


DOI: <https://dx.doi.org/10.21123/bsj.2022.6568>

3-D Packing in Container using Teaching Learning Based Optimization Algorithm

Linda Fitriyani 

Larissa Alva Zerinda 

Asri Bektı Pratiwi* 

Edi Winarko 

Department of Mathematics Faculty of Science and Technology, Universitas Airlangga, Surabaya, Indonesia.

*Corresponding author: asri.bekti@fst.unair.ac.id

E-mail addresses: lindafitriyani09@gmail.com, zerinda.alva98@gmail.com, edi_winarko@fst.unair.ac.id

Received 23/9/2021, Revised 18/4/2022, Accepted 19/4/2022, Published Online First 20/7/2022,
Published 1/2/2023



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

Abstract:

The paper aims to propose Teaching Learning based Optimization (TLBO) algorithm to solve 3-D packing problem in containers. The objective which can be presented in a mathematical model is optimizing the space usage in a container. Besides the interaction effect between students and teacher, this algorithm also observes the learning process between students in the classroom which does not need any control parameters. Thus, TLBO provides the teachers phase and students phase as its main updating process to find the best solution. More precisely, to validate the algorithm effectiveness, it was implemented in three sample cases. There was small data which had 5 size-types of items with 12 units, medium data which had 10 size-types of items with 106 units, and large data which had 20 size-types of items with 110 units. Moreover, it was also compared with another algorithm called Gravitational Search Algorithm (GSA). According to the computational results in those example cases, it can be concluded that higher number of population and iterations can bring higher chances to obtain a better solution. Finally, TLBO shows better performance in solving the 3-D packing problem compared with GSA.

Keywords: Container, Gravitational Search Algorithm, Mathematics, Teaching Learning Based Optimization Algorithm, 3-D Packing

Introduction:

Transportation plays a vital role in the social-economic aspect as a national, regional, and local distribution channel, both in urban and rural areas. Available items are distributed from companies to serve the customers by available distribution channels¹. Besides routing management optimization², critical activity in distribution is the packaging of items. Packaging in industrial activities especially logistics is based on the increased total cost of storing and distributing manufactured items. Using an optimal package, more items can be put into the container to minimize the use amount of the transport carrier and shipping costs³. Therefore, the companies can save more money in this aspect since the shipping costs are minimized.

A container 3-D packing problem is a complex problem in optimization. Besides having to consider the container volume which has length, width and height, this problem also considers the items

volumes and shapes. Some researchers have proposed several technical placement methods to solve this problem. The container packaging problem discussed in this paper focuses on items having rectangular shapes where one large rectangular box called a container will be filled with different sizes smaller rectangular⁴. The objective of this item's packaging problem is to minimize the space used in the container. The optimal items placement in the container can optimize the item distribution since it will reduce the number of delivery routes. Some methods have been used to solve this problem such as hybrid genetic algorithm^{4,5} and tabu search algorithm⁶. A co-evolutionary multi-objective genetic algorithm has been proposed by adding the encoding and decoding process in order to apply the items placement procedure⁴. While another genetic algorithm hybridization used some rules in the decoding process for replacements⁵.

An algorithm based on the influence of interaction between teacher and students and between students in the classroom has been proposed called Teaching Learning-Based Optimization (TLBO) algorithm. The TLBO algorithm presents two types of teaching-learning: (i) through the teacher known as the teacher phase and (ii) through interacting with other students known as the student phase. Similar to other optimization algorithms, the TLBO brings population and number of iterations as its parameters. A population is considered as student group, while the design variables is considered as various subjects offered to the students⁷. Some optimization problems have been solved using TLBO. It has been proven to solve specific issues, such as problems related to power⁸ and optimizing power flow problems⁹.

The TLBO algorithm has a special technique to find the optimal solution by removing the specific parameters from the process. It only requires general algorithmic parameters *i.e* size of population and design variables⁷. Since TLBO is free from tuning parameters, the usages of TLBO are more straightforward and the resulting solution is a global solution⁷. Based on research conducted by Rao et al.¹⁰, the TLBO algorithm saves the best solution in each iteration to improve the current solutions in order to obtain a convergence towards one point. A comparison between TLBO and several swarm intelligence-based metaheuristics algorithms such as Genetic Algorithm (GA), Artificial Bee Colony (ABC) and Ant Colony System (ANTS) has been conducted. Compared with those algorithms and Grenade Explosion Method (GEM), the TLBO algorithm is satisfactory effective in computation and consistency term¹¹. By using the TLBO, it is expected to find a better solution by minimizing space in containers. In order to validate its performance, the TLBO algorithm was compared with Gravitational Search Algorithm (GSA) introduced by Rashedi et al. in 2009¹². Based on the research that has been conducted, GSA requires a shorter computing time to get a better solution than the Particle Swarm Optimization (PSO)¹³.

This paper is designed thusly: next section is problem formulation continues with the overview of discussed algorithms. Then the procedure to solve the problem is presented in the next section. Last, before the conclusion, a short discussion is presented.

Problem Formulation:

The 3-D packing problem in the container is that items called items smaller than the container are arranged, both in the form of blocks or cubes.

