International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com‖ Volume 2 Issue 4‖ April 2016 ‖ PP 67-71

SMART EVOLUTIONARY ALGORITHM FOR CONSTRAINED CONTAINER LOADING PROBLEM

A.D. Jaisree, R. Umagowri, S.K. Rajesh Kanna
Student, Mahindera Engineering College, Chennai, India
Assistant Professor, Mahindera Engineering College, Chennai, India
Professor, Rajalakshmi Institute of technology, Chennai, India

ABSTRACT: This paper addresses the issue of identifying the best Bin Packing pattern from the available bins by satisfying the packing constraints. This research adopted the concept of smart operator that allows setting the genetic parameters for any kind of situations and thereby increasing the performance of any packing problem. The result obtained from this heuristic algorithm matches with several lower bounds proposed in the literature. The developed smart operators have many applications.

Keywords: Bin Packing, genetic algorithm, smart operator, constraints.

1. INTRODUCTION

The problem addressed in this paper is two dimensional bin packing problem using smart genetic operators. The objective of this research is to assign each bin into a rectangular container, such that the sum of the area of the packed bins and the area of container are more or less equal, by satisfying the constraints such as boundary crossing constraint and overlapping constraint. The boundary crossing constraint is used to maintain the area of the packed bins is not greater than the capacity of the container. The overlapping constraint is used to avoid the overlapping of bins with themselves and with the container boundary.

In general Martello and Toth (1990) defined the bin packing problem as NP-hard. So the result obtained from the traditional algorithm cannot be satisfied in all the unpredictable situations and created a need for a universal algorithm which can adopt in the unpredictable varying environment of bin packing. The contribution of this paper is to develop a general smart algorithm to solve the above mentioned problem using genetic approach. The algorithm provided is also allows to derive a best packing pattern by satisfying constraints.

2. PERVIOUS WORK

2D bin packing problems are used to optimize any two parameters and the third parameter assumed to remain same for all the boxes. Andrew Lim and Xingwen Zhang (2003) used the integer programming and simulated annealing for scheduling problems. He proved that the integer programming concepts works well for smaller problems and cannot be suitable for the larger problems. Armando Ponce Perez et al (2005) solved the 2D polygonal shaped bins into a rectangular container using genetic algorithm. Genetic parameters like population size, maximum generation, cross-over, and mutation probability were set to 100, 500, 0.8222 and 0.072 respectively. Andres Bortfeldt (2006) used genetic algorithm for orthogonally packing rectangular strips into a container of fixed width and variable length. Eleni Hadjiconstantinou and Manuel Iori (2007) solved the 2D cutting problem using genetic algorithm by modifying the operators to generate optimal value. This algorithm did not rotate the items and assumed that all the edges should be parallel. Even though the result obtained was satisfactory, they failed to concentrate on orientations and other major constraints.

Jakob Puchinger and Gunther Raidl (2007) developed an integer linear programming model, branch and price algorithm, fast greedy and evolutionary algorithm for packing items into minimum number of containers. Wenqi Huang and Duanbing Chen (2007) compared genetic algorithm, simulated annealing and heuristic algorithm for packing rectangles inside a rectangle. Zhang et al (2007) introduced recursive based heuristic genetic algorithm for rectangular strip packing problem. Lau et al (2009) compared the performance of three evolutionary algorithms for optimizing profit value in packing instead of reducing empty space for multi-pallet loading operations. Adewumi and Ali (2010) developed multi-level genetic approach for space allocation problem of hostel accommodation. Edmund Burke and Graham Kendall (2011) compared intelligent meta-heuristic algorithms (Genetic algorithm, Tabu search and simulated annealing) for 2D strip packing problem.

The above discussion shows that intelligent and evolutionary techniques produce better result than the heuristic techniques for complex NP-Hard problems. But most of the researchers assumed that the height of the boxes remains same and optimized the length and width. This assumption cannot be true for all the time and most of researcher didn’t consider the practical constraints.

3. BIN PACKING PROBLEM

Given a set 'B' of n bins, each with a length L_i, width W_i and height H_i, i = 1 to n, i € B, and rectangular containers of fixed size of fixed volume V_c. Let us assume, the height of all the bins remains same as one unit. Volume of bin i is V_i, then the objective function is $\sum_{i=1}^{n} Vi$ is at most equal to the bin volume V_c. The obtained volume then be normalized onto the interval [0,1], whereas 1 corresponds to the bin volume V_c. This value is achieved in this research work by applying genetic algorithm.

