Multi Objective Optimization of Heterogeneous Bin Packing using Adaptive Genetic Approach

R. Sridhar1*, M. Chandrasekaran1 and Tom Page2

1Department of Mechanical Engineering, Vels University, Chennai - 600117, Tamil Nadu, India; srisampany@gmail.com, dr.chandrasekar@yahoo.in
2Loughborough University, Loughborough, UK; T.Page@lboro.ac.uk

Abstract

Objectives: The packing of goods in any industry is a tedious work. The proposed system evaluates the optimal packing and prediction of 3D bin packing maximize the maximize profit. Methods/Statistical Analysis: The Adaptive Genetic Algorithm (AGA) is used to solve the 3D single bin packing problem by getting the user input data such as number of bins, its size, shape, weight, and constraints if any along with standard container dimension. These inputs were stored in the database and encoded to string (chromosomes) format which were normally acceptable by AGA. Findings: The performance of the hybrid GA the Tuning algorithm is satisfactory and gives the feasible solution when compared with the other standard search algorithms. The minimum number of boxes left unloaded by using this algorithm will helps to validating the developed bin packing system. The developed Adaptive Genetic Algorithm was validated using the mathematical function. This research work is the good background of further development and analysis in this transportation domain of the following cases- Case 1: Homogenous boxes of same dimensions: all the boxes packed without gap. Case 2: Homogenous boxes of arbitrary dimensions: all the boxes packed with small gaps. Case 3: Homogenous/Heterogeneous boxes of arbitrary dimensions: all the boxes packed with gaps. Application/Improvements: The proposed adaptive genetic approach is very helpful in the logistic industries, especially for cargo packaging for export this is very helpful and can be easily implement any logistic industry.

Keywords: AGA, Bin Packing, Genetic Approach, Optimization, Tuning Algorithm

1. Introduction

The three-Dimensional Bin Packing Problem (3D-BPP) has a high practical relevance in modern industrial processes such as plane cargo management, warehouse management, pallet loading, and container ship loading. Author in1 worked on ‘stack building approach for cutting problem optimization’ is a gateway to packing problems proposed. Author in1 introduced first container loading approach for solving the liquid loading problem. Only a limited number of researches dealt the practical cases in container loading problems are exist. Author in1 solved a problem by use of heuristic with practical constraints as box orientations, stability along with container loading and stack restrictions, Author in1 considered rectangular items in the problem and uses heuristic approaches to find efficient packing patterns for loading in freight containers. In2 dealt a 3D-BPP without considering orientation restriction and proposed a mixed integer programming formulation to solve it and highlighted that 3D-BPP is strongly NP-hard. Author in2 studied non-identical cartons problem and proposed a pallet loading heuristic. In2 dealt a case of pipe packing in multiple containers loading. Author in2-investigated 3D-BPP with filling a single bin approach and estimated lower bounds.

Author in3 presented a heuristic based on the wall-building approach for 3D-BPP, where the problem was decomposed into smaller sub-problems by introducing strips and layers. Author in3 suggested a heuristic approach based on guided local search for packing cartons into a minimum number of identical containers with the constraint of no carton rotation is allowed. Author
in\textsuperscript{11} suggested a method to find strip packing solutions by using an exact branch and bounds exhaustive search for problem sizes up to 30 rectangles. They found a perfect packing’s (zero trim loss) in that problem. Author in\textsuperscript{12} suggested a new heuristic algorithm for solving the 3D-BPP. They used rules to mimic human intelligence.

