Hydraulic Fracturing Design Optimization and Post Fracture Performance Prediction Using Hybrid Intelligent Evolutionary Computing and Fuzzy Support Vector Machines

[1]Odedele T. O, [2]Ibrahim H. D

Abstract- **In the oil and gas industries, many production wells have been hydraulically fractured to boost productivity. Much has been written regarding fracture propagation and treatment design optimization. The models already developed and applied in the industry include two-dimensional (2D) and three dimensional(3D) fracture propagation models. But the concern in the industry is the application of these models in view of large numbers of data required. This paper summarizes the efforts conducted towards the development of a new model and methodology for optimal design of hydraulic fracture treatments in oil industry. The motivation of this methodology is its capability to provide engineers with a near optimum design of fracturing job despite very little reservoir data availability. The unique design optimization method presented here is an application of soft-computing based on intelligent system. This method will accept available data on each well, which includes basic well information and production history, and provides engineer with a detail optimum hydraulic fracture design unique to that well, along with the expected post-fracture productivity using hybrid evolutionary computing strategies and fuzzy support vector machines.**

Index Terms-**Hydraulic fracturing, Particle swarm optimization, Differential evolution, Hybrid Particle swarm optimization, Differential evolution, Productivity ratio, Treatment parameters.**

I. INTRODUCTION

The concept of Evolutionary Computing method covers the process of searching for an optimal solution inspired by natural evolution. It can also be viewed as a family of trial and error problem solvers which can be considered as global optimization methods with a meta-heuristic or stochastic optimization concept, characterized by the use of a population of candidate solutions. Such methods include Genetic Algorithm, Particle Swarm Intelligence and Differential Evolution among others.

Hydraulic fracturing is a well-stimulation technique in which pressurized fluid made of water, sand and chemicals is pumped into a wellbore resulting to the creation of fractures. In most formations, a single vertical fracture is created which propagates in two directions from the wellbore. Initially the fluid which does not contain any propping agent (called pad) is injected to create a fracture that is wide enough to accept a propping agent. The propping agent usually serves the purpose to open the fracture once the pumping operations stops.

Stimulation operations are widely used in the oil industry to enhance the productive potential of wells and hydrocarbon bearing formations. These operations act to increase productivity of a given formation by creating channels in reservoir rock or removal of the damage, thus facilitating the flow of fluids to be produced.

The design of a hydraulic fracture treatment in a particular formation involves the selection of appropriate fracturing fluids, propping agents' concentrations, the injection rates and pressures. The proposed design is expected to give specific fracture geometry and conductivity, which is related to an enhanced production obtained from the fractured well. This implies that there are a significant number of possible fracture geometries arising from several possible combinations of the design parameters involved and their nonlinear interactions and this will result in a different post-fracture well production performance.

The hydraulic treatment design model started as far back as 1955 with Howard and Fast who published the mathematical model for 2D fracture propagation. Much has been written in the literature regarding fracture propagation and treatment design optimization. These include Perkins-Kern-Nordgren(PKN) and Kristonouch-Geertsma Daneshy(KGN) models. Today, with the advent of high powerful computers, 3D fracture propagation models have been developed.[1]

Specifically on fracture design optimization, Ralph and Veatch[2] introduced the basic concepts of hydraulic fracture treatments cash flow analysis by applying the net present value as a valuable tool for obtaining the optimal design of hydraulic fracture treatment. An optimal hydraulic fracture treatment design yields maximum net present value of the cash flow after the treatment, considering cash in-flows and the treatment costs.

.Correspondences
[1]Odedele T. O is Deputy Director (Computer Services Division), Raw Materials Research & Development Council, (RMRDC), Abuja Nigeria (email: odedelermrdc@yahoo.com)
[2]Ibrahim H. D is Director General of Raw Materials Research & Development Council, Raw Materials Research & Development Council, (RMRDC), Abuja Nigeria (email: hdibrahim2002@yahoo.co.uk)

Poulsen and Soliman [3] used fluid volume and propping agent concentration as design variables in the optimization process. The 2-dimensional fracture propagation model accounting for propping agent transport and sedimentation was adopted. No formal optimization procedure was used but sensitivity analysis aimed at minimizing the difference between calculated and desired fracture length and conductivity was adopted.

Balen et al.[4] used the fracturing fluid, injected fluid volume and propping agent concentration, pumping rate, and propping agent types as design variables. In their work, a two-dimensional fracture propagation model for predicting fracture geometry and a cash flow model was used. The optimization procedure was based on a sensitivity analysis of the design variables with respect to net present value.

