BAREKENG: Journal of Mathematics and Its Applications
December 2025 Volume 19 Issue 4 Page 2791–2802
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19no4pp2791-2802

PARTICLE SWARM OPTIMIZATION FOR CUTTING ALUMINUM STOCK AND ITS COMPARISON WITH THE EXACT METHOD

Bib Paruhum Silalahi [1*], **Siti Aminah** [2], **Hidayatul Mayyani** [3], **Amril Aman** [4]

[1,2,3,4] *School of Data Science, Mathematics, and Informatics, IPB University*
Jln. Dramaga, Campus IPB Dramaga, Bogor, 16680, Indonesia

*Corresponding author's e-mail: * **bibparuhum@gmail.com***

Article History:

Received: 1st March 2025
Revised: 2nd May 2025
Accepted: 10th June 2025
Available online: 1st September 2025

Keywords:

Cutting Stock Problem;
One-Dimensional;
Particle Swarm Optimization;
Trim Loss.

ABSTRACT

The Cutting Stock Problem (CSP) is a common challenge in many industries, involving the optimization of material cutting to minimize waste while meeting customer demands. Various methods can be used to address this issue. This paper applies the heuristic Particle Swarm Optimization (PSO) method to solve CSP in the case of one-dimensional aluminum roll cutting. First, we identify feasible cutting pattern combinations. A mathematical model and constraints are then formulated based on these patterns. Next, the PSO algorithm is employed to determine the optimal combination of cutting patterns, minimizing material waste. The results yield the optimal aluminum roller cutting pattern. Furthermore, we compare the results between the PSO method and the exact method.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-ShareAlike 4.0 International License (https://creativecommons.org/licenses/by-sa/4.0/).

How to cite this article:

B. P. Silalahi, S. Aminah, H. Mayyani, and A. Aman, "PARTICLE SWARM OPTIMIZATION FOR CUTTING ALUMINUM STOCK AND ITS COMPARISON WITH THE EXACT METHOD", *BAREKENG: J. Math. & App.*, vol. 19, iss. 4, pp. 2791-2802, Deember, 2025.

Copyright © 2025 Author(s)
Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

The cutting stock problem (CSP) is one of the problems that we often find in many manufacturing industries, such as textile, garment, metal equipment, paper, shipbuilding, and sheet metal. Cutting stock problem (CSP) is a well-known optimization problem where large materials need to be cut into smaller pieces to meet specific demands while minimizing waste.

The cutting stock problem (CSP) has several variants, including one-dimensional CSP [1], [2], which involves cutting rolls, rods, or beams; two-dimensional CSP [3], [4], which deals with cutting sheets of material like wood, fabric, or glass; and three-dimensional CSP [5] , which focuses on cutting blocks used in industries such as packaging and furniture making.

Several techniques have been proposed to solve the cutting stock problem. Exact methods include integer linear programming (ILP), branch-and-price, and dynamic programming, which were introduced by Gilmore and Gomory in 1961 and 1963 [6], [7]. These methods guarantee optimal solutions but can be computationally expensive for large problems. In contrast, heuristic methods such as genetic algorithms (GA) [8], [9], simulated annealing (SA) [10], [11], and particle swarm optimization (PSO) [12], [13] provide faster, though not necessarily optimal solutions, making them suitable for large-scale and complex cutting stock problems.

This study addresses the one-dimensional cutting stock problem in the context of cutting large aluminum rolls into smaller ones. The primary objective is to determine an optimal cutting pattern that satisfies demand requirements while minimizing trim loss. A comprehensive analysis is conducted on solving the CSP using a heuristic approach based on the Particle Swarm Optimization (PSO) algorithm. The paper presents the formulation of the CSP model specific to aluminum roll cutting and demonstrates how it is solved using PSO. Additionally, the trim loss results obtained from the PSO method are compared with those generated by an exact optimization method.

2. RESEARCH METHODS

This section describes how the particle swarm optimization method is applied to solve the one-dimensional cutting stock problem. Following the explanation of the particle swarm optimization method, the data used in the cutting stock problem model will also be presented.

2.1 Particle Swarm Optimization Method

Particle swarm optimization (PSO) was first introduced by Kennedy and Eberhart in 1995 [14]. PSO uses a simple mechanism that mimics the behavior of flocking birds and fish to guide particles to find the global optimal solution [15]. PSO has been applied to various problems, such as training artificial neural networks, combinatorial optimization problems, and multi-objective optimization problems, and also provides better results in many fields. This is because of the hybridization of evolutionary metaheuristics, multi-objective concepts, and swarm theory [16]. The implementation of the PSO algorithm makes PSO a popular optimizer and has been successfully applied in various fields, including the optimization of many vehicles and the total travel distance in goods delivery problems [17], 0-1 knapsack problems [18], lecture scheduling [19], etc.

Theoretical studies and algorithm performance improvements are important and interesting to do. Some improvements that have been made include convergence analysis and optimization stability [20], [21], [22], and [23]. While research on improving PSO performance, including parameter studies, combinations with additional operations, and topological structures has been widely carried out [24], [25].

The PSO algorithm is based on the social behavior of a flock of birds or a school of fish. This social behavior consists of the actions of each individual in a population and the influence of other individuals in the population. An example of social behavior in a flock of birds is if a bird finds the right or shortest path to a food source, other birds in the population will follow the path even though their location is far away. In PSO, the bird population is called a swarm, and individual birds are called particles. Each particle moves at a speed adapted to the search area and always stores the best position ever reached. The position of the i-th particle in a flock is denoted by x_i while the best position of a particle's journey is called $Pbest_i$. Each particle will know the best position globally found by one of the members of the flock. The best position of all particles in the flock is called $Gbest$.

