Petroleum Science 23 (2026) 2735–2757

Contents lists available at ScienceDirect

Petroleum Science

journal homepage: www.keaipublishing.com/en/journals/petroleum-science

Original Paper

Collaborative optimization of well operations and adjustment strategies in waterflooding reservoirs using an enhanced adaptive differential evolution algorithm

Xian-Min Zhang [a,b,*], Jian-Gang Yang [a,b], Qi-Hong Feng [a,b], Ya-Wei Hou [c], Lei Zhang [c]

[a] State Key Laboratory of Deep Oil and Gas, China University of Petroleum (East China), Qingdao, 266580, Shandong, China
[b] School of Petroleum Engineering, China University of Petroleum (East China), Qingdao, 266580, Shandong, China
[c] Bohai Petroleum Research Institute, Tianjin Branch of CNOOC (China) Ltd., Tianjin, 300450, China

ARTICLE INFO

Article history:
Received 30 July 2025
Received in revised form
24 November 2025
Accepted 12 March 2026
Available online 16 March 2026

Edited by Meng-Jiao Zhou

Keywords:
Waterflooding reservoir
Collaborative optimization
Enhanced adaptive differential evolution algorithm
Logarithmic spiral search
Shannon entropy

ABSTRACT

Efficient optimization of well operations and adjustment strategies in large-scale waterflooding reservoirs is a high-dimensional and complex challenge due to strong decision coupling and reservoir heterogeneity. This study proposes a collaborative optimization framework that integrates multiple adjustment strategies, including infill well drilling, shut-in of low-efficiency wells, and injection-production well conversion. A penalty mechanism is introduced to balance cumulative oil production maximization with minimum production constraints for infill wells. The core contribution is the development of a multi-strategy enhanced adaptive differential evolution algorithm (E-ADE), which incorporates the follower update mechanism of the Sparrow Search Algorithm (SSA) and the logarithmic spiral search strategy of the Whale Optimization Algorithm (WOA) into the differential evolution (DE) framework. By dynamically adjusting differential evolution vectors and adaptively regulating population size across optimization stages, E-ADE effectively balances global exploration and local exploitation, leading to significantly improved convergence speed and optimization accuracy. Benchmark tests on nine multimodal functions demonstrate that E-ADE consistently outperforms classical algorithms, including DE, GA, PSO, WOA, and SSA. The method is further applied to the PUNQ-S3 reservoir model and the S4 block of the W12-2 oilfield under high water-cut conditions. The results indicate that E-ADE enables adaptive optimization of infill well placement, shut-in schemes, and well-type conversions, achieving coordinated improvements in both field-scale production and single-well performance, and substantially enhancing the efficiency of waterflooding development.

1. Introduction

Adjusting well configurations is a key strategy in oilfield development, particularly critical for improving production efficiency in high water-cut reservoirs (Zhang et al., 2015). However, designing an effective adjustment scheme is a complex and systematic process that involves two fundamental aspects: accommodating reservoir heterogeneity and remaining oil distribution (Li et al., 2006), and achieving holistic optimization of the well configuration rather than isolated optimization of individual well groups (Onwunalu and Durlofsky, 2011). This process requires the integrated consideration of multiple factors, including infill well placement, well type selection (injection or production), trajectory design, and decisions regarding well conversion or shut-in. The core challenge lies in simultaneous optimization of these strongly coupled variables to improve both recovery efficiency and overall resource utilization.

Numerous studies have investigated well configuration deployment schemes, which can be broadly classified into three categories: traditional methods, vectorized methods, and optimization theory-based approaches. Traditional methods grounded in reservoir engineering theory (Prasun and Wojtanowicz, 2018; Xu et al., 2012) often

* Corresponding author.
E-mail address: spemin@126.com (X.-M. Zhang).
Peer review under the responsibility of China University of Petroleum (Beijing).

