A REVIEW: META-HEURISTIC APPROACHES FOR SOLVING RECTANGLE PACKING PROBLEM

1Dr. Leena Jain, 2GaganDeep Singh

1Associate Professor & Head-MCA, Global Institute of Mgt. &. Emerging Technologies, Amritsar (India)
2Ph. D., Research Scholar, Punjab Technical University Jalandhar (India), Head of Dep’t (Computer Science & Application), CKD Institute of Management & Technology, Amritsar (India)

ABSTRACT

Packing problems are optimization problem encountered in many areas of business and industries and have wide applications. These problems look for good arrangement of multiple items in some larger containing regions with an objective to maximize the utilization of resource materials. 2D packing problem has wide industrial applications starting from small scale industries related to leather, furniture, glass, metal, and wood to large scale industries dealing with textile, garments, paper, shipbuilding, automobiles and VLSI design. In this paper authors have summarized the different Metaheuristic approaches used to solve packing problem. Accordingly, this paper is a pure academic activity that catalogues latest research in the area of Rectangle packing Problem using Metaheuristic approaches.

Keywords: Ant colony optimization (ACO), genetic algorithm (GA), Swarm Particles Optimization (SPO), Variable Neighborhood Search (VNS)

I. INTRODUCTION

Rectangular Packing problems arise in many industries where desired rectangular items are laid on the rectangular object, which is bigger in size in comparison to items. This is of particular interest to industries involved with mass production as small improvements in design layouts can lead to considerable savings in raw material vis-à-vis reduction in production costs [1]. The wood, glass and paper industries are mainly concerned with the cutting of regular figures. The floor planning in VLSI (very large scale integrated design) is another tricky application of rectangle packing. On the other hand, in shipbuilding, textile and
leather industries irregular and arbitrary shaped items need to be packed. These items are first packed into rectangle enclosures and then rectangle enclosures are in turn packed onto rectangle stock sheet using rectangle packing approach [2]. Most of the standard problems related to Nesting are known to be NP-complete where computation time for an exact solution increases with N and become rapidly prohibitive in cost as N increases. The solution approach to these problems lies in reducing the exhaustive search of all possible arrangements of nesting the parts and subsequently checking upon the execution time. The development of exact algorithms, which are faster and produce near optimal solutions, is still a major research issue in this area. Proliferation of sophisticated desktops and faith of researchers in meta-heuristics have further allowed them to look beyond the traditional optimization techniques to solve this hard problem. Singh and Jain [3] undertook a study that catalogues solution methodologies popularly used to handle nesting problems. Fig. 1 represents survey of solution methodologies for Nesting Problem and it is clear from the figure now a days researchers work on Metaheuristic for solving the packing/nesting problem. This paper reviews the application of meta-heuristic methods to two dimensional rectangle packing problem.

Commonly used Metaheuristic methods in rectangle packing problem
- Simulated annealing[4,5]
- Tabu search [6,7,8,9,10,11]
- Iterated local search [12,13,14]
- Genetic Algorithm[15,16,17,18]
- Ant Colony Optimization. [19]
II. SYNOPSIS OF GENETIC ALGORITHM

Genetic Algorithms (GA) were developed by Holland (1975) and since then have been used in various fields of engineering. GA has been used quite successfully for combinatorial problems that are NP-complete. GAs have been used in a wide variety of optimisation tasks, including numerical optimisation and combinatorial optimisation problems such as travelling salesman problem (TSP) [20], packing problem [21,22,23], job shop scheduling[24] and video & sound quality optimisation. A genetic algorithm is a randomized parallel search method modelled on natural selection and genetics[25]. In contrast to more standard search algorithms, GAs base their progress on the performance of a population of candidate solutions, rather than that of a single candidate solution. The motivation behind this is that by simultaneously searching many areas of the design space the risk of getting stuck at local optima is greatly reduced. GAs are probabilistic in nature and start off with a population of randomly generated candidates and evolve towards better solutions by applying genetic operators, modelled on the natural genetic process. For solving any problem, Genetic Algorithm is started with a set of solutions (represented by chromosomes) called population. A member of the population is a genotype, a chromosome, a string or a permutation. Solutions from one population are taken and used to form a new population. This is motivated with a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are, the more chances they have to reproduce. When a genotype is decoded, a packing pattern, called a phenotype, is formed. We may calculate the fitness function even to restrict the mutating parents. Working of GA represents in fig. 2. A GA generally has five components [26]

1. A representation for solutions to the problem
2. A method to create initial population
3. An evaluation function to determine the relative fitness of the solutions
4. Genetic operators that effect the composition of the offspring during reproduction
5. Values for parameters that the GA uses (e.g. population size, probabilities of applying the genetic operators, etc.).

