Hindawi
Geofluids
Volume 2022, Article ID 2056323, 12 pages
https://doi.org/10.1155/2022/2056323

Research Article

Parameter Optimization Study of Gas Hydrate Reservoir Development Based on a Surrogate Model Assisted Particle Swarm Algorithm

Le Zhang,[1,2] Xin Huang,[1,2] Jiayuan He,[1,2] Xueqi Cen,[1,2] and Yongge Liu[3]

[1]*State Key Laboratory of Shale Oil and Gas Enrichment Mechanisms and Effective Development, Sinopec Petroleum Exploration & Production Research Institute, Beijing 102206, China*
[2]*Southern Marine Science and Engineering Guangdong Laboratory (Guangzhou), 511458, China*
[3]*China University of Petroleum (East China), 266580, China*

Correspondence should be addressed to Le Zhang; zhangle2017.syky@sinopec.com

Received 7 November 2021; Accepted 15 December 2021; Published 18 January 2022

Academic Editor: Zhenzhen Wang

Using surrogate model to assist parameter optimization of oil and gas reservoir development can greatly reduce the call times of numerical simulator and accelerate the optimization process. However, for serial simulators or parallel simulators with low speedup ratio, the conventional method is still time-consuming. Firstly, an improved surrogate model assisted particle swarm optimization (PSO) algorithm was proposed in this paper. Then, the performance of the algorithm was analyzed using the Rastrigin function. Finally, the key operation parameters of a gas hydrate reservoir by depressurization−to−hot−water−flooding method were optimized with the new method. The results show that the new method only affects the update of the global optimal particle without interfering with the calculation process of the local optimal particles at the early stage of optimization. It realizes the rapid addition of the particle samples through the good parallel features of the PSO algorithm, and therefore, improve the precision of surrogate model in a short time. At the late stage of optimization, it is transformed into a local surrogate model to achieve rapid convergence, when the training time of the surrogate model exceeds the calculation time of the simulator. Both the optimization of Rastrigin function and operation parameters of gas hydrate development reveal that the new algorithm greatly reduces the number of iterations under the same accuracy and thus successfully accelerates the optimization process.

1. Introduction

Multiparameter optimization is a common problem in oil and gas industry. At present, the optimization methods that can be combined with simulators mainly include gradient-based algorithms, approximate gradient-based algorithms, and intelligent algorithms [1–3]. Gradient-based algorithms need to accurately obtain the gradient of the objective function, and therefore, the simulator must be open source so that the code can be modified to obtain the gradient [4]. Approximate gradient algorithms mainly include Levenberg-Marquardt (LM) algorithm, simultaneous perturbation stochastic approximation (SPSA) algorithm, etc. These kinds of algorithms usually have fast convergence speed, but it is easy to obtain local optimum for nonconvex problems [5, 6]. Intelligent algorithms include genetic algorithm, simulated annealing algorithm, particle swarm optimization (PSO) algorithm, etc. Their optimization process does not depend on the gradient of the objective function, and the differentiability of the objective function is not necessary [7, 8]. In addition, the global search ability of these algorithms is very strong, so compared with the approximate gradient algorithms, the probability of obtaining the global optimum of nonconvex problems is greatly enhanced. Therefore, intelligent optimization algorithms have been widely used in multiparameter optimization problems in the oil and gas industry in recent years [9, 10].

However, many researchers found that the conventional intelligent optimization algorithms have a slow convergence speed. A large number of iterations are required, and the numerical simulator needs to be called hundreds or even thousands of times. Therefore, the calculation time for complex models can be up to several weeks or even longer [11, 12]. In recent years, with the development of machine learning technology, the surrogate model assisted optimization algorithm points out a new direction for rapid optimization [13]. Based on the known information about an objective function, a surrogate model can be trained to obtain the potential location of optimum, which then are verified by the numerical simulator. Compared with conventional intelligent optimization algorithms, the surrogate model assisted optimization algorithm can find the potential positions and accelerate the convergence quickly [14, 15]. Using radial basis function network to train the surrogate model, Yu et al. proposed a surrogate-assisted PSO algorithm and proved the effectiveness of the new method [16]. Cai et al. also proposed a surrogate-assisted PSO algorithm which focus on the balance between the prediction ability of surrogates and global search ability of PSO, and the results show that the new method can handle high-dimensional expensive problems well [17]. Zhang et al. trained the surrogate model with random forest algorithm and predicted the creep index. The results show that the prediction accuracy of the method is significantly higher than that of empirical model [18]. Chen et al. proposed a surrogate model assisted differential evolution method and optimized the operation parameters in waterflooding production. A higher net present value and better convergence speed are achieved by the new algorithm [19].

From the above analysis, it can be seen that many studies have proposed a variety of computing processes for different surrogate models and intelligent optimization algorithms. Meanwhile, the surrogate model shows a good ability to accelerate the convergence speed. However, most of these models focus on reducing the call times of numerical simulators. For serial simulators or parallel simulators with low speedup ratio, the computing resources are often idle in the calculation process, resulting the low computing efficiency. Making full use of the good parallel characteristics of intelligent optimization algorithms and how to design workflow to make rational use of computing resources have not been fully considered. Therefore, based on the parallel characteristics of PSO algorithm, this paper proposed an improved surrogate-assisted particle swarm optimization algorithm (i-SAPSO) and verified it with Rastrigin function. Then, the operation parameters of gas hydrate reservoir developed by depressurization–to–hot–water–flooding method were optimized by the new method. Finally, the performance of different algorithms and rationality of optimization results were analyzed.

2. Surrogate Model Assisted Particle Swarm Optimization

2.1. PSO Algorithm. PSO algorithm was proposed by James Kennedy and Russell Eberhart in 1995. Inspired from the activity behavior of animal clusters, the algorithm combines the individual information of particles together to make the movement of the whole group and produces an evolution process from disorder to order in the problem-solving space [20–23]. The algorithm randomly selects several particles in the N-dimensional search space, in which the position of particle i in the t-th iteration can be expressed as $\vec{x}_i^{\,t} = (x_{i1}^t, x_{i2}^t, \cdots, x_{\mathrm{iN}}^t)$. The historical optimal position of the particle i in the t-th iteration can be recorded as $\vec{p}_i^{\,t} = (p_{i1}^t, p_{i2}^t, \cdots, p_{\mathrm{iN}}^t)$, and the optimal position of all particles, that is, the global optimal position, can be recorded as $\vec{p}_g^{\,t} = (p_{g1}^t, p_{g2}^t, \cdots, p_{gN}^t)$. According to the PSO algorithm, the particle has the trend of moving to its historical optimal position and to the global optimal position. Thus, the update formula of the particle position can be expressed as follows:

$$\begin{cases} v_{id}^{t+1} = \omega v_{id}^t + c_1 \operatorname{rand}_1(\,)\left(p_{id}^t - x_{id}^t\right) + c_2 \operatorname{rand}_2(\,)\left(p_{gd}^t - x_{id}^t\right), \\ x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \end{cases} \tag{1}$$

where ω is the inertia weight, c_1 and c_2 are the constants, and rand$_1$() and rand$_2$() are the random functions. Inertia weight ω in this paper is 1.0, and c_1 and c_2 are both defined as 2.0 [20].

