An Ant Colony Optimization Heuristic for Solving the Two-Dimensional Level Packing Problems

AzzaGaber, RaafatElshaer† and MahassenKhater
Industrial Engineering Department
Faculty of Engineering, Zagazig University
Zagazig, Egypt
engazza10@zu.edu.eg, r_h_elshaer@zu.edu.eg, mahassenkhater@yahoo.com

Abstract

The two-dimensional packing problem (2PP) is one of the main problems encountered in many industries. Proper nested pattern layout can minimize the trim loss and maximize the utilization of the material available. This paper presents a hybrid ant colony algorithm coupled with a simple local search algorithm to solve the two-dimensional bin packing (2BP) and strip packing (2SP) problems with additional constraint, where the items to be packed by levels. The performance of the proposed algorithm is tested over a number of standard benchmark instances from the literature. Computational results indicate that the proposed algorithm is effective for solving these problems.

Keywords
Two-dimensional level packing, Bin packing, Strip packing, Ant colony optimization.

1. Introduction

For several industries, material saving is one of the most important factors to be considered, and it is well-known that a well nested pattern layout can result in a saving in the resource material. The two dimensional packing problem is a NP-complete combinatorial optimization problem (Fowler et al., 1981). This problem occurs in different real world applications such as glass, paper, cloth industries, cutting rectangular components from large sheet of material, placing goods on shelves in warehouse, arranging articles and advertisements in pages in newspaper paging.

There are two main specific problems of the two-dimensional packing problem considered in the literature (Lodi et al., 2004), bin packing and strip packing. In the two-dimensional bin packing problem (2BP), there is a set of rectangular items to be packed in an infinite number of identical rectangular containers (bins), these containers having width W and height H, and the objective is to minimize the number of bins used. In the two-dimensional strip packing problem (2SP), there is a set of rectangular items to be packed in a single container (strip), having width W and infinite height, and the objective is to minimize the height to which the strip is used. The 2BP is most suitable for the wood, glass, metal, and semiconductor industries, while the 2SP will generally apply to the paper and garment industries (Mohammed et al., 2009).

Exact solution methods, such as branch and bound algorithm (Cui et al., 2008) and column generation algorithm (Gilmore et al., 1965), can only be used for small problem instances. Martello et al. (1998, 2003) propose two exact solution approaches for 2BP and 2SP, respectively. For real world problems, heuristic solution methods have to be used, where these heuristic algorithms solve specific problem according to a set of rules, such as Next-Fit Decreasing Height (NFDH) algorithm, First-Fit Decreasing Height (FFDH) algorithm, and Best-Fit Decreasing Height (BFDH) algorithm (Lodi et al., 2002a). In the last decades, researchers have shown a lot of interest for metaheuristics approaches such as Genetic Algorithm (Bennell et al., 2013), Simulated Annealing (Leung et al., 2012), and Tabu Search (Alvarez-valdes et al., 2007) to solve the problem for a near-optimal solution. Hopper and Turton (2000, 2001) review the approaches developed to solve 2D packing problems using Genetic Algorithms,

†Corresponding author

simulated Annealing, Tabu Search, and Artificial Neural Networks. Recently, Jain and Singh (2013) review the metaheuristic approaches for solving rectangle packing problems.

Ant Colony Optimization (ACO) is a metaheuristic approach, which has been widely applied for solving different combinatorial optimization problems such as travelling salesman problem (Dorigo and Gambardella, 1997), job shop and flow shop scheduling (Yagmahan and Yenisey, 2008), and cell formation (Spiliopoulos and Sofianopoulou, 2008). Recently, A comprehensive review of different implementations of ant algorithms for packing problems can be found in (Singh et al., 2016), where few of these implementations are applied for the two-dimensional level packing (2LP) problem. So, the aim of this paper is to present an ant colony algorithm with local search for the two-dimensional level bin (2LBP) and strip packing (2LSP) problems. As stated in (Dorigo and Stutzle, 2003), ACO and local search can work as a complementary partnership.

The remainder of this paper is organized as follows. Section 2 describes the definition and formulation of the two level packing problems, 2LBP and 2LSP. Section 3 presents the proposed ant algorithm for solving the formulated problems. Computational results are given in Section 4. Section 5 concludes the paper and provides suggestions for future work.

2. Problem Definition and Formulation

The two-dimensional bin packing problem (2BP) aims to allocate a set of n rectangular items $j \in J = \{1, \dots, n\}$, each having width w_j and height h_j, and an unlimited number of finite identical rectangular bins, having width W and height H. The problem is to allocate, without overlapping, all the items to the minimum number of bins, with their edges parallel to those of the bins. It is assumed that the items have fixed orientation, i.e., they cannot be rotated. All input data are assumed to be positive integers, and $w_j \leq W$ and $h_j \leq H$ $(\forall j \in J)$. Most of the approximation algorithms for 2BP pack the items in rows forming levels. The first level is the bottom of the bin, and items are packed with their base on it. The next level is determined by the horizontal line drawn on the top of the tallest item packed on the level below, and so on. There are some assumptions mentioned in (Lodi et al., 2002b) for the level packing problems: In each level, the left most item is the tallest one, in each bin, bottom most level is the tallest level, and items are sorted and renumbered in non-increasing order of heights. Lodi et al. formulate the 2LBP problem, by assuming that there are n potential levels (the ith one associated with item i initializing it), and n potential bins (the kth one associated with potential level k initializing it), as follows:

