

Original Paper

Multi-surrogate framework with an adaptive selection mechanism for production optimization

Jia-Lin Wang^a, Li-Ming Zhang^{a, *}, Kai Zhang^{a, b}, Jian Wang^c, Jian-Ping Zhou^d,
Wen-Feng Peng^e, Fa-Liang Yin^a, Chao Zhong^a, Xia Yan^a, Pi-Yang Liu^b, Hua-Qing Zhang^c,
Yong-Fei Yang^a, Hai Sun^a

^a School of Petroleum Engineering, China University of Petroleum (East China), Qingdao, 266580, Shandong, China

^b Qingdao University of Technology, Qingdao, 266580, Shandong, China

^c College of Science, China University of Petroleum, Qingdao, 266580, Shandong, China

^d Zepu Oil and Gas Development Department, PetroChina Tarim Oilfield Company, Korla, 841000, Xinjiang, China

^e Hainan Branch of CNOOC Ltd, Haikou, 70311, Hainan, China

ARTICLE INFO

Article history:

Received 6 February 2023

Received in revised form

24 August 2023

Accepted 24 August 2023

Available online 26 August 2023

Edited by Yan-Hua Sun

Keywords:

Production optimization

Multi-surrogate models

Multi-evolutionary algorithms

Dimension reduction

Broad learning system

ABSTRACT

Data-driven surrogate models that assist with efficient evolutionary algorithms to find the optimal development scheme have been widely used to solve reservoir production optimization problems. However, existing research suggests that the effectiveness of a surrogate model can vary depending on the complexity of the design problem. A surrogate model that has demonstrated success in one scenario may not perform as well in others. In the absence of prior knowledge, finding a promising surrogate model that performs well for an unknown reservoir is challenging. Moreover, the optimization process often relies on a single evolutionary algorithm, which can yield varying results across different cases. To address these limitations, this paper introduces a novel approach called the multi-surrogate framework with an adaptive selection mechanism (MSFASM) to tackle production optimization problems. MSFASM consists of two stages. In the first stage, a reduced-dimensional broad learning system (BLS) is used to adaptively select the evolutionary algorithm with the best performance during the current optimization period. In the second stage, the multi-objective algorithm, non-dominated sorting genetic algorithm II (NSGA-II), is used as an optimizer to find a set of Pareto solutions with good performance on multiple surrogate models. A novel optimal point criterion is utilized in this stage to select the Pareto solutions, thereby obtaining the desired development schemes without increasing the computational load of the numerical simulator. The two stages are combined using sequential transfer learning. From the two most important perspectives of an evolutionary algorithm and a surrogate model, the proposed method improves adaptability to optimization problems of various reservoir types. To verify the effectiveness of the proposed method, four 100-dimensional benchmark functions and two reservoir models are tested, and the results are compared with those obtained by six other surrogate-model-based methods. The results demonstrate that our approach can obtain the maximum net present value (NPV) of the target production optimization problems.

© 2023 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

With the decrease in the number of newly discovered oil fields and the output of currently exploited oil fields, research is now focused on how to utilize existing oil fields to maximize the

development of reservoir potential. Production optimization (Oliveira and Reynolds, 2013; Isebor and Durlofsky, 2014; Gu et al., 2021; An et al., 2022; Xue et al., 2022; Zhong et al., 2022) is a crucial component of closed-loop reservoir management that can significantly increase production efficiency (Foss and Jensen, 2011; Hou et al., 2015; Mirzaei-Paiaman et al., 2021). Production optimization aims to find the optimal development scheme that can fully develop the reservoir potential and achieve the maximum

* Corresponding author.

E-mail address: zhangliming@upc.edu.cn (L.-M. Zhang).

Nomenclature

DB	Database
db	The set of samples to train the surrogate models
P	Excellent parent population in the database
C	Offspring population after evolutionary operator update
BLS	Broad learning system
NSGA-II	Non-dominated sorting genetic algorithm II
NPV	Net present value
SAEAs	Surrogate-assisted evolutionary algorithms
DE	Differential evolution
GWO	Grey wolf optimizer
PSO	Particle swarm optimization
PRG	Polynomial regression model
RBF	Radial basis function
MSBN	Multi-surrogate stage based on NSGA-II
ASSSME	Adaptive selection stage of a single surrogate with the multi-evolutionary algorithms

economic benefit (usually measured in NPV). The popular approaches used for reservoir production optimization can be roughly split into two groups: gradient-based methods and gradient-free methods. The gradient-based methods use the adjoint strategy (Forouzanfar et al., 2013; Zhang et al., 2014; Volkov and Bellout, 2017) to determine the gradient of the objective function. These methods have high computational efficiency, but obtaining information about the gradient is challenging, and it is prone to reaching the local optimal solutions during the optimization process. Approximate gradient methods (Chen et al., 2017; Chen and Xu, 2019) and evolutionary algorithms are two categories of gradient-free techniques. Among them, evolutionary algorithms such as differential evolution (DE) (Das and Suganthan, 2010), particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), and the grey wolf optimizer (GWO) (Mirjalili et al., 2014) have been popularly applied in production optimization in recent years.

Most of the evolutionary algorithms have efficient global searchability, allowing them to generate numerous new development schemes to ensure solution diversity. If all of these schemes are evaluated by the numerical simulator, the computational process is quite time-consuming, resulting in the so-called expensive production optimization problems (Chen et al., 2020a). To address this issue, different kinds of approaches using surrogate models have been developed in the evaluation process (Jin et al., 2019; Chen et al., 2020b; Zhao M. et al., 2020a; Zhao M. et al., 2020b; Zhao X. et al., 2020; Ma et al., 2022a; Ma et al., 2022b). The production regimes are fed into the surrogate models to quickly produce reservoir prediction results, such as NPV, without implementing the complex simulations. Surrogate models are mainly categorized into physics-based (Zhao et al., 2016; Ren et al., 2019; Wang et al., 2021) and data-driven-based cases. Data-driven surrogate models mainly include the Kriging model (Güygüler et al., 2002), the radial basis function (Luo et al., 2011), polynomial regression (Ostertagová, 2012), and so on. This paper focuses on data-driven surrogate models.

Instead of an actual numerical simulator, a surrogate model can save time. However, in practice, the optimization procedure uses just one surrogate model, which may not be applicable to all reservoir optimization problems as it lacks physical significance (refer to Section 3.1). The objective of this paper is to enhance

adaptability to different scenarios by incorporating multiple surrogate models, thus addressing the limitations of using a single surrogate model. The best solution cannot be obtained by simply superimposing multiple surrogate models without considering their relationship (Zerpa et al., 2005). Thus, a more appropriate approach is to use a multi-objective algorithm with different surrogate models as the objective functions to obtain Pareto solutions suitable for them (Dong et al., 2021). However, if the multi-objective algorithm's input population is poor, getting the optimal development scheme will be challenging during the subsequent optimization procedure. Moreover, there is not only one Pareto solution, and it's worth considering how to choose what we need. It's also crucial to note that the ability of evolutionary algorithms to adapt to different problems can vary (refer to Section 3.1). Furthermore, production optimization is a continuous process, and the performance of an evolutionary algorithm differs between the early and late stages of optimization (Li et al., 2022). In other words, different algorithms possess varying exploration and exploitation capabilities. By employing multiple evolutionary algorithms in conjunction with various surrogate models, adaptability to a wide range of issues can be enhanced.

This paper suggests a multi-surrogate framework with an adaptive selection mechanism (MSFASM) to address the above-mentioned issues. Two stages make up MSFASM. First, two surrogate models are used to assist the three evolutionary algorithms in updating the population and pre-screening the optimal solutions in each iteration process. Broad learning system (BLS) (Chen and Liu, 2018) is used to select the best one from the pre-screened optimal solutions of these surrogate models. The establishment of a multi-surrogate framework based on NSGA-II (Deb et al., 2000) occurs in the second stage. The two surrogate models mentioned above serve as the objective functions, and NSGA-II searches for a set of Pareto solutions that are appropriate for them. Then, an optimal point criterion is proposed to select Pareto solutions and input them into the database together with the optimal solution in the first stage. Repeat these two steps until the stop criteria is reached. The rest of this article is organized as follows. The production optimization problems are introduced in Section 2, and the surrogate-assisted evolutionary algorithms (SAEAs), BLS, and NSGA-II are introduced in Section 3. Section 4 contains details about the proposed method, MSFASM. In Section 5, the experimental results of four 100-dimensional benchmark functions and two reservoir models are given. Finally, Section 6 summarizes this paper.

2. Problem statement

2.1. Production optimization model

The objective of the reservoir production optimization is to find the optimal production control variable x^d , where d is the dimension of the variable, $d = w \times p$, w is the number of wells, and p is the number of time steps. We choose a net present value (NPV) as the objective function, which can be defined as follows:

$$f(\mathbf{x}, \mathbf{z}) = \text{NPV}(\mathbf{x}, \mathbf{z}) = \sum_{t=1}^p \Delta t \frac{Q_{o,t} \cdot r_o - Q_{w,t} \cdot r_w - Q_{i,t} \cdot r_i}{(1+b)} \quad (1)$$

where $\mathbf{z}$ is the state variable output by reservoir numerical simulation calculation; $Q_{o,t}$, $Q_{w,t}$, and $Q_{i,t}$ are the oil production rate, water production rate, and water injection rate at a time step t , respectively; r_o , r_w , and r_i are oil production revenue, water removal

cost, and water injection cost, respectively; b is the discount rate. Therefore, the water flooding production optimization can be defined in the following form:

$$\max_{\mathbf{x}} f(\mathbf{x}, \mathbf{z}), \quad \mathbf{x} \in \mathbb{R}^d, \quad \text{s.t. } \mathbf{z} = g(\mathbf{x}), \quad \mathbf{x}_{\text{low}} \leq \mathbf{x} \leq \mathbf{x}_{\text{up}} \quad (2)$$

where $\mathbf{x}_{\text{low}}$ and $\mathbf{x}_{\text{up}}$ are the lower and upper bounds of the control variable $\mathbf{x}$, respectively.

3. Related works

3.1. Surrogate-assisted evolutionary algorithms

Using evolutionary algorithms to generate a substantial number of promising candidate development schemes has gained popularity in the field. However, the optimization process often involves extensive numerical simulation calculations, which can be time-consuming. To mitigate this issue, data-driven surrogate models trained on samples are employed to replace the reservoir numerical simulators for evaluating the fitness (i.e., NPV) of the development schemes. These surrogate-assisted evolutionary algorithms (SAEAs) have proven to be effective in identifying optimal development schemes while reducing the overall time cost. To provide a clearer understanding, this paper introduces several definitions of SAEAs in the context of production optimization problems:

- **Sample:** A sample contains a development scheme and its corresponding NPV.
- **Database:** The database consists of all development schemes and their NPV (i.e., all samples).
- **Training:** Since the surrogate is driven by data, the surrogate model is constructed with the development schemes as inputs and the NPV as outputs during the training process.
- **Initial population:** The initial population is the set of development schemes that are input into the evolutionary algorithms. The initial population is usually selected from the database based on NPV.
- **Generation of offspring:** Generation of offspring is the process in which an evolutionary algorithm uses operators to generate new development schemes based on the initial population.
- **Selection:** Selection is the process of calculating the fitness between the initial population and the offspring generated by the evolutionary algorithms and selecting the better of the two.

3.1.1. Differential evolution

Differential evolution is an efficient global optimization algorithm widely used in engineering problems. The initial population of the differential evolution can be viewed as $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^n$, where n is the number of population, $\mathbf{x}_i$ can be expressed as $\mathbf{x}_i = \{x_i^1, x_i^2, x_i^3, \dots, x_i^d\}$, and d is the dimension of $\mathbf{x}_i$. DE consists of three main components: mutation, crossover, and selection. The mutation strategy selected in this paper can be expressed as follows:

$$\mathbf{v}_i = \mathbf{x}_{\text{best}} + Mu(\mathbf{x}_{r1} - \mathbf{x}_{r2}) \quad (3)$$

where $\mathbf{v}_i$ is the i th mutant vector; $\mathbf{x}_{\text{best}}$ is the best development scheme at present; Mu is a scaling factor; $r_1, r_2 \in [1, n]$ and they are randomly generated integers that differ from each other. The crossover strategy is as follows:

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

where u_i^j denotes the j th variable of the i th trial vector; $U_j(0, 1)$ is a uniformly distributed random number between 0 and 1; CR is the crossover rate; and j_{rand} is a random index to prevent u_i from being the same as x_i . The selection strategy is as follows:

$$x_i^* = \begin{cases} u_i, & \text{if } f(u_i) < f(x_i) \\ x_i, & \text{otherwise} \end{cases} \quad (5)$$

where x_i^* is the i th development scheme for the next generation; f is an objective function (i.e., a numerical simulator), and to reduce the time cost of using it, the surrogate models described next are used instead. To save space, the other two evolutionary algorithms (PSO and GWO) used in this paper are shown in Appendix A.