In PSO, a swarm of particles is represented as a set of problems, and each particle is associated with two vectors, namely, a velocity vector, where $V_i = [v_i^1, v_i^2, \dots, v_i^D]$ and a position vector $X_i = [x_i^1, x_i^2, \dots, x_i^D]$, where D is the dimension of the solution space. The algorithm is guided by personal experience ($Pbest$), overall experience ($Gbest$) and the current particle movement to determine the next position in the search space. Furthermore, the particle movement is accelerated by two factors c_1 and c_2 **[26]**. The mathematical model that describes the mechanism of updating the particle's status of velocity and position is as follows:

$$V_i^{t+1} = w\, V_i^t + c_1 r_1\left(Pbest_i^t - X_i^t\right) + c_2\, r_2\left(Gbest^t - X_i^t\right) \quad (1)$$
$$X_i^{t+1} = X_i^t + V_i^{t+1}$$

Where:
V_i^{t+1} : velocity of the individual i at iteration $t+1$
X_i^{t+1} : position of the individual i at iteration $t+1$
w : inertial weight parameter

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{t_{max}}\right) t \quad (2)$$

c_1, c_2 : acceleration constant (learning rate)
r_1, r_2 : random parameter 0 to 1
$Pbest_i^t$: local best of individual i at iteration t
$Gbest^t$: global best of all individuals.

This model will be simulated in a space with a certain dimension, with a number of iterations, so that in each iteration, the particle position will increasingly lead to the intended target (minimization or maximization of the function). This is done until the maximum iteration is reached, or another stopping criterion can also be used, namely, the value of the objective function or goal has converged to the optimum value. The following steps outline the use of the PSO algorithm.

1. Generate parameter values w_{min}, w_{max}, c_1 and c_2.
2. Initialize the position of each particle population X_i^t and the particle velocity V_i^t.
3. Set iteration $t = 1$.
4. Calculate the fitness value of each particle $F_i(t)$ based on the formula and model that have been determined according to the optimization problem. Then find the best fitness value $F_b(t)$.
5. Choose $Pbest_i^t = X_i^t$ and $Gbest^t = X_b^t$.
6. Calculate the inertial weight value w using **Equation (2)**.
7. Update the velocity value V_i^{t+1} and particle position X_i^t using **Equation (1)**.
8. Evaluate the fitness value $F_i(t+1) = f(X_i^{t+1})$ for each particle and find the best fitness value $F_{b1}(t+1)$.
9. Update the velocity value V_i^{t+1}, if $F_i(t+1) < F_i(t)$ then $Pbest_i^{t+1} = X_i^{t+1}$, otherwise $Pbest_i^{t+1} = X_i^t$.
10. Update the value of $Gbest^t$, if $F_{b1}(t+1) < F_b(t)$ then $Gbest^{t+1} = Pbest_{b1}^{t+1}$ and $b = b_1$, otherwise $Gbest^{t+1} = Gbest^t$.
11. If iteration $t < t_{max}$ then $t = t+1$. Then, if it has reached the final condition (reaching the optimum iteration value or the loop has reached a value that converges to the minimum function value), then the loop stops and the optimum value is obtained, but if not, then repeat from step 6.

The following is an example of a problem solved using the particle swarm optimization algorithm steps.

$$\min f(x) = (4 - x)^2; \quad -10 \le x \le 10 .$$

The following are the steps for the particle swarm optimization algorithm to solve the problem above:

1. Initialize the population size $N = 4$, determine the initial population randomly, for example, we get $X_1^0 = 7, X_2^0 = 2, X_3^0 = -1,\ X_4^0 = -5$.
2. Evaluate the value of the objective function for each particle X_i^0 for $i = 1,2,3,4$
   $f_1(0) = f(7) = 9,\ \ f_2(0) = f(2) = 4,\ f_3(0) = f(-1) = 25,\ f_4(0) = f(-5) = 81.$
3. Initialize the initial velocity for each particle, for example, $V_1^0 = V_2^0 = V_3^0 = V_4^0 = 0$.
   Set iteration $t = 1$ with $t = 1, 2, 3, 4$.
4. Find $Pbest_i^1$ and $Gbest^1$, for $i = 1,2,3,4$,
   $Pbest_1^1 = 7,\ Pbest_2^1 = 2,\ Pbest_3^1 = -1,\ Pbest_4^1 = -5,\ Gbest^1 = 2.$
5. Calculate the velocity of each particle with parameter values $w_{max} = 0.9, w_{min} = 0.4, c_1 = c_2 = 1$. Suppose the random values obtained are $r_1 = 0.3,\ r_2 = 0.6$ and $t_{max} = 4$.