III. SYNOPSIS OF ANT COLONY OPTIMIZATION

This optimization, first introduced by Colorni et al. [27,28] imitates the way ants search for food and find their way back to their nest. First an ant explores its neighborhood randomly. As soon as a source of food is found it starts to transport food to the nest leaving traces of pheromone on the ground which will guide other ants to the source. The intensity of the pheromone traces depends on the quantity and quality of the food available at the source as well as from the distance between source and nest, as for a short distance more ants will travel on the same trail in a given time interval. As the ants preferably travel along important trails their behavior is able to optimize their work. Pheromone trails evaporate and once a source of food is exhausted the trails will disappear and the ants will start to search for other sources. For the heuristic, the search area of the ant corresponds to a discrete set from which the elements forming the solutions are selected, the amount of food is associated with an objective function and the pheromone trail is modeled with an adaptive memory [29].
Aco Characteristics
- Exploit a positive feedback mechanism
- Demonstrate a distributed computational architecture
- Exploit a global data structure that changes dynamically as each ant transverses the route
- Has an element of distributed computation to it involving the population of ants
- Involves probabilistic transitions among states or rather between nodes

**Figure 2: Flowchart of working of GA**

### IV. SYNOPSIS OF SWARM PARTICLES OPTIMIZATION

**Particle swarm optimization** (PSO) is a population-based stochastic approach for solving continuous and discrete optimization problems. In particle swarm optimization, simple software agents, called particles, move in the search space of an optimization problem. The position of a particle represents a candidate solution to the optimization problem at hand. Each particle searches for better positions in the search space by changing its velocity according to rules originally inspired by behavioral models of bird flocking. Particle swarm
optimization belongs to the class of swarm intelligence techniques that are used to solve optimization problems. It was developed in 1995 by James Kennedy (social-psychologist) and Russell Eberhart (electrical engineer). It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution. Each particle is treated as a point in a N-dimensional space which adjusts its “flying” according to its own flying experience as well as the flying experience of other particles. Each particle keeps track of its coordinates in the solution space which are associated with the best solution (fitness) that has achieved so far by that particle. This value is called personal best, \( pbest \). Another best value that is tracked by the PSO is the best value obtained so far by any particle in the neighborhood of that particle. This value is called \( gbest \). The basic concept of PSO lies in accelerating each particle toward its pbest and the gbest locations, with a random weighted acceleration at each time step as shown in Fig 2. Fig 3 represents working of PSO algorithm[30].

![Figure 2: Searching point by PSO](image)

**PSO Characteristics**
- Population-based optimization technique – originally designed for solving real-valued function optimizations
- Applicable for optimizations in rough, discontinuous and multimodal surfaces
- Does not require any gradient information of the function to be optimized
- Conceptually very simple
- Each candidate solution of continuous optimization problem is described (encoded) by a real vector N-dimensional search space: \( x = x_1, \ldots, x_n \)
- Each candidate solution is called PARTICLE and represents one individual of a Population called SWARM.
- The particles change their components and FLY through the multi-dimensional search space.
- Particles calculate their FITNESS function as the quality of their actual position in the search space using w.r.t. the function to be optimized.
- Particles also compare themselves to their neighbors and imitate the best of that neighbors.
V. SYNOPSIS OF ITERATED LOCAL SEARCH