3.1.2. Polynomial regression model

The polynomial regression model is a simple and practical surrogate model. Polynomial regression analysis is used to study the quantitative relationship between the objective function and the observation variable. The polynomial regression used in this paper is a second-order polynomial model, and its mathematical expression is as follows:

$$\hat{f}(x) = \beta_0 + \sum_{i=1}^d \beta_i x_i + \sum_{i=1}^d \beta_{ii} x_i^2 + \sum_{i=1}^{d-1} \sum_{j=i+1}^d \beta_{ij} x_i x_j \quad (6)$$

where $\hat{f}(x)$ is the PRG prediction; x is the observation variable; $\beta = \{\beta_0, \beta_i, \beta_{ii}, \beta_{ij}\}$ and are polynomial coefficients that can be calculated by the least squares method (Dong et al., 2021); d denotes the dimension of the observation variable.

3.1.3. Radial basis function

The radial basis function is more suitable for high-dimensional problems than other surrogate models, so it is widely used for expensive optimization and engineering problems. Its expression is as follows:

$$\hat{f}(x) = \varepsilon^T \psi = \sum_{i=1}^n \varepsilon_i \psi(\|\mathbf{x} - \mathbf{s}^{(i)}\|) \quad (7)$$

where $\hat{f}(x)$ is the RBF prediction; x is the input variable; ψ represents the radial basis function, and the types are linear, cubic, gaussian, etc. In this paper, we use the cubic type, and its expression is shown as follows:

$$\psi(\|\mathbf{x} - \mathbf{s}^{(i)}\|) = (\|\mathbf{x} - \mathbf{s}^{(i)}\|)^3 \quad (8)$$

where $\mathbf{s}^{(i)}$ represents the i th center point equivalent to the known samples, all of the centers can be represented as $\mathbf{S} = (s^{(1)}, s^{(2)}, s^{(3)}, \dots, s^{(n)})^T$, where n is the number of sample centers and ε is the weight vector. The other detailed information about the RBF can be found in Yu et al. (2018).

3.1.4. The overall process of SAEAs

In Algorithm 1, we give the pseudo-code of SAEAs.

Algorithm 1. The framework of SAEAs (Viana, 2016).

```

01: Initialization: Generate  $N$  development schemes  $\{x_i\}_{i=1}^N$  using Latin hypercube sampling (LHS) (Viana,
    2016) in the whole searching space and then evaluate these points with the numerical simulator to get the
    corresponding NPV,  $\{y_i\}_{i=1}^N$ ; Archive the samples into database,  $DB$ ;
02: While the stopping criterion is not reached
03:     Sort  $DB$  according to the NPV and select the top samples to form a sample set,  $db$ ;
04:     Construct a surrogate model  $\hat{f}(x)$  (e.g., PRG) using  $db$ ;
05:     Select excellent development schemes from  $db$  to form a set  $P$  as the initial population input of the
    evolutionary algorithm (e.g., DE) to generate a new offspring population,  $C$ ;
06:     Estimate  $C$  using  $\hat{f}(x)$  to select the solution  $x_b$  with the best-predicted value.
07:     Evaluate  $x_b$  with the numerical simulator to get the NPV,  $y_b$ ;
08:     Store the new sample  $(x_b, y_b)$  in  $DB$ ;
09: End while
10: Output the best solution in  $DB$ .
  
```

In the above framework, the stopping criterion is whether the maximum number of calculations by the simulator reaches the specified value. In step 4, some samples are selected in the database based on the NPV to form a set $db = \{x_i, y_i\}_{i=1}^n$ to construct PRG or RBF model. In step 5, we choose a better part of the development schemes from db according to the NPV to form the initial population $P = \{x_i\}_{i=1}^{n^*}$ ($n^* \leq n$), which is input to the evolutionary algorithm to generate the offspring population C . The solution x_b with the best-predicted value among the population C is selected in step 6 using the surrogate model built in step 4. The NPV y_b of x_b is calculated by the numerical simulator. Then, the new sample (x_b, y_b) is added to the database.

The reason for using multiple evolutionary algorithms is that each one has its own adaptation problem. Furthermore, the performance of an evolutionary algorithm can vary between the early and late stages of optimization. To prove the above theory, we optimize three different problems using DE, GWO, and PSO, respectively. In Fig. 1(a)–(c), the 100-dimensional Ackley and Ellipsoid benchmark functions (minimizing optimization problems) and the 65-dimensional (i.e., d in Section 2.1) three-channel reservoir model (maximizing optimization problem) are optimized,

respectively. It is observed that GWO outperforms the other two algorithms on the Ackley function, PSO performs the best on the Ellipsoid function, and DE achieves the optimal result on the three-channel reservoir model. This demonstrates that no single evolutionary algorithm can consistently attain optimal solutions across all problems. Fig. 1(a) also shows that although PSO has the worst final optimization effect, it declines the fastest in the early stage, indicating a tremendous early exploration capability. In contrast, GWO and DE have strong later exploitation capabilities. Balancing exploration and exploitation throughout the optimization process can be achieved by utilizing various evolutionary algorithms. Based on these observations, this paper incorporates three evolutionary algorithms to tackle reservoir production optimization problems, ensuring a comprehensive approach that accounts for different problem characteristics.

Similar to evolutionary algorithms, the combination of multiple surrogate models is motivated by the fact that no single surrogate model can effectively adapt to all problems. As shown in Fig. 2(a) and (b), the DE algorithm is assisted by RBF and PRG, respectively, to optimize two different issues: the 100-dimensional Levy benchmark function (minimizing optimization problem) and the 72-dimensional PUNQS3 reservoir model (maximizing optimization

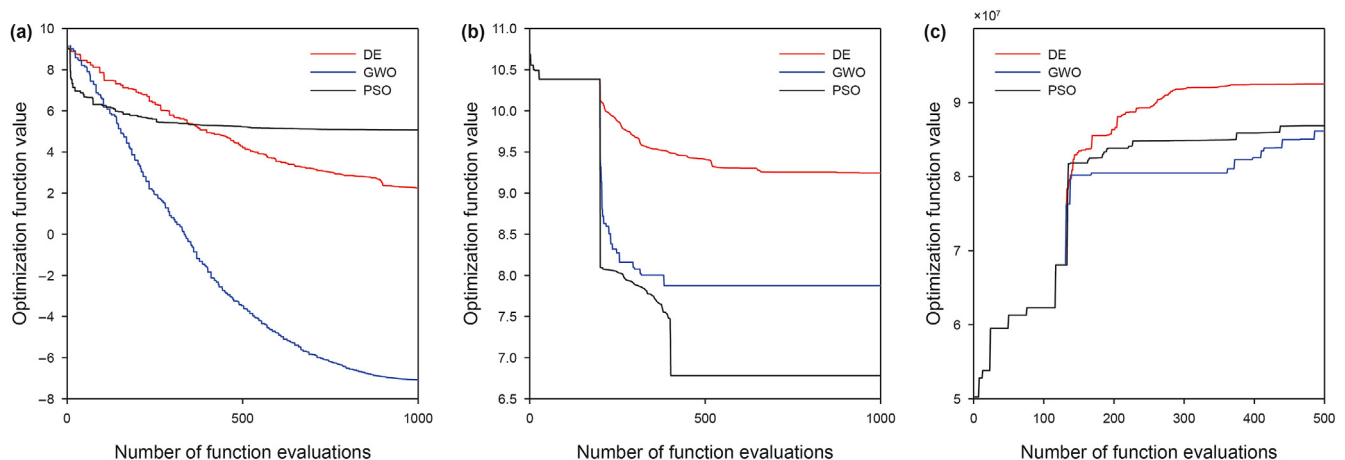


Fig. 1. Illustration of the reason for using multiple evolutionary algorithms. (a) Ackley function optimization curve; (b) Ellipsoid function optimization curve; (c) Three-channel model optimization curve.

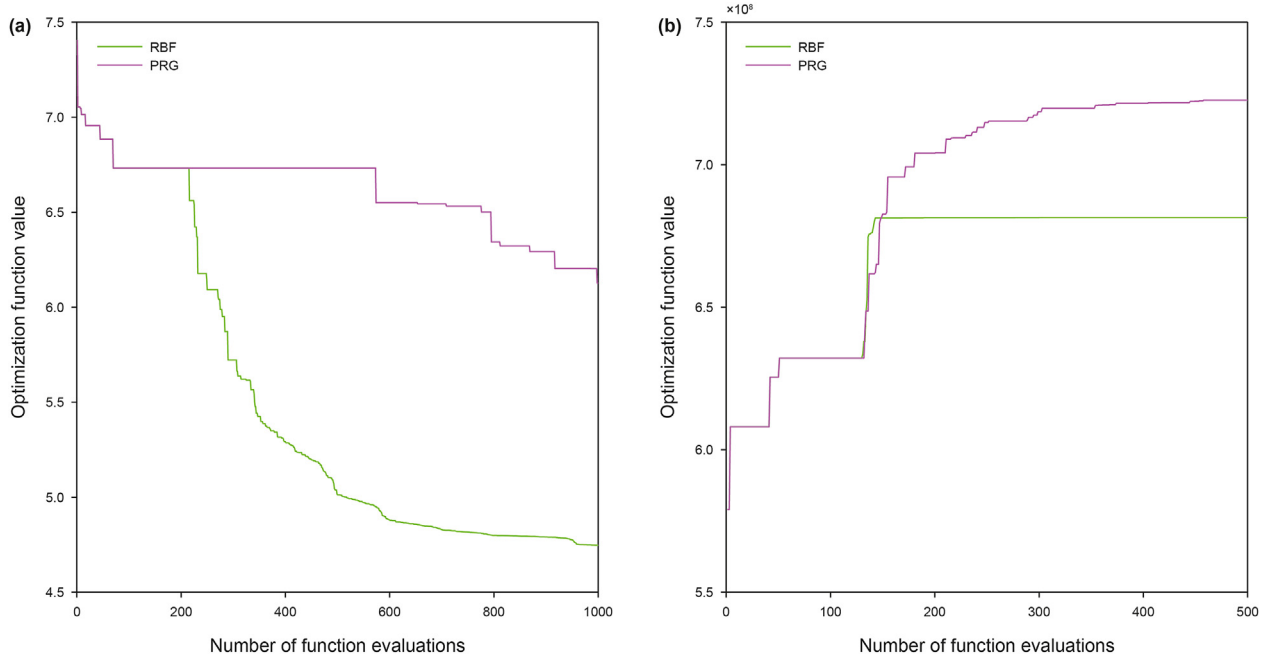


Fig. 2. Illustration of the reason for using multiple surrogate models. (a) Levy function optimization curve; (b) PUNQ-S3 model optimization curve.

problem). It is observed that RBF performs better in optimizing the Levy function, while PRG yields superior optimization results for the PUNQS3 reservoir model. These surrogate models exhibit different optimization performances across various problems. When it comes to production optimization problems, the surrogate models' capability to accurately represent the actual reservoir is constrained due to the limited number of available samples for model construction. As a result, it becomes challenging to determine which surrogate model will produce accurate reservoir prediction results. Consequently, the utilization of multiple surrogate

models becomes essential to enhance adaptability to a wide range of problems.

3.2. Broad learning system

The broad learning system is an efficient algorithm that does not have many parameters and machine resources to optimize compared to deep learning. Based on the above advantages, BLS is suitable for evaluating development schemes in production optimization problems. In essence, we also treat BLS as a surrogate model, but it is not combined with evolutionary algorithms for optimization, so we introduce it separately. BLS consists of input, a feature mapping layer, an enhancement layer, and output. The training process for BLS is described as follows: we input the data set $X^{N \times D}$, which contains N samples in D dimensions; Y is the output matrix corresponding to X . BLS first maps the input to construct a set of feature mappings. For n feature mappings, the following formula can be expressed:

$$Z_i = \varphi_i(XW_{ei} + \beta_{ei}), \quad i = 1, \dots, n \quad (9)$$

where W_{ei} and β_{ei} are random weights; φ_i is a random mapping; Z_i is the i th feature mapping, the feature mapping layer can be represented as $Z^n = [Z_1, \dots, Z_n]$. The feature mapping layer is then used to build the enhancement layer. The m th enhanced nodes can be expressed as:

$$H_m \equiv \xi_m(Z^n W_{hm} + \beta_{hm}) \quad (10)$$

where W_{hm} , β_{hm} and ξ_m are defined similarly to W_{ei} , β_{ei} and φ_i . The enhancement layer can be represented as $H^m = [H_1, \dots, H_m]$. Therefore, the generalized model can be expressed as:

$$\begin{aligned} Y &= [Z_1, \dots, Z_n | \xi_1(Z^n W_{h1} + \beta_{h1}), \dots, \xi_m(Z^n W_{hm} + \beta_{hm})] W^m \\ &= [Z_1, \dots, Z_n | H_1, \dots, H_m] W^m = [Z^n | H^m] W^m \end{aligned} \quad (11)$$

where $W^m = [Z^n | H^m]^+ Y$ is the connection weight of the broad

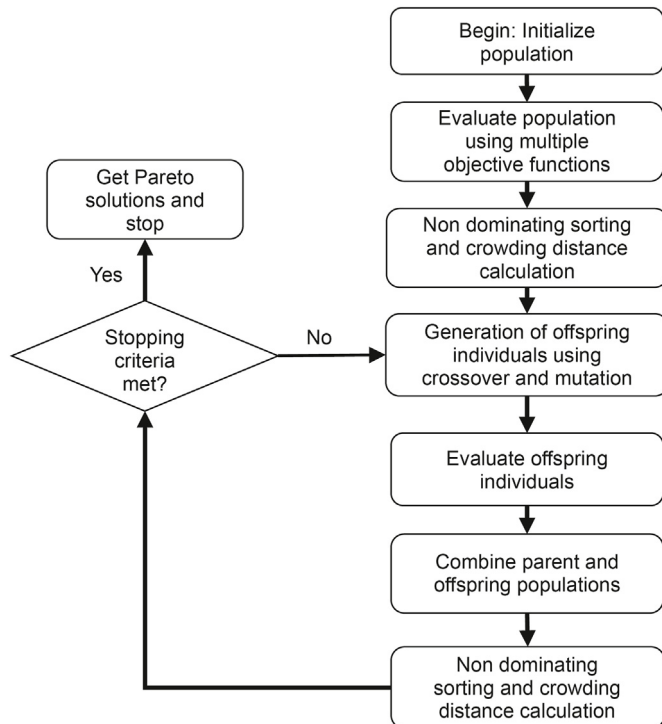


Fig. 3. The flow chart of the NSGA-II.

structure, which can be easily obtained by approximating the ridge regression $[Z^n | H^m]^+$ (Chen and Liu, 2018) when the output Y is determined.

