Journal of Petroleum Science and Engineering 195 (2020) 107512

Contents lists available at ScienceDirect

Journal of Petroleum Science and Engineering

journal homepage: http://www.elsevier.com/locate/petrol

Rain optimization algorithm (ROA): A new metaheuristic method for drilling optimization solutions

Ali Reza Moazzeni, Ehsan Khamehchi *

Department of Petroleum Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

ARTICLE INFO

Keywords:
Rain optimization algorithm
Drilling rate
Drilling optimization
Machine learning

ABSTRACT

With the development of powerful computers, some new methods for solving challenging problems are introduced. Some of these methods, which are called metaheuristic algorithms, such as bat algorithm or ant colony algorithm, are inspired by nature and simulate a natural phenomenon for finding the best solution. Today these algorithms with a robust approach can be used to find the solution of a complicated problem very fast although they might be trapped in the local minimums. Rain optimization algorithm (ROA) is a new metaheuristic algorithm that is inspired by the raindrops, which move toward minimum points after getting to the earth. This algorithm can find global extremum as well as local extremums if its parameters are correctly tuned. After the implementation of this algorithm, we compare it with some other existing optimization algorithms such as particle swarm optimization algorithm and bat algorithm by solving 26 benchmarks and three benchmarks in various dimensions as well as a drilling optimization problem. Simulations illustrate the performance and computational time in finding the global minimum. Also, ROA can find local minimum simultaneously and it can be confidently used in optimization problems.

1. Introduction

Drilling operation is a response to the oil and gas energy demand in the world. There are many drilling optimization problems in the various sequence of drilling operations such as casing design, drilling rate optimization, drilling cost minimization, bit size selection, bit type selection, directional drilling design, mud weight design, wellbore stability, and so on. Many of these optimization problems are highly nonlinear and have several local minima that make it difficult to solve these problems with traditional methods but metaheuristic optimization algorithms can solve many of these problems.

Metaheuristic optimization algorithms raindrops to solve many complex global problems in other fields (Gandomi et al., 2011). These algorithms try to simulate natural phenomena to find a fast and effective solution for complicated problems by using iterative sequences (Talbi, 2009).

In recent years, the researchers develop several population-based metaheuristic optimization algorithms. Some of most famous of them are as follow: the genetic algorithm which is developed based on Darwin's theory by Golberg (1989), Differential Evolution that was introduced by Storn and Price (1997, 1997), the Particle Swarm Optimization algorithm by Kenndy and Eberhart (1995), Harmony Search (Geem et al., 2001), Bacterial Foraging Optimization Algorithm (Passino, 2002), Estimation of Distribution Algorithms (Larra~naga and Lozano, 2002), Artificial Bee Colony algorithm (Basturk and Karaboga, 2006; Karaboga, 2005; Karaboga and Basturk, 2007, 2008), Firefly Algorithm (Yang, 2009; 2009), League Championship Algorithm (Kashan, 2009), Group Search Optimizer (He et al., 2009), Ant Colony. Optimization (Dorigo and Birattari, 2010), Cuckoo Search algorithm (Gandomi et al., 2011), Krill Herd algorithm (Gandomi and Alavi, 2012), Stochastic Fractal Search (Salimi, 2014), Symbiotic Organisms Search (Cheng and Prayogo, 2014), Optics Inspired Optimization (Husseinzadeh Kashan, 2015) and Sperm Whale algorithm (SWA) (Ebrahimi and Khamehchi, 2016) that are characterized by their names.

In this paper, we introduce a new metaheuristic algorithm, namely rain optimization algorithm (ROA), inspired by the natural behavior of rain droplets and raining phenomena for finding minimum locations in the earth's surface. We will first formulate the rain algorithm based on the natural behavior of the rain droplets. Then we will declare how it works and compares the proposed method with existing algorithms such as the genetic algorithm. In the end, the results of this algorithm will be discussed in detail by solving an optimization drilling problem.

* Corresponding author.
E-mail address: khamehchi@aut.ac.ir (E. Khamehchi).

https://doi.org/10.1016/j.petrol.2020.107512
Received 16 September 2019; Received in revised form 30 April 2020; Accepted 5 June 2020
Available online 1 July 2020
0920-4105/

2. Basic of rain behavior

When it starts raining, droplets of rainfall on the earth's surface. After a while, it can be seen that some of these droplets joint to each other and some more significant droplets forms which can move on the surface under the effect of their weight toward the lower locations of the earth's surface. In their path, some other marvelous happening would occur for these droplets too. Some of the other droplets might move toward the previous droplet and joint to it, or some fraction of each droplet might be evaporated or absorbed by the soil depending on different properties of the soil such as nature of the soil surface, porosity, permeability, wettability etc. Also, some of the soil would be dissolved in the water. In this process, droplets that are dropped on the flat area might be absorbed to the soil completely and disappear while dropped droplets on the inclined area will move downward and connect to other droplets to form a stream. Being lucky, some streams might connect to each other and form a river. If there is an obstacle in the path of the streams or rivers, some lakes will be created in which the volume of the water implies the importance of it. Very soon after finishing the rain, streams and rivers would be discharged to the local lakes, and after a while, small lakes might be vanished due to evaporation of water in the lake or absorption to the soil. Therefore, just a few significant lakes can be remained in the ground depending on the topology of the earth's surface and properties of the soil. These lakes show the local minimum of the ground surface and deeper lake shows the global minimum.

By changing the type of rain, the previously mentioned scenario might be changed a little. For example, if it is heavy rain with large droplets, all of the droplets will be connected to each other very fast without any absorption or evaporation resulting in a flood. In this case, just the global minimum can be detected as all local minimum are connected to each other due to a rainstorm. On the other hand, when there is a light rain with small droplets, all of the droplets might absorb to the soil resulting in no stream formation. Therefore, it can be realized that parameter tuning has significant importance while using ROA.

The movement of the particle in the proposed method is similar to gradient-based optimization methods and that of traditional single-point algorithms such as hill-climbing (HC) and gradient-descent and Rain-Fall Optimization algorithm (RFO) (Aghay Kaboli et al., 2016). These methods adjust only one parameter in each iteration to find if changing this parameter improves cost function or not. However, ROA uses a set of answers that all of them move toward the optimum simultaneously. In this movement, some of their properties will change in each iteration. For example, their size might change or they might eliminate. In addition, ROA is able to find all extremum points instead of just a minimum or maximum.

3. ROA algorithm

In this section, we would try to simulate rain behavior as it was described in the previous section. Each solution of the problem can be modeled by a raindrop. Depending on the problem, some points in the answer space can be selected randomly as the raindrops fall in the ground randomly. The main property of each drop of rain is its radius. The radius of every raindrop can be reduced as time goes by and it can be increased as a raindrop is connected to other drops. When the initial population of answers is produced, the radius of each droplet can be assigned randomly in an appropriate range. In each iteration, every droplet checks its neighborhood dependent on its size. Single droplets that are not still connected to any other droplet, just check for the end limit of the place that it has covered. When we are solving a problem in n-dimensional space, every droplet consists of n variable. So at the first step, the lower and upper limit of variable one will be checked as these limits would be determined by the radius of the droplet. At the next step, two endpoints of variable two would be tested and this is continued until the last variable. In this stage, the cost of the first droplet would be updated by moving it downward. This is not the end action for this droplet and while cost function is reducing, it will move downward in the same direction. This action will be performed for all droplets, then the cost and position of all droplets will be assigned. The radius of each droplet will be changed in two manners:

1 If two droplets with radius r_1 and r_2 are so close to each other that has a common area with each other; they can connect to form a larger droplet of radius R:

$$R = \left(r_1^n + r_2^n\right)^{1/n} \tag{1}$$

where n is the number of variables in each droplet.

2 If a droplet with radius r_1 does not move, depending on the soil properties, which is shown by α, some volume percentage of it can be adsorbed.

$$R = \left(\alpha\, r_1^n\right)^{1/n} \tag{2}$$

In fact, α shows the percentage of the volume of a droplet which can be absorbed in each iteration and is a number between 0 and 100 percent. We also can define a minimum for droplets radius r_{min}, where droplets with a smaller radius of that r_{min} will disappear.

As it can be considered, the population number would be decreased after a few iterations and larger droplets will be developed with a larger domain of investigations. By increasing the domain of investigation for each drop, the local searching ability of drops is increased proportionally to the diameter of the droplets. Therefore by increasing the number of iteration, weak droplets with a low domain of investigation disappear or connect to stronger drops with a higher domain of investigation and the initial population will decrease intensively caused increasing speed of finding the correct answer(s).

It should be considered that there are some important differences between the proposed optimization algorithm in this work Rain Optimization Algorithm (ROA) and the recently developed search algorithm by Aghay Kaboli et al. (2016) named Rain Fall Algorithm (RFA) which can be summarized as follow:

- In the ROA despite RFA and many other search algorithms, initial population number changes after each iteration due to the connection of adjacent drops or adsorption by the soil. This issue leads to an increase in the searching ability of the algorithm and decreases the optimization cost seriously.
- After each iteration size of each drop changes due to the connection of near droplets or adsorption by the soil. This action changes the searching ability of each droplet and categorizes the droplets from the viewpoint of importance.
- In the RFA and many other search algorithms, in each iteration, each population would be comprised by some other random neighbor points and the droplet would be improved one step randomly. On the other hand, in the ROA, each population finds the best path to the minimum point. After finding the path, it moves toward the downside step by step while the cost function is decreasing just in one iteration. This causes the initial population to leave the incompetent points very fast.

