

Article

Comparative Evaluation of the Application Effectiveness of Intelligent Production Optimization Methods in Offshore Oil Reservoirs

Chen Liu ^{1,2,3}, Qihong Feng ^{1,*}, Kai Zhang ¹ , Jialin Wang ¹ and Jingqi Lin ¹

¹ School of Petroleum Engineering, China University of Petroleum (East China), Qingdao 266555, China; liu099339@163.com (C.L.); zhangkai@upc.edu.cn (K.Z.); 1802010205@s.upc.edu.cn (J.W.); b22020057@s.upc.edu.cn (J.L.)

² State Key Laboratory of Offshore Oil Exploitation, Beijing 100028, China

³ CNOOC Research Institute Ltd., Beijing 100028, China

* Correspondence: fengqihong.upc@gmail.com; Tel.: +86-0532-86981229

Abstract: The development of offshore oil fields confronts challenges associated with high water cut and low displacement efficiency. Reservoir injection-production optimization stands out as an effective means to reduce costs and enhance efficiency in offshore oilfield development. The process of optimizing injection and production in offshore oil reservoirs involves designing strategies for a large number of wells and optimization time steps, constituting a large-scale, complex, and costly optimization computation problem. In recent years, with the rapid advancements in big data and artificial intelligence technologies, sophisticated evolutionary computation methods have found widespread application in reservoir injection-production optimization problems. However, the abundance of intelligent optimization algorithms raises the question of how to choose a method suitable for the complex optimization background of offshore oilfield injection-production optimization. This paper provides a detailed overview of the application of an existing differential evolution algorithm (DE), conventional surrogate-assisted evolutionary algorithm (CSAEA), and global–local surrogate-assisted differential evolution (GLSADE) in the context of practical offshore oilfield injection-production optimization problems. A comprehensive comparison of their performance differences is presented. The study concludes that the global–local surrogate-assisted evolutionary algorithm is the most suitable method for addressing the current challenges in offshore oilfield injection-production optimization.

Keywords: production optimization; surrogate model; offshore oil reservoirs; evolutionary algorithm; comparative effectiveness



Citation: Liu, C.; Feng, Q.; Zhang, K.; Wang, J.; Lin, J. Comparative Evaluation of the Application Effectiveness of Intelligent Production Optimization Methods in Offshore Oil Reservoirs. *Water* **2024**, *16*, 1732. <https://doi.org/10.3390/w16121732>

Academic Editor: Elias Dimitriou

Received: 15 April 2024

Revised: 31 May 2024

Accepted: 4 June 2024

Published: 19 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The offshore oil reservoirs face challenges such as high water cut, low permeability, and uneven reservoir development. How to fully exploit the potential of existing oil reservoirs and improve economic benefits is an issue of great engineering significance. Production optimization, by adjusting the injection-production relationship, can effectively increase oil recovery while reducing costs, and it is widely applied in offshore oil fields to reduce costs and enhance efficiency [1–5]. There are primarily two categories of production optimization methods, with the first being gradient methods. Two line search methods, namely steepest descent and conjugate gradient, were employed and compared by Asadollahi and Naevdal [6], and the results demonstrated that the conjugate gradient method can rapidly search for the optimal development scheme. Forouzanfar et al. utilized a gradient-based enhanced Lagrangian optimization code for the life cycle production optimization of oil reservoirs [7]. The objective was to identify the optimal well controls for a 15-year production period. In comparison with the reference case, the net present value increased by 8.9% over the 15-year production period. Liu and Reynolds [8] used gradients

calculated with the adjoint method to find boundary points by maximizing the augmented Lagrangian function, optimizing both the life-time net present value and short-term net present value. Bukshtynov et al. [9] developed and applied an efficient, robust, and flexible closed-loop reservoir management computational framework using a gradient-based production optimization method. This approach aimed to enhance the net present value through the utilization of automatic differentiation. Sefat et al. [10] provided guidance for selecting appropriate reservoir production optimization methods by employing state-of-the-art stochastic gradient approximation algorithms. Volkov and Bellout [11] developed an analytical framework to investigate the impact of enforcing simulator-based economic constraints during the execution of gradient-based production optimization. This approach enhanced the performance of the search process.

While gradient-based methods can quickly obtain computational results, for complex oil reservoir production optimization problems, obtaining gradient information is often challenging and may lead to difficulty in searching for the global optimal scheme. To address the aforementioned challenges, heuristic-based evolutionary algorithms are widely applied in the field of production optimization, adept at resolving high-dimensional optimization problems [12]. Evolutionary algorithms, such as differential evolution [13], particle swarm optimization [14], and genetic algorithm [15], possess strong global search capabilities. They gradually approach the optimal solution by generating a large number of candidate development schemes. Reddy and Kumar proposed an efficient and effective multi-objective optimization method based on the differential evolution algorithm, applying it to a case study of reservoir system optimization [16]. This approach can generate Pareto optimal solutions for multi-objective reservoir operation problems with convergence. Siavashi and Yazdani compared the performance of genetic algorithm (GA) and particle swarm optimization (PSO) in enhancing enhanced oil recovery (EOR) projects and combined the Newton method with PSO to improve its convergence speed [17]. Gu et al. utilized the differential evolution algorithm to optimize a reservoir water-cut minimization control model, effectively improving water displacement efficiency [18]. Zhang et al. employed the differential evolution algorithm to solve constrained reservoir production optimization problems, achieving higher net present values [19]. An et al. introduced conventional particle swarm optimization into reservoir engineering methods, reducing the computational costs of production optimization in commercial oil fields [20].

Although evolutionary algorithms can enhance global search capabilities, as mentioned earlier, they require generating a large number of candidate schemes to approximate the optimal solution. If all candidate solutions are evaluated and screened through numerical simulators, it can be extremely time-consuming. To address this time-consuming issue, in recent years, a method that utilizes surrogate models to replace reservoir numerical simulators and expedite the optimization process has gained popularity in the field of production optimization. A surrogate model is a mathematical model that takes development schemes as inputs and provides reservoir responses as outputs. In other words, by evaluating a small number of development schemes through a numerical simulator, a high-precision surrogate model is constructed. This surrogate model is then used in the subsequent optimization process to replace the numerical simulator, enabling the rapid generation of reservoir responses. Surrogate models are data-driven and come in various types, such as radial basis function network [21], support vector machine regression [22], and the Gaussian process [23], among others. Based on the type and quantity of surrogate models used, optimization methods can be classified into two categories: single surrogate optimization methods and multi-surrogate optimization methods. Let us first introduce single surrogate optimization methods. Golzari et al. proposed a novel approach for constructing an adaptive surrogate model, employing a dynamic artificial neural network as the surrogate model to achieve a rapid approximation of the actual reservoir simulation model [24]. This method demonstrates good accuracy, enhancing the overall optimization process. Aladeitan et al. utilized surrogate modeling methods to assess the optimal well designs for three different development strategies aiming to enhance the