3.3. Non-dominated sorting genetic algorithm II

NSGA-II is a fast, non-dominated multi-objective optimization algorithm with an elite retention strategy (Deb et al., 2000). We use NSGA-II to connect multiple surrogate models to build an efficient multi-surrogate framework. The NSGA-II is developed based on the Pareto dominance relationship. For a maximized multi-objective optimization problem, there are n_f objective functions $F_i(x)$, $i = 1, \dots, n_f$. Two arbitrary solutions x_a and x_b are given, x_a which is said to dominate x_b if the following two conditions are true:

$$\begin{aligned} 1. & F_i(x_b) \leq F_i(x_a) \forall i \in 1, \dots, n_f \\ 2. & F_i(x_b) < F_i(x_a) \exists i \in 1, \dots, n_f \end{aligned} \quad (12)$$

A solution is called a non-dominant solution (i.e., Pareto solution) if no other can dominate it. The Pareto rank of the non-dominated solutions is defined as 1. The flow chart of NSGA-II (Al-Aghbari and Gujarathi, 2022) is shown in Fig. 3. After generating the initial population, it is evaluated by multiple objective functions. A fast non-dominant sorting is used to determine the Pareto rank of each individual according to a process similar to Eq. (12). The crowding distance is calculated to ensure that the solutions have good space-filling characteristics. The larger the crowding distance, the better the space-filling characteristics of the individuals. Details of the fast non-dominated sorting and crowding distance calculation can be found in Farahi et al. (2021). Individuals with lower Pareto rank and higher crowding distance are selected to form an elite parent, and offspring individuals are generated by crossover and mutation (Whitley, 1994). After the objective function values of the offspring individuals are calculated, the parent and offspring populations are then combined for the above procedure until the stopping criterion is satisfied. This paper sets the stopping criterion for iterating for 100 generations. Finally, a set of Pareto solutions is output.

4. Proposed method

4.1. Adaptive selection stage of a single surrogate with the multi-evolutionary algorithms

As verified in the above section, different evolutionary

algorithms are suitable for different types of problems. Only one evolutionary algorithm is used in canonical SAEAs. However, production optimization is a complex problem, and each reservoir is unique. Meanwhile, certain evolutionary algorithms excel at exploration, while others demonstrate strong exploitation capabilities. They are suitable for different periods of the optimization process. If one evolutionary algorithm is used to solve the production optimization problems for all reservoir types and periods, optimal results cannot be obtained. Therefore, we propose an adaptive selection stage of a single surrogate with the multi-evolutionary algorithms (ASSSME) to solve the above problem. The term "single surrogate" does not mean that we only use one surrogate model, but that there is no connection between them. They simply act as evaluators of fitness values (i.e., NPV) during the update of the populations by the evolutionary algorithms, which also highlights the difference with Section 4.2.

Algorithm 2 describes the details of ASSSME. The first step, as outlined in Algorithm 1, involves extracting a sample set, denoted as db , from the database DB . This sample set is then utilized to construct PRG and RBF models, respectively. In step 4 of Algorithm 2, the set of development schemes, denoted as P , is provided as the initial population for the three optimizers: DE, GWO, and PSO. Instead of using numerical simulators, the evolutionary algorithms are assisted by the PRG and RBF surrogate models for population renewal. The utilization of two surrogate models serves two purposes. Firstly, as mentioned in Section 4.2, surrogate models are required and can be constructed in advance. Secondly, as indicated by Eq. (5), the evolutionary algorithms employed in this paper rely on the fitness values of offspring to determine their subsequent evolutionary direction. However, the prediction bias of surrogate models may lead to the population evolving in an unfavorable direction. By employing both PRG and RBF surrogate models, the prediction uncertainty can be reduced, increasing the likelihood of obtaining superior offspring. Consequently, PRG and RBF are independently integrated with the three evolutionary algorithms in each generation's population evolution. The discrepancies in their predicted values contribute to diverse directions of population evolution. As a result, each evolutionary algorithm will produce two offspring populations. This means that after K generations, a total of six optimized populations $\{C_i\}_{i=1}^6$ will be obtained.

Algorithm 2. The framework of ASSSME

```

01: Input: Database,  $DB$ ; The generation number,  $K$ ;
02: While the stopping criterion is not reached
03:     Train PRG ( $\hat{f}_1(x)$ ) and RBF ( $\hat{f}_2(x)$ ) models by extracting the sample set  $db$  from  $DB$ ;
04:     Select the development scheme set  $P$  from  $db$  to input into DE, GWO, and PSO;  $gen = 0$ ;
05:     while  $gen < K$ 
06:         Generate new generation of populations by using three evolutionary algorithms with the assistance
           of PRG and RBF;
07:          $gen = gen + 1$ ;
08:     end while
09:     Get the updated population  $\{C_i\}_{i=1}^6$ , find six suspected optimal solutions  $\{c_{bi}\}_{i=1}^6$  by the corresponding
       surrogate models;
10:     Reduce the dimension of the development schemes in  $DB$  and  $\{c_{bi}\}_{i=1}^6$  by using the Sammon mapping.
11:     Construct a BLS model ( $\hat{f}_3(x)$ ) using the development schemes after dimension reduction in  $DB$  and
       their corresponding NPV;
12:     Determine  $c_{best}$ , which is the optimal solution in  $\{c_{bi}\}_{i=1}^6$  by using ( $\hat{f}_3(x)$ );
13:     Evaluate  $c_{best}$  with the numerical simulator and store the new sample into  $DB$ ;
14: End while
15: Output

```

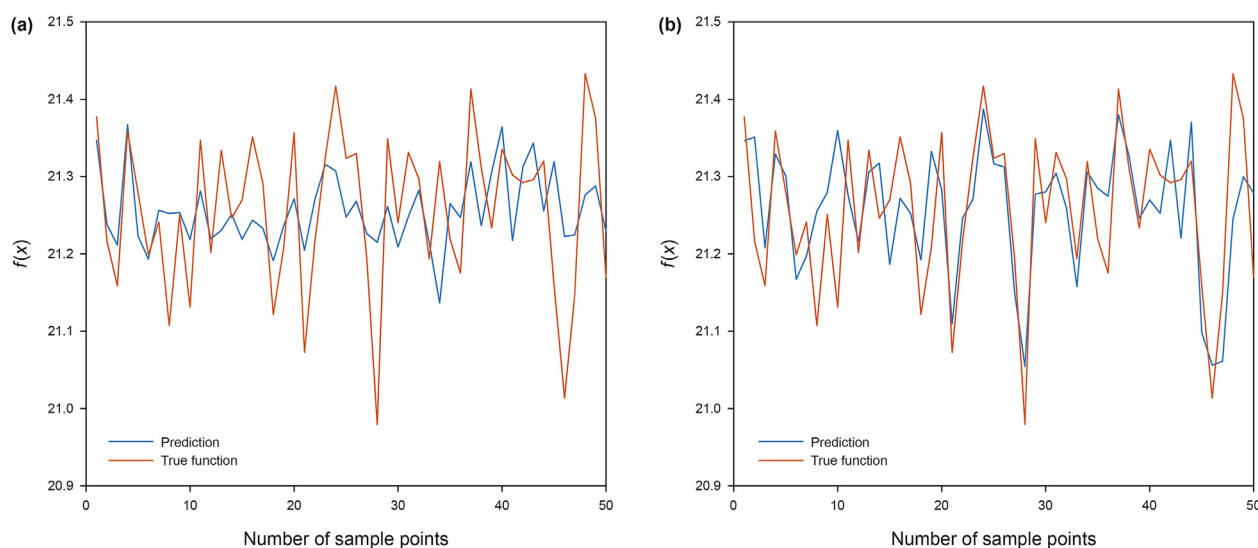


Fig. 4. BLS test accuracy of different data processing methods. (a) Test result for BLS without Sammon mapping; (b) Test result for BLS with Sammon mapping.

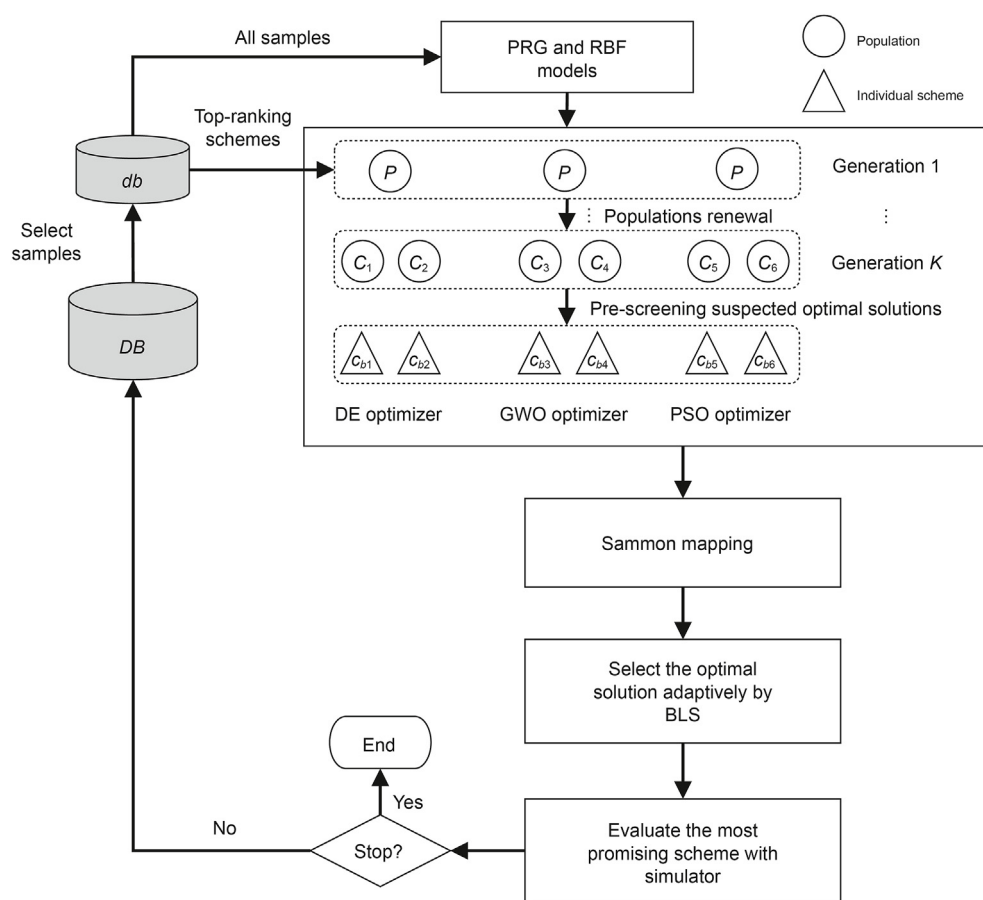


Fig. 5. Schematic of ASSSME.

In step 9, similar to step 6 of Algorithm 1, each population will be pre-screened by its corresponding surrogate model, and six suspected optimal solutions $\{c_{bi}\}_{i=1}^6$ will be obtained. The above process is illustrated with a concrete example. Take GWO and RBF, for instance. RBF assists GWO in updating the K generation to produce

the optimized population, C_1 . From this optimized population, the individual c_{b1} with the best fitness value, as predicted by RBF, is selected. Repeat the above steps for the three evolutionary algorithms and the two surrogate models to obtain the previous results $\{C_i\}_{i=1}^6$ and $\{c_{bi}\}_{i=1}^6$. What we seek is the individual with the highest genuine NPV in $\{c_{bi}\}_{i=1}^6$. However, as we are unsure now,

we refer $\{c_{bi}\}_{i=1}^6$ to the suspected optimal solutions. We cannot evaluate them directly with the numerical simulator because the simulator is too computationally intensive in each loop. We first reduce the dimensionality of the samples for training the BLS model and then use it to determine the best development scheme in $\{c_{bi}\}_{i=1}^6$. To our knowledge, this is the first time BLS has been applied to reservoir production optimization problems.

