IOP Conf. Series: Materials Science and Engineering **183** (2017) 012026 doi:10.1088/1757-899X/183/1/012026

Optimization of heterogeneous Bin packing using adaptive genetic algorithm

R Sridhar[1], M Chandrasekaran[2], P Sriramya[3]and Tom Page[4]

[1]Assistant Professor and Research Scholar, Department of Mechanical Engineering,Vels University, Chennai, TN, India.
[2]Professor, Department of Mechanical Engineering,Vels University, Chennai, TN, India.
[3]Associate Professor, Department of Computer Science and Engineering,Saveetha Engineering College, Chennai, TN, India.
[4]Lough borough University, U.K.

E-mail: srisampangy@gmail.com

Abstract. This research is concentrates on a very interesting work, the bin packing using hybrid genetic approach. The optimal and feasible packing of goods for transportation and distribution to various locations by satisfying the practical constraints are the key points in this project work. As the number of boxes for packing can not be predicted in advance and the boxes may not be of same category always. It also involves many practical constraints that are why the optimal packing makes much importance to the industries. This work presents a combinational of heuristic Genetic Algorithm (HGA) for solving Three Dimensional (3D) Single container arbitrary sized rectangular prismatic bin packing optimization problem by considering most of the practical constraints facing in logistic industries. This goal was achieved in this research by optimizing the empty volume inside the container using genetic approach. Feasible packing pattern was achieved by satisfying various practical constraints like box orientation, stack priority, container stability, weight constraint, overlapping constraint, shipment placement constraint. 3D bin packing problem consists of 'n' number of boxes being to be packed in to a container of standard dimension in such a way to maximize the volume utilization and in-turn profit. Furthermore, Boxes to be packed may be of arbitrary sizes. The user input data are the number of bins, its size, shape, weight, and constraints if any along with standard container dimension. This user input were stored in the database and encoded to string (chromosomes) format which were normally acceptable by GA. GA operators were allowed to act over these encoded strings for finding the best solution.

1. Introduction

Due to industrial revolution, today most of the industries are focusing towards globalization. Globalization paved way for their customers to purchase the goods as per their requirements all around the world under a single roof. In turn, industries have substantially increases the scope and magnitude of their global production and distribution networks all around the world. So a necessity arises for those firms to deliver and distribute their goods to their customers, warehouses, and to the distribution

IOP Conf. Series: Materials Science and Engineering **183** (2017) 012026 doi:10.1088/1757-899X/183/1/012026

centers all around the world in time. Thus the globalized market results in rapid development of international trade and creates rigorous competitions among the industries. Recently, Economical research proved that the survival of a firm in this competitive globalized environment will be possible only if those firms reduce their product cost without compromising the quality. So a necessity arises for those multinational firms to reduce their product cost by optimizing various factors which influencing the cost. The major factors which influences the product cost are generally categorized as Material cost, Manufacturing cost, Inspection cost, packing cost and Distribution cost.

In the logistic industries, determining which box should be packed on which container by which sequence can mean the difference between significant profit and debilitating loss. So this research work concentrates on perfect packing of goods into the container by considering major practical constraints using evolutionary algorithm which is essential for a firm to survive in today's the competitive globalized environment.

2. Literatures on Bin Packing

When packing problems were first proposed, simple models were used with exact methods for solving the problem. Given a problem, the exact methods find the best solution every time they are run. However, as constraints were added, the difficulty of solving the problem increased, and simply finding a good feasible solution. In addition, many methods take too long when applied to problems of significant size. Many methods focused exclusively on packing and assumed that the plan was static with specific assumptions. Later methods attempted to integrate more than one algorithm in an effort to find optimization possibilities that take advantage of the inherent best properties from those algorithms. Some methods focus on precedence or temporal constraints. Stochastic algorithms may find a different solution each time they are run. Review of the relevant literature confirms that the evolutionary approach works very well when compared to the traditional optimization procedures and computational time also reduced significantly [1]. Genetic algorithm is the best among the evolutionary algorithm. Interestingly majority of the Genetic approaches are hybrid / heuristic, i.e. combining GA with other algorithm or by modifying GA [2].