Based on the approximations and idealizations mentioned above, rain algorithm can be summarized in Fig. 1. Briefly, tunning parameters of this algorithm such as initial raindrops number (population number), initial raindrops radius (search space for each population), etc., will be entered in the first part of the algorithm. Then a value would be assigned to each droplet according to the cost function. After that, each droplet starts to move downward. For this issue, the endpoints of each droplet would be checked by the cost function. When a droplet starts to move, it will continue its route until getting to a minimum in its way. This

```
objective function f(X), X=(x1,x2,...,xn)

Input initial tuning parameters such as population number (nPop), maximum iteration (MaxIt), number of variables (nVar), the domain of variables ([VarMin, VarMax]), initial droplet's radius (InitR) and number of jointed droplets (size), rain speed (Speed) and Soil adsorption Constant(α).

Initialize droplets position, radius and size.

Evaluate each droplet with the objective function to obtain the cost of each droplet and sort population based on cost.

Main loop:
    While (iteration number < MaxIt)
        For( each droplet)
            Change each variable xi to xi+Ri and xi-Ri and evaluate the new position by the objective function.
            If the new cost is smaller then the previous cost, accept a new position for xi.

            while (cost reduces)
                move the droplet at the same direction with the same velocity,
                reduce size of droplet depending on the soil adsorption properties
                joint near droplets to each other, change size of new droplets
            end while

        end for
        omit weak droplets depending on soil adsorption
        generate new droplets depending on rain speed
    end while

Sort populations based on cost.

Show results and visualizations.
```

Fig. 1. Pseudocode for rain optimization algorithm (ROA).

scenario would be repeated for each droplet. In their path, near droplets could joint with each other, causing algorithm speed to increase significantly. When a droplet stops to a minimum point, its radius starts to reduce gradually causing the accuracy of the answer to increase notably. In this method, the algorithm is able to find all extremum points of the objective function. A simple version of the implementation of the ROA can be found in Appendix A.

Rain Optimization Algorithm.

4. Validation and comparison

It is not hard work to implement the rain algorithm from the mentioned steps in the previous section using any programming language. We implemented this algorithm using Matlab software for various test functions to visualize the results and compare it with other metaheuristic algorithms. As it was emphasized before, the strong point of this algorithm is in finding the local minimums with a high degree of accuracy, and this is what other algorithms cannot do it so easily. For validating and testing, standard tools have been used, in a similar way to test other new algorithms such as bat algorithm (yang, 2010). Therefore, we have considered the performance of this algorithm from three perspectives:

Perspective 1: considering the performance of the ROA using two benchmark functions in detail

Perspective 2: considering its performance on solving 26 benchmark functions regardless of the number of function evaluation (NFE) compared to some other optimization algorithm

Perspective 3: considering its performance in solving drilling optimization problems

4.1. Method of performance of ROA

We have chosen the following functions as the benchmark functions for considering the method of solving a problem using ROA:

1 Eggcrate function

$$z = x^2 + y^2 + 25\left(\sin^2(x) + \sin^2(y)\right), \qquad -5 < x < 5, \qquad -5 < y < 5 \quad (3)$$

Fig. 2 shows Eggcrate function in 3D view within the defined domain for x and y. We know that this function has a global minimum of zero at x = 0 and y = 0. Also following local minimum can be determined for this function:

We run the ROA algorithm for finding global and local minimums with the following algorithm parameter:

Initial raindrops number = 1000;
Variables number of each raindrop = 2;
Maximum iteration = 100;
Initial raindrops diameter = 0.038;

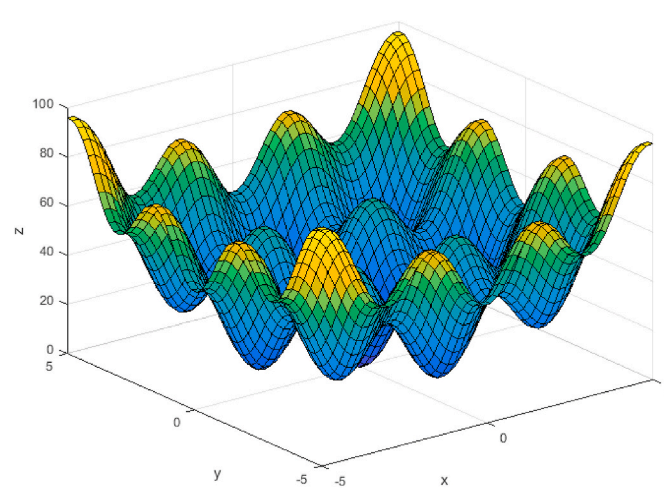

Fig. 2. 3D plot of Eggcrate function.

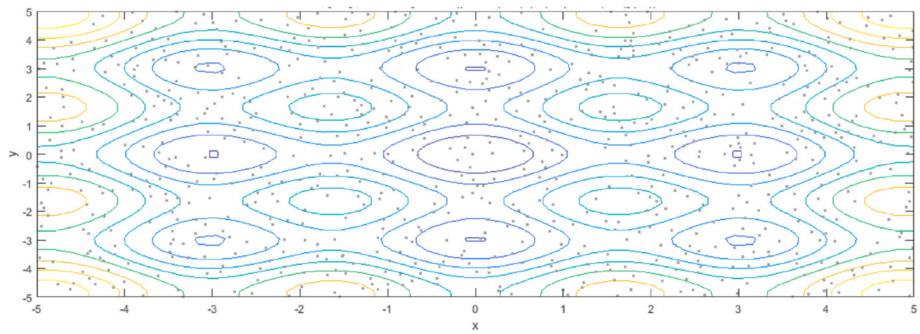

Fig. 3. Distribution of initial population on the problem area.

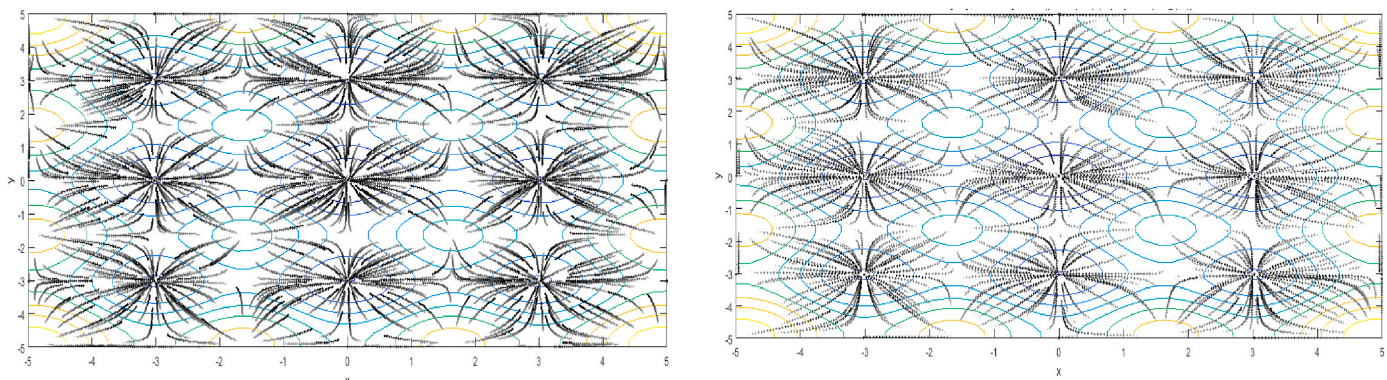

Fig. 4. (a): the moving path of the raindrops after 30 iterations on the problem area (left), 4(b): the moving path of the raindrops after 100 iterations on the problem area (right).

Table 1
Position and size of rain droplets after 100 iteration.
2)Rosenbrock's function

X	Y	Z	lake radius	Jointed droplet
0.00	−3.02	9.488197	0.497675	43
−3.02	−3.02	18.97639	0.467846	38
0.00	0.00	8.65E-35	0.448999	35
−3.02	3.02	18.97639	0.422564	31
0.00	3.02	9.488197	0.422564	31
3.02	0.00	9.488197	0.408705	29
−3.02	0.00	9.488197	0.39436	27
3.02	3.02	18.97639	0.363978	23
3.03	−3.01	18.97931	0.131453	3
3.05	3.97	38.86377	0.107331	2
−2.34	0.00	18.33812	0.107331	2
−2.37	0.53	24.6003	0.107331	2
3.55	0.00	16.60245	0.107331	2
0.47	3.02	14.8654	0.107331	2
0.01	4.11	33.88256	0.107331	2
3.02	3.47	24.23667	0.107331	2
−2.59	0.00	13.56718	0.107331	2
−3.02	0.40	13.44619	0.107331	2
0.00	5.02	47.90627	0.107331	2
3.04	3.86	35.26616	0.107331	2
−2.97	0.90	25.57681	0.107331	2
0.01	0.73	11.53468	0.107331	2
3.48	3.02	24.27566	0.107331	2
0.01	−2.13	22.54699	0.107331	2

Initially created droplets are distributed on the problem area as it can be seen in Fig. 3; also, the location of raindrops in iteration 1 to 30 can be seen in Fig. 4(a). Droplets of the latest iterations are darker and jointed droplets have a larger diameter. As it is obvious from Fig. 4(b), the raindrops are running from the maximum points toward the minimums and in their route, some droplets joint to create streams. The route of the raindrops after 100 iterations is shown in Fig. 4(b). After 100 iterations just 70 droplets with various size remain on the surface, some of these droplets are very big and has created some lakes. It is obvious that these lakes are local minimums of the function and the deepest one is the global minimum. The results of the algorithm are shown in Table 1. Some small and unimportant lakes can be seen in Fig. 4(b) and also in Table 1. If we run the algorithm for more iterations, these small lakes will move toward local minimums or might absorb to the soil, although this is not important and from the magnitude of the lakes we can find out which lake is more important.