In the framework above, steps 10, 11, and 12 show how the best development scheme in $\{c_{bi}\}_{i=1}^6$ is selected. First, the Sammon mapping (Sammon, 1969) is used to reduce the dimensions of DB and $\{c_{bi}\}_{i=1}^6$. The goal of the Sammon mapping is to minimize the following error function:

$$\begin{aligned} \operatorname{argmin}_y \frac{1}{\sum_{i < j} \operatorname{Dis}_{mn}} \sum_{m < n} \frac{(\operatorname{Dis}_{mn} - \operatorname{dis}_{mn})^2}{\operatorname{Dis}_{mn}} \\ = \frac{1}{\sum_{m < n} \|x_m - x_n\|} \sum_{i < j} \frac{(\|x_m - x_n\| - \|x_{lm} - x_{ln}\|)^2}{\|x_m - x_n\|} \end{aligned} \quad (13)$$

where x_m and x_n are the m th and n th development schemes in DB , respectively; x_{lm} and x_{ln} are the corresponding variables of x_m and x_n in the dimensional reduction projection space, respectively; Dis_{mn} is the distance between x_m and x_n ; dis_{mn} is the distance between x_{lm} and x_{ln} . Due to the high dimensionality of production optimization problems, dimensionality reduction is carried out to improve the accuracy of BLS. After Sammon mapping, the test accuracy of BLS is higher than that of Kriging after dimensionality reduction. For the 100-dimensional Ackley function, 200 sample points are used to train BLS, and 50 sample points are tested. As can be seen from Fig. 4, after Sammon mapping processing, the test accuracy of BLS is improved compared with before.

After dimensionality reduction, all development schemes in DB , together with NPV, are used to construct the BLS model, $\hat{f}_3(\mathbf{x})$. The following formulas are used to determine the actual best one among the suspected optimal solutions:

$$\begin{cases} \hat{y}_{c_{b1}} = \hat{f}_3(c_{b1}) \\ \hat{y}_{c_{b2}} = \hat{f}_3(c_{b2}) \\ \vdots \\ \hat{y}_{c_{b6}} = \hat{f}_3(c_{b6}) \end{cases} \quad (14)$$

where $\{\hat{y}_{c_{b1}}, \dots, \hat{y}_{c_{b6}}\}$ are the BLS model predicted values of $\{c_{bi}\}_{i=1}^6$ after dimensional reduction. The development scheme with the largest NPV is considered the best for production optimization. Therefore, BLS is used to adaptively select the one with the largest predicted value in $\{c_{bi}\}_{i=1}^6$, define it as c_{best} , calculate its real NPV through the numerical simulator, and then add the new sample to

the DB . The population that contains c_{best} in the $\{C_i\}_{i=1}^6$ is designated as C_{best} , and it is temporarily stored as part of the initial population of Section 4.2. The schematic diagram of ASSSME is shown in Fig. 5. The circle represents the population (i.e., the set of development schemes), and the triangle represents the individual (i.e., the development scheme). Within the ASSSME framework, multiple evolutionary algorithms are employed to update the child individuals. Despite the involvement of surrogate models in population updates, each evolutionary algorithm is assisted by the same PRG and RBF models. As a result, the disparities observed in the final populations can be attributed to the differences among the evolutionary algorithms themselves. Furthermore, the reduced-dimensional broad learning system (BLS) model is utilized to determine the optimal solution. This stage involves adaptively selecting the evolutionary algorithm that best aligns with the current optimization period, thereby enhancing its adaptability to the specific problem at hand. Moreover, ASSSME generates an excellent initial population (i.e., C_{best}), which serves as a foundation for the subsequent multi-surrogate stage.

4.2. Multi-surrogate stage based on NSGA-II

Due to the inherent complexity of reservoir blocks, it is evident that no single surrogate model can effectively address every issue. This observation aligns with the previous discussion on the adaptability of evolutionary algorithms. To account for the diverse challenges posed by different reservoir types, the utilization of multiple surrogate models becomes crucial. In order to establish a robust relationship between these surrogate models, we propose the implementation of a multi-objective algorithm. This approach enables us to leverage the capabilities of various surrogate models and build a comprehensive understanding of the problem at hand. Consequently, we introduce a multi-surrogate stage based on NSGA-II (MSBN), where the PRG and RBF surrogate models are employed as the optimized objective functions. By employing a multi-objective optimization framework, we can effectively tackle the complexities associated with multiple surrogate models. The formulation of the multi-objective optimization problem considering these surrogate models is defined as follows:

$$\begin{aligned} \operatorname{Max} \{ \hat{f}_1(\mathbf{x}), \hat{f}_2(\mathbf{x}) \} \\ \text{s.t. } lb \leq \mathbf{x} \leq ub \end{aligned} \quad (15)$$

In the formula, $\hat{f}_1(\mathbf{x})$ and $\hat{f}_2(\mathbf{x})$ are PRG and RBF, respectively; $\mathbf{x}$ is the independent variable development scheme; and lb and ub are the lower and upper limits of the injection rate and production rate, respectively. Algorithm 3 shows the details of the MSBN.

Algorithm 3. The framework of MSBN

```

01: Input:  $P, C_{\text{best}}, \hat{f}_1(\mathbf{x}), \hat{f}_2(\mathbf{x})$ .
02: While the stopping criterion is not reached
03:   Construct the initial population  $POP$  by the local sampling method for  $P$  and  $C_{\text{best}}$ ;
04:   Feed  $POP$  into NSGA-II, connect the two surrogate models  $\hat{f}_1(\mathbf{x})$  and  $\hat{f}_2(\mathbf{x})$  through the non-
       dominated sorting (refer to Section 3.3); Obtain an optimized set of Pareto solutions,  $pop$ ;
05:   Calculate the two sets of predicted values of  $pop$  using  $\hat{f}_1(\mathbf{x})$  and  $\hat{f}_2(\mathbf{x})$ , respectively;
06:   Sort  $pop$  according to the predicted values, select the two best development schemes,  $par_1$  and  $par_2$ ;
07:   Evaluate  $par_1$  and  $par_2$  with the numerical simulator to get the NPV,  $y_{par_1}, y_{par_2}$ ;
08:   Store the new samples  $(par_1, y_{par_1})$  and  $(par_2, y_{par_2})$  into  $DB$ ;
09: End while
10: Output

```

Table 1
Comparison between local sampling method and unifrnd.

The test case	$\bar{y}_1$	$\bar{y}_2$	σ_1	σ_2
Ellipsoid	1.99E+06	2.21E+06	1.14E+06	1.28E+06
Griewank	2733.896	3007.624	227.0892	249.1488
Rosenbrock	5.31E+08	6.38E+08	2.97E+08	3.70E+08
Dixonpr	3.27E+06	4.02E+07	4.64E+06	6.04E+06

In step 3, we use a local sampling method to form the initial input population of MSBN:

$$\begin{cases} lbound = \min([P, C_{best}]) \\ ubound = \max([P, C_{best}]) \end{cases} \quad (16)$$

$$POP = LHS(lbound, ubound)$$

The excellent development scheme set P selected from DB is combined with the C_{best} obtained from ASSSME to form a local search space by taking the upper and lower bounds of each dimension. Then the outstanding initial population POP is generated in the local region using the LHS method. In the conventional NSGA-II, the initial population is sampled uniformly and randomly in the original decision space by the unifrnd method (Dong et al., 2021).

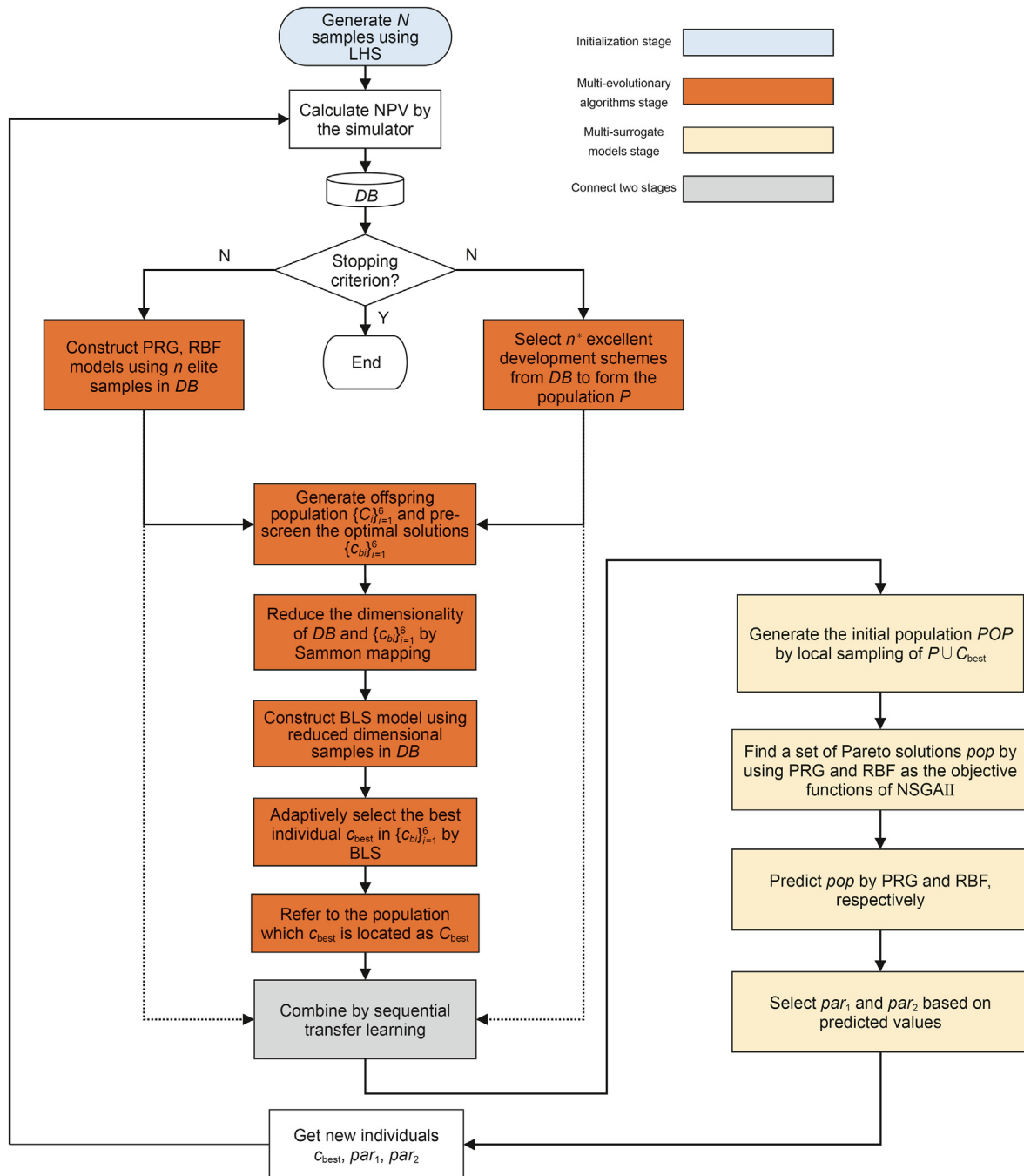


Fig. 6. Flow diagram of the proposed method.

Our local sampling method ensures that the superior sample points generated in ASSSME are not wasted and are added to MSBN's initial population generation session to achieve sequential migration learning. We tested four 100-dimensional functions (minimizing optimization problems) and compared the average function values and standard deviations of the initial populations (population size is 100) produced by the two sampling methods. As shown in Table 1, the initial population generated by the local sampling method has smaller function values ($\bar{y}_1$) and standard deviations (σ_1), which are better than those of the unifrnd method ($\bar{y}_2$ and σ_2). In step 4, our method flow is similar to Fig. 3. There are two differences. First, in the beginning stage, we use local sampling to generate the initial population instead of the unifrnd method. Second, we use multiple surrogate models as the objective functions. As can be seen from Section 3.3, in the fast non-dominant sorting, the Pareto dominance relation is determined according to the objective functions $\hat{f}_1(x)$ and $\hat{f}_2(x)$, and finally, a set of Pareto solutions, pop , can be obtained. In pop , there are many individuals with Pareto rank 1 who do not dominate each other. They perform better on at least one surrogate model compared to individuals of other ranks. We face the same problem as when picking c_{best} in ASSSME.

We cannot evaluate all individuals in the pop using a numerical simulator because this would significantly increase the computational effort. Therefore, selecting the most valuable sample point is a vital issue. In steps 5 and 6, when a set of Pareto solutions (pop) is obtained, PRG and RBF are used to predict it to obtain two sets of predicted values, respectively. The individuals par_1 and par_2 with the largest predicted values in the two sets are picked out, respectively. The numerical simulator calculates the corresponding NPV, and then the sample database is updated. We name this point selection criterion the optimal point criterion. It can have a greater possibility of selecting the best individual in the Pareto solution set according to the fitness value of each individual. In this way, we only need to

All development schemes are placed in the database along with the calculated NPV. In the stage of the multi-evolutionary algorithms (i.e., ASSSME), n sample points with higher NPV in DB are used to train PRG and RBF surrogate models, respectively. Simultaneously, the set P of n^* excellent development schemes serves as the parent population for optimization. In contrast to what is described above, for a clearer description of the proposed method structure, here P is considered to be selected from the DB rather than the db (since db is also selected from the DB). With the assistance of PRG and RBF, the three evolutionary algorithms generate six child populations $\{C_i\}_{i=1}^6$ and the suspected optimal solutions $\{c_{bi}\}_{i=1}^6$ for the corresponding surrogate models' pre-screening. All sample points in DB and $\{c_{bi}\}_{i=1}^6$ are processed by Sammon mapping, and the former is used as the training set to train the BLS model. The latter serves as the test set, and BLS adaptively selects the best individual c_{best} in the $\{c_{bi}\}_{i=1}^6$. The one that contains the c_{best} in the $\{C_i\}_{i=1}^6$ is called C_{best} and participates in the multi-surrogate stage (i.e., MSBN) through sequential transfer learning. The previously established PRG and RBF models and the population P are also involved in the MSBN (dashed part of Fig. 6). After generating the initial population by the local sampling method, NSGA-II is used to find Pareto solutions with good performance on PRG and RBF. The optimal point criterion is used to select excellent Pareto solutions, par_1 and par_2 , without burdening the numerical simulator. ASSSME and MSBN are linked by an initial population constructed by sequential transfer learning. In each cycle, we get three excellent new development schemes (c_{best} , par_1 , and par_2) through two stages, ASSSME and MSBN. The overall algorithm framework is shown in Algorithm 4, and some details have been introduced in Algorithms 2 and 3.