4. GENETIC ALGORITHM

The concept of genetic algorithm was introduced by John Holland in 1970. They have been widely studied and applied to optimize the non-linear functions bounded with constraints. The various genetic operators used to identify the best parent is given in the following sections.

4.1 Parent Generation

The first stage in the genetic algorithm is the initial population generation using random generator. In this research work, decimal encoding is used. The random numbers have been generated in the range between 1 and 3. In general, the parent size has to be set by the user. The items identified for the fixed parent size are as follows.

1. For smaller sized boxes, more empty space will be formed.
2. For larger sized boxes, some of the bins left unpacked.
3. For bins which are distributed equally, packing pattern obtained with least empty space.

As most of the practical cases, bin size cannot be predicted in advance. So in this work, smart operator is developed to fix the size of the parent based on the available bins. The smart operator varies the parent size for each and every iteration till the termination condition achieved. The sample generated parent of size 50 is given in the figure 1.

Parent 1 :
1 3 2 2 2 3 1 1 2 1 2 3 3 3 2 1 2 3 3 3 2 1 2 2 3 3 2 3 2 1 2 1 2 3 3 2 1 1 2 2 2 1 2 2 1 2 3 3 2 2
Parent II :
2 2 1 2 3 3 2 1 1 2 2 3 3 2 1 1 2 2 1 2 1 2 1 3 1 3 2 2 1 1 2 2 1 2 3 2 3 3 2 1 2 1 2 1 2 3 3 2 3 1

Figure 1 : A Sample Set Of Generated Parent

4.2 Encoding

Encoding is the process of converting the user defined data into genetic understandable data. In this research, the bins are classified as cubes, lengthier bins, and wider bins, they are encoded as 1, 2 and 3 respectively. The corresponding appearance of a decimal number in the parent chromosome represents the corresponding bin number in that category.

4.3 Crossover

The significant properties relevant to the problem are transferred from a set of parent to the child by means of crossover operation. Thereby the crossover operation identifies the better solution from the larger search space and in every generation it converges towards the better solution by inheriting the best properties from the best parents. The parents for the crossover were generated using random function. The developed smart operator also used to set the crossover sites and is given in the equation 1.

$$Cs = P_s/X \tag{1}$$

whereas Cs = number of crossover sites,
Ps = Number of strings in a parent

$$X = \begin{cases} Ps/3, \text{ if } 1 \le N \le (N/3) \\ Ps/2, \text{ if } (N/3) \le N \le (2N/3) \\ Ps/1, \text{ if } (2N/3) \le N \end{cases}$$

Once the crossover site has been decided by the smart operator, the crossover sites have been generated randomly. Thus the number of crossover sites will be higher at the start of the generation to increase the search space and thereby identifying the best solution. As the number of generation increases, the problem converges towards the optimal solution and the lower crossover sites preserve the optimal convergence deviation from its path. The sample crossover operation is shown in the figure 2. The next stage after the crossover operation is mutation.

Parent 1:
1 3 2 2 2 3 1 1 2 1 2 3 3 3 2 1 2 3 3 3 2 1 2 2 3 3 2 3 2 1 2 1 2 3 3 2 1 1 2 2 2 1 2 2 1 2 3 3 2 2

Parent II:
2 2 1 2 3 3 2 1 1 2 2 3 3 2 1 1 2 2 1 2 1 2 1 3 1 3 2 2 1 1 2 2 1 2 3 2 3 3 2 1 2 1 2 1 2 3 3 2 3 1
N = 01; Ps = 50; Cs = 3; { 5-12-31}

Offspring 1:
1 3 2 2 2 ***3 2 1 1 2 2 3*** 3 3 2 1 2 3 3 3 2 1 2 2 3 3 2 3 2 1 2 ***2 1 2 3 2 3 3 2 1 2 1 2 1 2 3 3 2 3 1***

Offspring 2:
2 2 1 2 3 3 1 1 2 1 2 3 ***3 2 1 1 2 2 1 2 1 2 1 3 1 3 2 2 1 1 2*** 1 2 3 3 2 1 1 2 2 2 1 2 2 1 2 3 3 2 2

Figure 2: A Sample Crossover Operation

4.4 Mutation

Mutation is the process of swapping the random gean at random location to avoid stagnation at a point in the solution space. the number of mutation points have been decided by the smart operator and is given in the equation 2.

$$Ms = (Cs * X)/100 \qquad (2)$$

The mutation site and the mutant string have to be selected at random. A sample mutation operation is shown in the figure 3.

