Annals of Electrical and Electronic Engineering, 2(11) 2019, Pages: 7-13

Published by IASE

Annals of Electrical and Electronic Engineering

Journal homepage: www.aeeej.org

Introducing an approach based on an advanced hybrid model for the two-dimensional bin packing problem

P. López *

Polytechnic University of Puerto Rico, San Juan, Puerto Rico

ARTICLE INFO

Article history:
Received 20 March 2019
Received in revised form
22 August 2019
Accepted 23 August 2019

Keywords:
Two-dimensional bin packing
Advanced hybrid model
Pareto optimization
Particle swarm optimization

ABSTRACT

As one of the classical problems in combinatorial optimization, the Bin Packing Problem (BPP) is assumed a branch of computer sciences and a special form of 0-1 knapsack problem and it is derived from the family of NP-Hard. This study aims at achieving optimal packing based on the minimum linear time and maximum applied bins with a focus on increasing the density of the solution set and decreasing in errors. This paper originates from the fact that it tries to achieve the minimum linear time and maximum applied bins for the two-dimensional bin packing problem (2DPP) as the methodology, and also, presents the optimal packing based on the minimum linear time and maximum applied bins for the packing problem in two-dimensional space by means of representation and comparisons based on the particle swarm optimization (PSO) algorithm and the hybrid model. Using the proposed model (the advanced hybrid model), the study covered one of the constraints which were present in the previous research, that is, holding the hard nature of the two-dimensional packing problem, which operates in a way that elongation of time in achieving the optimization state is accompanied with an increase in dimensions of the problems. Finally, the experimental and comparative results in the MATLAB software program approved the success and utility of the proposed model in minimizing the time of achieving the optimal packing and increasing in the number of applied bins in two-dimensional space (dual optimal packing).

1. Introduction

Optimization means to find values for maximizing and minimizing of some parameters in the objective function. Optimization denotes finding one or several solutions in regard to one or several objective functions. As clarified from this title, the multi-objective optimization problem deals with more than one objective function that is minimized or maximized. If the value chosen for the parameters satisfies all the requirements, it is called a possible solution. The solutions, in line with the value of the objective function, are called optimal solutions. Optimal solutions are employed in daily activities such as industrial design, allocation of sources, time-scheduling, decisions, and so on. Additionally, the optimization approaches are also used in many other areas such as industry, engineering and computer sciences. There is a variety of studies and research on optimization, and new optimization approaches are developed in this course. Of the most considerable optimization problems one can mention (a) problems without constraints, (b) compulsory problems, and (c) hybrid problems such as the bin packing (Deb et al., 2005).

Unlike the one-objective optimization problem, due to the presence of several contradictory purposes, there is more than one solution for problems with multiple contradictory objectives in combinatorial optimization that is a set of obtained solutions. The aim of multi-objective optimization is to find the Pareto (dominant) solutions for the problem at hand. There could be a set of solutions for a finite set of solutions and when choosing two solutions one dominates over the other. In other words, the solutions in this set are much better than other solutions. This set is called the dominant set to other sets of given solutions. The obtained dominant set is named the optimal set of Pareto. Any of the solutions in the set of Pareto is an optimal or near to optimal solution that is the give-and-take solution for objective functions. Although we know that we finally need one solution for the problem, the goal of such a procedure is that in many cases the user is not fully aware of the tradeoff relation among the objectives. Therefore, it is better to find a set of Pareto optimal solutions and thereby the user can select the best solution by considering a series of additional information or some presumptions the user keeps in his/her mind (Coello and Lechuga, 2002).

Considering that the problem is too complex and complicated, it can be said that its significance is still notable in two fields of sciences and industry. To give an example, in 1988 a study was conducted on a set of algorithm problems at Stony Brook University; it was reported that out of 75 algorithm problems, this problem was the seventeenth well-known problem and the third frequently used problem after *k*-d tree and tier (Lin et al., 2010). Of cases of use of this problem, one can refer to industry, glasswork, transportation, production planning, itemizing in the production line, industrial sewing, fundamental sciences and engineering, economy, management, and etc. In this study, the model resting on the combination of Pareto optimization and particle swarm optimization is presented and programmed to solve the two-dimensional bin packing problem when increasing the dimension of the problem. Then, after comparing of the programming results in MATLAB, the advanced hybrid model needs to shorter time for linear calculation in comparison with the basic model; that is to say, the particle swarm optimization algorithm is useful in particular when increasing the dimension of the problem. Moreover, it could prove its excellence and efficiency in achieving the optimal dual packing.

