



ARTICLE

Machine-Learning-Based Multi-Surrogate-Assisted Joint Optimization for Hydraulic Fracturing Design and Production Control

Xiaopeng Ma^{1,2,*}, Bin Zhang^{1,2}, Jinsheng Zhao^{1,2} and Di Zhu^{1,2}

¹College of Petroleum Engineering, Xi'an Petroleum University, Xi'an, China

²Key Laboratory of Exploration and Development of Complex and Difficult-to-Produce Oil & Gas Reservoirs (Xi'an Petroleum University), Ministry of Education, Xi'an, China

*Corresponding Author: Xiaopeng Ma. Email: xpma@xsyu.edu.cn

Received: 22 June 2026; Accepted: 29 July 2026; Published: 18 October 2026

ABSTRACT: The distribution of hydraulic fractures and production control strategies have significant influences on the fluid flow and production performance of low-permeability waterflooding reservoirs. Traditional approaches typically focus solely on fracture parameters while overlooking production control. To address this limitation, this work proposes a joint optimization framework that simultaneously integrates hydraulic fracturing design and production control. However, the joint optimization of hydraulic fracture distribution and production control requires a large amount of reservoir numerical simulation. To solve this problem, a novel adaptive multi-surrogate-assisted differential evolution (AMSADe) algorithm is developed. The AMSADe algorithm utilizes a surrogate model pool comprising radial basis functions, polynomial response surfaces, and deep neural networks. Additionally, an embedded discrete fracture model (EDFM) is adopted for simulation of fractured reservoir flow and optimization evaluation. The proposed method was applied to a two-dimensional waterflooding reservoir model. The results demonstrate that the algorithm converges rapidly, requiring only 200 numerical simulations to achieve optimal performance. Compared with the classical differential evolution algorithm, the net present value was improved by 17.5%. Overall, the proposed joint optimization framework based on the AMSADe algorithm successfully and simultaneously determines the optimal fracturing and production control parameters.

KEYWORDS: Hydraulic fracturing design; production control; Joint optimization; multi-surrogate model; differential evolution; embedded discrete fracture model

1 Introduction

Hydraulic fracturing technology plays an important role in the successful exploitation and development of low-permeability reservoirs. A network of fractures can be formed in the reservoir after hydraulic fracturing, including primary fractures, secondary induced fractures and the original natural fractures, which can enhance the flow capacity of reservoirs [1]. Generally, the hydraulic fracturing is combined with horizontal well technology. Multistage hydraulic fracturing operation is carried out along the horizontal section of the well to form a large-scale fracture network [2]. Since fractures are the main flow channels, the morphology and distribution of the fracture network have a significant impact on the development effect of reservoirs [3]. Consequently, optimizing hydraulic fracturing design [4] to identify the ideal fracture geometry and distribution pattern remains a critical task for the successful development of low-permeability tight reservoirs.

Hydraulic fracturing design problems have been widely studied over the past few decades. Yu et al. [5,6] conducted numerical sensitivity studies coupled with economic analysis to optimize the placement of multiple horizontal wells and fracture half-lengths. Using a dimensionless analysis approach, Luo et al. [7] evaluated the sensitivity of rock mechanical and fracturing parameters across different scales, providing a theoretical foundation for both laboratory and field-scale fracturing designs. Additionally, Hu et al. [8] explored the concept of hydraulic fracturing-induced energy storage in shale formations, analyzing the key factors influencing energy storage capacity and efficiency. However, such sensitivity analysis methods are often computationally expensive and rarely yield optimal solutions. To address this, optimization-based design methods have been developed, coupling numerical simulations with optimization algorithms. For instance, Ma et al. [9] optimized hydraulic fracture placement and compared solutions obtained via the gradient-based finite difference method. Rammay and Awotunde [10] utilized a differential evolution algorithm paired with local grid refinement to perform a multivariate optimization of fracture conductivity, length, and spacing. Furthermore, Ruiz Maraggi et al. [11] proposed a Rapid RTA-based workflow that optimizes cluster spacing and fracture count to maximize NPV of unconventional wells at DCA-like computational speeds. Although these studies demonstrate that optimization frameworks can effectively determine optimal fracturing parameters, the production controls in these methods remain predetermined and static throughout the optimization process. Consequently, the resulting optimal fracturing parameters may only be valid under a specific, predefined production regime, failing to achieve true global system efficiency.

To address this limitation, this work introduces a joint optimization framework that simultaneously couples hydraulic fracturing design and production control. Production control is a vital technology in reservoir management, primarily aimed at maximizing economic performance by regulating the operating constraints of injection and production wells. For production control optimization in conventional water-flooded reservoirs, Chen et al. [12] proposed a surrogate-assisted differential evolution algorithm, which achieved higher NPV and faster convergence than traditional evolutionary algorithms. In addition, Kor et al. [13] introduced a Bayesian optimization workflow for production control, which successfully reduced the computational overhead associated with fluid flow simulations. Recently, Wang et al. [14] provided a comprehensive review of intelligent optimization algorithms applied to production control problems across both conventional and unconventional reservoirs. By integrating fracturing design with production management, the joint optimization framework concurrently accounts for the interactive influences of fracture geometry and dynamic operating parameters on fluid flow, thereby ensuring optimal fracture utilization throughout the entire production lifecycle. However, executing such a joint optimization is inherently challenged by high-dimensional parameter spaces and prohibitive computational costs.

In the joint optimization of hydraulic fracturing design and production control, the decision variables span two categories: static fracture parameters and dynamic operational parameters for injection and production wells. The total dimensionality increases linearly with the number of fractures, active wells, and the frequency of production control adjustments, typically resulting in tens to thousands of variables. Although stochastic optimization algorithms can effectively navigate such high-dimensional spaces, they require excessive function evaluations, leading to prohibitive computational overhead. For joint optimization, each evaluation relies on high-fidelity reservoir numerical simulation, which is notoriously time-consuming and computationally restrictive. To mitigate this challenge, machine learning-based surrogate models have been increasingly integrated into optimization workflows. Notable examples include the dimensionality reduction of fracture parameters using deep autoencoders [15], a global and local surrogate-assisted algorithm for production optimization [16], a distributed surrogate system for history matching [17], a machine learning-assisted global optimization framework for shale gas reservoirs [18], and a convolutional-

recurrent neural network proxy engineered for fractured media under geological uncertainty [19]. In this work, we develop a novel Adaptive Multi-Surrogate-Assisted Differential Evolution (AMSADe) algorithm for the joint optimization problem. The AMSADe framework utilizes a heterogeneous surrogate pool comprising Radial Basis Functions (RBF), Polynomial Response Surfaces (PRS), and Deep Neural Networks (DNN). Crucially, a two-level multi-armed bandit (TL-MAB) mechanism is constructed to adaptively select the most suitable surrogate and infill strategy during the evolutionary process, effectively balancing global exploration and local exploitation. Candidate solutions generated by differential evolution are sequentially screened by the selected infill criterion and validated via a high-fidelity Embedded Discrete Fracture Model (EDFM) simulator [20]. Through this synchronized online learning and adaptive selection loop, the surrogate pool is progressively refined, efficiently guiding the heuristic search toward the global optimum.

The rest of this paper is organized as follows. Section 2 formulates the joint optimization problem and describes the EDFM model. Section 3 details the AMSADe framework and its adaptation to the problem, including the surrogate pool, the TL-MAB mechanism and the online learning procedure. Section 4 presents the experiments and compares AutoSAEA with several baseline algorithms, followed by an analysis of surrogate accuracy and a discussion of the results. Section 5 concludes the paper and outlines future research directions.

2 Problem Formulation

The integrated optimization of fracturing design and time-varying well controls is formulated as a bound-constrained expensive black-box problem. The decision vector, objective function, and constraints are described below.

2.1 Decision Variables

The design variables comprise two groups: fracturing fracture parameters and production schedule parameters. They are collected in a vector $\mathbf{x} \in \mathbb{R}^D$, where the dimension D depends on the number of fracture stages and the number of control cycles.

Fracture parameters. Considering the case of horizontal well steering and multi-stage fracturing, the fracturing parameters include the azimuth θ of the horizontal well, the stage spacing d_{well} , a dimensionless conductivity multiplier c_{tran} , and the individual stage lengths $L_1, \dots, L_{n_s}$. These parameters govern the geometry and conductivity of the fractures and consequently the reservoir flow paths.