   By using **Equation (2)**, $w = 0.9 - \left(\frac{0.9-0.4}{4}\right)1 = 0.775.$

   $V_i^1 = w\, V_i^0 + c_1 r_1\left(Pbest_i^1 - X_i^0\right) + c_2\, r_2\left(Gbest^1 - X_i^0\right)$
   $V_1^1 = (0.775)(0) + (1)(0.3\,)(7-7) + (1)(0.6)(2-7) = -3$
   $V_2^1 = (0.775)(0) + (1)(0.3\,)(2-2) + (1)(0.6)(2-2) = 0$
   $V_3^1 = (0.775)(0) + (1)(0.3\,)(-1+1) + (1)(0.6)(2+1) = 1.8$
   $V_4^1 = (0.775)(0) + (1)(0.3\,)(-5+5) + (1)(0.6)(2+5) = 4.2.$
   Meanwhile, the value of X_i^1 is $X_i^1 = X_i^0 + V_i^1$, so
   $X_1^1 = 7 - 3 = 4,\ X_2^1 = 2 + 0 = 2,\ X_3^1 = -1 + 1.8 = 0.8,\ X_4^1 = -5 + 4.2 = -0.8.$
6. Evaluate the value of the objective function for each particle X_i^1 for $i = 1,2,3,4$
   $f_1(1) = f(4) = 0,\ f_2(1) = f(2) = 4,\ f_3(1) = f(0.8) = 10.24,\ f_4(1) = f(-0.8) = 23.04.$
   Meanwhile, in the previous iteration, the objective function values obtained were as follows:
   $f_1(0) = f(7) = 9,\ \ f_2(0) = f(2) = 4,\ f_3(0) = f(-1) = 25,\ f_4(0) = f(-5) = 81.$
   The value of f in the previous iteration is not better so that $Pbest$ for each particle is equal to the value of X_i^1. $Gbest^2 = 4$.
   Advance to the next iteration $t = 2$.
7. Repeat step 4 to find $Pbest_i^2$ and $Gbest^2$, for $i = 1,2,3,4$,
   $Pbest_1^2 = 4,\ Pbest_2^2 = 2,\ Pbest_3^2 = 0.8,\ Pbest_4^2 = -0.8,\ Gbest^2 = 4.$
8. Calculate the velocity of each particle with parameter values $w_{max} = 0.9, w_{min} = 0.4, c_1 = c_2 = 1$. Suppose the random values obtained are $r_1 = 0.2,\ r_2 = 0.4$ and $t_{max} = 4$.

   By using **Equation (2)**, then $w = 0.9 - \left(\frac{0.9-0.4}{4}\right)2 = 0.65.$

   $V_i^2 = w\, V_i^1 + c_1 r_1\left(Pbest_i^2 - X_i^1\right) + c_2\, r_2\left(Gbest^2 - X_i^1\right)$
   $V_1^2 = (0.65)(-3) + (1)(0.2)(4-4) + (1)(0.4)(4-4) = -1.95$
   $V_2^2 = (0.65)(0) + (1)(0.2)(2-2) + (1)(0.4)(4-2) = 0.8$
   $V_3^2 = (0.65)(1.8) + (1)(0.2)(0.8-0.8) + (1)(0.4)(4-0.8) = 2.45$
   $V_4^2 = (0.65)(4.2) + (1)(0.2)(-0.8+0.8) + (1)(0.4)(4+0.8) = 4.65.$
   Meanwhile, the value of X_i^2 is $X_i^2 = X_i^1 + V_i^2$, so
   $X_1^2 = 4 - 1.95 = 2.05, X_2^2 = 2 + 0.8 = 2.8,\ X_3^2 = 0.8 + 2.45 = 3.25,\ X_4^2 = -0.8 + 4.65 = 3.85.$
9. Evaluate the value of the objective function for each particle X_i^2 for $i = 1,2,3,4$
   $f_1(2) = f(2.05) = 3.8025,\ f_2(2) = f(2.8) = 1.44,\ f_3(2) = f(3.25) = 0.5625,$
   $f_4(2) = f(3.85) = 0.0225$
   In the previous iteration, the objective function values obtained were as follows
   $f_1(1) = f(4) = 0,\ f_2(1) = f(2) = 4,\ f_3(1) = f(0.8) = 10.24,\ f_4(1) = f(-0.8) = 23.04.$
   When compared with the value of f in the previous iteration, there is one that is better, namely $f_1(1) = f(4) = 0$, so that the value of the objective function becomes

$f_1(2) = f(4) = 0$, $f_2(2) = f(2.8) = 1.44$, $f_3(2) = f(3.25) = 0.5625$,
$f_4(2) = f(3.85) = 0.0225$.
Continue to the next iteration $t = 3$.

10. Finding $Pbest_i^3$ and $Gbest^3$, for $i = 1,2,3,4$,
$Pbest_1^3 = 4$, $Pbest_2^3 = 2.8$, $Pbest_3^3 = 3.25$, $Pbest_4^3 = 3.85$, $Gbest^3 = 4$.
11. Calculate the velocity of each particle with parameter values $w_{max} = 0.9$, $w_{min} = 0.4$, $c_1 = c_2 = 1$. Suppose the random values obtained are $r_1 = 0.5$, $r_2 = 0.2$ and $t_{max} = 4$.