The objective of this problem is to minimize the space in the container¹⁴. It is assumed that all the items can be placed in the container. In 3-D packing problem, there is a number of items (n) that have length, width, and height, as well as containers with a fixed-length (L), fixed-width (W) and fixed-height (H). The items are placed in containers effectively and adequately to optimize the objective function¹⁵. The illustration of 3-D packing problem is presented in Fig. 1. Container shapes can be represented as layers and strips to make this model simpler. The container length is divided as layers representing the items placement while strips partite the container height. The placement of the items will be started from the first layer and strip. After the first layer is full filled, the next item will be placed on the next strip but the same layer. Therefore, the objective function is to minimize the total thickness of layers ($d_k, k = 1, 2, \dots, K$) where K denotes the number of layers in the container.

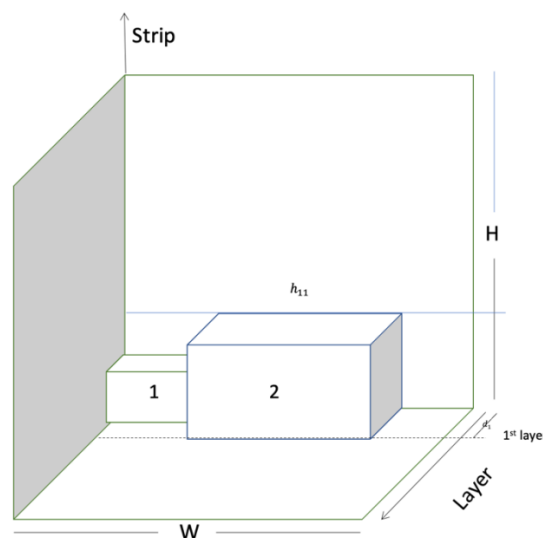


Figure 1. 3-D Packing Problem Illustration

Mathematically, for n number of items, this problem is described as follows while the notation description is presented in Table 1. Eq.1 explains the objective function of this 3-D packing problem while Eqs.2,3 are the constraints. As explained in the previous section, the objective of this problem is to minimize the space used in the container.

Table 1. Notation of 3-D Packing Problem

Notation	Description
Layer	Container length
Strip	Container height
d_k	Thickness of k -th layer
h_{jk}	Height of j -th strip k -th layer
K	Number of layers formed in the container
J	Number of strips included on the layer
p_{ijk}	Length of the i -th item on the j -th strip k -th layer
q_{ijk}	Width of the i -th item on the j -th strip k -th layer
r_{ijk}	Height of the i -th item on the J -th strip k -th layer
W	Container width
H	Container size

$$Z = \min\left(\sum_{k=1}^K d_k\right) \quad 1$$

$$\sum_{i=1}^n p_{ijk} \leq W, \forall j = 1, 2, \dots, J \text{ and } k = 1, 2, \dots, K \quad 2$$

$$r_{ijk} + \sum_{j=1}^{J-1} h_{jk} \leq H, \forall i = 1, 2, \dots, n \text{ and } k = 1, 2, \dots, K \quad 3$$

Eq.2 states that the entire length of the i -th item on the j -th strip k -th layer must not exceed the container's width. Meanwhile, Eq.3 is a constraint that states that the height i -th item on the J -th strip k -th layer that is summed with the height of the j -th strip k -th layer must not exceed the size of the container.

Overview of the Algorithm:

This section explains an overview of Teaching Learning Based Optimization (TLBO) algorithm and the compared algorithm, Gravitational Search Algorithm (GSA).

1. TLBO Algorithm

The teaching and learning process is the main idea for this algorithm. Teaching process involves an interaction between teacher and students while the learning process covers the interaction between students in the classroom based on its output. TLBO only requires general parameters: population and design variables. Population considers a group of students and the design variables as the various available subjects. Students learning outcomes or values are represented as the fitness values for the problems optimized. Two phases of learning are introduced in this algorithm: (i) teacher phase associates to outcomes learning process through the teacher and (ii) students phase associates to interaction with other students¹⁶.

The first TLBO algorithm phase is the teacher phase which the student studies through a teacher. In this learning process, the teacher lectures students supposing improves the average class performances in the subjects taught by the teacher depending on the teacher's abilities. At each iteration i , it is assumed that m number of subjects as design variables ($j = 1, 2, \dots, m$), n number of students as population size ($k = 1, 2, \dots, n$) and $M_{j,i}$ are average results students in certain subjects j - th in iteration i . The best learner is a student who has the best result, $fitness_{best,i}$, for all subjects tested compared with students in population for iteration i . However, teacher has a task to motivate students to obtain better scores in all subjects. Furthermore, student who gets the best score in all subjects can be selected as a teacher. The modification is given below and the description of the notation is in Table 2.

$$Difference_Mean_{j,k,i} = r_i(X_{j,best,i} - T_f M_{j,i}) \quad 4$$

$$T_f = round[1 + rand(0,1)] \quad 5$$

$$X_{j,k,i}^{new} = X_{j,k,i}^{old} + Difference_Mean_{j,k,i} \quad 6$$

Table 2. Notation of TLBO Algorithm

Notation	Description
$Difference_Mean_{j,k,i}$	Difference between the average of j -th subject to k -th student on iteration i
r_i	Random real numbers between 0 and 1 on iteration i
$X_{j,best,i}$	Best learner k -th result for subject j on iteration i
T_f	Teaching factor which decides the value of mean to be changed
$M_{j,i}$	Mean result of the learners in a particular j -th subject on iteration i
$X_{j,k,i}$	Result of j -th subjects to k -th students on iteration i
$rand(0,1)$	Real numbers generated randomly between 0 and 1

Eq.6 is used to update the solution in the teacher phase where $X_{j,k,i}^{new}$ is an updated value of $X_{j,k,i}^{old}$. The $X_{j,k,i}^{new}$ is approved if it provides a better fitness value. After the upgrading process in

teacher phase, the fitness values obtained at the end of this phase will continue to the student phase as an input.