* *Corresponding Author.*
Email Address: pablo.lopez@pupr.edu
https://doi.org/10.21833/AEEE.2019.11.002

2. Problem statement

Optimization problems seek for the best solution among all the feasible solutions. Combinatorial optimization problems have created a discrete search space of possible solutions. They have often high computational complexities and are placed in the category of NP-hard. The two-dimensional bin packing problem is a classical optimization problem that is categorized as NP-hard concerning its computational complexity. The two-dimensional bin packing problem is very functional in NP-hard and placed in the category of binary knapsack problems and of integer programming. This problem holds many applications in engineering, economy, industry, management, transportation, automotive industry, etc. To put it differently, combinatorial optimization plays a pivotal role in applied studies, management, fundamental sciences and even industry and it aims at solving the NP-hard combinatorial optimization problems. In several studies, one, two or three-dimensional bin packing problems have been examined. The present study puts the emphasis on the two-dimensional bin packing problem in which the mathematical model is described as follows:

Eqs. 1 to 5 show the mathematical formulation of the two-dimensional bin packing problem. Eq. 1 shows the objective function of the bin packing problem in that it is intended to optimize the number of bins. Eq. 2 illustrates the constraint of the space limit for any bin. Eq. 3 concerns with the constraint of bin placing and Eq. 4 and Eq. 5 express the binary constraint of decision variables. The indices *i* and *j* also represent the two-dimensional bin packing problem or emphasize in a type of two dimensions of this problem (Hong et al., 2014).

$$minimize\ B = \sum_{j=1}^{m} y_j \quad (1)$$
$$\sum_{i=0}^{n} \mathrm{w_i x_{ij}} = \mathrm{V_{y_i}}, \forall_{\mathrm{j}} \in \{1,..,\mathrm{m}\} \quad (2)$$
$$\sum_{j=1}^{m} x_{ij} = 1, \forall_j \in \{1, \ldots, n\} \quad (3)$$
$$y_j \in \{0,1\}, \forall_j \in \{1,..,m\} \quad (4)$$
$$x_{ij} \in \{0,1\} \forall_j \in \{1,..,m\}, \forall_i \in \{1,..,n\} \quad (5)$$

3. Related works

In this section, we will review and discuss the previous research in line with the integrated problem of bin packing and applied optimization algorithms in the actual modeling of problems that have been widely in use and it captured the attention. Of the previous research which has been conducted on promoting the particle swarm optimization algorithm, one can refer to adding the factor of equilibrium concerning the calculation of speed in these changes. This parameter creates a balance between local optimal and the search space of the problem as a large amount of this parameter is suitable for the search and the low amount is appropriate for local optimal. In a study, this gradual decrease in this parameter is stated. Shi and Eberhart (1998) focused on PSO parameters and their impact on the process of work. Shi and Eberhart (2001) introduced the non-linear reduction through fuzzing and in another project (Rantaweera et al., 2004) this amount was considered zero except for initializing times. Furthermore, the gradual reduction of speed was presented in another research (Fan and Shi, 2001). Promoting the PSO through designing different models is another active type of research. Kennedy and Mendes (2002) believed that PSO with small neighbors could act more efficiently in problems with higher complexity or vice-versa. In a study by Rao and Iyengar (1994), a simulated annealing algorithm was taken to solve the bin packing problem. The results of this study were in line with those of the previous research, and a more acceptable utility of simulated annealing was achieved as opposed to the previous researches. Another study (Wang et al., 2010) was devoted to solving BPP through the Ant Colony algorithm and the Simulated Annealing algorithm. This study does not aim to achieve the optimal solution but intends to find the best solution in a predetermined period of time. It is completely obvious that such an approach in solving the BPP could bring a solution that is much farther away from the optimal solution. According to tables presented by Correa and Epstein (2008), using dynamic programming, solving the problem is exposed to the difficulty by an increase in the dimensions of BPP. Moreover, Dósa and He (2006) proposed an iterative algorithm based on learning automata to solve multi-dimensional BPP. In this algorithm, multi-dimensional BPP was modeled by a complete graph in such a way that any node corresponds to an item in the graph; however, at higher scales, it is worthy for more consideration. In another study by Coffman et al. (2008), the Tabu algorithm was also employed to solve the one-dimensional problem, and the potential of the proposed method was specified. Notwithstanding the utility and efficiency of this methodology require further investigation. Because it is inevitable if the suggested model fails to deal with more complicated problems such as two-dimensional BPP and even at larger scales. In the proposed model in this research, the limitations of choosing items in the fuzzy BPP were taken into account and were resolved using a fuzzy linear programming method. The research results confirmed the efficiency of this method to solve large scale problems. Nevertheless, the presented method may fail to cope with all the practical applications of BPP.

Thus, in this research an efficient and proposed model known as “the advanced hybrid model” was proposed to solve the two-dimensional bin packing that is able to reduce the time calculation with achieving the maximum amount of bins for packing in two-dimensional space in a large scale and could offer better solutions in comparison with the other proposed algorithm, that is, the particle swarm optimization algorithm.

4. The reason for introducing the proposed approach

Dual bin packing problem or known as 2DBPP is the third most common problem in NP-hard and is one of the fundamental problems in computer science that includes many applications in sciences and industry. The algorithms utilized to solve this problem are classified into two classes. The first one is the exact algorithms that find the optimal solution of the problem for any input value. In this category, the algorithm execution time depends upon the value of the problem sample, and as the sample value increases, the algorithm execution time also exponentially increases and this is the most serious constraint of it. The other category is meta-heuristic algorithms that have drawn considerable attention in recent years due to certain benefits such as decreasing in the calculation time and proposing novel methods to avert failure in finding solutions and having BPP entangled in the local optimal; particle swarm optimization algorithm is a good example of it. According to recent studies, although the particle swarm optimization algorithm has operated more appropriately as opposed to the other existing methods, it might be still possible for it to be entangled in local optimal points. Accordingly, the structure of the particle swarm optimization algorithm is presented in two parts:

1. Rules of motion and/or searching in the space of the problem solution
2. The memory of the algorithm which offers information of any member required to make decisions about selecting the optimal path.