By using **Equation (2)**, $w = 0.9 - \left(\frac{0.9-0.4}{4}\right)3 = 0.525$

$V_i^3 = w\, V_i^2 + c_1 r_1 \left(Pbest_i^3 - X_i^2\right) + c_2\, r_2 \left(Gbest^3 - X_i^2\right)$
$V_1^3 = (0.525)(-1.95) + (1)(0.5)(4 - 2.05) + (1)(0.2)(4 - 2.05) = 0.34125$
$V_2^3 = (0.525)(0.8) + (1)(0.5)(2.8 - 2.8) + (1)(0.2)(4 - 2.8) = 0.66$
$V_3^3 = (0.525)(2.45) + (1)(0.5)(3.25 - 3.25) + (1)(0.2)(4 - 3.25) = 1.43625$
$V_4^3 = (0.525)(4.65) + (1)(0.5)(3.85 - 3.85) + (1)(0.2)(4 - 3.85) = 2.47125$
Next for $X_i^3 = X_i^2 + V_i^3$,
$X_1^3 = 2.05 + 0.34125 = 2.39125$, $X_2^3 = 2.8 + 0.66 = 3.46$, $X_3^3 = 3.25 + 1.43625 = 4.68625$,
$X_4^3 = 3.85 + 2.47125 = 6.32125$.
12. Evaluate the value of the objective function for each particle X_i^3 for $i = 1,2,3,4$,
$f_1(3) = f(2.39125) = 2.588077$, $f_2(3) = f(3.46) = 0.2916$
$f_3(3) = f(4.68625) = 0.470939$, $f_4(3) = f(6.32125) = 5.388202$.
In the previous iteration, the objective function value obtained were as follows
$f_1(2) = f(4) = 0$, $f_2(2) = f(2.8) = 1.44$, $f_3(2) = 0.5625$, $f_4(2) = f(3.85) = 0.0225$.
A better value has been obtained for $f_1(2)$ and $f_4(2)$ so that the objective function value becomes
$f_1(3) = f(4) = 0$, $f_2(3) = f(3.46) = 0.2916$, $f_3(3) = f(4.68625) = 0.470939$,
$f_4(3) = f(3.85) = 0.0225$.
Continue to the next iteration $t = 4$.
13. Finding $Pbest_i^4$ and $Gbest^4$, for $i = 1,2,3,4$
$Pbest_1^4 = 4$, $Pbest_2^4 = 3.46$, $Pbest_3^4 = 4.68625$, $Pbest_4^4 = 3.85$, $Gbest^4 = 4$.

Therefore the solution to the problem above is $x = 4$ and $f(x) = 0$.

Some stopping conditions that can be used in PSO are [23]:

1. Stop when the number of iterations reaches the maximum number of iterations.
2. Stop when an acceptable solution is found.
3. Stop when there is no progress after several iterations.

After five independent runs using mathematical software, with parameter values $N = 4$, $c_1 = c_2 = 1$, $w_{max} = 0.9$, $w_{min} = 0.4$ and $t_{max} = 4$, then only the best result will be taken. The results obtained are the objective function values $f(x) = 0$ and $Gbest = 4$, with an execution time of 0.730687 seconds. **Figure 1** shows the PSO convergence graph of the problem.

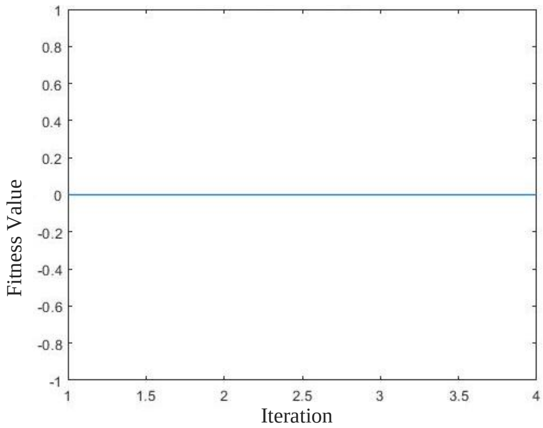

Figure 1. PSO Convergence Graph of the Example Problem

2.2 Data

Data on the standard width of available aluminum and the width of aluminum required in this work were obtained from Suliman and Octarina [27], [28].

A company that produces aluminum paper, called Aluminum Rolling Mill Company (ARMCO), receives an order from a customer. The problem faced by ARMCO is getting the right cutting pattern to produce minimal cutting waste. The standard width of the available aluminum paper rolls is 130 cm and 100 cm, with a certain standard length. The types of widths ordered consist of 50 cm, 40 cm, 30 cm, and 20 cm.

Table 1. Aluminum Roll Size Width and Quantity Requested

Roll Size Width (cm)	Number of Requests
50	64
40	58
30	70
20	48

3. RESULTS AND DISCUSSION

3.1 Combination Cutting Patterns

The cutting is planned by arranging several patterns by considering the following criteria:

1. The total width of the cutting result must be the same or shorter than 130 cm or 100 cm,
2. The width of the remaining cutting is less than 20 cm, and
3. There is no duplication of patterns.

The resulting pattern combinations are presented in Table 2.

Table 2. Combination of 100 cm and 130 cm Aluminum Roll Cutting Patterns

StandardWidth (L_i)	Pattern (j)	Aluminum Roll Width (cm) (l_k)				Trim Loss (cm)
		50	40	30	20	
100 cm	1	0	0	0	5	0
	2	0	0	1	3	10
	3	0	0	2	2	0
	4	0	0	3	0	10
	5	0	1	0	3	0
	6	0	1	1	1	10
	7	0	1	2	0	0
	8	0	2	0	1	0
	9	1	0	0	2	10
	10	1	0	1	1	0
	11	1	1	0	0	10
	12	2	0	0	0	0

StandardWidth (L_i)	**Pattern (j)**	**Aluminum Roll Width (cm) (l_k)**				**Trim Loss (cm)**
		50	**40**	**30**	**20**	
130 cm	13	0	0	0	6	10
	14	0	0	1	5	0
	15	0	0	2	3	10
	16	0	0	3	2	0
	17	0	0	4	0	10
	18	0	1	0	4	10
	19	0	1	1	3	0
	20	0	1	2	1	10
	21	0	1	3	0	0
	22	0	2	0	2	10
	23	0	2	1	1	0
	24	0	3	0	0	10
	25	1	0	0	4	0
	26	1	0	1	2	10
	27	1	0	2	1	0
	28	1	1	0	2	0
	29	1	1	1	0	10
	30	1	2	0	0	0
	31	2	0	0	1	10
	32	2	0	1	0	0
Number of Requests		64	58	70	48	