Offspring 1:
1 3 2 2 2 ***3 2 1 1 2 2 3*** 3 3 2 1 2 3 3 3 2 1 2 2 3 3 2 3 2 1 2 ***2 1 2 3 2 3 3 2 1 2 1 2 1 2 3 3 2 3 1***
Ms = 1; Mp = 10; Mt = 1
Offspring 1.1:
1 3 2 2 2 ***3 2 1 1 1 2 3*** 3 3 2 1 2 3 3 3 2 1 2 2 3 3 2 3 2 1 2 ***2 1 2 3 2 3 3 2 1 2 1 2 1 2 3 3 2 3 1***

Figure 3: A Sample Mutation Operation

4.5 Fitness Function

Fitness function is used to identify the best parent from the available solution. The fitness function for the bin packing problem can then be formulated and is given in the equation (3).

$$Min\ f(x) = Vc - \left(\sum_{iter=0}^{100} \mathrm{Vg} + \sum_{i=1}^{n} V_i\ X_{ij} \right) \qquad (3)$$

whereas f(x) is the minimization fitness function
X_{ij} =1, if the bin packed, else 0
Vg = occupied volume of the bins for iterations g = 1 to 100

The fitness function developed is used to identify the best packing pattern that yield packing pattern without or with less empty space inside the container. In this research, each iteration has to run for 100 generations and the best parent at the 100^{th} generation is the best in that iteration. The volume occupied by that parent is represented as Vg. The number of generation in each iteration is fixed as 100 and the iterations are terminated by satisfying the termination conditions.

4.6 Termination conditions

As there are various termination conditions available for terminating the genetic operations, but as this research used smart operators, the termination condition used is given as follows.

1. If 100 iterations reached.
2. If the parent size Ps = 0 achieved
3. If the Xij =1 for i,j = 1 to n.
4. If ∑Vg = Vc

5. CONSTRAINTS

The major constraint considered in this work are the overlapping and the boundary crossing constraint. Overlapping of the bins among themselves and with the container boundary can be avoided by generating placement coordinate for each and every bin. The placement coordinate has the x, y and z values in which the bins has to be placed. This values can be generated by the smart placement operator. The boundary crossing constraint was satisfied by generating the layer by layer packing concept. The pseudo code used to check the boundary crossing constraint and overlapping constraint is given in the figure 4.

```
CP[0,0,0] = {0,0,0};
i=j=k=0;
For x = 1 to noofpackedbins
  CP(i,j,k) = [0,0,0];
  CP(i+1,j,k) = {CP(i)+Li, j,k};
  Li = CP(i+1);
  i++  ;
  If Li ≥ Lc then
 { i = 0; j++ }
  If Wi ≥ Wc then
  { i=0; j=0; k++; }
```

Figure 4: Pseudo code to check the constraints

6. RESULTS AND DISCUSSION

The smart genetic algorithm was developed in visual basic language on personal computer Pentium i5 processor with 3.10 Ghz. The maximum number of iteration, number of parent in a generation and maximum number of generations in each iteration has been set to 100. The developed smart operator has been used to set the parent size, number of crossover sites and number of mutation sites. For the multi objective bin packing problem, the smart algorithm used to reduce the waste empty space inside the container. The constraints like boundary crossing and overlapping constraints have been eliminated by generating the feasible parent solution using smart operator. The developed module was tested with 13 different set of input data set and the obtained results were found quit superior then traditional genetic algorithm. The sample normalized output is shown in the figure 5. The normalized parent size and the empty space are represented in the Y axis. Number of iterations is represented in the X axis. From the figure it is clear that the parent size were reduced in four stages and the empty space converge to the minimum zero.

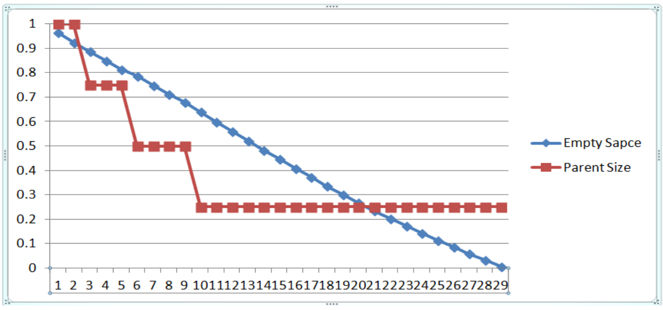

Figure 5: Sample output

The figure 6 shows the fitness value obtained from the traditional GA and the smart GA. The fitness function value of the traditional GA increases with increase in the iterations, but the fitness function value for the smart GA oscillates at the point of change in the number of strings in the parent and the fitness function value decreases as more it calculates only based on the available empty space instead of total container volume, thereby the computational complexity reduced.