2.2. Surrogate Model. From the iterative process of PSO, it can be seen that the PSO algorithm only updates the particle positions by simply recording the global optimal and historical optimal, but it does not mine the information of all the calculated particles. The introduction of surrogate model is to combine the information of all particles together, so as to obtain the potential position of the optimal value in a short time and improve the optimization speed of PSO. The methods of training surrogate model mainly include Gaussian process regression, support vector machine, radial basis function network, regression tree, artificial neural network, etc. [24] Among these methods, Gaussian process regression is a widely used method, and it has been proved that it can obtain satisfactory training performance. Therefore, this paper mainly uses Gaussian process regression to train the surrogate model [25].

Assume that the training data set is

$$D = \left\{ \left(\vec{x}_i, y_i\right) \middle| i = 1, \cdots, M \right\}, \tag{2}$$

where $\vec{x}$is the variable vector and y is the vector of fitness.

Gaussian process regression assumes that y follows the multivariate normal distribution, that is:

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_M \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1M} \\ k_{21} & k_{22} & \cdots & k_{2M} \\ \cdots & \cdots & \cdots & \cdots \\ k_{M1} & k_{M2} & \cdots & k_{MM} \end{bmatrix} \right), \tag{3}$$

where k is the covariance of the variable vector, and the matrix composed of the covariance of each vector can be represented by K.

When there is a new variable vector denoted by $x *$, the Gaussian process regression assumes that it still satisfies the multivariate normal distribution, the following equation can be obtained based on Equation (3):

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(x, x) & K(x, x^*) \\ K(x^*, x) & K(x^*, x^*) \end{bmatrix}\right). \tag{4}$$

The corresponding predicted value $y *$can be obtained from the properties of Gaussian distribution, which can be expressed as

$$y^* = K(x^*, x)K(x, x)^{-1}y. \tag{5}$$

From the above analysis, it can be seen that the key point affecting the regression performance of Gaussian process is the kernel function generating covariance matrix. In this paper, rational quadratic kernel is used for model training.

2.3. The Improvement of Yu's Method. For optimization problems in oil and gas industry, the fitness calculation of an objective function often takes a long time because of the calls of numerical simulators. In order to reduce the calculation time, many researchers improved the PSO algorithm by combining it with a surrogate model. Among them, the model proposed by Yu et al. is a quite typical model, and therefore, Yu's method was selected as the comparative model in this study. For the problem of finding the minimum value of the objective function, Yu's method mainly includes the following steps:

(1) Latin hypercube sampling is used to obtain M samples in the search space [26]. The objective function is called to calculate the fitness of samples, and the samples and corresponding fitness form the initial fitness sample database

(2) The fitness of the samples is arranged in ascending order, and the first N samples are selected to form the initial particle swarm. Then, the historical optimal and global optimal of the initial N particles are obtained according to the principle of PSO algorithm

(3) The first P samples are selected to train the surrogate model by a machine learning method

(4) The optimal value and optimal location of the surrogate model are obtained by PSO algorithm. After calculating the fitness at the optimal location by using the objective function, the optimal location of the surrogate model and its fitness are added into the sample database. Then, the particles are reordered in the sample database according to the fitness

(5) If the first P samples of the sample database have been changed, the surrogate model is retrained by the machine learning method

(6) Update the global best, and then update the particle swarm according to Equation (1)

(7) The finesses of the updated particles are estimated by the surrogate model. If the estimated fitness of a particle is smaller than the current historical optimal, the particle is selected as the potential particle

(8) Call the objective function to calculate the fitness of the potential particles, and add the potential particles and their fitness into the sample database

(9) Update the global best and the historical best position of each particle

(10) Judge whether the convergence condition is met. If not, return to step 3

It can be seen from the above steps that Yu's method greatly reduces the evaluation times of fitness by using surrogate model. However, the significant reduction of the evaluation times of fitness results in the slow growth of the sample number in the database. Therefore, the difference between the surrogate models trained in step 3 and step 5 may be small, and thus, the optimization convergence speed is slow at the late stage of optimization. In addition, Yu's method works well for parallel simulators with high speedup ratios, but it is prone to idle computing resources for serial or low speedup ratio simulators. For example, the number of potential particles screened in step 7 is far less than the total number of particles. Therefore, the number of cores called in the fitness calculation in step 8 is usually far less than the total number of cores of the computer.

Considering the good parallel characteristics of PSO algorithm, an improved surrogate-assisted particle swarm optimization (i-SAPSO) method which is based on Yu's method was proposed to make full use of computing resources and reduce the total number of iterations. The steps are as follows:

(1) Latin hypercube sampling is used to obtain the initial samples in the search space, and the fitness of each sample is calculated according to the objection function. Due to the independence between samples, parallel computing (MPI, CUDA, etc.) can achieve to make full use of computing resources. Considering that the number of samples to be calculated in the subsequent iteration process of i-SAPSO is much higher than that of Yu's method, the initial number of samples in i-SAPSO algorithm can be much less than that of Yu's method. The samples and their fitness form the initial sample database

(2) The fitness of the samples is arranged in ascending order, and the first N samples are selected to form the initial particle swarm

(3) Update the global best and the historical best of each particle according to the principle of PSO algorithm. Generate the new particle swarm according to Equation (1), and then obtain the fitness of each particle according to the objective function. Similarly, due

to the independence between particles, the fitness of each particle can be calculated in parallel. The new particle swarm and fitness are added to the database

(4) While performing step 3, appropriate computing resources are allocated to train the database to obtain the surrogate model. Then, the optimal value and location of the surrogate model are obtained by PSO algorithm

(5) Monitor the consumed time of step 4. In the early stage of optimization, the number of samples in the database is small, and the step 4 only takes a short time. Therefore, after step 4 is completed, use the computing resources occupied in step 4 to call the objective function to calculate the fitness of the optimal location screened by the surrogate model, and put the optimal location and its fitness into the sample database. However, when the number of samples is large in the late stage of optimization, the time spent in step 4 may be close to that of step 3. At this time, the fitness calculation of the optimal position of the surrogate model can be postponed to the next iteration; that is, the surrogate model is trained and optimized in the current iteration, and the fitness of potential particles is calculated in the next iteration. Moreover, when the number of samples is too large, the time spent in step 4 may exceed that of step 3. Then, the computing resources occupied by step 3 may be idle after the calculation is completed. At this time, the particles in the database shall be sorted, and the τ particles with the highest fitness are selected. At the same time, the optimization range is determined according to the following equation:

$$\begin{cases} lb^i = \min\left(x_1^i, x_2^i \cdots, x_\tau^i\right), \\ ub^i = \max\left(x_1^i, x_2^i \cdots, x_\tau^i\right). \end{cases} \quad (6)$$

(6) Judge whether the convergence condition is met. If not, return to step 3

It can be seen from the above steps that the main difference between the i-SAPSO method and Yu's method is that the surrogate model is no longer used to screen the potential historical optimal position. The good parallel feature of particle fitness calculation is used to quickly supplement the sample database, so as to realize the rapid accuracy improvement of the global surrogate model and acceleration of convergence. In addition, when the number of calculated particles is too large, the particles with high fitness are selected, and the local surrogate model is constructed. Due to the consideration of the invocation of computing resources in each step, the improved algorithm can almost achieve high availability of computing resources in the entire optimization process.

