IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 12, Issue 4 Ver. VII (Jul. - Aug. 2015), PP 65-70
www.iosrjournals.org

Optimization of 3D Constrained Rectangular Bin Packing Problem Using Recursive Ant Colony Algorithm

S.K. Rajesh Kanna[1], A.D. Jaisree[2], K. Balasundaram[3], S. Bharani Kumar[4]
[1,3,4]*(Department of Mechanical Engineering, Rajalakshmi Institute of Technology, Chennai, India)*
[2]*(Dept. of Communication and Network, Anna University, Chennai, India)*

Abstract: *This research presents an application of Ant colony optimization meta-heuristic to the bin packing problems. Hybridization of Ant tuning strategy has introduced to improve the results obtained from basic ant system by utilizing the behaviour of artificial ants to perform local search. The objective is to pack the arbitrary sized three dimensional rectangular prismatic bins into the container of standard sizes in such a way that the empty space inside the container should be least as minimum. The bins should be packed by satisfying the packing constraints such as placement constraint, overlapping constraint, shipment placement constraint, stability constraint, etc. The output from the ants algorithm have been further optimized and decoded to the user understandable graphical format using heuristic Ant tuning algorithm. Experiments using well-known benchmark bin packing problems show that this approach improves on the performance and is competitive with other evolutionary in computational time also.*
Keywords: *Ant Colony Algorithm; Ant tuning Algorithm; Bin packing problem; Optimization;*

I. Introduction

This research examines an application of meta-heuristic Ant Colony Optimization algorithm (ACO) for the Bin Packing Problems (BPP). In the static BPP, a finite number of three dimensional rectangular prismatic bins of arbitrary sizes are to be packed in to a finite number of standard sized rectangular prismatic containers with the objective of reducing the number of containers used for the packing by satisfying the packing constraints. In other words, the empty spaces inside the container need to be reduced. Each bin consists of a predetermined sequence of packing pattern based on the unloading centres, each which needs to be packed without interruption for unloading. Similar shaped/sized bins cannot be grouped and packed concurrently and each bin must analyze for its fragility, stability, orientation, etc. A feasible bin packing pattern is an assignment of available bins on a container without violation of the packing constraints. A bin packing fitness value is defined as the maximum volume occupied by the packed bins inside the container. The objective of the BPP is to find a packing pattern that minimizes the empty space inside the container or maximize the packed bin volume inside the container. A good packing pattern is one that minimizes the waste empty space inside the container, because theses empty spaces also include in the freight rate and in turn reflect in the total product cost without adding any value to the product.

The remainder of this paper is as follows: Section 2 gives simple survey on the past research work on bin packing problems and utilization of Ant colony algorithm. Section 3 is for a formal description of the bin packing problem and Section 4 introduces the Ant Colony Algorithm methodology. Hybridization of Ant colony algorithm with the Ant tuning algorithm is given in Section 5 followed by experimental results using benchmark problems in Section 6. Concluding remarks and future work is explored in Section 7.

II. Literature Survey

BPP is a multi-objective problem with ‘n’ number of conflicting constraints, so according to complexity theory, the BPP has been characterized under NP-hard combinatorial optimization problem, because Garey (1979) defined the NP hard problem as the problem for which obtaining the exact solutions is computationally intractable. Recent researchers proved that the NP-hard problems can be tackled by evolutionary algorithms and meta-heuristic algorithms. Dorigo and Di Caro (1999) proved the ACO is one among the best meta-heuristic algorithms. ACO is inspired by the study of argentine ants conducted by Goss et al.. Marco Dorigo et al. (1999) applied the principles of cooperative behaviour found in real ants colonies to solve NP hard optimization problems. Ant information exchange is achieved by mimicking the foraging and recruiting behaviour. Bullnheimer et. al (1999) proved that each ant in a colony constructively builds solution in several stepwise probabilities until a solution have been reached, so the same approach can be used to various hard combinatorial optimization problems.