Author in\textsuperscript{13} investigated 2D-BPP, and presented a new integer linear programming formulation and solved it by using CPLEX. Author in\textsuperscript{11} suggested a new heuristic algorithm for solving the 3D-BPP. They used rules to mimic human intelligence. Author in\textsuperscript{14} investigated 2D-BPP and suggested a hybrid genetic algorithm to maximize the number of objects packed in a single container with various volumes in nature. Author in\textsuperscript{14} presented a genetic approach for a kind of 2D-BPP called Rectangle Packing Area Minimization Problem (RPAMP). 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. Genetic Algorithm performance was 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. Author in\textsuperscript{16}, the Genetic Algorithm performce well in the optimization problems. Author in\textsuperscript{17} suggested that the GA based approach performs well in search algorithms. Author in\textsuperscript{18}, among the bio-Inspired Computational Algorithms, the GA based algorithms perormed well.

2. 3-D Bin Packing Problem

The 3-D Bin packing problem is advancement of 2D packing problem with additional difficulties and high practical relevance. In this problem the volume is considered as three dimensions like length, width and height or x, y and z axial dimensions to pack them in a bin efficiently. The three dimensional bin packing problems are classified as Single Container Loading problem, Multiple Container loading problems in which heterogeneous problems. The singular container packing problem involves only one container with either definite or infinite volume. Containers with infinite volume are defined with definite width and length, but height is extending to infinity. This will allow the 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 approach of this kind problem of is considering multiple containers. Each container has a definite volume. Thus, if the volume of the objects which is to be packed is exceeds than the volume of a container, an algorithm is employed to make choices for the next container. This research deals with heterogeneous of varying sizes were taken to pack into a single container of finite fixed volume.

3. Proposed Approach

This proposed approach includes 7 modules such as Input module, Encodcing module, Genetic algorithm module, Decoding module, Output module, Constraint implementation module and Tuning module. The proposed adaptive genetic approach furnished in Figure 1, which reveals the relationships in between modules.

The input module is used to interact with the user for getting the required data, the obtained input from

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{The proposed adaptive genetic approach.}
\end{figure}
3.2 Encoding Module

Encoding module is used to convert the user understandable data format into genetic acceptable format for further processing. Genetic algorithm can accept the binary digits, decimal values, alphabets and ASCII values. There is no standard hard rule existing for encoding process and the encoding digits will vary from application to application and the encoding methodology also varies from researcher to researcher. So a necessity arises to convert the user defined data into any of these formats. Most of the researchers used binary codes for genetic algorithm, as it is simple and easy to process. But the binary codes cannot be suitable for larger problems and the problems involved with more number of variables. In this work, for 10 boxes the data Size 400 X 100 Parents = 40,000, Binary Elements with 100 generation = 40,00,000, elements. So it requires more computational space, not possible to store in MS Excel Database, more computation time needed, understanding and interpreting difficult. Alphabets, ASCII values, Alpha-numeric values can lead to complexity in processing and understanding. So in this research work, decimal encoding was used and the boxes were converted to decimal numbers.

The second stage is the decimal encoding stage. Each box is represented by a decimal number and the set of decimal numbers is called as a chromosome which represents a set box packing pattern in the same sequence. In this four decimal number 1, 2, 3 and 4 were used to represents cubical box, rectangular prismatic box, cylindrical box and spherical box respectively (Table 1).

The sample chromosome of size 150 is given in Figure 3, each decimal number represents a box. The first appearance of the decimal number in the genetic chromosome represents the first box in that type, the second appearance of that decimal number represent the second box in that type from the same database and similarly the number of appearance represents the box serial number in the database. The dimension of the boxes can be retrieved directly from the database.

| Table 1 | Encoding format |
|---------|-----------------|
| **Decimal Digit** | **Box Type** |
| 1       | Cubical box     |
| 2       | Rectangular Prismatic box |
| 3       | Cylindrical box |
| 4       | Spherical box   |
3.3 Genetic Algorithm Module

The Genetic Algorithm is one of the best evolutionary algorithms used in this research to optimize the bin packing problem. In this research, five genetic operators were used to find the optimal solution and some of the operator modified for obtaining better solutions in less computational time.