$$\text{(2LBP):} \quad \min \sum_{k=1}^{n} q_k \tag{1}$$

Subject to:

$$\sum_{i=1}^{j-1} x_{ij} + y_j = 1 \qquad (j = 1, \dots, n) \tag{2}$$

$$\sum_{j=i+1}^{n} w_j x_{ij} \leq (W - w_i) y_i \qquad (i = 1, \dots, n-1) \tag{3}$$

$$\sum_{k=1}^{i-1} Z_{ki} + q_i = y_i \qquad (i = 1, \dots, n) \tag{4}$$

$$\sum_{i=k+1}^{n} h_i z_{ki} \leq (H - h_k) q_k \qquad (k = 1, \dots, n-1) \tag{5}$$

$$y_i, x_{ij}, q_k, z_{ki} \in \{0,1\} \forall i, j, k \tag{6}$$

The objective function (1) minimizes the number of bins used. Where, q_k $(k \in J)$ is a binary variable taking the value 1 if item k initializes bin k, and the value 0 otherwise. Constraint (2) imposes that each item is packed exactly once, either by initializing a level or in a level initialized by a preceding (taller) item. Where y_i $(i \in J)$ is a binary variable taking the value 1 if item i initializes level i, and the value 0 otherwise and $x_{ij} = 1$ if item j goes to the level initialized by item i and 0 otherwise. Constraint (3) imposes the width constraint to each used level. Constraint (4) imposes that each used level is allocated exactly once, either by initializing a bin or in a bin initialized by a preceding (taller) level. Where z_{ki} $(i \in J)$ is a binary variable taking the value 1 if item i initializes a

level in bin k. Constraint (5) imposes the height constraint to each used bin. Constraint (6) represents binary decision variables.

By modifying the objective function and eliminating all constraints (and variables) related to the packing of the levels into the bins, this will give the model for the 2LSP problem:

$$(2LSP):\ \min \sum_{i=1}^{n} h_i\, y_i \tag{7}$$

Subject to:

Constraints(2), (3),

$$y_i, x_{ij} \in \{0,1\} \forall i, j \tag{8}$$

The objective function (7) minimizes the height to which the strip is used. Constraint (8) represents binary decision variables.

3. Ant colony Optimization for solving the 2LP problem

ACO was introduced (Dorigo et al., 1996) as a nature-inspired metaheuristic. There are different ACO algorithms such as Ant System (Dorigo et al, 1996), Ant Colony System (Dorigo and Gambardella, 1997), and MAX-MIN Ant System (Stutzle and Hoos, 2000). The inspiring source of ACO is the foraging behavior of real ants. The characteristic of real ant colonies is exploited in artificial ant colonies in order to solve combinatorial and continuous optimization problems. The basic steps of the ACO algorithm are as shown in Figure 1. After initialization, the ACO algorithm iterates over three phases. At each iteration, a number of solutions are constructed by the ants. These solutions are then improved using a local search method. Finally, the pheromone trails are modified in order to bias ants in future iterations to construct solutions similar to the best ones previously constructed.

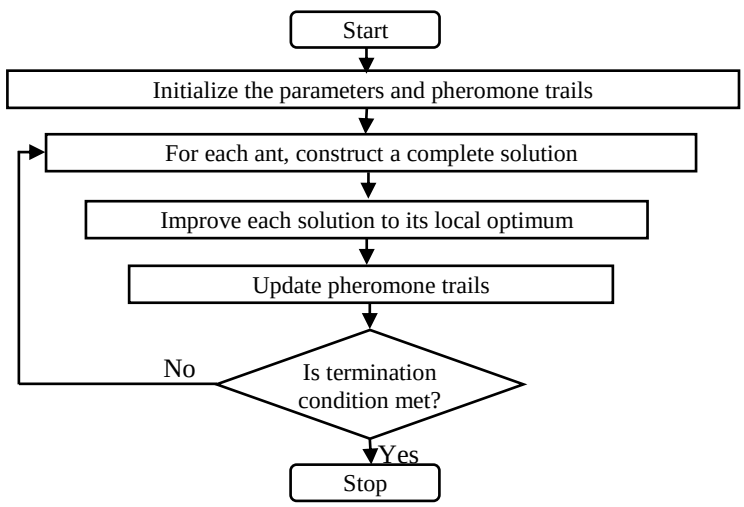

Figure 1. The general steps followed in the ACO algorithms (Pham et al., 2009)

3.1 Proposed ant algorithm

This section describes the key elements of the proposed ant algorithm.

3.1.1 Construction of solutions

The proposed ant algorithm (for 2LBP and 2LSP) works in an iterative way. At each iteration, a number m of artificial ants are considered. Each ant will fill the levels one by one, where each ant will start each level with a randomly selected item, then it will add the items one by one to its level, until none of the items fit in the level. Then the level is closed, based on the tallest item, and a new one is started. In order to find all feasible solutions, the allocated items are added to a $tabu_a$ list for ant a. This list saves the indices of the items already allocated and forbids the ant to allocate these items again before a complete solution has been constructed.