Iterated local search [Stiitzle (1999)] is a general meta-heuristic [31]. It has two basic operators for generating new solutions. One is a local search and other is a perturbation operator. When the local search is trapped in local optimal solution, a perturbation operator is applied to the local optima to generate a new starting point for its local search. It is desirable that the generated starting point should be in a promising area in the search space. A commonly-used perturbation operator is a conventional mutation, which can produce a starting point in a neighboring area of the local optimum. Another perturbation operator is guided mutation operator [32,33]. Let $C$ be the cost function of our combinational problem, $c_i$ to be minimizes. We label candidate solution by $S$, solution in $S$ by $s$, cost function value of solution by $C(s)$, Locally optimal solution by $s^*$, set of locally optimal solutions by $S^*$ and Local Search defines mapping from $S \rightarrow S^*$. Figure 4 depict procedure of Iterated Local Search [31.]

```
procedure Iterated Local Search
S_0=Generate InitialSolution
S^*=LocalSearch(S_0)
Repeat
S = Perturbation(S^*,history)
S^*=LocalSearch(S')
S^*=AcceptanceCriterion(S, S^*, History)
until termination condition met
end
```

Figure 4: procedure of Iterated Local Search
VI. A BRIEF REVIEW OF LITERATURE TO RECTANGULAR PACKING PROBLEM USING METAHEURISTIC APPROACH

The majority of literature available is on hybrid algorithms, where a GA is combined with a heuristic placement routine. In this two-stage approach a GA is used to determine the sequence, in which the items are to be packed. A second algorithm is then needed, which describe how this sequence is allocated onto the object. One of the first few researchers who implemented GAs in the domain of packing was Smith (1985) [34]. He experimented with two heuristic packing routines, one of which implements backtracking and produces denser layouts. However, this method was computationally more expensive. Comparisons between the two hybrid approaches showed that the combination with the more sophisticated heuristic generated better packing patterns. The packing problem Smith studied was special in that the orientation of the rectangles was fixed.

The GA was based on a directed binary tree to encode the problem [35, 36]. This representation fixes one dimension of the position of an item in the layout. The second dimension was determined by the bottom left condition. Since its performance was compared to well-known packing heuristics, a relative comparison with our work is possible.

Falkenauer and Delchambre (1992) [37] have developed a genetic algorithm for the classical two-dimensional bin packing problem, where the objective was to minimise the number of bins. Consequently, the fitness function has to take into account how efficiently the bin capacity is utilised. The encoding technique uses one gene per item to represent the bin in which it is packed. This method of encoding does not perform well in combination with the classic crossover and mutation operators. Therefore, an alternative data structure was proposed with a chromosome consisting of two parts, an object part and a group part. The object part identifies the items which form a bin and the group part contains one gene per bin introduced in the object part.

In the hybrid approach by Hwang et. al. (1994) [38] a GA was combined with a well-known heuristic from bin packing, the so-called First-Fit-Decreasing-Height algorithm (FFDH). Although this technique produces guillotineable layouts, and is suggested for the general case of the rectangle packing problem. It also used a directed binary tree, which combines two rectangles to a larger rectangle by either placing them horizontally or vertically next to each other. The position within the larger rectangle was left justified. As mentioned before, comparisons with a hybrid GA technique showed that this method was less efficient in terms of packing height.

Herbert and Dowsland (1996) [39] developed a 2D coding technique for a pallet-loading problem of identical rectangles. The layout was represented by 2D matrix indicating available positions for vertical and horizontal placement, where in the horizontal one has priority over the vertical one. This technique works well for small problems. In order to improve the outcome for medium sized problems additional repair and enhance operators have been introduced by the authors.

Jakobs (1996) [40] first suggested genetic algorithms to two dimensional packing problem and employed the bottom left (BL) strategy to pack the small rectangles.

The method developed by Ratanapan and Dagli (1997) [41] is different from the other approaches described so far, since it does not make use of a data structure to represent the problem. In their approach, the items were represented as 2D pieces with their true geometric dimensions. After the initialisation process, which places all items onto non-overlapping positions on the object, a series of genetic operators which consist of moving, relocation and recombination operations were applied.
Dagli and Poshyanonda (1997) [42] also used the GA to generate an input sequence for the placement algorithm, which was based on a sliding method combined with an artificial neural network. In this method, every incoming item was placed next to the partial layout and all scrap areas generated were recorded. If there was a match between an incoming item and one of the scrap areas, the neural network selects the best match. A second category of solution approach with GAs aims at incorporating some of the layout information into the data structure of the GA. However, some additional rules are still needed to fix the position in the layout.