Rosenbrock's function was introduced by Howard Rosenbrock in 1960 (Rosenbrock, 1960) and has a global minimum inside a long, narrow, parabolic shaped flat valley and can be defined by

$$f(x) = \sum_{i=1}^{d-1} \left(1 - x_i^2\right)^2 + 100\left(x_{i+1} - x_i^2\right)^2, \qquad -2.048 < x_i < 2.048 \tag{4}$$

Rosenbrock's function has a global minimum at $(x_1, x_2) = (1, 1)$, where f(x) = 0. Fig. 5 shows Rosenbrock's function in 3D view within the defined domain for x and y with a = 1 and b = 100.

Fig. 5 shows the shape of Rosenbrock's function in 3D view and Fig. 6 (a) shows the initially selected population for solving the problem which is produced randomly. The results of the algorithm after 100 iterations are shown in Fig. 6(b).

As it can be noticed from the produced answer for Egg crate and Rosenbrock functions, in the rain algorithm, initial populations move gradually and slowly move toward the minimums and in their path can joint to each other to form larger droplets with more local searching ability. In addition, week droplets that cannot improve themselves would be absorbed into the soil and vanished. Therefore, at the initial iteration, we have a lazy algorithm that should check lots of probable answers, but just after a few iteration lots of these answers or droplets

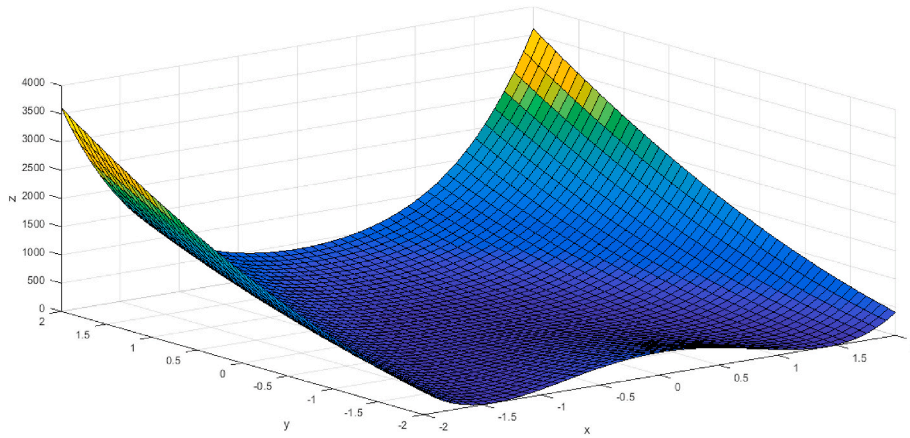

Fig. 5. Rosenbrock function in 3D view within the defined domain for x and y with a = 1 and b = 100.

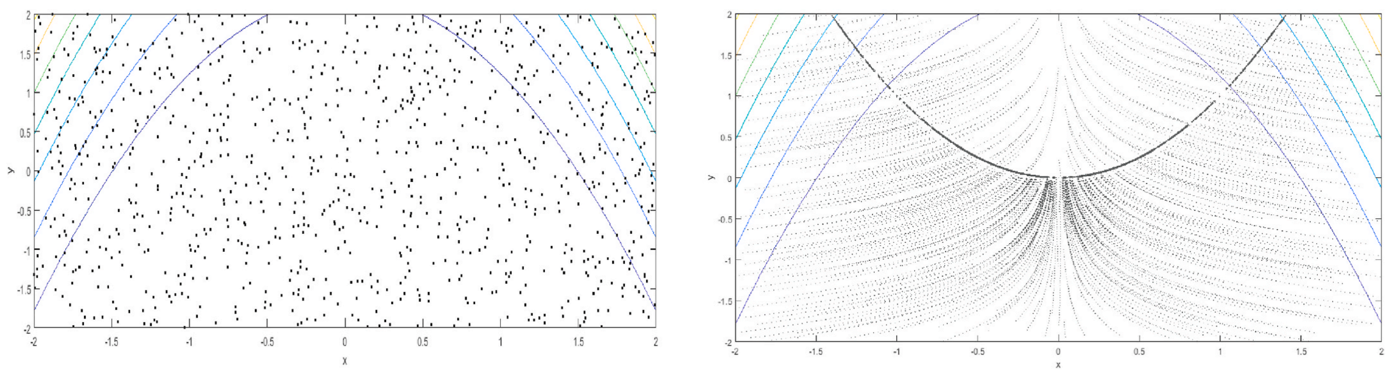

Fig. 6. (a): Rosenbrock function in 2D view and position of the initial population (left), 6(b): Results of the rain algorithm after 100 iterations (1000 rain droplets and Rosenbrock's function) (right).

will joint to each other or absorbed to the soil and speed of the algorithm will increase rapidly.

4.2. Solving 26 benchmark functions

As it can be seen in Table 2, we choose 26 important benchmark functions and solve them using Rain Optimization Algorithm and compare its efficiency with some other optimization algorithms such as Genetic algorithm (GA), Particle swarm (PSO), bat algorithm (BA) and Sperm Whale Algorithm (SWA) (Cheng and Lien, 2012). These problems were solved and published by Cheng and Lien (2012) except SWA that was solved by Ebrahimi and Khamechi (Ebrahimi and Khamehchi, 2016). Optimization algorithm parameters that are used for solving these benchmark functions can be found in Table 3.

For solving these benchmark functions, maximum number of function evaluations greater than $5*10^5$ was not allowed. Also in this algorithm, any value less than 10^{-12} was assumed to be zero. We perform the simulations using Matlab software on a 2 GHz laptop. Furthermore, we have tried to use different population sizes from n = 10 to 250, and a fixed population size n = 50 for all simulations were applied. Tuning parameters for ROA was as follow: Population size = 50, rain speed = 10, rain radius = 0.05 (Xmax-Xmin), soil adsorption = 50%. In addition, the results of the power of various algorithms for solving these benchmark functions are shown in Table 4. As it may be notified, the score and rank of each algorithm is shown in the two last columns of Table 4. Results show the ROA, SWA, BA, PSO and GA algorithms are respectively the best algorithms. In addition, ROA has the best rank with a slight difference compared to SWA.

5. Solving a drilling optimization problem using ROA

Today, the cost of oil and gas production is a deterministic factor in the industry (Skjerpen et al., 2018). Therefore, the main objective of drilling optimization is to reduce drilling time and cost. For this purpose there are two main methods, reducing drilling time by selecting optimum drilling parameters before drilling (for example, selecting a suitable drilling fluid or bit) and reducing drilling time by selecting optimum drilling parameters in real-time drilling operations (for example, optimizing weight on bit or pump pressure) (see e.g. Eren and Ozbayoglu, 2010; Payette et al., 2017). For optimizing drilling parameters, there should be an accurate predictive model (Barbosa et al., 2019). This model should be able to relate important drilling parameters (such as rotary bit speed, weight on bit, etc.) to the drilling rate with acceptable accuracy (Soares and Gray, 2019). Despite many efforts for developing an effective model for ROP (analytically or experimentally), the results are not so satisficing (Soares and Gray, 2019). Therefore, many investigators prefer to employ machine-learning methods (for example, artificial neural networks, genetic algorithms, random forest, etc.) for ROP prediction. (Hornik et al., 1989). Most studies on the comparison between analytical modeling methods and intelligent methods concluded that more accurate models could be obtained using intelligent methods. (for example: Arabjamaloei and Shadizadeh, 2011; Amar and Ibrahim, 2012; Bataee et al., 2014; Hegde et al., 2017). Lots of work on ROP prediction using machine learning (ML) methods can be found in the. The newest method employed for ROP Prediction combines traditional methods with a machine learning method that is called a hybrid method (Yavari et al., 2018). So, ROP prediction methods can be

Table 2
Details of benchmark functions (Ebrahimi and Khamehchi, 2016).

