Journal of Petroleum Science and Technology 12(1): 33, 2022, Pages 21-35
DOI:10.22078/JPST.2022.4869.1814

Journal of Petroleum Science and Technology
Research Paper
https://jpst.ripi.ir/

Optimization of Hole Cleaning in Deviated Wells Using Metaheuristic Algorithms

Mahdi Nazari Sarem and Arash Ebrahimabadi*
Department of Petroleum, Mining and Material Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran

Abstract
Field experience shows that the cutting transportation and hole-cleaning phenomena are essential during the drilling phase. Particularly in directional drilling, when the accumulation of cutting has caused some drilling problems such as drill string sticking, formation failure, slow rate of penetration, drill bit abrasion, and the like. Through the study, a novel method for efficient hole cleaning, considering different parameters such as flow rate, the drill bit nozzles' flow area, the consistency and flow behavior indices in the same time using PSO and ACO algorithms were implemented. Moreover, Power Law has been considered for the fluid rheology model. Based on this, the research parameter shows that the PSO algorithm is much more accurate than the ACO algorithm, improving objective function by 50% and 4%, respectively. The performance of each algorithm was evaluated, and the results show that hole cleaning has been significantly improved. The flow rate and the bit nozzle size, which play key roles, were selected as optimization variables. Effective parameters on hole cleaning were evaluated, and the results before and after optimization showed a significant improvement in the model. The PSO and ACO algorithms have been coded in MATLAB software, and the results are compared to the results of the ant colony. The amount of PV and YP has an inverse effect on the increment of minimum velocity required for cutting transport. Various model analyses reveal that the PSO algorithm is more accurate and robust than the Ant colony algorithm.

Keywords: Optimization, Hole Cleaning, Cutting Bed Height, PSO Algorithm, Ant Colony Algorithm.

Introduction
The growth of the demand for energy in the world has increased the need for drilling deeper wells with high deviation degrees. Hole cleaning is one of the most challenging problems in highly deviated wells. Causing problems such as stuck pipe, hole pack-off, high ECD, formation fracture, and cutting accumulation, poor hole cleaning can create hard situations. The key to achieving a successful hole cleaning is based on combining optimum drilling hydraulics with new drilling methods. If the accumulation of drilling cuttings is not evaluated properly, the situation will cause side-tracking of the well or packing it off. As the hole-cleaning problem is often encountered, it needs to be managed appropriately in the drilling operation [1].
The drilling fluid capability of cutting transportation from the borehole to the surface depends on fluid hydraulics and rheology. Proper hole cleaning is important in increasing the drilling rate and ROP. As soon as the rock was broken by the bit, the bit nozzle transports the cuttings to the surface by jet force. The effective parameters in cutting transfer and hole cleaning include: drilling fluid density, borehole size, hole angle, drill pipe eccentricity, cutting diameter, circulation drill pipe, flow rate, ROP, drilling fluid rheology, etc. Flow rate and drilling fluid rheology are two key parameters that affect the efficiency of cutting transport. The optimum setting of these parameters is a key factor in controlling the cutting transfer, and it depends on their controllability through drilling operations. For example, drill pipe eccentricity significantly affects cutting transport. However, controlling or estimating eccentricity is complicated in drilling operations [2].
This study addresses the hole-cleaning problems using an unconventional practice and a novel method. In addition to conventional methods considering maximizing hydraulic horsepower and jet impact force, a new criterion was achieved in which flow rate and cutting transport are considered. The reason for using this new criterion is that the conventional methods are not successful in deep and ERD wells [3].
Artificial intelligence algorithms have been extensively developed and used in various engineering problems in recent decades. In this study, an optimization problem

**Corresponding author:* Arash Ebrahimabadi, Department of Petroleum, Mining and Material Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran
E-mail addresses: a.ebrahimabadi@iauctb.ac.ir
Received 2022-07-28, Received in revised form 2022-09-29, Accepted 2022-10-18, Available online 2022-12-13

for the hole-cleaning parameters was addressed with the particle swarm optimization algorithm. The developed model for the optimization problem was used under different constraints to achieve the optimum value for the objective function (here, minimization of the cutting bed height). This research was conducted in one of the most famous oil fields in Iran, one of the main problems of which is hole cleaning wells.

Materials and Methods

The Effect of Various Parameters on Minimum Critical Velocity Required for Cutting Transport

The Effect of Inclination Angle

Figure 1 depicts the impact of various deviation angles on minimum critical velocity cutting transport. As can be seen, increasing the deviation angle will enhance the minimum velocity required for cutting transport; in other words, by increasing the deviation angle, a higher velocity is needed for the transportation of the cuttings. However, it will decrease slightly in about 80 and 90 degrees inclinations, and lower velocity will be required [4].

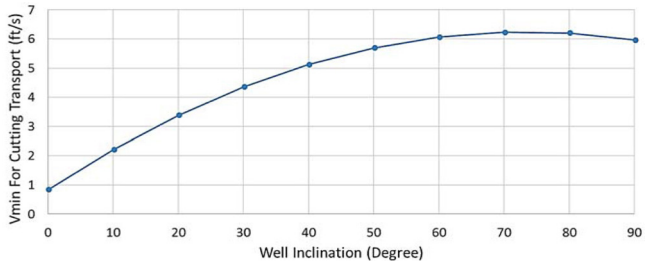

Fig. 1 The effect of inclination angle.

Drilling Fluid Properties

Figure 2 depicts PV and YP variations effect on the required velocity for cutting transport. From this figure, it can be concluded that an increase in PV and YP will decrease the minimum required velocity for cutting transport in short deviated wells [5]. Therefore, to enhance the cutting transport capability of the fluid in vertical wells, YP to PV ratio should be increased [6]. However, for highly deviated wells such as 60 degrees, it is vice versa, meaning that with simultaneous increasing of YP and PV, the minimum velocity required for cutting transport will increased, and consequently, the capability of the cutting transport is decreased [7].

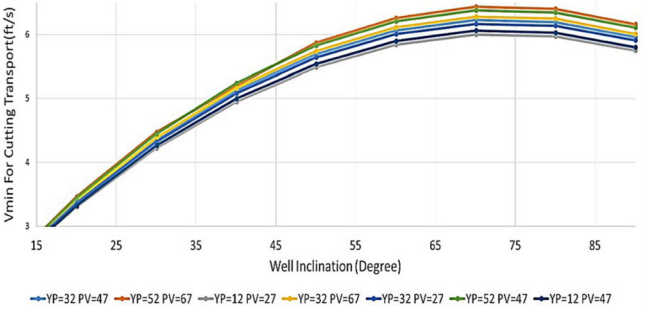

Fig. 2 Drilling fluid properties.

Drilling Penetration Rate

The drilling penetration rate is another factor that can be changed to reduce the minimum velocity required for cutting transport. A high penetration rate produces more drilling cutting; thus, more velocity is needed for cutting transport. As a result, more flow rate is required for cutting transport. In Figure 3, the decline in drilling penetration rate from 80ft/hr to 20ft/hr reduce minimum velocity required for cutting transport is shown [8].

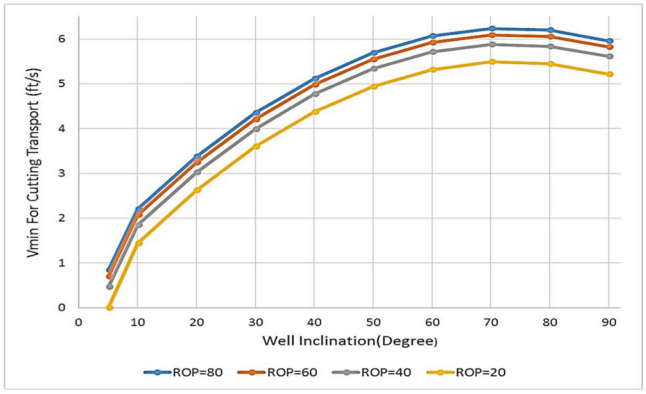

Fig. 3 Drilling penetration rate.

Hole Cleaning Optimization

While planning or drilling a deviated or horizontal well, two key parameters must be determined: the minimum flow rate required for drilled cuttings to the surface and the rheological properties of the drilling fluid. Different parameters are involved in optimizing hole cleaning (Figure 4). It seems obvious, but many incidents so far are linked to poor hole-cleaning practices. However, to optimize hole cleaning efficiency in deviated and horizontal wells, a balance must be struck between minimizing particle settling velocity and promoting fluid velocity under eccentric drill pipe.

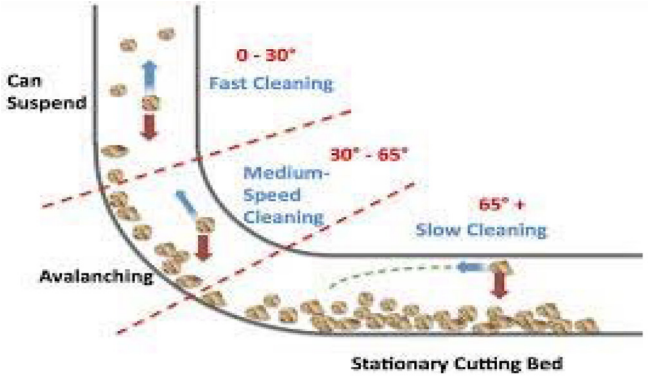

Fig. 4 Schematic of hole cleaning problem in deviated well.