The second basic model of learning is the student phase. The student phase is the second stage of the algorithm, where students interact among themselves in order to improve their knowledge. Students interact with other students and learn new things if other students have more ability. It is assumed that two students interact with P and Q so that $fitness_{P,i} \neq fitness_{Q,i}$.

For the objective function of minimizing using the equation:

$$X_{j,P,i}^{new} = X_{j,P,i}^{old} + r_i(X_{j,P,i}^{old} - X_{j,Q,i}^{old}) \quad 7$$

$$X_{j,P,i}^{new} = X_{j,P,i}^{old} + r_i(X_{j,Q,i}^{old} - X_{j,P,i}^{old}) \quad 8$$

For the objective function of maximizing using the equation:

$$X_{j,P,i}^{new} = X_{j,P,i}^{old} + r_i(X_{j,Q,i}^{old} - X_{j,P,i}^{old}) \quad 9$$

$$X_{j,P,i}^{new} = X_{j,P,i}^{old} + r_i(X_{j,P,i}^{old} - X_{j,Q,i}^{old}) \quad 10$$

If $fitness_{P,i}$ is better than $fitness_{Q,i}$ then update $X_{j,P,i}$ using Eq.7 for minimizing problem and Eq.9 for maximizing problem. If $fitness_{P,i}$ is worse than $fitness_{Q,i}$ then update $X_{j,P,i}$ using Eq.8 for minimizing problem and Eq.10 for maximizing problem. $X_{j,P,i}^{new}$ is approved if it provides better fitness value¹⁴. The overall steps of this algorithm are presented in Fig. 2.

2. GSA

In 2009, Rashedi *et al.* introduced GSA. The GSA is inspired by the object gravity law considering the mass presences. In this algorithm, each individual is assumed to be an object, and their performance is seen from its mass. Each object interacts with other things through the force of gravity. The heavier the mass of the object, the better solution. A more severe group expresses slow movement¹⁰.

There are four specifications described in GSA *i.e* the objects positions, mass of inertial, their mass in active gravitational, and their mass in passive gravitational. The part of the object is the solution to the problem. The fitness value determines the gravitational mass and inertial mass. This algorithm is directed by adapting the mass of gravitational and inertial. There are three parameters used in GSA, such as the initial gravitational constant (G_0), the coefficient of decrease (α), the epsilon value (ϵ). With G_0 , α , ϵ is a non-negative number. Following are the steps in GSA¹⁷:

- i) Initialization of parameter
Initialization of parameters is used to solve the problem, such as the number of objects, initial gravitational constant, alpha value

(α), epsilon value (ϵ), and maximum iteration.

- ii) Initialization population
Generating the initial population of objects consisting of random real numbers at the interval (0,1). The process of developing a population is done as many as N (number of objects).
- iii) Initialization velocity of an object
It is the same as generating an initial population of objects. Effecting the initial velocity of things consisting of random real numbers at the interval (0,1) of the number of units of items. The process of generating a population is done as many as N (number of objects).
- iv) Grouping items by the type
Each dimension of the object will be transformed into a sequence of items to be put into the container. The first step is to sort the dimensions from the smallest to the biggest and then number the dimension. From the numbering, the data is modified in such a way as to order items by type.
- v) Objective function evaluation
Evaluating the objective function for each object.
- vi) Update the gravitational constant, *best*, *worst*

Gravitational constant (G), *best*, *worst* is always updated for each iteration. To determine the gravitational constant obtained using Eq.11 in the following formula:

$$G(t) = G_0 e^{\left(\frac{-\alpha t}{T}\right)} \quad 11$$

where

$G(t)$: the gravitational constant in the t -th iteration

G_0 : initial gravitational constant ($G_0 > 0$)

α : coefficient of decrease ($\alpha > 0$)

T : maximum iteration

Then determine *best* and *worst*.