3.2 Mathematical Model

The cutting problem can be formulated into a mathematical model as follows:

$$min\ Z = \sum_{j=1}^{32} X_j$$

Constraint

$$\sum_{j=1}^{32} P_{jk}\, X_j \geq\ B_k\,, \qquad\qquad X_j \geq 0$$

Sets and Indexes

I : {1, 2}, set of standard width variations available with index i

J : {1, 2, 3, . . ., 32}, set of aluminum roll cutting pattern with index j

K : {1, 2, 3, 4}, set of variations in the width of the aluminum roll required with index k

Parameter

L_i : width of aluminum roll i-th variant

l_k : the required width of the k-th variant of the aluminum roll

P_{jk} : the number of small rolls with width k-th on the j-th cutting pattern

B_k : the number of requests for small rolls with width k-th

Decision variables

X_j : the number of aluminum rolls cut in the j-th cutting pattern

Therefore, we have:

Objective Function

$$min\ Z = \sum_{j=1}^{32} X_j$$

Constraint

$X_9 +\ X_{10} +\ X_{11} + 2X_{12} + X_{25} + X_{26} + X_{27} +\ X_{28} +\ X_{29} + X_{30} +\ 2X_{31} + 2X_{32} \geq 64$

$X_5 + X_6 + X_7 + 2X_8 + X_{11} + X_{18} + X_{19} + X_{20} + X_{21} + 2X_{22} + 2X_{23} + 3X_{24} + X_{28} + X_{29} + 2X_{30} \geq 58$
$X_2 + 2X_3 + 3X_4 + X_6 + 2X_7 + X_{10} + X_{14} + 2X_{15} + 3X_{16} + 4X_{17} + X_{19} + 2X_{20} + 3X_{21} + X_{23} + X_{26} + 2X_{27} + X_{29} + X_{32} \geq 70$
$5X_1 + 3X_2 + 2X_3 + 3X_5 + X_6 + X_8 + 2X_9 + X_{10} + 6X_{13} + 5X_{14} + 3X_{15} + 2X_{16} + 4X_{18} + 3X_{19} + X_{20} + 2X_{22} + X_{23} + 4X_{25} + 2X_{26} + X_{27} + 2X_{28} + X_{31} \geq 48$
$X_j \geq 0; j = 1, 2, \ldots, 32.$

3.3 Solving Cutting Stock Problem Using Particle Swarm Optimization and Exact Method

Based on the problem example above, we solve the cutting problem with the mathematical model in Section 3.2 and the cutting combination data contained in **Table 2** using the heuristic PSO method and the exact method using the help of mathematical software and Lingo 11.0.

The PSO method begins with parameter initialization first; the parameters used consist of c_1, c_2, N (population size), t, w_{max}, and w_{min}, with $w_{max} = 0.9$ and $w_{min} = 0.4$, so the parameter variations used in this case are:

1. Different parameters c_1 and fixed parameters c_2, N, t.
2. Different parameters c_2 and fixed parameters c_1, N, t.
3. Different parameters N and fixed parameters c_1, c_2, t.
4. Different parameters t and fixed parameters c_1, c_2, N.

For each parameter variation, the algorithm will be run independently ten times, then from all the results, only the best result will be taken. The results are displayed in **Table 3** to **Table 6**. PSO Convergence Graph for the best result for each case is presented in **Figure 2** to **Figure 5**.

Table 3. Comparison of Cutting Results with Different Parameters c_1

c_1	c_2	N	t	Total Cuts	Time (seconds)
1	2	100	500	76	10.423897
2	2	100	500	74	10.279768
2.05	2	100	500	68	8.000135

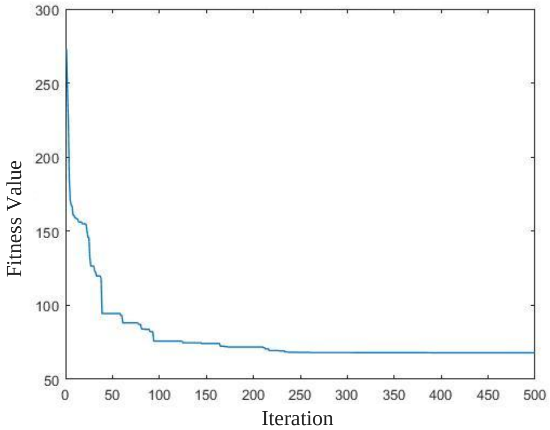

Figure 2. PSO Convergence Graph With Parameter Values $c_1 = 2.05$, $c_2 = 2$, $N = 100$, and $t = 500$

Table 4. Comparison of Cutting Results with Different Parameters c_2

c_1	c_2	N	t	Total Cuts	Time (seconds)
2	1	100	500	72	12.745466
2	2	100	500	74	10.279768
2	2.05	100	500	68	7.493547

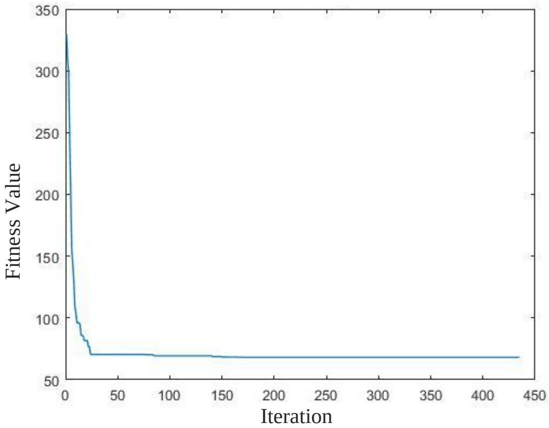

Figure 3. PSO Convergence Graph With Parameter Values $c_1 = 2$, $c_2 = 2.05$, $N = 100$, and $t = 500$