Production control parameters. In general, the water injection rate of injectors and the bottom-hole pressure (BHP) of producers are controlled to vary over the production horizon. The time horizon is discretized into C control timesteps, and the schedule can be expressed in matrix form as follows:

$$\mathbf{Q} = \begin{bmatrix} q_{1,1} & \cdots & q_{1,C} \\ \vdots & \ddots & \vdots \\ q_{N_{\text{inj}},1} & \cdots & q_{N_{\text{inj}},C} \end{bmatrix} \quad (1)$$

$$\mathbf{p} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,C} \\ \vdots & \ddots & \vdots \\ p_{N_{\text{pro}},1} & \cdots & p_{N_{\text{pro}},C} \end{bmatrix} \quad (2)$$

where $q_{i,c}$ denotes the water injection rate of i -th well during cycle c , and $p_{j,c}$ is the BHP (barsa) of the j -th production well in cycle c . N_{inj} and N_{pro} represent the number of injectors and producers, respectively.

All variables are restricted to physically meaningful intervals that ensure stable operations and respect field constraints.

2.2 Objective Function Based on Economic Benefits

Hydraulic fracturing enhances reservoir flow capacity by forming complex fracture networks and is critical for the effective development of tight reservoirs. The morphology and characteristics of hydraulic fractures, such as horizontal well azimuth, number of fracturing stages, fracture spacing, half-length, and conductivity, significantly impact reservoir production performance, making their optimized design essential for guiding field implementation. To evaluate the economic performance of different fracturing schemes and production control parameters, the net present value (NPV) is adopted as the core economic objective, which accounts for oil revenue, water handling costs, water injection costs, fracturing investment, and fixed capital expenditure.

For a given decision vector $\mathbf{x}$ which represents the hydraulic fracturing design parameters and production control parameters, the NPV is defined as:

$$J(\mathbf{x}) = \sum_{k=1}^{N_{ts}} \frac{\Delta t_k (Q_{o,k} R_o - Q_{w,k} R_w - Q_{i,k} R_i)}{(1+b)^{t_k/365}} - C_{\text{frac}}(\mathbf{x}) - C_{\text{fixed}} \quad (3)$$

where: N_{ts} is the total number of simulation time steps; Δt_k is the time span of the k -th time step; $Q_{o,k}$, $Q_{w,k}$, and $Q_{i,k}$ are the oil production rate, water production rate, and water injection rate at time step k (STB/D); R_o , R_w , and R_i are the oil price, water disposal cost, and water injection cost (USD/STB), respectively; b is the annual discount rate; t_k is the cumulative time at the k -th time step; $C_{\text{frac}}(\mathbf{x})$ is the total fracturing-related cost, split into horizontal well and stage-specific costs:

$$C_{\text{frac}}(\mathbf{x}) = \sum_{j=1}^{N_w} \left(C w_j + \sum_{i=1}^{N f s_j} C f s_{ij} \right) \quad (4)$$

where N_w is the number of horizontal wells, $N f s_j$ is the number of fracturing stages for well j , the horizontal well cost is $C w_j = 300 \times L w_j$ ($L w_j$ = length of the horizontal section of well j), and the cost of stage i in well j follows the correlation [21]:

$$C f s_{ij} = 0.96 \times (C d_{ij}/1000)^{1.084} \times x f_{ij} \times l h f_{ij} \quad (5)$$

with $C d_{ij}$, $x f_{ij}$, and $l h f_{ij}$ representing fracture conductivity, height, and half-length, respectively; C_{fixed} represents fixed expenditures (e.g., well drilling, surface facilities). The production terms $Q_{o,k}$, $Q_{w,k}$, and $Q_{i,k}$ are not explicit functions of the design variables $\mathbf{x}$, but are obtained from high-fidelity reservoir simulations. The simulator acts as a black-box mapping $\mathbf{s} = g(\mathbf{x})$, where $\mathbf{s}$ denotes the reservoir state variables (e.g., pressure and saturation fields) over time, from which all production rates are extracted.

2.3 Optimization Model

The joint optimization problem is formulated as a computationally expensive black-box optimization problem, which can be compactly stated as:

$$\max_{\mathbf{x}} J(\mathbf{x}) \quad \text{s.t.} \quad \begin{cases} x_k^{\min} \leq x_k \leq x_k^{\max}, & k = 1, \dots, D \\ c(\mathbf{x}) \leq 0 \end{cases} \quad (6)$$

where $\mathbf{x} \in \mathbb{R}^D$ is the D -dimensional vector of design variables, with $x_k^{\min}$ and $x_k^{\max}$ as the lower and upper bounds of each component, and $c(\mathbf{x}) \leq 0$ denotes additional practical constraints. Each evaluation of $J(\mathbf{x})$ requires a full-field reservoir simulation, which typically takes several minutes or hours to complete. As a result, the problem is computationally expensive, and the affordable number of function evaluations is severely limited. This motivates the use of surrogate-assisted evolutionary algorithms [22,23], which can efficiently locate high-quality optimal solutions within a limited simulation budget.

2.4 Embedded Discrete Fracture Model

To evaluate different fracturing schemes and control patterns, dynamic reservoir simulation is essential. This study adopts the EDFM, which embeds fractures into a fixed matrix grid and uses non-neighboring connections (NNCs) to compute fracture-matrix flow exchange, preserving geometric accuracy without altering the underlying grid topology. Fig. 1 illustrates the EDFM discretization and NNC-based modeling workflow with a representative case. Fig. 1a presents the fundamental geometric discretization, showing a 2×2 Cartesian matrix grid intersected by two hydraulic fractures. The fractures are automatically split into four segments: F1–F4 at matrix grid boundaries, creating three distinct scenarios that require NNCs: flow between embedded fracture segments and their host matrix cells, continuity across segments of the same physical fracture, and fluid communication at the intersection of two fractures. Fig. 1b visualizes the corresponding NNC connection topology, where colored lines distinguish different flow links: green lines represent fracture-matrix NNCs, red lines denote intra-fracture NNCs that maintain continuity across grid-split fracture segments, and the orange and blue line indicates the NNC between intersecting fracture segments. Black lines are also included to show standard adjacent matrix connections, highlighting the difference between conventional grid links and EDFM-specific NNCs.

We implement the EDFM model within the open-source MATLAB Reservoir Simulation Toolbox (MRST) [24–26], which provides built-in functionalities including automatic fracture discretization on Cartesian grids, NNC transmissibility calculation, and two-phase black oil simulation, forming a reliable foundation for subsequent NPV optimization. Fig. 1c shows the full-scale reservoir model constructed using this framework, including the $500 \text{ m} \times 500 \text{ m}$ Cartesian matrix grid, multi-stage hydraulic fractures, the well trajectory passing through fracture stage centers, and the locations of injection and production wells. This subfigure provides an intuitive view of the integrated fracture-grid-well communication in the field-scale simulation.

3 Related Techniques

This section briefly reviews the fundamental techniques that constitute the building blocks of the proposed AMSADE algorithm: radial basis function networks, polynomial response surfaces, and deep neural networks with uncertainty quantification. The three surrogates form a complementary heterogeneous pool differing in approximation style, smoothness, and uncertainty representation. Their adaptive selection and coordination are detailed in Section 4.

3.1 Radial Basis Function Model

A radial basis function (RBF) model [27] approximates an unknown function $f(\mathbf{x})$ by a weighted sum of radially symmetric basis functions. Given N evaluated points $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the predictor is:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N \omega_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (7)$$

where we adopt the Gaussian basis function $\phi(r) = \exp(-r^2/\sigma^2)$. The shape parameter is set to $\sigma^2 = d_{\max}(DN)^{-1/D}$, with $d_{\max}$ being the maximal pairwise distance among training inputs. The weight vector $\boldsymbol{\omega} = [\omega_1, \dots, \omega_N]^\top$ is solved from $\Phi\boldsymbol{\omega} = \mathbf{y}$, where $\Phi_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$. Because the kernel matrix is positive definite for the Gaussian kernel, a unique solution exists. RBF networks are insensitive to problem dimensions and provide accurate interpolation with limited training data, making them suitable for moderate- to high-dimensional expensive optimization.