Hareland et al.[5,6,7] adopted fluid injection rate and fracturing fluid as design variables and a pseudo three-dimensional fracture propagation model together with a post-fracture production and cash flow models. The optimization procedure was similar to that used by Balen et al

Rueda et al. used the injected fluid volume, fracturing fluid type, propping agent type, and pumping rate as treatment design variables. The two-dimensional fracture propagation model for the prediction of fracture closure behavior, and a post-fracture production model coupled with cash flow model were used in their work. The optimization was a mixed integer linear programming (MILP) problem and solved accordingly.

Mohaghegh et al.[8] used the fluid volume injected, propping agent concentration, and fluid injection rate as design variables. In their work used a three dimensional fracturing simulator for predicting fracture propagation and closure behavior, and prop pant transport and sedimentation were adopted. The optimization procedure used was a Genetic Algorithm.

This paper presents a methodology called hybrid evolutionary computing and fuzzy support vector machines for the optimal design of hydraulic fracture stimulation treatments and prediction of post–fracture deliverability. This methodology includes the construction of an objective function, whose evaluation involves the analytical solution of mathematical model. Using evolutionary computing approaches (Particle Swarm Optimization (PSO), Differential Evolution (DE), hybrid DEPSO) promising solution domains are searched considering the information provided. The proposed optimization methodology provides a global evolutionary optimization, hence avoiding the potential problem of convergence to a local minimum in the objective function.

II. OVERVIEW OF PARTICLE SWARM OPTIMIZATION (PSO)

Particle swarm optimization (PSO) is an evolutionary computation technique, first introduced by Kennedy and Eberhart.[9,10,11] The main idea is used to model a group social behavior such as the way birds travel when trying to find sources of food, or fish schooling. The flowchart of the method is given in Fig.1.0. c1 and c2 are two positive constants, called the cognitive and social parameter respectively; r_{i1} and r_{i2} are random numbers uniformly distributed within the range [0, 1]. In each iteration, Eq. (1) is used to determine the i-th particle's new velocity, while Eq. (2) provides the new position of the i-th particle by adding its new velocity, to its current position. The performance of each particle is measured according to a fitness function, which depends on the problem. The role of the inertia weight w is considered important for the PSO's convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity. Thus, the parameter w regulates the trade-off between the global (wide-ranging) and the local (nearby) exploration abilities of the swarm. A large inertia weight facilitates exploration (searching new areas), while a small one tends to facilitate exploitation, i.e. fine-tuning the current search area. A proper value for the inertia weight w provides balance between the global and local exploration ability of the swarm, and, thus results in better solutions. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. After finding the two best values, the particle updates its velocity and positions with following equations (1) and (2).

$$v[n+1] = v[n] + c1 * rand() * (pbest[n] - X[n]) + c2 * rand()*(gbest[n]-X[n]) \quad (1)$$

$$X[n+1]=X[n]+v[n+1] \quad (2)$$

v[n] is the particle velocity, X[n] is the current particle (solution). pbest[n] and gbest[n] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors and usually c1 = c2 = 2.

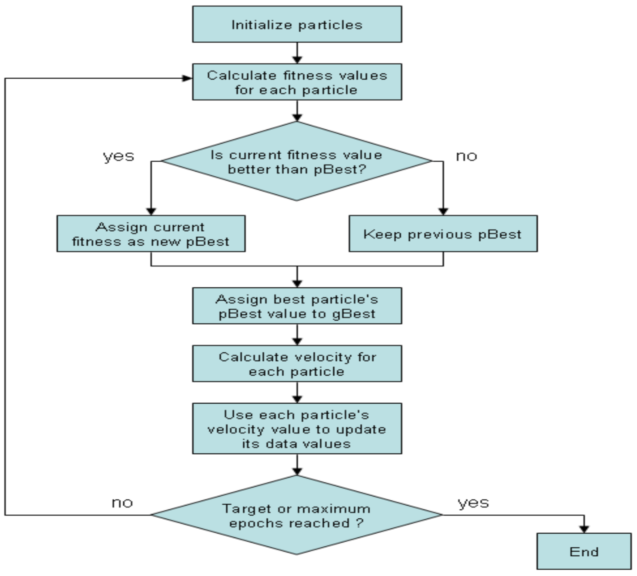
Initialize particles
Calculate fitness values for each particle
Is current fitness value better than pBest?
yes
no
Assign current fitness as new pBest
Keep previous pBest
Assign best particle's pBest value to gBest
Calculate velocity for each particle
Use each particle's velocity value to update its data values
Target or maximum epochs reached ?
no
yes
End

Fig. 1. Flow diagram illustrating the particle swarm

The procedure describing proposed PSO approach is as follows.

1. Initializing PSO with population size, inertia weight and generations.

2. Evaluating the fitness of each particle.
3. Comparing the fitness values and determines the local best and global best particle.
4. Updating the velocity and position of ea particle till value of the fitness function converges.