3. Existing solution

Artificial Intelligence (AI) is a very great innovation in the field of Engineering now-a-days. This algorithm uses human intelligence like techniques in most cases to identify the solution for a new set of input pattern. Recent research proved that the AI techniques is a most reliable and important part in this transportation domain. Most commonly used Artificial Intelligence techniques are [3]

- Artificial Neural Network
- Case Based Reasoning
- Fuzzy Logic
- Expert Systems

3.1 Artificial Neural Network

An Artificial Neural Network (ANN) is an information processing system that is inspired by the way of biological nervous systems, such as the brain and process information. The key element of this structure is the information processing system which is formed with a large number of highly interconnected processing elements called neurons. Even though it resembles to human brain, it requires some training set of data to train the neural network. But in the packing problems it may not be possible to define the training set. As the type, dimensions of the boxes, constraints and conditions vary from time to time. So it may not be feasible to use these techniques in the bin packing applications.

3.2 Case Based Reasoning

Case based reasoning is a technique of Artificial Intelligence that attempts to solve new problems by using past experience. The main task involves is matching the given problem details against records of previous cases. This technique is effective and less time consuming, but the box varieties were limited and the input can be predicated. So it may not be suitable for wide varieties of box types with varying nature.

3.3 *Fuzzy Logic*

Fuzzy Logic is a mathematical technique for imprecise data and problems that have many solutions rather than one. Although it is implemented in digital computers which ultimately make only yes-no decisions, fuzzy logic works with ranges of values, solving problems in a way that more resembles to human logic. Fuzzy logic is used for solving problems with expert systems and real-time systems that must react to an imperfect environment of highly variable, volatile or unpredictable conditions. Some of the researchers used this technique for the bin packing applications.

3.4 Expert Systems

Expert Systems are Artificial intelligence based system that converts the knowledge of an expert in a specific subject into a software code. These techniques may not be suitable for the bin packing applications, as it lacking to solve the new constraint facing in the real time packing.

4. Evolutionary algorithms

An evolutionary algorithm uses some mechanisms inspired by biological evolution, reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the environment within which the solutions "live". Evolution of the population then takes place after the repeated application of the above operators. Evolutionary algorithms often perform well approximating solutions to all types of problems because they ideally do not make any assumption about the underlying fitness landscape [4].
These families of approaches include, but are not limited to,

- Genetic Algorithm
- Ants Colony Algorithm
- Tabu Search
- Simulated Annealing
- Quantum Annealing
- Stochastic Tunneling
- Reactive Search Optimization
- Stochastic Gradient Descent Algorithm
- Cross Entropy Method
- Particle Swarm Optimization
- Intelligent Water Drops
- Bee’s Algorithm

4.1 Genetic algorithm

A genetic algorithm is a search technique used in engineering science to find approximate solutions to optimization and search problems. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, natural selection, and recombination. This was found that the genetic approach will provide best suitable solution for the multi objective and multi constraint problems.

4.2 Ants Colony Algorithm

In operations research, the ant colony optimization algorithm(ACO) is aiming to search for an optimal path based on the behavior of ants seeking a path between their colony and a source of food. If other ants find such a path, they are likely not to keep travelling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food. They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically. It is clear that the ACO is used to identify the shortest path, more over it may not produce the better solution for the bin packing problems and it can be best suited for container transportation and distribution problems.

4.3 Tabu Search

Tabu search is a mathematical optimization method, belonging to the class of trajectory based techniques. Tabu search enhances the performance of a local search method by using memory structures that describe the visited solutions, once a potential solution has been detected, it should be marked as "taboo". Thus the algorithm will not be allowed to visit those possibilities repeatedly, thereby reducing the search space and converging towards the optimal solution. As the algorithm climbs over the better solution to reach the peak, it may not be suitable for bin packing optimization problems.

4.4 Simulated Annealing

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. After making many trades, lowering the temperature several times to a low value, one may then "quench" the process by accepting only "good" trades in order to find the local minimum of the cost function.
Simulated annealing may be more efficient to find an acceptably better solution in the predefined time instead of producing best solution with large set of parameters. So it may not be used for bin packing applications, as the application required only the best solution.

4.5 Quantum Annealing

It uses "quantum fluctuations" instead of thermal fluctuations to get through high, but thin barriers in the target function. So this methodology also may not be suitable for the bin packing problem.

4.6 Stochastic Tunneling

This method attempts to overcome the increasing difficulty of the simulated annealing runs have in escaping from local minima as the temperature decreases, by 'tunneling' through barriers. As this methodology works similar to simulated annealing, it may not be used for bin packing.