https://doi.org/10.1016/j.petsci.2026.03.030

fails to adequately account for reservoir heterogeneity, resulting in poor adaptation to complex geological structures and compromised oil recovery. Vectorized methods (Chen et al., 2017; Li et al., 2006; Tian et al., 2018) aim to align well locations and spacing with sedimentary trends, permeability orientations, and structural features, but their performance remains limited under highly complex geological conditions. Optimization theory-based methods (Güyagüler and Horne, 2004; Li and Jafarpour, 2012; Wen et al., 2014) formulate well pattern design as a mathematical optimization problem and employ computational algorithms to determine optimal well layouts. This approach has become a mature and widely adopted technique, particularly when integrated with reservoir simulation and computational intelligence. With the rapid development of intelligent optimization algorithms, especially evolutionary and stochastic methods, well pattern optimization has been extended to the simultaneous optimization of key parameters such as well number, location, trajectory, type, and drilling timing (Zhang et al., 2015). Current research focuses on vertical and horizontal well placement (Alghareeb et al., 2014; Humphries et al., 2014; Malallah et al., 2021), trajectory optimization (Ali et al., 2015; Cao and Sui, 2022; Forouzanfar et al., 2012), optimization of complex well structures including well location, branch number, and branch trajectories (Hamid et al., 2015; Schuetter et al., 2015), as well as well pattern configuration optimization (Khasanov et al., 2014; Nasir et al., 2022; Onwunalu and Durlofsky, 2011; Zhang et al., 2017). However, for large-scale well placement problems, stochastic optimization algorithms typically require hundreds of reservoir simulations (Chen et al., 2017), resulting in extremely high computational burdens and prohibitive time costs (Guo and Reynolds, 2018), which severely restrict their practical application in large-scale oil reservoirs.

Most existing studies remain at the theoretical or small-scale level and lack comprehensive solutions for the multidimensional challenges encountered in large-scale reservoirs. In practical applications, effective reservoir management requires coordinated adjustments beyond infill well location, type, and trajectory, including the shut-in of low-efficiency wells and the conversion between injection and production wells (Li et al., 2024; Lu and Reynolds, 2020). These adjustments significantly increase problem dimensionality, with optimization variables and constraints growing in a superliner manner, and make optimization costly and inefficient (Park et al., 2017). Moreover, in large-scale heterogeneous reservoirs, issues such as local optima traps and low search efficiency further hinder coordinated optimization (Chen et al., 2021). At present, research on well pattern optimization, particularly on coordinated optimization for comprehensive adjustment strategies, remains limited and lacks sufficient engineering adaptability. Therefore, there is a strong need to develop accelerated solution strategies and intelligent optimization algorithms for high-dimensional problems to improve the feasibility and timeliness of well pattern optimization in large-scale reservoirs.

Metaheuristic methods are widely recognized for their efficiency, simplicity, and broad applicability in solving complex engineering optimization problems (Ahmad et al., 2022). Representative algorithms include Genetic Algorithm (GA) (Katoch et al., 2021), Differential Evolution (DE) (Piotrowski et al., 2023), Particle Swarm Optimization (PSO) (Houssein et al., 2021), Ant Colony Optimization (ACO) (Priyadarshi and Kumar, 2025), Whale Optimization Algorithm (WOA) (Nadimi-Shahraki et al., 2023; Yang et al., 2022), Sparrow Search Algorithm (SSA) (Gharehchopogh et al., 2023). Among these, DE is particularly favored for global optimization due to its fast convergence, simple structure, few control parameters, and strong robustness, and has been widely applied to complex practical optimization problems (Ahmad et al., 2022). DE evolves candidate solutions through mutation, crossover, and selection, exhibiting strong global search ability and promising performance in reservoir development optimization (Chen et al., 2021; Zuo and Guo, 2022). However, conventional DE often suffers from premature convergence, stagnation, and degraded efficiency when addressing high-dimensional and strongly nonlinear problems (Ahmad et al., 2022), which limits its effectiveness in large-scale well pattern optimization. To overcome these drawbacks, extensive efforts have been devoted to improving DE through parameter adaptation, enhanced mutation strategies, hybridization, and multi-population coevolution (Ahmad et al., 2022; Deng et al., 2022; Reyes-Davila et al., 2025). As classical variants, SaDE (Qin and Suganthan, 2005) adaptively updates the scaling factor and crossover probability based on historical performance, while jDE (Brest et al., 2006) performs self-adaptation at the individual level. JADE (Zhang and Sanderson, 2009) introduces the “DE/current-to-pbest” mutation strategy and parameter updating based on the means of successful samples, followed by SHADE (Tanabe and Fukunaga, 2013) and L-SHADE (Tanabe and Fukunaga, 2014), which further enhance robustness using historical memory archives. Nevertheless, parameter adaptation alone remains insufficient for highly complex optimization problems. Recently, hybrid DE frameworks that integrate complementary intelligent algorithms have received increasing attention (Deng et al., 2022; Jayabarathi et al., 2007; Mohammadi et al., 2021; Reyes-Davila et al., 2025). Typical examples include CADE (Awad et al., 2017), which couples DE with the Cultural Algorithm; HyGADE (Chaudhary et al., 2019), which embeds DE mutation into GA crossover; AHFADE (Rosić et al., 2021), which introduces DE mutation into different evolutionary stages of the Firefly Algorithm; and ADECS (Wang et al., 2022), which combines DE with the Clonal Selection Algorithm using history-informed mutation and premature convergence detection. Based on the L-SHADE-RSP framework, IDE-EDA (Li et al., 2023) integrates estimation-of-distribution modeling to enhance convergence speed and global optimization. Similarly, nH-WDEOA (Mishra and Pooja Tripathi, 2024) incorporates the backtracking search strategy of WOA into DE to improve constrained optimization performance. These cross-algorithm hybridization strategies represent a major trend in contemporary optimization research.