Algorithm 4. The framework of the proposed method

-
- | | |
|-----|---|
| 01: | Initialization: Generate N development schemes $\{x_i\}_{i=1}^N$ using LHS in the whole searching space and then evaluate these points with the numerical simulator to get the corresponding NPV, $\{y_i\}_{i=1}^N$; Archive the samples into database, DB ; Set the generation number, K ; |
| 02: | While the stopping criterion is not reached |
| | // Adaptive selection stage of a single surrogate with the multi-evolutionary algorithms |
| 03: | Sample set db and development scheme set P are selected from DB to construct surrogate models PRG ($\hat{f}_1(x)$), and RBF ($\hat{f}_2(x)$) and form the initial population of evolutionary operators; |
| 04: | Get c_{best} , C_{best} by the process of Algorithm 2; |
| | // Multi-surrogate stage based on NSGA-II |
| 05: | Construct the initial population POP by the local sampling method; |
| 06: | Get par_1 and par_2 by the process of Algorithm 3; |
| 07: | Evaluate c_{best} , par_1 , and par_2 with the numerical simulator to get the NPV, y_{best} , y_{par_1} , and y_{par_2} ; |
| 08: | Store the new samples (c_{best} , y_{best}), (par_1 , y_{par_1}) and (par_2 , y_{par_2}) in DB ; |
| 09: | End while |
| 10: | Output the best solution in DB . |
-

evaluate par_1 and par_2 , which guarantees the quality of the solutions without occupying much computation power in the numerical simulator. This paper presents a multi-surrogate framework based on the PRG and RBF models. When the framework is determined, if a new surrogate model with a better effect appears, it can be directly used in the framework.

4.3. The whole framework of the proposed method

The entire process of MSFASM is shown in Fig. 6. In the initialization stage, the development schemes are generated using the LHS method, and the numerical simulator is used to calculate their NPV.

5. Experimental results and discussion

To demonstrate MSFASM's performance, four benchmark functions and two real reservoir examples are tested in this paper. The benchmark functions are set to 100 dimensions: Ackley, Ellipsoid, Griewank, and Rosenbrock (Jamil and Yang, 2013). The variables of the two reservoirs are 65 and 72, respectively. One is the three-channel model; the other is the PUNQS3 model. To effectively test the performance of MSFASM, it is compared with six other surrogate-model-based methods, including two multi-surrogate frameworks, SGOP (Dong et al., 2021) and Direct (Zerpa et al., 2005), a SADE-Sammon (Chen et al., 2020a) algorithm that also

utilizes Sammon dimensional reduction, and three DE-MSBN, GWO-MSBN, and PSO-MSBN algorithms that combine a single evolutionary algorithm with the MSBN stage. In SGOP, the NSGA-II is also used to connect the PRG and RBF models to construct a multi-surrogate framework to find more excellent solutions. The unifrnd method and the KNN algorithm (Guo et al., 2003) are used in the initial population generation and the selection of the final obtained Pareto solutions in the optimization process, respectively. Direct uses a naive multi-surrogate approach, the weighting method, which divides different weights according to the prediction accuracy of each surrogate model. The final prediction result is the prediction value of each model multiplied by their respective weights and summed. A modified Lipschitzian approach (Jones et al., 1993) is used as an optimizer for Direct, and the two surrogate models used in this paper are also used in its multi-surrogate framework. The comparison with SGOP and Direct highlights the advantages of the multi-surrogate framework of our approach. The SADE-Sammon algorithm uses Sammon mapping to solve the disadvantages of the Kriging model in high-dimensional problems and uses DE as the evolutionary algorithm. The overall processes of DE-MSBN, GWO-MSBN, and PSO-MSBN are similar to those of MSFASM. Still, the difference is that the above three methods do not have the ASSSME stage but only use DE, GWO, and PSO as optimizers, respectively. The comparison with them is to highlight the effectiveness of the ASSSME stage.

The maximum number of function evaluations (i.e., the stopping criterion) of the seven methods tested on the benchmark functions is set to 1000. The maximum number of numerical simulations in the actual reservoir cases is set to 500. The LHS method sets the number of initial sample points (i.e., N) to 200 on the benchmark functions and 130 on the reservoir cases, which is also the number of sample points (i.e., n) in db used to train the surrogate models on each cycle. The number of development schemes (i.e., n^*) in the P that is input to the evolutionary algorithms for optimization is set to 50. In the ASSSME stage, the population is updated for 20 generations in each evolutionary algorithm (i.e., $K = 20$). Sammon mapping reduces the dimensions of variables to 3 dimensions. The number of feature mapping layers, nodes in the feature mapping layer, and nodes in the enhancement layer of BLS are set to 10, 1, and 15, respectively. The number of individuals in the initial population of the MSBN stage is set to 100. The parameter settings of the DE optimizer are consistent with those in Chen et al. (2020b). The learning factors c_1 , c_2 of the PSO algorithm are set to 2, 2, ω_{ini} and ω_{end} are set to 0.8, 0.6 (the specific meaning of symbols can be found in Appendix A).

5.1. Example 1: benchmark function

In this part, we test a total of four benchmark functions. The expressions of these functions and their upper and lower bound constraints are as follows:

$$\text{Ackley: } f(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i)} + 20 + e, \quad x_i \in [-32.768, 32.768] \quad (17)$$

$$\text{Ellipsoid: } f(\mathbf{x}) = \sum_{i=1}^d ix_i^2, \quad x_i \in [-5.12, 5.12] \quad (18)$$

$$\text{Griewank: } f(\mathbf{x}) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad x_i \in [-600, 600] \quad (19)$$

$$\text{Rosenbrock: } f(\mathbf{x}) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right], \quad x_i \in [-5, 10] \quad (20)$$

where d is the number of variables of the benchmark functions, which is set to 100 in this paper, the minimal values of these four tested functions are $f(\mathbf{x}^*) = 0$. To prove that MSFASM is suitable for a variety of problems, we chose different types of test functions, including unimodal and multimodal characteristics.

Fig. 7(a)–(d) show the convergence curves of SGOP, Direct, SADE-Sammon, DE-MSBN, GWO-MSBN, PSO-MSBN, and MSFASM after 10 independent operations on the four benchmark functions. The multi-surrogate methods (SGOP, Direct, DE-MSBN, GWO-MSBN, PSO-MSBN, and MSFASM) perform better than the single-surrogate method (SADE-Sammon) in the optimization of the last three functions, which proves that combining multiple surrogate models can enhance the adaptability to the problems. SADE-Sammon uses Sammon mapping to reduce the dimensions of the development schemes input into the surrogate model, thus improving the prediction accuracy of the Kriging model. However, it only pre-screens the newly generated development schemes with a single surrogate model and cannot find the optimal solution through collaboration among multiple surrogate models as other methods do. Although Direct can converge quickly, its multiple surrogate models are not effectively connected to each other and only sum up their respective predictions based on weights. Even if the surrogate models with higher prediction accuracy are assigned larger weights, the prediction bias from the poorly performing surrogate models will still mislead the optimization direction. Therefore, it is not as effective as MSFASM. Although SGOP also uses NSGA-II to optimize PRG and RBF, it uses the unifrnd method to generate the initial population, which is inferior to the strategy in this paper. Additionally, in the selection of Pareto solutions, SGOP uses the KNN algorithm to select the solution with the largest Euclidean distance from the database as the new sample. This method does not guarantee the quality of the obtained solutions. As a result, SGOP is less effective than MSFASM in terms of overall optimization. It can be seen from the convergence curves that none of these three combined MSBN methods (DE-MSBN, GWO-MSBN, and PSO-MSBN) can obtain optimal solutions for all issues. The reason is that they do not have the adaptive selection process (ASSSME) that uses BLS for multiple evolutionary algorithms, so they do not achieve wide adaptability to the problems, and the final results are not as good as MSFASM's. ASSSME and MSBN are used in MSFASM to enhance the adaptability of both the evolutionary algorithms and the surrogate models to different problems so that the best results can be obtained in all the functions tested. Fig. 7(e)–(h) present the box graphs of the optimal values of seven methods described above after 10 independent operations on four benchmark functions. According to the box graphs, MSFASM has better optimization results and stability than the other six methods.

5.2. Example 2: three-channel model

The first practical reservoir example selected in this paper is the three-channel model. This model's well position and permeability field are shown in Fig. 8. This reservoir is developed under

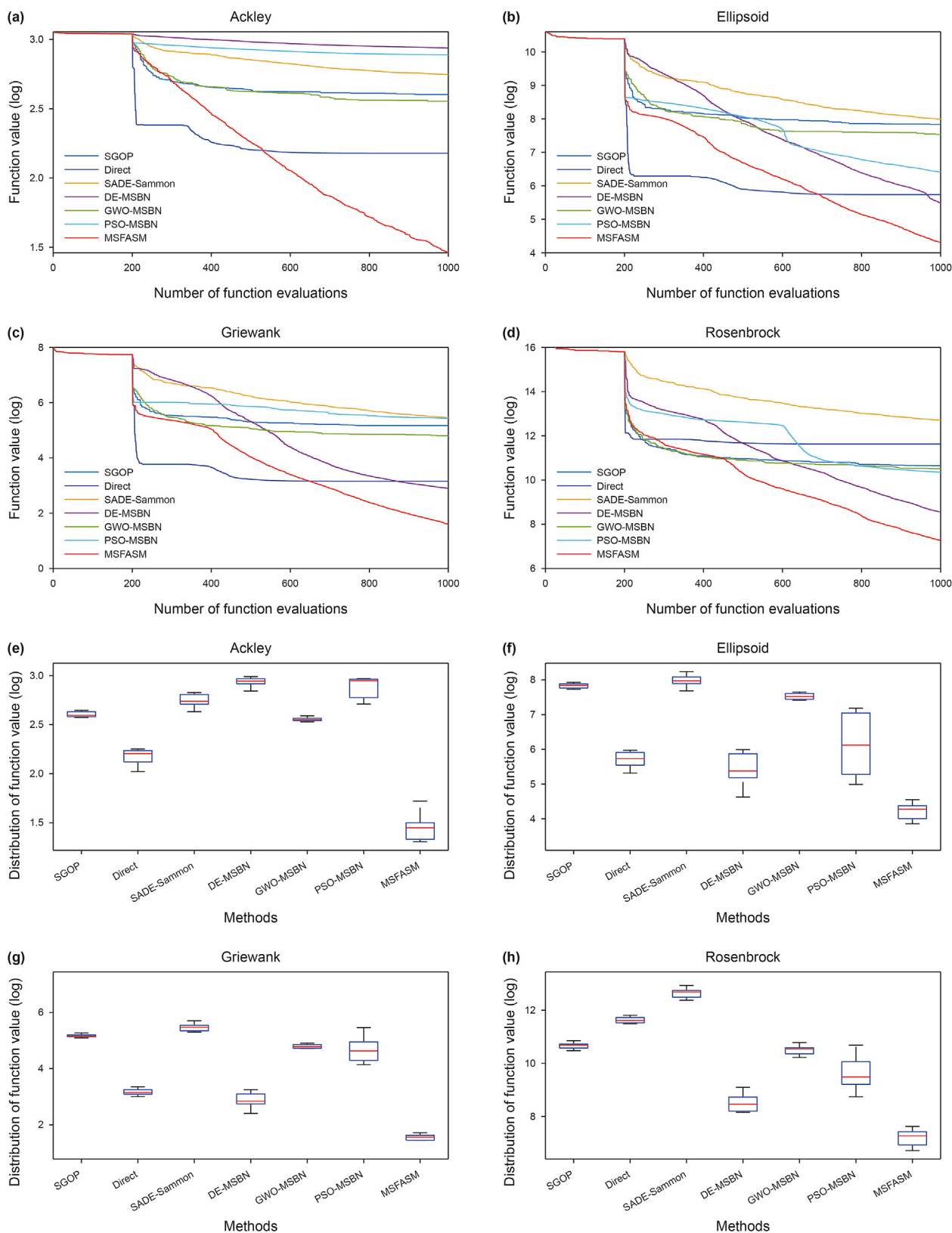


Fig. 7. Results for the benchmark functions with 10 independent runs. (a)–(d) Convergence curves of the benchmark functions; (e)–(h) Distribution of the optimal values of the benchmark functions.