4.7 Reactive Search Optimization

This algorithm focuses on combining machine learning with optimization, by adding an internal feedback loop to self-tune the free parameters of an algorithm to suit with the characteristics of the problem. As the bin packing problem taken in this research is not a dynamic problem, so it doesn't have any feedback parameters to optimize.

4.8 Stochastic Gradient Descent Algorithm
It runs many greedy searches from random initial locations and achieves the good solution. But in bin packing the input are well known.

4.9 (i) Cross Entropy Method
This method generates solutions via a parameterized probability distribution. The parameters are updated via cross-entropy minimization function, so as to generate better samples in the next iteration. This methods consumes more times as the parameter increases and not suitable for large sized problems.

4.10 Particle Swarm Optimization
Particle swarm optimization (PSO) is an exciting new methodology in evolutionary computation that is somewhat similar to a genetic algorithm in that the system is initialized with a population of random solutions. Unlike other algorithms, however, each potential solution (called a particle) is also assigned a randomized velocity and then flown through the problem hyperspace. Particle swarm optimization has been found to be extremely effective in solving a wide range of engineering problems.

4.11 Intelligent Water Drops (IWD)
IWD which mimics the behavior of natural water drops to solve optimization problems.

4.12 Bee's Algorithm
Bee's algorithm is based on the foraging behavior of honey bees. It has been applied in many applications such as routing and scheduling. These algorithms may provide better solution for particular category of the problems and can not be used universally to all types of problems.

5. Heuristic and Hybrid Methods
Heuristics make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, Heuristics do not guarantee an optimal solution is ever found. As the bin packing problems having more number of parameters and the constraints, the algorithm used should be modified accordingly to produce optimal solution [5].
Generally, the parameters of the bin packing problems and its constraints can not be feed directly in to the intelligent evolutionary algorithms. These data should be encoded to a format acceptable by the evolutionary algorithms. Similarly, the output from the algorithm needs to be decoded to user understandable format. For encoding and decoding the data, separate algorithms required and also the procedure vary from application to application. Some time a need arises to synchronize the application with the algorithm. So a separate algorithm can be developed and integrated with the intelligent evolutionary algorithms to create a hybrid algorithm to solve an application.

6. Bin Packing
Bin-packing has been defined in several different forms such as stock cutting, vehicle loading, air cargo loading, container loading, scheduling, knapsack problem, etc depending on the application. In

this research work, bin packing problem was taken for consideration. The various terminologies and the standards used in this research work will be explained in this section

Bin packing is a finite collection of items with varying specifications to be packed into one or more containers or baskets [6]. Each container can hold any subset of the collection of objects without exceeding its capacity. Bin packing is also called as container loading, box packing, cargo loading, knapsack, etc. It not only denotes the packing of boxes in to the container but also arranging of good and items into the bags, baskets, pallets. The figure shows the sample set of boxes packed into the container.

7. Bin packing problem

The common problems faced by the bin packers are to pack the boxes in to the container in such a way that to utilize maximum volume of the container and the constraints satisfaction. This problem may be thought of as either a decision making problem or an optimization problem. In the decision making problems, a determination must be made, whether all of the objects fit into the container? In other words, the decision problem states that enough free space exists to hold the objects or not and having a Boolean answer. Alternatively, the optimization problem attempts to minimize the number of containers or minimize the amount of wasted space inside the container. This way of framing the problem attempts to optimize the containers utilization and having 'n' number of numeric answers, so mathematicians and computer scientists consider the bin packing problem as an optimization problem. General problems faced in bin packing are [7]

- Load the container with least empty space.
- Packed boxes should not exceed the container boundary.
- Total weight of the packed boxes should be less the pay load capacity of the container.
- Boxes to be unloaded first should be placed in the front portions of the container and vice-versa.
- Boxes of heavier weight should be kept at the bottom layers of the containers and vice-versa.
- Fragile items and items with constraints should be kept at the top layers of the container.
- Changing in the orientation of the boxes may leads to variation in the packed volume.
- For some of the items like bottles, liquid containers should allow to change the orientation.
- Boxes to be packed should not overlap with each other and with the container.
- Output from the software should be user understandable format.
- Base of a box to be packed should be completely lying on the face of bottom box.
- Surface profile of upper box to be packed should be the same as the lower box surface profile.

8. Types of bin packing

Form the above section it become very clear that the optimization of the bin packing problems need to satisfy many constraints are shown in Figure 1.