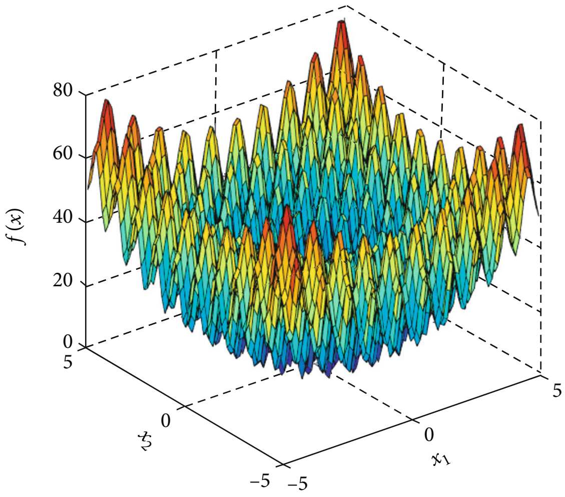


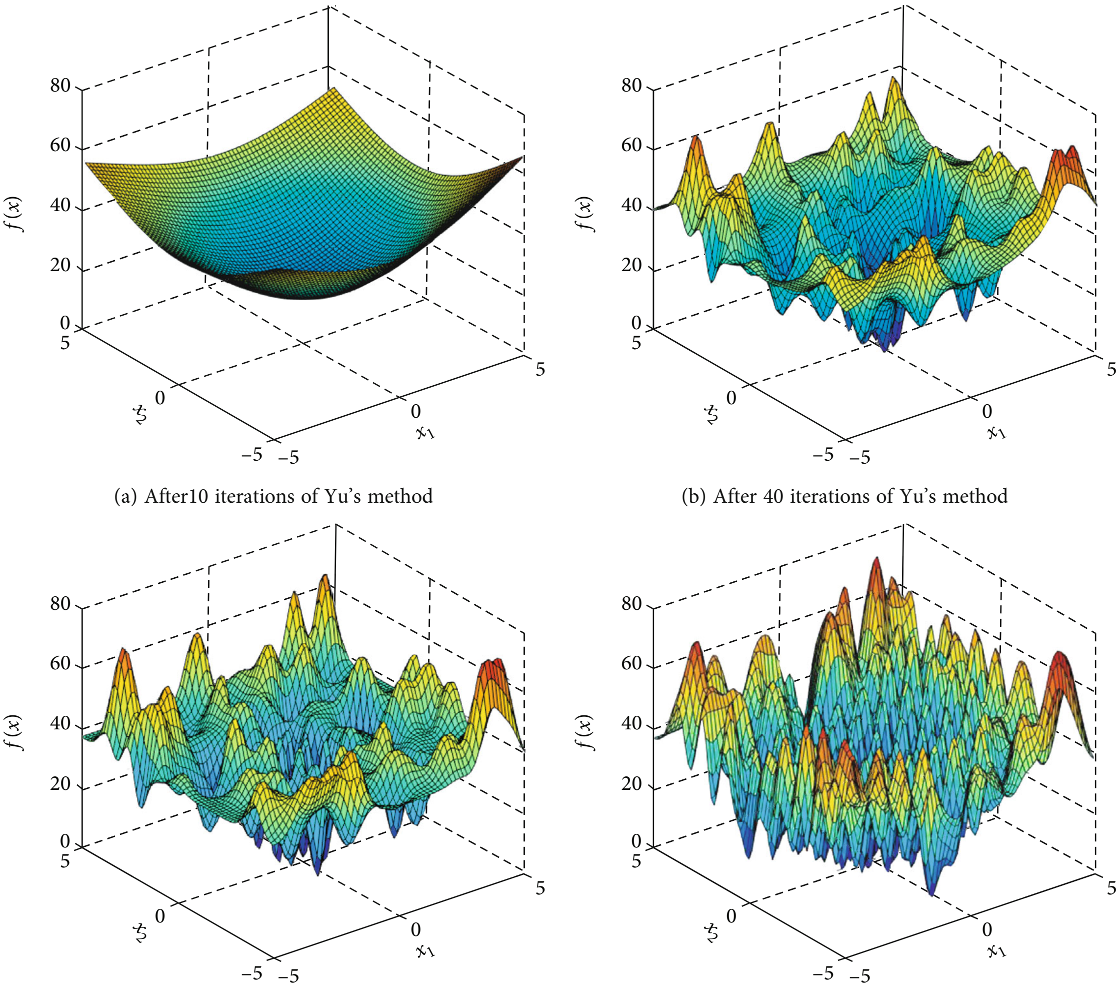
Figure 1: 2D Rastrigin function.

3. Performance Analysis and Comparison

Rastrigin function is a widely used function for testing optimization algorithms, and its expression is

$$f(x) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos\left(2\pi x_i\right)\right]. \quad (7)$$

The model is a nonconvex function, and the global minimum value 0 is obtained when x_i are all 0. It can be seen from Equation (7) that changing the n value can construct a Rastrigin function of any dimension. In order to more intuitively compare the differences between the Yu's method and i-SAPSO method in terms of surrogate model training, a 2-dimensional Rastrigin function was first used for testing and analysis. Figure 1 shows the values of the 2-dimensional Rastrigin function on the interval [-5 5], from which it can be seen that the function has a multipeaked distribution, and there are many extreme value points, which can effectively test the optimum-seeking ability of global optimization algorithms. The optimization processes of the PSO algorithm, Yu's method, and i-SAPSO method are compared, and the initial Latin hypercube sampling points of the three methods are the same, and the number is 100. The maximum number of iteration is 300, and the iteration is stopped when the objective function value is lower than 1×10^{-10}. Figure 2 shows the comparison of the surrogate model evolution during the iterations of the Yu's method and i-SAPSO method. It can be seen from the figure that the Yu's method greatly reduces the calls of the objective function, and thus, its surrogate model is updated slowly. The surrogate model of Yu's method greatly differs from the Rastrigin function at the 10th iteration, and only after 40 iterations does the surrogate model show more local features of the Rastrigin function. Meanwhile, the surrogate model updates slowly in the subsequent iterations. The surrogate model in the i-SAPSO method can already characterize the local features of the Rastrigin function well after 10