3.1.2 Pheromone trail definition

The definition of the meaning of the pheromone trail is the main factor that affects on the quality of the ACO algorithm (Dorigo and Stutzle, 2003). This definition should conform the nature of the problem. So, the pheromone trial for the 2LBP and the 2LSP problems is defined in a similar way to Costa and Hertz (1997): τ_{ij} encodes the favorability of having an item i and an item j in the same level. The pheromone matrix is symmetric ($\tau_{ij} = \tau_{ji}$) and square (has n rows and n columns where n is the number of items to be packed).

3.1.3 Heuristic information definition

The heuristic information, denoted by η_{lj} and also called visibility of item j in level l, is a simple heuristic guiding the ant to make a decision (Levine and Ducatelle, 2004). For the 2LBP and the 2LSP problems, the visibility is calculated based on (Burke et al., 1999) Eq. (9):

$$\eta_{lj} = \begin{cases} \frac{Total\_area_{lj}}{Best\_placement\_area_{lj}} & if\ l \neq \phi \\ 1 & otherwise \end{cases} \quad \forall\, j \in J_a(s,l) \tag{9}$$

where $Total\_area_{lj} = \sum_{i \in l \cup \{j\}} h_i . w_i$, the combined area of all items in level l plus the area of item j, $Best\_placement\_area_{lj} = \sum_{i \in l \cup \{j\}} w_i \,.\, \max_{i \in l \cup \{j\}} \mathrm{h}_i$, the best placement area of the same items using the No Fit Polygon, $J_a(s,l)$ is the set of prospective (candidate) items that are still left after partial solution s is formed, don't appear in the $tabu_a$, and are fit in the current level l by ant a. Visibility is defined as how items i, just placed in the level l, fit with the item j about to be placed as shown in Figure 2. This returns a value between 0 and 1.

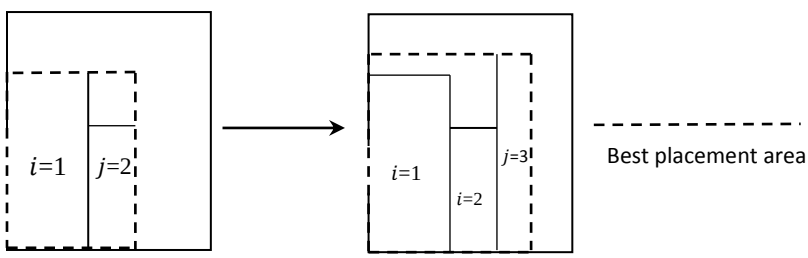

Figure 2. Best placement area for items ***i*** (just placed) and item ***j*** (to be placed)

3.1.4 Selection probability

An ant a chooses the next item j for its current level l by applying the rule given by Dorigo and Gambardella (1997):

$$j = \begin{cases} \arg\max_{j \in J_{a(s,l)}} \left\{ [\bar{\tau}_{lj}] . [\eta_{lj}]^{\beta} \right\} & if q \leq q_0 \quad \text{(exploitation)} \\ j^* & otherwise \quad \text{(exploration)} \end{cases} \tag{10}$$

Where $\bar{\tau}_{lj}$ is the pheromone value for piece j in level l, The pheromone value $\bar{\tau}_{lj}$ for an item j in a level l is given in Eq. (12). It is the sum of all the pheromone values between item j and the items i that are already in level l, divided by the number of items in l. If l is empty, $\bar{\tau}_{lj}$ is set to 1. η_{lj} is the heuristic information guiding the ant, and β is a parameter which determines the relative importance of pheromone information versus heuristic information ($\beta > 0$). q is a random number uniformly distributed in $[0,1]$, q_0 is a parameter ($0 < q_0 < 1$), determines the relative importance of $exploitation$ versus $exploration$, and j^* is a random variable selected according to the probability distribution given in Eq. (11).

$$p_a(s,l,j) = \begin{cases} \frac{[\bar{\tau}_{lj}] . [\eta_{lj}]^{\beta}}{\sum_{g \in J_{a(s,l)}} [\bar{\tau}_{lg}] . [\eta_{lg}]^{\beta}} & if\ j \epsilon J_a(s,l) \\ 0 & otherwise \end{cases} \tag{11}$$

$$\bar{\tau}_{lj} = \begin{cases} \frac{\sum_{i \varepsilon l} \tau_{ij}}{|l|} & if\ l \neq \phi \\ 1 & otherwise \end{cases} \tag{12}$$

3.1.5 Updating the pheromone trail

The aim of the pheromone update is to increase the pheromone values associated with good or promising solutions, and to decrease those that are associated with bad ones (Dorigo et al., 2006). This paper uses ACS (Dorigo and Gambardella, 1997) to update the pheromone trail where there are two main aspects for updating, local and global. The local updating (online update) applied while ants construct the solution according to the following rule:

$$\tau_{ij} = (1-\varphi)\,.\,\tau_{ij} + \varphi\,.\,\tau_0 \tag{13}$$

where $0 < \varphi < 1$ is the local pheromone decay coefficient, and τ_0 is the initial value of the pheromone, can be set to a small and positive arbitrary value. The main goal of the local update is to diversify the search performed by subsequent ants during any iteration. The global updating (offline update) applied at the end of each iteration by only the best ant according to the following rule:

$$\tau_{ij} = (1-\rho).\tau_{ij} + \rho.f(s^{best}) \tag{14}$$

where $0 < \rho < 1$ is the global pheromone decay coefficient, $f(s^{best})$ is the objective value of s^{best} in the current iteration.

