a reward mechanism is applied to individuals. In Multi-Meme, memes are considered as a part of individual (genotype) so inheritance is used to passed memetic behavior of individuals to their offspring. Third generation of MA uses Co-evolution [34] and self-generation [40]. Rule-based local search is applied to heuristic so repeated patterns are captured and individuals are nourished by captured patterns. Recombination and mutation of population are proposed for global search and Ahuja and Murty (AM) tree improvement heuristic (AM-H) and the Random Sampling Ahuja and Murty (RSAM) algorithm are used as local search heuristics [50,41].

A study composed of three different algorithms, Artificial Immune System (AIS), GA and Particle Swarm Optimization (PSO) shows that AIS is better than GA and PSO [44]. A Greedy Randomized Adaptive Search Procedure (GRASP) is used in order to meet critera California Vehicle Code (CVC) in [16]. Goncalves and Resende presented a new way of 2DBP and 3DBP with Biased Random-Key GA (BRKGA) [28]. Lopez-Camacho et al. suggested a heuristic and hyper-heuristic method for packing problems [24]. Heuristics are FFD, Filler, BFD, Djang and Finch with initial fullness of DJD$\frac{1}{2}$, DJD$\frac{1}{3}$ and DJD$\frac{1}{2}$. Bennell et al. proposed a MultiCrossover GA (MXGA) with BFB heuristic for NO-2DBPP with Due Dates (DD) [12]. Fernandez et al. considered rotation of items and Load Balancing (LB) of bins as a multiobjective problem [4]. Blum and Schmid described an EA that is based on LGFi heuristic for free guillotine oriented two-dimensional bin packing problem (FG-O-2DBPP) [3]. Bansal and Khan used Round and Approximation Framework to pack oriented/non-oriented 2D rectangles [52].