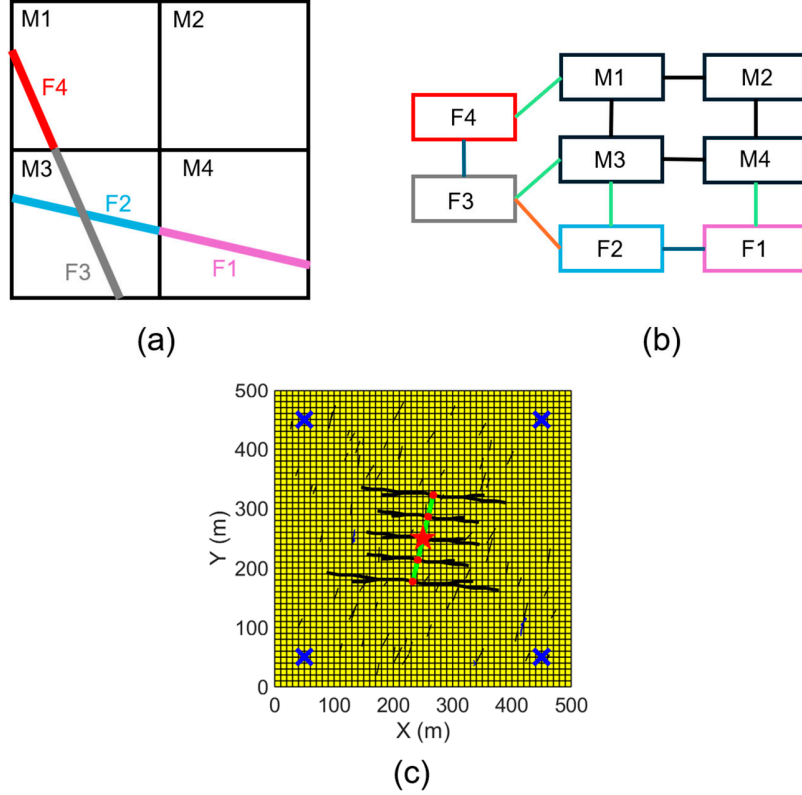


Figure 1: Illustration of (a) the EDFM discretization, (b) NNC topology, and (c) reservoir model.

3.2 Polynomial Response Surface

The polynomial response surface (PRS) model [28] the objective via a low-order polynomial. We employ the second-order (quadratic) form as follows:

$$\hat{f}_{\text{PRS}}(\mathbf{x}) = \beta_0 + \sum_{k=1}^D \beta_k x_k + \sum_{k=1}^D \sum_{l=k}^D \beta_{kl} x_k x_l \quad (8)$$

The coefficients $\boldsymbol{\beta}$ estimated by least-squares regression on the evaluated archive. PRS is computationally cheap and captures the global trend. In high dimensions its flexibility is limited, but its smoothness avoids overfitting and offers a stable baseline.

3.3 Deep Neural Network with Uncertainty Quantification

Deep neural networks (DNNs) [29] can capture complex nonlinear relationships. We design a feed-forward network that simultaneously outputs the predicted objective function, aleatoric data uncertainty, and a low-dimensional latent embedding. The architecture comprises a shared trunk of three dense layers

with 256, 256, and 128 units, ReLU (Rectified Linear Unit) activation, batch normalization, and dropout, followed by three heads: a regression head that produces a scalar prediction $\hat{y}$, an uncertainty head that outputs log-variance $\log \sigma^2$, and an embedding head that generates a latent space for diversity sampling.

Uncertainty is estimated via Monte Carlo (MC) dropout [30]. Dropout remains active during inference, and the network is evaluated T times. The mean prediction and total uncertainty can be calculated as follows:

$$\sigma_{\text{total}}^2 = \frac{1}{T} \sum_{t=1}^T (\hat{y}_t - \hat{y})^2 + \frac{1}{T} \sum_{t=1}^T \exp(\log \sigma_t^2) \quad (9)$$

The total standard deviation σ_{total} constructs the upper confidence bound $\text{UCB}(\mathbf{x}) = \hat{y} + \kappa \sigma_{\text{total}}$ for convergence-driven sampling.

4 Proposed Methods

The proposed adaptive multi-surrogate-assisted differential evolution (AMSADE) is designed to solve the computationally expensive high-dimensional joint optimization of hydraulic fracturing parameters and time-varying production schedules. AMSADE integrates three key components: (1) a surrogate model pool consisting of three heterogeneous surrogates; (2) a two-level multi-armed bandit (TL-UCB) mechanism that adaptively selects the most suitable surrogate model at the upper level and the best infill sampling strategy at the lower level; and (3) a differential evolution (DE) candidate generator that produces diverse offspring for subsequent surrogate screening. The overall framework is illustrated in Fig. 2.

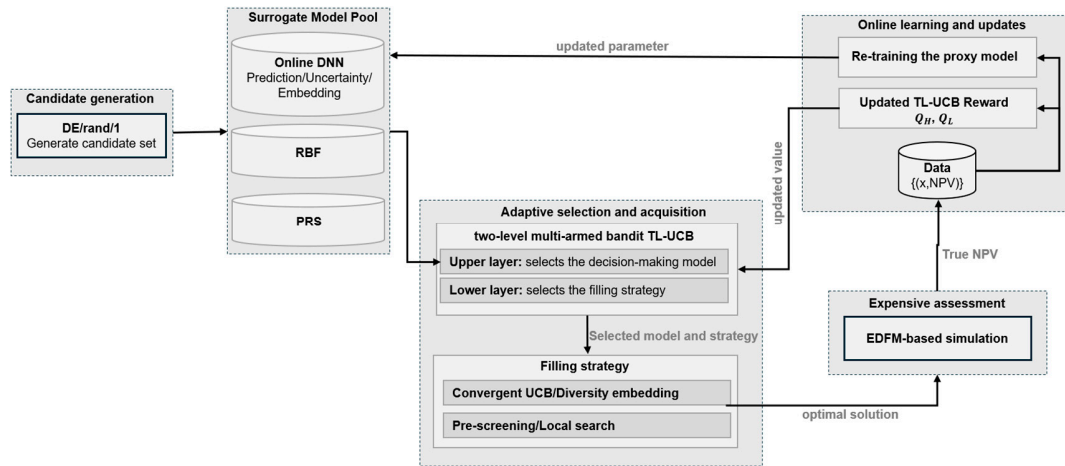


Figure 2: Overall framework of the proposed AMSADE algorithm.

Specifically, the DE algorithm generates candidate solutions, which are initially screened by the heterogeneous surrogate model pool. To alleviate the computational burden of high-fidelity evaluations, the TL-UCB mechanism adaptively selects the optimal surrogate model and infill strategy to identify the most promising candidate solution for expensive numerical simulation. Upon executing the simulation, the training database is dynamically expanded. Concurrently, both the surrogate models and the multi-armed bandit rewards are updated online, thereby establishing a closed-loop adaptive optimization workflow. The novelty of AMSADE lies in its application of an adaptive multi-surrogate-assisted evolutionary framework to the coupled fracturing–scheduling optimization problem. Unlike most existing surrogate-assisted DE algorithms, the AMSADE algorithm constructs a pool of surrogate models and adaptively selects both the surrogate model and its corresponding infill strategy. The following subsections detail each module: the

candidate generation strategy, the surrogate model pool and online training, the TL-UCB adaptive selection mechanism, and the infill criteria.

4.1 Candidate Generation

The initial population of pop_size solutions is generated by random uniform sampling in the parameter space, ensuring coverage of the prescribed parameter bounds. In each subsequent iteration, a diverse set of candidate solutions is required for surrogate screening. We employ the classical DE/rand/1 mutation strategy to generate these candidates, as it has been shown to maintain good population diversity in high-dimensional spaces.

Given the current population $P = \{\mathbf{p}_1, \dots, \mathbf{p}_{\text{pop_size}}\}$, for each target vector $\mathbf{p}_i$ three distinct individuals $\mathbf{p}_{r1}, \mathbf{p}_{r2}, \mathbf{p}_{r3}$ are randomly selected from P . A mutant vector is formed as

$$\mathbf{v}_i = \mathbf{p}_{r1} + F \cdot (\mathbf{p}_{r2} - \mathbf{p}_{r3}) \quad (10)$$

where the scaling factor is set to $F = 0.3$ to avoid excessive perturbation in the high-dimensional setting. Binomial crossover with rate $CR = 0.5$ is then applied to produce a trial vector $\mathbf{u}_i$. If any component of $\mathbf{u}_i$ violates the design bounds, it is repaired by reflection bouncing back into the feasible region, and subsequently clipped to the bounds to strictly enforce feasibility.

This procedure is repeated for every member of the current population, generating exactly pop_size candidate solutions. These candidates are passed to the infill selection module, where they are screened by the selected surrogate model and infill strategy combination.

4.2 Surrogate Model Pool and Online Training

To balance prediction accuracy, uncertainty quantification, and the ability to measure diversity, we construct a surrogate model pool with three heterogeneous surrogates. The RBF model provides a fast and stable prediction when data is scarce, and the PRS surrogate can capture the global low-order trends. As shown in Fig. 3, the DNN surrogate outputs three quantities simultaneously: (i) the predicted NPV $\hat{y}$, (ii) the total predictive variance σ^2 , and (iii) a low-dimensional latent embedding $\mathbf{z}$. This design enables two unique infill strategies (convergence UCB and diversity exploration) that are impossible with the other two surrogates. The network receives an input $\mathbf{x}$ and passes it through shared hidden layers with batch normalization and dropout. Three parallel heads then produce the predicted NPV $\hat{y}$, the log-variance $\log \sigma^2$ (representing predictive uncertainty), and a low-dimensional latent embedding $\mathbf{z}$. During online training, the three outputs are jointly optimized via a composite loss comprising the negative log-likelihood (NLL) of the prediction and an embedding regularization term $\lambda \|\mathbf{z}\|^2$. The Adam optimizer updates the network weights.