Calculation of Cutting Bed Height (Optimization Objective)

When drilling a deviated well, usually, a layer is formed from the accumulation and deposition of drilling cuttings in the low side of the wellbore. If the height of this layer is high, it can cause problems such as high torque and drag and stuck pipe, indicating poor hole cleaning. Thus, cutting bed height can be used as a criterion for the analysis of hole cleaning. The cutting bed height should be maintained as low as possible to minimize drilling operation problems. Dimensionless cutting bed height is calculated as follows [9]:

$$H = \frac{100 T_{cb}}{D_h} \quad (1)$$

The formula for calculating cutting bed height developed by Zhou and Wang is as follows [10]:

$$T_{cb} = 0.015D_h(1000\mu_e + 194.48\mu_e^{0.5})(1+0.587\varepsilon)(V_{cr} - V_a) + D_h(0.0001N^2 - 0.35468N + 0.16236N \times V_a - 0.09465N \times \varepsilon + 0.00034N \times V_a \times \varepsilon)/100 \quad (2)$$

$$\mu_e = K\left[(2n+1)/(3n)\right]^n (D_h - d_{po})^{1-n}(12V_a)^{n-1}/1000^n \quad (3)$$

$$V_{cr} = 40.09\left[\frac{(\rho_s - \rho_f)}{\rho_f} d_s\right]^{0.667}\left[\frac{1+0.71\theta+0.55\sin(2\theta)}{(\rho_f \mu_e)^{0.333}}\right] \quad (4)$$

where is dimensionless cutting bed height, H is cutting bed height, μ_e is the diameter of the open hole, D_h is the effective viscosity, ε is the eccentricity, V_{cr} is the critical velocity, is the average velocity, V_a is the rotary speed of the drill pipes, K is the consistency, n, is the flw behavior index, d_{po} is the outer diameter of the drill pipe, is the cutting density, ρ_s is the fluid density, ρ_f is the diameter of the drill cuttings, and θ is the inclination angle of the hole.

Variable and Constraint

With respect to current research work, Figure 5 illustrates the profile of the studied well. Moreover, the data used through the optimization project listed in Table 1.

For this optimization, three constraints must be considered:

• Maximum pressure loss in the system

The total pressure loss includes pressure drop at the drill bit and parasitic pressure losses, in which parasitic pressure is pressure loss in the drill collars, drill pipes, annular space, and surface equipment. Total pressure loss can be expressed as:

$$\Delta P_{loss} = \Delta P_{bit} + \Delta P_{\text{parasitic}} \quad (5)$$

where ΔP_{loss} is the total pressure loss of the circulating system, ΔP_{bit} is the pressure loss at the drill bit, and the parasitic pressure loss in the system is $\Delta P_{Parasitic}$.

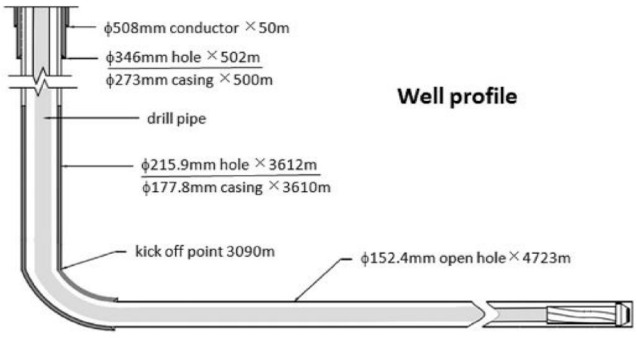

Fig. 5 Profile of the well [10].

Table 1 The data used in this project.

Parameter	Value
Drill Pipe Diameter	m 0.1016
Drilling Fluid Density	kg/m^3 10^3×1.2
Flow Behavior Index	0.5442
Drill Pipe Rotary Speed	rot/min 50
Max Pressure Pump	Pa 10^7×2.82
Min jet velocity	m/s 40
Max Flow Rate	m^3/s 0.0234

In drilling practice, maximum pressure loss is limited by pump capacity and pressure bearing capacity of equipment, and if pressure loss exceeds pressure bearing capacity, the facility will fail. Therefore, the maximum pressure loss in the system must be lower than the equipment's pump capacity and pressure bearing capacity to avoid causing damage to them.

The maximum allowable pressure loss can be written as follows:

$$\Delta P_{loss} < P_{\max} \quad (6)$$

• The maximum flow rate in the system

The maximum flow rate must be lower than the maximum pump horsepower to be able to create the desired flow rate, so the maximum flow rate constraint can be written as follows:

$$Q < Q_{\max} \quad (7)$$

where Q is the flow rate, and Q_{max} is the maximum allowable flow rate that the pump can provide.

• Minimum jet velocity

After breaking rocks by the bit, the cutting must be removed from the borehole and transported to the surface. Jet velocity from the bit nozzle should be high enough to do this task and remove the cutting. Thus the minimum jet velocity can be written as follows:

$$V_{jet} > V_{jet\,\min} \quad (8)$$

where V_{jet} is the jet velocity, and $V_{jet\ min}$ is the minimum velocity of the jet.

Optimization Description

Objective Function Minimum is obtained using the following equation: ($H = \frac{100T_{cb}}{D_t}$).

Constraints: the first constraint is $\Delta P_{loss} < P_{\max}$ the second constraint $Q < Q_{\max}$ and the third constraint is $V_{jet} > V_{jet\,\min}$

Basic of Ant Colony Algorithm

In Figure 6, the basic idea of ant colony optimization is illustrated. The left picture shows moving ants directly to the food (i.e., objective). The middle picture shows the status after a barrier is inserted between the nest and the food (i.e., objective). To pass through the barrier, at first, each ant chooses to turn left or right randomly. Let us presume that ants move at the same speed, depositing pheromones in the trail uniformly. However, the ants that turn left will reach the food sooner, while the ants that go around the barrier turning right will follow a longer path and take longer to circumvent the barrier. Finally, the pheromone accumulates faster in the shorter path. Since ants prefer to follow trails with larger amounts of pheromone, in the long run, all the ants converge to the shorter path around the barrier, as shown in Figure 6 [11,12].

The Proposed Ant Colony Algorithm

The optimization objective of this study is dimensionless cutting bed height, and (100H) is used as the pheromone accumulation value. For preliminary design, ant colony algorithm is considered in discrete domains, but when the algorithm is applied in continuous domains, it should be modified [14].

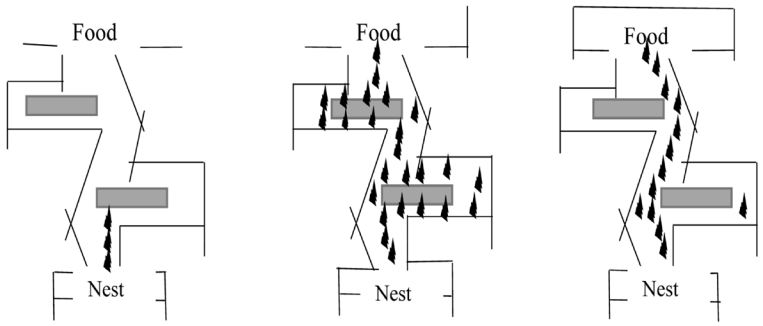

Fig. 6 Basic ACO [13].

For example, the probability of selection is calculated by accumulating the fitness value, which is related to the optimization objective value rather than the distance between two discrete cities in the TSP problem. When the optimization is constrained, it should combine those constraints when searching for solutions. The selection probability of one ant for one particular path in one generation is defined as [15,16]:

$$\text{prob}=\left[\frac{\tau_{best}^{p}-\tau_{i}^{p}}{\tau_{best}^{p}}\right] \quad (9)$$

where

τ_{best}^{p} : largest value of pheromone accumulation among all of the ants.

τ_{p}^{i} : value of pheromone accumulation for the ith ants.

Updating Pheromone

After all ants have completed their journey through the problem space and produced their solutions, their worthiness is recognized. Then, the pheromone update for all nodes is done according to Equation 10, then the amount of pheromone changes associated with each node is updated according to Equation 11. In this equation, the left side demonstrates the pheromone evaporation for all nodes, and the right side represents pheromone release which is only considered for the best ants [17].

$$\tau_{ij}(t+1)=\tau_{ij}(t)+\Delta\tau_{ij} \quad (10)$$

$$\Delta\tau_{ij}=-\lambda\tau_{ij}(t)+\begin{cases}\dfrac{Q}{num} & \text{for node of the best ant}\\ 0 & \text{otherwise}\end{cases} \quad (11)$$

where

$\tau_{ij}^{(t)}$: pheromone between nodes i, j (previous iteration).