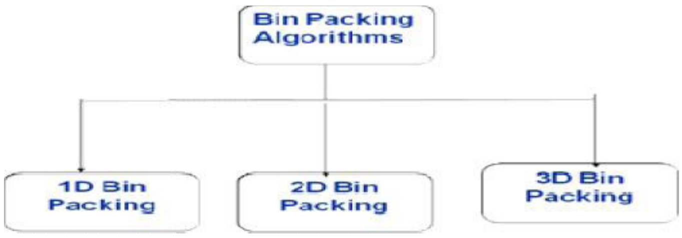

Figure 1. Types of Pin Baking Problem

So the researcher decomposes the problem into simple and builds to the complex by adding the parameter step by step. Generally, Bin packing problem are classified based on the dimension as

9. 1D Bin Packing

One dimensional bin packing problem optimizes only one parameter at a time. For example, writing maximum number of songs on the disc of standard memory size. In this example, by selecting the suitable song length, memory can be utilized properly. For bin packing problems, each object had fixed width and variable height or variable width and fixed height [8]. 1D problem will optimize either the length or the width of the box in a direction. Commonly used 1D Bin Packing Algorithms are:

- First Fit Algorithm (FFA).
- First Fit Decreasing Algorithm (FFDA).
- Refined First Fit Algorithm (RFFA).
- Best Fit Algorithm (BFA).
- Best Fit Decreasing Algorithm (BFDA) and many more.

10.2D Bin-Packing

Two dimensional bin packing problem is expanding the one dimensional bin packing problem by adding a variable to optimize. In the two dimensional version, both parameters will be vary to produce different solutions and need to optimize both the parameters simultaneously. The two dimensional bin-packing problem may be thought of as placing rectangles on a flat surface or inside a rectangle. Rectangle packing equates to efficiently fitting varying sized flat rectangles into a larger flat rectangular space most efficiently [9]. The classic video game 'Tetris' relates to the problem of rectangle bin-packing. However, unlike Tetris, classic two dimensional packing requires each object to orthogonal, so objects may or may not be rotated. Commonly used 2D Bin Packing Algorithms are as follows.

- Slide Pack Algorithm
- Skyline Pack Algorithm
- Nesting Algorithm
- Branch and Bound Algorithm.
- Knapsack Algorithm.

11.3D Bin Packing

The three-dimensional bin-packing problem retains the difficulty of lesser dimensional bin-packing problems, but holds unique and important applications. As one would expect, each object and bin exists in three dimensions. These objects and box represents triplets containing three values: width, length and height. Each box should fit into the container most efficiently [10]. If two dimensional bin-packing equates to rectangle-to-floor plan packing, three dimensional bin-packing equates to box-to-room packing. The 3D bin packing problems are classified as follows

IOP Conf. Series: Materials Science and Engineering **183** (2017) 012026 doi:10.1088/1757-899X/183/1/012026

A. Based on number of Container used
    - Single Container Loading problem
    - Multiple Container loading problems.
B. Based on type of box to be packed
    - Homogenous Problems.
    - Heterogeneous problems.

3D bin-packing may involve a single container or multi container problems which is shown in Figure 2. The singular container packing problem involves only one container with either definite or infinite volume. Containers with infinite volume are defined with finite length and width, but with height extending to infinity. This allows packing solutions to pack until the set of boxes are exhausted. Solutions dealing with infinitely sized containers generally care most to compressing objects effectively. But most of the real time applications use the container of definite volumes. In the definite volume container, the boundaries of the container were fixed and the boxes should be placed within that boundary. In this some of the boxes may left unpacked. Another way to approach this problem is by considering multiple containers. Each container has definite volume. In this way, if the volume of the objects exceeds the volume of a container, an algorithm must make choices of which boxes to include in the packing and which to throw away to another container. In this work, heterogeneous of varying sizes were taken to pack into a single container of finite fixed volume.

3D-Bin packing Algorithms

There are various algorithms and methodologies available for solving the 3D bin packing problems. Some of them may consider the constraints or few. Commonly used 3D Bin Packing Algorithms of varying methodologies are

(A)Smile Algorithm.
(B) Wall building Algorithm
(C) Wall building Algorithm along Width
(D) Wall building Algorithm in L
(E) Tree Generating Heuristic
(F) Geometric Method
(G) Strip Packing
(H) Stability method
(I) Hybrid Algorithms
(J) Heuristic Algorithms
(K) Intelligent Algorithms
(L)Evolutionary Algorithm