Enhancing each of the said parts would result in escaping from a locally optimal point. Put it differently, simple but at the same time, significant search rules in particle swarm optimization algorithm have made it preferable to other algorithms in terms of its convergence speed. In this research, therefore, the two-dimensional bin packing problem is solved by the exertion of appropriate and effective changes in particle swarm optimization algorithm. So, the objective of this study is to propose an efficient and updated algorithm that follows the two conditions of optimization in drawing the two-dimensional bin packing problem by evaluating and comparing the proposed algorithms. The implementation of the two-dimensional bin packing problem is based on any proposed method including particle swarm optimization algorithm and the advanced hybrid model.

5. Research methodology

To achieve the optimal dual packing based on the proposed methods, the two-dimensional bin packing is implemented in such a way that the input data were generated randomly and through a uniform distribution method. In other words, the implemented codes randomly generated the data. The programming language of MATLAB also made use of the uniform distribution method to generate these random numbers and input data are made up of two parts, that is to say:

- The size of a bin
- The area of the size† is obtained by multiplying the length by the width

In this study, the MATLAB software program was put into practice to meet the final goal, that is, the optimal dual packing. MATLAB holds a software environment for conducting numerical analysis and is a fourth-generation programming language. The word MATLAB also means the environment for numerical analysis and implies the programming language, deriving from a combination of two words namely; matrix and laboratory. As a multi-purpose software program, MATLAB has many applications in engineering problems.

The steps are explained in tables to achieve the optimizing two-dimensional BPP based on the proposed methods to acquire the optimal dual optimization as follows. Put it differently, the proposed methods are implemented in three sets of standard, finite data and with the value of 20, 50 and 100 data and with the given parameters, and the results of implantation of two-dimensional bin packing are presented and elaborated to accelerate achieving the optimal state and to enhance the accuracy based on each of the proposed methods.

5.1. Optimizing 2DBPP and based on the PSO algorithm

The optimization of the particle swarm optimization algorithm is a kind of meta-heuristic algorithm that has been inspired by the motion of a flock of birds to achieve the goal. The particle swarm optimization algorithm was proposed by Kennedy and Eberhart (1995). This algorithm is associated with both artificial life particularly collective theories and with evolutionary algorithms especially the evolutionary strategy and genetic algorithm.

In this section, the process of optimization of two-dimensional BPP is illustrated with a meta-heuristic PSO algorithm in Fig. 1 and to solve the bin packing problem. In other words, the particle swarm optimization algorithm was conducted on three sets of standard, finite data with the value of 20, 50 and 100 data and with the given parameters to accelerate finding an optimal dual state and to increase the accuracy.

Considering the value of p-best and g-best, any particle makes use of Eq. 6 and Eq. 7 for locating the next position as follows:

$$V_{i,t+1} = V_{i,t-1} + C_1 r1.(P_{besti} - P_{i,t}) + C_2 r2(G_{besti} - P_{i,t}) \quad (6)$$
$$P_{t+1} = P_t + V_t \quad (7)$$

where:
$V_{i,t-1}$: denotes the speed of the *i*th particle in the replication of t+1; $P_{i,t}$: the position of the *i*th particle in the replication of t, and P_{besti} and G_{besti} are sum of the best position of *i*th particle in the path of motion and the best position of the flock or the position of the current population, respectively. The numbers r_1 and r_2 are random numbers with uniform distribution in [0 – 1] range. C_1 and C_2 are other parameters that should be regulated according to the type of the problem for enhancing the efficiency and controlling the algorithm behavior. P_{t+1} is the position of the particle in the replication of t+1, and P_t is the position of the particle in the replication of t and V_t is the speed of the particle in the replication of t and $V_{i,t+1}$ is the speed of *i*th in the replication of t-1 (Alatas et al., 2009).

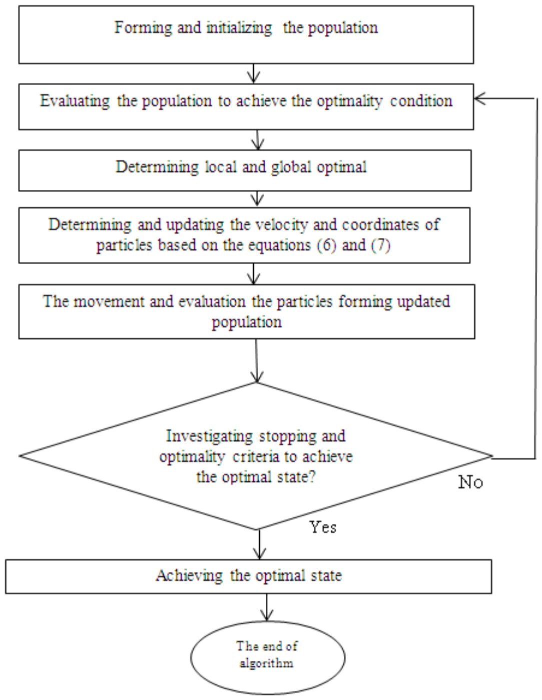

Fig. 1. The procedure of 2DBPP optimization based on the PSO algorithm.