$\tau_{ij}^{(t+1)}$: pheromone between nodes i, j (current iteration).

Q: The amount of pheromone release on nodes relating to the way of best ants.

Num: Priority number of ants between release pheromone candidates

λ: evaporation coefficient of pheromone.

Analysis of the Advantages/ Disadvantages of the Ant Colony Algorithm

Advantages of the Ant Colony [14]:

1. It can be used for Traveling Salesman Problem and similar problems
2. Intrinsic parallelism
3. Efficient for use in dynamic applications.
4. Positive feedback accounts for the rapid discovery of good solutions

Disadvantages of the Ant Colony [18]:

1. It's difficult to the theoretical analysis
2. Time of convergence is uncertain
3. Probability distribution changes by iteration

The optimization process is shown in Figures 7 to 9. First, the ant positions are assigned randomly in the optimization domain, as shown in Figure 6. After 150 iterations, the ant moves toward the positions where the objective functions are lower, as shown in Figure 7. After 700 iterations, all of the ant converge at one position, as shown in Figure 8.

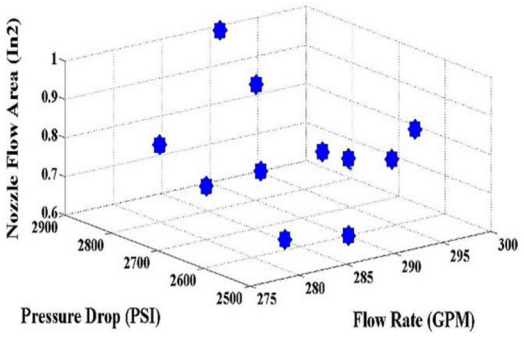

Fig. 7 Initial distribution of ant (Input data).

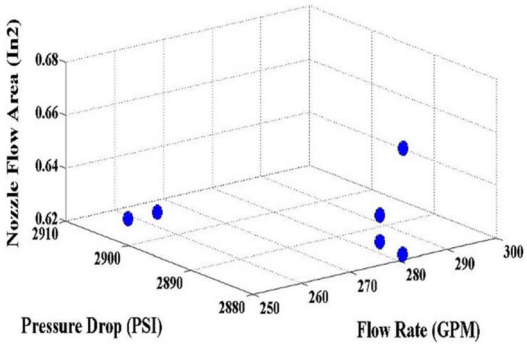

Fig. 8 Ant distribution at 150 iterations.

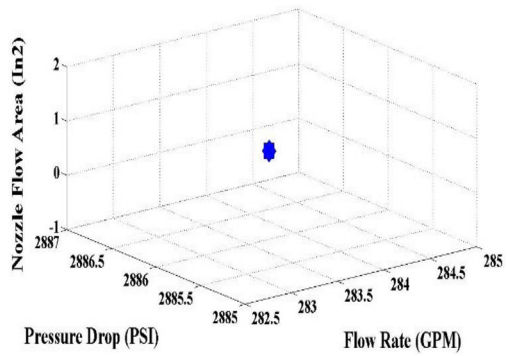

Fig. 9 Final distribution.

Basis of Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is an algorithm modeled on swarm intelligence, which finds a solution to an optimization problem in a search space. The PSO is a stochastic, population-based computer algorithm modeled on swarm intelligence. Swarm intelligence is based on social-psychological principles, providing insights into social behavior and contributing to engineering applications. The particle swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart. The algorithm simulates the social behavior of fish and birds. In the PSO algorithm, each solution is called a particle. The particle swarm simulates this kind of social optimization. First, a problem is given, and some way to evaluate a proposed solution exists in the form of a fitness function. A communication structure or social network is also defined, assigning neighbors with each individual to interact. Afterward, a population of individuals defined as random guesses at the problem solutions is initialized. These individuals are candidate solutions. They are also known as the particles, hence they can be named particle swarm. An iterative process to improve these candidate solutions is set in motion. The particles iteratively evaluate the fitness of the candidate solutions and remember the location where they had their best success. The individual's best solution is called the particle best or the local best. Each particle makes this information available to its neighbors [18]. Figure 10 depicts ACO flowchart proposed by the authors.

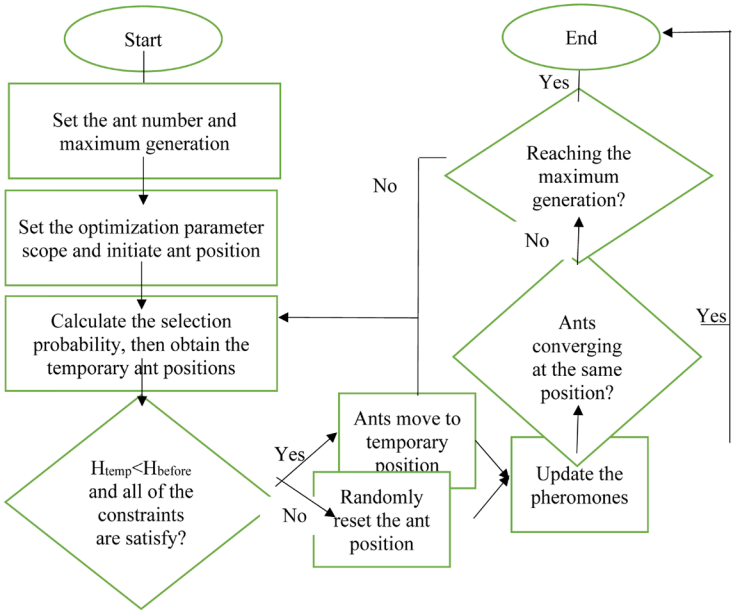

Fig. 10 ACO flowchart (Author).

The Proposed PSO Algorithm

An objective function is defined for optimization problems that are minimized or maximized. In this study, cutting bed height is set as the objective function, and it is minimized concerning three constraints.

At each iteration, each particle in the swarm moves to a new position in the search space through simple equations. The movement of each particle is affected by its personal best, moving to the global best position as well. This trend moves the swarm to the global best positions [19,20].

PSO Algorithm Steps:

1. The initial population is created and evaluated.
2. Personal best and global best positions are determined
3. Particles velocity and their positions are updated by Equations 12 and 13.

$$\overrightarrow{V_{ij}(t+1)} = W\overrightarrow{V_{ij}(t)} + r_1 c_1(P_{ij}(t) - \overrightarrow{x_{ij}(t)}) + r_2 c_2(g_j(t) - \overrightarrow{x_{ij}(t)}) \quad (12)$$

$$\overrightarrow{X_{ij}(t+1)} = \overrightarrow{X_{ij}(t)} + \overrightarrow{V_{ij}(t+1)} \quad (13)$$

4. This procedure is continued until the stop criterion is satisfied.
5. End

Furthermore, Figure 11 shows PSO velocity component and update position for a particle

In Equation 12, ω, c_1 and c_2 are weights, r_1 and r_2 are uniformly distributed random vectors in the range of [0 1], and ij refers to the ith particle in the jth iteration. In this study, we set ω=0.721 and c_1=c_2=1.194. These values were determined from numerical experiments by [21]. Figure 12 shows PSO flowchart proposed by the authors. With this regard, a pseudo coding of the PSO Algorithm is also presented in Table 2.

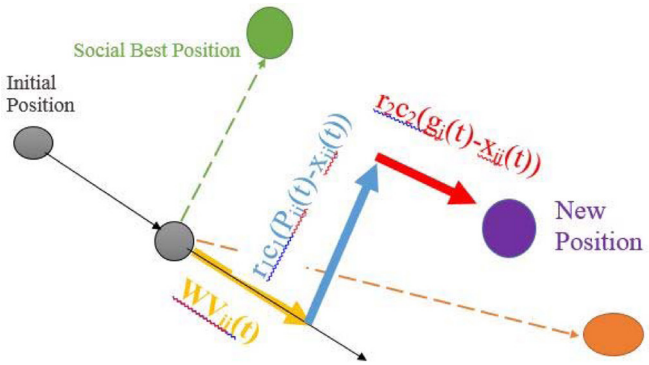

Fig. 11 PSO Velocity component and update position for a particle (Author).

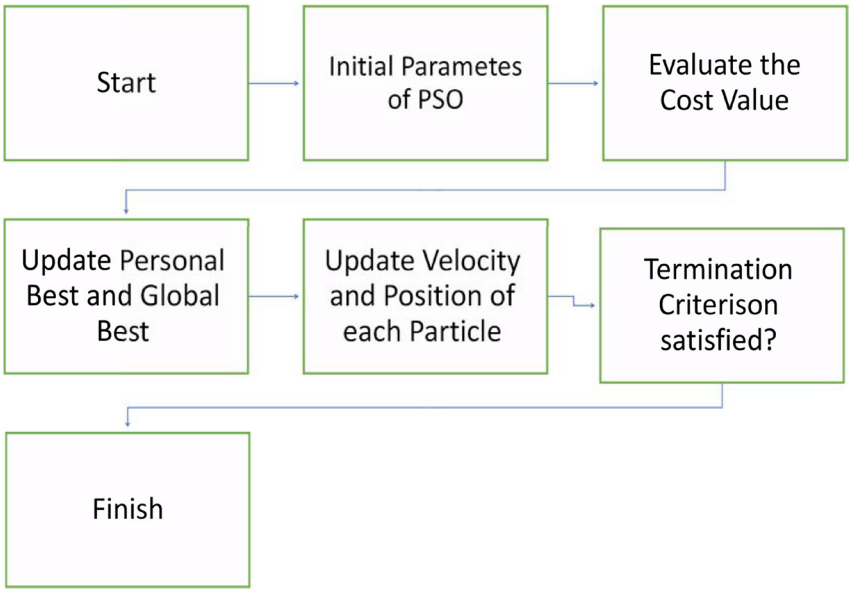

Fig. 12 Flowchart of PSO (Author).