(a) After10 iterations of Yu's method

(b) After 40 iterations of Yu's method

(c) After 10 iterations of i-SAPSO method

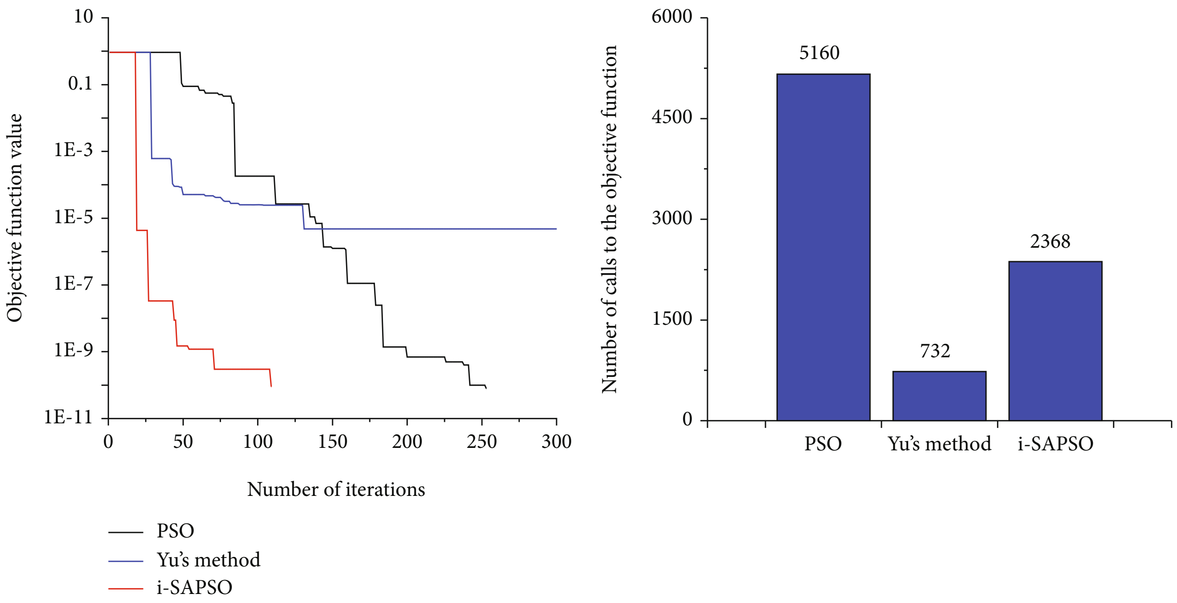
(d) After 40 iterations of i-SAPSO method

Figure 2: Evolution of surrogate model.

iterations, and the local features of the function are more accurately characterized by the surrogate model after 40 iterations.

Figure 3(a) shows the comparison of the change of the objective function value with the number of iterations. From the results, it can be seen that since both the Yu's method and i-SAPSO method use the surrogate model to predict the potential particle positions; the decrease of the objective function is obviously faster than that of PSO at the early stage of optimization. Meanwhile, because the number of particles in the sample database of the i-SAPSO increases rapidly and the surrogate model evolution quickly in the early stage, the probability of obtaining potential particles is much greater than that of Yu's method. Correspondingly, the objective function of i-SAPSO method decreases more rapidly than that of Yu's method. The PSO and i-SAPSO methods reach the preset accuracy after 253 and 109 iterations, respectively, but the Yu's method cannot further reduce the objective function value after 131 iterations. Finally, Yu's method exits the calculation when it reaches the maximum number of iterations of 300. From Figure 3(b), it can be seen that the PSO algorithm has the highest number of objective function calls, while the Yu's model has the lowest number of calls, and the i-SAPSO is in between. From the comparison results, it can be seen that the Yu's method is more suitable for cases where the requirements for optimization results are not critical and the computational process of the objective function is highly parallel. On the contrary, if a higher fitness is expected and the computational process of the objective function is not parallel, i-SAPSO can reasonably organize the computation steps to achieve fast parallel computation.

The number of unknowns can often reach tens or more when optimizing actual oil and gas field development parameters. For this reason, the dimension of the Rastrigin model is changed, and the comparison of the algorithms is carried out. The number of iterations is set to 300, and Table 1 shows the optimized objective function values for the three algorithms. It can be seen from the table that the performance of each algorithm in multidimensional case is basically the same as that in two-dimensional case. That is, under the same number of iterations, the performance of i-



(a) Objective function value

(b) Number of calls to the objective function

Figure 3: Performance comparison of different methods.

Table 1: Optimized objective function values of different methods.

	PSO	Yu's method	i-SAPSO
10-dimension	1.98	2.83	0.21
20-dimension	19.84	12.65	1.78
30-dimension	78.63	110.90	4.73

SAPSO is the best, followed by the PSO algorithm, and the call times of the objective function of Yu's method is the least, so the fitness is the worst under the same number of iterations. In addition, the nonlinearity of the objective function increases with the increase of dimension, and therefore, the probability of falling into the local optimum increases for all three algorithms as the number of dimensions increases.