In this study, a systematic optimization framework is proposed to address the theoretical complexity and practical challenges associated with multi-measure well pattern adjustment in large-scale waterflood reservoirs. The framework integrates various adjustment strategies, including infill well deployment, shut-in of low-efficiency wells, and conversion between injection and production wells, to construct a collaborative optimization model suitable for large-scale reservoir development. To effectively handle the challenges arising from high dimensionality, strong nonlinearity, and discrete constraints, acceleration strategies are investigated from the perspectives of parameter self-adaptation, mutation strategy enhancement, and multi-algorithm hybridization. Based on these developments, an improved differential evolution algorithm with enhanced robustness, fast convergence, and high-precision search capability is established. This study aims to improve the practicality and computational efficiency of well pattern optimization under complex engineering scenarios, thereby providing theoretical support and decision-making tools for large-scale waterflood developments.

2. Method and theory

2.1. Well pattern optimization problem and model construction

In practical reservoir development, adjusting well pattern adjustment typically requires the integrated application of

multiple measures, including well pattern densification, thinning through the shut-in of inefficient wells, and conversion to injection. Therefore, optimization of well pattern adjustments must comprehensively account for the synergistic effects among these measures, so as to formulate an optimal and coordinated adjustment scheme for overall reservoir performance.

The mathematical model for well pattern adjustment optimization consists of an objective function, optimization variables, and associated constraints. With respect to the objective function, existing studies primarily adopt either net present value (Arouri and Sayyafzadeh, 2022; Chen et al., 2018; Mousavi et al., 2024; Semnani et al., 2021; Wang et al., 2016) or cumulative oil production (Ding et al., 2020; Kwon et al., 2021; Raji et al., 2022) as the optimization target. However, under specific offshore engineering constraints, key economic parameters such as the number of allowable infill wells or total drilling budget are often predetermined at the project level. In such situations, the role of optimization shifts from pursuing purely economic optimality to identifying the most technically feasible and operationally robust well configuration within a fixed investment framework. Motivated by this engineering background, a comprehensive objective function is constructed by balancing the maximization of cumulative oil production with requirement that each infill well satisfies a minimum production threshold. By introducing a penalty mechanism, the objective function integrates two coupled goals: maximizing overall cumulative oil production while suppressing the generation of low-efficiency infill wells and mitigating their adverse impact on development performance. Accordingly, the comprehensive objective function is defined as follows:

$$\max f(\mathbf{X}) \tag{1}$$

where:

$$f(\mathbf{X}) = \begin{cases} C_{\mathrm{o,total}}, \text{if } C_{\mathrm{o},n} \geq C_{\mathrm{o,threshold}} \\ C_{\mathrm{o,total}} - f_{\mathrm{penalty}} \sum_{n=1}^{N} \left(C_{\mathrm{o,threshold}} - C_{\mathrm{o},n} \right), \text{if } C_{\mathrm{o},n} < C_{\mathrm{o,threshold}} \end{cases} \tag{2}$$