Table 2 A pseudo coding of the PSO Algorithm.

1:	procedure PSO
2:	for particle i $\in\{1,2,\ldots,N\}$ do
3:	for dimension j $\in\{1,2,\ldots,n\}$ do
4:	set $x_{ij} \sim U(\text{lowerBoundary}_j, \text{upperBoundary}_j)$
5:	set d_j $\|\text{upper Boundary}_j - \text{lower Boundary}_j\|$
6:	set $v_{ij} \sim U(-d_j, d_j)$
7:	set p_{ij} x_{ij}
8:	if $f(p_{ij}) < f(p_{gj})$ then
9:	set p_{gj} p_{ij}
10:	for timestep t $\in\{1,2,\ldots,I_{max}\}$ do
11:	for particle i $\in\{1,2,\ldots,N\}$ do
12:	set $r_p \sim U(0, 1)$
13:	set $r_g \sim U(0, 1)$
14:	for dimension j $\in\{1,2,\ldots,n\}$ do
15:	Update v_{ij},x_{ij} from Eq (12), (13)
16:	if $f(x_{ij}) < f(p_{ij})$ then
17:	set p_{ij} x_{ij}
18:	if $f(p_{ij}) < f(p_{gj})$ then
19:	set p_{gj} p_{ij}
20:	Print best solution p_{gj}

Results and Discussion

The optimization process is shown in Figures 13 and 14. It should be stated that the input data are similar to ones shown in Figure 7. First, the particle positions are assigned randomly in the optimization domain. After 50 iterations, the particle moves toward the positions where the objective functions are lower, as shown in Figure 12. Finally, after 300 iterations, all particle converge at the one position, as shown in Figure 13.

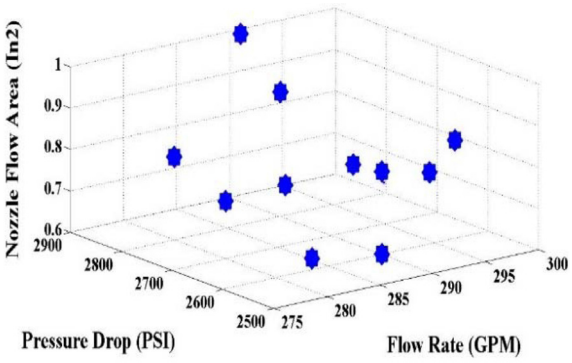

Fig. 13 Particle distribution at 50 iteration.

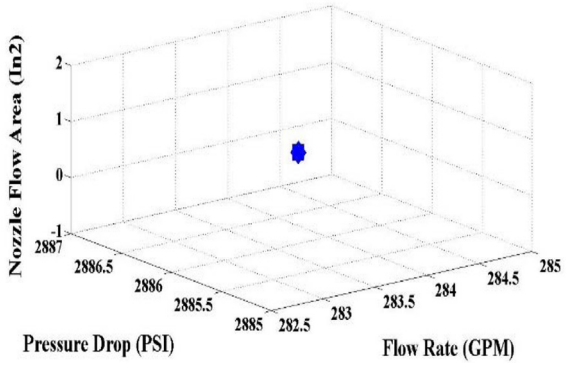

Fig. 14 Final distribution.

Analysis of the Advantages/ Disadvantages of the PSO algorithm

Advantages of the PSO Algorithm

1. PSO is based on intelligence. Therefore, it can be applied to both.
2. PSO has no overlapping and mutation calculation. The search can be carried out by the speed of the particle. During the development of several generations, only the most optimist particle can transmit information onto the other particles, and the speed of the researching is very fast.
3. The calculation in PSO is very simple. Compared with the other developing calculations, it occupies the bigger optimization ability, and it can be completed easily.
4. PSO adopts the real number code, and it is decided directly by the solution. The number of the dimension is equal to the constant of the solution [22].

Disadvantages of the PSO Algorithm

1. The method easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and direction.
2. The method cannot work out the problems of scattering and optimization.
3. The method cannot work out the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field [23].

Comparison of particle swarm optimization with ant colony in hole cleaning problem [9] addressed hole cleaning optimization using ant colony algorithm. In their study, flow rate and rate and bit nozzle size were the optimization parameters for hole cleaning optimization. The optimization domains for the optimization variable shows in Table 3, and the ACO parameters in the algorithm are presented in Table 4. The results of their study are shown in Table 5: Furthermore, the ant colony trend is shown in Figure 15.

Table 3 The optimization domains for ACO optimization variable [9].

Variable	Domain
Q	0.005-0.025 m^3/s
A_n	0.0001-0.0004 m^2
K	0.60-0.95 $Pa.S^n$

Table 4 ACO Parameters Adopted [9].

Parameters	Value
Population Size	10
Max generation	1000
Pheromone evaporation coefficient	0.8
Number of Iteration	840

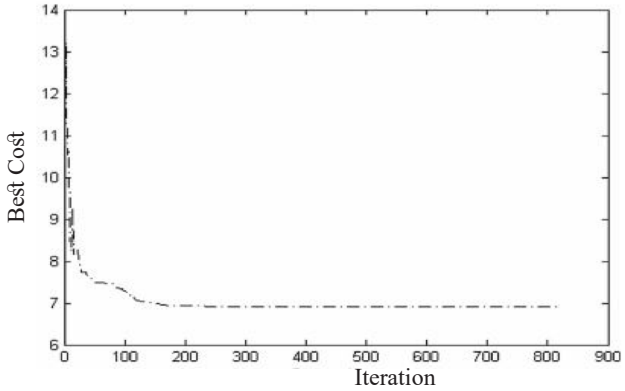

Fig. 15 Ant Colony trend [9].

Table 5 The results from the Ant colony Algorithm [9].

	Nozzle Flow Rate (m^2)	Flow Rate (m^3/s)	Total Pressure Loss (Pa)
Before Optimization	3.79×10^{-4}	0.0130	1.65×10^{7}
After Optimization	4×10^{-4}	0.0179	1.99×10^{7}

The evolution trend in their study is shown in Figure 14. There was 840 iterations at the end of optimization.

To find out which of the PSO algorithm and Ant Colony are efficient for hole cleaning optimization; the Zhi-Chuan problem was solved using PSO and ACO algorithms. The PSO parameters in the algorithm are presented in Table 6, and the optimization domains for PSO optimization are listed in Table 7, and the results are shown in Table 8. Moreover, we developed Ant colony algorithm which is better than Zhi-chuan Ant colony has been developed, but PSO algorithm is better than two Ant Colony algorithms. Similarly, Tables 9, 10 and 11 list the ACO parameters, the optimization domains and the results, respectively. Moreover, PSO trend for this study is depicted in Figure 16.

Table 6 PSO Parameters Adopted.

Parameters	Value
Number of Particles	700
Number of Iteration	300
Local Weights(C_1)	1.496
Social Weights(C_2)	1.496
Inertia weight	0.7298

Table 7 The domain of variables for the PSO optimization.

Variable	Domain
Q	0.01-0.04 m^3/s
A_n	0.0000064-0.0005 m^2
K	0.50-0.99 $Pa.S^n$

Table 8 The results from the PSO Algorithm.

	Nozzle Flow Rate	Flow Rate	Total pressure Loss
After Optimization (SI)	346×10^{-6}	0.0170	1.7960×10^{7}
After Optimization (Field)	270.9672	0.5371	2605

Table 9 The domain of variables for ACO optimization (This Study).

Variable	Domain
Q	0.007-0.021 m^3/s
A_n	0.0002-0.0004 m^2
K	0.71-0.83 $Pa.S^n$

Table 10 ACO Parameters Adopted (This Study).

Parameters	Value
Population Size	500
Max generation	2000
Pheromone evaporation coefficient	0.9
Number of Iteration	700

Table 11 The results from the Ant colony Algorithm (This Study).