4. Model Application

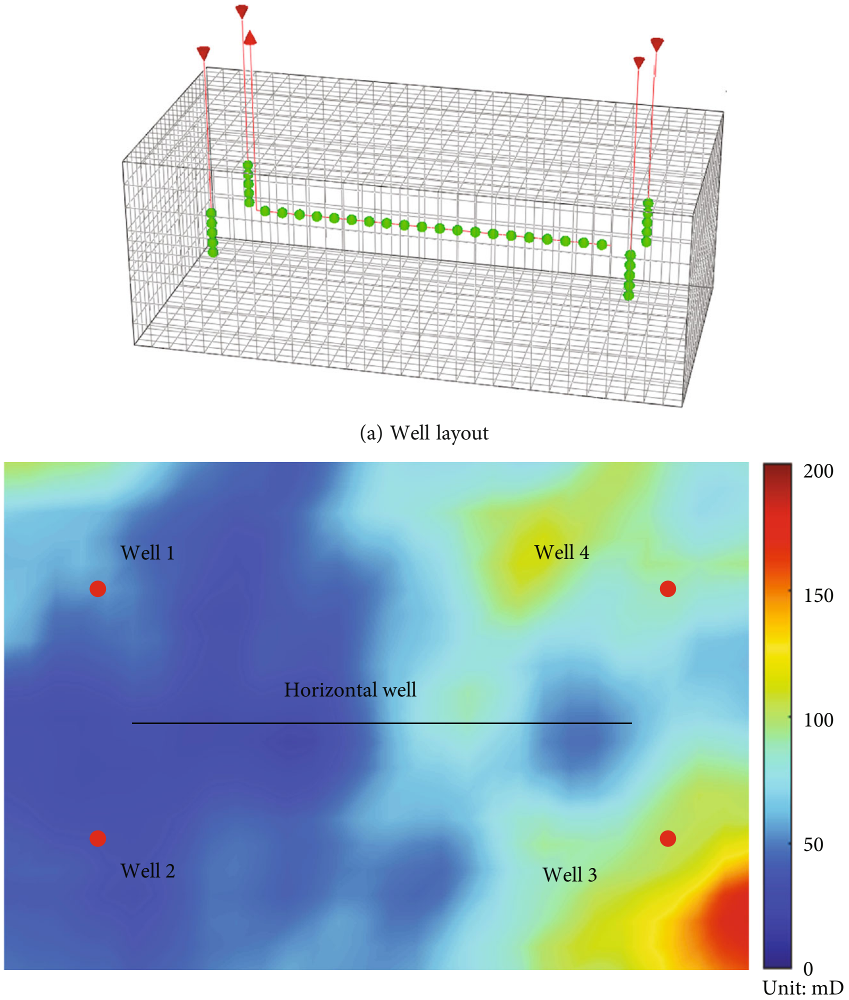
China has implemented two trial production tests of hydrate bearing layers in the Shenhu area, and the average daily gas production rate of the second trial production test is 2.87 × 104 m^3, which is quite lower than the minimum gas production rate required for commercial development [27–29]. Moridis et al. investigated the depressurization performance of the hydrate bearing layers in Mallik area and Alaska North Slope. The results show that the dissociation area is mainly located around the well, and the gas production is low only through depressurization method [29, 30]. Zhang et al. investigated the decomposition conditions of methane hydrate and the effect of hydrate saturation on the methane hydrate dissociation, and the results are similar to those of Moridis [32–34]. In order to enhance gas production, more and more attention has been paid to the combined method of depressurization and thermal stimulation [35, 36]. In this paper, the Tough+Hydrate software was used to establish a numerical simulation model for a Class III hydrate reservoir at station SH7 in the Shenhu area of the South China Sea, and the basic geological parameters of the model are shown in Table 2 [37].

The model is a five-point well pattern composed of four vertical wells and one horizontal well (Figure 4(a)). The grid system is 30 × 21 × 11, and the grid size is 15 m × 10 m × 2 m . The length of the horizontal well in the center is 300 m, and the distance between the vertical well and the nearest perforation of the horizontal well is 75 m. In order to represent the heterogeneity of the hydrate reservoir, a nonuniform permeability distribution is generated by sequential Gaussian simulation method which is a widely used in geostatistics, as shown in Figure 4(b) [38]. The average permeability is 75 md, and the coefficient of variation is 0.4 [39]. Neumann boundary condition is used, and in order to avoid the usage of well fraction, each vertical well has a distance from the boundary. Four vertical wells and one horizontal well first produce at a constant pressure of 4 MPa. The gas production rate continually decreases due to the pressure drawdown and temperature decline. When the gas production rate reaches a critical value (critical gas production rate), the four vertical wells are converted to hot water injection well to enhance gas production. In order to get a better performance, the injection rate of each well is adjusted once during the heat injection process, and thus, the total number of optimized parameters is 11. The range for each parameter is shown in Table 3, where the sum of the injection rates of the four injection wells was always maintained at 300 m^3/d.

The development of gas hydrate reservoirs needs to consider both productivity and economy. Therefore, the evaluation index is divided into production index and economic index. The production index is mainly the methane recovery,

Table 2: Basic parameters of the hydrate reservoir model.

Parameter	Value	Parameter	Value
Seabed depth	1108 m	Reservoir thickness	22 m
Initial hydrate saturation	0.44	Initial water saturation	0.56
Average permeability	75 mD	Porosity	0.41
Initial pressure	13.83 MPa	Initial temperature	11.7°C



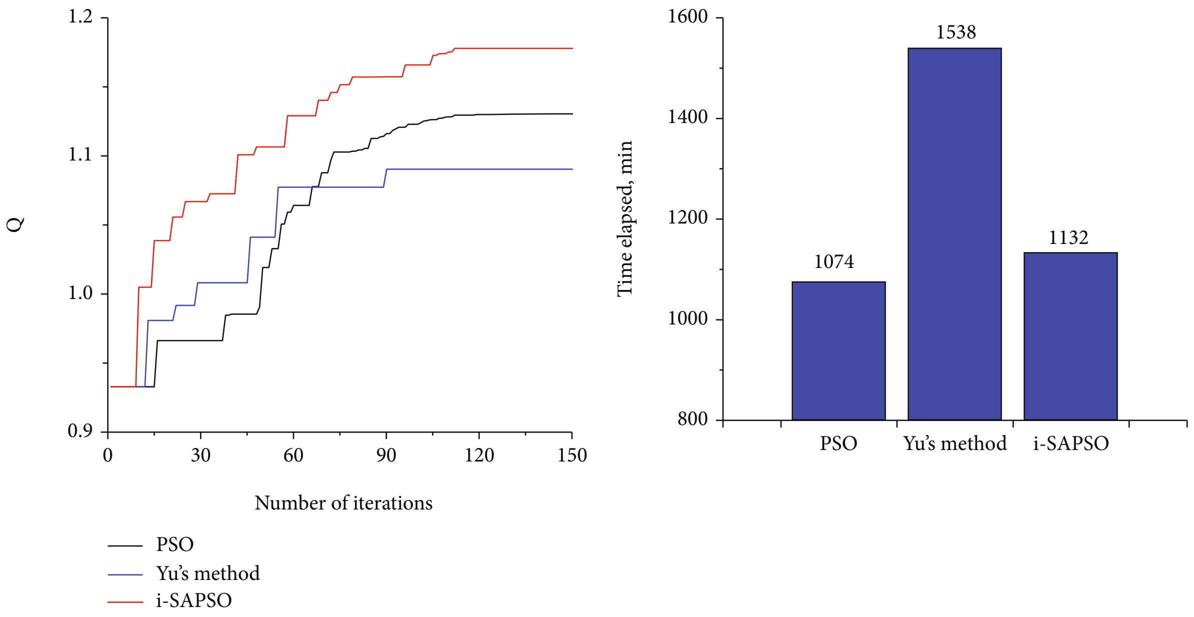
(a) Well layout

(b) Permeability distribution

Figure 4: Physical model of numerical simulation and permeability distribution.

Table 3: Ranges of optimized parameters.