Leung et. al. (2003) [15] found that genetic algorithms consistently outperform simulated annealing, and that a genetic algorithm combined with simulated annealing outperforms a pure genetic algorithm. However, a flaw in the study of Leung et al. was the assumption that the orientations of the small rectangles are fixed.

Goncalves (2007) [43] proposed an algorithm, which hybridizes a placement procedure with a genetic algorithm based on random keys for two-dimensional (2D) orthogonal packing problem. The approach was tested on a set of instances taken from the literature and was compared with other approaches. The computation results validate the quality of the solutions and the effectiveness of the proposed algorithm.

Ranjan et. al. (2009) [16] explored the basics of genetic programming (emerging evolutionary technique) with the possibilities of its application in interdisciplinary research. This paper illustrated the difference between genetic programming and genetic algorithm. It stated that genetic programming created computer program in the LISP computer language as the solution and the output of genetic programming is another computer program, which could suitably be applied in applications, whereas the genetic algorithm created a string of numbers that represents the solution. Therefore, it was concluded that genetic programming was more powerful than genetic algorithm.

Lai and Chan (1997) [44] developed an algorithm based on evolution strategy. This paper provides a basis for exploring the integration of the evolutionary algorithm technique in artificial intelligence with classic two dimensional cutting stock problems, with minimum trim loss. The algorithm developed could be used to solve non-guillotine, two dimensional cutting problems. The algorithm addresses the problem of placing differently sized small rectangles on a larger rectangle or box, in order to minimize trim loss. The advantage of the proposed heuristic was its easy implementation and lesser requirement of computation memory.

Jain and Gęa (1998) [45] presented a technique for applying the genetic algorithm for the two dimensional packing problem. This algorithm was quite effective and could be used for any level of discretization. The algorithm worked for both concave and complex objects with holes. In this work, random seeds were tried on the same problems and a new concept of a two dimensional genetic chromosome was introduced. The total layout space is divided into a finite number of cells for mapping it into a 2D genetic algorithm chromosome. The mutation and crossover operators have been modified and are applied in conjunction with connectivity analysis for the objects to reduce the creation of faulty generation. A new feature was added to the genetic algorithm in the form of a new operator called compaction. Several examples of GA based layout were presented. Thus it was concluded that the problem of compacting general shaped objects could be resolved quite accurately and in a very small period. A limitation of this approach was that it allows rotations in steps of 90 degree.
Hopper and Turton (1999) [46] described two GAs for rectangular packing problems. Both the GAs were hybridized with a heuristic placement algorithm, one of which is the well-known bottom left routine (BLR). A second placement method which overcame some of the disadvantages of the BLR was developed. The two hybrid genetic algorithms were compared with heuristic placement algorithms. The GA combined with the improved BLF heuristic outperformed the GA, using both the BL method as well as the heuristic method. The improved heuristic was sufficient to achieve high quality layouts for application in industry.

Faina (1999) [47] proposed two new algorithms global optimization and simulated annealing, involving guillotine and non-guillotine constraints, to solve the two dimensional rectangular cutting stock problem. The simulated annealing is an algorithm that continuously attends to transform the current configuration into one of its neighbours. The mechanism is mathematically best described by means of a Markov chain: a sequence of trials, where the outcome of each trial depends only on the outcome of the previous one. Several tests proved the validity of these algorithms. It was derived that, for a large number of items, the NON-GA is beyond comparison with the GA, since it gets much better cutting patterns in almost the same computational time. The result of the study illustrates the additional effort necessary for realizing computer codes based on the non-guillotine constraints.

Burke and Kendall (1999)[48] in their research work considered a simplified version of stock cutting problem. They found that all three metaheuristic techniques genetic algorithm (GA), tabu search (TS) and simulated annealing (SA)) were able to find the optimum solution to a small problem. When presented with larger problems, tabu search and simulated annealing produced good quality solution, whereas genetic algorithm produced lower quality solution despite many test during the small problem to try and ascertain the best combination of parameters. It was concluded that SA and TS algorithms produced similar solutions and both outperformed GA.