3.4 Decoding Module

The genetic output will be in the chromosomal format and need to be converted to the user understandable format by decoding the chromosomes. To make the easy understanding of the output, Cartesian coordinate system was used to define the position and orientation of the boxes. Some of the criteria considered while decoding are:

- The container origin should be the lower most left corner (0, 0, 0) and that corner should be opposite to the front door of the container.
- Placement coordinate for the first box should always be the origin of the container.
- The length, width and height of the container should be along the global x, y and z-axis respectively.
- The length, width and height of the boxes should be along the local x, y and z-axis respectively.
- The container is placed in the Cartesian coordinate system to be able to relate the placement of the different boxes to one another.
- The layer by layer filling concept was used.

4. Output Modules

The output module calculates the output coordinate points in the user understandable format. In the Table 2, the sixth column 'Placement Corner' represents the Cartesian coordinate point of the container volume into which the

![Figure 3](image_url)  
Figure 3. A sample parent with string size of 150.

| S.No. | Chromosome | Box Type | Box Dimension | Box no | Placement Corner | Box Corner | Box Position | Volume |
|-------|------------|----------|---------------|--------|------------------|------------|--------------|--------|
| 1     | 1          | Cube     | 10 10 10      | CB1    | 0 0 0            | 10 10 10   | 1 1 1         | 1000   |
| 2     | 4          | Sphere   | 10 10 10      | CB-4-1 | 10 0 0            | 20 10 10   | 1 1 2         | 1000   |
| 3     | 2          | Rect. prism 5 | 10 5 2 | RP-5-1 | 20 0 0            | 30 5 2     | 1 1 3         | 100    |
| 4     | 2          | Rect. prism | 10 5 2 | RP1    | 30 0 0            | 40 5 2     | 1 1 4         | 200    |
| 5     | 3          | Cylinder  | 5 5 50        | CY1    | 40 0 0            | 45 5 50    | 1 1 5         | 3450   |
| 6     | 2          | Rect. prism | 10 5 2 | RP2    | 45 0 0            | 55 5 2     | 1 1 6         | 100    |
| 7     | 3          | Cylinder  | 5 5 50        | CY-6-1 | 55 0 0            | 60 5 50    | 1 1 7         | 4800   |
| 8     | 4          | Sphere   | 10 10 10      | CB-4-2 | 60 0 0            | 70 10 10   | 1 1 8         | 5800   |
| 9     | 2          | Rect. prism | 10 5 2 | RP3    | 70 0 0            | 80 5 2     | 1 1 9         | 5900   |
| 10    | 2          | Rect. prism | 10 5 2 | RP4    | 80 0 0            | 90 5 2     | 1 1 10        | 6000   |
|       |            |          |               |        |                  |            |              |        |
| 150   | 4          | Sphere   | 20 20 20      | CB-4-23 | 25 45 250        | 45 65 270  | 6 4 3         | 8000   |
|       |            |          |               |        |                  |            |              | 280750 |
lower left most corner of the first box has to be placed. The placement corner of the first box given in the row 1 is (0, 0, 0) i.e., the first box has to be placed in the origin of the container. The second box has to be placed in the coordinate point (10, 0, 0) and so on. These coordinate points represent X, Y and Z axis respectively. The X axis represents along the length of the container, the Y axis represents along the width of the container and the Z axis represents along the height of the container. The location of the last box is given as (25, 45, 250).

The seventh column ‘Box Corner’ represents the opposite diagonal corner of the packed box. These value also represented in Cartesian coordinate points along X, Y and Z axis. The Box corner and its placement corners were illustrated in the Figure 4. These coordinate points used to calculate the next placement point and these check the overlapping of the boxes and helps in compact and perfect packing of the boxes without empty spaces between them. The Figure 5 shows the layout Boxes for Packaging along the Longitudinal Direction. There are eighth column ‘Box Position’ represents the location of the box inside the container. The ‘Layer’ represents the Cartesian Z-axis and along the height of the container. The layer-1 is the bottom most layers inside the container, Layer-2 represent the second layer and so on. In the Figure 6 and as per the Table 2, six layers of boxes were packed inside the container. The number of boxes in the each layer will vary from layer to layer.