Parameter	Value	Parameter	Value
Critical gas production rate	4000~20000 m^3/d	Temperature of injected water	20~90°C
Water injection rate of each well	0 ~ 200 m^3/d	Gas production rate when water injection rates are adjusted	4000~20000 m^3/d
Water injection rate of each well after adjustment	0 ~ 200 m^3/d	—	—



(a) Objective function value

(b) Time elapsed

Figure 5: Performance comparison of different methods.

Table 4: Optimal parameter values of i-SAPSO method.

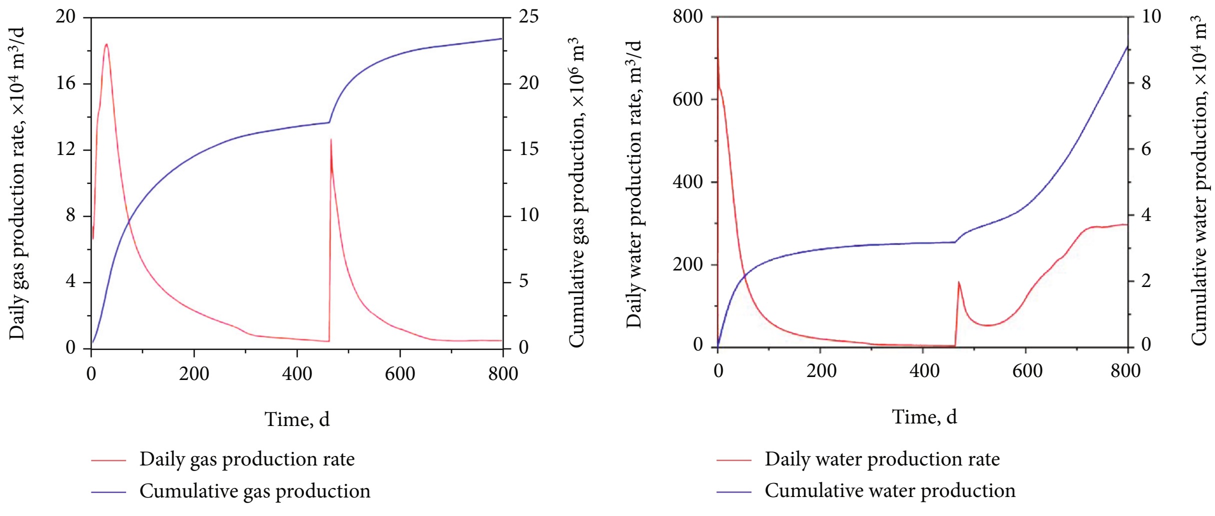
Parameter	Value	Parameter	Value
Critical gas production rate	4254 m^3/d	Temperature of injected water	33°C
Water injection rate of well 1	58 m^3/d	Water injection rate of well 2	51 m^3/d
Water injection rate of well 3	104 m^3/d	Water injection rate of well 4	87 m^3/d
Gas production rate when water injection rates are adjusted	22518 m^3/d	Water injection rate of well 1 after adjustment	73 m^3/d
Water injection rate of well 2 after adjustment	64 m^3/d	Water injection rate of well 3 after adjustment	89 m^3/d
Water injection rate of well 4 after adjustment	74 m^3/d	—	—

and the economic index is mainly the ratio of produced energy to injected energy, that is, energy efficiency ratio. Thus, the objective function of this paper is

$$\begin{cases} Q = E_R + \alpha\eta, \\ E_R = \dfrac{N_D}{N_T}, \\ \eta = \dfrac{E_{\text{out}}}{E_{\text{in}}}, \end{cases} \tag{8}$$

where Q is the objective function; E_R is the methane recovery; η is energy efficiency ratio; N_D is the cumulative volume of methane produced, m^3; N_T is the total volume of methane that can be generated from hydrate dissociation, m^3; E_{out} is the total heat of produced methane, J; E_{in} is the total heat of injected hot water, J; and α is the weight coefficient of E_R and η. According to the research results of Liu et al., the value α of is 0.025 [40].

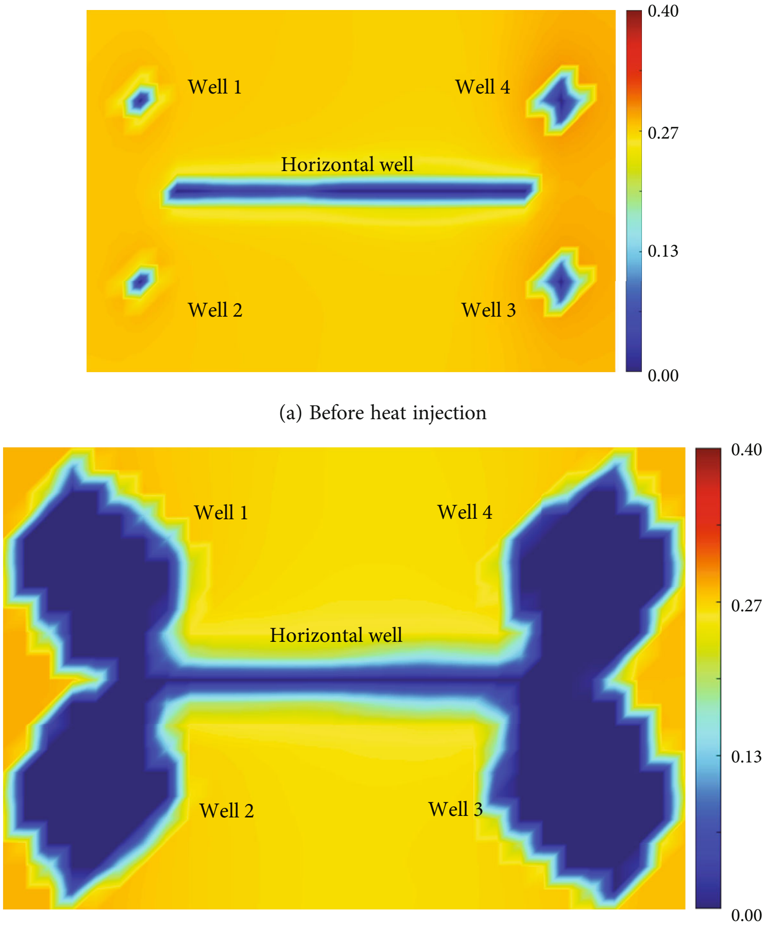
Three algorithms are compared to optimize the key parameters. The simulated time of depressurization stage is 800 days, and the maximum number of iterations is 150. The test platform is XeonSP, which contains a 40-core processor, and the memory is 64G. From the perspective of making full use of computing resources, the number of particles is set to 35. In parallel computing of i-SAPSO method, each particle occupies one CPU core based on MPI, and thus, 35 cores are occupied by the particles. Three CPU cores are used for surrogate model training, and the remaining two CPU cores are used to process system applications. Figure 5 shows the performance comparison of the three algorithms. It can be seen from the figure that the change law of the objective function is consistent with that when Rastrigin function is used as the objective function. The Yu's method and i-SAPSO method can quickly find the potential position through the surrogate model. Therefore, the fitness quickly increases in the early stage of iteration, but the fitness increase of Yu's method becomes slower in the later stage which means that Yu's method falls into local optimum. Comparing the fitness at the end of the iteration,



(a) Gas production

(b) Water production

FIGURE 6: Gas and water production of the optimal case.