Martello (2000) developed exact branch-and-bound algorithm for 3D-BPP with the assumptions such as bins should not be rotated, bins have to be packed with each edge parallel to the corresponding bin edge and all data should be positive integers. Ambrosino (2006) satisfied some of the main constraints that must be considered are constraints related to the bin structure which focused on the type, size and weight, standard sized containers, Weight constraints, locations of reefer containers, etc. Dubrovsky et al. (2002) use a genetic algorithm for minimising the number of container required for shipment for the given set of bins by including the stability criteria and reduced the search space using a compact and efficient encoding scheme. Imai et al. (2002) developed simplification hypotheses using Linear Programming models and Integer Programming models which make heuristic algorithms unsuitable for entire bin packing applications. Martello and Pisinger (2000) presented the review of relevant literatures and confirms that the evolutionary approach works very well when compared to traditional optimization procedures and the computational time was also reduced significantly. George and Jennifer (1995) given various algorithms given in the literature for box packing are the First Fit Decreasing Algorithm, Best-Fit Decreasing Algorithm, Improved First Fit Algorithm, Wall Building Algorithm, Layer-by-Layer Algorithm, Matrix Methods, Greedy Algorithm, Tree Search Algorithm, Traditional Optimization Algorithms, Stack-Building Algorithm, Guillotine Cutting Algorithms, Cuboids Arrangement Algorithms, etc.

In this paper, an ant colony optimization approach had been utilized to optimize the space utilization inside the container by considering the packing constraints.

III. Bin Packing Problem

This research adopts a standard model for three dimensional bin packing problem, which have n-bins to be packed into m-container of standard size (m = 1) denoted by $n/m/C/V_{max}$, where the parameter 'C' represents the packing rules associated with the bins (their packing order in the container, stability of the bins, etc). The parameter 'V_{max}' is the packed volume of bins inside the container, which indicates the performance measure and should be maximized (maximum occupied volume of all the packed bins including allowances).

The inputs to the problems are the bin details and the container specification along with the constraints, if any are as follows.

1. A finite set of rectangular prismatic bins with its dimensions should be the input to the bin packing problems. $B = \{b_1, b_2, \ldots, b_N\}$, Whereas $b_N = \{L_n, B_n, H_n\}$.
2. Dimension of the rectangular prismatic container can be as per the user requirement. The container considered in this research is open top external container. A sample size of the 20' top open container considered for the experimental purpose is of size $C_N = \{5.80, 2.44, 2.66\}$.

The constraints considered in this research are as follows.

1. The sum of volume of set of packed bins ∈ = B should be less than the container volume Ω1. i.e. $\sum|V(B)| \leq V(M)$.
2. The bins to be unloaded should be kept near to the container door for easy unloading Ω2.
3. The packed bins should be stable enough Ω3.
4. For every bin ∈ *B*, a *freight cost di,j* have to be included by calculating the volume of the bin along with the empty space created around that bin, representing the typical empty space Ω4.
5. As a finite number constraints Ω defined over the packing elements. So some times, in some optimal sequence, the constraint may conflict with each other. Thus priority rule have to be developed to resolve the confliction. i.e. Given a set *S* of all possible sequences <si,sj,…sk,…sn>, it is required that a set $\hat{S} \subseteq S$ exist representing the *feasible sequences* with respect to Ω.

The following is an example of a packed bin matrix P[L].

		C1	C2	C3	C4	C5	C6	
	[	01	05	10	22	23	24	R1
P[L] =		06	02	25	12	10	21	R2
		07	04	11	14	08	16	R3
		03	09	15	13	17	00	R4
		18	19	20	00	00	00]	R5

Whereas, each row {R1, R2,...R5} of the matrix represents a sequence of the bins packing order inside the container for the R^{th} row. i.e. the matrix denotes the packing order of the bins for five rows in a layer L. The vertical elements {C1, C2, ... C6} of the matrix represent the column order inside the container for a layer L. i.e., R1C1 is the first bin to be packed in the origin of the container followed by R1C2 and so on till the boundary of the container along the length-wise. The packed volume of each sequence is given in the matrix V[L].