All surrogates share the same database $D = \{(\mathbf{x}^i, y^i)\}_{i=1}^N$ of previously evaluated solutions. The RBF and PRS surrogates are refitted on the entire database whenever new data arrives. DNN is retrained online at regular intervals using mini-batch gradient descent with a negative log-likelihood loss and a small regularization on the embedding.

4.3 Infill Criteria for Surrogate Models

Four distinct infill criteria are designed to leverage the specific predictive outputs of each surrogate model and to identify the most promising solution from the DE-generated candidate set for high-fidelity

numerical evaluation. Given a set of candidate solutions C generated by DE, the selected (model, strategy) picks one promising candidate for real evaluation.

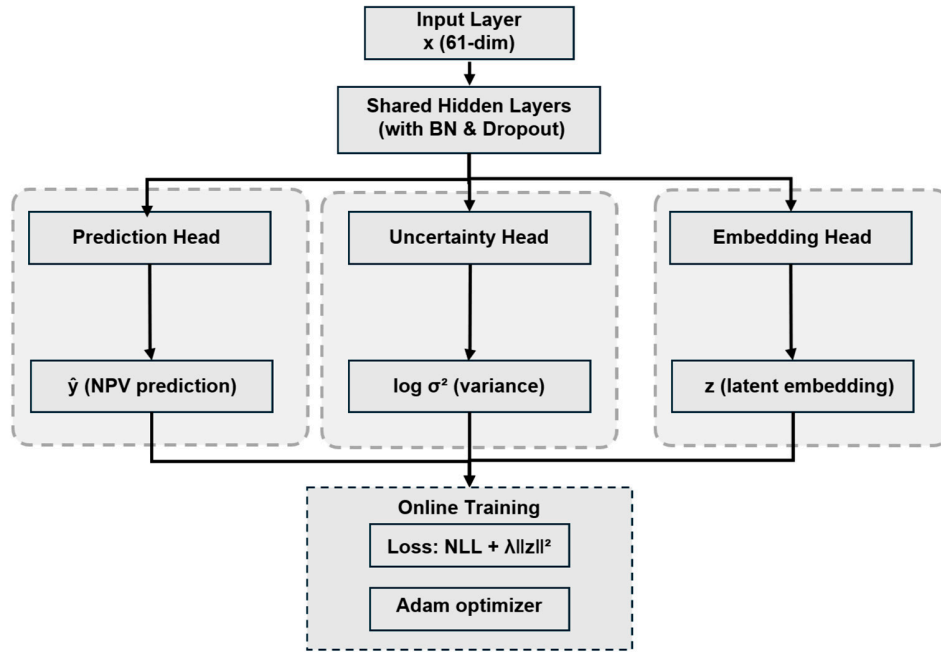


Figure 3: Architecture of the multi-output DNN surrogate model structure.

DNN-Based Infill Strategies. For the DNN surrogate model, two distinct infill strategies are formulated to leverage their predictive and representation learning capabilities. The first is the UCB, where the DNN predicts the mean $\hat{y}(x)$ and total standard deviation $\sigma(x)$ via Monte Carlo dropout for each candidate solution. The upper confidence bound is $UCB(x) = \hat{y}(x) + 2\sigma(x)$. The candidate that maximizes UCB is selected, promoting exploitation guided by uncertainty. The second strategy is Diversity Embedding, which harnesses the DNN to project each candidate into a low-dimensional latent space as a low-dimensional embedding $z(x)$, and the candidate that maximizes the Euclidean distance between its embedding and those of the current population is then selected, thereby promoting global exploration within the learned feature space.

RBF/PRS-Based Infill Strategies. For the RBF and PRS models, the infill strategies focus on exploiting geometric landscapes and spatial distributions. The prescreen strategy operates as a pure exploitation mechanism, wherein the RBF or PRS model predicts the NPV for all candidates, and the individual with the highest predicted value is directly selected for high-fidelity evaluation. Conversely, the local-search strategy balances local refinement with spatial exploration by first identifying the top five candidates with the highest predicted NPVs; among these elite candidates, the one that exhibits the maximum distance to the current population within the original decision space is selected, preventing premature convergence in localized optima.

As shown in Fig. 4, for each panel, black filled circles represent evaluated population members, gray open circles denote candidate solutions generated by DE/rand/1, and the red star indicates the candidate selected by the corresponding strategy. For the DNN surrogate, the Convergence UCB strategy selects the candidate maximizing the upper confidence bound to favor high predicted NPV and large uncertainty, whereas the Diversity Embedding approach chooses the point with the largest distance in the latent space to promote exploration. For RBF and PRS models, the prescreen method directly identifies the candidate

with the highest predicted value. Concurrently, the local-search strategy filters the top five candidates (orange diamonds) with the highest predicted NPV and selects the furthest one from its nearest evaluated point (indicated by dashed lines), balancing exploitation and exploration.

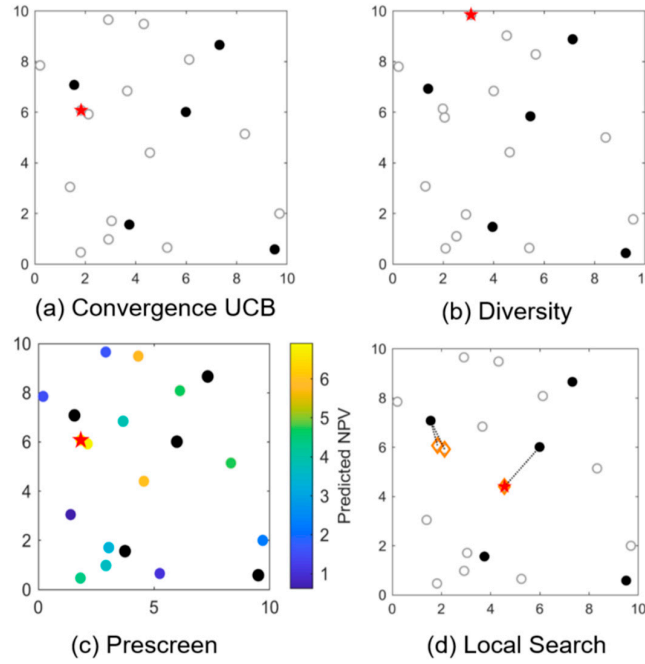


Figure 4: Illustration of the different infill criteria.

4.4 Adaptive Selection Mechanism of Surrogate Model and Infill Strategy

A key innovation of the proposed AMSADE algorithm is the two-level multi-armed bandit (TL-UCB) mechanism that dynamically selects which surrogate model to use and which infill criterion to apply at each iteration. The approach is inspired by the Upper Confidence Bound (UCB) algorithm for multi-armed bandits [31], extended to a hierarchical structure. Let $M = \{\text{RBF}, \text{PRS}, \text{DNN}\}$ denote the pool of surrogate models. For each model $m \in M$, a set of available infill strategies S_m is defined in Section 4.3.

The adaptive selection process is executed hierarchically across two distinct levels, consisting of a high-level selection that determines an individual surrogate model from the predefined heterogeneous pool, followed by a low-level selection which, contingent upon the activated model, dynamically chooses the most appropriate infill criterion from the corresponding strategy space. Each arm is associated with an estimated quality Q and a counter T of how many times it has been selected.

At iteration t , the UCB value for the high-level arm of model m is computed as:

$$U_H(m) = Q_H(m) + \sqrt{\frac{\alpha \ln t}{T_H(m)}} \quad (11)$$

where α is a exploration–exploitation trade-off parameter (set to 2.5 in our work). The model with the highest U_H is selected.

Subsequently, for the chosen model m , the UCB for each of its strategies is calculated:

$$U_L(m, s) = Q_L(m, s) + \sqrt{\frac{\alpha \ln t}{T_L(m, s)}} \quad (12)$$

The strategy with the highest U_L is used to select the next candidate for expensive evaluation.

After the true NPV y_{new} is obtained, the rewards are updated based on the ranking of the new point within the current population (the NP best individuals in the archive). The population $\mathcal{P}$ consists of the top NP points according to their true NPV. The rank of the new point among $\mathcal{P} \cup \{y_{\text{new}}\}$ is denoted by r (where $r = 1$ indicates the best). The low-level reward for the chosen strategy is defined as

$$r_L = -\frac{1}{NP}r + \frac{NP+1}{NP} \quad (13)$$

This formulation gives a reward close to 1 when the new point ranks highly (i.e., small r) and close to 0 when it ranks poorly. The quality value for the low-level arm (m, s) is updated incrementally:

$$Q_L^{\text{new}}(m, s) = \frac{Q_L^{\text{old}}(m, s) \cdot T_L(m, s) + r_L}{T_L(m, s) + 1} \quad (14)$$

The change in the low-level quality, $\Delta = Q_L^{\text{new}}(m, s) - Q_L^{\text{old}}(m, s)$, is used to update the high-level quality of model m :

$$Q_H^{\text{new}}(m) = Q_H^{\text{old}}(m) + \frac{\Delta}{|\delta_m|} \quad (15)$$

Thus, the high-level rewards accumulate the average improvement contributed by the strategies of each model, guiding the selection toward models that consistently yield high-quality candidates.