3.1.6 Fitness function

Evaluating results of the solutions is very important to guide the algorithm towards good solutions. So there is a need to the fitness function. For the 2LBP problem, the objective value of a solution s is defined as the average filling rate for each bin used in this solution. The function proposed by Falkenauer and Delchambre (1992) is suitable to define the fitness of a solution s:

$$f(s) = \frac{\sum_{i=1}^{N}(F_i\,/W * H)^{\gamma}}{N} \tag{15}$$

where N is the number of bins used, F_i is the sum of areas of the items in the bin i, γ is a constant, $\gamma > 1$, expresses the concentration on the well-filled 'elite' bins in comparison to the less filled ones. If $\gamma = 1$, only the total number of bins used would matter. Falkenauer and Delchambre (1992) reported that a value of 2 gives good results. However, for the 2LSP problem, the objective value of a solution s is defined as the strip height needed to build the corresponding packing pattern. Therefore this paper uses the following objective function proposed by Salto et al. (2008):

$$f(s) = 1 - \frac{area\_waste}{strip\_height * W} \tag{16}$$

Where $strip\_height$ is the length of the packing pattern corresponding to the permutation s and $area\_waste$ is the area of reusable trim loss in the last level of the packing pattern.

3.1.7 Adding Local Search

The performance of ACO algorithms can be highly improved when coupled with local or neighborhood search algorithms. ACO creates a population of solutions, and then these solutions are improved via local search. The proposed local search routine selects two items randomly and exchanges their positions for each newly generated solution as shown in figure 3. If a better solution is found, it replaces the current solution. The improved solutions are then used to update the pheromone trail. This process is repeated until no further improvement is possible.

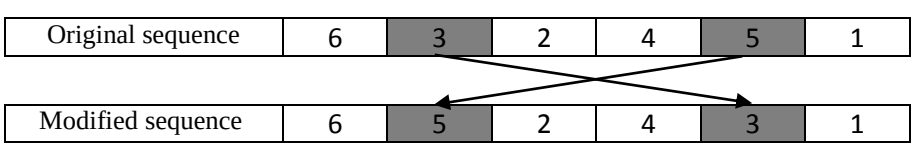

Figure 3. Swapping two randomly selected items

3.1.8 Termination condition

The proposed algorithm stops if the solution is equal to the lower bound or the last max_iter is satisfied, where max_iter is a user-specified parameter and the two lower bounds (Eqs. 17 and 18) proposed by Lodi et al. (2004) are used, where L_{cut}^{s} is the lower bound for the 2LSP problem using CUT-S algorithm, H_i is the height of level i and l is the resulting levels (with $H_i \geq H_{i+1}\ \forall i \in l$). The CUT-S sorts the items in non-increasing h_j values. Initialize the first level at height h_1, consecutively pack into it items $1, 2, \ldots$, until the first item i is found which doesn't fit. Split item i into two slices: one having width $\delta = W - \sum_{j=1}^{i-1} w_j$, the other having $w_i - \delta$ width. Pack the first slice

(possibly null, if $\delta = 0$) into the first level, and initialize the next level at height h_i by packing the second slice into it. Proceed in the same way until all items are packed. L^b_{cut} is the lower bound for the 2LBP problem using CUT-B algorithm. The CUT-B algorithm has two steps; in the first step execute the previous algorithm CUT-S. In the second step pack levels 1, 2,…, in the first bin, until the first level l is found which does not fit. Horizontally split level l (i.e., the items and slices packed into it) into two sectors: one having height $\delta = H - \sum_{i=1}^{l} H_i$, the other having height $H_l - \delta$. Pack the first sector (possibly null, if δ = 0) into the first bin, and initialize the next bin with the second sector. Proceed in the same way until all levels are packed.

$$for\ the\ 2LSP:\ \ L^s_{cut} = \sum_{i=1}^{l} H_i \tag{17}$$

$$for\ the\ 2LBP: L^b_{cut} = \left\lceil \frac{L^s_{cut}}{H} \right\rceil \tag{18}$$

3.2 The complete ant algorithm

The proposed ant algorithm (Ant2DLP) can be summarized as shown in Figure 4.

Step 1: Initialize values of the parameters $m, q_0, \beta, \rho, \varphi$ and max_iter
Initialize values of the pheromone trails τ_{ij} between each two items (i, j).
Step 2:
Repeat
For each ant $\{a = 1, 2, \dots, m\}$ do,
Repeat
Initialize a tabu list of ant a,tabu$_a$, to empty
Open an empty strip {for the 2LSP problem}
Repeat
Open an empty bin {for the 2LBP problem}
Repeat
Open an empty level l
Choose the first item i in the level randomly from the candidate items
Put the chosen item in tabu list
Choose the next item j according to the calculated probabilities in Eq$s.$ (10)and (11)
Put the chosen item in the tabu list
Update the pheromone trail using Eq. (13) {local update }
until no remaining items fits in the level
until no remaining items fits in the bin {for the 2LBP problem}
until all items are packed
Step 3: Calculate the fitness function value for that solution using Eq. (15) for the 2LBP and Eq. (16) for the 2LSP
Step 4: Improve that solution using the local search described in section(3.1.7)
End for
Step 5: Find the iteration best solution
Step 6: Update the pheromone trail using Eq. (14){ global update }
Step 7: Stop, if the solution is equal to the lower bound.
Until the maximum number of iterations, max_iter, is reached.