$$V[L] = \begin{array}{c} L1 \quad L2 \quad L3 \quad L4 \quad L5 \\ \begin{bmatrix} v1 & v6 & v11 & v16 & v21 \\ v2 & v7 & v12 & v17 & v22 \\ v3 & v8 & v13 & v18 & v23 \\ v4 & v9 & v14 & v19 & v24 \\ v5 & v10 & v15 & v20 & v25 \end{bmatrix} \end{array}$$

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & j \in allowed_k \\ 0 & \text{else} \end{cases}$$

Matrices P and V define the bin packing problem (Note that not all entities of P are given here for simplicity). Gehring and Bortfeldt (1997) had the objective of packing all the available bins into the minimum number of containers. The objective of the matrix P is to pack all the available bins in some sequence. The objective of the matrix V is maximizing its value. Furthermore, unlike the most local search techniques and heuristic algorithms, which are problem specific, the Ant tuning based strategy applied is more flexible as it does not alter the ant colony algorithm, but rather analyzed the information gathered by the ants to improve on the current solution.

IV. Ant Colony Algorithm

The basic idea of the ant colony algorithm is to mimic the behaviour of the real ants in identifying the shortest path from its nest to the food source. So, a population or colonies of *x* artificial ants have been generated to identify the optimal bin packing sequence. Those ants iteratively build a solution by continually applying a probabilistic decision policy *i* times until a solution has been found. Ants that found a good solution, mark their path in the network by depositing some amount of pheromone on the path. Ants of the next iteration/generation have been obviously attracted towards the higher pheromone path, resulting in a higher probability to follow the already traversed good paths. Thereby the best optimal solution had been retained. In addition to it and to escape from the local maxima, the pheromone deposited by the ants will allow evaporating with time by storing the best path in the memory (tabu list).

As indicated above, in order to apply the Ant colony algorithm, a graphical representation *G*, of the bin packing problem must first be constructed. The nodes in the network should be the bins. All the nodes have to be inter-connected with the adjacent nodes by edges. The sample generated ant colony network is given in the Figure 1.

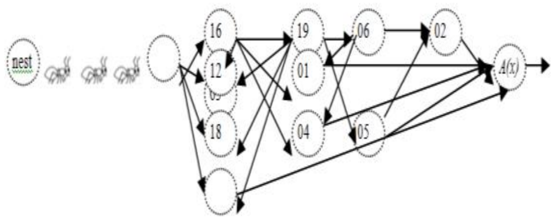

Figure 1 : Ant Colony Network

The meta-heuristic begins by initializing the amount of pheromone along each edge of *G* to some positive real value. In this research, it has been set to unit value. Each ant is then designated with a starting position from the nest to any of the nodes. The initially anta are allowed to select its positions and path along the edges randomly. Upon completion of the initialization phase, each ant has to be allowed independently to construct a solution/sequence, employing that an ant should not be allowed to visit the nodes twice in a path. After identification of valid solutions, feasibility and the fitness function value have to be calculated and normalized to identify the best optimal solution. The processes have to be continued until a complete solution has been found. The pheromone amount along each edge (i, j) is recomputed accordingly from the Equation 1.

$$\tau_{ij}(t+1) = \Delta\tau_{ij} + (1-\rho)\,\tau_{ij}(t) \qquad (1)$$

Whereas, 'ρ' denotes the evaporation rate. 'τ' & 't' represent the pheromone value and trails respectively. 'Δτ' represents the proportional change. Once it has been calculated for the edges in the network, the fourth coming ants has to follow the path having higher pheromone value. The ant moving rule can be identified based on the Equation 2.