recovery factor of a marginal oil reservoir [25]. Chen et al. proposed a new framework based on surrogate-assisted evolutionary algorithm and dimensionality reduction [26]. This framework utilizes Sammon mapping to map high-dimensional decision variables to low-dimensional space, achieving better performance. Zhao et al. proposed a new method for solving multi-objective production optimization problems [27]. This method employs radial basis function networks as surrogate models to predict the dominant relationship between candidate solutions and pre-screening. Ogbeiwi et al. deployed a surrogate model for total oil production to perform robust optimization of water flooding in the Niger Delta reservoir, significantly reducing computational costs [28]. In contrast to single surrogate optimization methods, multi-surrogate optimization involves the participation of multiple surrogate models, either of the same or different types, in the optimization process. Chen et al. proposed a production optimization method that utilizes both global and local surrogate models to assist differential evolution [29]. The collaboration between global optimization and local search contributes to making the production optimization process more efficient and effective. Du et al. established a multi-objective production optimization framework that combines Bayesian Random Forest and Particle Swarm Optimization algorithms [30]. This framework effectively balances oil production and the gas–oil ratio. Wang et al. employed a multi-surrogate model framework using polynomial response and radial basis function networks, reducing prediction uncertainty in the production optimization process [31]. Wang et al. [32] employed a multi-surrogate model production optimization framework based on Pareto criteria, expediting the search for optimal solutions.

Although evolutionary algorithms and surrogate models have been widely applied in production optimization, there is currently a lack of comparative studies in the context of offshore oil reservoirs. Offshore oil reservoirs pose challenges with a multitude of optimization variables and harsh production conditions, making it essential to compare various optimization methods for informed decision-making. Therefore, this paper selects representative algorithms from evolutionary optimization, single surrogate optimization methods, and multi-surrogate optimization methods. Performance tests for production optimization problems are conducted on two real offshore oil reservoir cases. The performance comparison of these methods aims to assist field engineers in better applying advanced intelligent computing methods to the field of offshore oil reservoir production optimization. The remaining structure of the article is as follows. Chapter two introduces the production optimization issues in offshore oil reservoirs, providing the foundational context for the problems addressed in this paper. Chapters three to five, respectively, delve into the three categories of methods being compared. Chapter six presents a comparative analysis of the application effectiveness of these three methods on two oil reservoir cases. Finally, the paper concludes with the findings and conclusions drawn from the study.

2. Problem Statement

Production optimization in offshore oil reservoirs is primarily achieved through controlling the development schemes of production and injection wells to maximize cumulative oil production or economic benefits. The term “development schemes” here refers to the bottom-hole pressure or flow rate of each well. In practical production, economic benefits are typically chosen as the optimization objective, specifically the net present value (NPV). NPV can be expressed using the following formula:

$$J(x, v) = \sum_{t=1}^{N_t} \left\{ \frac{\Delta t}{(1+b)^{\frac{\Delta t}{365}}} \left[\sum_{j=1}^{N_p} (r_o \cdot q_{o,j}^t - r_w \cdot q_{w,j}^t) - \sum_{j=1}^{N_I} (r_i \cdot q_{i,j}^t) \right] \right\} \quad (1)$$

where J is the objective function for optimization, representing the NPV; x represents the development scheme, and v is the state variable containing reservoir characteristics. N_t represents the total number of optimization time steps; Δt represents the time interval for each time step in the optimization; b is the discount rate; and $q_{o,j}^t, q_{w,j}^t$ are the oil production rate, water production rate of the j -th production well at time step t (STB/D), respectively;

$q_{i,j}^t$ is the water injection rate of the j -th injection well at time step t (STB/D); r_o , r_w , and r_i are oil production revenue, produced water treatment cost, and injection cost, all in USD/STB, respectively; N_P and N_I represent the number of production and injection wells, respectively.

In addition, this paper also considers the boundary constraint of the development scheme. Therefore, the production optimization problem for offshore oil reservoir can be defined by the following formula:

$$\max_x J(x, v), x \in \mathbb{R}^d, \text{ s.t. } x^{\text{low}} \leq x \leq x^{\text{up}} \quad (2)$$

where $d = A \times N_t$ is the dimension number of the development scheme, A denotes the number of production and injection wells, i.e., $N_P + N_I$; x^{low} and x^{up} represent the upper and lower bounds of the development scheme, respectively.

3. Differential Evolutionary Algorithm for Solving Production Optimization Problems in Offshore Reservoirs

The differential evolution (DE) algorithm, as an optimization algorithm with global search capabilities, has been widely applied in the industrial field in recent years [33]. Although many new optimization algorithms are emerging in the field of intelligent computing, due to the stability of the DE, it performs better in reservoir production optimization [32]. Therefore, this paper chooses it as a comparative algorithm for solving offshore oil reservoir production optimization problems. The overall process of DE is shown in Algorithm 1.

Algorithm 1: Pseudocode of DE

01:	Initialization: Generate development schemes $\{x_i\}_{i=1}^N$ using Latin hypercube sampling [34] in the boundary constraint space, evaluate them with the numerical simulator to obtain NPV $\{y_i\}_{i=1}^N$, and then store both the schemes and NPV in a database DB ;
02:	While the stopping criterion is not met
03:	Mutation: Generate mutation vectors by applying the mutation factor to $\{x_i\}_{i=1}^N$;
04:	Crossover: Generate new schemes by applying the crossover factor to $\{x_i\}_{i=1}^N$ and mutation vectors;
05:	Selection: Calculate the NPV for all new schemes using the numerical simulator and store the best one along with its NPV in the DB ;
06:	End while
07:	Output: The best scheme in DB .

The core idea of DE is to use the existing development schemes as a foundation, generate new schemes using crossover and mutation operators, and select the optimal individuals in the new schemes based on their NPV. This process is repeated until the predetermined maximum number of numerical simulation iterations is reached. Initially, the existing development schemes $x = \{x_i\}_{i=1}^N$ are generated by applying the Latin hypercube sampling method according to the boundary constraints outlined in Equation (2). Their NPV $\{y_i\}_{i=1}^N$ is calculated using a numerical simulator, and a sample library DB is constructed, where N is the number of development schemes. The first step of DE is to generate mutation vectors based on the existing development schemes using a mutation factor:

$$v_i = x_{\text{best}} + F(x_{m1} - x_{m2}) \quad (3)$$

where v_i is the mutation vector corresponding to the i -th development scheme; x_{best} is the individual in the existing schemes with the highest NPV; F is the pre-defined muta-

tion factor; m_1 and m_2 are two random integers between 1 and N . The next step is the crossover process:

$$x_i^{*j} = \begin{cases} v_i^j, & \text{if } R_j(0, 1) \leq C \text{ or } j = j_{\text{rand}} \\ x_i^j, & \text{otherwise} \end{cases} \quad (4)$$

The individual x_i in the existing development schemes can be represented as the following: $x_i = \{x_i^1, x_i^2, x_i^3, \dots, x_i^d\}$; d is the dimension number of the development scheme. x_i^{*j} represents the j -th dimension of the new scheme obtained by updating the i -th existing scheme through the DE algorithm; $R_j(0, 1)$ is a random number between 0 and 1; C is a pre-defined crossover factor. During the interaction in each dimension, a random number $R_j(0, 1)$ is generated. If $R_j(0, 1)$ is less than the pre-defined crossover factor, then in that dimension, the new scheme will be the j -th dimension of the mutation vector. Otherwise, the existing scheme will be retained. j_{rand} is a random integer between 1 and d , ensuring that the new scheme cannot be identical to the existing scheme. After generating new schemes, the selection process is to select the one with the highest NPV among all the new schemes and add it to the database:

$$x^{\text{new}} = \max_x J(x^*, v) \quad (5)$$

where x^{new} is the scheme after the selection process and it is placed in the database together with the corresponding NPV; x^* represents all the new schemes generated after the crossover phase, and their NPV is calculated entirely using a numerical simulator.