	Nozzle Flow Area (In2)	Flow Rate (G P M)	Total pressure Loss (PSI)
After Optimization	0.60	279.24	2802

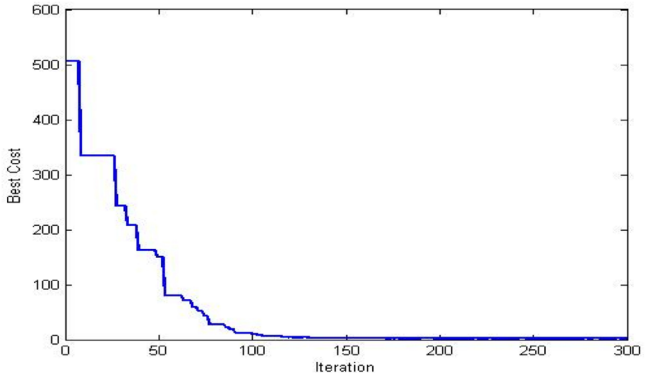

Fig. 16 PSO trend (This Study).

In this study, the evolution trend is shown in Figure 17. There were 700 iterations at the end of optimization.

Sensitivity Analysis for PSO Algorithm

In this section, the impact of efficient parameters on quick and accurate algorithm convergence has been checked out. To better determine the algorithm result, the chart shows a Cartesian and logarithmic scale [24,25]. Finally, in Table 12, the result analyses are shown numerically.

Number Population

The effects of number population on the objective function are illustrated in Figures 18 and 19. This figure represents that a population of less than 700 is not suitable for hole cleaning, and there is no good result. As a result, a 700-number population has been selected for this optimization, and the cutting bed height has been minimized (i.e., Objective Function).

Iteration

Figures 20 and 21 illustrate the effect of iteration on the objective function. This figure represents that an iteration of less than 300 is unsuitable for hole cleaning and is not a good result. For example, 50 or 100 iterations are not enough for good optimization, and by increasing the iteration, the more optimal results are achieved. As a result, 300 iterations for this optimization have been selected, and the cutting bed height (i.e., Objective Function) has been minimized.

Table 12 Numerical results of sensitivity analysis

	Q	An	Dp Loss	H
Run 1 (Iteration 50)	97.3605	0.5452	1.0328e+03	123.790
Run 2 (Iteration100)	463.3276	0.6154	2.6061e+03	7.516
Run 3 (Npop100)	287.2360	0.5929	2.7417e+03	3.704
Run 4 (Npop50)	358.1696	0.5248	3.6626e+03	5.260
Run 5 (Npop250)	89.8055	0.7094	2.7038e+03	133.25
Run 6 (W=1)	110.3598	0.5770	1.1148e+03	100.02
Run 7 (W=2)	286.8217	0.7224	2.6666e+03	5.02
Run 8 (W=0)	337.2268	0.6042	3.3053e+03	5.342
Run 9 (C1=0)	371.3067	0.4235	4.0791e+03	6.334
Run 10 (C2=2)	157.0139	0.6183	1.4538e+03	83.225
Run 11 (C1=2)	178.1708	0.3474	1.7800e+03	52.358
Run 12 (C2=0)	295.7996	0.5312	2.8935e+03	781.6404

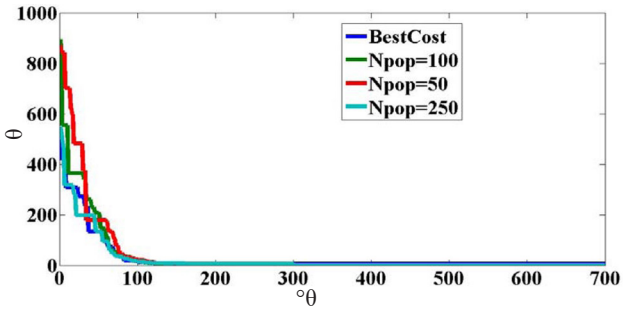

Fig. 18 Effect of number population on objective function (Cartesian).

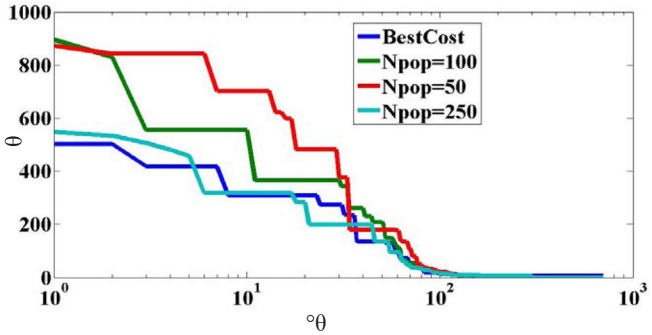

Fig. 19 Effect of number population on the objective function(x-log).

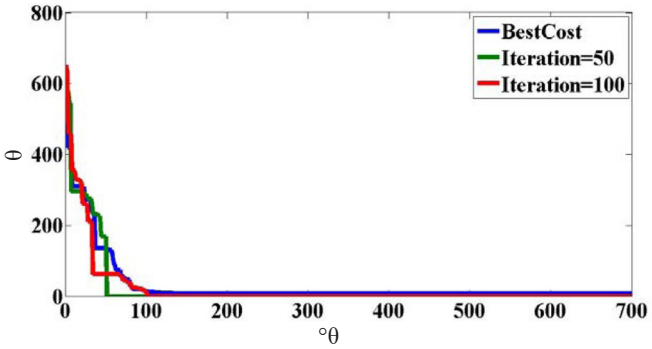

Fig. 20 Effect of iteration on the objective function (Cartesian).

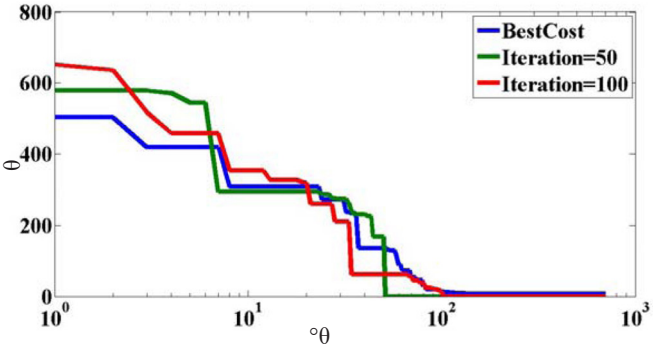

Fig. 21 Effect of Iteration on the objective function(x-log).

Inertia Weight

The W parameter in particle swarm optimization is inertia weight. This parameter is efficient on the quick and accurate algorithm convergence that its domain change between 0-2. By comparing different values for W, it is found out that the best optimization is when w=0.7298 (Figure 22 and 23).

Local and Social Weights

The C1 and C2 parameters in particle swarm optimization are local and social weights. These are other efficient parameters on the quick and accurate algorithm convergence that its domain change between 0-2.

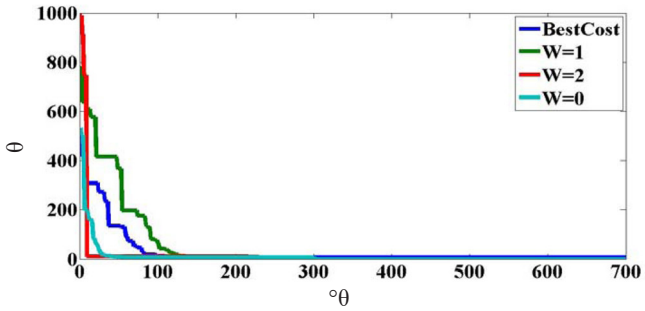

Fig. 22 Effect of Inertia weight on the objective function (Cartesian).

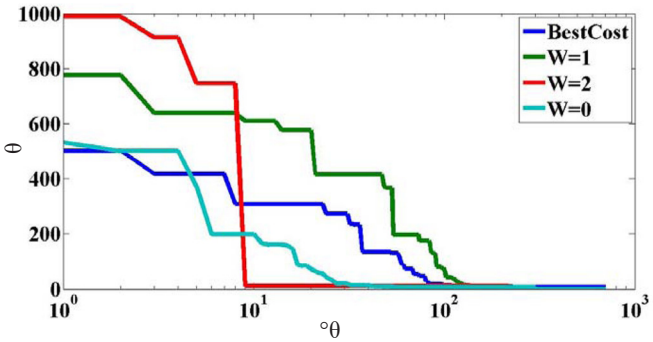

Fig. 23 Effect of Inertia weight on the objective function (x-log).

By comparing different values for local and social weights, it is found out that the best optimization is when C1=C2=1.4962 (Figures 24 and 25).

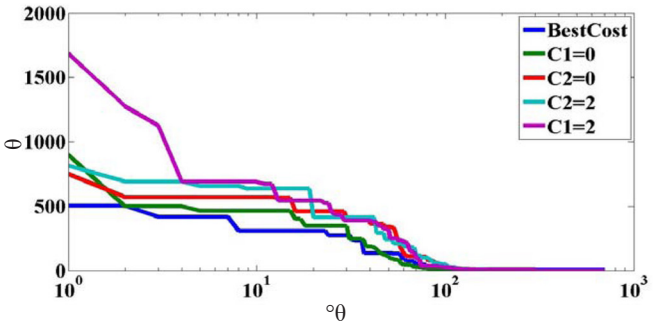

Fig. 24 Effect of Local and social Weights on the objective function (Cartesian).