5.2. Optimizing two-dimensional BPP and based on the advanced hybrid model

The proposed strategies are generated by applying changes in the status of the particle swarm optimization algorithm that is called 'the advanced hybrid model'. In fact, it searches for the PSO algorithm from several points and the p-best and g-best of these points are close to the optimal point. The PSO algorithm could be used for discrete and continuous problems and is able to search thoroughly in the problem space. But it operates poorly in local searching and it might be entangled in local optima. In fact, such an approach was proposed because of the non-monotony of the solution set of Pareto optimization, the high level of errors, the low number of the optimization set, and the non-response of the algorithm on the problems at high levels. Thus, the proposed model is implemented based on the advanced hybrid model on two-dimensional BPP then elaborated in Fig. 2. The results of the optimization are presented in the rest part of the paper.

$$V_{id}(t+1) = WV_{id}\,t + C_3 rand(X_{id}(t) - gworst_{id}(t)) \quad (8)$$
$$V_{id}(t+1) = WV_{id}\,t + C_1 rand(pbest_{id}(t) - X_{id}(t)) + C_2 rand(gbest_{id}(t) - X_{id}(t)) \quad (9)$$

where:
$V_{id}(t+1)$: the speed of the particle *i* with *d* dimension in the replication of (t+1), W, C_1, C_2, and C_3 are learning parameters and are set by considering the type and kind of the problem for promoting the efficacy and control of the algorithm behavior.
$V_{id}\,t$: the speed of particle *i* with *d* dimension in the replication of *t* and *rand* is the random number with a consistent distribution between 0 and 1.

† The bins are rectangles.

$X_{id}(t)$: the position of particle i with d dimension in the replication of t, d(t) P_{besti} and d(t) g_{besti} show the best position of particle i in the length of the motion and the best position of the set or the best current population and $gworst_{id}(t)$: the worst position of the set.

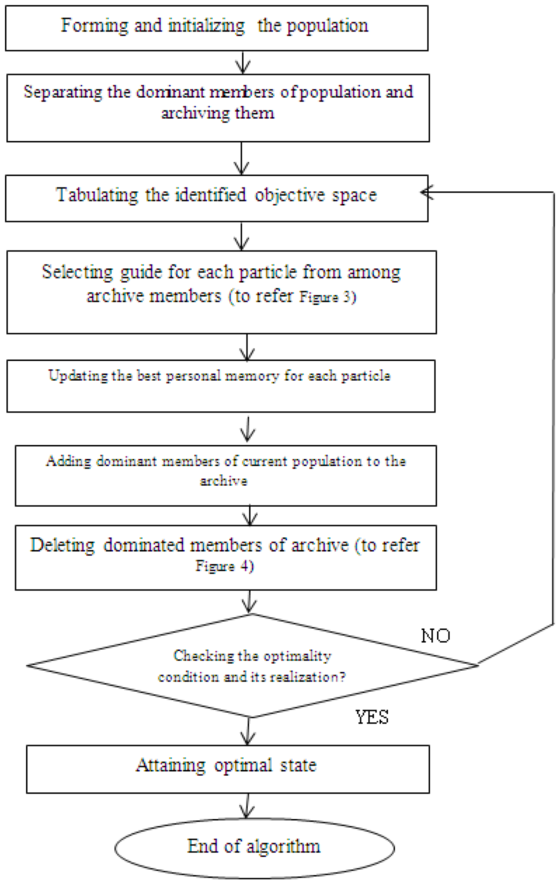

Fig. 2. The procedure of 2DBPP optimization based on the AHM algorithm.

6. Results and discussion

In this section, some steps were taken to achieve the dual optimal optimization at the proposed scales of algorithms.

6.1. Implementing the two-dimensional BPP with a population of 20, 50 and 100

The results of the implementation of two-dimensional bin packing are explained based on the proposed and comparative algorithms. At this stage, the proposed and comparative algorithms are implemented on a set of standard and finite data with the value of 20, 50, 100 and with the given parameters, as shown in Table 1, and the results are presented in Table 2.

At this step, the value of the population was increased from 20 to 50 and 50 to 100 due to an increase in searching, the dimension and complexity of the search space, and more convergence and promoting the results.

The results of the implementation of the two-dimensional bin packing based on the proposed algorithms and with given parameters are as follows:

As it could be seen that reducing of the time needed for achieving the optimal dual packing, the number of bins under packing in the two-dimensional space decreased, and the proposed AHM algorithm could achieve the dual optimization.