III. DIFFERENTIAL EVOLUTION(DE)

Differential Evolution, like other evolutionary computation methods, starts with an initial population that is generally randomly initialized. After determining the population, a new candidate individual is generated by applying mutation and crossover operators [12,13]. The mutation operator creates mutant candidate by perturbing a randomly selected candidate with the difference of two other randomly selected candidates. This candidate then becomes the input of selection operator and is examined if the candidate is better than the current member. If it is better, it will enter the next generation otherwise the current member remains in the population[20].

IV. HYBRID DIFFERENTIAL EVOLUTION WITH PARTICLE SWARM OPTIMIZATION ALGORITHM(DEPSO)

The proposed DE-PSO as mentioned earlier is a hybrid version of DE and PSO. DE-PSO starts like the usual DE algorithm up to the point where the trial vector is generated. If the trial vector satisfies the specified condition, then it is included in the population otherwise the algorithm enters the PSO phase and generates a new candidate solution. The method is repeated iteratively till the optimum value is reached. The inclusion of PSO phase creates a perturbation in the population, which in turn helps in maintaining diversity of the population and producing a good optimal solution. [14,15,16]

V. OVERVIEW OF SUPPORT VECTOR MACHINES

Vapnik proposed the support vector machines(SVMs) which was based on statistical learning theory. The governing principles of support vector machines is to map the original data x into a high dimension feature space through a non-linear mapping function and construct hyper plane in new space. The problem of regression can be represented as follows. Given a set of input-output pairs Z = {(x1, y1), (x2, y2), . . . ,(xℓ, yℓ)}, construct a regression function f that maps the input vectors x € X onto labels y € Y . The goal is to find a classifier f €F which will correctly predict new samples. There are two main cases to consider when we use a separating hyper-plane:
1. A linearly separable case
2. The data might not be linearly separable.
SVMs tackle the first problem by finding the hyper-plane that realizes the maximum margin of separation between the classes. A representation of the hyper-plane solution used to classify a new sample xi is:

$$Y=f(x)=wi\phi(x)+b \qquad (3)$$

Where wi,ϕ(x) is the dot-product of the weight vector w and the input sample, and b is a bias value. The value of each element of w can be viewed as a measure of the relative importance of each of the sample attributes for the prediction of a sample. Various research studies have shown that the optimal hyperplane can be uniquely constructed through the solution of the following constrained quadratic optimization problem.

$$\text{Minimise}1/2\|w\|+C\textstyle\sum_{i=1}^{l} \xi_I \qquad (4a)$$

subject to _ yi(||w||+ b) ≥ 1 − ξi, i= 1, . . . , ℓ

$$\xi i\geq 0, i=1,...,\ell \qquad (4b)$$

In linearly separable problem, the solution minimizes the norm of the vector w which increases the flatness(or reduces the complexity) of the resulting model and hence the generalization ability is improved. With non-linearly separable hard-margin optimization, the goal is simply to find the minimum ||w|| such that the hyperplanef(x) successfully separates all ℓ samples of the training dataset. The slack variables ξi are introduced to allow for finding a hyperplane that misclassifies some of the samples (soft-margin optimization) because many datasets are not linearly separable. The complexity constant C >0 determines the trade-off between the flatness and the amount by which misclassified samples are tolerated. A higher value of C means that more importance is attached to minimizing the slack variables than to minimizing||w||. Instead of solving this problem in its primal form of (4a) and (4b), it can be more easily solved in its dual formulation by introducing Langrangian multiplier α [17,18]:

$$\text{Maximize } W(\alpha)=\textstyle\sum_{i=1}^{l} \alpha i+½\sum_{i,j=1}^{l} \alpha i\alpha jyiyj\langle xi, xj\rangle \qquad (5a)$$

$$\text{Subject to } C\geq\alpha i\geq 0,\textstyle\sum_{i=1}^{l} \alpha iyi=0 \qquad (5b)$$

In this solution, instead of finding w and b the goal now is find the vector α and bias value b, where each αi represents the relative importance of a training sample I in the classification of a new sample. To classify a new sample, the quantity f(x) is calculated as:

$$f(x)=\textstyle\sum_{i=1}^{sv} \alpha iyiK\langle xi, xj\rangle+b \qquad (6)$$

where b is chosen so that yif(x) = 1 for any I with C > αi>0. Then, a new sample xs is classed as negative if f(xs) is less than zero and positive if f(xs) is greater than or equal to zero. Samples xi for which the corresponding αi are non-zero are called as *support vectors* since they lie closest to the separating hyperplane. Samples that are not support vectors have no influence on the decision function.