In the above equation, to ensure that the cumulative oil production of each infill well satisfies the predetermined minimum threshold $C_{\mathrm{o,threshold}}$, a penalty coefficient f_{penalty} is introduced to quantify the impact of insufficient infill well production on the overall optimization performance. This penalty term is constructed upon the cumulative oil production of individual infill wells and their corresponding production threshold, and is defined as follows:

$$f_{\mathrm{penalty}} = \omega \times \frac{C_{\mathrm{o,total}}}{N \times C_{\mathrm{o,threshold}}} \tag{3}$$

where $f(\mathbf{X})$ is the objective function; $C_{\mathrm{o,total}}$ represents the total cumulative oil production of the oilfield, which is a key indicator of the overall development performance and reflects the core value of the objective function, i.e., improving the cumulative oil production of the oilfield through well pattern adjustment and optimization; $C_{\mathrm{o,threshold}}$ is the cumulative oil production threshold for infill wells; $C_{\mathrm{o},n}$ is the cumulative oil production of the n-th infill well; N is the maximum number of infill wells; $\mathbf{X}$ is the vector of optimization variables; f_{penalty} is the penalty coefficient, which reflects the deviation of overall production performance from the prescribed target. For each infill well, if its cumulative oil production falls below the threshold, a penalty term is added. In addition, an amplification factor ω is incorporated to strengthen the constraints on low-efficiency infill wells. By embedding this penalty mechanism into the objective function, the sensitivity of the optimization model to inadequate single-well performance is significantly enhanced, thereby promoting a more balanced development strategy that simultaneously maximizes field production and satisfies minimum performance requirements for individual wells. This formulation enables the optimization framework to better reflect the practical engineering constraints of offshore reservoir development. Moreover, this technically oriented objective function is particularly suitable for scenarios in which high-level economic decisions are predetermined and optimization is focused on engineering feasibility and deployment robustness, demonstrating strong adaptability and practical value in reservoir applications.

In the well pattern optimization problem, wells are categorized into two types according to their roles in the optimization objectives: newly drilled infill wells and existing old wells. This classification reflects the diversity of well pattern adjustments strategies, and underscores the necessity of adopting differentiated optimization measures for each well type. Specifically, the optimization variables and associated constraints differ between these two categories. Therefore, in constructing the well pattern adjustment optimization model, it is essential to define distinct optimization variables and constraints separately for new infill wells and existing old wells.

2.1.1. New infill wells

New infill wells are designed to effectively utilize remaining oil reserves and improve reservoir development performance. The key optimization variables for these wells include their spatial locations and well trajectories, which are characterized by optimizing the coordinates of the heel and toe points. Additionally, the well type of each new well (i.e., injection or production) is treated as an optimization variable. Accordingly, for the i-th new well, the optimization variables are defined as follows:

$$\mathbf{X}_i^{\mathrm{infill}} = \left[c_i, x_i^{\mathrm{heel}}, y_i^{\mathrm{heel}}, z_i^{\mathrm{heel}}, x_i^{\mathrm{toe}}, y_i^{\mathrm{toe}}, z_i^{\mathrm{toe}} \right] \tag{4}$$

where c_i represents the well type of the i-th new infill well, with $c_i = -1$ indicating an injection well, $c_i = 1$ indicating a production well, and $c_i = 0$ meaning that this well will not be drilled, i.e., it will not participate in the infill optimization. $x_i^{\mathrm{heel}}, y_i^{\mathrm{heel}}, z_i^{\mathrm{heel}}$ denote the x, y, and z coordinates of the heel position for the i-th infill well, whereas $x_i^{\mathrm{toe}}, y_i^{\mathrm{toe}}, z_i^{\mathrm{toe}}$ correspond to the coordinates of the toe position. Although the heel and toe are treated as discrete control points in the optimization framework, a post-processing procedure is subsequently performed using standard directional drilling design software to generate continuous, smooth, and drillable trajectories. This ensures that the final well paths strictly honor the optimized target coordinates while fully satisfying mechanical feasibility and operational constraints.

The optimization of new infill wells is subject to several critical constraints to ensure the feasibility and effectiveness of the proposed well locations and trajectories. These constraints include boundary constraints, azimuth angle constraints, and minimum well spacing constraints to prevent inter-well interference. Additionally, special attention is given to the injection-production relationship between new infill wells and existing wells to ensure that the new deployed wells contribute effectively to

reservoir development without negatively impacting the performance of the existing wells.