4. Canonical Surrogate Assisted Evolutionary Algorithm for Solving Production Optimization Problems in Offshore Reservoirs

As mentioned earlier, when solving the offshore oil reservoir production optimization problem using only the differential evolution algorithm, it is necessary to use a numerical simulator to calculate the generated numerous candidate solutions, significantly increasing computational costs. This issue is reflected in step 5 of Algorithm 1, where the NPV of all new schemes needs to be computed using the numerical simulator, leading to a reduced number of optimization iterations within the loop. To solve this problem, the surrogate model method is used to assist the evolutionary algorithm in the rapid determination of the optimal development scheme. This method is referred to as the canonical surrogate-assisted evolutionary algorithm (CSAEA) [35]. CSAEA employs a training phase to train surrogate models using real samples, thereby reducing the reliance on numerical simulators during subsequent optimizations. The surrogate model is constructed by taking development schemes $\{x_i\}_{i=1}^N$ as input and NPV $\{y_i\}_{i=1}^N$ as output. After the surrogate model is established, it can quickly generate reservoir responses as a substitute for numerical simulation. Among these, the radial basis function network (RBFN) has been widely applied in the field of production optimization. In this paper, the RBFN is employed to assist the differential evolution as the second comparative algorithm. Let us start by introducing the RBFN. RBFN is an artificial neural network that approximates the numerical simulator by summing weighted basis functions. The expression is as follows:

$$f(x) = \sum_{i=1}^N \{w_i \varphi(|x - x_i|)\} \quad (6)$$

where $f(x)$ is the predicted value of the RBFN; w_i represents the weight coefficients; x is the development scheme to be predicted; x_i is a development scheme in the DB; N is the total number of development schemes, and φ is the basis function. In this paper, the cubic basis function is chosen, and the expression is as follows:

$$\varphi(x) = x^3 \quad (7)$$

After the establishment of RBFN, it can be used in subsequent optimization to replace numerical simulators for screening new development schemes. The overall process of CSAEA is shown in Algorithm 2. Its overall process is similar to DE, with the most crucial difference lying in step 5. In this step, the surrogate model is used to predict the NPV of all new schemes, reducing the need for the numerical simulator to evaluate only the selected solution with the highest predicted value. This significantly reduces the number of numerical simulations required. The flowchart of CSAEA is shown in Figure 1.

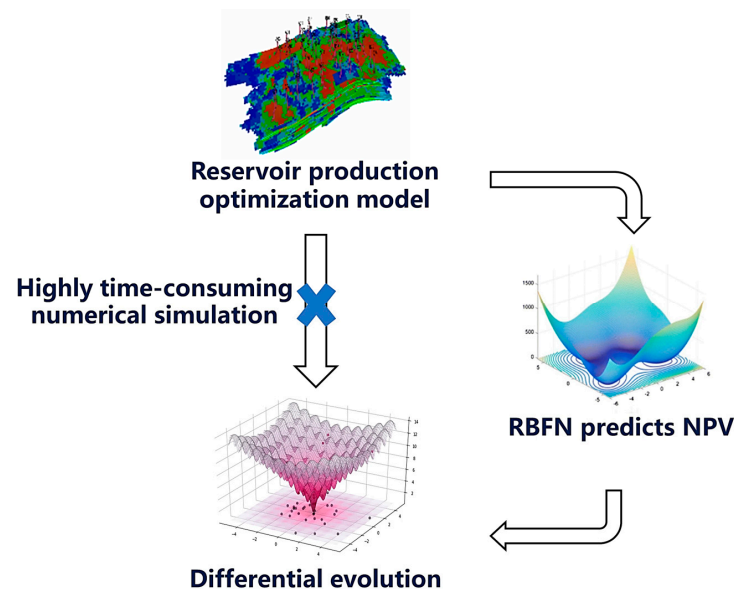


Figure 1. Flow diagram of CSAEA.

Algorithm 2: Pseudocode of CSAEA

01:	Initialization: Generate development schemes $\{x_i\}_{i=1}^N$ using Latin hypercube sampling in the boundary constraint space, evaluate them with the numerical simulator to obtain NPV $\{y_i\}_{i=1}^N$, and then store both the schemes and NPV in a database <i>DB</i> ;
02:	While the stopping criterion is not met
03:	Train an RBFN surrogate model using the development schemes in <i>DB</i> and NPV;
04:	Generating new schemes using the DE algorithm.
05:	Select the best one from the new schemes based on the predicted values of the RBFN model, calculate its NPV using the numerical simulator, and update the database;
06:	End while
07:	Output: The best scheme in <i>DB</i> .

5. Global–Local Surrogate-Assisted Differential Evolution for Solving Production Optimization Problems in Offshore Reservoirs

CSAEA utilizes all samples in the database for the construction of the RBFN model, which results in a higher model accuracy. However, due to the uneven quality of samples in the database, it is challenging to quickly pinpoint the optimal development scheme. To address this issue, Chen et al. proposed a global–local surrogate-assisted differential evolution (GLSADE) algorithm, constructing two surrogate models to accelerate the optimization process [29]. One of them is the global surrogate model, which, similar to the RBFN established in CSAEA, is trained using all the samples in the database *DB*. It is noteworthy that the second surrogate model established is a local surrogate model, trained with outstanding schemes selected based on NPV ranking from the *DB*. The advantages of establishing a local surrogate model to participate in the search can be illustrated using Figure 2:

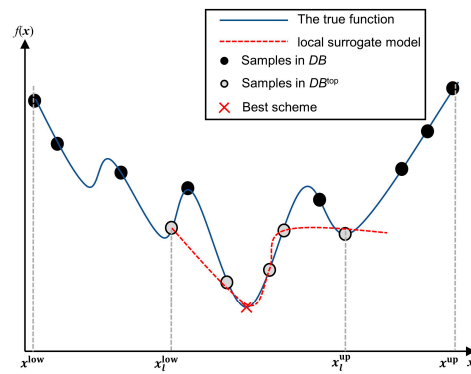


Figure 2. Flow diagram of the advantages of the local surrogate model.