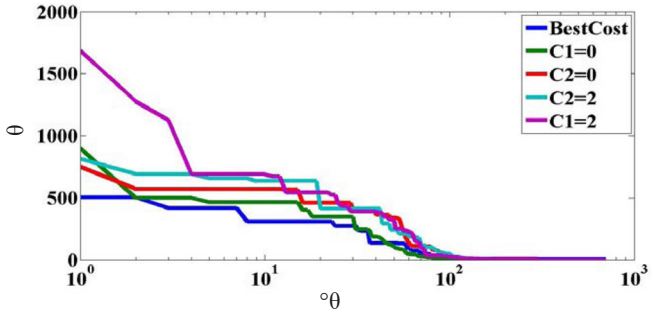

Fig. 25 Effect of local and social weights on the objective function (x-log).

As shown in Figures 17-25, each run was associated with a lack of adequate improvement and parameters not optimized enough. Therefore, the data in Table 12 offered the optimized condition.

Evaluation Process

In this section, we checked out the selection process of optimum parameters (i.e. number of population, iteration, inertia weight, etc.) [26,27].

As can be seen, the optimal parameters are shown in green, and the not optimal parameters are shown in red. The intention of optimization is to minimize the objective function (Cutting bed height).
Table 13 represents that iteration of less than 300 is not suitable for hole cleaning, and a good result is not obtained. For example, 50 or 100 iterations are not good enough for optimization, and by increasing the iteration, the more optimal result is achieved. As a result, 300 iterations are selected for this optimization, and the cutting bed height is minimized (i.e. Objective Function).

Table 13 Process selection of iteration.

	Q	An	Dp Loss	H
Run 1(Iteration 50)	97.3605	0.5452	1.0328e+03	123.790
Run 2 (Iteration100)	463.3276	0.6154	2.6061e+03	7.516
Best Run (Iteration300)	270.9672	0.5371	2605	3

In Table 14, it is represented that number population of less than 700 is not suitable for hole cleaning and a good result is not obtained. As a result, a 700-number population has been selected for this optimization, and the cutting bed height (i.e., Objective Function) has been minimized. Furthermore, process selection of intertia weight is represented in Table 15.

Table 14 Process selection of Number Population.

	Q	An	Dp Loss	H
Run 3 (Npop100)	287.2360	0.5929	2.7417e+03	3.704
Run 4 (Npop50)	358.1696	0.5248	3.6626e+03	5.260
Best Run (Npop 700)	270.9672	0.5371	2605	3

Table 15 Process selection of Inertia weight.

	Q	An	Dp Loss	H
Run 6 (W=1)	110.3598	0.5770	1.1148e+03	100.02
Run 7 (W=2)	286.8217	0.7224	2.6666e+03	5.02
Run 8 (W=0)	337.2268	0.6042	3.3053e+03	5.342
Best Run (W=0.7298)	270.9672	0.5371	2605	3

The inertia weight parameter is an efficient parameter on the quick and accurate algorithm convergence that its domain changes between 0-2. By comparing different values for W, it is found that the best optimization is when w=0.7298.
The C1 and C2 parameters in particle swarm optimization are local and social weights. These are other efficient parameters on the quick and accurate algorithm convergence that their domain change between 0-2. By comparing different values for local and social weights, It is found out that the best optimization is when C1=C2=1.4962. In addition, process selection of social and local weight is presented in Table 16.

Assessment

Finally, the function of algorithms must be evaluated. Table 17 and Figure 26 or Figure 27 show the comparison of ant colony algorithm and particle swarm optimization algorithm.

Table 16 Process selection of social and local weight.

	Q	An	Dp Loss	H
Run 9 (C1=0)	371.3067	0.4235	4.0791e+03	6.334
Run 10 (C2=2)	157.0139	0.6183	1.4538e+03	83.225
Run 11 (C1=2)	178.1708	0.3474	1.7800e+03	52.358
Run 12 (C2=0)	295.7996	0.5312	2.8935e+03	781.6404
Best Run (C1=C2=1.4962)	270.9672	0.5371	2605	3

Table 17 Comparison of algorithms.

	Ant Colony Algorithm (ZhichuanGuan, 2016)	Ant Colony Algorithm (This study)	PSO Algorithm (This study)
Iteration	840	700	300
Objective Function	6	5.8	3
		4% improvement	50% improvement
Flow Rate	283.72	279.24	270.96
Total Pressure Loss	2886	2802	2605
Nozzle Flow Area	0.62	0.60	0.53
Jet velocity	44.75	43.95	42.26

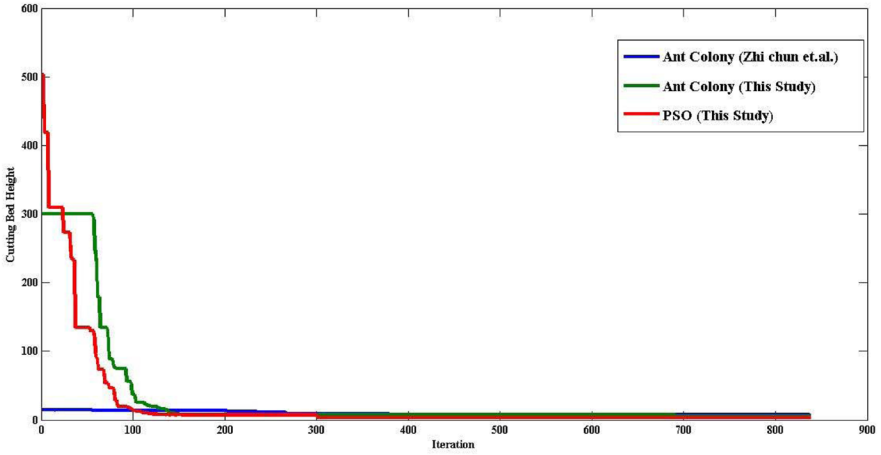

Fig. 26 Compression between Ant Colony and Particle Swarm Optimization (Cartesian).

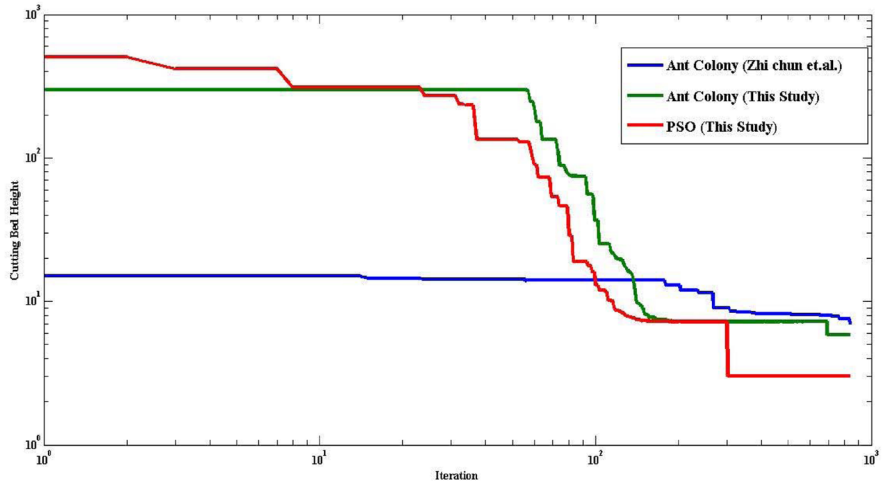

Fig. 27 Compression between Ant Colony and Particle swarm optimization (Log-Log).

As expressed in Table 12 and Figures 26 and 27, the results of the three algorithms are close. Still, the Particle Swarm Optimization algorithm showed faster convergence, better solutions, and improved objective function than the ant colony algorithm, because the PSO algorithm is better than the ant colony algorithm for continuous search spaces. Also, the PSO algorithm has suggested a lower flow rate and pressure drop, which is valuable and cheaper, and lower capacity pumps can be used. In addition, the objective function has also improved and showed that the PSO algorithm is more efficient than Ant Colony algorithm.

Improvement in This Study

By analysing the resulting parameters, it is shown that PSO algorithm is more quick and accurate than Ant colony algorithm, so that PSO, 50% and ACO, 4% improved ACO [9]. As a result, it is concluded that the PSO algorithm showed faster convergence, better solutions, and more improvement in the objective function, as shown in Figure 28.

Conclusions

Many investments in oil and gas development projects are there in the drilling section; therefore, optimum conditions and situations in drilling operations are necessary.

1. The common existing methods in the literature for hole cleaning optimization are not appropriate for deep and ERD wells because they are not practical in an oil field and are also time-consuming.
2. The flow rate is the dominant parameter in the growth of the cutting layer. Therefore, the cutting bed height would be lessened by increasing the flow rate.
3. As shown in the figures, the effect of rheological properties variation in different hole angles is different. For example, in low deviation degrees, the amount of PV and YP has an inverse effect on the increment of minimum velocity required for cutting transport.
4. As mentioned before, the flow rate is of great importance in the case of optimizing the cutting bed height, so to set and choose the optimum flow rate, the constraints like pump horsepower should be considered.
5. The resulting parameters show that the PSO algorithm is much more accurate than the ACO algorithm, and they improved objective function by 50% and 4%, respectively.
6. The PSO algorithm can be used in hole cleaning optimization to optimize the cutting bed height in deviated wells, so it could reduce the costs and problems such as cutting accumulation.