As shown in Fig. 5, the upper level selects one of the three surrogate models via the UCB criterion. The lower level then selects an infill strategy from the available options for the chosen model. Operating on a shared candidate pool, the TL-UCB framework adaptively switches among the surrogate models and infill strategies throughout the optimization. At each iteration, the candidate selected by the active infill criterion undergoes high-fidelity evaluation via the EDFM simulator, with the resulting NPV appended to the database to trigger an online surrogate update. This closed-loop ‘sample-evaluate-update’ paradigm progressively refines the surrogates mapping of the NPV landscape, steering subsequent searches toward high-NPV regions and efficiently converging toward the optimum.

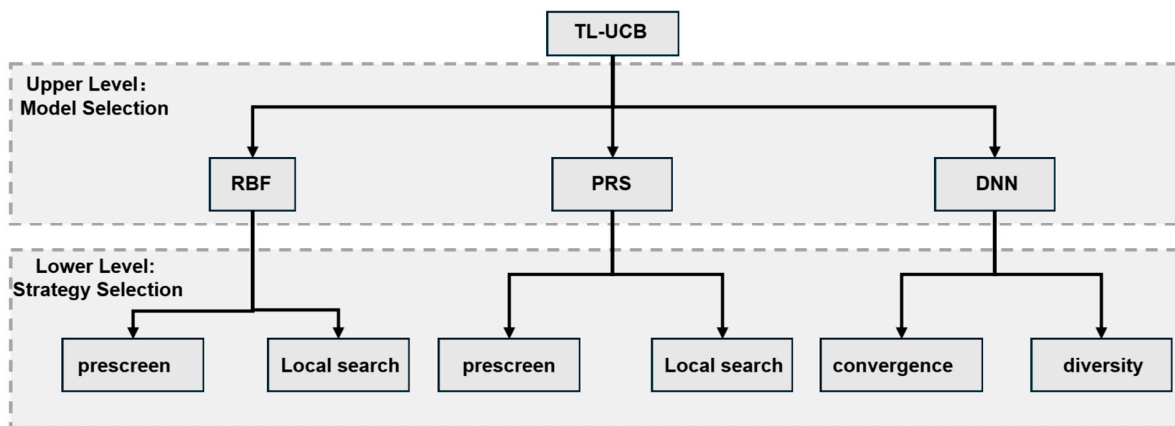


Figure 5: Two-level UCB architecture for adaptive surrogate and infill strategy selection.

5 Numerical Experiments

The experimental evaluation is divided into two parts: performance benchmarking on analytical test functions, and application to a coupled fracturing design and production control problem. In both scenarios, the proposed AMSADE algorithm is rigorously evaluated against three baselines under an identical computational budget. The first baseline, Random Search (RS), uniformly draws new solutions from the design space at each iteration without leveraging surrogate models or prior landscape knowledge, serving as a purely stochastic baseline. The second is the standard Differential Evolution (DE) algorithm, which acts as a model-free heuristic engine that relies solely on mutation and crossover operations to maintain population diversity. The third is an offline DNN-assisted DE algorithm, which trains a deep neural network surrogate exclusively on the initial samples and utilizes it as a static proxy thereafter.

5.1 Test on Benchmark Function

The Ackley function was adopted as a representative benchmark to simulate computationally expensive scenarios, serving to evaluate the optimization performance of the proposed AMSADE algorithm against the three baseline methods. The optimization task was formulated as maximizing the 7-dimensional Ackley function within the search domain $[-5, 5]^7$. For all algorithms, the maximum number of expensive evaluations was set to 300, and an initial population of 20 points was generated via Latin hypercube sampling and shared across all methods to ensure a fair comparison. For offline DNN-assisted DE, 300 initial solutions are generated to construct a DNN model for the optimization. Each algorithm was run independently 10 times, and the mean and standard deviation of the results were recorded. Fig. 6 illustrates the landscape of the Ackley function, this function is multimodal and has many local optima, defined as follows:

$$f(\mathbf{x}) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1) \quad (16)$$

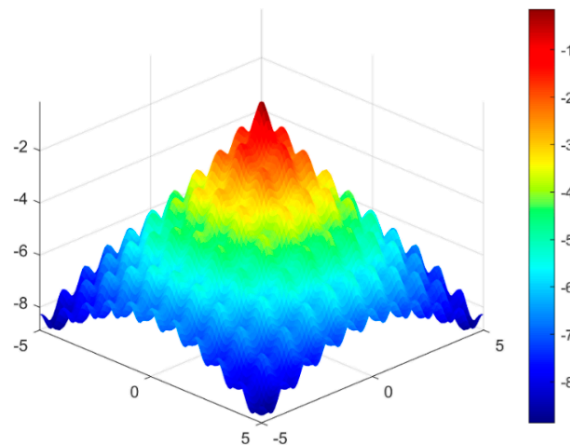


Figure 6: The landscape of 2-D Ackley function.

The parameters are set as $a = 20$, $b = 0.2$ and $c = 2\pi$. The function has a global minimum $f(0) = 0$ located at the origin. Fig. 7 presents the comparative convergence curves and box-plots of each algorithm on the Ackley function. For this benchmark test, AMSADE significantly outperforms standard DE, RS, and offline DNN-DE in terms of both convergence velocity and optimization stability, delivering the superior

overall performance. Standard DE achieves the second-best results, whereas Random Search and Offline DNN-DE exhibit severely limited optimization efficacy, with the offline DNN variant yielding the poorest outcomes. This failure stems from the highly complex, multimodal landscape of the Ackley function, where the initial 300 samples are profoundly insufficient to construct a reliable static surrogate model. Conversely, the proposed AMSADE algorithm progressively approaches the global optimum within a few iterations. This underscores that its online dynamic updating mechanism effectively refines the surrogate pool's prediction accuracy over time, thereby successfully guiding the DE search toward the optimal parameter space.

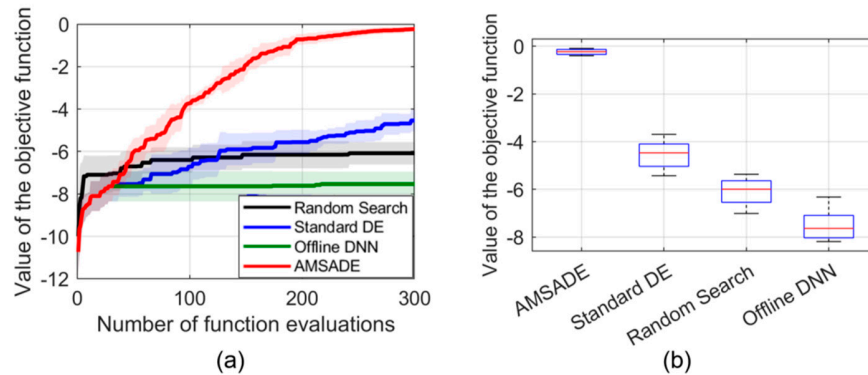


Figure 7: The (a) convergence curve and (b) distribution of optimal function value after 300 function evaluations with 10 independent runs for the Ackley function.

5.2 Joint Optimization of a 2D Reservoir Model

To investigate the effectiveness of the proposed joint optimization of hydraulic fracturing design and production control, a comprehensive case study was conducted on a two-dimensional (2D) tight reservoir model. The foundational reservoir properties are summarized in Table 1. An inverted five-spot well pattern was implemented within the reservoir domain, consisting of four water injection wells positioned at the corners and a central horizontal production well. Assuming a homogeneous formation where the hydraulic fractures fully penetrate the reservoir thickness, the model was simplified into a 2D plane; the well configuration is illustrated in Fig. 8.

Table 1: Parameters of the 2D reservoir case.