The ‘longitude’ column represents the position of the boxes along the Cartesian Y-axis or along the width of the container. The sum of the width of the packed boxes along the width of the container should be less than the container width and once the sum of the width of the boxes exceeds the container width, then the forthcoming boxes should be packed in the second layer of the container. The longitudinal placement of the boxes were shown in the Figure 7 and the boxes along the row were shown in Figure 8. The number of longitudinal columns in the layer 1, 2, 3, 4, 5 and 6 form the table was 4, 4, 3, 3, 4 and 4 respectively.

The next column ‘transverse’ represents the position of the boxes along the transverse direction of the container or along the Cartesian X-axis. In this the boxes are packed along the length of the container and when the sum of length of packed boxes exceeds the container length, then the boxes should be placed in the second row and so on. From the table, then number of rows in the first layer and

![Figure 4. Box represents placement corner and box corner.](image1)

![Figure 5. The boxes packed along the longitudinal direction.](image2)

![Figure 6. The layers of boxes.](image3)

![Figure 7. The longitudinal placement of the boxes.](image4)
Many experiments were conducted for different bench mark problems with various conditions and the items observed were proved that the increase in number of box types, shapes and sizes was directly proportional to the amount of empty space formation inside the container. Because the fitness function of the Genetic Algorithm was developed to optimize the container volume and not the box packing sequence optimization. So there exist a gap between the volume optimization and the packing sequence optimization. The result shown in the table can be easily understood by the engineers and it may be difficult for the layman. So the result generated should be in the layman understandable formats i.e., by getting the result, the layman has to pack the boxes without expecting any assistance and guidance to minimize the time involved in packing. A separate algorithm developed in this work to solve the above mentioned problems was the tuning algorithm.

5. Tuning Algorithm Module

The tuning algorithm is a special heuristic algorithm developed in this research work to convert the genetic output in the user understandable format and also it tries to fill the remaining empty spaces. Hence this format helps the user to understand and visualize the bin packing inside the container easily and also it helps to find the placement corners and the amount of gap formed in each layer. Once an empty volume is found, it can be filled by suitable boxes to achieve complete packing inside the container and thereby obtaining the minimum empty space using the tuning algorithm. Figure 9 shows the top view of a first layer of boxes arranged for the data given Table 4.

| Layer Number | Column Number (Longitude) | Row Number (transverse) | Number of boxes |
|--------------|---------------------------|-------------------------|-----------------|
| 1            | 1                         | 11                      | 40              |
|              | 2                         | 10                      |                 |
|              | 3                         | 11                      |                 |
|              | 4                         | 08                      |                 |
| 2            | 1                         | 08                      | 29              |
|              | 2                         | 05                      |                 |
|              | 3                         | 08                      |                 |
| 3            | 1                         | 06                      | 17              |
|              | 2                         | 06                      |                 |
|              | 3                         | 05                      |                 |
| 4            | 1                         | 05                      | 18              |
|              | 2                         | 08                      |                 |
|              | 3                         | 05                      |                 |
| 5            | 1                         | 07                      | 23              |
|              | 2                         | 06                      |                 |
|              | 3                         | 05                      |                 |
| 6            | 1                         | 05                      | 18              |
|              | 2                         | 05                      |                 |
|              | 3                         | 05                      |                 |
|              | 4                         | 03                      |                 |
Table 4. Standard dimension of the shipment container

| Internal dim.  | 20’ standard | 40’ standard | 20’ open top | 40’ open top | 20’ flat track |
|---------------|--------------|--------------|--------------|--------------|---------------|
| Length(mm)    | 5886 – 5940  | 12040-12062  | 5890 - 5905  | 19905-12020  | 5920 - 5952   |
| Width(mm)     | 2340 - 2370  | 5920 - 5952  | 2330 - 2348  | 2285 – 2315  | 2310 - 2389   |
| Height(mm)    | 2380 - 2410  | 2370 - 2377  | 2300 - 2345  | 2300 – 2360  | 2090 - 2175   |