12.Genetic algorithm

Recent revolutions in molecular genetics made clear that the modular organization of genes is highly important for evolution of complexity. Evolutionary concept of Genetic Algorithm (GA) was introduced by John Holland in 1975 at the University of Michigan. First stage in using Genetic Algorithm is initial population generation randomly based on probability logics and each individual in the randomly generated population is represented by chromosomes in biology. The method of random generation may vary from researcher to researcher. Chromosomes have many segments and ‘n’ number of strings depends on the complexity of the given problem [2]. These chromosomes were allowed for crossover. Thus the generated children have the properties of both the parents. Mutation operator may applied to the generated off spring to prevent from the stagnation condition. Finally the fitness function will be calculated and the worst parents need to be replaced by the best parents. This process need to be continued till the optimal solution reached. The various stages of the genetic algorithm are explained in the following sections.

12.1 Population generation
First stage in using the Genetic Algorithm is an initial population generation with 'n' number of strings generated randomly, based on probability logics [1, 2]. In a genetic algorithm, a population of strings called chromosomes or the genotype of the genome, which encodes candidate solutions called individuals, creatures, or phenotypes to an optimization problem, evolves toward better solutions are shown in Figure 3.

Figure 2. 3D Bin Packing Algorithm

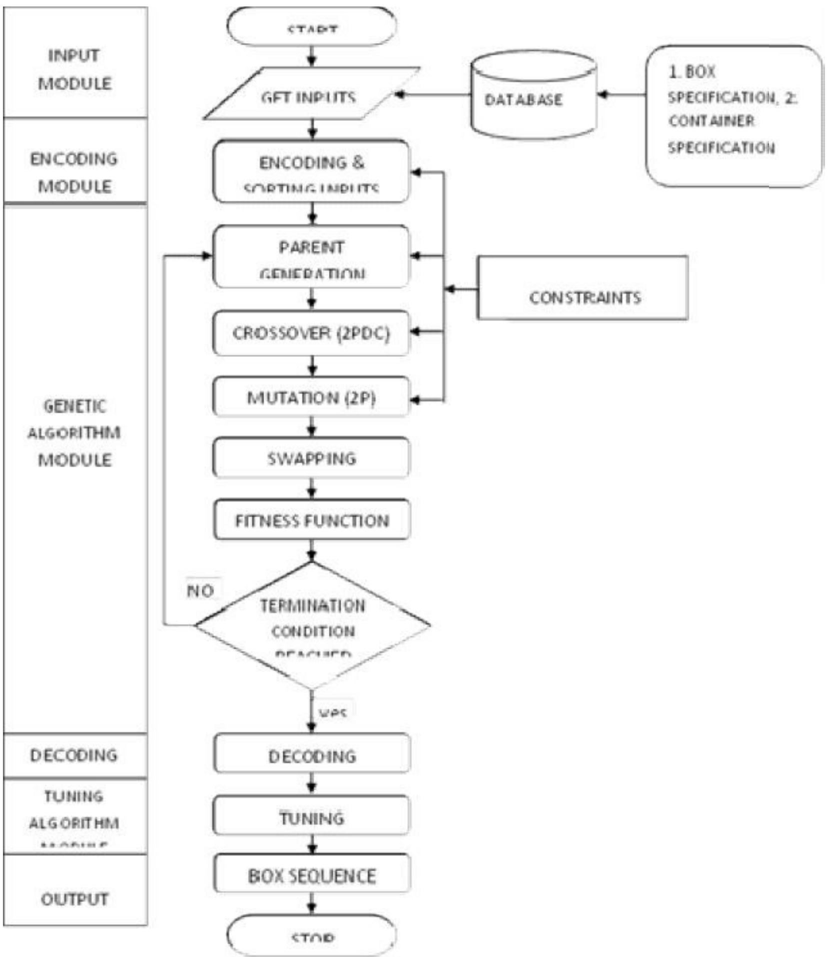

Figure 3. Over view of the solution method

Traditionally, solutions are represented in binary as strings of 0's and 1's, but other encodings are also possible. Decimal numbers, alphabets, alphanumeric values, ASCII values can also be used. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions [11]. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space)..

12.2 Reproduction

Reproduction or cross over is the second stage of GA, where the properties of the parents mixed to form best solution. The number of cross over site will depend up on the size of each parent. As the size increases, the number of cross over site should be increased to avoid the stagnation at a point. I.e. the set of strings may repeat in all the generation. Percentage of cross over also very from application to application. Many crossover techniques exist for organisms which use different data structures to store themselves.