(1) Maximum infill well constraint

The number of infill wells is typically constrained by the limited number of well slots available on drilling platforms or offshore facilities (Jesmani et al., 2016). Therefore, the total number of deployable infill wells must satisfy the maximum infill well capacity constraint, which can be formulated as:

$$\sum_{i=1}^{N} |c_i| \le N \tag{5}$$

(2) Boundary constraints

The heel and toe coordinates of each infill well must remain within the predefined densification area to ensure that the new wells are deployed within the feasible reservoir region. Mathematically, this boundary constraint is expressed as:

$$\left(x_i^{\text{heel}}, y_i^{\text{heel}}, z_i^{\text{heel}}\right) \in \mathbf{\Omega}_i, \left(x_i^{\text{toe}}, y_i^{\text{toe}}, z_i^{\text{toe}}\right) \in \mathbf{\Omega}_i \tag{6}$$

where $\mathbf{\Omega}_i$ is the densification boundaries of the i-th infill well; $i = 1, 2, \ldots, N$.

(3) Azimuth angle constraint

In offshore drilling operations, precise control of the drilling direction is essential due to the spatial limitations imposed by platform well slots. The azimuth angle, which governs the horizontal direction of the wellbore, must be strictly constrained to ensure precise penetration of the target reservoir while avoiding interference with neighboring wells. Accordingly, the azimuth angle constraint is expressed as follows:

$$\theta_{\min} \le \arctan\left(\frac{y_i^{\text{heel}} - y_i^{\text{toe}}}{x_i^{\text{heel}} - x_i^{\text{toe}}}\right) \le \theta_{\max} \tag{7}$$

where $\theta_{\min}$ and $\theta_{\max}$ define the allowable range for the azimuth angle of the i-th infill well.

(4) Maximum well section length constraint

The wellbore length is a critical parameter in drilling engineering, as it directly impacts both economic efficiency and operational performance. Particularly in the development of long horizontal wells or multilayered reservoirs, excessively long well sections may increase construction difficulty, compromise wellbore stability, and substantially raise drilling costs. Therefore, a rational upper limit on wellbore length is required to improve operational efficiency, and optimize economic benefits:

$$\sqrt{\left(x_i^{\text{heel}} - x_i^{\text{toe}}\right)^2 + \left(y_i^{\text{heel}} - y_i^{\text{toe}}\right)^2 + \left(z_i^{\text{heel}} - z_i^{\text{toe}}\right)^2} \le L_{\max} \tag{8}$$

(5) Minimum well spacing constraints

To avoid inter-well interference and ensure efficient reservoir drainage, a minimum spacing must be maintained between any two wells. This constraint applies not only to the spacing between new infill wells, but also to the spacing between new infill wells and existing wells. The coordinates (x_i, y_i, z_i) and (x_j, y_j, z_j) represent the spatial positions of any points along the trajectories of two wells, rather than only their surface or heel locations. Accordingly, a three-dimensional Euclidean distance constraint is enforced along the entire wellbore paths to guarantee sufficient clearance between vertical, deviated, and horizontal sections:

$$\sqrt{\left(x_i^{\text{infill}} - x_j^{\text{infill}}\right)^2 + \left(y_i^{\text{infill}} - y_j^{\text{infill}}\right)^2 + \left(z_i^{\text{infill}} - z_j^{\text{infill}}\right)^2} \ge d_{\min} \tag{9}$$

$$\sqrt{\left(x_i^{\text{infill}} - x_j^{\text{exist}}\right)^2 + \left(y_i^{\text{infill}} - y_j^{\text{exist}}\right)^2 + \left(z_i^{\text{infill}} - z_j^{\text{exist}}\right)^2} \ge d_{\min} \tag{10}$$

where $d_{\min}$ is the minimum allowable inter-well distance.

This formulation inherently accounts for potential interference between wellbores and effectively avoids collision or adverse pressure interaction among horizontal and deviated sections. During optimization, each well trajectory is discretized into a series of control points, and the spacing constraint is enforced for each corresponding point pair, thereby guaranteeing safe and sufficient inter-well spacing along the entire well path.