No	Name	Range	D	Formulation	Min
1	Rastrigin	[-5.12,5.12]	n	$f_1(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	0
2	De Jong (Sphere)	[-5.12,5.12]	n	$f_2(x) = \sum_{i=1}^{n} {x_i}^2$	0
3	Griewank	[-600,600]	n	$f_3\left(x\right) = \frac{1}{4000}\sum_{i=1}^{n} {x_i}^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0
4	Beale	[-4.5,4.5]	2	$f_4(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + \ldots$ $\ldots(2.625 - x_1 + x_1x_2^3)^2$	0
5	Easom	[-100,100]	2	$f_5x) = -\cos x_1)\cos x_2)\exp - (x_1 - \pi)^2 - (x_2 - \pi)^2)$	−1
6	Matyas	[-10,10]	2	$f_6(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	0
7	Bohachevsky1	[-100,100]	2	$f_7(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	0
8	Booth	[-10,10]	2	$f_8(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	0
9	Michalewicz2	[0, π]	2	$f_9(x) = -\sum_{i=1}^{D}\sin x_i)\sin(ix_i^2/\pi))^{20}$	−1.8013
10	Schaffer	[-100,100]	2	$f_{10}\left(x\right) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	0
11	Six Hump Camel Back	[-5,5]	2	$f_{10}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	−1.03163
12	Boachevsky2	[-100,100]	2	$f_{12}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)(4\pi x_2) + 0.3$	0
13	Boachevsky3	[-100,100]	2	$f_{13}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	0
14	Shubert	[-10,10]	2	$f_{14}(x) = (\sum_{i=1}^{5} i\cos(i + 1)x_1 + i)(\sum_{i=1}^{5} i\cos((i + 1)x_2 + i))$	−186.73
15	Colville	[-10,10]	4	$f_{15}(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + ..$ $..10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$	0
16	Michalewicz5	[0, π]	5	$f_{16}(x) = -\sum_{i=1}^{D}\sin x_i)\sin(ix_i^2/\pi))^{20}$	−4.6877
17	Zakharov	[-5,10]	10	$f_{17}(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5ix_i^2)^2 + (\sum_{i=1}^{D} 0.5ix_i^2)^4$	0
18	Michalewicz10	[0, π]	10	$f_{18}(x) = -\sum_{i=1}^{D}\sin x_i)\sin(ix_i^2/\pi))^{20}$	−9.6602
19	Step	[-5.12,5.12]	30	$f_{19}(x) = \sum_{i=1}^{D}(x_i + 0.5)^2$	0
20	SumSquares	[-10,10]	30	$f_{20}(x) = \sum_{i=1}^{D} ix_i^2$	0
21	Quartic	[-1.28,1.28]	30	$f_{21}(x) = \sum_{i=1}^{D} ix_i^4 + Rand$	0
22	Schwefel 2.22	[-10,10]	30	$f_{22}(x) = \sum_{i=1}^{D}\lvert x_i\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$	0
23	Schwefel 1.2	[-100,100]	30	$f_{23}(x) = \sum_{i=1}^{D}(\sum_{j=1}^{D} x_j)^2$	0
24	Rosenbrock	[-30,30]	30	$f_{24}(x) = \sum_{i=1}^{D} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	0
25	Dixon-Price	[-10,10]	30	$f_{25}(x) = (x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_i - 1)^2$	0
26	Ackley	[-32,32]	30	$f_{26}\left(x\right) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	0

Table 3
Optimization algorithm parameters used in solving benchmark functions in this work.

Optimization algorithm	parameters
GA	population size = 50; mutation rate = 0.01; crossover rate = 0.8; generation gap = 0.9
PSO	population size = 50; inertia weight = 0.9–0.7; limit of velocity = X_{max}/10- X_{min}/10
BA	population size = 50; elite bee number = n/2; best bee number = n/4; random bee number = n/4; elite bee neighborhood number = 2, best bee neighborhood number = 1
SWA	Number of main groups = 10; group size = 5; good gang size = 2; local search iteration = 10

classified as follow (Fig. 7):

- ✓ analytical models,
- ✓ statistical models (e.g. multiple regression),
- ✓ machine learning models (e.g. genetic algorithm or artificial neural networks),
- ✓ hybrid models (e.g. combining analytical models with machine learning models).

In this work, we used a new hybrid method for ROP prediction and optimization. In this method, a new analytical model was developed in the first part. Then this model was solved using ROA (the new search model that was described in the previous section).

5.1. Developing an analytical model for ROP

A useful method for increasing the drilling rate is to reduce 'mechanical specific energy (MSE)'-the amount of work done for the excavating unit volume of rock- (Teale, 1965). In rotary table drilling, work is done both by the piercing the bit, WOB (lb), and the exerted force while rotation of bit or torque, T (lb-ft). It can be shown that the total work done by bit in 1 h in lb-ft is:

$$W = WOB \times ROP + 60 \times 2\pi N \times T \tag{5}$$

where:

N is rotation speed in rev/min
WOB is the weight on the bit in lb
T is the torque in lb-ft
ROP is penetration rate in ft/hr
W is the work done for removing the rock in lb-ft/hr

Therefore the volume of excavated rock in 1 h is:

Table 4
Optimization algorithms performance comparison on benchmark functions.

f	D	Min		GA	PSO	BA	SWA	ROA
$f_1(x)$	30	0	Mean	52.92259 (3)	43.9771369 (2)	0 (1)	0 (1)	0 (1)
			SD	4.56486	11.728676	0	0	0
$f_2(x)$	30	0	Mean	1.11 E+03 (2)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	74.21447	0	0	0	0
$f_3(x)$	30	0	Mean	10.63346 (3)	0.01739 (2)	0 (1)	0 (1)	0 (1)
			SD	1.16146	0.02081	0	0	0
$f_4(x)$	2	0	Mean	0 (1)	0 (1)	1.88E-05 (2)	0 (1)	0 (1)
			SD	0	0	1.94E-05	0	0
$f_5(x)$	2	−1	Mean	−1 (1)	−1 (1)	−0.99994 (2)	−1 (1)	−1 (1)
			SD	0	0	4.50E-05	0	0
$f_6(x)$	2	0	Mean	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	0	0	0	0	0
$f_7(x)$	2	0	Mean	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	0	0	0	0	0
$f_8(x)$	2	0	Mean	0 (1)	0 (1)	0.00053 (2)	0 (1)	0 (1)
			SD	0	0	0.00074	0	0
$f_9(x)$	2	−1.8013	Mean	−1.8013 (1)	−1.57287 (2)	−1.8013 (1)	−1.8013 (1)	−1.8013 (1)
			SD	0	0.11986	0	0	0
$f_{10}(x)$	2	0	Mean	0.00424 (2)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	0.00476	0	0	0	0
$f_{11}(x)$	2	−1.0316	Mean	−1.03163 (1)	−1.03163 (1)	−1.03163 (1)	−1.03163 (1)	−1.03163 (1)
			SD	0	0	0	0	0
$f_{12}(x)$	2	0	Mean	0.06829 (2)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	0.07822	0	0	0	0
$f_{13}(x)$	2	0	Mean	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	0	0	0	0	0
$f_{14}(x)$	2	−186.73	Mean	−186.73 (1)	−186.73 (1)	−186.73 (1)	−186.73 (1)	−186.73 (1)
			SD	0	0	0	0	0
$f_{15}(x)$	4	0	Mean	0.01494 (4)	0 (1)	1.1176 (5)	0.00544 (3)	0.00053 (2)
			SD	0.00736	0	0.46623	0.00063	0.00032
$f_{16}(x)$	5	−4.6877	Mean	−4.64483 (2)	−2.49087 (3)	−4.6877 (1)	−4.6877 (1)	−4.6877 (1)
			SD	0.09785	0.25695	0	0	0
$f_{17}(x)$	10	0	Mean	0.01336 (2)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	0.00453	0	0	0	0
$f_{18}(x)$	10	−9.6602	Mean	−9.49683 (3)	−4.00718 (4)	−9.6602 (1)	−9.61387 (2)	−9.6602 (1)
			SD	0.14112	0.50263	0	0.00236	0
$f_{19}(x)$	30	0	Mean	1.17 E+03 (3)	0 (1)	5.1237 (2)	0 (1)	0 (1)
			SD	76.56145	0	0.39209	0	0
$f_{20}(x)$	30	0	Mean	1.48 E+02 (2)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	12.40929	0	0	0	0
$f_{21}(x)$	30	0	Mean	0.1807 (4)	0.00116 (3)	1.72 E−06 (2)	0 (1)	0 (1)
			SD	0.02712	0.00028	1.85E-06	0	0
$f_{22}(x)$	30	0	Mean	11.0214 (3)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	1.38686	0	0	0	0
$f_{23}(x)$	30	0	Mean	7.40 E+03 (2)	0 (1)	0 (1)	0 (1)	0 (1)
			SD	1.14 E+03	0	0	0	0
$f_{24}(x)$	30	0	Mean	1.96 E+05 (3)	15.088617 (2)	28.834 (3)	13.36393 (2)	9.4536 (1)
			SD	3.85 E+04	24.170196	0.10597	4.0295	3.4381
$f_{25}(x)$	30	0	Mean	1.22 E+03 (3)	0.66667 (2)	0.66667 (2)	0 (1)	0 (1)
			SD	2.66 E+02	E−08	1.16E-09	0	0
$f_{26}(x)$	30	0	Mean	14.67178 (3)	0.16462 (2)	0 (1)	0 (1)	0 (1)
			SD	0.17814	0.49387	0	0	0
Score				**48**	**36**	**35**	**30**	**27**
Final Rank				**5**	**4**	**3**	**2**	**1**

$$V = A \times ROP \tag{6}$$

So, the mechanical specific energy (in lb/in^2) can be computed by dividing work by volume:

$$MSE = \frac{WOB}{A} + \frac{60 \times 2\pi N \times T}{A \times ROP} \tag{7}$$

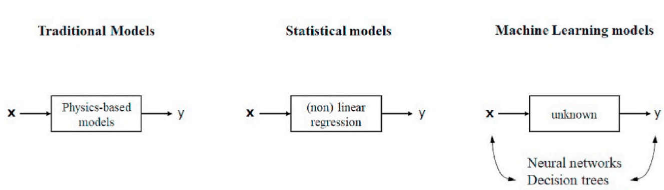

Fig. 7. Approaches for ROP modeling (Barbosa et al., 2019).

where A is the area of the hole in^2

Teale (1965) pointed out that the minimum energy required for cracking a rock in all cases is of the order of the uniaxial compressive strength (UCS) of that rock.