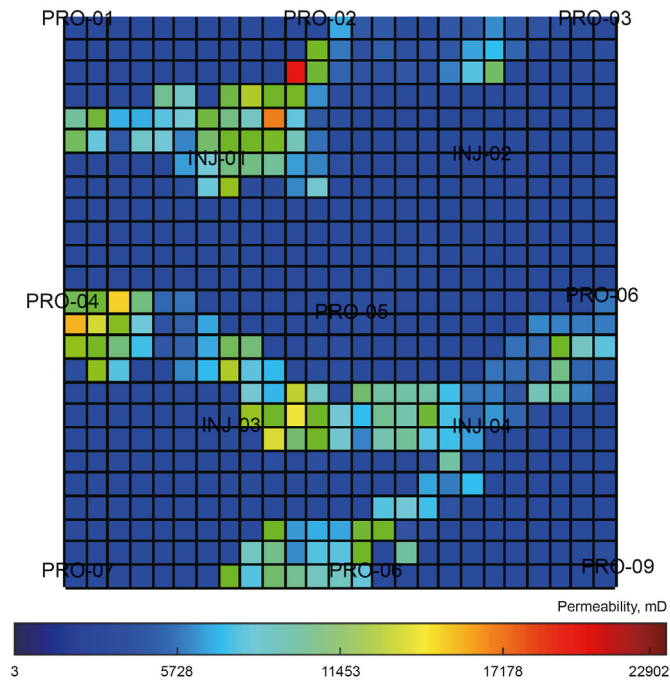


Fig. 8. Well position and permeability field of the three-channel model.

Table 2
Properties of the three-channel model.

Properties	Value
Reservoir grid	25 × 25 × 1
Depth, ft	4800
Initial pressure, psi	4000
Porosity	0.2
Water compressibility, psi ⁻¹	3.74 × 10 ⁻⁶
Rock compressibility, psi ⁻¹	6.10 × 10 ⁻⁵
Density, kg/m ³	911.93
Initial water saturation	0.2
Oil viscosity, cP	1.2

waterflooding with four water-injection wells and nine production wells. The properties of the three-channel model are summarized in Table 2. The model has 25 × 25 × 1 grid blocks. The thickness of each grid block is 20 ft, and the porosity of all grid blocks is 0.2. This paper's control variables are liquid production rate of each producer and water injection rate of each injector. The production rates of the nine production wells are 200 STB/D on the upper bounds and 0 STB/D on the lower bounds. The upper bounds of the injection rates of the four-water injection wells are 500 STB/D, and the lower bounds are 0 STB/D. The production lifetime is 1800 days, divided into 5-time steps on average (360 days for each step), so the total number of decision variables is (4 + 9) × 5 = 65. The oil revenue, water injection, and water-production costs are set to 80, 5, and 5 USD/STB, respectively, and the discount rate is 0%.

Fig. 9(a) shows the optimization result after running the three-channel model independently 10 times. As can be seen from the figure, the multi-surrogate methods using the NSGA-II (SGOP, GMO-MSBN, PSO-MSBN, and MSFASM) outperform the single-surrogate method (SADE-Sammon) in the three-channel model, which proves that solutions suitable for multiple surrogate models can be obtained based on Pareto sampling criteria and improves the optimization performance. Although the SADE-Sammon improves prediction accuracy from the perspective of constructing a surrogate model, the limitation of a single surrogate makes it less adaptable to problems. Direct has the worst performance because it inevitably suffers from the influence of surrogate models with lower prediction accuracy, which leads the optimization in a poorer direction. On the one hand, MSFASM employs BLS to choose the most promising evolutionary algorithm out of a variety while avoiding the limitations of the problems it adapts to. On the other hand, the generation of the initial population of the NSGA-II is guided by the updated population of the evolutionary algorithm, which can provide an excellent starting point for the subsequent optimization. In addition, Pareto solutions are selected from the optimal point criterion according to the predicted value of each surrogate model, so as to ensure the most desirable development schemes. Due to the above advantages, MSFASM is the most effective method for the three-channel model optimization

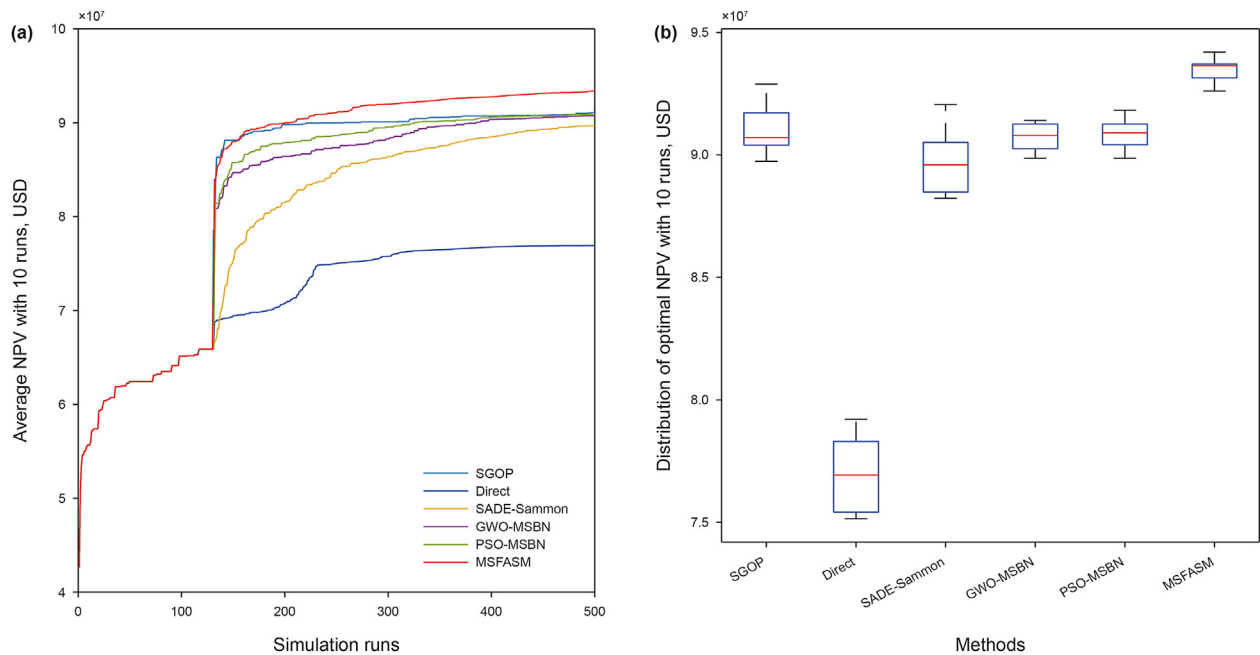


Fig. 9. Results for the three-channel model with 10 independent runs. (a) Average NPV versus simulation runs; (b) Boxplots of the optimal NPV.

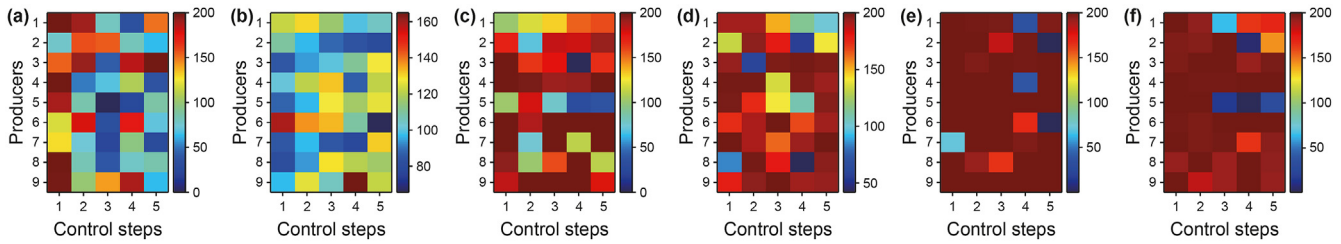


Fig. 10. Optimal liquid-production-rate well controls provided for the three-channel model. (a) SGOP; (b) Direct; (c) SADE-Sammon; (d) GWO-MSBN; (e) PSO-MSBN; (f) MSFASM. Color scale indicates liquid production rate in STB/D.

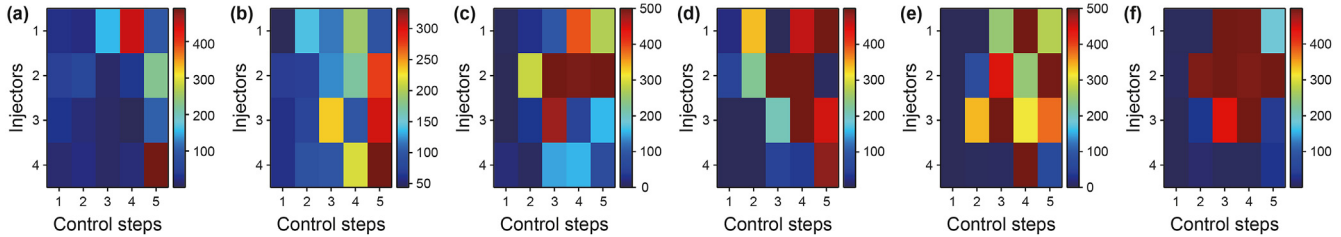


Fig. 11. Optimal water-injection-rate well controls provided for the three-channel model. (a) SGOP; (b) Direct; (c) SADE-Sammon; (d) GWO-MSBN; (e) PSO-MSBN; (f) MSFASM. Color scale indicates water injection rate in STB/D.

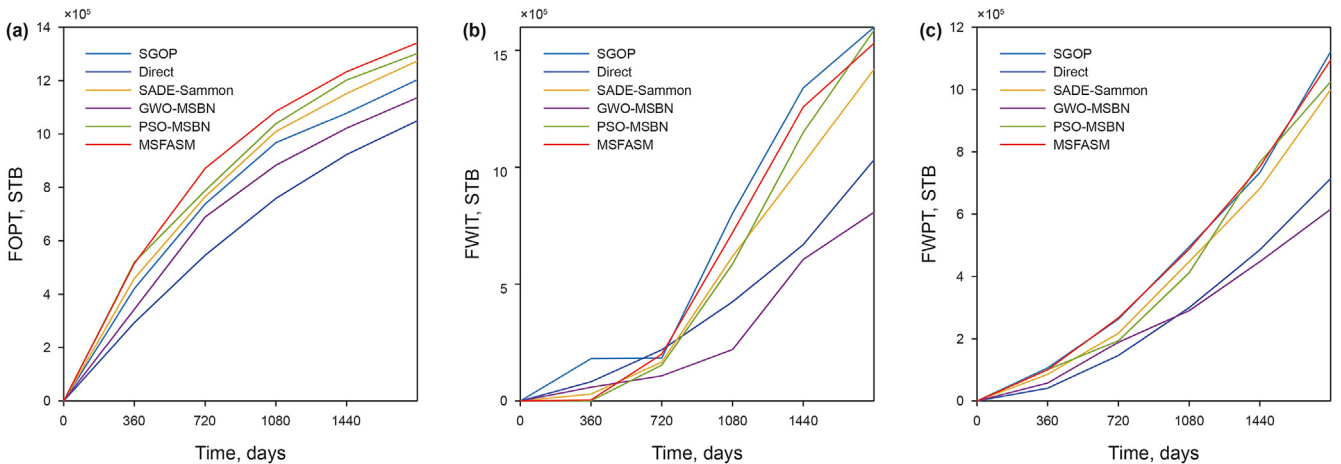


Fig. 12. Results of optimal control for the three-channel model. (a) Cumulative oil production versus time; (b) Cumulative water injection versus time; (c) Cumulative water production versus time.

problem. At the same time, box diagram of the optimal NPV after 10 independent operations (Fig. 9(b)) shows that MSFASM is more stable than other methods.

The optimal well control schemes for the production and injection of wells resulting from all methods are shown in Figs. 10 and 11. Each row represents the optimal scheme of a well throughout the entire optimization time, and each square denotes the control scheme of the well at a time step. The changes in cumulative oil production (FOPT), cumulative water injection (FWIT), and cumulative water production (FWPT) with development time are shown in Fig. 12. MSFASM can obtain the highest cumulative oil production, and although the cumulative water production is also higher, the highest economic benefit is ultimately obtained due to the lower cost of produced water.

5.3. Example 3: PUNQ-S3 model

The second practical reservoir example selected for this paper is the PUNQ-S3 model (Bush et al., 2002). It is a three-dimensional,

three-phase model. The model has $19 \times 8 \times 25$ grid blocks. Only oil and water phases are considered in this paper. The model consists of three water-injection wells and six production wells. The well position and permeability field of this model are shown in Fig. 13. Details of the model can be found in Gao et al. (2006). The production rates of the six production wells are 1000 STB/D on the upper bounds and 0 STB/D on the lower bounds. Water-injection wells have the same water-injection rate limits as production wells. The production lifetime is 5760 days, divided into 8-time steps on average (720 days for each step), so the total number of decision variables is $(3 + 6) \times 8 = 72$. The oil revenue, water injection, and water-production costs are set to 80, 5, and 5 USD/STB, respectively, and the discount rate is 0.