Table 5. Comparison of Cutting Results with Different Parameters N

c_1	c_2	N	t	Total Cuts	Time (seconds)
2	2	30	500	76	5.067679
2	2	50	500	71	5.258329
2	2	100	500	74	10.279768

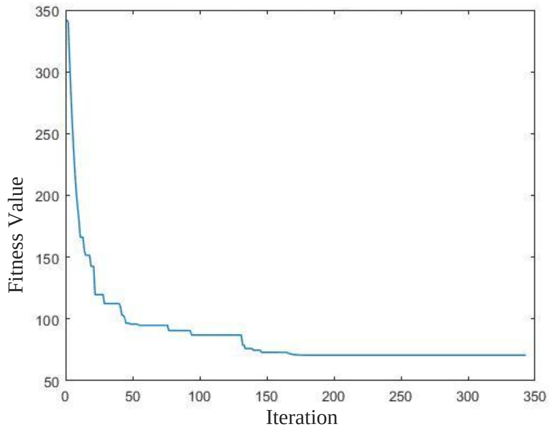

Figure 4. PSO Convergence Graph With Parameter Values $c_1 = 2$, $c_2 = 2$, $N = 50$, and $t = 500$

Table 6. Comparison of Cutting Results with Different Parameters t

c_1	c_2	N	t	Total Cuts	Time (seconds)
2.05	2.05	100	100	75	3.873903
2.05	2.05	100	500	71	10.381683
2.05	2.05	100	1000	67	8.971090

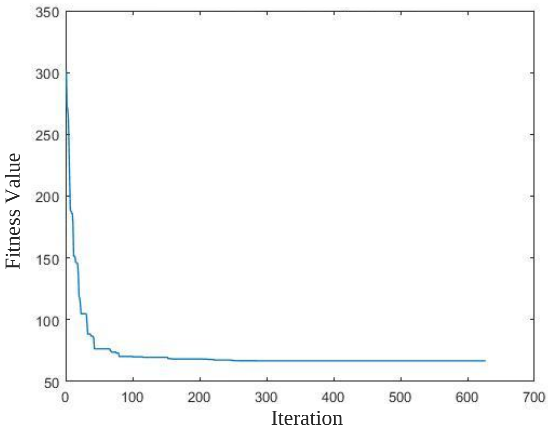

Figure 5. PSO Convergence Graph With Parameter Values $c_1 = 2.05$, $c_2 = 2.05$, $N = 100$, and $t = 1000$

After conducting several experiments with each different parameter value, the minimum total cutting of aluminum rolls was 67 rolls with an execution time of 8.971090 seconds. Table 7 shows the number of aluminum rolls that will be used to meet customer demand. A total of 32 patterns can be made from aluminum rolls with a length of 100 cm and 130 cm (as in Table 2). Of the 32 patterns, four patterns were obtained that were used to obtain minimum cutting residue/trim loss, namely, patterns X_1, X_{23}, X_{27}, and X_{32}.

Table 7. The Results of the Cutting Pattern Using PSO

Standard Width	Pattern	Aluminum Roll Width (cm)				Trim Loss (cm)	Number of Aluminum Rolls Used
		50	40	30	20		
100 cm	1	0	0	0	5	0	3
130 cm	23	0	2	1	1	0	29
	27	1	0	2	1	0	6
	32	2	0	1	0	0	29
Total		64	58	70	50	0	67

Based on the results obtained above, the total number of aluminum rolls used is obtained from the number of selected patterns. Table 7 shows that the total number of aluminum rolls used is 67 rolls with a combination of patterns used, namely 3 rolls for pattern 1, 29 rolls for pattern 23, 6 rolls for pattern 27, and 29 rolls for pattern 32. Then, from each roll width will produce 64 pieces for a width of 50 cm, 58 pieces for a width of 40 cm, 70 pieces for a width of 30 cm and 50 pieces, for a width of 20 cm. These results, when compared with the number of orders, will produce an excess number of pieces, namely two pieces for a width of 20 cm. The excess cutting results can be used for the next order. Then the remaining cutting produces 0 cm, meaning there is no material left or wasted. Furthermore, the output results above will be compared with the exact method using LINGO 11.0 software.

Table 8. The Results of the Cutting Pattern Using Exact Method

Standard Width	Pattern	Aluminum Roll Width (cm)				Trim Loss (cm)	Number of Aluminum Rolls Used
		50	40	30	20		
130 cm	16	0	0	3	2	0	18
	25	1	0	0	4	0	3
	30	1	2	0	0	0	29
	32	2	0	1	0	0	16
Total		64	58	70	48	0	66

Based on the results obtained in Table 8, the number of aluminum rolls used is 66 rolls with a combination of patterns used, namely $X_{16} = 18, X_{25} = 3, X_{30} = 29$, and $X_{32} = 16$. Then from each roll width will produce 64 pieces for a width of 50 cm, 58 pieces for a width of 40 cm, 70 pieces for a width of 30 cm and 48 pieces for a width of 20 cm. These results are sufficient to meet customer orders and do not produce trim loss. So, from the two methods, it can be concluded that both have met the order constraints and do not produce trim loss. The difference lies in the selected patterns, and in the PSO method, there are excess cuts, namely, two cuts for a width of 20 cm.