As shown in Figure 2, the blue solid line represents the true function curve, i.e., the numerical simulation; the red dashed line represents the local surrogate model; the black solid dots represent all samples in the database DB ; the gray solid dots represent samples with NPV under a specific threshold, forming a subset of the database denoted as DB^{top} ; the red cross represents the optimal sample. From the figure, it can be observed that if a surrogate model is built for optimization using samples within the range of x^{low} and x^{up} , the search scope is too broad, making it difficult to quickly locate the optimal solution. However, by constructing a local surrogate model with the gray excellent samples, the search can be performed within the narrowed range of x_l^{low} and x_l^{up} , leading to a rapid identification of the optimal solution. x_l^{low} and x_l^{up} can be calculated using the following expression:

$$\begin{aligned} x_l^{low} &= \min(DB^{top}) \\ x_l^{up} &= \max(DB^{top}) \end{aligned} \quad (8)$$

x_l^{low} and x_l^{up} are composed of the minimum and maximum values of each dimension across all samples in the DB^{top} , respectively. The overall process of GLSADE is shown in Algorithm 3.

Algorithm 3: Pseudocode of GLSADE

01:	Initialization: Generate development schemes $\{x_i\}_{i=1}^N$ using Latin hypercube sampling in the boundary constraint space, evaluate them with the numerical simulator to obtain NPV $\{y_i\}_{i=1}^N$, and then store both the schemes and NPV in a database DB ;
02:	While the stopping criterion is not met
03:	//Global surrogate model search phase
04:	Construct a global RBFN model using all the samples in DB ;
05:	Generating new schemes using the DE algorithm;
06:	Select the best one from the new schemes based on the predicted values of the global RBFN model, calculate its NPV using the numerical simulator, and update the database;
07:	//Local surrogate model search phase
08:	Sort the samples in the DB based on NPV and select the n top-ranked samples to generate a subset database DB^{top} ;
09:	Construct a local RBFN model using all the samples in DB^{top} ;
10:	Calculate x_l^{low} and x_l^{up} based on the minimum and maximum values of each dimension for all samples in the DB^{top} ;
11:	Generate optimized initial schemes using LHS within the range of x_l^{low} and x_l^{up} ;
12:	Generating new schemes using the DE algorithm within the range of x_l^{low} and x_l^{up} ;
13:	Select the best one from the new schemes based on the predicted values of the local RBFN model, calculate its NPV using the numerical simulator, and update the database;
14:	End While
15:	Output: The best scheme in DB .

As illustrated in Algorithm 3, the global surrogate model search phase of GLSADE is roughly similar to CSAEA. In the local search phase, firstly, the n top-ranked samples are selected from the DB to construct a data subset DB^{top} . Subsequently, DB^{top} is used to train a local RBFN model. Then, the upper and lower limits of the DB^{top} are determined. In both CSAEA and DE, new schemes are generated based on all schemes in the DB . However, in the local search phase of GLSADE, optimized initial schemes are generated within the determined superior upper and lower limits using LHS. These solutions are then updated using DE, and finally, the local RBFN model is employed for filtering to expedite the optimization process.

6. Experimental Results and Discussion

To verify the performance of DE, CSAEA, and GLSADE in solving the production optimization problem of offshore oil fields, tests are conducted using two offshore actual oil fields. It is worth noting that the heterogeneity of offshore oil reservoirs is relatively strong, and generally, their well networks are sparse, resulting in higher development costs. The two reservoirs selected in this article are both offshore oil reservoirs with low permeability and high-water content, and low development potential. One of them is optimized in a moderate dimension, while the other is set as a high-dimensional optimization problem (the specific dimensions will be detailed below), which corresponds to the two main types of problems addressed in the field of reservoir development. Conducting case studies on them can effectively demonstrate the performance of the algorithms. All simulation experiments in this paper are carried out using ECLIPSE as the numerical simulator. The stopping criterion for testing on the two reservoir cases reaches 1000 evaluations of the numerical simulator. The initial sample size N for all methods is set to 200, which also serves as the training phase for the surrogate models of CSAEA and GLSADE. The mutation factor F is set to 0.5, and the crossover factor C is set to 0.5. The number of top-ranked samples n in GLSADE is set to 200.

Example 1: The offshore actual reservoir model A. The first case tested in this paper is model A, which has $50 \times 30 \times 147$ grid blocks, and the grid size is $51 \text{ m} \times 31 \text{ m} \times 148 \text{ m}$. This reservoir is developed using water flooding, with a total of three water-injection wells and nine production wells. The locations of these wells and the permeability of the model field are shown in Figure 3. The properties of the A model are presented in Table 1. The three water-injection wells are controlled by injection rates, with the lower limits set at 20, 200, and 100 STB/D, respectively, and upper limits set at 100, 978, and 1200 STB/D, respectively. The nine production wells are controlled by production rates, with the lower limits set at 20, 10, 60, 20, 100, 90, 15, 70, and 15 STB/D, respectively, and upper limits set at 160, 38, 330, 73, 220, 296, 260, 290, and 93.8 STB/D, respectively. The reservoir production period is 720 days, divided into four control steps, each spanning 180 days on average. Therefore, the total number of decision variables is $(3 + 9) \times 4 = 48$. The oil revenue, water injection cost, and water treatment cost are USD/STB 80, 2, and 2, respectively. The discount rate is 0%.

Figure 4 shows the optimization results obtained from independently running the three methods on A model for 10 iterations. As shown in Figure 4, DE exhibits the poorest performance. This is because it lacks a surrogate model to assist in optimization and relies solely on evaluating all candidate development schemes using the numerical simulator, greatly increasing computational burden and resulting in fewer optimization iterations. The CSAEA methodology employs surrogate models to expedite the optimization process, thereby achieving superior results compared to DE. This observation aligns with the research conducted by Golzari et al. [24], affirming that the utilization of surrogate models enables significantly more iterations within a finite timeframe, facilitating the exploration of optimal outcomes. However, CSAEA's reliance on a single surrogate model for the search process constrains its capacity to rapidly pinpoint the optimum solution. In contrast, GLSADE leverages multiple surrogate models to balance exploration and exploitation, resulting in superior solutions within the same iteration period, corroborating the un-

derstanding posited by Chen et al. [29]. GLSADE employs a global surrogate model to preliminarily screen advantageous development schemes, and a local surrogate model to conduct in-depth exploration of high-potential regions. This approach markedly augments the probability of identifying the optimum solution, thereby yielding optimal performance.