(a) Before heat injection

(b) After heat injection

FIGURE 7: Hydrate saturation distribution.

it can be concluded that although it is difficult to prove that the optimization result of i-SAPSO method is globally optimal, the performance of i-SAPSO method is better than that of the PSO method and Yu's method under the same number of iterations. Figure 5(b) shows the comparison of simulation time, which indicates that the elapsed time of the PSO method is close to that of i-SAPSO method, while that of Yu's method is the longest. This is mainly due to the good independence between particles of PSO algorithm, which can easily realize parallel computing. Therefore, the elapsed time for 100 iterations is theoretically close to the time required to run the simulator 100 times continuously. Although the surrogate model training is added to the i-SAPSO method, additional computing resources are allocated. At the same time, the parallel computing framework between surrogate model training and particle fitness calculation is considered in the workflow, so the computing time is close to that of PSO method. Although Yu's method greatly reduces the total number of calls to the simulator, the computation process includes one simulator call in both step 4 and step 8, and the two calls cannot be computed parallelly. Therefore, the total calculation time is much higher than the other two algorithms.

Table 4 shows the values of the parameters obtained after the optimization of the i-SAPSO method. Combined with Table 3, it can be seen that the critical gas production rate is 4254 m^3/d, which is only slightly higher than the lower limit of the allowable range (4000~20000 m^3/d) which indicates that the time of depressurization should be long enough. This is because on one hand, the heat contained in rock and fluid can be fully used to promote the dissociation of hydrate if the depressurization period is long, and on the other hand, if hot water is injected too early, the hot water injected is easy to enter the production well directly along the dissociation area, which will lead to a waste of heat energy and reduction of energy efficiency. The optimized injected water temperature is 33°C, which is a relatively low injection temperature. Consistent with the conclusions of other researchers, a relatively low temperature of injected water is conducive to improve energy efficiency. The water injection rates of wells 1 and 2 located in the low permeability area is significantly lower than those of wells 3 and 4 located in the high permeability area, which helps to prevent the excessive pressure rise near the wellbore and the secondary formation of hydrate. Water injection rates of wells 3 and 4 after adjustment are lower than those before the adjustment, which can effectively avoid the rapid flow of injected hot water from the high permeability area to the bottom of the production well. Thus, the energy efficiency ratio can be enhanced.

Figures 6 and 7 show the gas and water production curves and hydrate saturation distributions of the optimal case, respectively. It can be seen that the gas production rate increases significantly after hot water injection. During depressurization development stage, the hydrate dissociation areas around wells 3 and 4 which are located in the high permeability area are significantly larger than those of wells 1 and 2, which indicates that higher permeability can achieve faster reservoir depressurization and hydrate dissociation. There is little difference in the shapes of the dissociation areas formed by the four wells after hot water injection, which means that balanced exploitation of hydrate reservoir can be realized through injection rate adjustment.

In conclusion, i-SAPSO method can effectively accelerate the optimization process, and the optimization results are satisfactory. Therefore, it can be used for parameter optimization of oil and gas reservoir development.

5. Conclusions

Based on the parallel computing framework, this paper proposed an improved surrogate model assisted particle swarm optimization method. Then, the effectiveness of the method was verified by Rastrigin function. Finally, the key parameters of depressurization–to–hot–water–flooding development of natural gas hydrate reservoir are optimized. The main conclusions are as follows:

(1) In a single iteration, the fitness calculation of each particle of PSO algorithm is independent. Therefore, parallel computing can be used to realize the rapid improvement of accuracy of global surrogate model. When the particle and fitness database is large, local surrogate model helps to achieve a quick convergence. The new method fully considers the parallel features of the calculation process and thus can get a better performance

(2) Yu's method greatly reduces the number of calls to the objective function. However, the accuracy of the surrogate model increases slowly with the iteration. Due to the independence of particles in an iteration, the purpose of the new model is to reduce the number of iterations rather than the number of calls to the objective function. The accuracy of the surrogate model increases significantly with the increase of iterations, and therefore, the iteration is expected to quickly converge for the new method

(3) The key parameters of a gas hydrate reservoir by depressurization–to–hot–water–flooding method are optimized. The results show that the fitness of the optimal case of the new method is significantly higher than those of Yu's method and PSO method under the same number of iterations. Meanwhile, the optimization results are satisfactory and consistent with the conclusions of other researchers. Therefore, the effectiveness of the new method is verified

Data Availability

The figure data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Sinopec Excellent Youth Innovation Fund Project (Grant No. P20025), the China Petroleum & Chemical Corporation for financial support (Grant No. P20040-4), and the Key Special Project for Introduced Talents Team of Southern Marine Science and Engineering Guangdong Laboratory (Guangzhou) (GML2019ZD0102).

References

[1] V. Machairas, A. Tsangrassoulis, and K. Axarli, "Algorithms for optimization of building design: a review," *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 101–112, 2014.

[2] J. Hou, K. Zhou, X. Zhang, X. Kang, and H. Xie, "A review of closed-loop reservoir management," *Petroleum Science*, vol. 84, pp. 363–370, 2016.

[3] B. Pouladi, A. Karkevandi-Talkhooncheh, M. Sharifi, S. Gerami, A. Nourmohammad, and A. Vahidi, "Enhancement of SPSA algorithm performance using reservoir quality maps: application to coupled well placement and control optimization problems," *Journal of Petroleum Science and Engineering*, vol. 189, 2020.

[4] L. Zhang, K. Zhang, Y. Chen et al., "Smart well pattern optimization using gradient algorithm," *Journal of Energy Resources Technology*, vol. 138, no. 1, pp. 1–13, 2016.

[5] T. Foroud, A. Baradaran, and A. Seifi, "A comparative evaluation of global search algorithms in black box optimization of oil production: a case study on Brugge field," *Journal of Petroleum Science and Engineering*, vol. 167, pp. 131–151, 2018.

[6] K. Khan and A. Sahai, "A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context," *International Journal of Intelligent Systems and Applications*, vol. 4, no. 7, pp. 23–29, 2012.

[7] C. Kim, R. Batra, L. Chen, H. Tran, and R. Ramprasad, "Polymer design using genetic algorithm and machine learning," *Computational Materials Science*, vol. 186, 2021.