Fig. 14(a) shows the optimization result after the independent running of the PUNQ-S3 model 10 times. As can be seen from the figure, similar to the optimization results of the three-channel model, the multi-surrogate methods using NSGA-II (SGOP, GWO-MSBN, PSO-MSBN, MSFASM) have better performance. This shows that the effect of ensemble surrogate optimization

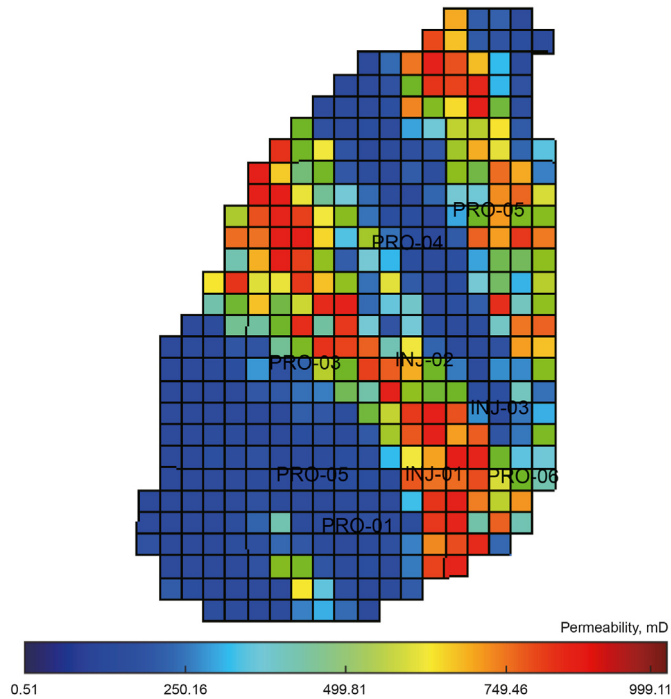


Fig. 13. Well position and permeability field of the PUNQ-S3 model.

constructed by the multi-objective algorithm is better than that of the single-surrogate method (SADE-Sammon) and weighted multi-surrogate method (Direct). For the PUNQS3 reservoir model optimization problem, the overall optimization performance of GWO and PSO is similar, with the former having a strong late-stage exploitation capability and the latter having a high exploration capability in the early stage of optimization, as explained in Section 3.1. Therefore, the difference between the final optimization results

of GWO-MSBN and PSO-MSBN is not significant. MSFASM adaptively selects the best evolutionary algorithm in each iteration using ASSSME, which fully exploits the performance of each optimizer in different optimization periods so that MSFASM can get the best result. The final distribution box diagram of optimal NPV after 10 independent runs in Fig. 14(b) also shows that MSFASM has satisfactory effectiveness and stability.

Figs. 15 and 16 show the optimal well control schemes for the production and injection wells resulting from all methods in the PUNQ-S3 model. In the control schemes of our method, injector 2 has a high water injection rate at each time step because it is located in the middle of the reservoir blocks, and increasing its water injection rate can help other producers. On the contrary, injector 1 is far away from producer 4 and producer 5. If the injection rate of injector 1 is increased, it will not affect them effectively. Therefore, the injection rate of injector 1 is not the same as that of injector 2. Fig. 17 shows the changes in development time for three critical metrics. Although MSFASM has the highest FWPT, it has the most increased cumulative oil production, leading to the highest economic benefits.

6. Conclusions

Using surrogate-assisted evolutionary algorithms (SAEAs) can solve the time-consuming problem of reservoir numerical simulators in the evaluation process while finding the optimal development scheme. However, one surrogate model cannot solve all problems, and various evolutionary algorithms affect different issues differently. Therefore, it is necessary to use a method that effectively combines multiple surrogate models with multiple evolutionary algorithms to have broad adaptability to reservoir blocks and optimization periods. This paper proposes an efficient approach, MSFASM, to solve production optimization problems that combines multi-evolutionary algorithms and multi-surrogate models through sequential migration learning. MSFASM is

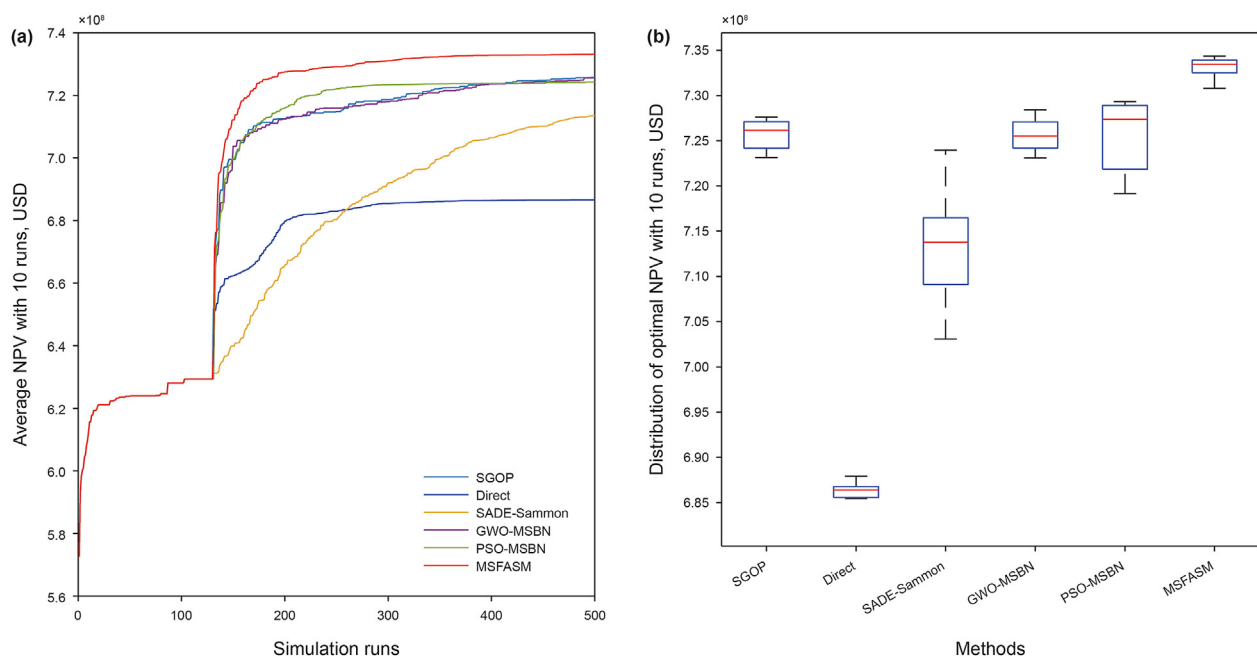


Fig. 14. Results for the PUNQ-S3 model with 10 independent runs. (a) Average NPV versus simulation runs. (b) Boxplots of the optimal NPV.

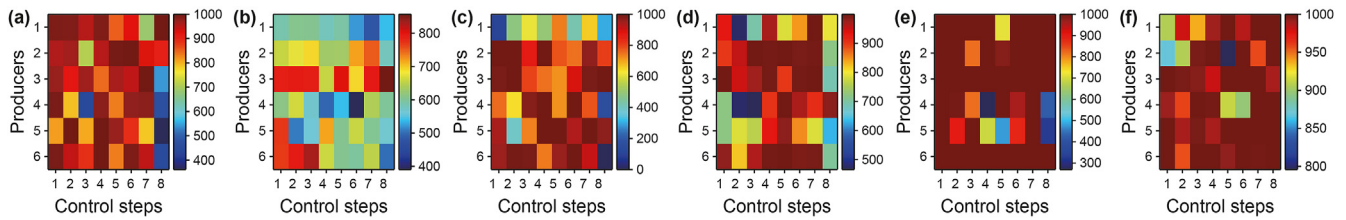


Fig. 15. Optimal liquid-production-rate well controls provided for the PUNQ-S3 model. (a) SGOP; (b) Direct; (c) SADE-Sammon; (d) GWO-MSBN; (e) PSO-MSBN; (f) MSFASM. Color scale indicates liquid production rate in STB/D.

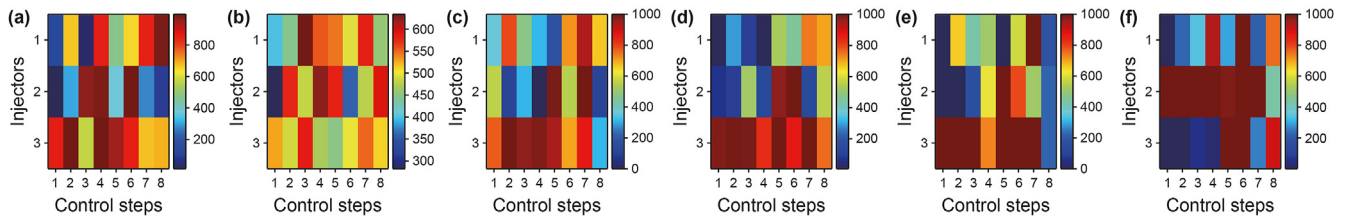


Fig. 16. Optimal water-injection-rate well controls provided for the PUNQ-S3 model. (a) SGOP; (b) Direct; (c) SADE-Sammon; (d) GWO-MSBN; (e) PSO-MSBN; (f) MSFASM. Color scale indicates water injection rate in STB/D.

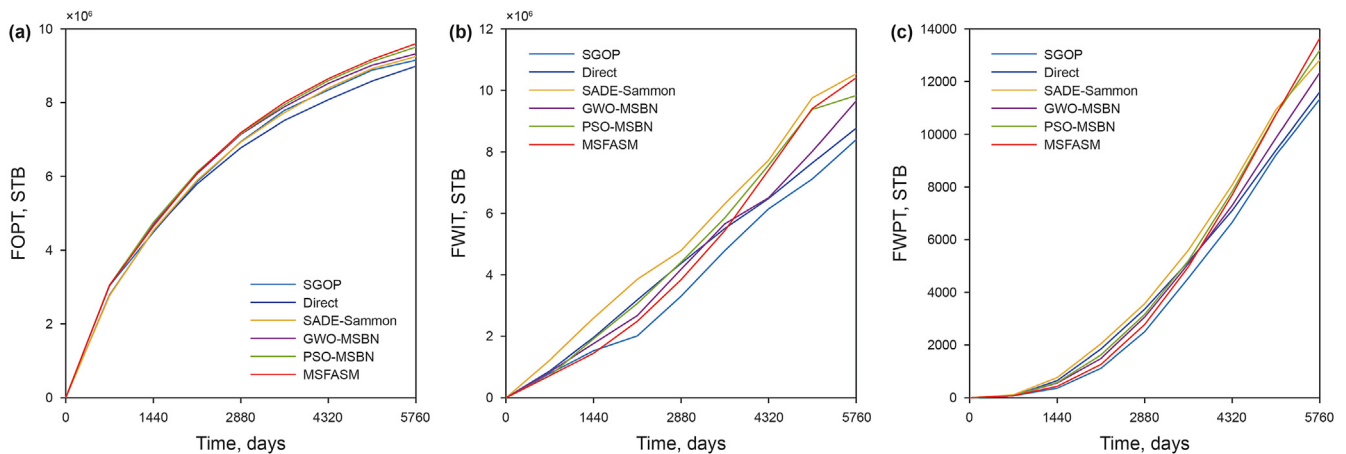


Fig. 17. Results of optimal control for the PUNQ-S3 model. (a) Cumulative oil production versus time; (b) Cumulative water injection versus time; (c) Cumulative water production versus time.

divided into two stages: ASSSME and MSBN. The former combines the advantages of various evolutionary algorithms and provides a suitable population for the next stage, while MSBN fully exploits the benefits of different surrogate models. Combining the two can finally lead to optimal solutions with wide adaptability to various problems. To verify the validity of MSFASM, four benchmark functions and two reservoir models are tested, and the results are compared with those obtained from six other surrogate-model-based methods. Experimental results show that the MSFASM can obtain the minimum values of the benchmark functions and the optimal well control schemes for the production optimization problems, thus achieving the highest NPV. Compared with other methods, MSFASM has better stability. This also shows that the framework proposed in this paper can be adapted to various problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have

appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 52274057, 52074340 and 51874335, the Major Scientific and Technological Projects of CNPC under Grant ZD2019-183-008, the Major Scientific and Technological Projects of CNOOC under Grant CCL2022RCP0397RSN, the Science and Technology Support Plan for Youth Innovation of University in Shandong Province under Grant 2019KJH002, 111 Project under Grant B08028.

Appendix A

In order to save the length of Section 3.1, the other two evolutionary algorithms used in this paper, grey wolf optimization (GWO) and particle swarm optimization (PSO), are introduced here.

The GWO algorithm simulates grey wolf predation and has strong convergence performance. It takes the optimal development

scheme as α , and the second and third best schemes are β and δ , respectively, and they can be regarded as the leaders of the pack. The remaining candidate solutions are represented by ϖ and guided by the three wolves. In the process of population evolution, α , β , and δ are first found according to fitness values. The distances between them and other individuals are calculated using the following formula:

$$\begin{aligned} D_{\alpha} &= |C_1 \cdot \alpha(t) - \varpi(t)|, \\ D_{\beta} &= |C_2 \cdot \beta(t) - \varpi(t)|, \\ D_{\delta} &= |C_3 \cdot \delta(t) - \varpi(t)|, \end{aligned} \quad (A1)$$

where t represents the current iteration number; D_{α} , D_{β} , and D_{δ} represent the distances between α , β , and δ from other individuals, ϖ , respectively; C is a coefficient vector that can be calculated as follows:

$$C = 2 \cdot r_1 \quad (A2)$$

where r_1 is a random number between 0 and 1. The individuals for the next generation are generated by the following formulas:

$$\varpi(t+1) = \frac{\alpha(t) - A_1 \cdot (D_{\alpha}) + \beta(t) - A_2 \cdot (D_{\beta}) + \delta(t) - A_3 \cdot (D_{\delta})}{3} \quad (A3)$$

where A is a random vector that can be calculated as follows:

$$A = 2a \cdot r_2 - a \quad (A4)$$

where a is the vector between 0 and 2 that decreases linearly during the iteration; r_2 is a random number between 0 and 1.