Training an SVM entails solving the quadratic programming problem of (5a) and (5b). There are many standard methods that are be applied to SVMs, these include the Newton method, conjugate gradient and primal-dual interior-point methods, but this study used the Sequential Minimal Optimization. In SVMs, kernel functions are used to map the training data into a higher dimensional feature space via some mapping φ(x) and construct a separating hyperplane with maximum margin. This yields a non-linear decision boundary in the original input space. Typical types of kernels are:

− Linear Kernel: $K(x, z) = \langle x, z\rangle$
− Polynomial Kernel: $K(x, z) = (1 + \langle x, z\rangle)^d$
− RBF Kernel: $K(x, z) = \exp(-\|x-z\|^2/2\sigma 2)$
− Sigmoid Kernel: $K(x, z) = \tanh(\gamma * \langle x, z\rangle - \theta)$

This condition ensures that the solution of (5a) and (5b) produces a global optimum. The functions that satisfy Mercer’s conditions can be as kernel functions. As promising as SVM is compared with ANN as regards generalization performance on unseen data, the major disadvantage is its black box nature. The knowledge learnt by SVM is represented as a set numerical parameters value making it difficult to understand what SVM is actually computing.

VI. FUZZY LOGIC OVERVIEW

Fuzzy Logic which was introduced by Lotfi A. Zadeh was based on fuzzy sets in 1965 [19,20,21]. The basic concept of fuzzy logic is to consider the intermediate values between [0,1] as degrees of truth in addition to the values 1 and 0. The following sections will briefly discuss the general principles of fuzzy logic, membership functions, linguistic variables, fuzzy IF-THEN rules, combining fuzzy sets and fuzzy inference systems (FISs).

Fuzzy inference systems (FISs) are otherwise known as fuzzy-rule-based systems or fuzzy controllers when used as controllers. A fuzzy inference system (FIS) is made up of five functional components. The functions of the five components are as follows:

1. A fuzzification *is an* interface which maps the crisp inputs into degrees of compatibility with linguistic variables.
2. A rule base is an interface containing a number of fuzzy if-then rules.
3. A database defines the membership functions (MFs) of the fuzzy sets used in the fuzzy rules.
4. A decision-making component which performs the inference operation on the rules.
5. A defuzzification interface which transforms the fuzzy results of the inference into a crisp output. The qualified consequents are combined to produce crisp output according to the defined methods such as: centroid of area, bisector of area, mean of maximum, smallest of maximum and largest of maximum etc. This final step is also known as defuzzification. The major disadvantage of standard fuzzy logic is the curse of dimensionality nature for high dimensional input space. For instance, if each input variable is allocated m fuzzy sets, a fuzzy system with n inputs and one output needs on the order of m^n rules.

VII. EXTRACTING FUZZY RULES FROM SUPPORT VECTOR MACHINE

In this fuzzy SVM section, we will first give an insight into how to extract fuzzy rules from Support Vector Machine (SVM), and then explain the process of optimizing the fuzzy rules and highlight an algorithm that will convert SVM into interpretable fuzzy rules. This method has both good generalization performance and ability to work in high dimensional spaces of support vector machine algorithm with high interpretability of fuzzy rules based models. Suppose a set of training dataset denotes the input space patterns. Their main concept is to construct a hyperplane that acts as a decision space such that the margin of separation between positive and negative samples is maximized. This is generally referred to as the Optimal Hyperplane". This property is achieved as the support vector machines are an approximate implementation of the method of structural risk minimization[17]. Despite the fact that a support vector machine does not provide domain-specific knowledge, it provides good generalization ability, a unique property among the different types of machine learning techniques. Instead of solving this problem in its primal form of (4a) and (4b), it can be more easily solved in its dual formulation by introducing Langrangian multiplier α: as highlighted in section II. The crucial step in fuzzy SVM is to build a reliable model on training samples which can correctly predict class label and extract fuzzy rules from SVM. On the other hand, fuzzy rule-base which consists of set of IF-THEN rules constitutes the core of the fuzzy inference. Suppose there are m fuzzy rules, it can be expressed as following forms:

Rule j: If x_1 isAj^1 AND x_2 isAj^2 AND ………x_n is. Aj^n THEN bj (7)

Where x_k is the input variables; b_j is the output variable of the fuzzy system; and A^k are linguistic terms characterized by fuzzy membership functionsa_j^k. If we choose product as the fuzzy conjunction operator, addition for fuzzy rule aggregation, and height defuzzification, then the overall fuzzy inference function is

$$F(x) = \frac{\sum_{j=1}^{m} bj \prod_{k=1}^{n} a_j^k(xk)}{\sum_{j=1}^{m} \prod_{k=1}^{n} a_j^k(xk)} \quad (8)$$

where F(x)
is the output value when the membership function achieves its maximum value. If on the other hand, the input space is not wholly covered by fuzzy rules, equation(7) may not be defined. To avoid this situation, Rule0 can be added to the rule base