Teale's model for specific energy although was new at that time but contained some significant source of error. Some of these errors were corrected by some researchers, and some of them still are existing. The weakness of Teale model are as follow:

1 Teale conducted all his tests under atmospheric conditions, so he underestimated minimum specific energy. Some other researchers by performing more exact experiments show that the minimum energy required for cracking an in-situ rock is of the order of the confined compressive strength (CCS) of that rock.
2 Although Teale used surface measured torque for solving his equation but it is clear that surface measured torque is quite different

Table 5
Summary of previous researches for optimizing drilling operations.

Researcher(s)	Main developed idea or work	Weakness (es) of work
Simon (1963)	Experimentally measured the magnitude of the work required to break out a unit volume of rock.	Did not develop a model for MSE.
Teale (1965)	Developed first equation for computing MSE. Developed a method for ROP optimization using MSE.	T and WOB where used from surface data instead of bottom hole data. The hydraulic effect was missing. Bit efficiency was ignored, …
Bourgoyne and Young (1974)	Developed a comprehensive model for predicting ROP	There were several constants in the equation which should be determined in the various situation
Warren (1987) Winters et al. (1987)	develop models and formulate related parameters to ROP	Could not make an accurate and comprehensive estimation
Pessier and Fear (1992)	Developed a relation for T based on WOB for using in Teale's model. Improved Teale's method for ROP optimization based on MSE.	WOB was used from surface data which can be different from real bottom hole WOB. The hydraulic effect was missing.
Waughman et al. (2003)	Provided a method for including bit wear to the Teale's equation.	The hydraulic effect was missing. Measurement of real WOB was missing
Dupriest and Koederitz (2005)	Included bit mechanical efficiency to the Teale's equation. Improved Teale's method for ROP optimization based on MSE.	Bit efficiency was not exact. The hydraulic effect was missing. Measurement of real WOB was missing
Rahimzadeh et al. (2010) Edalatkhah et al. (2010) Monazami et al. (2012)	Used intelligent methods (ANN) for predicting ROP	Their method was not real-time Solving method was time-consuming Parameter tuning was required for the network
Cherif and Bits (2012) Amadi and Iyalla (2012)	Included bit mechanical efficiency to the Teale's equation. Improved Teale's method for ROP optimization based on MSE.	Bit efficiency was not exact. The hydraulic effect was missing. Measurement of real WOB was missing
Chen et al. (2014)	developed a formula between bottom hole weight on bit and surface weight on bit improved Teale's MSE model	The hydraulic effect was missing.
Mohan et al. (2015)	Included hydraulic effect to the Teale's MSE model. Introduced HMSE concept for the first time.	Bit efficiency was not exact.

from real exerted torque to the bit. Also measuring exerted torque from bit to the beneath rock has proven difficult.

3 Measured WOB in the surface can be completely different from real WOB at the bottom of the hole, especially in deviated and horizontal wells due to the effect of drag and pump-off force of drilling fluid.
4 There is not any term for the hydraulic effect of drilling fluid in Teale's equation, but it is clear that the hydraulic power of mud can effect on drilling process especially in soft formations.
5 In directional drilling, when a downhole motor is used, exerted torque and RPM by bit are quite different from measured torque and RPM on the surface.
6 Effect of bit type and bit efficiency is neglected in the equation but it is obvious that different bits have different efficiencies in the same conditions.
7 The effect of bit wear is missing in this equation.

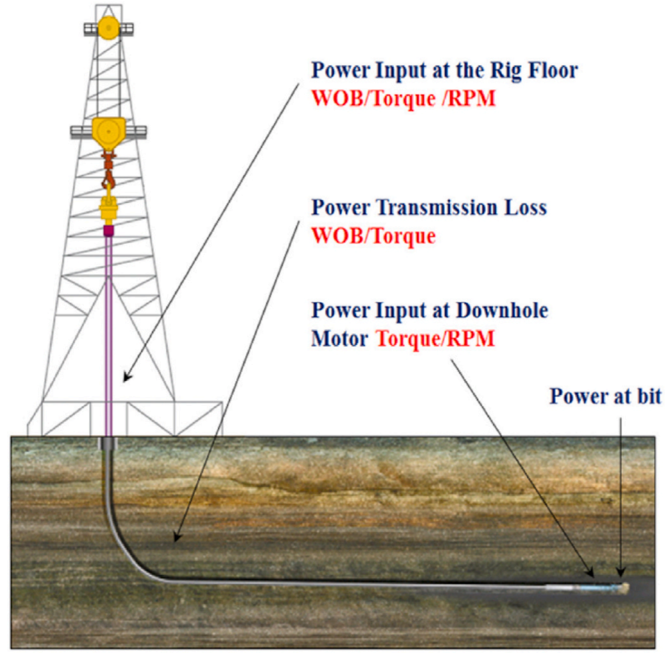

Fig. 8. Rotary drilling system with PDM (Chen et al., 2016).

8 Drilling problems such as bit balling and drill pipe vibration can effectively reduce the drilling rate and change drilling efficiency that is not seen in the Teal's equation.

Some researchers solve some of the mentioned problems, but some of them are still existing (Table 5).

5.2. Improving the model of ROP

Many researchers try to improve the weaknesses of Teal's equation. Pessier and Fear, 1992 stated that in rotary-drilling with PDM (Fig. 8) the total mechanical work done by the bit in 1 h can be estimated by

$$W_t = (WOB_b \times N) + (60 \times 2\pi \times N_s \times T_s) + (60 \times 2\pi \times N_m \times T_m) \tag{8}$$

where:

N_s: bit rotary speed provided by surface rotation;
T_s: torque at bit provided by surface rotation;
N_m: PDM output rotary speed;
T_m: PDM output torque.

Cherif and Bits, 2012 stated that every bit has a mechanical efficiency related to its cutter size and structure. Including mechanical efficiency to the Pessier's equation, the mechanical work required to break the rock drilled in 1 h can be nearly expressed as:

$$W_v = W_t.E_m \tag{9}$$

The volume of rock drilled in 1 h is

$$V = \mathrm{A}.ROP \tag{10}$$

MSE was defined as the mechanical work done to excavate a unit volume of rock (Teale, 1965). By combining Eqs. (8)–(10), then the MSE for rotating drilling with PDM can be expressed by

$$MSE = \frac{W_v}{V} = E_m.\frac{WOB_b.ROP + 60.2\pi.N_s.T_s + 60.2\pi.RPN_m.T_m}{\mathrm{A}.ROP} \tag{11}$$

However, the mechanical energy provided by the surface has a significant transmission loss in horizontal and directional drilling. Chen et al. (2014) formulated a relationship between bottom hole WOB and the surface measured WOB and presented a method to calculate torque of bit in directional and horizontal drilling.

Chen et al. (2014) stated that provided mechanical energy on the surface has a great difference with the mechanical energy received by the bit due to friction between pipes and borehole, especially in directional and horizontal drilling. So he formulated a relation between surface measured WOB and bottom hole WOB as well as Surface and bottom hole Torque:

$$WOB_b = WOB.e^{-\mu_s.\gamma_b} \tag{12}$$

$$\mu_b = 36\frac{T}{D_b.WOB.e^{-\mu_s.\gamma_b}} \tag{13}$$

Then the mechanical specific energy provided by the surface can be estimated as

$$E_m.\frac{WOB_b.ROP + 60.2\pi.N_s.T_s}{\mathrm{A}.ROP} = E_m.WOB.e^{-\mu_s.\gamma_b}.\left(\frac{1}{\mathrm{A}} + \frac{13.33.\mu_b.N_s}{D_b.ROP}\right) \tag{14}$$

Also According to Equations (13) and (14), Chen et al. (2014) deduced that the mechanical specific energy provided by the PDM can be estimated as

$$E_m\frac{60.2\pi.\mathrm{N_m}.T_m}{\mathrm{A}.ROP} = E_m.\frac{1155.2.\eta.\Delta P_m.Q}{\mathrm{A}.ROP} \tag{15}$$

Finally, substitute Equations (15) and (14) into Equation (11), Chen et al. (2014) get a new MSE model for rotating drilling with PDM

$$MSE = E_m.\left(WOB.e^{-\mu_s.\gamma_b}.\left(\frac{1}{\mathrm{A}} + \frac{13.33.\mu_b.N_s}{D_b.ROP}\right) + \frac{1155.2.\eta.\Delta P_m.Q}{\mathrm{A}.ROP}\right) \tag{16}$$

where in this equation:

ΔP_m: Pressure drop across the PDM, psi
Q: Pump flow rate, gpm
D_b: Bit diameter, in
η: Efficiency of PDM
Ns: Drill pipe rotary speed, rpm
Em: Mechanical efficiency of the bit
γ_b: Bit sliding coefficient (between 0.3 and 0.85)
μ_s: Drill string sliding coefficient (between 0.25 and 0.4)

Although derived equation by Chen, improved lots of weaknesses of Teal's equation, but there are some coefficients ($\mathrm{E_m}$, μ_s,γ_b, μ_b, η) in this equation that changes in the various situation of drilling. Therefore, ROA was used to find out these coefficients in various conditions of drilling.