4. CONCLUSION

This paper shows that the problem of one-dimensional cutting of aluminum rolls can be solved using the PSO method. The problem of one-dimensional cutting of aluminum rolls was initially modeled in a mathematical model in the form of linear optimization. The implementation of the model in cutting aluminum rolls produced an optimum solution, namely 67 aluminum rolls to be used, with a combination of selected patterns, namely three rolls for pattern one, 29 rolls for pattern 23, six rolls for pattern 27, and 29 rolls for pattern 32. Then the remaining cutting results in 0 cm, meaning there is no material left or wasted. By using the exact method, the number of aluminum rolls used is 66 rolls with a combination of patterns used, namely 18 rolls for pattern 16, 3 rolls for pattern 25, 29 rolls for pattern 30, and 16 rolls for pattern 32. The two methods have met the order constraints and do not produce trim loss. The difference lies in the selected patterns, and in the PSO method, there are excess cuts, namely, two cuts for a width of 20 cm.

AUTHOR CONTRIBUTIONS

Bib Paruhum Silalahi: Writing - Original Draft, Writing - Review and Editing. Danupun Visuwan: Data Curation, Funding Acquisition, Writing - Original Draft, Writing - Review and Editing. Siti Aminah: Data curation, Investigation. Hidayatul Mayyani: Project administration, Supervision. Amril Aman: Resources. All authors discussed the results and contributed to the final manuscript.

FUNDING STATEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ACKNOWLEDGMENT

The authors are indebted to the colleagues in the operations research division, School of Data Science, Mathematics, and Informatics, IPB University, for their support during this research.

CONFLICT OF INTEREST

The authors declare no conflicts of interest regarding this study.

REFERENCES

[1] D. Tanir, O. Ugurlu, A. Guler, and U. Nuriyev, "ONE-DIMENSIONAL CUTTING STOCK PROBLEM WITH DIVISIBLE ITEMS: A CASE STUDY IN STEEL INDUSTRY," *Turkish World Mathematical Society Journal of Applied and Engineering Mathematics*, vol. 9, no. 3, pp. 473–484, 2019.

[2] B. P. Silalahi, F. Hanum, F. Setyawan, and P. T. Supriyo, "ONE-DIMENSIONAL CUTTING STOCK PROBLEM THAT MINIMIZES THE NUMBER OF DIFFERENT PATTERNS," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 16, no. 3, 2022, doi: https://doi.org/10.30598/barekengvol16iss3pp805-814

[3] O. Ayadi, M. Masmoudi, M. Ben Ameur, and F. Masmoudi, "A NEW PSO-BASED ALGORITHM FOR TWO-DIMENSIONAL NON-GUILLOTINE NON-ORIENTED CUTTING STOCK PROBLEM," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 376–393, 2017, doi: https://doi.org/10.1080/08839514.2017.1346966.

[4] F. Clautiaux, R. Sadykov, F. Vanderbeck, and Q. Viaud, "PATTERN-BASED DIVING HEURISTICS FOR A TWO-DIMENSIONAL GUILLOTINE CUTTING-STOCK PROBLEM WITH LEFTOVERS," *EURO Journal on Computational Optimization*, vol. 7, no. 3, 2019, doi: https://doi.org/10.1007/s13675-019-00113-9 .

[5] M. Atika, B. P. Silalahi, and F. Bukhari, "PATTERN GENERATION FOR THREE DIMENSIONAL CUTTING STOCK PROBLEM," *JTAM (Jurnal Teori dan Aplikasi Matematika)*, vol. 6, no. 4, 2022, doi: https://doi.org/10.31764/jtam.v6i4.9933.

[6] P. C. Gilmore and R. E. Gomory, "A LINEAR PROGRAMMING APPROACH TO THE CUTTING STOCK PROBLEM—PART II," *Oper Res*, vol. 11, no. 6, 1963, doi: https://doi.org/10.1287/opre.11.6.863.

[7] P. C. Gilmore and R. E. Gomory, "A LINEAR PROGRAMMING APPROACH TO THE CUTTING-STOCK PROBLEM," *Oper Res*, vol. 9, no. 6, 1961, doi: https://doi.org/10.1287/opre.9.6.849.

[8] S. A. de Araujo, K. C. Poldi, and J. Smith, "A GENETIC ALGORITHM FOR THE ONE-DIMENSIONAL CUTTING STOCK PROBLEM WITH SETUPS," *Pesquisa Operacional*, vol. 34, no. 2, 2014, doi: ttps://doi.org/10.1590/0101-7438.2014.034.02.0165.

[9] Bambang Santoso and Heri Haerudin, "GENETIC ALGORITHM APPROACH FOR CUTTING STOCK PROBLEMS IN CONSTRUCTION INDUSTRIES," *International Journal of Integrative Sciences*, vol. 2, no. 7, 2023, doi: https://doi.org/10.55927/ijis.v2i7.4810.

[10] A. R. Botsali and K. Alaykiran, "ANALYSIS OF TSP: SIMULATED ANNEALING AND GENETIC ALGORITHM APPROACHES," *International Journal of Computational and Experimental Science and Engineering*, vol. 6, no. 1, 2020, doi: https://doi.org/10.22399/ijcesen.637445.

[11] Z. Zheng and Y. Li, "AUTOMATED BOARDING LAYOUT AND WASTE OPTIMIZATION FOR EXTERIOR WALLS USING SIMULATED ANNEALING ALGORITHM," in *Mechanisms and Machine Science*, 2024. doi: https://doi.org/10.1007/978-3-031-44947-5_58.