Rule0: If A_0^1 AND A_0^2 AND ………. A_0^n THEN b_0

$$F(x) = \frac{b0+\sum_{j=1}^{m} bj \prod_{k=1}^{n} a_j^k(xk)}{1+\sum_{j=1}^{m} \prod_{k=1}^{n} a_j^k(xk)} \quad (9)$$

In regression analysis, F(x) shows the predicted value of each input x and since the denominator is always positive, predicted value of each input is computable by

Label(x) =$(b0 + \sum_{j=1}^{m} bj \prod_{k=1}^{n} a_j^k\ (xk)$ (10)

In order to let equation (6) and (10) are equivalent, at first we have to let the kernel functions in (6) and the membership functions in (10) are equal. The Gaussian membership functions can be chosen as the kernel functions to satisfy the Mercer condition[22,23,24]. Besides, the bias term of the expression (6) should be zero. If the Gaussian function is chosen as the kernel function and membership functions, and the number of rules equals the number of support vectors then (6) and (10) becomes equal and then output of fuzzy system (10) is equal to the output of SVM (6). A membership function $\boldsymbol{\mu}$(x) is reference function if and only if $\boldsymbol{\mu}(x)=\boldsymbol{\mu}(-x)$ and $\boldsymbol{\mu}(0)=1$. A reference function with location transformation has the following property for some locations $m_j \in R$

$$a_j^k(xk) = a^k(x_k - m_j^k)$$

A translation invariant kernel k is given by

K(x,m_j)=$\prod_{k=1}^{n} a^k(x_k - m_j^k)$

Examples of reference functions are as shown in Table I

TABLE 1
REFERENCE FUNCTIONS

	Reference functions
Symmetric Triangle	$\boldsymbol{\mu}(x)=\text{Max}(1- g\ \lvert x\rvert,0)\quad g>0$
Gaussian	$\boldsymbol{\mu}(x)=e^{-gx^2} g>0$
Cauchy	$\boldsymbol{\mu}(x)=\frac{1}{1+gx^2} g>0$
Laplace	$\boldsymbol{\mu}(x)=e^{-g\lvert x\rvert} g>0$
Hyperbolic Secant	$\boldsymbol{\mu}(x)=\frac{2}{e^{g\lvert x\rvert}+e^{-g\lvert x\rvert}} g>0$

VIII. METHODOLOGY

The system is divided into two parts: as it makes use of different evolutionary strategies for hydraulic fracturing design optimization and prediction of post fracturing well performance.

a. Hydraulic Fracturing design using evolutionary computing approach(PSO, DE, Hybrid DE-PSO)

Every individual of the population represents a potential solution of the oil/gas production problem. The evolution is guided by a strategy of selection of the individuals, with the intention of improving their "fitness", a measure based on the identified restrictions in the operational scenario determined by the fuzzy system. The optimization strategy involves construction of objective function which necessitates finding solution to hydraulic fracturing propagation model.

b. Prediction of post fracturing well performance using fuzzy support vector machines.

Various authors have used 2D, 3D/P-3D hydraulic fracturing models in their research work. In this paper, an analytical solution of 2D PKN-C fracture model was adopted. The basic solution for estimating the extent of the fracture takes into account the effects of fluid leakage into
the formation and fracture propagation is derived from Carter equation(10). On basis of material balance,[25,26,27] Carter formulated the fracture model by assuming that at any injection time t, the injection rate entering the fracture is equal to the sum of the different leak-off rates plus the growth rate of the fracture volume. Hence, we have

$$q_i=\int_0^t \frac{CL}{\sqrt{t-\tau}}\left(\frac{dA}{dt}\right)d\tau + w\frac{dA}{dt} + A\frac{dw}{dt} \qquad (10)$$

where q_i=total injection rate bbl/min
C_L=Overall leak-off coefficient
τ=Opening time at which filtration starts
W=fracture width
The analytical solution using Laplace Transforms of equation (10) is given by

$$A(t)=\frac{wqi}{4\pi C_L^2}\left[\exp(x^2)\, erfc(x) + \frac{2x}{\sqrt{\pi}} - 1\right] \qquad (11)$$

$$\chi = \frac{2\mathrm{CL}\sqrt{\pi t}}{w}$$

The problem in this paper is an optimization problem with many design variables and evaluation of objective function
The design variables include:

- Injection time, mins
- Volume of propping agent, gal
- Fracture Area, ft^2
- propping agent concentration, Ib/gal
- Propping agent required, Ib
- Surface Injection pressure, Psi
- Hydraulic horsepower, Hp
- Fluid loss coefficient, ft/$\sqrt{min}$

The objective function is
Total fracture cost= (Fluid cost + Pumping cost)
To calculate the total fracture treatment cost for a specific formation, evaluation of the fracture propagation model to determines the fluid cost and pumping costs are required. The design with lowest total cost yields optimum fracture design. Having obtained an optimal design, the post-fracture performance prediction model is derived using the fuzzy support vector machines highlighted in section(VII).