Parameter	Value
Region size	500 m × 500 m × 10 m
Matrix porosity	0.08
Matrix permeability	0.5 mD
Fracture porosity	0.2
Fracture permeability	Natural fractures: 1 D; Hydraulic fractures: 1 D ~ 100 D
Fracture aperture	0.01 m
Fluid phase	Two-phase (oil/water)
Initial pressure	15 MPa
Initial water saturation	0.3
Viscosity of oil	1.5 cp
Viscosity of water	1 cp

The total project lifetime is set to 15 years (5400 days), which is evenly divided into 10 control cycles of 540 days each. Within each cycle, the well controls are held constant. The injection wells are operated under rate control, with the injection rate for each well-being an independent variable that can vary

between 0 and 10 STB/D across different control cycles. The production well is operated under bottom-hole pressure (BHP) control, with the BHP allowed to vary between 80 and 150 Barsa. Internal time stepping is handled automatically by the MRST simulator using ramp-up from an initial step of approximately 60 days, ensuring numerical stability while capturing the multiphase flow dynamics. The economic objective NPV is calculated from the simulated oil and water production rates using an oil price of USD 50/STB, a water treatment cost of USD 10/STB, a water injection cost of USD 10/STB, a discount rate of 1% per year and a fixed capital cost of USD 100,000. These economic parameters, together with the well control ranges, define the 61-dimensional coupled optimization problem.

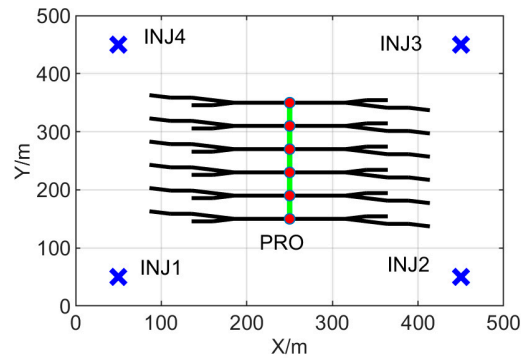


Figure 8: The well pattern and fracturing fracture schematic diagram of the 2D reservoir case.

5.3 Results and Analysis of Joint Optimization

Each algorithm was run independently for 5 times, Fig. 9 compares the optimization results of RS, standard DE, the offline DNN-assisted DE, and the proposed AMSADE algorithm on the fracturing–scheduling joint optimization problem. Fig. 9a shows the evolution of the best NPV over function evaluations, while Fig. 9b summarizes the final best NPV after 200 simulator calls. RS exhibits no meaningful improvement, ending with an average best NPV of 3.42×10^7 USD ($\pm 6.45 \times 10^5$). Standard DE achieves moderate gains, reaching 3.60×10^7 USD on average ($\pm 3.32 \times 10^5$). The offline DNN surrogate attains an average of 3.81×10^7 USD ($\pm 1.02 \times 10^6$), but its progress levels off after early iterations. By contrast, AMSADE demonstrates rapid and robust convergence: it surpasses all baselines within the first 50 function evaluations and continues to improve through adaptive sampling, ultimately delivering the highest average best NPV of 4.15×10^7 USD ($\pm 7.70 \times 10^5$) and the single-best run reaching 4.23×10^7 USD. The consistently lower standard deviation and higher mean confirm that AutoSAEA reliably balances exploration and exploitation under tight computational budgets. Fig. 9 further underscores that AMSADE achieves the highest net present value while maintaining efficiency in costly simulator calls, highlighting its practical advantage for fracturing parameter optimization.

A direct comparison of the convergence curves in Fig. 9 clearly illustrates the computational efficiency of AMSADE. All four algorithms were allocated exactly the same budget of 200 high-fidelity simulations, and their total wall-clock times were dominated by these expensive evaluations. The surrogate-training overhead of AMSADE, which occurred every five evaluations, accounted for only a negligible fraction of the overall cost. Random search stagnates early at a low NPV, and standard DE improves only marginally, effectively wasting most of the evaluation budget on unpromising regions. The offline DNN exhibits a rapid initial surge but quickly plateaus after approximately 30 evaluations, failing to translate its early lead into further gains. In stark contrast, AMSADE continues to improve steadily throughout the entire 200-evaluation budget, overtaking the offline DNN around the 80th evaluation and ultimately reaching the

highest NPV. This sustained upward trend demonstrates that AMSADE converts each additional simulation into meaningful progress, achieving substantially superior solution quality for the same computational expenditure. The convergence behavior therefore confirms that the modest online learning cost of AMSADE yields a decisive advantage in cost-normalized optimization performance.

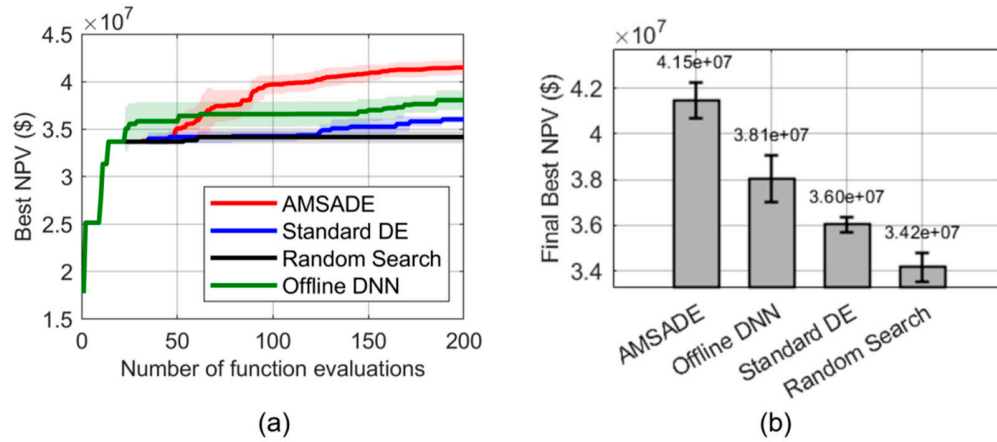


Figure 9: The (a) convergence curve and (b) distribution of optimal function value after 200 function evaluations with 5 independent runs for the joint optimization problems.

A comparison of the optimized outcomes is provided in Fig. 10 below, which displays the 15-year post-production oil saturation distributions and the corresponding fracture topologies for each algorithm. As can be seen in Fig. 10a, the AMSADE-derived design establishes a highly synergistic and well-connected fracture network, which achieves the most extensive sweep efficiency and leaves minimal residual oil. Conversely, Fig. 10b reveals that the OfflineDNN scenario generates a sub-optimal fracture deployment, thereby resulting in elevated oil saturation. Furthermore, compared to these results, both Standard DE and Random Search exhibit sparse or poorly oriented fracture conduits, failing to ensure sufficient reservoir sweep, as respectively depicted in Fig. 10c,d. As a consequence, these saturation patterns confirm that the proposed AMSADE-based joint optimization method simultaneously secures maximized economic returns and superior volumetric sweeping.

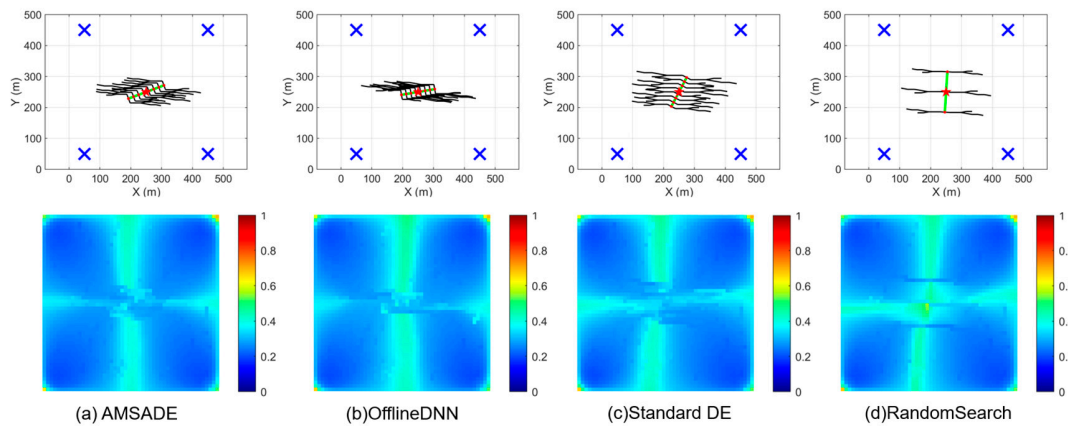


Figure 10: Comparison of fracture distribution and oil saturation results of four algorithms.

Fig. 11 presents the predictive accuracy of the heterogeneous surrogate pool at 30 and 200 function evaluations. As illustrated in Fig. 11a, after 30 evaluations, the surrogate pool exhibits limited predictive

capability ($R^2 = -0.273$) due to the scarcity of training data. However, after 200 evaluations, as shown in Fig. 11b, the predicted values are tightly clustered along the diagonal line, achieving a high deterministic accuracy of $R^2 = 0.86$. This contrast demonstrates that the adaptive surrogate pool, guided by the TL-UCB bandit mechanism, progressively captures the complex, non-linear NPV landscape and provides highly reliable guidance for the heuristic search.