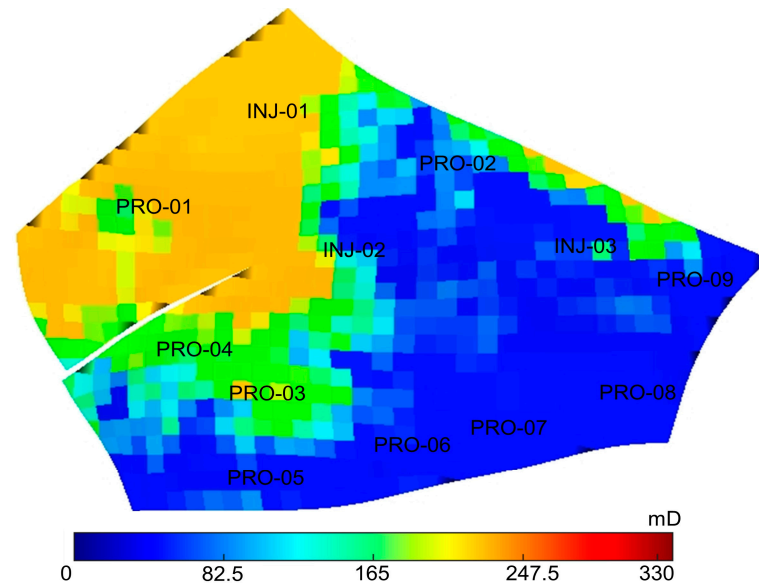


Figure 3. Well locations and permeability field of the A model.

Table 1. Properties of the A model.

Properties	Value
Reservoir grid	$50 \times 30 \times 47$
Depth	2000 m
Initial pressure	22.4 MPa
porosity	0.086
Water compressibility	$5.45 \times 10^{-5}/\text{MPa}$
Rock compressibility	$1 \times 10^{-8}/\text{MPa}$
Density	850 kg/m^3
Initial oil saturation	0.6
Oil viscosity	0.7 cP

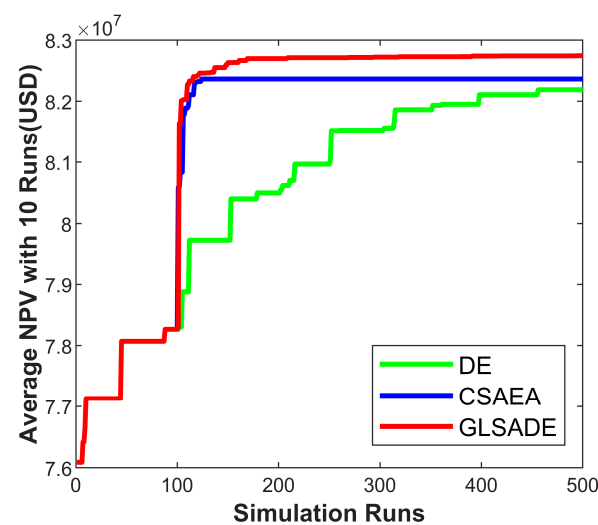


Figure 4. Average NPV with 10 runs for the A model.

Figures 5 and 6 show the optimal control schemes for producers and injectors obtained using the three methods. Each row represents the control scheme of a particular well over the entire optimization period, with each box indicating the control scheme of a particular well for a specific control step. Figure 7, respectively, shows the variations in cumulative oil production (FOPT), cumulative water injection (FWIT), and cumulative water production (FWPT) over the development time under optimal well control. The well control scheme of DE provides the lowest cumulative oil production, the highest cumulative water production, and cumulative water injection, thus resulting in the lowest economic benefits. The well control scheme of GLSADE can provide the highest cumulative oil production while not necessarily resulting in the highest cumulative water production or injection. Therefore, the ultimate economic benefits are the highest.

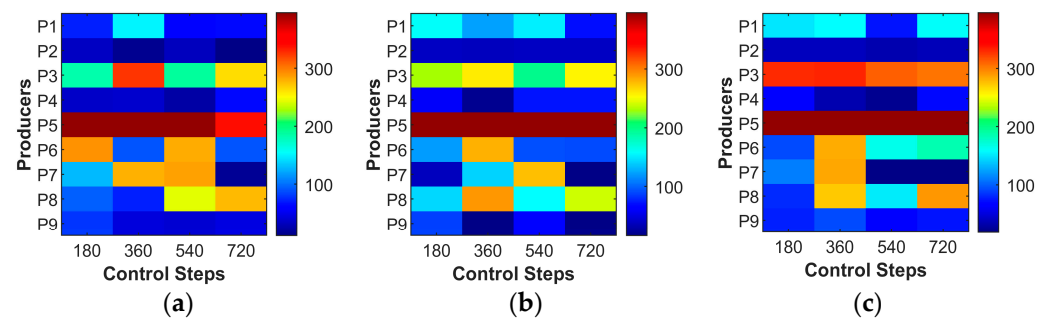


Figure 5. The optimal production-rate well controls provided for the A model. (a) DE. (b) CSAEA. (c) GLSADE.

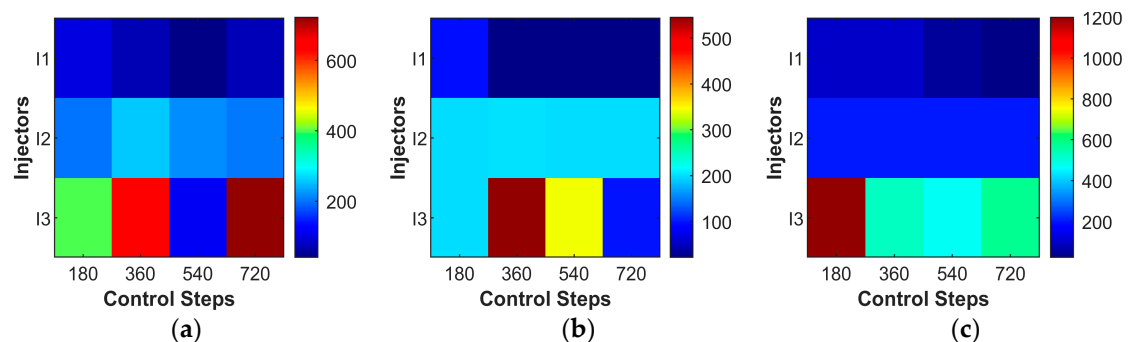


Figure 6. The optimal water-injection-rate well controls provided for the A model. (a) DE. (b) CSAEA. (c) GLSADE.

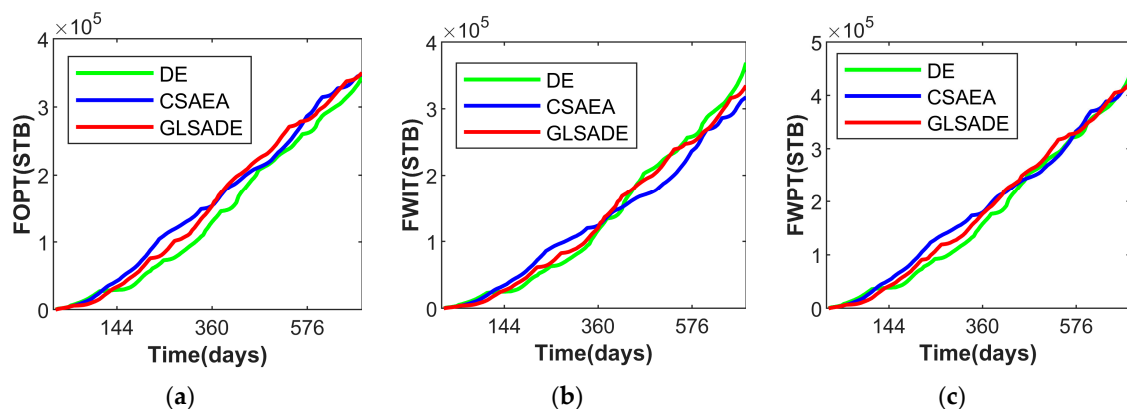


Figure 7. Results of optimal control for the A model. (a) Cumulative oil production vs. time. (b) Cumulative water injection vs. time. (c) Cumulative water production vs. time.