- vii) Calculates inertial mass for each object
Calculating the mass of inertial, mass of active gravitational and passive gravitational. To determine these masses, the thing to do is calculate the mass value of objects for each t iteration with Eq.12, which is in the formula as follows:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad 12$$

where

- $fit_i(t)$: objective function value of i -th object in t -th iteration
- $worst(t)$: object having the worst objective function value in the t -th iteration
- $best(t)$: object having the best objective function value in the t -th iteration

After determining the mass value, then calculate the inertial mass, active gravitational mass, and passive gravitational mass with Eq.13, which is in the following formula:

$$M_{ai} = M_{pi} = M_{ii} = M_i \quad 13$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad 14$$

where

- M_{ai} : mass of active gravitational for object i
- M_{pi} : mass of passive gravitational for object i
- M_i : mass of inertial mass for object i
- $M_i(t)$: inertial mass of object i on t -th iteration
- $m_i(t)$: the mass value of object i on t -th iteration
- $\sum_{j=1}^N m_j(t)$: the sum of the mass weight of all objects on t -th iteration

viii) Calculates the force for each object

The next step is to calculate the total force on each object. To find out the actual object, the thing to do is to figure the object against other objects in each dimension. Therefore, Eq.15 is used as follows:

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \cdot M_{aj}(t)}{R_{ij}(t) + \varepsilon} \left(x_j^d(t) - x_i^d(t) \right) \quad 15$$

with the calculation of $R_{ij}(t)$, which is the distance of object i with object j in Eq.16 as follows¹¹:

$$R_{ij}(t) = \sqrt{\sum_{d=1}^D (x_i^d(t) - x_j^d(t))^2} \quad 16$$

where

- $F_{ij}^d(t)$: force of object i from object j in the d -th dimension in t -th iteration.
- $x_j^d(t)$: position of object j in d -th dimension in t -th iteration

From the force of one object to another with the formula above, the total energy of each object can be determined using Eq.17 as follows:

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad 17$$

where

- $F_i^d(t)$: total force of object i -th in d -th dimension on iteration t
- $rand_j$: uniform real numbers generated randomly in (0,1) for object j

ix) Calculates the acceleration for each object

After the total force of each object has been obtained, the next step is to calculate the acceleration for each object in each dimension. Determine the using acceleration Eq.18 as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad 18$$

where

- $a_i^d(t)$: acceleration of object i in d -th dimension on iteration t

x) Update velocity and position of the object for the next iteration

With the acceleration of each object that has been obtained, the velocity and position for the next iteration can be calculated. To find out the velocity of each object can be calculated using Eq.19 as follows:

$$v_i^d(t+1) = rand_i v_i^d(t) + a_i^d(t) \quad 19$$

where

- $v_i^d(t+1)$: velocity of object i in d -th dimension on iteration $t+1$
- $rand_i$: uniform real numbers generated randomly in (0,1) for object i

After the velocity for each object is obtained, then the position for the next iteration can be calculated using Eq.20 as follows:

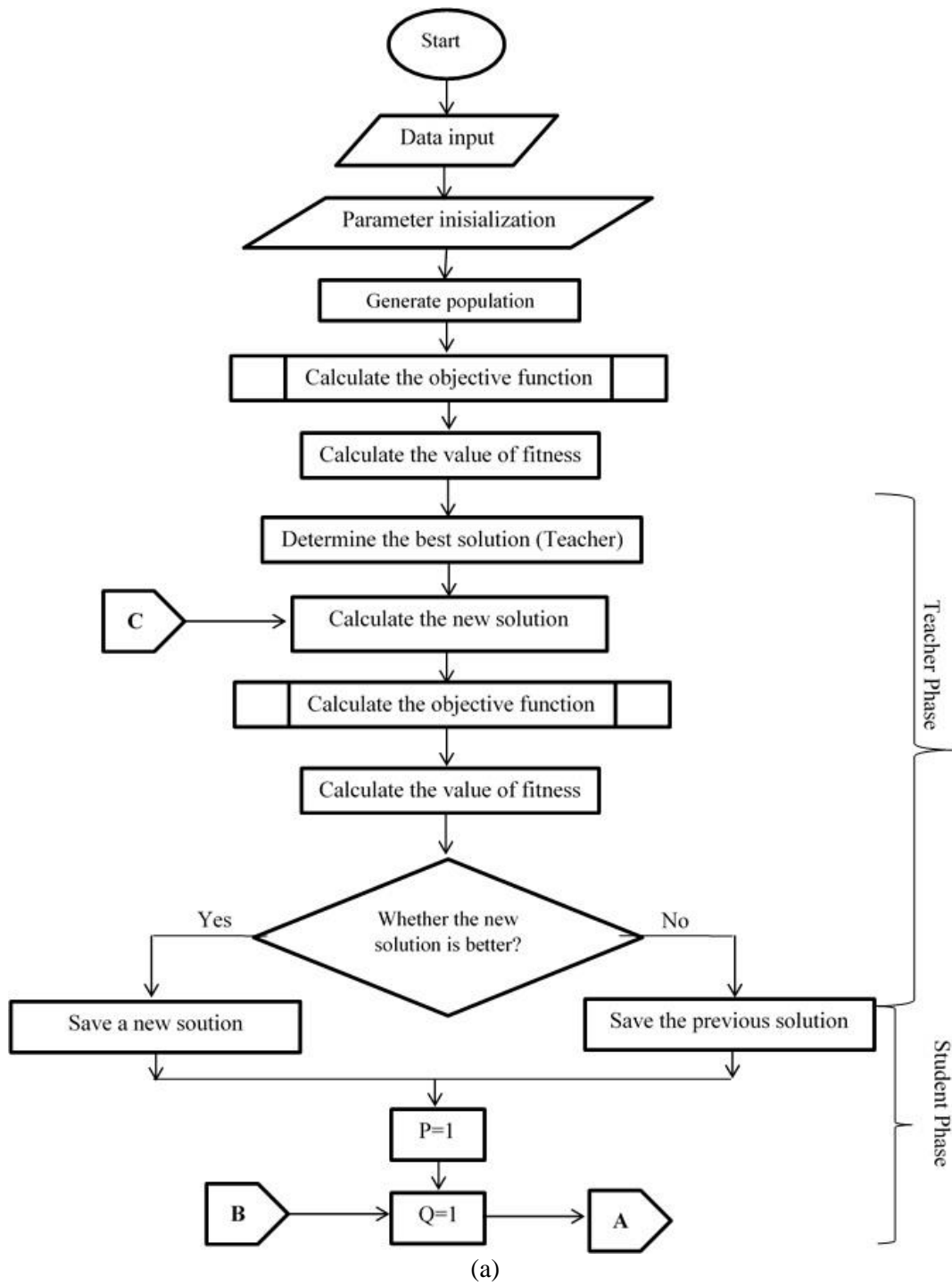
$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad 20$$