Table 1
The value of given parameters for the implementation of the two-dimensional bin packing.

Parameter	Value
Iteration	1000
Size of population	20-50-100
C_1	0.5
C_2	0.5
Size of Data Set1	20
Size of Data Set2	50
Size of Data Set3	100

6.2. Simulating the advanced hybrid models in calculation systems of grid

The scheduling algorithms play a pivotal role in the calculation of networks, the parallel distributed networks for scheduling the tasks and dispatching them towards the appropriate sources. The efficient scheduling algorithms could have maximum exploitation from the sources and load balance with the minimum cost. The problem of Grid Scheduling of the tasks is a technique for the equal distribution of the calculation sources to achieve the optimal optimization of the sources with the lowest response time and more importantly to avoid the extra load over the sources. Here, the load balance, Makespan and the cost are the three important purposes in the multi-objective optimization. The Grid calculation is a new area of calculation that has emerged as an excellent technology in the scope of parallel and distributed calculation. Three of the main objectives in the optimization related to the Grid scheduling which has been observed in the related research are as follows: Makespan, cost, and load balance.

Table 2
The results of the implementation of two-dimensional BPP with proposed algorithms for small, average and great data.

Type of Algorithm	The Time Needed for Processing for Small Data (20)	The Number of Bins for Small Data (20)	The Time Needed for Processing for Average Data (50)	The Number of Bins for Average Data (50)	The Time Needed for Processing for Great Data (100)	The Number of Bins for Great Data (100)
PSO	150.93	14	261.09	34	328.29	66
AHM	134.00	22	244.11	45	299.31	81

6.2.1. Minimizing the makespan

Makespan or the longest time to complete the duty scheduling is the first objective that is introduced as the objective function. Makespan means the longest completion time among all the processors of the system contributing to the timing. One of the main purposes of optimization is to minimize the Eq. 10, implying the performance of the tasks given to the sources in the shortest time.

$$Makespan = Max\{Tcomplete(j)\} \quad 1 \le j \le m \quad (10)$$

$$Tcomplete(j) = \frac{\sum_{k \epsilon Aj} T_k}{c_j} \quad 1 \le j \le m \quad (11)$$

$$c_j = \frac{\sum_{k \epsilon Aj} T_k}{c_j} \quad (12)$$

where:
$Tcomplete(j)$: The time needed to complete the task, T_k: the Kth task, C_j: the processing speed of the processor j, A_j: the set of molecule indexes to the source, *Makespan j*: the longest time needed to complete the task.

6.2.2. Minimizing the cost

As implied earlier, the source providers could charge the users for the sources they use in the Grid calculation based on the market. Therefore, the scheduling algorithms in the market-based Grid calculation consider the needs of the users to complete their applied programs in the most possible economical way. Hence, the second objective function is the total cost of performing the scheduling of the task that must be minimized.

$$Total\ Cost=\sum_{j=1}^{m} Price(j) \quad (13)$$

$$Price(j) =) T_{complete}(j) * Wj \quad (14)$$

where:
Total cost: the whole cost, W_j: the price of each source of j per second, $T_{complete}$: the time needed to complete the task, *Price (j)*: the price of performing of task *i* on the source *j*.

6.2.3. Maximizing the load balance

The mechanism of the load balance is the equal distribution of the load on any source of calculation. This maximizes the load balance, the productivity of the sources and system, and minimizes the time for performing the task. To achieve these objectives, the load balance mechanism should be equal in the distribution of load on the sources. This also requires that the difference should be minimal among the source with the heaviest load and the source with the lightest load. Thus, the load balancer is in its maximum state in the third objective function for optimization that is achieved by minimizing the Eq. 15.

$$Pmsd=\sqrt{\frac{\sum_{j=1}^{m}(Pu(j)-P)^2}{m}} \quad (15)$$

$$Pu(j)=\frac{Tcomplete\ (j)}{Makespan} \qquad 1 \le j \le m \quad (16)$$

$$p = \frac{(\sum_{j=1}^{m} Pu(j))}{m} \quad (17)$$

where:
P: the average productivity of the source, P_{msd}: the load balance all over the sources, *m*: the applied source, *Pu (j):* the expected productivity of any source based on the attribution of the tasks (the average deviation of the squared productivity), $T_{complete}$: the time needed for the tasks, *Makespan*: the longest time needed for the scheduling of the task.

The Makespan and cost in the scheduling of the tasks are two objectives among the significant and fundamental objectives in the Grid economical calculation that is important for the users and the owners of the sources. Additionally, the load balance holds more benefits including decreasing the response time and increasing the productivity of the sources and system. That is the reason why optimizing these objectives is useful in a multi-objective way. Although in the achieved responses each of them holds a special feature, the responses between these areas should be appropriate in terms of these objectives.

6.2.4. Acquired results

The idea of Grid was born in the 1990s and aimed to utilize the unemployed calculation sources around the world for large scientific and research applications. Thus, appropriate scheduling should suggest the lowest cost for the tasks in minimal time with the largest load balance.