Hopper and Turton (2001a) [49] considered 2D rectangular packing problem, where a fixed set of items have to be allocated on a single object. Two heuristics belonging to the class of packing procedure that preserve bottom left (BL) stability, are hybridized with three meta heuristic algorithms (genetic algorithm, simulated annealing and naive evolution and local search heuristic). This study compares the hybrid algorithms in terms of solution quality and computation time, on a number of packing problems of different size. In order to show the effectiveness of the design of different algorithms, their performance is compared to random search (RS) and heuristic packing routines.

In their next study Hopper and Turton (2001b) [50] compared the performances of genetic algorithms and simulated annealing and found that the latter yields better results than the former but it requires more execution time when the size of a problem is large.

Babu and Babu (2001) [51] proposed both genetic and heuristic algorithms to arrange multiple 2-D shaped parts in multiple 2-D shaped sheets with the objective of minimizing the wastage of the sheet material. The paper proposed a new method of representing the sheet and part of geometries in a discrete form to arrange the parts on the sheet quickly, irrespective of the complexity in the geometry of the sheets and parts. The proposed heuristic algorithm, using the bottom-left positioning strategy, arranges the parts with their orientation, on the sheets, in an effective manner. The genetic algorithm that generates the best sequence of the sheets and parts with their orientation can be obtained for nesting the parts in an optimal manner. Authors considered the geometry of parts and sheet with
line and arc features. The discrete representation scheme adopted in this approach can easily incorporate free-form features for the parts and sheets. It concluded that the computational complexity of the proposed approach can be reduced with the help of parallel processing algorithms.

In 2003, Onwubolu and Mutingi [52] proposed a genetic approach for the guillotine rectangle packing problem. The approach firstly places the initial rectangle and then searches for a rectangle that can be attached to the first whilst maintaining the guillotine cut constraint. The chromosome within the genetic algorithm represents the piece pairs that are to be attached and the orientation in which they should be placed.

Lin (2006) [53] in his study proposed a genetic algorithm to solve the two dimensional assortment problems. The proposed genetic algorithm uses a problem-specific encoding scheme that incorporates a novel packing process. Numerical examples indicated that the coding method is effective for solving the assortment problem. Using this coding scheme the proposed genetic algorithm appeared to be more efficient and effective than a state of the art integer programming approach and provided acceptable solutions to large problems.

Huang et al. (2007) [54] recommended a new heuristic algorithm based on two important concepts namely, the corner — occupying action and caving degree. Authors used the following rectangle packing principle: the rectangle to be packed into the box always occupies a corner, and the caving degree of the packing action should be as large as possible. 21 rectangle packing instances were tested by the algorithm developed and 16 of them had achieved optimum solution within the reasonable runtime. Experimental results demonstrated that the algorithm developed was fairly efficient for solving the rectangle packing problem.

Goncalves (2007) [43] in his paper addressed a hybrid meta-heuristic algorithm for two dimensional (2D) orthogonal packing problem, where a fixed set of small rectangles had to be placed on a larger stock rectangle in such a way that the amount of trim loss is minimized. This algorithm combines random keys based genetic algorithm with a novel fitness function and new heuristic placement policy. The experimental result demonstrated the excellent quality of the proposed algorithm when compared with other meta-heuristics in absolute terms.

Alvarez-Valdes et al. (2007)[55] have developed a new heuristic algorithm based on Tabu search techniques for a non-guillotine two-dimensional cutting stock problem. Two moves have been proposed, based on the reduction and insertion of blocks of pieces. The efficiency of the moves is based on a merge and fill strategy that accommodates the empty rectangles to the pieces still to be cut. Some intensification and diversification strategies based on long-term memory, have also been included in this study.

Hadjiconstantinou and Iori (2007) [56] in their paper presented a new greedy algorithm and hybrid genetic algorithm with elitist theory, immigration rate, heuristics online and tailored crossover operators, for two dimensional knapsack problems. The algorithm was evaluated on a very large number of test problems of varying size and complexity. The results indicated that as the problem size increased the heuristics performed better and the results outperformed existing metaheuristics.

Mohamed and Adnan (2009) [19] proposed new pretreatments for the two-dimensional bin-packing problem, the non-oriented case. The problem of bin-packing in two dimensions (2BP) consists of placing a given set of rectangular items in a minimum number of rectangular and identical containers, called bins. This study treats the case of objects
with a free orientation of 90 degree. The authors proposed an approach of resolution combining optimization by colony of ants (ACO) and the heuristic method IMA to resolve this NP-Hard problem. New heuristic method (SACO) was developed for the resolution of the problem. Pretreatments on benchmarks taken from literature were tested and their efficiency to simplify the instances was shown. It was concluded that on an average better results were shown on average by using the proposed method.