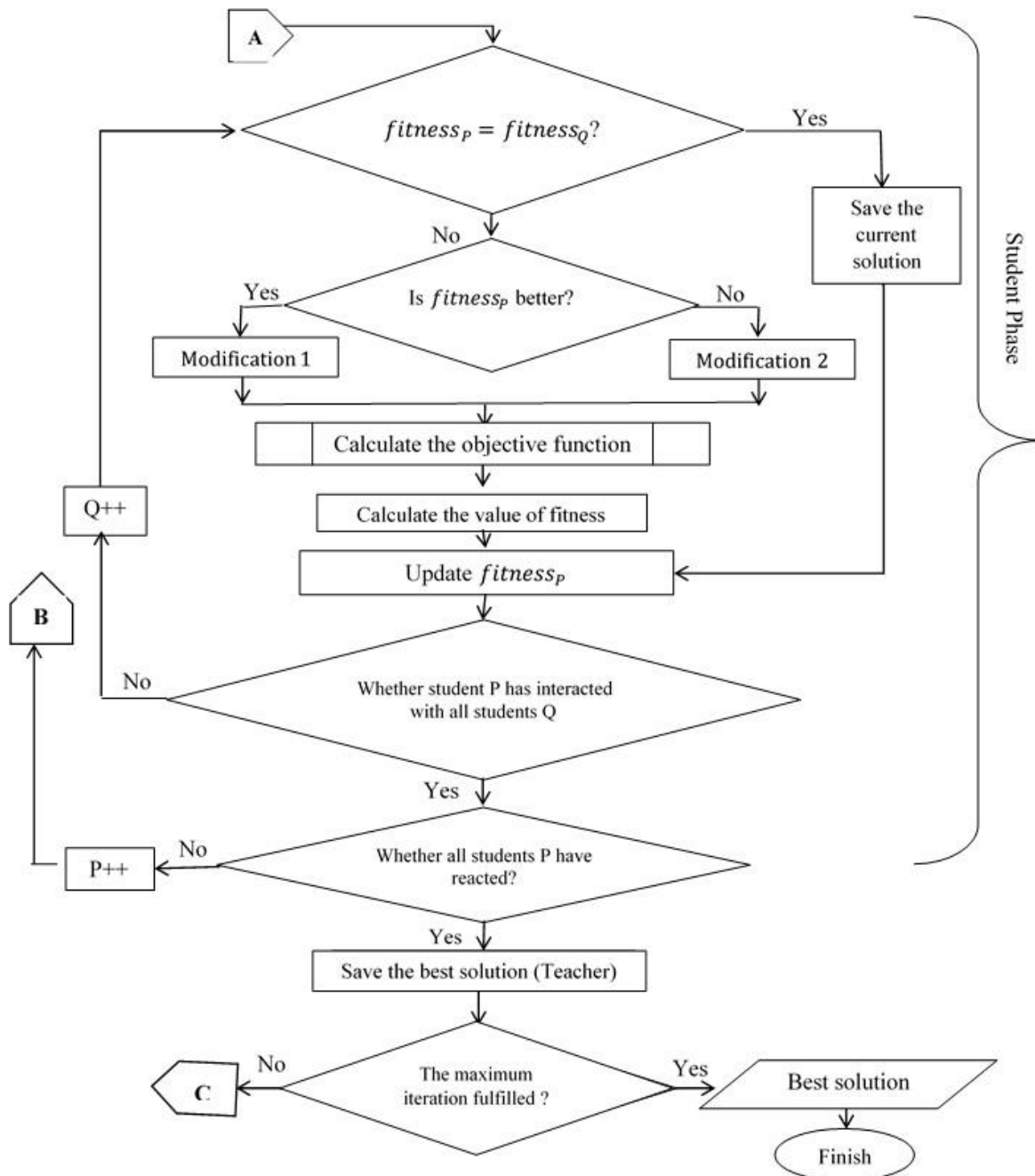
where

- $x_i^d(t+1)$: position of object i in d -th dimension on iteration $t+1$

xi) Check the maximum iteration

In this step, a maximum iteration is checked. If the ultimate iteration has been reached, the results of the best objective function can be displayed. If the complete iteration has not been called, a new position has been formed, and the objective function is calculated.