2.1.2. Existing old wells

In well pattern optimization, thinning involves shutting in certain existing wells to enhance development efficiency while redistributing injection and production rates to maintain waterflooding stability. This process improves waterflooding performance by increasing the pressure differential between injection and production wells, optimizing their relationship, and expanding sweep efficiency. The optimization task focuses on determining which existing wells should be shut in, subject to a constraint on the maximum number of closed wells. Unlike infill well optimization, this procedure does not alter the spatial locations of existing wells, but adjusts their operating status (on/off) of existing wells according to development requirements and remaining oil distribution, with the objective of improving overall reservoir performance. To formalize this, a decision vector $\mathbf{X}_{\text{o}}^{\text{exist}}$ is defined to indicate whether each existing well is selected for shut-in (well pattern thinning), as follows:

$$\mathbf{X}_{\text{o}}^{\text{exist}} = \left[c_{\text{o}1}^{\text{exist}}, c_{\text{o}2}^{\text{exist}}, \cdots, c_{\text{o}j}^{\text{exist}}, \cdots, c_{\text{o}N}^{\text{exist}}\right] \quad j = 1, 2, \cdots, M \tag{11}$$

where $C_{\text{o}j}^{\text{exist}}$ is a binary variable indicating the shut-in status of the j-th old well, with 1 indicating that the j-th well remains active and 0 indicating that the j-th well is to be shut in (thinned); M is the total number of existing old wells. This approach ensures that the well pattern optimization accounts for the strategic closure of wells to maximize the efficiency of the oilfield's development,

while considering the redistribution of production and injection to improve overall reservoir recovery.

Well type conversion is a key adjustment measure in well pattern optimization, which generally includes two modes: converting production wells to injection wells and vice versa (Sayyafzadeh et al., 2010). The former is a more commonly applied strategy, but the latter has also proven effective in field practices. Well type conversion alters the number and spatial relationship of injection and production wells, thereby modifying the flowline distribution between wells. This adjustment is beneficial for mobilizing previously poorly swept remaining oil. The objective of well type conversion optimization is to determine which wells should be converted, while adhering to a constraint on the maximum allowable number of conversions, so as to improve waterflooding performance.

It should be noted that, in practical optimization, well thinning and well type conversion are mutually exclusive measures. This means that for any given well, these two operations cannot be applied simultaneously. To effectively incorporate this constraint, the decision vector $\mathbf{X}_0^{\text{exist}}$ for existing wells is extended to simultaneously represent both shut-in and well type conversion measures. Specifically, the variable $C_{oj}^{\text{exist}} = -1$ is used to indicate that the j-th existing well undergoes well type conversion. In this manner, $\mathbf{X}_0^{\text{exist}}$ provides a unified representation of both thinning and conversion decisions. By integrating these two aspects into a unified formulation, it significantly reduces the dimensionality of the optimization variables and simplifies the model structure, thereby improving computational efficiency and solution feasibility.

In addition, the constraints of well thinning and well type conversion optimization play a crucial role in ensuring the rationality and feasibility of the optimization strategy.

(1) Mutual exclusivity constraint of adjustment measures

Unlike infill well optimization, the adjustment of existing wells is restricted to three discrete operational states: well type conversion, shut-in (thinning), or no adjustment. Moreover, within the same optimization, thinning and type conversion are mutually exclusive for any individual well. For the j-th existing well, its adjustment variable C_{oj}^{exist} can only take three values: $C_{oj}^{\text{exist}} = -1$ indicates well type conversion, $C_{oj}^{\text{exist}} = 0$ represents shut-in, and $C_{oj}^{\text{exist}} = 1$ denotes that the well retains its original status (no adjustment):

$$c_{oj}^{\text{exist}} \in \{-1, 0, 1\} \tag{12}$$

This constraint strictly limits the adjustment states of each existing well to one of three admissible options—well type conversion, shut-in (thinning), or no adjustment—thereby excluding infeasible intermediate states and ensuring the logical consistency of the optimization model. By enforcing this rule, the optimization process avoids conflicting or impractical measures, ensuring that the resulting adjustment strategy is both effective and implementable in reservoir management scenarios.

(2) Maximum well type conversion constraint

This constraint limits the total number of wells that can be converted, ensuring that the number of well type conversion does not exceed the maximum allowable threshold. Mathematically, this constraint can be expressed as follows:

$$\sum_{j=1}^{M} \left\lfloor \frac{1 - c_{oj}^{\text{exist}}}{2} \right\rfloor \le s_2 \tag{13}$$

where $\lfloor x \rfloor$ represents the floor function, defined as the largest integer less than or equal to x; s_2 is the predetermined maximum number of wells allowed for type conversions. By incorporating the floor function, the term $\left\lfloor \frac{1-C_{oj}^{\text{exist}}}{2} \right\rfloor$ is introduced to mathematically represent the status of the j-th existing well in the optimization model. This term helps distinguish between the different measures applied to the well (i.e., well type conversion, thinning, or no adjustment) in a computationally efficient way. When the well undergoes well type conversion ($C_{oj}^{\text{exist}} - 1$), the term $\left\lfloor \frac{1-C_{oj}^{\text{exist}}}{2} \right\rfloor$ evaluates to 1. This is because:

$$\left\lfloor \frac{1 - c_{oj}^{\text{exist}}}{2} \right\rfloor = \left\lfloor \frac{1 - (-1)}{2} \right\rfloor = \left\lfloor \frac{2}{2} \right\rfloor = 1$$

For other measures (no adjustment or thinning), the result is 0. Specifically.