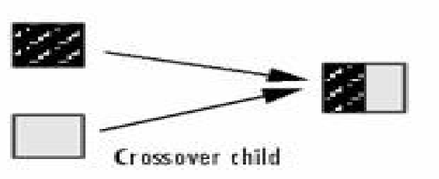

Figure 4. Crossover child

The various crossover techniques commonly used by the researchers are shown in Figure 4

- One Point Crossover
- Two Point Crossover
- Multi Point Crossover
- Uniform Crossover
- Half uniform Crossover
- Arithmetic Crossover

12.3 Mutation

Sometimes, the Fitness function value of the obtained offspring's may seem to stagnate around the optimal point [1]. Because the same strings may repeats again and again in the offspring leads to stagnation. This problem can be solved by applying the mutation operator, which helps to attain the best optimal point. One of the best known mechanisms for producing variations is mutation, where new trial solutions are created by making small, random changes in representation of prior trial solutions. If binary representation is used, the

Mutation is achieved by randomly flipping one bit (figure).

Generated Offspring	:	000|00000	before mutation
Mutant Offspring	:	000|10000	after mutation

Figure: Mutation

Fitness Function

The fitness function measures the quality of the generated solution. The fitness function is always problem dependent. Once the fitness function was defined, GA proceeds to generations repetitively by applying crossover, mutation, inversion and selection operators to produce optimal solution. Fitness function is a minimization or a maximization function used to calculate the relative value of the obtained solution with the expected.

12.4 Selection

Selection operation plays a major role in reducing the computational time. Proper selection of the parent will leads to faster convergence. Selection process may be by random process or by sequential process. Commonly used selection method is Roulette–wheel selection.

12.5 Swapping

Sometimes even after many iterative results, the strings in the starting position of the parents in the population remains same, even after many generations and may lead to stagnation. This can be avoided by using swapping operator. Swapping is a special type of Genetic operator used in this research, to widen the search space instead of being focused towards particular local optima. In this research, a single point random swapping was implemented. Strings beyond the swapping point should swap to the front, so that the strings get interchanged with in the same parent.

12.6 Termination

This is the stage in which the optimal solution was identified. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. The termination conditions may vary from application to application and some of the commonly used termination conditions are.

1. If the solution reached the minimum criteria, fitness function value reaches the desired minimum value.
2. If the fixed number of generation reached.
3. If the successive iterations no longer produce the better results, fitness function value remains the same for continuous iterations.
4. If allotted time/money/cost reached. By manual inspection.

13. Conclusion

In this Work, heuristic GA used for effective packing was described to solve multi constrained packing problem. It seems like, that these methods can be effectively used for these kinds of logistic problems. In the previous chapters, it was described, why to use and why not to use these algorithms. Solving the multi constrained bin packing problems are really hard. That is the reason in this work; GA is used to create a effective solution to solve optimization problems in a rather effective way.

14. References

[1] Goldberg D E 1988 Genetic Algorithm in Search*Optimization and Machine Learning* Addison Wesley.
[2] Mitchell M 1996 An Introduction to Genetic AlgorithmMIT Press.
[3] Hopper E and Turton B 1997Application of Genetic Algorithm to Packing Problems – A Review *Springer Verlag*, London, 279 -288.
[4] Korf R 2002 A New Algorithm of Optimal Bin PackingIn *proc. AAAI* 731-736.
[5] Korf R 2003 A Improved Algorithm for Optimal Bin Packing In proc. IJCAI 1252 – 1258.
[6] Pisinger D2002 Heuristic for Container Loading Problem*European Journal of Operation Research***141**292 – 382.
[7] Gehring H and Bortfeldt A 1997 A Genetic Algorithm for Solving the Container Loading Problem*International transactions in Operation Research***44** 401 – 418.
[8] Dowsland K A and Herbet E A 2004 Using Tree Search Bounds To Enhance A Genetic Algorithm Approach Two Rectangle Packing Problems *European Journal of Operation Research***168**390-402.
[9] Martello S and Pisinger D 2000 The Three Dimensional Bin Packing Problem *Operation research***48**256-267.
[10] Bischoff EE 2004 Three Dimensional Packing of Items with Limited Load Bearing Strength *European Journal of Operation Research***168** 952-966.
[11] Bortfeldt A and Gehring H2001 A Hybrid Genetic Algorithm for Container Loading Problem *European Journal of Operation Research***131**143 -16.