The figure 7 compares the best fitness value obtained from the traditional GA and smart GA for random ten runs. It become clear that the traditional GA is giving different solution for every run and the smart GA giving standard solution for every run. Thus it proves that the smart GA is the stable and provide best result every time and it can be used for all type of optimization problems.

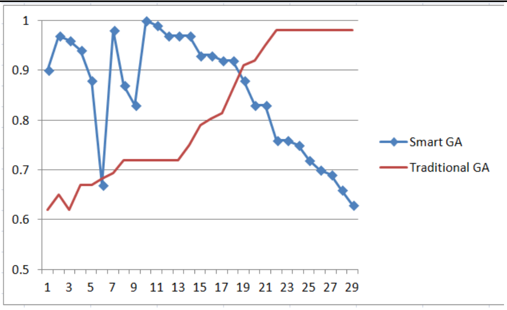

Figure 6: Fitness Function Value Comparison of traditional and Smart GA

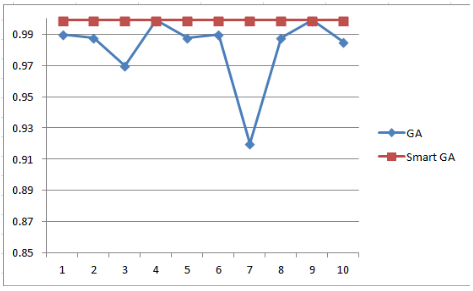

Figure 7: Best Parent Comparison of traditional and Smart GA

7. CONCLUSIONS

In this paper, we introduced a smart operator to set the genetic parameters to yield best result. As the parent size has been reduced and makes the complete packing of bins into the container. As some of the evolutionary algorithms operate over the random numbers, the result obtained from each generation can differ, but the smart operator has been developed to generate the best result all the time and each run. As the computational time is concerned, smart algorithm consumes more compared to the traditional GA, but the latest digital machines and parallel processing machines can process in less time.

REFERENCES

[1]. Adewumi A.O. and Ali M.M. 'A Multi-Level Genetic Algorithm for A Multi-Stage Space Allocation Problem' Journal of Mathematical and Computer Modelling, Vol. 51, Issues 1-2, 2010, pp. 109-126.

[2]. Andrew Lim and Xingwen Zhang, 'Integer Programming and Simulated Annealing for Scheduling Sports Competition on Multiple Venues', The Fifth Meta-heuristics International Conference, Kyoto, Japan, August 25–28, 2003.

[3]. Armando Ponce Perez, Arturo Perez-Garcia and Victor Ayala-Ramirez, 'Bin Packing Using Genetic Algorithms', Proceedings of the 15th International Conference on Electronics, Communications and Computers, IEEE, 2005.

[4]. Andreas Bortfeldt, 'A Genetic Algorithm for the Two-Dimensional Strip Packing Problem with Rectangular Pieces', European Journal of Operational Research, Vol. 172, 2006, pp. 814–837.

[5]. Edmund Burke and Graham Kendall, 'Comparison of Meta-Heuristic Algorithms for Clustering Rectangles', Journal of Computers and Industrial Engineering, Citeseer, Vol. 37, Issue 1, 1999, pp.383-386.

[6]. Eleni Hadjiconstantinou and Manuel Iori, 'A Hybrid Genetic Algorithm for the Two-Dimensional Single Large Object Placement Problem', European Journal of Operational Research, Vol. 183, 2007, pp. 1150–1166.

[7]. Jakob Puchinger, Gaunther R. Raidl and Martin Gruber, 'Cooperating Memetic And Branch-And-Cut Algorithms For Solving The Multidimensional Knapsack Problem', MIC2005. The 6th Meta-heuristics International Conference, Vienna, Austria, August 22-26, 2005, pp. 775- 780.

[8]. Lau H.C.W., Chan T.M., Tsui W.T., Ho G.T.S. and Choy K.L. 'An AI Approach For Optimizing Multi-Pallet Loading Operations', Journal Of Expert System With Applications, Vol. 36, Issue 3, Part 1, April 2009, pp. 4296-4312.

[9]. Wenqi Huang and Kun He, 'A Caving Degree Approach for the Single Container Loading Problem', European Journal of Operational Research, Vol. 196, 2009, pp. 93–101.

[10]. Zhang De-Fu Chen, Sheng-Da1 and Liu Yan-Juan, 'An Improved Heuristic Recursive Strategy Based on Genetic Algorithm for the Strip Rectangular Packing Problem', Acta Automatica Sinica, Vol. 33, No 9, 2007, pp. 911-916.