- When the well retains its original status (no adjustment, $C_{oj}^{\text{exist}} = 1$):

$$\left\lfloor \frac{1 - c_{oj}^{\text{exist}}}{2} \right\rfloor = \left\lfloor \frac{1 - 1}{2} \right\rfloor = \left\lfloor \frac{0}{2} \right\rfloor = 0$$

- When the well is shut in (thinning, $C_{oj}^{\text{exist}} = 0$):

$$\left\lfloor \frac{1 - c_{oj}^{\text{exist}}}{2} \right\rfloor = \left\lfloor \frac{1 - 0}{2} \right\rfloor = \left\lfloor \frac{1}{2} \right\rfloor = 0$$

By incorporating this term, the model can effectively manage constraints related to well type conversion, ensuring that the optimization scheme remains both mathematically rigorous and practically implementable. This formulation simplifies the mathematical representation of the optimization problem and enables a clear distinction among different adjustment measures applied to the wells.

(3) Maximum well thinning constraint

The total number of wells to be thinned (i.e., shut in) must be limited to ensure that the post-thinning well density remains within a reasonable range. This constraint prevents excessive thinning, which could otherwise lead to insufficient well density and degraded development efficiency. It can be represented by summing the shut-in status variables of all wells, as follows:

$$\sum_{j=1}^{M} \left(1 - \left| c_{oj}^{\text{exist}} \right|\right) \le s_1 \tag{14}$$

where M is the total number of wells; s_1 is the predetermined maximum number of wells allowed to be shut in. The term $1-|C_{oj}^{\text{exist}}|$ serves as an indicator function that specifically identifies wells undergoing well thinning (where the result is 1). For other measures, such as no adjustment or well type conversion, the

corresponding value is 0. This term can be integrated into the constraint formulation to limit the number of wells undergoing thinning. This constraint prevents excessive thinning, which could lead to insufficient reservoir recovery, while also mitigating the negative impact of over-thinning on development performance. By controlling the number of shut-in wells, different thinning strategies can be effectively compared, ensuring that the optimized well pattern aligns with development objectives.

(4) Development stability constraint

This study focuses on optimizing well pattern configuration and well type conversion, without involving injection–production parameter optimization. To ensure field feasibility of the optimization results, a development stability constraint is incorporated into the model. This constraint requires that existing wells without conversion or shut-in operations, as well as newly infill wells, satisfy the predefined production and injection allocation targets, thus avoiding adverse impacts on overall field performance. Under this constraint, the total production and injection rates after adjustment are expressed as follows:

$$\sum_{i=1}^{N}\frac{1+c_i}{2}q_{\mathrm{pro},i}+\sum_{j=1}^{M_{\mathrm{pro}}}\left\lfloor\frac{1+c_{oj}^{\mathrm{exist}}}{2}\right\rfloor q_{\mathrm{pro},j}=q_{\mathrm{pro},t} \tag{15}$$

$$\sum_{i=1}^{N}\frac{1-c_i}{2}q_{\mathrm{inj},i}+\sum_{j=1}^{M_{\mathrm{inj}}}\left\lfloor\frac{1-c_{oj}^{\mathrm{exist}}}{2}\right\rfloor q_{\mathrm{inj},j}=q_{\mathrm{inj},t} \tag{16}$$

where $q_{\mathrm{pro},i}$ and $q_{\mathrm{inj},i}$ denote the liquid production and water injection rates of infill wells, m^3/d; $q_{\mathrm{pro},j}$ and $q_{\mathrm{inj},j}$ represent the liquid production and water injection rates of existing wells, m^3/d; $q_{\mathrm{pro},t}$ and $q_{\mathrm{inj},t}$ are the total target production and injection rates after adjustment, m^3/d.