AHM is based on the three objectives including cost, Makespan, and load balance. Naturally, Makespan and cost are in conflict with each other. For example, as the time decreases, the Makespan increases or vice versa. The reason is that sources with higher processing are more expensive than the sources with lower processing, and this may result in conflict and incompatibility. For example, when the time decreases, the Makespan increases or vice versa. The details of the optimization data are presented in Table 3.

The details of the optimization data are given in Fig. 3 as follows. According to Fig. 3, this problem was solved so that the density of a set of solutions increased and the number of errors decreased; as a result, a better solution was obtained.

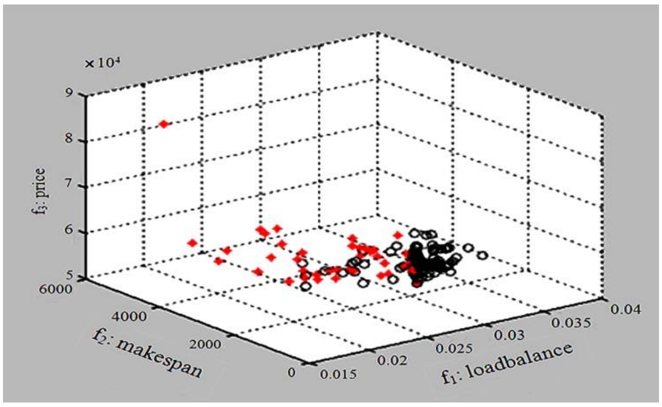

Fig. 3. The proposed approach (AHM).

Table 3
The conditions of optimization algorithms and the parameter of the problem.

Parameter	Amount
The number of generations	100
The number of guides	40
The amount of C2, C1	C1=0.5, C2=0.5
The amount of W	0.2
The number of tasks	500
The number of sources	50
The limit of task size	20-100 (MI)
The amount of source price	1-5(G$/sec)
The limit of processor speed	2-10(MI/sec)
The number of optimization objectives	3

In Fig. 3 and Fig. 4 the results of the particle swarm optimization approach and the proposed approach may be seen and the proposed approach has a higher Makespan and cost.

The proposed approach was compared with the particle swarm optimization and the results were presented in Table 4.

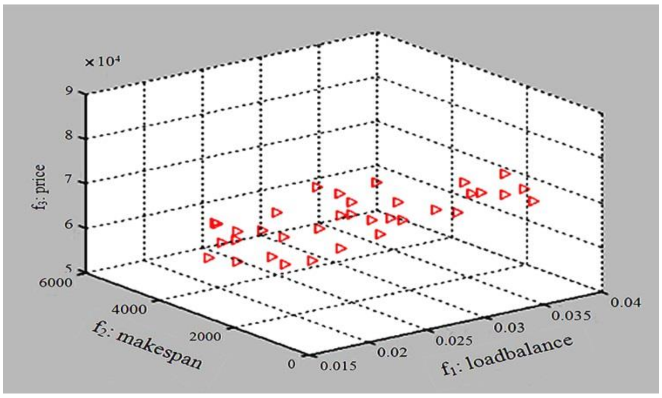

Fig. 4. The particle swarm optimization approach.

Table 4
The optimal solutions of Pareto for AHM and PSO.

Factors	The Best Makespan			The Best Cost			The Best Load Balance		
Approach	Makespan	Cost	The Mean of deviation of squared Productivity	Makespan	Cost	The Mean of deviation of squared productivity	Makespan	Cost	The Mean of deviation of squared productivity
PSO	187.5	101625	0.0196	5498	40426	0.0198	798.89	90323	0.0186
AHM	38.25	26671.08	0.0268	38.25	23704	0.0278	140.998	27449	0.0194

As Table 4 shows, the configuration of the proposed model holds a more efficient Makespan and cost, as opposed to PSO and the quality of the produced AHM solutions of Pareto, is better in comparison with the PSO.

7. The evaluation of the proposed algorithms

Given that not knowing the range of global optimization, the optimization of three non-linear functions is determined via the PSO and AHM methods to investigate the proposed algorithms. In the same way, we use some examples.

Example 1:

$$Minf_1(x) = (x^2 + x) * cos(x) \quad (18)$$

The minimum value of this function is -100.22 occurs at the point x=9.6204. In this example, it is assumed we do not know the range of solution, and the range -11<x<-4 is considered for searching the solution. By choosing this range, the initial position of the members of the set would be in this range.

Fig. 5 shows the process of achieving the solution by the PSO and AHM algorithms. As a matter of fact, Fig. 5 shows that in case of a lack of knowledge about the range of global optimization, the PSO algorithm would be trapped in the local optimal. The proposed algorism is able to get rid of the local optimal and achieve global optimization. The value of optimization found by the PSO and AHM algorithms for the $f_1(x)$ function is presented in Table 5.