Example 2: The offshore actual reservoir model B. The second case tested in this paper is model B, which has $99 \times 99 \times 80$ grid blocks, and the grid size is $51 \text{ m} \times 31 \text{ m} \times 148 \text{ m}$. This reservoir is developed using water flooding, with a total of two water-injection wells and six production wells. The locations of these wells and the permeability field are shown in Figure 8 of the model. The properties of the B model are presented in Table 2. The two water-injection wells are controlled via injection rates, with lower limits set at 300 and 300 STB/D, respectively, and upper limits set at 627 and 677 STB/D, respectively. The six production wells are controlled via production rates, with lower limits set at 20, 10, 20, 20, 10, and 30 STB/D, respectively, and upper limits set at 230, 61, 240, 113, 50, and 150 STB/D, respectively. The reservoir production period is 720 days and divided into 20 control steps, each spanning 36 days on average. Therefore, the total number of decision variables is $(2 + 6) \times 20 = 160$. The oil revenue, water injection cost, and water treatment cost are USD/STB 80, 2, and 2, respectively. The discount rate is 0%.

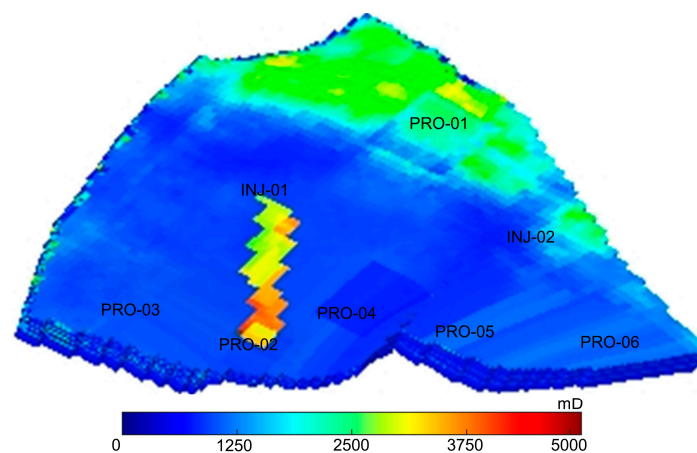


Figure 8. Well locations and permeability field of the B model.

Table 2. Properties of the B model.

Properties	Value
Reservoir grid	$99 \times 99 \times 80$
Depth	2110 m
Initial pressure	23.16 MPa
porosity	0.15
Water compressibility	$5 \times 10^{-5}/\text{MPa}$
Rock compressibility	$1 \times 10^{-8}/\text{MPa}$
Density	850 kg/m^3
Initial oil saturation	0.56
Oil viscosity	0.85 cP

Figure 9 shows the optimization results obtained from independently running the three methods on B model for 10 iterations. The experimental result similar to those exhibited using the A model shows that DE still performs the worst due to the absence of using a surrogate model, hence, it is unable to expedite the solving process. In the experiments of B model, it can be observed that GLSADE's performance is significantly superior to CSAEA. The variable dimension in B model reaches 160, indicating a large-scale optimization problem. From the experimental results, it can be inferred that GLSADE is more suitable for offshore oilfield production optimization problems with a large number of wells or time steps compared to conventional surrogate optimization methods. This is in line with other research on multi-surrogate models [31,32], where the synergistic coupling of multiple surrogate models significantly enhances the global optimization capability for complex problems with large search spaces. This approach circumvents premature

convergence to locally optimal solutions at certain times, thereby achieving more optimal outcomes over the entire development cycle.

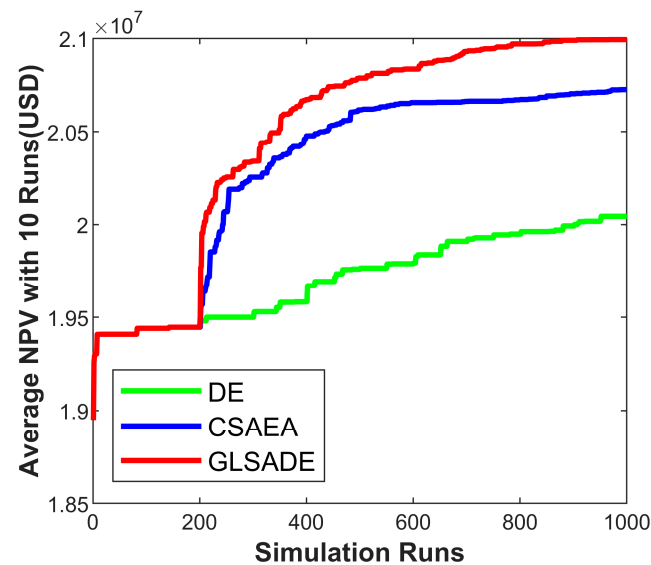


Figure 9. Average NPV with 10 runs for the B model.

Figures 10 and 11 show the optimal control schemes for producers and injectors obtained using the three methods. Figure 12 illustrates the curves of FOPT, FWIT, and FWPT over time for the three methods. It can be seen from Figure 12 that although GLSADE injects more water and produces more water, it can yield the highest cumulative oil production, thereby achieving the maximum economic benefit.

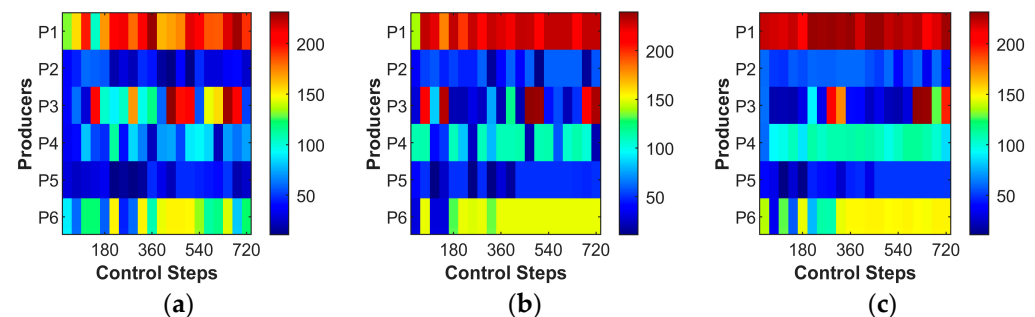


Figure 10. The optimal production-rate well controls provided for the B model. (a) DE. (b) CSAEA. (c) GLSADE.