Particle swarm optimization is a kind of evolutionary computing technology that is derived from the study of the predation behavior of birds and then builds a model by using swarm intelligence. PSO makes use of the information sharing among individuals in a group to make the movement of the whole group have a trend from disorder to order, and then obtains the optimal solution. PSO mainly defines two concepts, one of which is speed of movement:

$$\begin{aligned} v_i^{(t+1)} &= \omega \times v_i^{(t)} + c_1 \times rand \times (pbest_i - x_i^{(t)}) + c_2 \times rand \\ &\times (gbest - x_i^{(t)}) \end{aligned} \quad (A5)$$

where $v_i^{(t+1)}$ is the velocity of the i th particle when the evolutionary algebra is $t + 1$; c_1 and c_2 are learning factors; $rand$ is a random number between 0 and 1; $pbest_i$ is the current best development scheme for the i th particle; $gbest$ is the current global optimal production system of the group; $x_i^{(t)}$ is the position of the i th particle with the evolution algebra t ; ω is the inertia weight. In this paper, the strategy of linearly decreasing weight is used for ω :

$$\omega^{(t)} = \omega_{ini} - (\omega_{ini} - \omega_{end}) \times \frac{t}{t_{max}} \quad (A6)$$

where $\omega^{(t)}$ is the inertia weight when the evolutionary algebra is t ; ω_{ini} is the initial inertia weight; ω_{end} is the inertia weight when iterating to the maximum evolutionary algebra t_{max} . Another important concept is the position of movement:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t)} \quad (A7)$$

where $x_i^{(t+1)}$ is the position of the i th particle in evolutionary algebra $t + 1$.

References

- Al-Aghbari, M., Gujarathi, A.M., 2022. Hybrid optimization approach using evolutionary neural network & genetic algorithm in a real-world waterflood development. *J. Petrol. Sci. Eng.* 216, 110813. <https://doi.org/10.1016/j.petrol.2022.110813>.
- An, Z., Zhou, K., Hou, J., et al., 2022. Accelerating reservoir production optimization by combining reservoir engineering method with particle swarm optimization algorithm. *J. Petrol. Sci. Eng.* 208, 109692. <https://doi.org/10.1016/j.petrol.2021.109692>.
- Bush, M., Cuypers, M., Roggero, F., et al., 2002. Comparison of production forecast uncertainty quantification methods—An integrated study. Delft, The Netherlands. <http://www.nitgton.nl/punq/cases/punqs3/PUNQS3paper/index.htm>.
- Chen, B., Fonseca, R.-M., Leeuwenburgh, O., et al., 2017. Minimizing the risk in the robust life-cycle production optimization using stochastic simplex approximate gradient. *J. Petrol. Sci. Eng.* 153, 331–344. <https://doi.org/10.1016/j.petrol.2017.04.001>.
- Chen, B., Xu, J., 2019. Stochastic simplex approximate gradient for robust life-cycle production optimization: applied to brugge field. *J. Energy Resour. Technol.* 141 (9). <https://doi.org/10.1115/1.4043244>.
- Chen, C.L.P., Liu, Z., 2018. Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. *IEEE Transact. Neural Networks Learn. Syst.* 29 (1), 10–24. <https://doi.org/10.1109/TNNLS.2017.2716952>.
- Chen, G., Zhang, K., Xue, X., et al., 2020a. Surrogate-assisted evolutionary algorithm with dimensionality reduction method for water flooding production optimization. *J. Petrol. Sci. Eng.* 185, 106633. <https://doi.org/10.1016/j.petrol.2019.106633>.
- Chen, G., Zhang, K., Zhang, L., et al., 2020b. Global and local surrogate-model-assisted differential evolution for waterflooding production optimization. *SPE J.* 25 (1), 105–118. <https://doi.org/10.2118/199357-PA>.
- Das, S., Suganthan, P.N., 2010. Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* 15 (1), 4–31. <https://doi.org/10.1109/TEVC.2010.2059031>.
- Deb, K., Agrawal, S., Pratap, A., et al., 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: *Proceedings of Parallel Problem Solving from Nature PPSN VI*. https://doi.org/10.1007/3-540-45356-3_83.
- Dong, H., Wang, P., Chen, W., et al., 2021. SGOP: surrogate-assisted global optimization using a Pareto-based sampling strategy. *Appl. Soft Comput.* 106, 107380. <https://doi.org/10.1016/j.asoc.2021.107380>.
- Farahi, M.M.M., Ahmadi, M., Dabir, B., 2021. Model-based water-flooding optimization using multi-objective approach for efficient reservoir management. *J. Petrol. Sci. Eng.* 196, 107988. <https://doi.org/10.1016/j.petrol.2020.107988>.
- Forouzanfar, F., Rossa, E.D., Russo, R., et al., 2013. Life-cycle production optimization of an oil field with an adjoint-based gradient approach. *J. Petrol. Sci. Eng.* 112, 351–358. <https://doi.org/10.1016/j.petrol.2013.11.024>.
- Foss, B., Jensen, J.P., 2011. Performance analysis for closed-loop reservoir management. *SPE J.* 16 (1), 183–190. <https://doi.org/10.2118/138891-PA>.
- Gao, G., Zafari, M., Reynolds, A.C., 2006. Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF. *SPE J.* 11 (4), 506–515. <https://doi.org/10.2118/93324-MS>.
- Gu, J., Liu, W., Zhang, K., et al., 2021. Reservoir production optimization based on surrogate model and differential evolution algorithm. *J. Petrol. Sci. Eng.* 205, 108879. <https://doi.org/10.1016/j.petrol.2021.108879>.
- Guo, G., Wang, H., Bell, D., et al., 2003. KNN model-based approach in classification. In: *Meersman, R., Tari, Z., Schmidt, D.C. (Eds.), On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science*, vol. 2888. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-39964-3_62.
- Güyüğüler, B., Horne, R.N., Rogers, L., et al., 2002. Optimization of well placement in a Gulf of Mexico waterflooding project. *SPE Reservoir Eval. Eng.* 5 (3), 229–236. <https://doi.org/10.2118/78266-PA>.
- Hou, J., Zhou, K., Zhang, X.-S., et al., 2015. A review of closed-loop reservoir management. *Petrol. Sci.* 12 (1), 114–128. <https://doi.org/10.1007/s12182-014-0005-6>.
- Isebor, O.J., Durllofsky, L.J., 2014. Biobjective optimization for general oil field development. *J. Petrol. Sci. Eng.* 119, 123–138. <https://doi.org/10.1016/j.petrol.2014.04.021>.
- Jamil, M., Yang, X.-S., 2013. A literature survey of benchmark functions for global optimisation problems. *Int. J. Math. Model. Numer. Optim.* 4 (2), 150–194. <https://doi.org/10.1504/IJMMNO.2013.055204>.
- Jin, Y., Wang, H., Chugh, T., et al., 2019. Data-driven evolutionary optimization: an overview and case studies. *IEEE Trans. Evol. Comput.* 23 (3), 442–458. <https://doi.org/10.1109/TEVC.2018.2869001>.
- Jones, D.R., Perttunen, C.D., Stuckman, B.E., 1993. Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theor. Appl.* 79 (1), 157–181. <https://doi.org/10.1007/BF00941892>.
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: *Proceedings of ICNN'95—international Conference on Neural Networks*. <https://doi.org/10.1109/ICNN.1995.488968>.
- Li, F., Li, Y., Cai, X., et al., 2022. A surrogate-assisted hybrid swarm optimization algorithm for high-dimensional computationally expensive problems. *Swarm*

- Evol. Comput. 72, 101096. <https://doi.org/10.1016/j.swevo.2022.101096>.
- Luo, C., Zhang, S.-L., Wang, C., et al., 2011. A metamodel-assisted evolutionary algorithm for expensive optimization. *J. Comput. Appl. Math.* 236 (5), 759–764. <https://doi.org/10.1016/j.cam.2011.05.047>.
- Ma, X., Zhang, K., Zhang, J., et al., 2022a. A novel hybrid recurrent convolutional network for surrogate modeling of history matching and uncertainty quantification. *J. Petrol. Sci. Eng.* 210, 110109. <https://doi.org/10.1016/j.petrol.2022.110109>.
- Ma, X., Zhang, K., Zhao, H., et al., 2022b. A vector-to-sequence based multilayer recurrent network surrogate model for history matching of large-scale reservoir. *J. Petrol. Sci. Eng.* 214, 110548. <https://doi.org/10.1016/j.petrol.2022.110548>.
- Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. *Adv. Eng. Software* 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- Mirzaei-Paiaman, A., Santos, S.M.G., Schiozer, D.J., 2021. A review on closed-loop field development and management. *J. Petrol. Sci. Eng.* 201, 108457. <https://doi.org/10.1016/j.petrol.2021.108457>.
- Oliveira, D.F., Reynolds, A., 2013. An adaptive hierarchical algorithm for estimation of optimal well controls. In: *Proceedings of SPE Reservoir Simulation Symposium*. <https://doi.org/10.2118/163645-MS>.
- Ostertagová, E., 2012. Modelling using polynomial regression. *Procedia Eng.* 48, 500–506. <https://doi.org/10.1016/j.proeng.2012.09.545>.
- Ren, G., He, J., Wang, Z., et al., 2019. Implementation of physics-based data-driven models with a commercial simulator. In: *Proceedings of SPE Reservoir Simulation Conference*. <https://doi.org/10.2118/193855-MS>.
- Sammon, J.W., 1969. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* 100 (5), 401–409. <https://doi.org/10.1109/T-C.1969.222678>.
- Viana, F.A., 2016. A tutorial on Latin hypercube design of experiments. *Qual. Reliab. Eng. Int.* 32 (5), 1975–1985. <https://doi.org/10.1002/qre.1924>.
- Volkov, O., Bellout, M.C., 2017. Gradient-based production optimization with simulation-based economic constraints. *Comput. Geosci.* 21 (5), 1385–1402. <https://doi.org/10.1007/s10596-017-9634-3>.
- Wang, Z., He, J., Milliken, W.J., et al., 2021. Fast history matching and optimization using a novel physics-based data-driven model: an application to a diatomite reservoir. *SPE J.* 26 (6), 4089–4108. <https://doi.org/10.2118/200772-PA>.
- Whitley, D., 1994. A genetic algorithm tutorial. *Stat. Comput.* 4 (2), 65–85. <https://doi.org/10.1007/BF00175354>.
- Xue, X., Chen, G., Zhang, K., et al., 2022. A divide-and-conquer optimization paradigm for waterflooding production optimization. *J. Petrol. Sci. Eng.* 211, 110050. <https://doi.org/10.1016/j.petrol.2021.110050>.
- Yu, H., Tan, Y., Zeng, J., et al., 2018. Surrogate-assisted hierarchical particle swarm optimization. *Inf. Sci.* 454, 59–72. <https://doi.org/10.1016/j.ins.2018.04.062>.
- Zerpa, L.E., Queipo, N.V., Pintos, S., et al., 2005. An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. *J. Petrol. Sci. Eng.* 47 (3), 197–208. <https://doi.org/10.1016/j.petrol.2005.03.002>.
- Zhang, K., Zhang, L.-m., Yao, J., et al., 2014. Water flooding optimization with adjoint model under control constraints. *Journal of Hydrodynamics, Ser. B.* 26 (1), 75–85. [https://doi.org/10.1016/S1001-6058\(14\)60009-3](https://doi.org/10.1016/S1001-6058(14)60009-3).
- Zhao, H., Kang, Z., Zhang, X., et al., 2016. A physics-based data-driven numerical model for reservoir history matching and prediction with a field application. *SPE J.* 21 (6), 2175–2194. <https://doi.org/10.2118/173213-PA>.
- Zhao, M., Zhang, K., Chen, G., et al., 2020a. A surrogate-assisted multi-objective evolutionary algorithm with dimension-reduction for production optimization. *J. Petrol. Sci. Eng.* 192, 107192. <https://doi.org/10.1016/j.petrol.2020.107192>.
- Zhao, M., Zhang, K., Chen, G., et al., 2020b. A classification-based surrogate-assisted multiobjective evolutionary algorithm for production optimization under geological uncertainty. *SPE J.* 25 (5), 2450–2469. <https://doi.org/10.2118/201229-PA>.
- Zhao, X., Zhang, K., Chen, G., et al., 2020c. Surrogate-assisted differential evolution for production optimization with nonlinear state constraints. *J. Petrol. Sci. Eng.* 194, 107441. <https://doi.org/10.1016/j.petrol.2020.107441>.
- Zhong, C., Zhang, K., Xue, X., et al., 2022. Surrogate-reformulation-assisted multi-tasking knowledge transfer for production optimization. *J. Petrol. Sci. Eng.* 208, 109486. <https://doi.org/10.1016/j.petrol.2021.109486>.