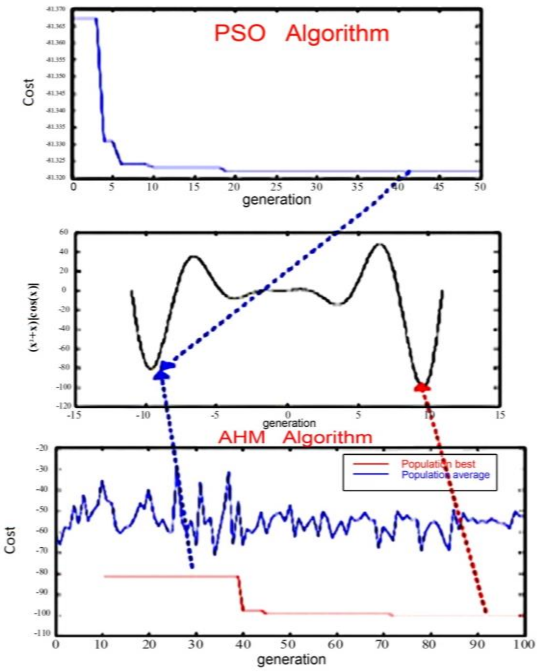

Fig. 5. The process of achieving the optimization of the $f_1(x)$ function by AHM and PSO algorithms.

Table 5
The results of the implementation of PSO and AHM algorithms based on the $f_1(x)$ function.

Method	X	$f_1(x)$
PSO	-9.7	-81.36
AHM	9.62	-100.22

Example 2:

$$Minf_2(x) = y * sin(4x) + 1.1x * sin(2x) \quad (19)$$

The minimum value of this function is -18.544 occurs at the point y=8.668 and x=9.039. In this example, the range 0<x<5 is considered for searching for the solution.

As a matter of fact, Fig. 6 shows that in case of a lack of knowledge about the range of global optimization, the PSO algorithm may be trapped in the local optimal. The proposed algorism is able to get rid of the local optimal. The value of optimization found by the PSO and AHM algorithms for the $f_2(x)$ function is presented in Table 6.

Table 6
The results of the implementation of PSO and AHM algorithms based on the $f_2(x)$ function.

Method	X	Y	$f_2(x)$
PSO	4.3320	5.5503	-10.2825
AHM	8.5729	8.5729	-18.2278

Example 3:

$$Minf_3(x) = -e^{-0.2\sqrt{x^2+y^2}\ +(cos(2x)+sin(2x))} \quad (20)$$

The optimal value of this function is -16.947 occurs at the point x=-2.773 and y=-5. In this example, the range 0<x<5 is considered for searching for the solution.

As a matter of fact, Fig. 7 shows that in case of a lack of knowledge about the range of global optimization, the PSO algorithm might be trapped in the local optimal. The proposed algorism is able to get rid of the local optimal. The value of optimization found by the PSO and AHM algorithms for the $f_3(x)$ function is presented in Table 7.

Table 7
The results of implementation of PSO and AHM algorithms based on the f3(x) function.

Method	X	Y	$F_3(x)$
PSO	3.5533	5.0000	-10.6152
AHM	-2.7678	-5.000	16.9481-

8. Conclusion and further suggestions

The calculation time is very significant to solve the NP-Hard problem at large scales e.g. the two-dimensional BPP. The two-dimensional bin packing problem is regarded as a serious and applied problem in computer sciences. We explored briefly different ways of achieving a solution for the given issue, and the aim of the proposed algorithm was based on obtaining the minimum linear time and maximum applied packing in this case, considering the nature of NP-Hard, could achieve a better optimal performance when increasing the time, compared to the previous research. Compared to the PSO algorithm, the results of the configuration of AHM and PSO to solve the two-dimensional packing issue with similar parameters and under similar conditions illustrated that the rate of convergence of the AHM algorithm increased. The results of simulation show that the proposed approach searches for more optical particles with higher density and fewer errors as opposed to the PSO. This could be substituted as an appropriate solution to solve the problem of multi-objective optimization bin packing. Also, the configuration of the proposed solution on the Grid calculation system involves a more optimal Makespan and cost in comparison with the PSO algorithm, and the quality of AHM generated optimization solutions of Pareto is better than the PSO. At last, by evaluating the efficiency and simulation based on the benchmark function, it was found that the proposed AHM method is more efficient than the PSO algorithm even when we do not have accurate information on the range of optimization to achieve the optimal packing and the proposed method is able to achieve more efficient results.

The following suggestions may be mentioned for further research:

The PSO algorithm is one of the discrete and continuous problems and offers good solutions for various optimization problems. Thus, we select the particle swarm optimization algorithm for the combinatorial optimization and reporting results after implementing different types of PSO including Fuzzy

PSO, Continuous PSO, and Discrete PSO and so on. One of the algorithms is utilized i.e. more powerful in finding the local optimization after implementing any replication of the PSO algorithm; for example the SA algorithm, and then it is combined with the PSO algorithm. Furthermore, the genetic algorithm could be also used and the optimal packing is achieved by choosing the chromosome with the larger size.