Figs. 12 and 13 display the injection rate schedules for the four injectors and the BHP profiles of production well over 10 control steps optimized by four algorithms. The AMSADE algorithm delivers a highly sophisticated and balanced control scheme: it initiates with elevated injection rates to rapidly replenish reservoir energy, followed by stabilized, moderate adjustments to sustain long-term sweep efficiency. Concurrently, its production BHP profile is regulated in a progressive manner, eliminating severe fluctuations or abrupt drawdowns that could destabilize fluid dynamics. By comparison, offline DNN and standard DE suffer from notable oscillations in both injection and pressure controls. These operational dynamics strongly corroborate the previously observed sweep efficiencies and economic outcomes.

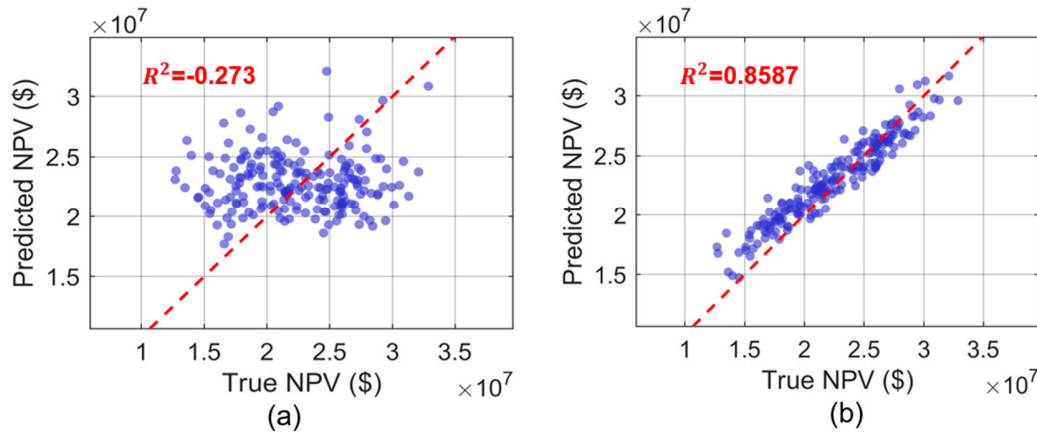


Figure 11: Predictive accuracy of the surrogate pool (a) before and (b) after online learning.

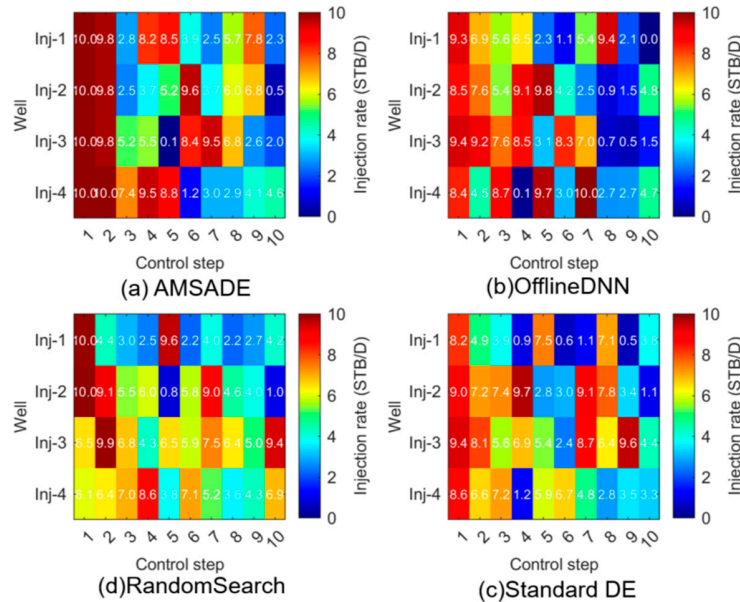


Figure 12: Comparison of injection rates of the optimal solution.

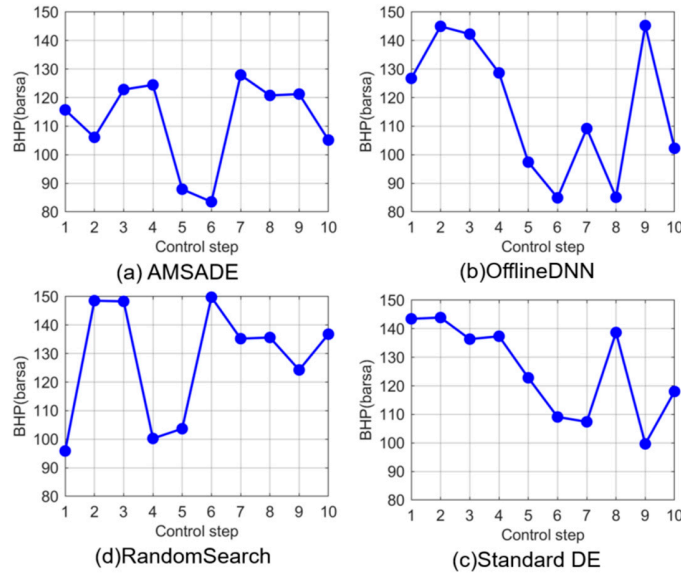


Figure 13: Comparison of production BHP profiles of optimal solutions.

Fig. 14 presents the evolution of key production indicators over the reservoir operational life cycle, focusing on water cut and cumulative oil production for the four algorithms. Compared to the cumulative oil production from random search, the standard DE and offline DNN models improved by 1.09% and 0.88%, respectively, after optimization, while the algorithm proposed in this study achieved a 3.29% increase in cumulative oil production after optimization. The water cut profiles clearly underscore the distinct advantages of the AMSADE-optimized policy: it effectively delays water breakthrough and mitigates the early premature water production that plagues the alternative methods. Consequently, AMSADE consistently maintains a leading trajectory in cumulative oil production throughout the production lifespan, culminating in the highest ultimate recovery volume. These performance trends tightly align with the previously observed reservoir sweep dynamics and economic valuations, demonstrating that optimized control parameters of AMSADE ensure superior long-term recovery efficiency and production stability.

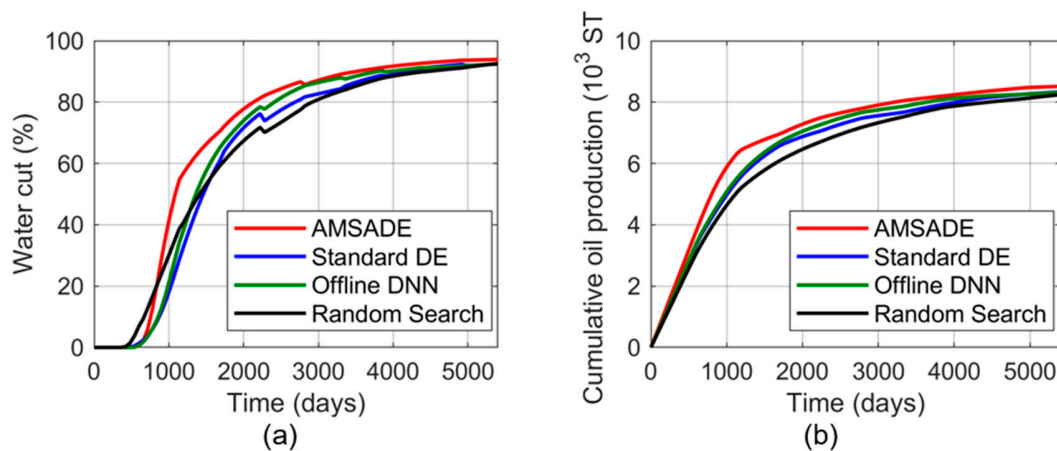


Figure 14: Comparison of (a) water cut and (b) cumulative oil production for the optimal solution.

6 Conclusions

In this work, an adaptive multi-surrogate-assisted differential evolution (AMSADE) framework is developed for the joint optimization of hydraulic fracturing design and time-varying production control. This approach effectively addresses a challenging, high-dimensional mixed-variable decision space that simultaneously governs complex fracture geometries and dynamic well controls. Based on comprehensive mathematical benchmarking and reservoir case studies, the core contributions and conclusions are summarized as follows.

First, a deep neural network surrogate is designed to jointly output the predicted NPV, aleatoric uncertainty, and low-dimensional latent space embeddings, enabling the simultaneous deployment of both convergence-driven and diversity-driven infill strategies within a single, cohesive architecture. Second, by establishing a heterogeneous surrogate pool alongside a hierarchical two-level multi-armed bandit mechanism, the proposed AMSADE algorithm adaptively synchronizes the optimal surrogate-infill pairing at each iteration. This localized online feedback loop systematically balances global exploration and local exploitation throughout the optimization workflow. For the 2D reservoir case, under a constrained budget of 200 high-fidelity numerical simulations, the proposed framework achieves a superior average best NPV across independent runs, consistently outperforming standard differential evolution, static offline DNN proxies, and random search.