Rewriting Equation (16) yields:

$$\mathrm{MSE} = E_m\left(\frac{WOB.e^{-\mu_s.\gamma_b}}{\mathrm{A}}\right) + \left(\frac{\left(\frac{\pi}{4}.D_b\right).E_m.13.33.\mu_b.WOB.e^{-\mu_s.\gamma_b}.N_s}{\mathrm{A}.ROP}\right) + \left(\frac{E_m.1155.2.\eta.\Delta P_m.Q}{\mathrm{A}.ROP}\right) \tag{17}$$

Letting

$$a_1 = \mu_s.\gamma_b$$

$$a_2 = \left(\frac{\pi}{4}.D_b\right).13.33.\mu_b$$

$$a_3 = 1155.2.\eta$$

Equation (17) can be summarized as follow:

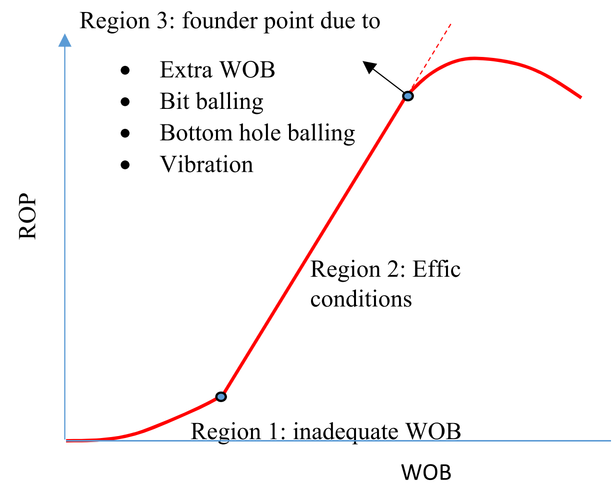

Fig. 9. a typical drill off test in drilling operations (Dupriest and Koederitz, 2005).

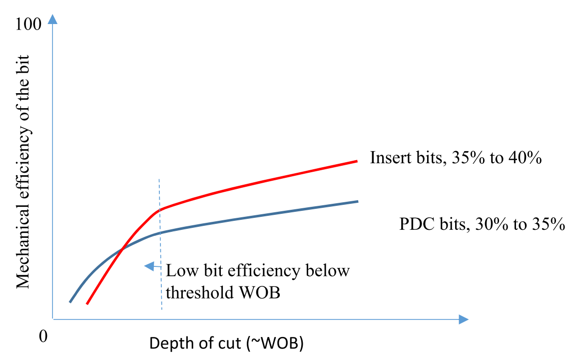

Fig. 10. Typical relation between depth of cut (equivalent to WOB) and bit efficiency (Pessier and Fear, 1992)

$$\mathrm{MSE} = E_m\left(\left(\frac{WOB.e^{-a_1}}{\mathrm{A}}\right) + \left(\frac{a_2.WOB.e^{-a_1}.N_s}{\mathrm{A}.ROP}\right) + \left(\frac{a_3.\Delta P_m.Q}{\mathrm{A}.ROP}\right)\right) \tag{18}$$

As it can be seen in Equation (18), there are some empirical constants named a_1 *to* a_3 in the model. We will try to find these constants for various situations using ROA.

Rearranging Equation (18) for ROP we have:

$$ROP = \frac{a_2N_s.WOB.e^{-a_1} + a_3\Delta p_m.q}{\frac{A.MSE}{E_m} - WOB.e^{-a_1}} \tag{19}$$

5.3. A brief discussion on the value of MSE

In order to know the manner in which the MSE works in Equation (19), it is necessary to discuss the method that bits drill the rock and factors that affect the bit performance. Fig. 9 shows a typical drill off test in the drilling operations. This curve is divided into three regions.

In region 1, the ROP is very low, and it increases gradually by increasing WOB. In this region, drilling operation is suffering from low WOB and it can be said that operation is nearly stopped. In region 2, there is a linear relation between ROP and WOB. In this section, WOB is so enough that drilling can be started. In this section by increasing WOB, ROP increases linearly. This region continues until region 3 where ROP increase stops and reversely starts to decrease.

Fig. 10 shows the typical relation between the depth of cut (equivalent to WOB) and bit efficiency. As the WOB and resulting depth of cut increases, bit efficiency increases. Bit efficiency can be defined as the

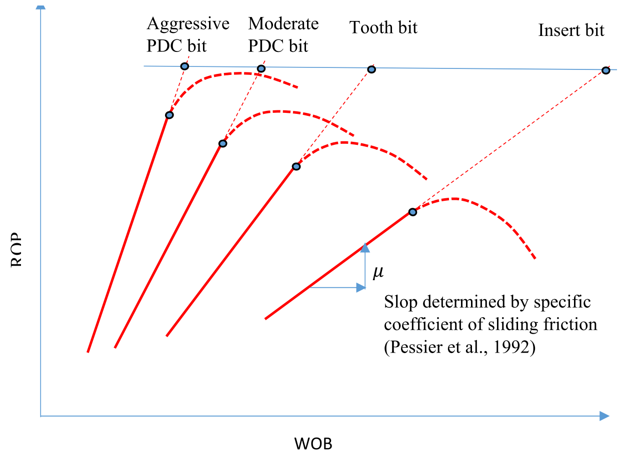

Fig. 11. National plot showing that the slop of drill off test is determined by μ and RPM, but the maximum ROP is limited by founder point (Dupriest and Koederitz, 2005).

required energy to remove a specific volume of rock to the actual energy used for drilling this volume. Bits tend to transfer 30%–40% of their input energy to the rock, even when operating at peak performance (Dupriest and Koederitz, 2005).

In the linear portion of Fig. 9, although by increasing WOB, drilling efficiency does not change but the provided energy to the bit would be increased, causing the drilling rate to increase. The slope of the line is constant for a special formation, bit and rotary speed. Fig. 11 shows the national relation between the slope of the straight line in the drill off test and the bit type. When a bit is loaded enough to a formation to reach to the linear region, it can transfer only 30%–40% of its energy to the rock due to friction coefficient. In the best situation, the minimum amount of MSE has a value of the order of confining compressive strength (CCS) of the rock, as it was mentioned by Pessier and Fear (1992). Therefore, in the most efficient situation, the minimum amount of MSE is equal to CCS, and just about 35% of the energy is transferred by the bit. This means in Equation (18), we can use CCS instead of MSE. Also, we can suppose the mechanical efficiency of the bit to be 0.35. The amount of CCS can be found from adjacent wells or can be calculated from the log or empirical equations. Therefore, Equation (19) can be rewritten as follow:

$$ROP = \frac{a_2 N_s.WOB.e^{-a_1} + a_3 \Delta p_m.q}{\frac{A.CCS}{0.35} - WOB.e^{-a_1}} \tag{20}$$

5.4. Solving the developed model using ROA

Maximizing ROP for reducing drilling cost is the permanent objective of researchers in the drilling industry. Many parameters affect the drilling rate, some of them are controllable such as WOB and RPM, and some of them are uncontrollable such as formation type. Changing and optimization of controllable drilling parameters can lead to drilling rate maximization. For optimizing drilling parameters, there should be an exact relation between these parameters and the drilling rate of penetration (ROP). In this section, the Rain Optimization Algorithm would be used for ROP modeling and prediction. Therefore, drilling parameters would be optimized, leading to drilling cost-effectively reduced.

Using mud logging data of a drilling rig in a specific formation, we have ROP, WOB, N_s, Δp_m, q and CCS in several points. For this work, we used 500 data series in the Asmary formation of one of the Iranian oil fields.

The cost function for the ith data set is equal to:

$$Cost_i = ROP_i - \frac{a_2 N_{si}.WOB_i.e^{-a_1} + a_3 \Delta p_{mi}.q_i}{\frac{A.CCS}{0.35} - WOB_i.e^{-a_1}} \tag{21}$$

Moreover, the cost function for the total data can be obtained as follow:

$$Cost = \sqrt{\sum_{i=1}^{50} Cost_i^2} \tag{22}$$

In this case, using ROA, we will find a_1 *to* a_3 so that the amount of cost minimized. For this work, ROA will guess the amount of the a_1 *to* a_3 first time and amount of cost in Equation (22) will be calculated. At the next iterations, this algorithm tries to change a_1 *to* a_3 to reduce the cost.

Therefore, briefly, we used Equation (19) as the main equation. Then in each data point of the data set, the real ROP was compared with the computed ROP by Equation (19) as it can be seen in Equation (20). In Equation (21), it was tried to minimize some of the errors with changing the constants a_1 *to* a_3. In the end, by obtaining these constants a special relation for predicting ROP in a certain formation was obtained as it can be seen in Equation (22).