Figure 4.A pseudo-code description of the proposed ant algorithm (Ant2DLP)

Table 1: The distributions used to generate instances of each class

Class	$W \times H$	$w_i \times h_i$	Proposed by
1	10X10	$Uniform[1, 10]$	Berkey and Wang, 1987
2	30X30		
3	40X40	$Uniform[1, 35]$	
4	100X100		
5		$Uniform[1, 100]$	
6	300X300		
7	100X100	*70%Type1, 10%for each Type2,3,4	Martello and Vigo, 1998
8		*70%Type2, 10%for each Type1,3,4	
9		*70%Type3, 10%for each Type1,2,4	
10		*70%Type4, 10%for each Type1,2,3	

*Type 1: w_i uniformly random in $[\frac{2}{3}W, W]$, $h_i$uniformly random in $[1, \frac{1}{2}H]$,
Type 2: w_i uniformly random in $[1, \frac{1}{2}W]$, $h_i$uniformly random in $[\frac{2}{3}H, H]$,
Type 3: w_i uniformly random in $[1, \frac{1}{2}W]$, $h_i$uniformly random in $\left[\frac{1}{2}H, H\right]$,
Type 4: w_i uniformly random in $[1, \frac{1}{2}W]$, $h_i$uniformly random in $[1, \frac{1}{2}H]$,

Table 2. Computational results of the 2LBP and 2LSP problems

Class	W×H	n	2LBP			2LSP		
			$\bar{z}$(CPLEX)	$\bar{z}$(Ant2DLP)	%deviation	$\bar{z}$(LINGO)	$\bar{z}$(Ant2DLP)	%deviation
1	10X10	20	7.3	7.3	0.00	65.8	66	0.30
		40	13.8	14	1.45	127.6	128.8	0.94
		60	20.3	21	3.45	194.7	198.6	2.00
		80	27.7	28.9	4.33	268.4	274.4	2.24
		100	32.4	34.6	6.79	314.2	323.5	2.96
2	30X30	20	1	1	0.00	24.1	24.4	1.24
		40	2	2	0.00	44	44.4	0.91
		60	2.8	2.8	0.00	65.1	65.4	0.46
		80	3.3	3.3	0.00	88.7	88.7	0.00
		100	4.1	4.1	0.00	105.5	105.5	0.00
3	40X40	20	5.4	5.4	0.00	180.8	181.3	0.28
		40	9.8	10	2.04	352.8	361.5	2.47
		60	14	15.3	9.29	533.1	556.6	4.41
		80	19.7	21.8	10.66	741.1	788.6	6.41
		100	22.8	25.8	13.16	873.6	933.5	6.86
4	100X100	20	1	1	0.00	77.1	78.6	1.95
		40	2	2	0.00	141	146.8	4.11
		60	2.7	2.7	0.00	212.2	220.1	3.72
		80	3.4	3.4	0.00	288.2	299.4	3.89
		100	4.2	4.2	0.00	345.1	359.6	4.20
5	100X100	20	6.7	6.7	0.00	568	568.9	0.16
		40	12.3	12.6	2.44	1116.4	1135.6	1.72
		60	18.3	19.3	5.46	1699.7	1750.7	3.00
		80	25	26.8	7.20	2349.4	2442.8	3.98
		100	28.8	31.8	10.42	2793.6	2891.2	3.49
6	300X300	20	1	1	0.00	206.5	213.2	3.24
		40	1.9	1.9	0.00	375.4	385.3	2.64
		60	2.3	2.5	8.70	561.7	583.7	3.92
		80	3	3	0.00	762.8	803.1	5.28
		100	3.7	3.7	0.00	910.7	965.6	6.03
7	100X100	20	5.7	5.7	0.00	520.7	520.7	0.00
		40	11.5	11.7	1.74	1103.6	1108.9	0.48
		60	16.2	16.8	3.70	1585.4	1589.9	0.28
		80	23.3	24	3.00	2271.5	2283.9	0.55
		100	27.6	29	5.07	2703	2735.1	1.19
8	100X100	20	6.1	6.1	0.00	498.5	502	0.70
		40	11.5	11.7	1.74	996.3	1021.1	2.49
		60	16.4	16.8	2.44	1453.8	1482.5	1.97
		80	22.7	23.3	2.64	2019.9	2101.8	4.05
		100	28.2	29	2.84	2464.7	2582.6	4.78
9	100X100	20	14.3	14.3	0.00	1113.7	1117.1	0.31
		40	27.8	27.8	0.00	2202.9	2204.2	0.06
		60	43.7	43.7	0.00	3423.6	3431.6	0.23
		80	57.7	57.7	0.00	4591.3	4604	0.28
		100	69.5	69.5	0.00	5440.4	5475.7	0.65
10	100X100	20	4.5	4.5	0.00	384	386.7	0.70
		40	7.7	8.1	5.19	709.8	724	2.00
		60	10.5	11.3	7.62	988.4	1028.7	4.08
		80	13.5	14.6	8.15	1270.6	1338.9	5.38
		100	16.4	18	9.76	65.8	66	0.30
		Average	14.75	15.27	3.52	1074.82	1098.23	2.35