(b)

Figure 2. Flowchart of TLBO

Procedure for 3-D Packing:

The optimization problem that appeared in this 3-D packing problem is to find the ideal placement of available items in a container so that the space uses can be optimized. The solution in the algorithm represents the sequence of items placement. After the solution is obtained, then the objective function of each solution is determined in order to find the best items placement. This procedure refers to the objective function calculation of TLBO algorithm presented in Fig.2. For $i = 1, 2, \dots, n$ items where i is the placement order index, the placement method can be described as follows¹⁵:

- 1) Assign the i th item in the container based on the available layer and strip.
- 2) Check whether the item can be placed on a given layer. If it is possible, then continue to the next step. Otherwise, go to step 7.
- 3) Put the i th item in the container in the given layer and strip.
- 4) Update the length remains in the layer or height remains in the strip. Go to step 1 for the next item.
- 5) Check whether the item can be placed on a given strip. If it is possible, then continue to step 3. Otherwise, continue to the next step.

- 6) Add a new layer and back to step 2.
- 7) Add a new strip on a given layer and back to step 5.

Steps in items placement in the container needed for the objective function calculation are shown in Fig.3.

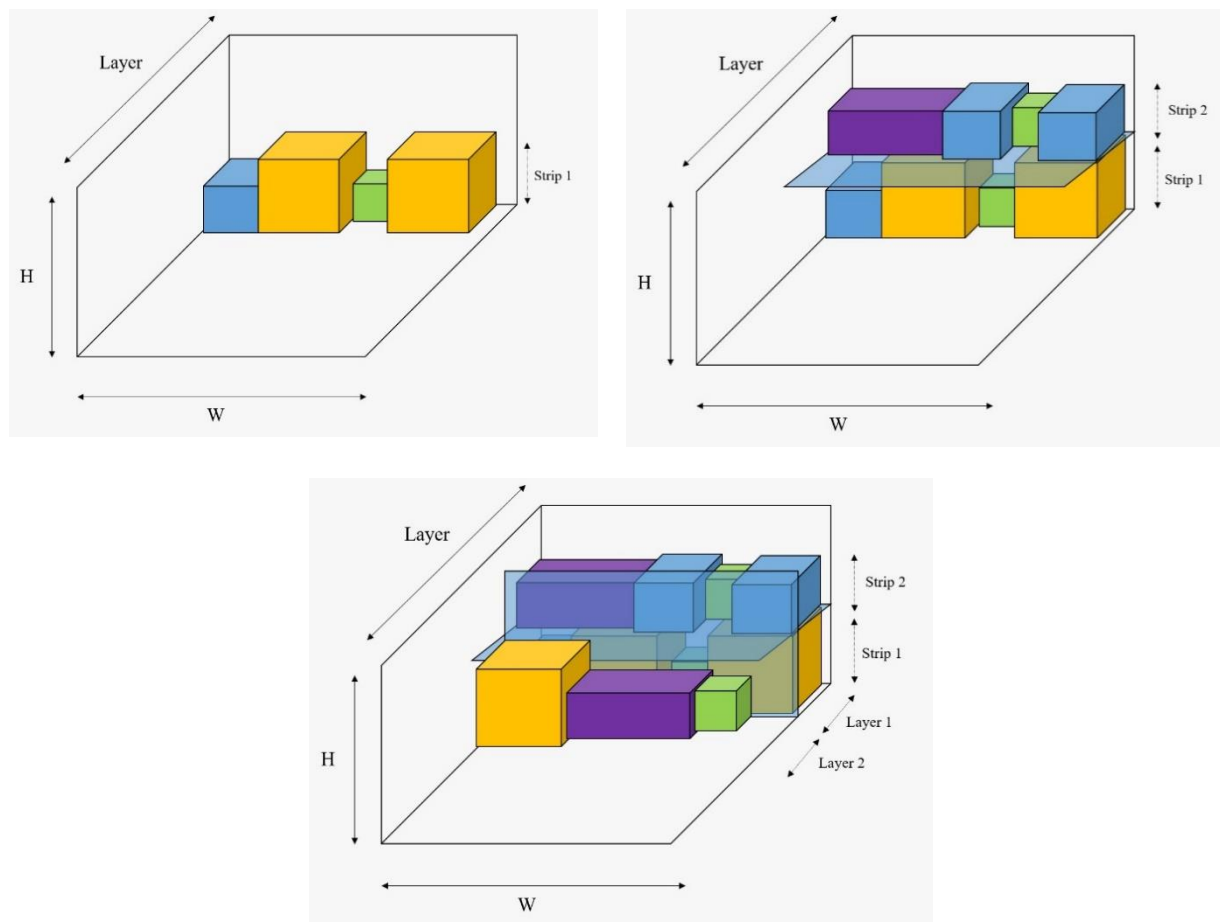


Figure 3. Placement Steps in 3-D Packing

Result and Discussion:

TLBO algorithm program to obtain the solution for the 3-D Packing problem has been implemented in three types of cases. The program is implemented in the C++ programming language using Borland C++ 5.02 Software. From the data obtained from *OR-Library* by J. E. Beasley, this paper used different number of items per type. In this case, there are as many as three types of data, which are small data¹⁸, medium¹⁹, and large²⁰. The result of running the calculation of the objective function value using small data are packing data consisting of 5 types of items with 81 units of items, medium data composed of 10 kinds of items with

106 units of items, and big data consisting of 20 types of items with 110 units of items. The following tables are the calculation results of the objective function values obtained from the program created using two general parameters both in TLBO and GSA: the number of students/ objects and the maximum iteration. Moreover, the parameters used are the initial gravitational constant (G_0) = 100 and coefficient of decrease (α) = 20^{21} , epsilon value (ϵ) is various depending on the best result after experiments. The computational result and its comparison with GSA for small, medium, and large data are presented in Table 3, Table 4, and Table 5, respectively.