For solving this problem using ROA, the initial population was 100, the minimum amount of each variable was zero, the maximum amount was one and After 100 iteration amount of cost was reduced to 1e-16.

After 100 iterations calculated amount of a_1 *to* a_3 using ROA were as follow:

$a_1 = 0.07;$

$a_2 = 0.58;$

$a_3 = 0.99 \cong 1;$

Fig. 12 shows the process of finding the answer. So the ROP model for this formation can be obtained as follow:

$$ROP = \frac{0.58 N_s.WOB.e^{-0.07} + \Delta p_m.q}{\frac{A.CCS}{0.35} - WOB.e^{-0.07}} \tag{23}$$

Having A = 29.5 in^2 and CCS = 2000 psi, ROP equation for this formation will be:

$$ROP = \frac{0.58 N_s.WOB.e^{-0.07} + \Delta p_m.q}{\frac{29.5*2000}{0.35} - WOB.e^{-0.07}} \tag{24}$$

This equation can be more simplified as follow:

$$ROP = \frac{0.54 N_s.WOB + \Delta p_m.q}{168571 - 0.93WOB} \tag{25}$$

This formula was tested in another drilling point (at the same formation with the same drilling conditions) with following drilling parameters for obtaining drilling rate:

WOB = 9000 lb;
A = 29.5 in^2;
N_s = 100 rpm;
Δp_m = 2000 psi;
q = 150 gpm;
CCS = 2000 psi;
ROP = 1.2 ft/h;

Moreover, the obtained ROP was close to 1.1 as it was expected.

It should be emphasized that Equation (20) was developed for directional drilling using a down hole motor. If it is interested to use this equation in vertical drilling conditions when there is no a down hole motor in the well, letting $\Delta p_m = 0$ yields:

$$ROP = \frac{a_2 N_s.WOB.e^{-a_1}}{\frac{A.CCS}{0.35} - WOB.e^{-a_1}} \tag{26}$$

Equation (26) is simpler than Equation (20) and it is just necessary to find a_1 and a_2 to solve the equation for specific drilling conditions. Equations (20) and (26) can be used for drilling optimization using the

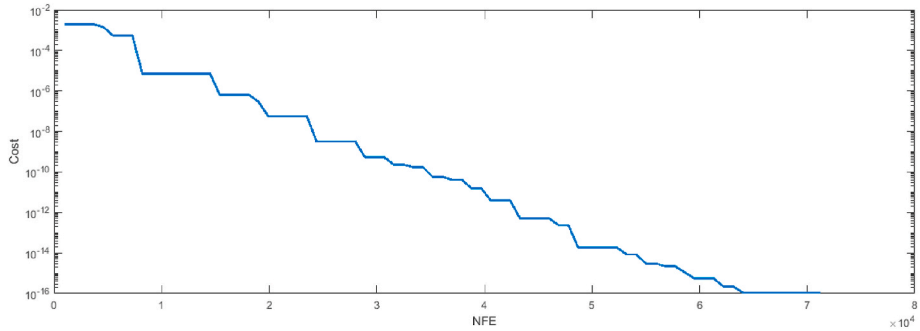

Fig. 12. The process of cost function reduction versus the number of function evaluation for solving drilling problem.

Table 6
Comparison of the speed and exactness of the ROA with BA, PSO, SWA and GA in solving the drilling optimization problem in this work.

Algorithm name	NFE	Error	Rank
ROA	64,350	1e-16	1
BA	90,100	1.7e-15	2
PSO	78,500	1.6e-12	3
SWA	145,597	1.5e-6	4
GA	63,024	2.5e-2	5

developed method by Dupriest and Koederitz (2005) more effectively. Dupriest and Koederitz (2005) stated a method for hydraulics optimization during drilling. It is recommended to optimize Hydraulics using the Dupriest method and optimize WOB and bit RPM using the proposed method in this work.

At the next attempt, we tried to solve Equation (20) using some other metaheuristic algorithms. For this purpose, GA, PSO, BA and SWA with the available parameters in Table 3 (except population number that was set to 100) were used. Table 6 compare the power and exactness of these 4 algorithms with the ROA. As it can be seen from this table, BA could find the answer with almost the same exactness of the ROA but nearly 2 times NFE. PSO, SWA and GA could get the next ranks respectively. Fig. 13 shows the process of finding the answer for these four algorithms that can be compared with Fig. 12 that shows the process of finding an answer for the ROA.

6. Conclusions

In this study, a new metaheuristic optimization algorithm called ROA that was inspired from raining phenomena was introduced and developed in detail. This algorithm was used to solve some important and standard benchmark functions as well as one drilling problem and its performance was compared with the genetic and particle swarm optimization algorithms as well as Sperm Whale and Bat algorithms. Results of this work summarized as follows:

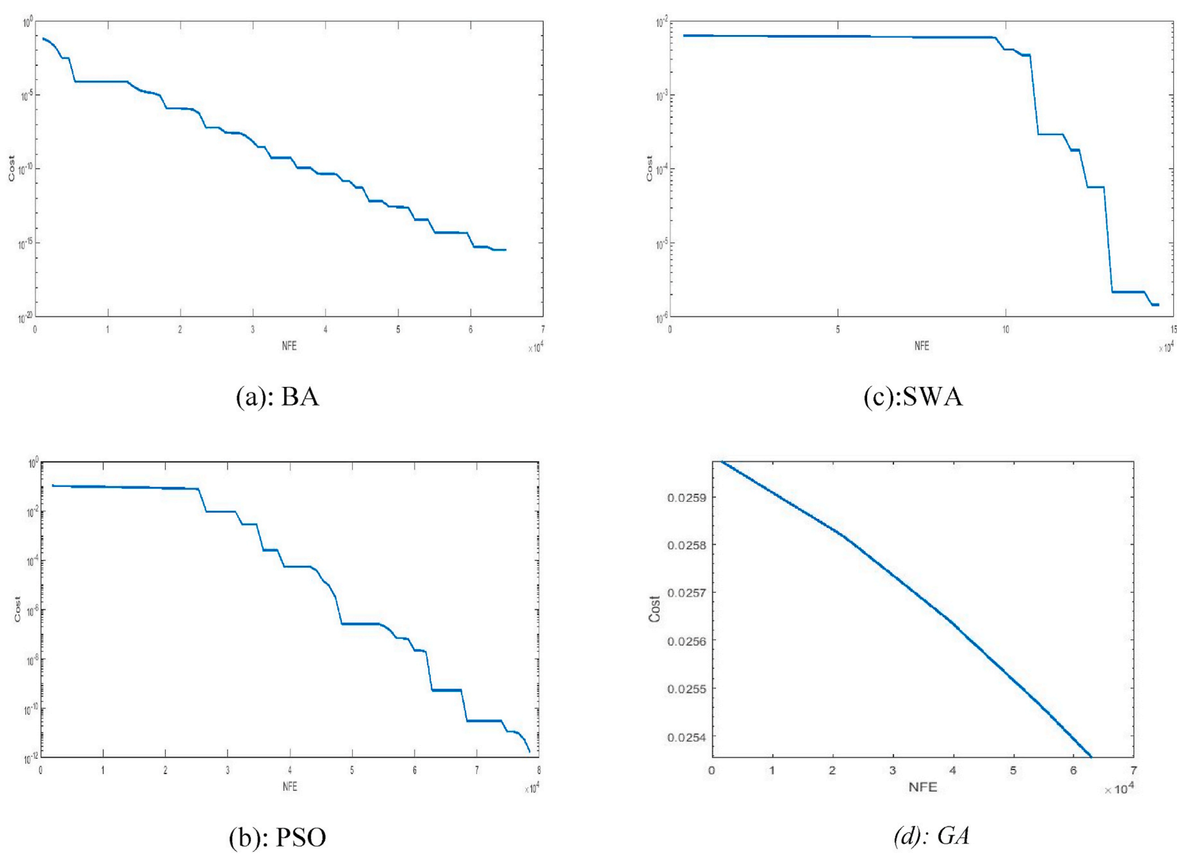

Fig. 13. The process of cost function reduction versus the number of function evaluation for solving drilling problem using (a): BA, (b): PSO, (c): SWA, (d): GA.

The developed algorithm, in addition to obtaining absolute extremums, was able to obtain local extremums with a high degree of accuracy.

ROA needed a fewer number of cost function evaluations, time and cost to solve most of the problems compared to GA and PSO that is very important in solving complicated engineering problems. Results show that ROA could get ranking 1 between other Compared optimization algorithms.

ROA was used to solve a drilling problem and was able to find the answers very quickly and accurately. Also, the proposed algorithm could reduce the number of function evaluations (NFE) to half of the BA that has the best performance between selected algorithms.

A new hybrid method for developing a new ROP model was introduced. This model was able to predict the drilling rate in directional drilling as well as vertical drilling.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ali Reza Moazzeni: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Visualization, Investigation. **Ehsan Khamehchi:** Supervision, Software, Validation, Writing - review & editing.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.petrol.2020.107512.

References

Aghay Kaboli, S.Hr, Selvaraj, J., Rahim, N.A., 2016. Rain-fall optimization algorithm: a population based algorithm for solving constrained optimization problems. Journal of Computational Science. https://doi.org/10.1016/j.jocs.2016.12.010.