4. Computational results

The proposed ant algorithm was coded using Visual C# and run on a Pentium IV computer with 2.13 GHz processor and 2 GB of memory. The algorithm contains six parameters $m,\ q_0, \beta, \rho, \varphi$ and max_iter whose values affect its

performance. A number of experiments were conducted to find appropriate parameter combinations for the test problems. Based on these experiments, the parameters were set to $m = 10$ ants, $q_0 = 0.9$, $\beta = 2$, $\rho = 0.1$, $\varphi = 0.8$, and $max\_iter = 1000$.

To evaluate the performance of the proposed algorithm (Ant2DLP), it was tested on a set of standard problem instances. These instances are factorized in ten classes of processes, generated at random. The first six classes have been proposed by (Berkey and Wang, 1987) while the last four ones have been proposed by (Martello and Vigo, 1998). Each class is composed of five groups which differ by the number of items ($n = 20, 40, 60, 80, 100$). Each group contains ten different instances. The benchmark contains in total 500 different instances which can be downloaded from http://www.or.deis.unibo.it/ORinstances/2BP/. Table 1 shows the details information of each class and the distributions used to generate its instances.

Table 2 shows the computational results of applying the proposed algorithm on the benchmark classes for the two types of problems, 2LBP and 2LSP. Regarding 2LBP, $\bar{z}(CPLEX)$ is the average best solutions of number of packed bins used in each group (10 instances) obtained by CPLEX 8.1 within 500 seconds as stated in (Puchinger and Raidl, 2007), where $\bar{z}(Ant2DLP)$ is the corresponding value obtained by the proposed ant algorithm. The performance of the proposed algorithm is measured using the percentage deviation calculated as:

$$\%deviation = 100 * (\bar{z}(Ant2DLP) - \bar{z}(CPLEX))/\bar{z}(CPLEX)$$

Regarding 2LSP, $\bar{z}(LINGO)$ is the average height of the best solutions of the 10 instances of each group obtained by LINGO17.0 within 500 seconds, where $\bar{z}(Ant2DLP)$ is the corresponding value obtained by the proposed ant algorithm. The performance of the proposed algorithm is measured using the percentage deviation calculated as:

$$\%deviation = 100 * (\bar{z}(Ant2DLP) - \bar{z}(LINGO))/\bar{z}(LINGO)$$

Table 2 also shows the performance of $Ant2DLP$ algorithm for the tested benchmark measured as the average number of needed bins (15.27) with a total average %deviation (3.52), and the average strip height (1098.23) with a total average %deviation (2.35) for 2LBP and 2LSP respectively. For the 2LBP the maximum %deviation is 13.16 at class 3 (n=100), and the minimum deviation is 0. For the 2LSP the maximum %deviation is 6.86 at class 3 (n=100), and the minimum deviation is 0.

Figure 5 shows the impact of the bin width on the performance of the algorithm. Although the classes (1 and 2), (3 and 4) or (5 and 6) each pair has the same uniform distribution and different bin widths, it is clear from the figure that the average %deviation for both problems reduces with increasing the width. Except in case of classes 5 and 6, the average %deviation decreases for the 2LBP, and increases for the 2LSP problem.

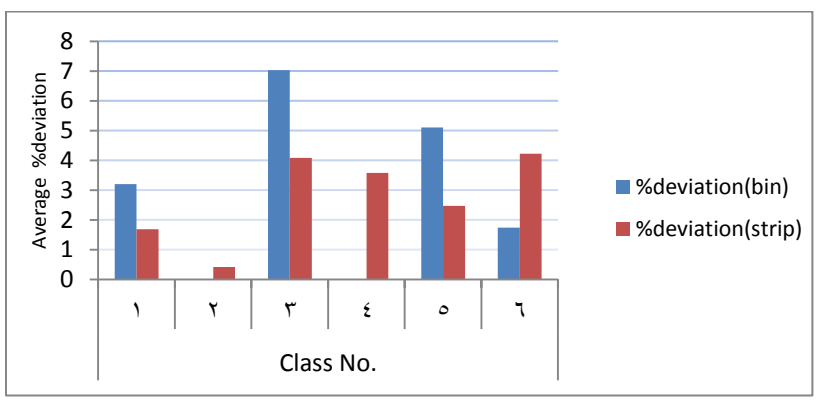

Figure 5. Average %deviation for 2LBP and 2LSP problems within the first six classes (Berkey and Wang classes)

Figure 6 shows the impact of the number of parts in the performance of the proposed algorithm. The results in the figure depend on the class 7, 8, 9, and 10 where the width W for all classes is constant, 100. It is clear that the average %deviation increases with the increase of the number of items for both problems. Moreover, although the algorithm solves the 20 items problem optimally in case of 2LBP, the average %deviation of the 2LBP is greater than the average %deviation of the 2LSP in case of the number of items is greater than 20.