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & j \in allowed_k \\ 0 & \text{else} \end{cases} \qquad (2)$$

The parameters α, β are user defined parameters that determine the degree to which the pheromone is used and helps the ant in deciding where to move. Setting β = 0 will result in only the pheromone information being used. The global updation rule is given in the Equation 3.

$$\tau_{ijk}(t+1) = {}_{\tau min}\int^{\tau max} \{\Delta\, \tau_{ijk\,(best)} + (1-\rho)\, \tau_{ijk}(t)\} \qquad (3)$$

whereas τ_{max} and τ_{min} are upper and lower bounds imposed on the pheromone respectively and Δ_{best} belongs a tour which yields better fitness function. Furthermore, the definition of a given problem may impose some constraints on the sequence of nodes resulting in valid solutions/sequence. In this research, if a path is found to be not a valid sequence, its probability of selection would be set to negative value. This implies that nodes with negative pheromone edges, and having very low probability of being selected.

Apart from identification of feasible and optimal solutions, Ant colony algorithm incorporates *pheromone evaporation* and optional *daemon activities*. Pheromone evaporation causes the amount of pheromone on each edge of *G* to evaporate over time and it prevents premature convergence to a local optimal solution. Thus the bad solution had been eliminated from the search and favours the exploration of the search space further. Daemon activities can be any action such as a local search technique over the solutions of ants.

V. Hybrid Ant Colony Optimization For BPP

The general idea is to adapt a basic Ant colony algorithm and investigate further exploration of the solution space. The exploration technique, termed Ant tuning strategy (ATS) has been provided to fill the left out empty space inside the container. The key feature of the Ant tuning strategy is that it does not alter the ant colony algorithm, but rather enhance the information gathered by the ants further by including more bins and sequence of packing. i.e. another local search technique based on probabilistic rules to further improve solutions which already found by the ants. Unlike most heuristic techniques which geared towards problem specific, the developed heuristic can be applied to packing, cutting, stacking problems, etc.

The ants are allowed to pass through the network and ants are free to select its path and the network has been framed bidirectional. The ant colony output has to be decoded to the packing sequence. The sample sequence of fifty bins obtained from an ant trial is given in the Figure 2.

B91,B39,B81,B66,B32,B41,B76,B98,B71,B67,B44,B03,B55,B32,B36,B66,B58,B45,B50,B17,B01,B33,B94,B36,B80,B45,B45,B11,B49,B58,B41,B11,B15,B33,B38,B78,B43,B82,B21,B06,B05,B88,B41,B07,B87,B21,B95,B85,B10,B12,

Figure 2: Sequence of bins from an Ant trial

The sequence obtained from the ant trial has to be decoded to the real bin data. The decoded bin data for the sequence given in the Figure 2 is given in the Figure 3. Each bin has been represented by its length, width and height respectively. The length represents along the container X axis and the width represents along the container Y axis.

15,10,05; -10,10,05; -20,10,10; -15,20,05; -10,10,10; -10,10,10; -10,10,10; -15,10,05; -10,10,05; -20,10,10; -15,20,05; -10,10,10; -15,15,05; -10,20,10; -10,10,05; -15,20,05; -15,10,05; -20,10,10; -10,10,05; -15,10,05; -20,10,10; -15,20,05; -15,10,05; -15,10,05; -10,10,05; -10,10,10; -20,10,10; -15,20,05; -10,10,05; -15,10,05; -10,10,10; -20,10,10; -10,10,05; -15,10,05; -15,20,05; -10,10,10; -15,10,05; -15,10,05; -10,10,05; -20,10,10; -10,10,10; -10,10,10; -20,10,10; -10,10,05; -15,10,05; -10,10,10; -10,10,10; -20,10,10; -10,10,10; -20,10,10;

Figure 3: Decoded Bin Sequence

Graphically, the packing pattern has been developed and the empty space formations are calculated for each layer of packed bins inside the container. In this research, a virtual Cartesian coordinate system had formed and the bins are packed on it to identify the position and location of the bins and the empty spaces. The graphical representation of the bins inside the container for the sequence given in Figure 2 for first layer is shown in the Figure 4.