IX. APPLICATION EXAMPLE

This section gives an example of hydraulic fracturing design. Consider a well with characteristics as shown in Table II.

TABLE II

A TYPICAL HYDRAULIC FRACTURING DESIGN WELL DATA

Well Depth (ft)	5000.00
Tubing size (in)	2.0
Oil Production rate (q) bbl/day	240
Oil gravity	35
Productivity Index(J) bbl/day*Psi	15
Reservoir Pressure Psi	1900
Surface Temperature ^{0}F	72
Bottom Hole Temperature ^{0}F	200
Oil formation volume factor (B_o)	1.25
Porosity	0.135
Initial Reservoir Pressure	1900
Reservoir outer boundary rad, re ft	660
Well bore radius rw ft	0.0333
Casing diameter in	4.982
Perforation diameter, in	0.375

It is required to determine an optimum hydraulic fracturing treatment in order to obtain maximum production rate.
The optimized hydraulic fracturing design parameters are as shown below (20/40 mesh sand, Lease oil fracturing fluid):

X. RESULTS AND DISCUSSION

Figure 2 shows the prediction of productivity ratio after fracturing using FuzzySVM with Cauchy kernel and membership function. The productivity ratio is 3.6 so that if the initial production was 240bbl/day, the production after fracturing is 864bbl/day.
Figure 3 also show the performance comparison of DEPSO with DE and PSO for fitness function in Eq.(13) for 40 generations and varying particle size. As it can be seen DEPSO is preferred to DE, PSO. However, DE and DE-PSO seem to begin to converge as from the particle size greater than 60. The optimized fracturing design is as shown in Table III. The prediction performance of productivity ratio after fracturing using FuzzySVM with Cauchy and Gaussian kernel and membership functions are shown in Table IV. The training dataset are as provided in the TableV. The testing dataset given in table VI is therefore used for validation and prediction of the productivity ratio which in turn is used to predict the production rate after fracturing.

PSO/DE PARAMETERS FOR HYDRAULIC FRACTURING PROBLEM

Parameter	**Value**
C1	1.2
C2	0.8
CR	0.5

F	1.0
Inertia *w* factor	0.5
Particle size	40
Searching iterations	60

TABLE III
OPTIMIZED HYDRAULIC FRACTURING DESIGN

Design Parameters	PSO	DE	DE-PSO	Conventional Technique
Injection time, mins	203.6	155	197.0	209.4
Injection Rate, bbl/min	20.8	27.4	21.5	20.8
Volume of proppant, gal	170018.3	170018.3	170182	174537.7
Fracture Area ft^2	199854	199954	193028	198800
Proppant conc. Ib/gal	1.00	1.046	1.008	1.0125
Sand Required, Ib	177747	177747	171590	176721
Surface Inj. pressure, Psi	4804	5178.5	4841	4803
Hydraulic horsepower, Hp	2452	3470	2548	2443
Fluid loss coeff., ft/$\sqrt{min}$	0.0024285	0.0024285	0.0024285	0.0024285
Total Treatment cost $	428346.2	427937	427579	438380
Efficiency	7.5	8.7	8.5	7.1

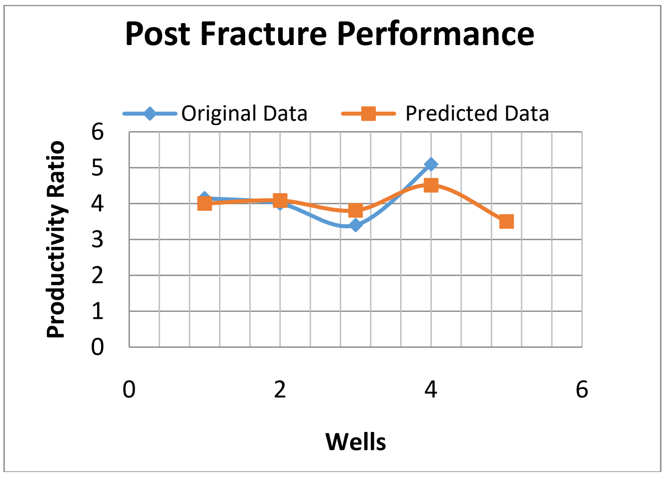

Fig 2 A plot of the experimental and predicted data versus the input data

TABLE IV
FUZZYSVM PERFORMANCE WITH DIFFERENT KERNEL FUNCTION

Method/ kernel function	Samples	Fuzzy rules	C	Gama	RMSE	Accuracy%
FuzzySVM(Gauss)	5	3	2.0	0.1	0.387	81.6
Fuzzy SVM(Cauchy)	5	3	1.9	0.45	0.366	85.0