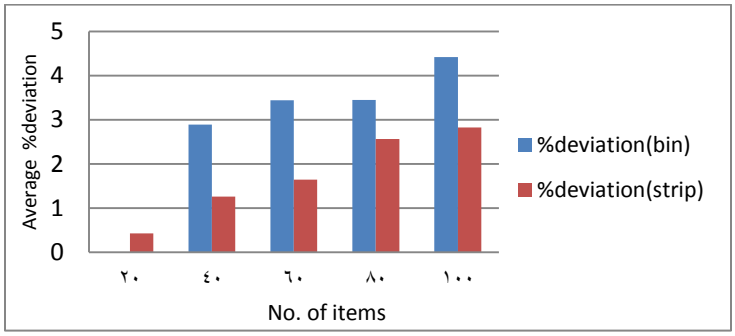

Figure 6. Average %deviation for the two problems within the last four classes (Martello and Vigo classes)

Figure 7 shows the %frequency within the 500 instances and the corresponding %deviation range. The proposed algorithm achieves optimality by 70.4% and 27.8% for the 2LBP and 2LSP respectively. The algorithm is capable of solving around 75% and 90% of the instances with less than 4% and 8% deviation of the optimal solution respectively for both types of problems. No deviation is registered more than 18%. The figure also show that the proposed algorithm gives good results for the 2LBP comparing with the 2LSP.

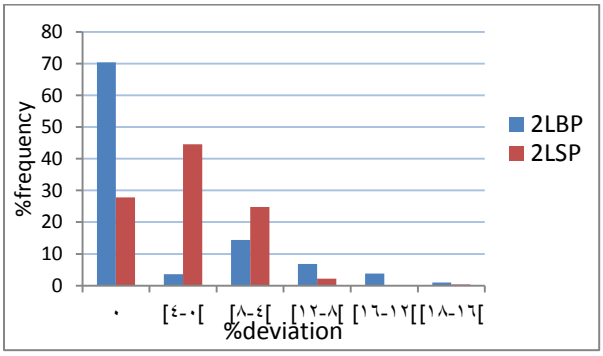

Figure 7. %frequency for the %deviation within 500 instances

5. Conclusions and Future Work

This paper has presented an ACO approach to solve the 2LBP and 2LSP problems. During solution construction, the ants make the best possible choice, as indicated by the pheromone trail and heuristic information. The performance of the proposed ant system has been tested on sets of well-known benchmark problems. The Computational results show that the algorithm is capable of solving the small problems optimally, and gives good performance for large scale ones and has achieved optimality by 70.4%, 27.8% for 2LBP and 2LSP problems, respectively. Also, total averages percentage deviation comparing with the optimal solution are 3.52%, 2.35% for the 2LBP and 2LSP problems, respectively.

More work could be carried out to optimize the algorithm parameters and to find more effective local search method in order to obtain better results for large size problems. Further research might also investigate the impact of the items' rotation on the objective value.

References:

Alvarez-Valdés, R., Parreño, F., and Tamarit, J. M., A tabu search algorithm for a two-dimensional non-guillotine cutting problem, *European Journal of Operational Research*, vol. 183 no. 3, pp. 1167-1182, 2007.

Bennell, J. A., Lee, L. S., and Potts, C. N., A genetic algorithm for two-dimensional bin packing with due dates. *International Journal of Production Economics*, vol. 145, no. 2, pp.547-560, 2013.

Berkey, J. O., and Wang, P. Y., Two-dimensional finite bin-packing algorithms, Journal of the operational research society, pp. 423-429, 1987.

Burke, E., and Kendall, G., Applying ant algorithms and the no fit polygon to the nesting problem, *Advanced Topics in Artificial Intelligence*, pp. 453-464, 1999.

Costa, D. and Hertz, A., Ants can colour graphs, *Journal of the Operational Research Society*, vol. 48, pp. 295-305, 1997.

Cui, Y., Yang, Y., Cheng, X., and Song, P. , A recursive branch-and-bound algorithm for the rectangular guillotine strip packing problem, *Computers and Operations Research*, vol. 35, no. 4, pp. 1281-1291, 2008.

Dorigo, M., Birattari, M., and Stutzle, T. (2006).Ant colony optimization. *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28-39, 2006.

Dorigo, M., and Stutzle, T., The ant colony optimization metaheuristic: Algorithms, applications, and advances, *International series in operations research and management science*, pp. 251-286, 2003.

Dorigo, M., and Gambardella, L. M., Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53-66, 1997.

Dorigo, M., Maniezzo, V., and Colorni, A., Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems*, Man, and Cybernetics, Part B (Cybernetics), vol. 26, no. 1, pp. 29-41,1996.

Falkenauer, E., &Delchambre, A. (1992, May), A genetic algorithm for bin packing and line balancing, In Robotics and Automation, *IEEE International Conference on Proceedings*, pp. 1186-1192, May, 1992.

Fowler, R. J., Paterson, M. S., and Tanimoto, S. L., Optimal packing and covering in the plane are NP-complete, Information processing letters, vol. 12, no. 3, pp. 133-137, 1981.

Gilmore, P. C., and Gomory, R. E., Multistage cutting stock problems of two and more dimensions, *Operations research*, vol. 13, no. 1, pp. 94-120, 1965.