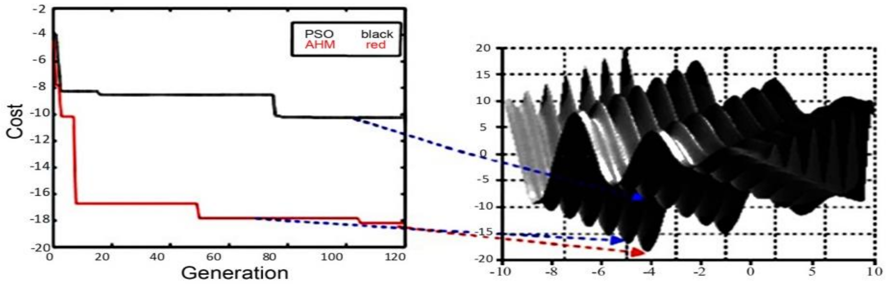

Fig. 6: The process of achieving the optimization of the $f_2(x)$ function by AHM and PSO algorithms.

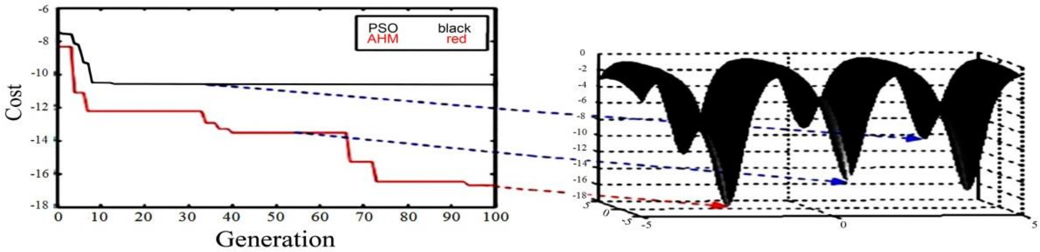

Fig. 7. The process of achieving the optimization of the $f_3(x)$ function by AHM and PSO algorithms.

References

Alatas B, Akin E, and Ozer AB (2009). Chaos embedded particle swarm optimization algorithms. Chaos, Solitons & Fractals, 40(4): 1715-1734. https://doi.org/10.1016/j.chaos.2007.09.063

Coello CC and Lechuga MS (2002). MOPSO: A proposal for multiple objective particle swarm optimization. In the Congress on Evolutionary Computation, IEEE, Honolulu, USA: 1051-1056. https://doi.org/10.1109/CEC.2002.1004388

Coffman Jr EG, Csirik J, Rónyai L, and Zsbán A (2008). Random-order bin packing. Discrete Applied Mathematics 156(14): 2810-2816. https://doi.org/10.1016/j.dam.2007.11.004

Correa JR and Epstein L (2008). Bin packing with controllable item sizes. Information and Computation, 206(8): 1003-1016. https://doi.org/10.1016/j.ic.2008.04.005

Deb K, Burke EK, and Kendall G (2005). Multi-objective optimization in search methodologies. Springer-Verlag, New York, USA.

Dósa G and He Y (2006). Bin packing problems with rejection penalties and their dual problems. Information and Computation, 204(5): 795-815. https://doi.org/10.1016/j.ic.2006.02.003

Fan HY and Shi YH (2001). Study on Vmax of particle swarm optimization. In the Workshop on Particle Swarm Optimization. Purdue School of Engineering and Technology, Indianapolis, USA.

Hong S, Zhang D, Lau HC, Zeng X, and Si YW (2014). A hybrid heuristic algorithm for the 2D variable sized bin packing problem. European Journal of Operational Research, 238(1): 95-103. https://doi.org/10.1016/j.ejor.2014.03.049

Kennedy J and Eberhart R (1995). Particle swarm optimization (PSO). In the IEEE International Conference on Neural Networks, Perth, Australia: 1942-1948.

Kennedy J and Mendes R (2002). Population structure and particle swarm performance. In the Congress on Evolutionary Computation, IEEE, Honolulu, USA: 1671-1676. https://doi.org/10.1109/CEC.2002.1004493

Lin M, Yang Y, and Xu J (2010). On lazy bin covering and packing problems. Theoretical Computer Science, 411(1): 277-284. https://doi.org/10.1016/j.tcs.2009.10.006

Rao RL and Iyengar SS (1994). Bin-packing by simulated annealing. Computers and Mathematics with Applications, 27(5): 71-82. https://doi.org/10.1016/0898-1221(94)90077-9

Ratnaweera A, Halgamuge SK, and Watson HC (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Transactions on Evolutionary Computation, 8(3): 240-255. https://doi.org/10.1109/TEVC.2004.826071

Shi Y and Eberhart RC (1998). Parameter selection in particle swarm optimization. In: Porto VW, Saravanan N, Waagen D, and Eiben AE (Eds.), Evolutionary Programming VII. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. Germany. https://doi.org/10.1007/BFb0040810

Shi Y and Eberhart RC (2001). Particle swarm optimization with fuzzy adaptive inertia weight. In the Workshop on Particle Swarm Optimization. Purdue School of Engineering and Technology, Indianapolis, USA.

Wang S, Wang L, Shi R, and Ge M (2010). Study on improved ant colony optimization for bin-packing problem. In the International Conference on Computer Design and Applications, IEEE, Qinhuangdao, China. https://doi.org/10.1109/ICCDA.2010.5540875