[8] W. Zhang, A. Maleki, M. Rosen, and J. Liu, "Optimization with a simulated annealing algorithm of a hybrid system for renewable energy including battery and hydrogen storage," *Energy*, vol. 163, pp. 191–207, 2018.

[9] Y. Liu, J. Hou, Z. Chen et al., "Enhancing hot water flooding in hydrate bearing layers through a novel staged production method," *Energy*, vol. 217, 2021.

[10] K. Mohammadi and F. Ameli, "Toward mechanistic understanding of Fast SAGD process in naturally fractured heavy oil reservoirs: application of response surface methodology and genetic algorithm," *Fuel*, vol. 253, no. 1, pp. 840–856, 2019.

[11] Z. Ma and J. Y. Leung, "Design of warm solvent injection processes for heterogeneous heavy oil reservoirs: a hybrid workflow of multi-objective optimization and proxy models," *Journal of Petroleum Science and Engineering*, vol. 191, 2020.

[12] H. Rahmanifard and T. Plaksina, "Application of fast analytical approach and AI optimization techniques to hydraulic fracture stage placement in shale gas reservoirs," *Journal of Natural Gas Science and Engineering*, vol. 52, pp. 367–378, 2018.

[13] A. Bhosekar and M. Lerapetritou, "Advances in surrogate based modeling, feasibility analysis, and optimization: a review," *Computers & Chemical Engineering*, vol. 108, pp. 250–267, 2018.

[14] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, vol. 45, no. 1-3, pp. 50–79, 2009.

[15] A. Golzari, M. H. Sefat, and S. Jamshidi, "Development of an adaptive surrogate model for production optimization," *Journal of Petroleum Science and Engineering*, vol. 133, pp. 677–688, 2015.

[16] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Information Sciences*, vol. 454-455, pp. 59–72, 2018.

[17] X. Cai, H. Qiu, L. Gao, C. Jiang, and X. Shao, "An efficient surrogate-assisted particle swarm optimization algorithm for high-dimensional expensive problems," *Knowledge-Based Systems*, vol. 184, 2019.

[18] P. Zhang, Z. Yin, Y. Jin, and T. H. T. Chan, "A novel hybrid surrogate intelligent model for creep index prediction based on particle swarm optimization and random forest," *Engineering Geology*, vol. 265, 2020.

[19] G. Chen, K. Zhang, L. Zhang et al., "Global and local surrogate-model-assisted differential evolution for waterflooding production optimization," *SPE Journal*, vol. 25, no. 1, pp. 105–118, 2020.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[21] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, pp. 1945–1950, Washington, DC, USA, 1999.

[22] Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546),*, pp. 81–86, Seoul, Korea (South), 2001.

[23] Y. Gong, J. J. Li, Y. Zhou et al., "Genetic learning particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2277–2290, 2016.

[24] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Computing*, vol. 19, no. 6, pp. 1461–1475, 2015.

[25] Z. Zhou, Y. S. Ong, M. H. Nguyen, and D. Lim, "A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm," in *2005 IEEE Congress on Evolutionary Computation*, pp. 2832–2839, Edinburgh, UK, 2005.

[26] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Reliability Engineering & System Safety*, vol. 81, no. 1, pp. 23–69, 2003.

[27] Y. Liu, J. Hou, H. Zhao, X. Liu, and Z. Xia, "A method to recover natural gas hydrates with geothermal energy conveyed by CO2," *Energy*, vol. 144, no. 1, pp. 265–278, 2018.

[28] Z. Xu, T. Hu, X. Pang et al., "Research progress and challenges of natural gas hydrate resource evaluation in the South China Sea," *Petroleum Science*, 2021.

[29] X. Pang, Z. Chen, C. Jia et al., "Evaluation and re-understanding of the global natural gas hydrate resources," *Petroleum Science*, vol. 18, no. 2, pp. 323–338, 2021.

[30] G. J. Moridis, "Numerical studies of gas production from Class 2 and Class 3 hydrate accumulations at the Mallik site, Mackenzie delta, Canada," *SPE Reservoir Evaluation & Engineering*, vol. 7, no. 3, pp. 175–183, 2004.

[31] G. J. Moridis, S. Silpngarmlert, M. T. Reagan, T. Collett, and K. Zhang, "Gas production from a cold, stratigraphically-bounded gas hydrate deposit at the Mount Elbert Gas Hydrate Stratigraphic Test Well, Alaska North Slope: implications of uncertainties," *Marine and Petroleum Geology*, vol. 28, no. 2, pp. 517–534, 2011.

[32] Y. Zhang, X. Li, Y. Wang, Z. Chen, and G. Li, "Methane hydrate formation in marine sediment from South China Sea with different water saturations," *Energies*, vol. 10, no. 4, pp. 561–561, 2017.

[33] Y. Zhang, X. Li, Y. Wang, Z. Chen, and K. Yan, "Decomposition conditions of methane hydrate in marine sediments from South China Sea," *Fluid Phase Equilibria*, vol. 413, pp. 110–115, 2016.

[34] Y. Zhang, X. Li, Z. Chen, Y. Wang, and X. Ruan, "Effect of hydrate saturation on the methane hydrate dissociation by depressurization in sediments in a cubic hydrate simulator," *Industrial & Engineering Chemistry Research*, vol. 54, no. 10, pp. 2627–2637, 2015.

[35] Q. Wan, H. Si, B. Li et al., "Energy recovery enhancement from gas hydrate based on the optimization of thermal stimulation modes and depressurization," *Applied Energy*, vol. 278, no. 15, pp. 115612–115614, 2020.

[36] Z. Yang, H. Si, and D. Zhong, "AI-based composition model for energy utilization efficiency optimization of gas hydrate recovery by combined method of depressurization and thermal stimulation," *Journal of Natural Gas Science and Engineering*, vol. 92, pp. 104001–104015, 2021.

[37] Y. Song, L. Yang, J. Zhao et al., "The status of natural gas hydrate research in China: a review," *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 778–791, 2014.

[38] R. Dimitrakopoulos and X. Luo, "Generalized sequential gaussian simulation on group size v and screen-effect approximations for large field simulations," *Mathematical Geology*, vol. 36, no. 5, pp. 567–591, 2004.

[39] G. F. Reed, F. Lynn, and B. D. Meade, "Use of coefficient of variation in assessing variability of quantitative assays," *Clinical and Vaccine Immunology*, vol. 9, no. 6, pp. 1235–1239, 2002.

[40] Y. Liu, Y. Bai, Z. Xia, and J. Hou, "Parameter optimization of depressurization–to–hot–water–flooding in heterogeneous hydrate bearing layers based on the particle swarm optimization algorithm," *Journal of Natural Gas Science and Engineering*, vol. 53, pp. 403–415, 2018.