Amadi, W.K., Iyalla, I., 2012. Application of mechanical specific energy techniques in reducing drilling cost in deepwater development. In: Presented at SPE Deepwater Drilling and Completions Conference, Texas, 20-21 June. https://doi.org/10.2118/156370-MS. SPE-156370-MS.

Amar, K., Ibrahim, A., 2012. Rate of penetration prediction and optimization using advances in artificial neural networks, a comparative study. In: Proceedings of the 4th International Joint Conference on Computational Intelligence, pp. 647–652.

Arabjamaloei, R., Shadizadeh, S., 2011. Modeling and optimizing rate of penetration using intelligent systems in an Iranian southern oil field (Ahwaz oil field). Petrol. Sci. Technol. 29, 1637–1648.

Barbosa, L., Nascimento, A., Mathias, M.H., Carvalho, J.A., 2019. Machine learning methods applied to drilling rate of penetration prediction and optimization - a review. J. Petrol. Sci. Eng. 183 (December), 106332.

Basturk, B., Karaboga, D., 2006. An Artificial Bee Colony (ABC) Algorithm for Numeric Function Optimization. IEEE swarm intelligence symposium.

Bataee, M., Irawan, S., Kamyab, M., 2014. Artificial neural network model for prediction of drilling rate of penetration and optimization of parameters. J. Jpn. Petrol. Inst. 57, 65–70.

Bourgoyne Jr., A.T., Young, F.S., 1974. A multiple regression approach to optimal drilling and abnormal pressure detection. SPE J. 14, 371–384.

Chen, X., Honghai, F., Guo, B., Gao, Deli, Hongshu, W., Ye, Zhi, 2014. Real-time prediction and optimization of drilling performance based on a new mechanical specific energy model. Arabian J. Sci. Eng. 39, 8221–8231. https://doi.org/10.1007/s13369-014-1376-0.

Chen, X., Gao, D., Guo, B., 2016. April 1. A method for optimizing jet-mill-bit hydraulics in horizontal drilling. SPE J. https://doi.org/10.2118/178436-PA.

Cheng, M.-Y., Lien, L.-C., 2012. Hybrid artificial intelligence-based PBA for benchmark functions and facility layout design optimization. J. Comput. Civ. Eng. 26, 612–624.

Cheng, M.-Y., Prayogo, D., 2014. Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput. Struct. 139, 198e112.

Cherif, H., Bits, S., 2012. FEA Modelled MSE/UCS Values Optimise PDC Design for Entire Hole Section. SPE, 149372.

Dorigo, M., Birattari, M., 2010. Ant colony optimization. In: Encyclopedia of Machine Learning. Springer, p. 36e39.

Dupriest, F.E., Koederitz, W., 2005. Maximizing drill rates with real-time surveillance of mechanical specific energy. In: SPE-116667-MS, Paper SPE/IADC 92194-MS Presented at the SPE/IADC Drilling Conference, Amsterdam, The Netherlands, Feb.23-25. https://doi.org/10.2118//92194-MS.

Ebrahimi, A., Khamehchi, E., 2016. Sperm whale algorithm: an effective metaheuristic algorithm for production optimization problems. J. Nat. Gas Sci. Eng. 29, 211e222.

Edalatkhah, S., Rasoul, R., Hashemi, A., 2010. Bit selection optimization using artificial intelligence systems. Petrol. Sci. Technol. 28, 1946–1956.

Eren, T., Ozbayoglu, M.E., 2010. Real time optimization of drilling parameters during drilling operations. In: SPE Oil and Gas India Conference and Exhibition. Society of Petroleum Engineers. https://doi.org/10.2118/129126-MS.

Gandomi, A.H., Alavi, A.H., 2012. Krill herd: a new bio-inspired optimization algorithm. Commun. Nonlinear Sci. Numer. Simulat. 17 (12), 4831e4845.

Gandomi, A.H., Yang, X.-S., Alavi, A.H., 2011. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. 29, 17e35.

Geem, Z.W., Kim, J.H., Loganathan, G., 2001. A new heuristic optimization algorithm: harmony search. Simulation 76 (2), 60e68.

Golberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addion wesley, p. 1989.

He, S., Wu, Q.H., Saunders, J., 2009. Group search optimizer: an optimization algorithm inspired by animal searching behavior. Evol. Comput. IEEE Trans. 13 (5), 973e990.

Hegde, C., Daigle, H., Millwater, H., Gray, K., 2017. Analysis of rate of penetration (ROP) prediction in drilling using physics-based and data-driven models. J. Petrol. Sci. Eng. 159, 295–306.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Network. 2, 359–366.

Husseinzadeh Kashan, A., 2015. A new metaheuristic for optimization: optics inspired optimization (OIO). Comput. Oper. Res. 55, 99e125.

Karaboga, D., 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization. Erciyes university, engineering faculty computer engineering department. Technical report-tr06.

Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J. Global Optim. 39 (3), 459e471.

Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (abc) algorithm. Appl. Soft Comput. 8 (1), 687e697.

Kashan, A.H., 2009. League championship algorithm: a new algorithm for numerical function optimization. Soft Computing and Pattern Recognition. In: SOCPAR'09. International Conference of. 2009. IEEE.Yang, X.-S., 2008. Nature-Inspired Metaheuristic Algorithms. Luniver Press, UK.

Kenndy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks.

Larrãnaga, P., Lozano, J.A., 2002. Estimation of Distribution Algorithms: a New Tool for Evolutionary Computation. Kluwer Academic Publishers, Boston.

Mohan, K., Faraaz, A., Samuel, R., 2015. Comprehensive hydromechanical specific energy calculation for drilling efficiency. J. Energy Resour. Technol., Contributed by the Petroleum Division of ASME for Publication in the J. Energy Resour. Technol., Jan. 137/012904-1.

Monazami, M., Hashemi, A., Shahbazian, M., 2012. Drilling rate of penetration prediction using artificial neural network: a case study of one of iranian southern oil fields. Electron. Sci. J. Oil Gas Bus. 6, 21–31.

Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. Control Systems IEEE 22 (3), 52e67.

Payette, G.S., Spivey, B.J., Wang, L., Bailey, J.R., Sanderson, D., Kong, R., Pawson, M., Eddy, A., 2017. Real-time Well-Site Based Surveillance and Optimization Platform for Drilling: Technology, Basic Workflows and Field Results. SPE/IADC Drilling Conference and Exhibition, Society of Petroleum Engineers, Hague, The Netherlands. https://doi.org/10.2118/184615-MS.

Pessier, R.C., Fear, M.J., October 4–7 1992. Quantitative common drilling problems with mechanical specific energy and bit-specific coefficient of sliding friction. In: Paper SPE 24584 Presented at the 67th Annual Technical Conference and Exhibition, Washington, DC.

Rahimzadeh, H., Mostofi, M., Hashemi, A., Salahshoor, K., 2010. Comparision of the penetration rate models using field data for one of the gas fields in Persian Gulf area. In: SPE 131253. CPS/SPE International Oil and Gas Conference and Exhibition in China Held in Beijing, China, 8-10 June.

Rosenbrock, H.H., 1960. An automatic method for finding the greatest or least value of a function. Comput. J. 3 (3), 175–184.

Salimi, H., 2014. Stochastic fractal search: a powerful metaheuristic algorithm. Knowl. Base Syst. 75, 1e18.

Simon, R., Jan. 23–24 1963. Energy balance in rock drilling. In: Paper SPE 499 Presented at the SPE-U. Texas Drilling and Mechanics Symposium, Austin, TX.

Skjerpen, T., Storrøsten, H.B., Rosendahl, K.E., Osmundsen, P., 2018. Modelling and forecasting rig rates on the Norwegian continental shelf. Resour. Energy Econ. 53, 220–239. https://doi.org/10.1016/j.reseneeco.2018.05.004.

Soares, C., Gray, K., 2019. Real-time predictive capabilities of analytical and machine learning rate of penetration (ROP) models. J. Petrol. Sci. Eng. 172, 934–959. https://doi.org/10.1016/j.petrol.2018.08.083.

Storn, R., Price, K., 1997. Differential evolution e a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11, 341e359.

Talbi, E., 2009. Metaheuristics: from Design to Implementation. John Wiley & Sons, Hoboken, New Jersey, USA.

Teale, R., 1965. The concept of specific energy in rock drilling. Int. J. Rock Mech. Min. Sci. 2, 57–73.

Warren, T.M., 1987. Penetration-rate performance of roller-cone bits. SPE Drill. Eng. 2 (1), 9–18.

Waughman, R.J., Kenner, J.V., Moore, R.A., 2003. Real-time specific energy monitoring enhances the understanding of when to pull worn PDC bits. SPE Drill. Complet. 18 (1), 59–68.

Winters, W.J., Warren, T.M., Onyia, E.C., 1987. Roller bit model with ductility and cone offset. In: SPE Paper 16696, Presented at SPE 62nd Annual Technical Conference and Exhibition Held in Dallas, Texas, September 27-30.

Yang, X.-S., 2009. Firefly algorithms for multimodal optimisation. In: 5th Symposium on Stochastic Algorithms, Foundations and Applications.

Yavari, H., Sabah, M., Khosravanian, R., Wood, D.A., 2018. Application of an adaptive neuro-fuzzy inference system and mathematical rate of penetration models to predicting drilling rate, Iran. J. Oil Gas Sci. Technol. 7, 73–100.