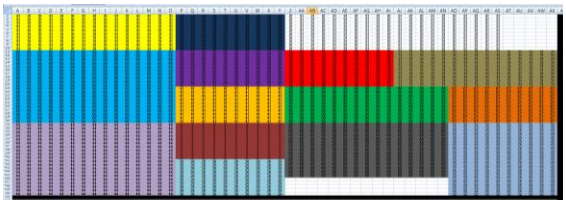

Figure 4: Graphical Bin Packing Pattern for a Sequence

Once the empty spaces had identified, then Ant tuning algorithm have to be applied to further fill the empty space formed inside the container. Ant tuning algorithm is also inspired by the ants behaviour in identifying the path between the obstacles. In real-life, ants can modify their paths easily, if it found an obstacle occurs in their path. In this research, the packed bin nodes are considered to be the obstacles and the path will be blocked. So the ants are forced to select its next nearby path around the obstacle. In this trial, ants are allowed to jump the packed bin nodes to the next layer, as the real ants are climbing over the obstacles. Unlike the pervious trial, in the recursion trial, the fitness value has been calculated node by node and the maximum fitness value for each layer inside the container will be calculated before the trial starts i.e. the empty space inside the container layer by layer. The termination condition for the recursive trial is that the actual fitness value is more or less equal to the calculated fitness value, then the ant has to terminate its trial and it might not be necessary for the ants to pass through the entire nodes in the network. The goal of ant tuning algorithm is to cause fluctuations or alteration in the environment in which the ants exist along with good solutions which already discovered and to identify further good solutions based on requirement by the ants behaviour. As this recursive trails took more computational time, the ant tuning algorithm can be applied only after or near stagnation. i.e. after the good solution had been identified by the colony of ants. Since ants would have already constructed a reasonable solution, any fluctuation will cause a local search around the current solution. Thus, the radius of local search from the already good solution should be determined by conducting sensitivity analysis as it is vary from application to application. A new layer of pheromone have been lay over the edges and the fourth coming ants will follow the path as described above. This process will continue until the empty space in all the layers filled by the remaining empty space. In each and every iteration, the numbers of obstacles are increasing and the pheromones of the path have to be revised and updated every iteration. The output obtained from the ant tuning algorithm is shown in the Figure 5.

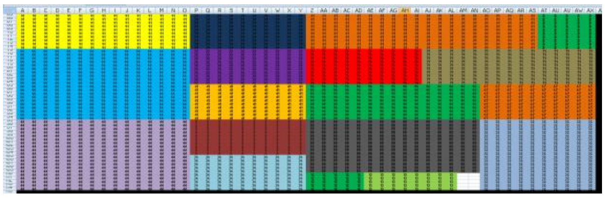

Figure 5: Graphical Pattern from Ant Tuning Algorithm

Form the Figure 5, it clearly proved that the ant tuning algorithm fills more empty space left out by the ant colony algorithm.

VI. Results And Discussion

The proposed ant based optimization approach had tested using well-known BPP instances found at OR-Library [13]. The results were compared to the best-known published results (based on various meta-heuristics). The module was developed in Visual basic programming language and the simulations were performed in Pentium IV processors, 1 GB RAM running in Microsoft Windows 7. The ant colony network was tested with various parameter sets and by conducting sensitivity analysis the parameters have been finalized. Theses parameters were used for both the ant colony optimization and ant tuning optimization. The parameters used for the test runs are: α =0.85, β= 0.10, ρ= 0.90. The number of ants in each generation was m = number of bins, number of iterations = 5000. The initial pheromone along each edge was set to unit value. The parameters α and β are proportional to exploration, whereas ρ value is proportional to convergence rate. On average, 90 % of the available bins have been packed by the Ant colony algorithm and the Ant tuning algorithm packs on average of 4 %. Sorting of the bins based on the unloading location and size before applying Ants colony algorithm reduces the computational time in checking the placement constraint and stability constraint respectively. A sample output is given in the Table 1. From the table 1, user can understand the position and the location of the bins to be packed into the container and the sequence of loading the bins in the Cartesian coordinate system. At every stage, the packed volume and the empty space have been calculated which helps the user to plan for the spacers and the position of the spacers. The freight rate for packing the bins also calculated for every bin placement and the cost for the spacers also added with the freight rate. So the logistic industries can know all the required details before packing the bins into the containers.