Table 3. Performance Comparison for Small Data

Number of iteration	Objective Function Value					
	10		50		100	
Number of students/objects	10	50	10	50	10	50
TLBO	855	837	855	837	855	819
GSA	855	855	855	855	875	855

Table 4. Performance Comparison for Medium Data

Number of iteration	Objective Function Value					
	10		50		100	
Number of students/objects	10	50	10	50	10	50
TLBO	919	914	919	904	912	894
GSA	939	942	930	921	927	899

Table 5. Performance Comparison for Large Data

Number of iteration	Objective Function Value					
	10		50		100	
Number of students/objects	10	50	10	50	10	50
TLBO	979	918	951	888	906	888
GSA	950	954	955	951	947	951

Based on Table 3, 4 and 5, the TLBO performs better using higher number of iterations. On the other hand, higher number of iterations failed to obtain a better solution in GSA as seen in Table 5. A similar pattern has happened in population size parameter. For TLBO, higher number of population size can bring a better solution in 3-D packing problem while as seen in Table 4 and 5, this parameter does not bring much effect in GSA. It is proven that the general parameters applied in TLBO is sufficient to find a better solution in this problem. Therefore, it does not require any specific or tuning parameters. Furthermore, the TLBO gives better performance compared with the GSA in all three types of data.

Conclusion:

According to the previous discussion, it can be terminated that TLBO algorithm can be implemented to solve the problem of 3-D Packing. The simulation results demonstrate that TLBO algorithm provides a better solution when the number of iteration and the number of students increases. Moreover, it performs better compared with GSA in solving the 3-D Packing problem.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Universitas Airlangga.

Authors' contributions statement:

L. F. and L. A. Z. performed the computations. A. B. P. and E. W. verified the methods. A. B. P. supervised the findings of this work. All authors discussed the result and contributed to the paper.

References

1. Sarder MD. Chapter 1 Overview of transportation logistics. Logistics Transportation Systems. Elsevier Inc, Netherlands; 2021; 1-35.
2. Sheng, NL, Arbaiy N, Wen CC, Lin PC. Delivery Route Management based on Dijkstra Algorithm. Baghdad Sci J. 2021;18(1) (Suppl.March); 728-736. Available from: [http://dx.doi.org/10.21123/bsj.2021.18.1\(Suppl.\).0728](http://dx.doi.org/10.21123/bsj.2021.18.1(Suppl.).0728).
3. Sheng L, Xiuqin S, Changjian C, Hongxia Z, Dayong S, Feiyue W. Heuristics algorithm for the container loading problem with multiple constraints. Comput Ind Eng. 2017; 108: 149-164. Available from : <https://doi.org/10.1016/j.cie.2017.04.021>.
4. Zheng JN, Chien CF, Gen M. Multi-Objective Multi-Population Biased Random-Key Genetic Algorithm for the 3-D Container Loading Problem. Comput Industrial Eng. 2015; 89: 80-87. Available from: <https://doi.org/10.1016/j.cie.2014.07.012>.
5. Alvarez-Valdes R, Carravilla MA, Oliveira JF. Cutting and packing. Handbook of Heuristics. Springer Cham, Switzerland; 2018; 1-46
6. Zhuo Q, Liu X. A swarm optimization algorithm for practical container loading problem. Proceedings IECON 2017 – 43rd Annual Conference of the IEEE Industrial Electronics Society. 2017; 133535. Available from: <https://doi.org/10.1109/IECON.2017.8216987>.
7. Rao RV, Savsani V, Balic J. Teaching-Learning-Based Optimization Algorithm for Unconstrained and Constrained Real-Parameter Optimization Problem. Eng Optim. 2012; 44(12): 1447-1462. Available from: <https://doi.org/10.1080/0305215X.2011.652103>.
8. Joshi PM, Verma HK. An improved TLBO based economic dispatch of power generation through