Lueng et. al. (2010) [57] presented a fitness strategy based on which a constructive heuristic algorithm was developed. The fitness strategy evaluated the fitness of every unpacked rectangle so that one appropriate rectangle is selected to pack into the sheet. Based on this fitness strategy, a constructive heuristic algorithm was developed to generate a solution, i.e., a given sequence of rectangles for packing. Then, a greedy strategy was used to search for a better solution. By combining the greedy strategy and simulated annealing algorithm, a hybrid heuristic algorithm was proposed to solve the rectangular knapsack packing problem. The computational results on zero-waste and non-zero-waste instances have shown that HSA is very fast and effective in solving the rectangular knapsack packing problem.

Ducatelle and Levine (2002) [58] adopted ACO to decipher bin packing and employed simulated ants to assemble solutions stochastically, with heuristic information obtained by First Fit Decreasing (FFD) and an simulated pheromone trail, which is determined by the favourability of having two items in the similar bin. This approach may get sound performance and a high-quality explanation even not finding the optimum. However, it’s computing time is extensive than others. As a result, Levine and Ducatelle (2004) [60] perk up the performance by combining ACO and an iterated local search approach, since a local search algorithm can significantly advance the performance of an ACO approach. The mixed approach they projected to the rectangle packing is relatively generic, and could be accomplished of being tailored to solve similar grouping problems.

Thiruvady [59] engaged a cross approach by combining ACO and Bottom-LeftFill (BLF) to solve two dimensional rectangular packing problems. The planning and orientations of items are produced by ACO and the items are owed one by one with BLF heuristic according to the ordering and orientations given by ACO. They evaluate and analyzed four combinations between ordering and orientation. The results showed that packing items with learning ordering and orientations obtained by ACO outperformed the other combinations.

Xu, Y.C. et.al (2010) [60] recommended a productive heuristic to pack weighted items in a rounded container. They utilize an ant-based algorithm and optimize the wadding order with the base of this heuristic. In their ant algorithm, encode of pheromone matrix considers the favourability of choosing an item and the product of the size and the weight of the next packed item. By doing so, hefty and weighty items have higher priority and this is advantageous for the packing performance. They also evaluated two edition of the ant-based algorithm, AS and Min-Max AS, with existing approaches, such as the genetic algorithm, and the hybrid PSO. However, since their study only centre on the regular objects (square and rectangle), it’s a problem for them whether their methodology can obtain fine performance for irregular formed objects.

Qi yang and wang jin-min (2011)[61] solve the rectangle packing problem using PSO. The algorithm optimizes the parameter of dynamic attractive factors by updating the position and the velocity of the particles, and applies perturbation strategy to solve the matter that it is easy to stick at local optima. The experimental result shows that the algorithm can get a better packing result by less time.
CONCLUSION

Rectangular Packing problems arise in many industries where desired rectangular items are laid on the rectangular object, which is bigger in size in comparison to items. Most of the standard problems related to Nesting/Rectangle Packing Problem are known to be NP-complete where computation time for an exact solution increases with N and become rapidly prohibitive in cost as N increases. The solution approach to these problems lies in reducing the exhaustive search of all possible arrangements of nesting the parts and subsequently checking upon the execution time. The development of exact algorithms, which are faster and produce near optimal solutions, is still a major research issue in this area. Proliferation of sophisticated desktops and faith of researchers in meta-heuristics have further allowed them to look beyond the traditional optimization techniques to solve this hard problem. The paper is a pure academic activity that catalogues latest research in the area of Rectangle packing Problem using Metaheuristic approaches.

REFERENCES


57. Ducatelle, F., Levine, J., “Ant Colony Optimization for Bin Packing and Cutting Stock Problems.” In 2001 proc. UK Workshop on Computational Intelligence, Edinburgh


59. Thiruvady, D.R., Meyer, B., Ernst, ” Strip packing with hybrid ACO Placement order is learnable.”’ In 2008 proc. IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1207–1213