Hopper, E., and Turton, B. C., A review of the application of meta-heuristic algorithms to 2D strip packing problems, *Artificial Intelligence Review*, vol. 16, no. 4, pp. 257-300, 2001.

Hopper, E., *Two-dimensional packing utilising evolutionary algorithms and other meta-heuristic methods*, PhD thesis, University of Wales, Cardiff, U.K., 2000.

Jain, L., and Singh, G., A Review: Meta-heuristic Approaches for solving Rectangle Packing Problem, *International Journal of Computer Engineering and Technology*, vol. 4, no. 2, pp. 410-424, 2013.

Leung, S. C., Zhang, D., Zhou, C., and Wu, T., A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem, *Computers and Operations Research*, vol. 39, no.1, pp. 64-73,2012.

Levine, J., and Ducatelle, F., Ant colony optimization and local search for bin packing and cutting stock problems, *Journal of the Operational Research Society*, vol. 55, no.7, pp. 705-716, 2004.

Lodi, A., Martello, S., and Vigo, D., Models and bounds for two-dimensional level packing problems, *Journal of Combinatorial Optimization*, vol. 8, no. 3, pp. 363-379, 2004.

Lodi, A., Martello, S., and Vigo, D., Recent advances on two-dimensional bin packing problems, *Discrete Applied Mathematics*, vol. 123, no.1, pp.379-396, 2002a.

Lodi, A., Martello, S., and Monaci, M., Two-dimensional packing problems: A survey, *European journal of operational research*, vol. 141, no. 2, pp. 241-252, 2002b.

Martello, S., and Vigo, D., Exact solution of the two-dimensional finite bin packing problem, Management science, vol. 44,no. 3, pp. 388-399, 1998.

Martello, S., Monaci, M., and Vigo, D., An exact approach to the strip-packing problem, *INFORMS Journal on Computing*, vol. 15, no. 3, pp.310-319, 2003.

Mohamed, B. M. A., and Adnan, Y. A. S. S. I. N. E., Optimization by ant colony hybryde for the bin-packing problem, *World Academy of Science, Engineering and Technology*, vol. 49, pp. 354-357, 2009.

Puchinger, J., and Raidl, G. R., Models and algorithms for three-stage two-dimensional bin packing, *European Journal of Operational Research*, vol. *183*, no. 3, pp. 1304-1327, 2007.

Pham, D.T., Afify, A.A. ,Otria, S., Layout optimization of manufacturing cells using a hybrid ant algorithm, *The 5th I*PROMS Virtual International Conference on Innovative Production Machines and Systems*, Cardiff, UK, July, 2009.

Salto, C., Leguizamón, G., Alba, E., and Molina, J. M., Hybrid ant colony system to solve a 2-dimensional strip packing problem, *Eighth International Conference on Hybrid Intelligent Systems, HIS'08, IEEE*, pp. 708-713, September, 2008.

Singh, G., and Jain, L., Increasing Impact of ACO in solving Rectangular Packing Problems, *International Journal of Recent Trends in Engineering and Research* (IJRTER), vol. 02, no. 05, 2016 .

Spiliopoulos, K., and Sofianopoulou, S., An efficient ant colony optimization system for the manufacturing cells formation problem, *The International Journal of Advanced Manufacturing Technology*, vol. 36, no. 5, pp. 589-597, 2008.

Stützle, T., and Hoos, H. H., MAX–MIN ant system, *Future generation computer systems*, vol. 16, no. 8, pp. 889-914, University of Wales, Cardiff, U.K., 2000.

Yagmahan, B., and Yenisey, M. M., Ant colony optimization for multi-objective flow shop scheduling problem, *Computers and Industrial Engineering*, vol. 54, no. 3, pp. 411-420, 2008.

Biographies

Azza Gaber is a Teaching Assistant in Industrial Engineering Department, Faculty of Engineering, Zagazig University, Zagazig, Sharkia, Egypt. She earned B.S. in Industrial Engineering from Zagazig University in 2010. She is a master student working in developing an Ant Colony Optimization Heuristic for solving the 2D Level Packing Problems.

Raafat Elshaer is an Associate professor in Industrial Engineering Department, Faculty of Engineering, Zagazig University; Zagazig, Sharkia, Egypt. He received his B.S. degree in Production Engineering from Faculty of Engineering, Helwan University in 1996, M.Sc. degree in Industrial Engineering from Faculty of Engineering, Zagazig University in 2004, and Ph.D. in Industrial Engineering from Faculty of Engineering, Zagazig University in 2009 as a joint program between Zagazig University and Rutgers University, USA. He has published journal and conference papers. His research interests include optimization, scheduling, project management, earned value management and others.

Mahassen Khater is a Professor emeritus in Industrial Engineering Department, Faculty of Engineering, Zagazig University; Zagazig, Sharkia, Egypt. She received her B.S. degree in Electrical Power and Machines Engineering from Faculty of Engineering, Cairo University in 1973, M.Sc. degree in Electrical Power and Machines Engineering from Faculty of Engineering, Cairo University in 1978, and Ph.D. in System Engineering from Faculty of Engineering, Zagazig University in 1985. She has published journal and conference papers. Her research interests include optimization, scheduling, logistics, project management, and others.