- distributed energy resources considering environmental constraints. *Sustain Energy Grids Netw.* 2019; 18: 100207. Available from: <https://doi.org/10.1016/j.segan.2019.100207>.
9. Farahani HF, Rashidi F. An improved teaching-learning-based optimization with differential evolution algorithm for optimal power flow considering HVDC system. *J Renew Sustain Energy.* 2017; 9: 035505. Available from: <https://doi.org/10.1063/1.4989828>.
10. Rao RV, Savsani VJ, Vakharia DP. An Optimization Method for Continuous Non-Linear Large Scale Problem. *Inf Sci.* 2012; 183(1): 1-15. Available from: <https://doi.org/10.1016/j.ins.2011.08.006>.
11. Yasear SA, Ku-Mahamud KR. Taxonomy of Memory Usage in Swarm Intelligence-Based Metaheuristics. *Baghdad Sci J.* 2019;16(Special Issue); 445-452. Available from: [https://doi.org/10.21123/bsj.2019.16.2\(SI\).0445](https://doi.org/10.21123/bsj.2019.16.2(SI).0445).
12. Rashedi E, Rashedi E, Nezamabadi-pour H. A comprehensive survey on gravitational search algorithm. *Swarm Evol. Comput.* 2018; 41: 141-158. Available from: <https://doi.org/10.1016/j.swevo.2018.02.018>.
13. Siddique N, Adeli, H. Chapter 2 Gravitational Search Algorithm. *Nature-Inspired Computing Physics and Chemistry-Based Algorithms.* Taylor and Francis Group, London; 2017; 51-110.
14. Parreno F, Alvarez-Valdes R, Oliveira JF, Tamarit JM. Neighborhood Structures for the Container Loading Problem: A VNS Implementation. *J Heuristics.* 2010; 16: 1-22. Available from: <https://doi.org/10.1007/s10732-008-9081-3>.
15. Dereli T, Sena Das G. A Hybrid Simulated-Annealing Algorithm for Two-Dimensional Strip Packing Problems. *International Conference on Adaptive and Natural Computing Algorithms.* 2007; 4431: 508-516. Available from: https://doi.org/10.1007/978-3-540-71618-1_56.
16. Rao RV. Chapter 2 Teaching-Learning-Based Optimization Algorithm. *Teaching Learning Based Optimization Algorithm and Its Engineering Applications.* Springer Cham, Switzerland. 2016; 9-39.
17. Mittal H, Tripathi A, Pandey AC, Pal R. Gravitational search algorithm: a comprehensive analysis of recent variants. *Multimed Tools Appl.* 2021; 80: 7581-7608. Available from: <https://doi.org/10.1007/s11042-020-09831-4>.
18. Beasley JE. Available from: *OR-library*: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/wtpack2.txt>. [12 January 2020]. *OR-Library*: distributing test problems by electronic mail. *J Oper Res Soc.* 1990; 41 (11): 1069-1072.
19. Beasley JE. Available from: *OR-library*: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/wtpack4.txt>. [12 January 2020]. *OR-Library*: distributing test problems by electronic mail. *J Oper Res Soc.* 1990; 41 (11): 1069-1072.
20. Beasley JE. Available from: *OR-library*: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/wtpack7.txt>. [12 January 2020]. *OR-Library*: distributing test problems by electronic mail. *J Oper Res Soc.* 1990; 41 (11): 1069-1072.
21. Sahu RK, Panda S, Padhan S. Optimal Gravitational Search Algorithm for Automatic Generation Control of Interconnected Power Systems. *Electr Eng.* 2014; 5(3); 721-733. Available from: <https://doi.org/10.1016/j.asej.2014.02.004>.

التعبئة ثلاثية الأبعاد في الحاوية باستخدام خوارزمية التحسين القائم على التعلم

ليندا فيتراني لاريسا الفازاريندا اسري بكتي براتيوي* ادي ويناركو

قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة إيرلانجا، سورابايا، إندونيسيا.

الخلاصة:

تهدف الورقة إلى اقتراح خوارزمية التحسين القائم على التعلم (TLBO) لحل مشكلة التعبئة ثلاثية الأبعاد في الحاويات. الهدف الذي يمكن تقديمه في نموذج رياضي هو تحسين استخدام المساحة في الحاوية. تراقب هذه الخوارزمية أيضاً، إلى جانب تأثير التفاعل بين الطلاب والمدرسة، عملية التعلم بين الطلاب في الفصل والتي لا تحتاج إلى أي معلمات تحكم. وبالتالي، يوفر TLBO لمرحلة المعلمين ومرحلة الطلاب كعملية تحديث رئيسية لإيجاد أفضل حل. بتعبير أدق، للتحقق من فعالية الخوارزمية، فقد تم تنفيذها في ثلاث حالات نموذجية. كانت هناك بيانات صغيرة تحتوي على 5 أنواع من الأحجام مع 12 وحدة، وبيانات متوسطة تحتوي على 10 أنواع من الأحجام مع 106 وحدة، وبيانات كبيرة تحتوي على 20 نوعاً من الأحجام مع 110 وحدة. وتمت مقارنتها، علاوة على ذلك، بخوارزمية أخرى تسمى خوارزمية البحث عن الجاذبية (GSA). وفقاً للنتائج الحسابية في تلك الحالات النموذجية، يمكن استنتاج أن العدد الأكبر من السكان والتكرارات يمكن أن يجلب فرصاً أكبر للحصول على حل أفضل حل. ويُظهر TLBO أداءً أفضل في حل مشكلة التعبئة ثلاثية الأبعاد مقارنةً بـ GSA.

الكلمات المفتاحية: حاوية، خوارزمية البحث الجاذبية، الرياضيات، تدريس خوارزمية التحسين القائم على التعلم، تغليف ثلاثي الأبعاد