[12] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "PARTICLE SWARM OPTIMIZATION: A COMPREHENSIVE SURVEY," *IEEE Access*, vol. 10, 2022, doi: https://doi.org/10.1109/ACCESS.2022.3142859.

[13] K. Ren, L. Jia, J. Huang, and M. Wu, "RESEARCH ON CUTTING STOCK OPTIMIZATION OF REBAR ENGINEERING BASED ON BUILDING INFORMATION MODELING AND AN IMPROVED PARTICLE SWARM OPTIMIZATION ALGORITHM," *Developments in the Built Environment*, vol. 13, 2023, doi: https://doi.org/10.1016/j.dibe.2023.100121.

[14] R. Eberhart and J. Kennedy, "NEW OPTIMIZER USING PARTICLE SWARM THEORY," in *Proceedings of the International Symposium on Micro Machine and Human Science*, 1995. doi: 10.1109/mhs.1995.494215.

[15] N. Franken and A. P. Engelbrecht, "PARTICLE SWARM OPTIMIZATION APPROACHES TO COEVOLVE STRATEGIES FOR THE ITERATED PRISONER'S DILEMMA," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, 2005, doi: https://doi.org/10.1109/TEVC.2005.856202.

[16] G. Ben Lagha, N. Dahmani, and S. Krichen, "PARTICLE SWARM OPTIMIZATION APPROACH FOR RESOLVING THE CUTTING STOCK PROBLEM," in *2014 International Conference on Advanced Logistics and Transport, ICALT 2014*, 2014. doi: https://doi.org/10.1109/ICAdLT.2014.6866321.

[17] B. P. Silalahi, K. Fatihin, P. T. Supriyo, and S. Guritman, "ALGORITME SWEEP DAN PARTICLE SWARM OPTIMIZATION DALAM OPTIMISASI RUTE KENDARAAN DENGAN KAPASITAS," *Jurnal Matematika Integratif*, vol. 16, no. 1, p. 29, Apr. 2020, doi: https://doi.org/10.24198/jmi.v16.n1.27474.29-40.

[18] B. P. Silalahi, M. Novanto, and P. T. Supriyo, "ALGORITME MIGRATING BIRDS OPTIMIZATION DAN ALGORITME PARTICLE SWARM OPTIMIZATION: PENYELESAIAN MASALAH KNAPSACK 0-1," *PYTHAGORAS Jurnal Pendidikan Matematika*, vol. 17, no. 1, 2022, doi: https://doi.org/10.21831/pythagoras.v17i1.35660.

[19] F. Yunita, Pranowo, and A. J. Santoso, "HYBRID MODEL OF PARTICLE SWARM AND ANT COLONY OPTIMIZATION IN LECTURE SCHEDULE PREPARATION," in *AIP Conference Proceedings*, 2018. doi: https://doi.org/10.1063/1.5042895.

[20] M. Clerc and J. Kennedy, "THE PARTICLE SWARM-EXPLOSION, STABILITY, AND CONVERGENCE IN A MULTIDIMENSIONAL COMPLEX SPACE," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, 2002, doi: https://doi.org/10.1109/4235.985692.

[21] I. C. Trelea, "THE PARTICLE SWARM OPTIMIZATION ALGORITHM: CONVERGENCE ANALYSIS AND PARAMETER SELECTION," *Inf Process Lett*, vol. 85, no. 6, 2003, doi: https://doi.org/10.1016/S0020-0190(02)00447-7.

[22] V. Kadirkamanathan, K. Selvarajah, and P. J. Fleming, "STABILITY ANALYSIS OF THE PARTICLE DYNAMICS IN PARTICLE SWARM OPTIMIZER," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, 2006, doi: https://doi.org/10.1109/TEVC.2005.857077.

[23] F. Van Den Bergh and A. P. Engelbrecht, "A STUDY OF PARTICLE SWARM OPTIMIZATION PARTICLE TRAJECTORIES," *Inf Sci (N Y)*, vol. 176, no. 8, 2006, doi: https://doi.org/10.1016/j.ins.2005.02.003.

[24] R. A. Krohling and L. Dos Santos Coelho, "COEVOLUTIONARY PARTICLE SWARM OPTIMIZATION USING GAUSSIAN DISTRIBUTION FOR SOLVING CONSTRAINED OPTIMIZATION PROBLEMS," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 6, 2006, doi: https://doi.org/10.1109/TSMCB.2006.873185.

[25] B. Liu, L. Wang, and Y. H. Jin, "AN EFFECTIVE PSO-BASED MEMETIC ALGORITHM FOR FLOW SHOP SCHEDULING," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, 2007, doi: https://doi.org/10.1109/TSMCB.2006.883272.

[26] M. Nabab, "PARTICLE SWARM OPTIMIZATION: ALGORITHM AND ITS CODES IN MATLAB," *ResearchGate*, no. 1, 2016.

[27] S. M. A. Suliman, "PATTERN GENERATING PROCEDURE FOR THE CUTTING STOCK PROBLEM," *Int J Prod Econ*, vol. 74, no. 1–3, 2001, doi: https://doi.org/10.1016/S0925-5273(01)00134-7.

[28] S. Octarina, P. B. J. Bangun, and M. Avifana, "REDUKSI POLA PEMOTONGAN KERTAS PADA CUTTING STOCK PROBLEM (CSP) SATU DIMENSI," *Prosiding Annual Research Seminar 2016*, vol. 2, no. 1, 2016.