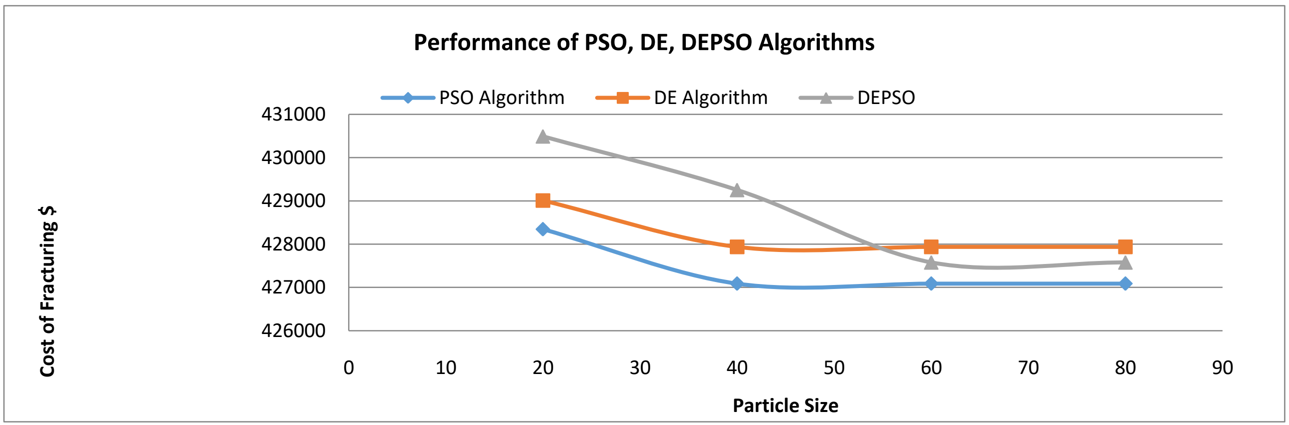

Fig.3 Performance of PSO, DE and DE-PSO Optimization Algorithm

TABLE V
TRAINING DATA

Well Name	Fracturing Fluid Coefficient	Injection Rate	Total Injection Volume	Sand Required	Fracture Area	Sand Concentration	Surface Injection Pressure	Pre Frac Production Rate	HorsePower Required	Fracturing Total Cost	Productivity Ratio
Eyak2	0.0364646	28.1993	1249.782	33545.8	37253.6	1.63	4814.60	230	3326.33	5896.39	4.2
Eyak3	0.00111	36.1	40000	17600	198900	2.23	3263	280	2886	19500	4
Eyak4	0.00144	30	30000	200200	75140	2.63	3300	240	2677	20450	3.4
Eyak5	0.0012	35.9	40000	121000	193000	2.00	1511	230	1329	22300	3.8
Eyak6	0.0011	30.9	126000	88000	100000	1.056	3058	210	2320	23755	5.1

TABLE V I

TESTING DATA

Well Name	Fracturing Fluid Coefficient	Injection Rate	Total Injection Volume	Sand Required	Fracture Area	Sand Concentration	Surface Injection Pressure	PreFrac Production Rate	HorsePower Required	Fracturing Total Cost	Productivity Ratio
Eyak2	0.0364646	28.1993	1249.782	33545.8	37253.63	1.63	4814.604	230	3326.325	5896.391	4.2
Eyak3	0.00111	36.1	40000	17600	198900	2.23	3263	280	2886	19500	4
Eyak4	0.00144	30	30000	200200	75140	2.63	3300	240	2677	20450	3.4
Eyak5	0.0012	35.9	40000	121000	193000	2.00	1511	230	1329	22300	3.8
Eyak6	0.0011	30.9	126000	88000	100000	1.056	3058	210	2320	23755	5.1
Eyak8	0.0024285	21.5	174538	171590	193028	1.046	4841	240	2548	427579	?

REFERENCES

[1] Geertsma, J., de Klerk, F., 1969. A rapid method of predicting width and extent of hydraulically induced fractures. Journal of Petroleum Technology 246, 1571–1581.

[2] Ralph, W., Veatch Jr., W. 1986. Economics of fracturing: some methods, examples and case studies. SPE Paper 15509 presented at SPE 61th Annual Technical Conference and Exhibition held in New Orleans, Atlanta, USA, pp. 1 –16.

[3] Poulsen, K.D., Soliman, M.Y., 1986. A Procedure for Optimal Hydraulic Fracturing Treatment Design. SPE Paper 15940 presented at SPE Eastern Regional Meeting held in Columbus, OH, USA, 221–230.

[4] Balen, M.R., Meng, H.Z., Economides, M.J., 1988. Application of net present value (NPV) in the optimization of hydraulic fracture SPE Paper 18541 presented at the SPE Eastern Regional Meeting in Charleston, USA, 181– 191.