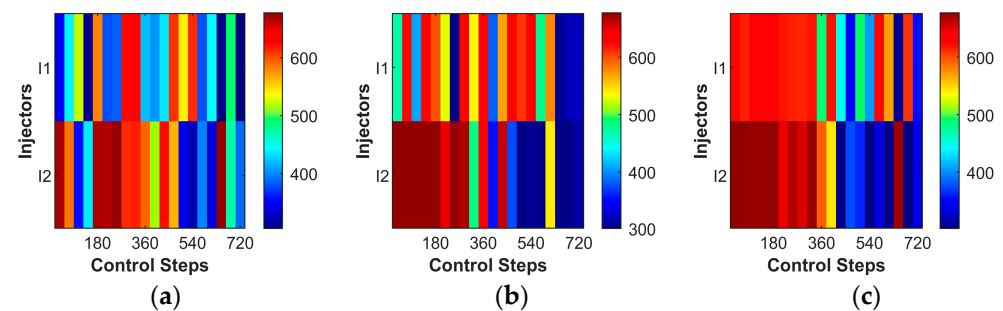


Figure 11. The optimal water-injection-rate well controls provided for the B model. (a) DE. (b) CSAEA. (c) GLSADE.

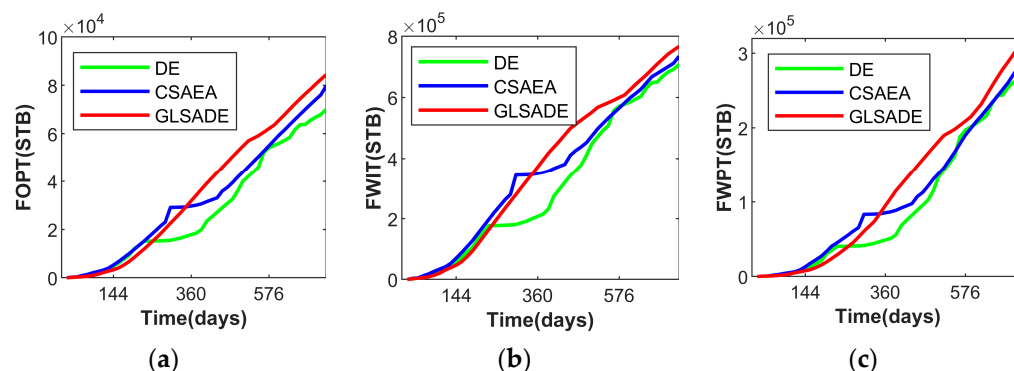


Figure 12. Results of optimal control for the B model. (a) Cumulative oil production vs. time. (b) Cumulative water injection vs. time. (c) Cumulative water production vs. time.

7. Conclusions

In addressing the problem of offshore oilfield production optimization, this paper compares the application effectiveness of the differential evolution algorithm (DE), conventional surrogate-assisted evolutionary algorithm (CSAEA), and global-local surrogate-assisted differential evolution (GLSADE) in two actual oilfields. The experimental results indicate that GLSADE, through constructing two surrogate models to participate in optimization, is capable of grasping the overall optimization trend globally while also delving into local information, thus accelerating the optimization process. It achieves the best optimization results in both oilfield cases. Particularly noteworthy is its outstanding performance in the second oilfield case characterized by large-scale optimization, indicating its suitability for addressing offshore oilfield production optimization problems with numerous wells and time steps, thereby possessing significant engineering application value.

Author Contributions: Conceptualization, Q.F. and K.Z.; methodology, C.L.; software, C.L.; validation, C.L. and J.W.; formal analysis, C.L.; investigation, C.L. and J.L.; resources, Q.F. and K.Z.; data curation, C.L. and K.Z.; writing—original draft preparation, C.L.; writing—review and editing, Q.F.; visualization, C.L.; supervision, Q.F.; project administration, K.Z.; funding acquisition, Q.F. and K.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under Grant 52325402, 52274057, 52074340 and 51874335, the National Key R&D Program of China under Grant 2023YFB4104200, the Major Scientific and Technological Projects of CNOOC under Grant CCL2022RCPS0397RSN, 111 Project under Grant B08028. The authors have no competing interests to declare that are relevant to the content of this article. The datasets generated during the current study are available from the corresponding author on reasonable request.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: Author Chen Liu was employed by the company CNOOC Research Institute Ltd., Beijing. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Wang, C.; Li, G.; Reynolds, A.C. Production optimization in closed-loop reservoir management. *SPE J.* **2009**, *14*, 506–523. [\[CrossRef\]](#)
- Chen, Y.; Oliver, D.S.; Zhang, D.J. Efficient ensemble-based closed-loop production optimization. *SPE J.* **2009**, *14*, 634–645. [\[CrossRef\]](#)
- Su, H.-J.; Oliver, D.S. Engineering, Smart-well production optimization using an ensemble-based method. *SPE Reserv. Eval. Eng.* **2010**, *13*, 884–892. [\[CrossRef\]](#)
- Wang, L.; Yao, Y.; Luo, X.; Adenutsi, C.D.; Zhao, G.; Lai, F. A critical review on intelligent optimization algorithms and surrogate models for conventional and unconventional reservoir production optimization. *Fuel* **2023**, *350*, 128826. [\[CrossRef\]](#)
- Wang, Z.-Z.; Zhang, K.; Chen, G.-D.; Zhang, J.-D.; Wang, W.-D.; Wang, H.-C.; Zhang, L.M.; Yan, X.; Yao, J. Evolutionary-assisted reinforcement learning for reservoir real-time production optimization under uncertainty. *Pet. Sci.* **2023**, *20*, 261–276. [\[CrossRef\]](#)