Door opening

| Height(mm)    | 2275 - 2286  | 2270 - 2275  | 2240 - 2317  | 2220 – 2315  |
| Width(mm)     | 2317 - 2335  | 2330 - 2355  | 2206 - 2325  | 2265 – 2315  |

Weights

| Max. gross(Kg)| 20320        | 30480        | 20320        | 30480        | 20320         |
| Tare(Kg)      | 2100 - 2240  | 3560 - 3670  | 2250 - 2350  | 3850 – 3950  | 2870 - 3170   |
| Max. pay load | 18100-18300  | 26810-27000  | 18070-18140  | 26530-26630  | 17150-17420   |
| Cubage(Kg)    | 32 - 34      | 67 - 68      | 31.6 – 32.7  | 65 – 67      | 30.5 – 32.5   |

The left figure shows the top view of packing of the bin for the data shown in Table 4, before applying Tuning algorithm. Figure 9 shows some empty spaces formed between the boxes. This empty space formation may be because of the many reasons. This type of spaces cannot be formed for the homogenous boxes or the boxes with same shapes. As the boxes are of heterogeneous shapes and arbitrary sizes with unpredictable in nature, the space formation becomes an unavoidable issue. Therefore a heuristic algorithm was developed to fill this type of left out empty spaces.

The positions of all the bins in the Cartesian coordinate system were given by the software as the output table. The placing arrangement of the bins inside the container in the layer by layer format was also given in the output in a graphical format as shown in the Figure 9. By reading this data, the tuning algorithm calculates the empty space formed in each layer. Then the tuning algorithm reads the output and finds the details of the packed and unpacked boxes. All the results were stored in a MS excel/Access database for analysis. The tuning algorithm checks the size of unpacked boxes with the empty space. If any of the box dimension and the empty space formation is same or more-or-less equal, then the tuning algorithm fixes that box into that gap. There by tuning algorithm fix the possible boxes into the empty spaces formed in each and every layer. Thus, the container volume was utilized properly and maximum number of bins was packed into the container. The right side of the Figure 8 shows the boxes filled by the tuning algorithm. Even though the tuning algorithm may fill the boxes, all the packed boxes should satisfy the constraints involved in the bin packing problems.

6. Discussion

As the genetic algorithm operates only on the chromosomes, in this research work, input format was in chromosome format. Each segment in the chromosomes represents a box and the chromosome as the whole represents a set of box packing pattern. The number of chromosomes is normally denoted by population size and that population size should be fixed based on the complexity which gives the empty space inside the container. Fitness function value was calculated for all the parents in the population and sorted based on fitness value. Two point Cross over operator was applied between the selected parents and allowed to reproduce two offspring’s. The generated offspring were allowed to crossover with the third parent. Each generated offspring inherits best properties from all the three parent’s i.e., best position of packing boxes in to the container. Packing and positioning of bins found may not be optimal in first generation itself and optimal fitness seems to be stagnating around the optimal point, so mutation operator were used to attain the best optimal point.

In this research, mutation operator was overloaded with the orientation changing function. The various constraints were applied at various stage of the module. Finally generated chromosome has optimal and feasible solution for packing of boxes into the container. In this research work several preliminary experiments were conducted to determine the suitable parameter values shown Table 2. Final result obtained from the genetic algorithm was fine tuned by the tuning algorithm.

The two point double crossover genetic approach consistently found better solutions than the steady-state
Originally, the intention when starting this work was to create a way to handle bin packing problem with the binary coded genetic approach, but leads to computational memory problem. While this sounds like a minor problem, it should become obvious when problem size increases, that this is an important feature for many real time packing problems. There are several constraints have to face with, when choosing an algorithm. The goal of this is to make a good tradeoff between time, space and complexity: this is rather hard. At the end it can be said, that no better or no hard algorithm exists.