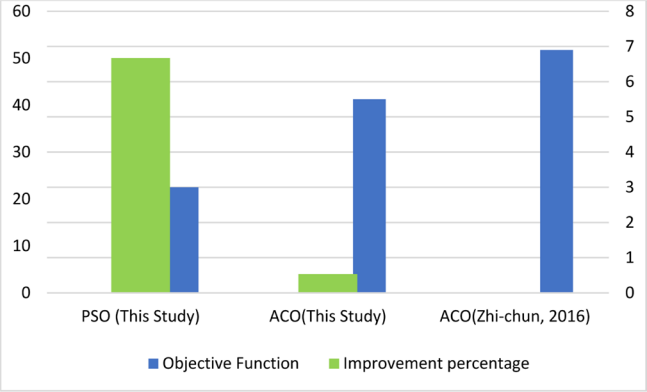

Fig 28 Comparison between ACOs and PSO.

1. Algorithm code

```
clc;
clear;
close all;


%% Problem Definition
global NFE;
NFE=0;
model=Creatmodel();

%    CostFunction=@(xhat)  Mycost(xhat,model);  %  Cost
Function


nVar=model.N;     % Number Of Decision Variables

Varsize=[1 nVar]; % Size Of Decision variable matrix

VarMin=[0 0 0];          % Lower Bound Of Vaiables
VarMax=[1 1 1];           % Upper Bound Of Vaiables
%% PSO Parameters
MaxIt=600;          %Max Of Iterations


nPop=400;           % Population Size (Swarm Size)

w=.729;             % Inertia Weight
wdamp=0.99;         % Inertial Weight Damping Ratio

c1=1.494;           % Personal learning coeffcient
c2=1.494;           % Global learning coeffcient

% Velocity Limits
Velmax=1*(VarMax-VarMin);
Velmin=-Velmax;



%% Initilization

empty_particle.Position=[];
empty_particle.Cost=[];
empty_particle.Velocity=[];
empt_particle.Sol=[];
empty_particle.Best.Position=[];
empty_particle.Best.Cost=[];
empty_particle.Best.Sol=[];
particle=repmat(empty_particle,nPop,1);

GlobalBest.Cost=inf;


for i=1:nPop
   % Initialize Position
   particle(i).Position=CreatRandomSolution(model);
   % Initialize Velocity
   particle(i).Velocity=zeros(Varsize);
   % Evaluation
    [particle(i).Cost, particle(i).Sol]=Mycost(particle(i).Posi-
tion,model);
   %Update Personal Best
   particle(i).Best.Position=particle(i).Position;

   particle(i).Best.Cost=particle(i).Cost;
    particle(i).Best.Sol=particle(i).Sol;
  % Update Global Best
   if particle(i).Best.Cost< GlobalBest.Cost

      GlobalBest=particle(i).Best;

   end

end

BestCost=zeros(MaxIt,1);
nfe=zeros(MaxIt,1);
%%   PSO Main Loop

for it=1:MaxIt
   for i=1:nPop
      % Update Velocity
      particle(i).Velocity=w*particle(i).Velocity...
          +c1*rand(Varsize).*(particle(i).Best.Position-parti-
cle(i).Position)...
         +c2*rand(Varsize).*(GlobalBest.Position-particle(i).
Position);
      % Apply Velocity Limits
```

```
        particle(i).Velocity=max( particle(i).Velocity,Velmin);
        particle(i).Velocity=min( particle(i).Velocity,Velmax);

        % Update Position
          particle(i).Position=particle(i).Position+particle(i).Ve-
locity;

        % Velocity Mirror Effect
         IsOutside=(particle(i).Position<VarMin | particle(i).Po-
sition>VarMax );
          particle(i).Velocity(IsOutside)=- particle(i).Velocity(I-
sOutside);
        % Apply Position  Limits
         particle(i).Position=max( particle(i).Position,VarMin);
         particle(i).Position=min( particle(i).Position,VarMax);
        % Evaluation
      [particle(i).Cost, particle(i).Sol]=Mycost(particle(i).Posi-
tion,model);
       %Update Personal Best
      if   particle(i).Cost<particle(i).Best.Cost

          particle(i).Best.Position=particle(i).Position;
          particle(i).Best.Cost=particle(i).Cost;
          particle(i).Best.Sol=particle(i).Sol;
       % Update Global Best
       if particle(i).Best.Cost<GlobalBest.Cost

          GlobalBest=particle(i).Best;

       end

      end

    end


    BestCost(it)=GlobalBest.Cost;
    nfe(it)=NFE;
    disp(['Iteration' num2str(it) 'NFE=' num2str(nfe(it)) ': Best
Cost=' num2str(BestCost(it))]);
   w=w*wdamp;
end



%% Results


figure;
plot(BestCost,'Linewidth',2);
%semilogy(nfe,BestCost,'Linewidth',2);
xlabel('Iteration');
ylabel('Best Cost');
```

2. My Cost Function Code

```
function [z ,sol] = Mycost(xhat,model)
% N = model.n;


Dh = model.Dh ;
Pmax = model.Pmax;
Qr = model.Qr ;
Vjmin= model.Vjmin;
EQDEpore = model.EQDEpore;
EQDEcave = model.EQDEcave ;
EQDEFrac = model.EQDEFrac ;
eps = model.eps;
Va = model.Va;
Nrotry = model.Nrotry;
nFLBH = model.nFLBH;
ROs= model.ROs;
ROf = model.ROf;
ds= model.ds;
Inc = model.Inc;
Hv = model.Hv;
Lp = model.Lp;
g= model.g;
Di=model.Di;
PV= model.PV;
YP= model.YP;
Dic= model.Dic;
Lc= model.Lc;
ODp= model.ODp;
ODc= model.ODc;



Qmax=800;
Qmin=100;
Kmax=500;
Kmin=0;
Anmax=0.9;
Anmin=0.1;



Q = (Qmax-Qmin)* xhat(1)+Qmin;
K = (Kmax - Kmin)*xhat(2)+Kmin;
An = (Anmax - Anmin)*xhat(3)+Anmin;


Miooe=          ((((Dh-ODp)^(1-nFLBH))*(((2*nFLBH+1)/
(3*nFLBH))^nFLBH)*K)*(20*Va)^(nFLBH-1))/1000^n-
FLBH;

Vcrit= 20.09*abs((((ROs-ROf)*ds/ROf))^0.667)...
            *((1+0.71*Inc+0.55*sind(2*Inc))/abs((ROf*Mio-
oe)^0.333));

Tcb=              (0.015*Dh)*(1000*Miooe+194.48*(Mio-
oe^0.5))*(1+0.587*eps)*(Vcrit-Va)+Dh*(0.0001*(Nro-
try^2)…
   -(0.35468*Nrotry)+((0.16236*Nrotry)*Va)...
   -(0.09465*Nrotry*eps)+(0.00034*Nrotry*Va*eps))/100;

H = 100*Tcb/Dh;

%%%%%%%'Pressure Drop at Drillpipe '%%%%%%

V1=(24.5*Q)/Di^2;
```

```
Vc1=(97*PV+97*sqrt(PV^2+8.2*ROf*Di^2*YP))/(ROf*-
Di);
 if V1<Vc1
deltapi1=(Lp*(YP+(PV*V1/(300*Di))))/(300*Di);
 else
    deltapi1=((8.91*10^-5)*(ROf^(0.8))*(Q^(1.8))*(PV^(0.2
))*Lp)/Di^4.8;
 end
%disp('Pressure Drop at Drillpipe= ');
%disp (deltapi1);



%%%%%%%Pressure Drop at DrillCollar%%%%
V2=(24.5*Q)/Dic^2;
Vc2=(97*PV+97*sqrt(PV^2+8.2*ROf*Dic^2*YP))/
(ROf*Dic);
 if V2<Vc2
deltapi2=(Lc*(YP+(PV*V2/(300*Dic))))/(300*Dic);
 else
    deltapi2=((8.91*10^-5)*(ROf^(0.8))*(Q^(1.8))*(PV^(0.2
))*Lc)/Dic^4.8;
 end
%disp('Pressure Drop at DrillCollar= ');
%disp (deltapi2);



%%%%%%%%Pressure Drop at Annulus (PIPE)%%%%%
V3=(24.5*Q)/(Dh^2-ODp^2);
Vc3=(97*PV+97*sqrt(PV^2+6.2*ROf*(Dh-ODp)^2*YP))/
(ROf*(Dh-ODp));
if V3<Vc3
deltapi3=(Lp*PV*V3)/(60000*(Dh-ODp)^2)+(Lp*YP)/
(200*(Dh-ODp));
else
   deltapi3=(8.91*10^-5*ROf^(0/8)*Q^(1/8)*PV^(0/2)*Lp)/
((Dh-ODp)^(3)*(Dh+ODp)^1/8);
end
%disp('Pressure Drop at Annulus (PIPE)= ');
%disp (deltapi3);



%%%%%%%%Pressure Drop at Annulus (Collar)%%%%%%
V4=(24.5*Q)/(Dh^2-ODc^2);
Vc4=(97*PV+97*sqrt(PV^2+6.2*ROf*(Dh-ODc)^2*YP))/
(ROf*(Dh-ODc));
if V4<Vc4
deltapi4=(Lc*PV*V4)/(60000*(Dh-ODc)^2)+(Lc*YP)/
(200*(Dh-ODc));
else
   deltapi4=(8.91*10^-5*ROf^(0/8)*Q^(1/8)*PV^(0/2)*Lc)/
((Dh-ODc)^(3)*(Dh+ODc)^1/8);
end
%disp('Pressure Drop at Annulus (Collar)= ');
%disp (deltapi4);

%%%%%%%%Pressure Drop at Bit%%%%%%
Dpbit= (9.21*10^-5*ROf*Q^2)/(An^2);
%disp('Pressure Drop at Bit= ');
%disp (Dpbit);

%%%%%%%Pressure Drop Surface%%%%%%
Dpsur=(5.3*10^-5)*(ROf^0.8)*(Q^1.8)*(PV^0.2);
%disp (Dpsur)

%%%%%%%%Total Pressure Drop%%%%%%
DpLoss=deltapi1+deltapi2+deltapi3+deltapi4+Dpbit+Dp-
sur;


ECD= ROf+(188/(g*Hv));


beta = 1;
Violation_pmax = max(1- DpLoss/Pmax ,0);
Violation_Qr = max(1- Q/Qr ,0);
Violation_Vjmin = min((Q/An)/Vjmin-1,0) ;
Violation_ECD1    =    min(ECD/(max(EQDEpore,EQDE-
cave))-1,0)   ;
Violation_ECD2 = max(1- ECD/EQDEFrac,0);

z = H+beta*(Violation_pmax+Violation_Qr+abs(Violation_
Vjmin)...
   +abs(Violation_ECD1)+Violation_ECD2);

sol.Q = Q ;
sol.K = K;
sol.An = An ;
sol.DpLoss = DpLoss;
sol.Vj = Q/An;
sol.H = H ;
end
```