[5] Hareland, G., and Rampersad, P. R. 1994. Hydraulic fracturing design optimization in low permeability gas reservoirs. SPE 27033, Latin American/Caribbean Petroleum Engineering Conference, Buenos Aires, Argentina, April 27–29.

[6] Hareland, G., Rampersad, P., Dharaphop, J., Sasnanand, S., 1993.Hydraulic fracturing design optimization. SPE Paper 26950 presented at the SPE Eastern Regional Conference and Exhibition help in Pittsburgh, PA, USA, 493– 500.

[7] Daneshy, A., 1978. Numerical solution of sand transport in hydraulic fracturing. Journal of Petroleum Technology, 132– 140. January.

[8] Mohaghegh, S., Popa, A., Ameri, S., 1999. Intelligent systems can design optimum fracturing jobs. SPE Paper 57433 presented at the SPE Eastern Regional Meeting held in Charleston, West Virginia, USA, pp. 1– 9.

[9] Jones, D., Schonlau, M., Welch, W., 1998. Efficient global optimization of expensive black-box functions. Journal of Global Optimization 13, 455– 492. MATLAB, Ver. 5.3. The MathWorks.

[10] Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proc. IEEE int'l conf. on neural networks Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995.

[11] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the sixth international symposium on micro machine and human science pp. 39-43. IEEE service center, Piscataway, NJ, Nagoya, Japan, 1995

[12] Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series). Springer-Verlag New York, Inc., (2005)

[13] Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of

[14] Global Optimization **11**(4), 341-359 (1997). doi:10.1023/A:1008202821328

[15] Hao, Z-F, Gua, G-H, Huang, H., "A Particle Swarm Optimization Algorithm with Differential Evolution", Sixth International conference on Machine Learning and Cybernetics, pp. 1031 – 1035, 2007

[16] Hendtlass, T., "A Combined Swarm differential evolution algorithm for optimization problems', fourteenth international conference on industrial and engineering applications of artificial intelligence and expert systems, Lecture notes in computer Science, Volume 2070, 11 – 18, Springer Verlag, 2001.

[17] Talibi, H., and Bautouche "Hybrid Particle Swarm with Differential Evolution for Multimodal Image Regression", IEEE International Conference on Industrial Technology, Volume 3, pp. 1567-1573, 2004.

[18] Sch¨olkopf, A. Smola, R. C. Williamson, and P. L. Bartlett "New support vector algorithms". *Neural Computation*, 12:1207–1245 2000.

[19] Cristianini N, Shawe-Taylor J: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods Cambridge, U.K.: Cambridge UniversityPress; 2000..

[20] Zadeh, L. A., 1973. Outline of a new approach to analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 28–44.

[21] Zadeh L . A, "The concept of a linguistic variable and its application to approximate reasoning," *Information Sciences*, no. 8, pp. 199–249, 301–357, 1975.

[22] E. H. Mamdani, "Applications of fuzzy algorithms for control of a simple dynamic plant," in *Proc. IEE*, no. 121, 1974, pp. 1585–1588

[23] J. L. Castro and M. Delgado, "Fuzzy systems with defuzzification are universal approximators," *IEEE Transactions on System, Man and Cybernetics*, vol. 2,

[24] .Chen Y, Wang J: Support vector learning for fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 2003, 11(6):716-728.

[25] Lin CF, Wang SD: Fuzzy support vector machine. IEEE Transactions on Neural Networks 2002, 13(2):464-471.

[26] Bouteca, M. J. 1988. Hydraulic fracturing model based on a three-dimensional closed form: Tests and analysis of fracture geometry and containment. SPE Prod. Eng. November:445–454.

[27] Economides, M. J., Hill, A. D., and Ehlig-Economides, C. 1994. Petroleum Production Systems. Upper Saddle River, New Jersey: Prentice Hall PTR.

[28] Elbel, J. L. 1985. Considerations for optimum fracture geometry design. SPE/DOE 13866, SPE/DOE Low Permeability Gas Reservoirs Symposium, Denver, CO, May 19–22.

.

Dr Ibrahim Doko Hussaini who hails from Doko Niger state of Nigeria was born on 23rd September1962. The educational background is as follows:* The University of Leeds, United Kingdom (1993)Ph.D Textile Science & Engineering), He is currently Director General of Raw Materials Research & Development Council,(RMRDC) , Abuja Nigeria

EngrTimothy.O Odedele who hails from Ipetumodu, Osun state of Nigeria was born on 29th April 1958.The educational background is as follows:* B.Sc.Petroleum Eng. (Second class Upper Division1984) University of Ibadan, Ibadan,* M.Sc Computer Science –University of Ibadan, Ibadan. He is currently Deputy Director(Computer Services Division) with Raw Materials Research & Development Council, Abuja Nigeria.