7. Conclusion

Solving the multi constrained bin packing problems are really hard, so there is need to find the new techniques. In this research work an Adaptive Genetic Algorithm (AGA) is used for effective solution to optimization problem. The developed AGA algorithm turned out to perform well, despite the fact that many improvements like 2PDC, orientation constraint overloaded mutation operator, swapping operator, tuning algorithm were made to increase the search Space.

In this research work most of the practical constraints were considered and the results obtained by a combination of the hybrid GA with the Tuning algorithm were found satisfactory and feasible for packing, when compared to other standard search algorithms. Minimum number of boxes left unloaded using this algorithm will helps in validating the developed bin packing module. The developed Adaptive Genetic Algorithm was validated using the mathematical functions. Finally, it can be expected that this paper may be a good background for the further development and analysis in this transportation domain.

Case 1: Homogenous boxes of same dimensions: all the boxes packed without gap.
Case 2: Homogenous boxes of arbitrary dimensions: all the boxes packed with small gaps.
Case 3: Homogenous/Heterogeneous boxes of arbitrary dimensions: all the boxes packed with gaps.

8. References

1. Goldberg DE. Genetic Algorithm in search. Optimization and Machine Learning. USA: Addison Wesley; 1989.
2. Mitchell M. An introduction to Genetic Algorithm. Cambridge: MIT Press; 1996. p. 1–162.
3. Hopper E, Turton B. Application of genetic algorithm to packing problems– A review. London: Springer Verlag; 1997.
4. Korf R. A new algorithm of optimal bin packing. Proceeding of AAAI; 2002. p. 731–6.
5. Gehring H, Bortfeldt A. A genetic algorithm for solving the container loading problem. International Transactions in Operation Research. 1997; 4(5-6):401–18.
6. Dowsland KA, Herbet EA. Using tree search bounds to enhance a genetic algorithm approach to two rectangle packing problems. European Journal of Operation Research. 2004; 168(2):390–402.
7. Martello S, Pisinger D. The three dimensional bin packing problem. Operation Research. 2000; 48(2):256–67.
8. Pisinger D. Heuristic for container loading problem. European Journal of Operation Research. 2002; 141(1):292–382.
9. Bischoff EE. Three dimensional packing of items with limited load bearing strength. European Journal of Operation Research. 2004; 168(3):952–66.
10. Bortfeldt A, Gehring H. A hybrid genetic algorithm for container loading problem. European Journal of Operation Research. 2001; 131(1):143–61.
11. Container loading with multi-drop constraint. masters thesis. Informatics and mathematical modeling. Lyngby: Technical University of Denmark, DTU. Available from: http://www2.imm.dtu.dk/pubdb/p.php?5225
12. Davies AP, Bischoff EE. Weight distribution considerations in container loading. European Journal of Operation Research. 1999; 114(3):509–27.
13. George, George JM. Packing different sized circles into a rectangular container. European Journal of Operation Research. 1995; 84(3):693–712.
14. Kang K, Moon I, Wang H. A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem. Applied Mathematics and Computation. 2012; 219(3):1287–99.
15. Bortfeldt A. A reduction approach for solving the rectangle packing area minimization problem. European Journal of Operational Research. 2013; 224(3):486–96.
16. Pugazhenthi. Optimization of permutation flow shop with multi-objective criteria. International Journal of Applied Engineering Research. 2013; 8(15):1807–13.
17. Swapna BS, Sivanandam N. A survey on cryptography using optimization algorithms in WSNs. Indian Journal of Science and Technology. 2015; 8(3):216–21.
18. Anita CJ, Ramesh R, Vaishali D. Bio-inspired computational algorithms for improved image steganalysis. Indian Journal of Science and Technology. 2016; 9(10):1–10.