Despite the promising results, several limitations of the present study should be acknowledged. The current AMSADE framework is designed for single-objective optimization and has only been validated on a two-dimensional synthetic reservoir model with homogeneous matrix properties and a fixed set of natural fractures. Geological heterogeneity, three-dimensional flow effects, and field-scale complexities have not yet been incorporated. These limitations point to clear directions for future work. Future research will focus on extending this adaptive framework to multi-objective optimization contexts that account for complex geological uncertainties. Additionally, efforts will be directed toward evaluating its computational scalability and algorithmic robustness on field-scale hydrocarbon reservoirs.

Acknowledgement: Not applicable.

Funding Statement: This work is supported by the National Natural Science Foundation of China under Grants 52404037, 52304035, and 52174031, and the Natural Science Basic Research Program of Shaanxi grant number 2024JC-YBQN-0506.

Author Contributions: Conceptualization, Xiaopeng Ma, Bin Zhang, Jinsheng Zhao, Di Zhu; methodology, Xiaopeng Ma, Bin Zhang, Jinsheng Zhao, Di Zhu; investigation, Xiaopeng Ma, Bin Zhang, Jinsheng Zhao, Di Zhu; writing—original draft, Xiaopeng Ma, Bin Zhang; writing—review & editing, Xiaopeng Ma, Bin Zhang, Jinsheng Zhao, Di Zhu; supervision, Xiaopeng Ma, Jinsheng Zhao, Di Zhu; visualization, Bin Zhang, Di Zhu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data will be made available on request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no competing financial interests or personal relationships that could have influenced the work reported in this paper.

References

1. Wang W, Shahvali M, Su Y. A semi-analytical fractal model for production from tight oil reservoirs with hydraulically fractured horizontal wells. *Fuel*. 2015;158:612–8. [[CrossRef](#)].

2. Xi Y, Li J, Liu G, Tao Q, Lian W. A new numerical investigation of cement sheath integrity during multistage hydraulic fracturing shale gas wells. *J Nat Gas Sci Eng.* 2018;49:331–41. [[CrossRef](#)].
3. Lu Y, Li H, Lu C, Wu K, Chen Z, Wang K, et al. The effect of completion strategy on fracture propagation from multiple cluster perforations in fossil hydrogen energy development. *Int J Hydrogen Energy.* 2019;44(14):7168–80. [[CrossRef](#)].
4. Qiao J, Tang X, Hu M, Rutqvist J, Liu Z. The hydraulic fracturing with multiple influencing factors in carbonate fracture-cavity reservoirs. *Comput Geotech.* 2022;147:104773. [[CrossRef](#)].
5. Yu W, Sepehrnoori K. Optimization of multiple hydraulically fractured horizontal wells in unconventional gas reservoirs. *J Petrol Eng.* 2013;2013:151898. [[CrossRef](#)].
6. Yu W, Luo Z, Javadpour F, Varavei A, Sepehrnoori K. Sensitivity analysis of hydraulic fracture geometry in shale gas reservoirs. *J Petrol Sci Eng.* 2014;113:1–7. [[CrossRef](#)].
7. Luo H, Xie J, Huang L, Wu J, Shi X, Bai Y, et al. Multiscale sensitivity analysis of hydraulic fracturing parameters based on dimensionless analysis method. *Lithosphere.* 2022;12:9708300. [[CrossRef](#)].
8. Hu Z, Wang H, Zheng S. Numerical modelling of energy storage using hydraulic fracturing in shale formations. *Geoenergy Sci Eng.* 2025;244:213424. [[CrossRef](#)].
9. Ma X, Plaksina T, Gildin E. Optimization of placement of hydraulic fracture stages in horizontal wells drilled in shale gas reservoirs. In: *Proceedings of the Unconventional Resources Technology Conference*; 2013 Aug 12–14; Denver, CO, USA. p. 1479–89. [[CrossRef](#)].
10. Rammay MH, Awotunde AA. Stochastic optimization of hydraulic fracture and horizontal well parameters in shale gas reservoirs. *J Nat Gas Sci Eng.* 2016;36:71–8. [[CrossRef](#)].
11. Ruiz Maraggi LM, Walsh MP, Lake LW, Male FR. Fast optimization of the net present value of unconventional wells using rapid rate-transient analysis. *SPE J.* 2024;29(11):6500–15. [[CrossRef](#)].
12. Chen G, Luo X, Jiao JJ, Xue X. Data-driven evolutionary algorithm for oil reservoir well-placement and control optimization. *Fuel.* 2022;326:125125. [[CrossRef](#)].
13. Kor P, Hong A, Bratvold R. Reservoir production management with Bayesian optimization: Achieving robust results in a fraction of the time. *SPE J.* 2024;29(2):620–40. [[CrossRef](#)].
14. Wang X, Wang Z, Zhang L, Zhang Z, He Y. A comparison analysis of intelligence algorithms for oil reservoir production optimization. *J Eng Sci Technol Rev.* 2021;14(4):169–78. [[CrossRef](#)].
15. Zhang K, Zhang J, Ma X, Yao C, Zhang L, Yang Y, et al. History matching of naturally fractured reservoirs using a deep sparse autoencoder. *SPE J.* 2021;26(4):1700–21. [[CrossRef](#)].
16. Chen G, Zhang K, Zhang L, Xue X, Ji D, Yao C, et al. Global and local surrogate-model-assisted differential evolution for waterflooding production optimization. *SPE J.* 2020;25(1):105–18. [[CrossRef](#)].
17. Ma X, Zhang K, Zhang L, Wang Y, Wang H, Wang J, et al. A distributed surrogate system assisted differential evolutionary algorithm for computationally expensive history matching problems. *J Petrol Sci Eng.* 2022;210:110029. [[CrossRef](#)].
18. Xiao C, Zhang S, Ma X, Zhou T, Li X. Surrogate-assisted hydraulic fracture optimization workflow with applications for shale gas reservoir development: A comparative study of machine learning models. *Nat Gas Ind B.* 2022;9(3):219–31. [[CrossRef](#)].
19. Do Kim Y, Durlafsky LJ. Neural network surrogate for flow prediction and robust optimization in fractured reservoir systems. *Fuel.* 2023;351:128756. [[CrossRef](#)].
20. Zhou B, Chen Z, Song Z, Tang Z, Wang B, Olorode O. A new analytically modified embedded discrete fracture model for pressure transient analysis in fluid flow. *J Hydrol.* 2024;636:131330. [[CrossRef](#)].
21. Zhang J, Kamenov A, Zhu D, Hill ADD. Laboratory measurement of hydraulic-fracture conductivities in the Barnett Shale. *SPE Prod Oper.* 2014;29(3):216–27. [[CrossRef](#)].
22. Liang J, Lou Y, Yu M, Bi Y, Yu K. A survey of surrogate-assisted evolutionary algorithms for expensive optimization. *J Membr Comput.* 2025;7(2):108–27. [[CrossRef](#)].
23. He C, Zhang Y, Gong D, Ji X. A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. *Expert Syst Appl.* 2023;217:119495. [[CrossRef](#)].
24. Poli REB, Barbosa Machado MV, Sepehrnoori K. Advancements and perspectives in embedded discrete fracture models (EDFM). *Energies.* 2024;17(14):3550. [[CrossRef](#)].

25. Lie KA, Krogstad S, Ligaarden IS, Natvig JR, Nilsen HM, Skafflestad B. Open-source MATLAB implementation of consistent discretisations on complex grids. *Comput Geosci*. 2012;16(2):297–322. [[CrossRef](#)].
26. Bao K, Lie KA, Møyner O, Liu M. Fully implicit simulation of polymer flooding with MRST. *Comput Geosci*. 2017;21(5):1219–44. [[CrossRef](#)].
27. Li F, Shang Z, Liu Y, Shen H, Jin Y. Inverse distance weighting and radial basis function based surrogate model for high-dimensional expensive multi-objective optimization. *Appl Soft Comput*. 2024;152:111194. [[CrossRef](#)].
28. Zhao Y, Ye S, Chen X, Xia Y, Zheng X. Polynomial Response Surface based on basis function selection by multitask optimization and ensemble modeling. *Complex Intell Syst*. 2022;8(2):1015–34. [[CrossRef](#)].
29. Kontou MG, Asouti VG, Giannakoglou KC. DNN surrogates for turbulence closure in CFD-based shape optimization. *Appl Soft Comput*. 2023;134:110013. [[CrossRef](#)].
30. Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *Int Conf Mach Learn*. 2016;48:1050–9.
31. Liu X, Derakhshani M, Lambbotharan S, van der Schaar M. Risk-aware multi-armed bandits with refined upper confidence bounds. *IEEE Signal Process Lett*. 2021;28:269–73. [[CrossRef](#)].