6. Asadollahi, M.; Naevdal, G. Waterflooding optimization using gradient based methods. In Proceedings of the SPE/EAGE Reservoir Characterization & Simulation Conference, Abu Dhabi, United Arab Emirates, 19–21 October 2009.
7. Forouzanfar, F.; Della Rossa, E.; Russo, R.; Reynolds, A. Life-cycle production optimization of an oil field with an adjoint-based gradient approach. *J. Pet. Sci. Eng.* **2013**, *112*, 351–358. [\[CrossRef\]](#)
8. Liu, X.; Reynolds, A.C. Gradient-based multiobjective optimization with applications to waterflooding optimization. In Proceedings of the ECMOR XIV-14th European Conference on the Mathematics of Oil Recovery, Catania, Italy, 8–11 September 2014.
9. Buktynov, V.; Volkov, O.; Durlinsky, L.J.; Aziz, K. Comprehensive framework for gradient-based optimization in closed-loop reservoir management. *Comput. Geosci.* **2015**, *19*, 877–897. [\[CrossRef\]](#)
10. Sefat, M.H.; Muradov, K.M.; Elsheikh, A.H.; Davies, D.R. Proactive optimization of intelligent-well production using stochastic gradient-based algorithms. *SPE Reserv. Eval. Eng.* **2016**, *19*, 239–252. [\[CrossRef\]](#)
11. Volkov, O.; Bellout, M.C. Gradient-based production optimization with simulation-based economic constraints. *Comput. Geosci.* **2017**, *21*, 1385–1402. [\[CrossRef\]](#)
12. Li, L.; Zhou, F.; Wang, B. Method investigation on intelligent optimization of high dimension HWMHF parameters. *Pet. Sci. Bull.* **2023**, *2023*, 347–359.
13. Mayer, D.G.; Kinghorn, B.; Archer, A.A. Differential evolution—An easy and efficient evolutionary algorithm for model optimisation. *Agric. Syst.* **2005**, *83*, 315–328. [\[CrossRef\]](#)
14. Vesterstrom, J.; Thomsen, R. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753), Padua, Italy, 19–23 June 2004.
15. Mirjalili, S.J. Evolutionary algorithms and neural networks. In *Studies in Computational Intelligence*; Springer: Cham, Switzerland, 2019; Volume 780.
16. Reddy, M.J.; Kumar, D.N.J. Multiobjective differential evolution with application to reservoir system optimization. *J. Comput. Civ. Eng.* **2007**, *21*, 136–146. [\[CrossRef\]](#)
17. Siavashi, M.; Yazdani, M.J. A comparative study of genetic and particle swarm optimization algorithms and their hybrid method in water flooding optimization. *J. Energy Resour. Technol.* **2018**, *140*, 102903. [\[CrossRef\]](#)
18. Gu, J.; Liu, W.; Zhang, K.; Zhai, L.; Zhang, Y.; Chen, F. Reservoir production optimization based on surrogate model and differential evolution algorithm. *J. Pet. Sci. Eng.* **2021**, *205*, 108879. [\[CrossRef\]](#)
19. Zhang, K.; Zhao, X.; Chen, G.; Zhao, M.; Wang, J.; Yao, C.; Sun, H.; Yao, J.; Wang, W.; Zhang, G. A double-model differential evolution for constrained waterflooding production optimization. *J. Pet. Sci. Eng.* **2021**, *207*, 109059. [\[CrossRef\]](#)
20. An, Z.; Zhou, K.; Hou, J.; Wu, D.; Pan, Y. Accelerating reservoir production optimization by combining reservoir engineering method with particle swarm optimization algorithm. *J. Pet. Sci. Eng.* **2022**, *208*, 109692. [\[CrossRef\]](#)
21. Er, M.J.; Wu, S.; Lu, J.; Toh, H.L. Face recognition with radial basis function (RBF) neural networks. *IEEE Trans. Neural Netw.* **2002**, *13*, 697–710.
22. Castro-Neto, M.; Jeong, Y.-S.; Jeong, M.-K.; Han, L.D. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst. Appl.* **2009**, *36*, 6164–6173. [\[CrossRef\]](#)
23. Schulz, E.; Speekenbrink, M.; Krause, A.J. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [\[CrossRef\]](#)
24. Golzari, A.; Sefat, M.H.; Jamshidi, S. Development of an adaptive surrogate model for production optimization. *J. Pet. Sci. Eng.* **2015**, *133*, 677–688. [\[CrossRef\]](#)
25. Aladeitan, Y.M.; Arinkoola, A.O.; Udebhulu, O.D.; Ogbe, D.O. Surrogate modelling approach: A solution to oil rim production optimization. *Cogent Eng.* **2019**, *6*, 1631009. [\[CrossRef\]](#)
26. Chen, G.; Zhang, K.; Xue, X.; Zhang, L.; Yao, J.; Sun, H.; Fan, L.; Yang, Y. Surrogate-assisted evolutionary algorithm with dimensionality reduction method for water flooding production optimization. *J. Pet. Sci. Eng.* **2020**, *185*, 106633. [\[CrossRef\]](#)
27. Zhao, M.; Zhang, K.; Chen, G.; Zhao, X.; Yao, J.; Yao, C.; Zhang, L.; Yang, Y. A classification-based surrogate-assisted multiobjective evolutionary algorithm for production optimization under geological uncertainty. *SPE J.* **2020**, *25*, 2450–2469. [\[CrossRef\]](#)
28. Ogbeiwi, P.; Stephen, K.D.; Arinkoola, A.O. Robust optimisation of water flooding using an experimental design-based surrogate model: A case study of a Niger-Delta oil reservoir. *J. Pet. Sci. Eng.* **2020**, *195*, 107824. [\[CrossRef\]](#)
29. Chen, G.; Zhang, K.; Zhang, L.; Xue, X.; Ji, D.; Yao, C.; Yao, J.; Yang, Y. Global and local surrogate-model-assisted differential evolution for waterflooding production optimization. *SPE J.* **2020**, *25*, 105–118. [\[CrossRef\]](#)
30. Du, S.-Y.; Zhao, X.-G.; Xie, C.-Y.; Zhu, J.-W.; Wang, J.-L.; Yang, J.-S.; Song, H.-Q. Data-driven production optimization using particle swarm algorithm based on the ensemble-learning proxy model. *Pet. Sci.* **2023**, *20*, 2951–2966. [\[CrossRef\]](#)
31. Wang, J.-L.; Zhang, L.-M.; Zhang, K.; Zhou, J.-P.; Peng, W.-F.; Yin, F.-L.; Zhong, C.; Yan, X.; Liu, P.-Y.; Zhang, H.-Q.; et al. Multi-surrogate framework with an adaptive selection mechanism for production optimization. *Pet. Sci.* **2024**, *21*, 366–383. [\[CrossRef\]](#)
32. Wang, J.; Zhang, K.; Zhang, L.; Peng, W.; Yan, X.; Wang, H.; Zhang, H.; Yang, Y.; Sun, H.; Liu, P.; et al. Adaptive Basis Function Selection Enhanced Multisurrogate-Assisted Evolutionary Algorithm for Production Optimization. *SPE J.* **2023**, *28*, 3192–3209. [\[CrossRef\]](#)
33. Yin, F.; Xue, X.; Zhang, C.; Zhang, K.; Han, J.; Liu, B.; Wang, J.; Yao, J. Multifidelity genetic transfer: An efficient framework for production optimization. *SPE J.* **2021**, *26*, 1614–1635. [\[CrossRef\]](#)

34. Viana, F.A.; Haftka, R.T.; Watson, L.T. Efficient global optimization algorithm assisted by multiple surrogate techniques. *J. Glob. Optim.* **2013**, *56*, 669–689. [[CrossRef](#)]
35. Agada, S.; Geiger, S.; Elsheikh, A.; Oladyshkin, S. Data-driven surrogates for rapid simulation and optimization of WAG injection in fractured carbonate reservoirs. *Pet. Sci.* **2017**, *23*, 270–283. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.