3. CreatRandomSolution Code

```
function xhat = CreatRandomSolution(model)
N = model.N;
xhat = unifrnd(0,1,1,N);
end
```

4. Model Function Code

```
function model= Creatmodel()
model.N = 3; % Number of variables
model.Dh = 6;
model.Pmax = 4089;
model.Qr = 300;
model.Vjmin = 131.23;
model.EQDEpore = 69.48 ;
model.EQDEcave = 63.24;
model.EQDEFrac = 142.77 ;
model.Inc = 65; % Degree of hole Inclination
model.eps = 0.001;
model.Va = 2.2;
model.Nrotry = 50;
model.nFLBH = 0.5442;
model.ODp=4;
model.ODc=5.5;
model.ROs = 2.6;
model.ROf = 10.01;
model.ds = .01;
```

```
model.Hv = 10137;
model.Lp = 2549;
model.Dic=2.25;
model.Lc=354;
model.PV=47;
model.YP=32;
model.Di=3;
model.g =32.17;
end
```

References

1. Ogunrinde J O, Dosunmu A (2012) Hydraulic optimization for efficient hole cleaning in deviated and horizontal wells, paper SPE presented at the 2012 SPE Nigerian Annual Intenational, 6-8.
2. Lim K M, Chukwu G A (1996) Bit Hydraulics analysis for efficient hole cleaning, Paper presented at the SPE Western Regional Meeting, Anchorage, Alaska.
3. Bernt S A (2010) Modern well design, CRC Press; 2nd edition.
4. Mohammadsalehi M, Malekzadeh N (2012) Optimization of hole cleaning and cutting removal in vertical, deviated and horizontal wells, Paper Presented at the International Petroleum Technology Conference held in Bangkok, Thailand, 7–9.
5. Cayeux E, Leulsegend A, Kluge R, Haga J (2016) Use of a transient cutting transport model in the planning, monitoring and post analysis of complex drilling operation in the North Sea, In IADC/SPE Drilling Conference and Exhibition. OnePetro.
6. Heitmann N, Molero R, Molero R , Graterol W , Ouali Y, Chamat Burgos E , Cisneros A (2014) Novel integrated hole-cleaning concept reduces well construction by 3 rig days, Paper presented at the IADC/SPE Asia Pacific Drilling Technology Conference, Bangkok, Thailand.
7. Zhang F, Miska S, Yu M, Ozbayoglu E, Takach N, Osgouei R E (2015) Is well clean enough? a fast approach to estimate hole cleaning for directional drilling, Paper presented at the SPE/ICoTA Coiled Tubing and Well Intervention Conference and Exhibition, The Woodlands.
8. Thompson J, Wilkes J, Marland C, Martin A, Bindl B, Brunneder M (2015) Hole-cleaning optimization: a case history from three high-angle wells in Austria, Paper Presented at the SPE Western Regional Meeting, Garden Grove, California, USA.
9. Guan Z C, Liu Y M, Liu Y W, Xu Y Q (2016) Hole cleaning optimization of horizontal wells with the multidimensional ant colony algorithm, Journal of Natural Gas Science and Engineering, 28:347-355.
10. Zhou F, Pu C (1998) Study on cuttings bed height prediction in eccentric annulus of horizontal well, Petrolum Drill Technology.
11. Selvi V, Umarani R (2010) Comparative analysis of ant colony and particle swarm optimization techniques, International Journal of Computer Applications, 5:1-6.
12. Abdulkader M S, Gajpal Y, ElMekkawy Y T (2015) Hybridized ant colony algorithm for the multi Compartment Vehicle Routing Problem, Applied Soft Computing, 37: 196-203.
13. Benhala B, Ahaitouf A, Mechaqrane A(2012) Multi objective optimization of an operational amplifier by the ant colony optimization algorithm, Electrical and Electronic Engineering, 2, 4: 230-235.
14. Jovanovic R, Tuba M, Voß S (2016) An ant colony optimization algorithm for partitioning graphs with supply and demand, Applied Soft Computing, 41: 317-330.
15. Dosumnu P A, Cosmas O, Orun C, Anyanwu C, Ekeinde E (2015) Optimization of hole cleaning using dynamic real-time cuttings monitoring tools, Paper presented at the SPE Nigeria Annual International Conference and Exhibition, Lagos, Nigeria.
16. Purian F Kh, Farokhoi F, Nadooshan R S(2013) Comparing the performance of genetic algorithm and ant colony optimization algorithm for mobile robot path planning in the dynamic environments with different complexities, Journal of Academic and Applied Studies, 3.
17. Puymbroeck LV (2013) Increasing drilling performance using hydro-mechanical hole cleaning devices, Paper Presented at the SPE Unconventional Gas Conference and Exhibition, Muscat, Oman.
18. Kaveh A, Talatahari S (2009) Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, Computer and Sructures, 87, 5-6: 267-283.
19. Mohanty B, Panda S (2013) Hybrid BFOA-PSO algorithm for automatic generation control of linear and nonlinear interconnected power systems, Applied Soft Computing, 13, 12: 4718-4730.
20. Assareh E, Behrang MA, Assari M R, Ghanbarzadeh A (2010) Application of PSO and GA Techniques on demand estimation of oil in Iran, Energy, 35, 12: 5223-5229.
21. Clerc M (2006) Stagnation analysis in particle swarm optimization or what happens when nothing happens, Technical Report CSM-460, Department of Computer Science, University of Essex, Edited by Riccardo Poli.
22. Karimi Nasab M, Modarres M, Seyedhosseini S M (2015) A self-adaptive PSO for joint lot sizing and job shop scheduling with compressible process times, Applied Soft Computing, 27: 137-147.
23. Tsai C Y, Chen C J (2015) A PSO-AB classifier for solving sequence classification problems, Applied Soft Computing, 27: 11-27.
24. Nwagu C, Awobadejo T, Gaskin K (2014) Application of mechanical cleaning device: hole cleaning tubulars, to Improve Hole Cleaning, Paper presented at the SPE Nigeria Annual International Conference and Exhibition, Lagos, Nigeria.
25. Kennedy J, Eberhardt RC (1995) Particle swarm optimization, In Proceedings of the IEEE International Joint Conference on Neural Networks, IEEE, December, 1942 -1947.
26. Bizhani M, Rodriguez-Corredor F E, Kuru E (2015) Hole cleaning performance of water vs. polymer-based fluids under turbulent flow conditions, Paper Presented at the SPE Canada Heavy Oil Technical Conference, Calgary, Alberta, Canada.
27. Lu Y, Zhou K, Li W, Tian W, Wang X (2014) Research for control parameters optimization of 6-DOF flight simulator based on particle swarm optimization title, In the Twenty-fourth International Ocean and Polar Engineering Conference. OnePetro.