Table 1: Sample user Understandable Output

Bin No	L	W	H	X	Y	Z	Spacers	Packed Volume	Empty Space	Constraint	Freight Rate
B91	15	10	05	0	0	0	-	750	9250	-	0.75
B39	10	10	05	15	0	0	-	1250	8750	-	0.50
B81	20	10	10	25	0	0	45,0,0-5	3250	6750	-	2.0+5.0
B66	15	20	05	0	10	0	-	4750	5250	-	1.50
B32	10	10	10	10	10	0	-	5750	4250	-	0.10

VII. Conclusion

This paper presented an ant colony optimization algorithm approach to the bin packing problem. The main goal is to optimize the container volume utilization by applying pheromone updating techniques and their effects on solution space exploration. A rudimentary Ant colony algorithm has been used to identify the best bin packing pattern by satisfying the packing constraints and the recursive application of Ant tuning algorithm had enhanced the utilization of the container volume. Ant tuning algorithm was proposed and proved that the heuristic improves the solution quality of the ant colony algorithm. Considering the packed bins as the obstacle, cause a higher degree of exploration for ants to search. Experimental results show that the proposed approach gave competitive packing pattern length compared to the benchmark solutions for the problem instances tested.

VIII. Future Scope

Some more behaviours of the ant trial such as blowing off the pheromone due to wind or by external force, instead of searching in the same solution space again, a sub-Path can constructed that holds all valid nodes and edges to propagate, in recursive searching too many steps occur, making it difficult for the ants to discover small improvements in already obtained good solution and requires heuristics to reduce the searching.

References

[1] M.Dorigo and G.Di Caro, "The ant colony optimization meta-heuristic," in New Ideas in Optimization, edited by D. Corne, M.Dorigo, and F. Glover, McGraw-Hill, pp. 11-32, 1999.

[2] M.R.Garey, and D.S, "Computers and Intractability, A Guide to the Theory of NP-Completeness", W.H Freeman and Company, 1979.

[3] S. Goss, S.Aron, J. L. Deneubourg, and J. M. Pasteels.Self-organized shortcuts in the Argentine Ant.Naturwissenschaften, 76:579-581, 1990

[4] Dorigo M., G. Di Caro & L. M. Gambardella. "Ant Algorithms for Discrete Optimization", Artificial Life, 5(2):137-172, 1999.

[5] Bullnheimer B., R. F. Hartl & C. Strauss, "A New Rank Based Version of the Ant System: A Computational Study", Central European Journal for Operations Research and Economics, 7(1):25-38, 1999.

[6] Martello, S., Pisinger, D., Vigo, D., 2000. The three-dimensional bin packing problem. Operations Research 48 (2), 256–267.

[7] Ambrosino, D., Sciomachen, A., Tanfani, E., 2006. A Bay Assignment Algorithm for the containership stowage problem. Journal of Heuristics 12 (3), 211–233.

[8] Dubrovsky, O., Levitin, G., Penn, M., 2002. A genetic algorithm with compact solution encoding for the container ship stowage problem. Journal of Heuristics 8, 585–599.

[9] Imai, A., Nishimura, E., Papadimitriu, S., Sasaki, K., 2002. The containership loading problem. International Journal of Maritime Economics 4, 126–148.

[10] S. Martello, D. Pisinger, (2000). "The Three Dimensional Bin Packing Problem", European Journal of Operation research 48, 256-267.

[11] John A George and Jennifer M George, (1995). "Packing Different Sized Circles into a Rectangular Container", European Journal of Operation Research 84, 693-712.

[12] H.Gehring, A. Bortfeldt, (1997). "A Genetic Algorithm for Solving the Container Loading Problem", International transactions in Operation Research, 44, pages 401 – 418.

[13] OR Library, URL http://mscmga.ms.ic.ac.uk