During the optimization, adaptive scaling of injection–production rates is applied to existing wells that are neither shut in nor converted, subject to their capacity constraints:

$$q_{\mathrm{pro},j}=\left\lfloor\frac{c_{oj}^{\mathrm{exist}}+1}{2}\right\rfloor\frac{q_{\mathrm{pro},j}}{\max\left(q_{\mathrm{pro},t},q_{\mathrm{inj},t}\right)} \tag{17}$$

$$q_{\mathrm{inj},t}=\left\lfloor\frac{1-c_{oj}^{\mathrm{exist}}}{2}\right\rfloor\frac{q_{\mathrm{inj},t}}{\max\left(q_{\mathrm{pro},t},q_{\mathrm{inj},t}\right)} \tag{18}$$

This treatment prevents imbalance or fluctuations in the injection–production system after structural adjustments, ensuring the engineering feasibility and operational continuity of the optimized well-pattern configuration.

In practical well-pattern optimization, the most computationally intensive component is the repeated execution of the reservoir simulator, which dominates the overall computational cost. Therefore, minimizing unnecessary simulation runs is essential for improving optimization efficiency. To achieve this, a rejection strategy is applied during the optimization process. Specifically, any candidate solution that violates predefined operational or engineering constraints is directly discarded without invoking the reservoir simulation. Only feasible solutions are retained for simulation-based evaluation, while infeasible candidate solutions do not participate in the evolutionary search. This strategy effectively prevents computational resources from being wasted on invalid simulations.

2.2. Multi-strategy enhanced adaptive differential evolution algorithm

The optimization problem integrates multiple adjustment measures for both new and existing wells, forming a highly combinatorial and complex decision space. In large-scale oilfields, the number of decision variables grows exponentially, and the coexistence of discrete variables and diverse engineering constraints further complicates the problem. Motivated by the follower position update strategy of SSA (Gharehchopogh et al., 2023) and the logarithmic spiral search mechanism of WOA (Yang et al., 2022), a multi-strategy Enhanced Adaptive Differential Evolution (E-ADE) algorithm is proposed in this study. The proposed E-ADE framework is established on adaptive regulation of key parameters such as the scaling factor and crossover probability. By considering the relationship between each population individual and the global best solution, a variable spiral search-driven adaptive elite differential mutation strategy is introduced, alongside an adaptive population size adjustment mechanism. These strategies allow for flexible adjustments of differential vectors and population sizes at different evolutionary stages, thereby ensuring strong global exploration capability. Meanwhile, E-ADE realizes an adaptive transition between global exploration and local exploitation, significantly enhancing local search ability and accelerating convergence. As a result, the proposed E-ADE algorithm provides a more robust and efficient solution for complex engineering problems, and effectively overcome the limitations of conventional DE in handling large-scale, high-dimensional, and strongly constrained optimization tasks.

2.2.1. Chaotic initialization strategy

In the standard DE algorithm, population initialization is typically done by pseudo-random sampling. For high-dimensional problems, this approach often leads to locally clustered individuals and reduced population diversity, which in turn weakens global search capability and slows convergence. To address this issue, chaotic mappings have been widely used to improve optimization performance. Among the available methods such as Logistic mapping, Tent mapping, and Circle mapping (Demir et al., 2020; Yang et al., 2014), the SPM mapping (Ban et al., 2020), which combines Sine and PWLCM mapping, provides enhanced chaotic behavior and superior ergodicity. In this study, SPM mapping is adopted to initialize the population, enabling a more uniform distribution of individuals across the search space and improving initial population diversity. This enhancement strengthens global exploration and accelerates convergence. The SPM mapping is formulated as:

$$z_{i+1}=\begin{cases}\mathrm{mod}\left(\frac{z_i}{\eta}+\mu\sin(\pi z_i)+r,1\right),0\le z_i<\eta\\ \mathrm{mod}\left(\frac{z_i/\eta}{0.5-\eta}+\mu\sin(\pi z_i)+r,1\right),\eta\le z_i<0.5\\ \mathrm{mod}\left(\frac{(1-z_i)/\eta}{0.5-\eta}+\mu\sin(\pi(1-z_i))+r,1\right),0.5\le z_i<1-\eta\\ \mathrm{mod}\left(\frac{1-z_i}{\eta}+\mu\sin(\pi(1-z_i))+r,1\right),1-\eta\le z_i<1\end{cases} \tag{19}$$

where $\eta\in(0,1)$; $\mu\in(0,1)$ indicates that the system is in a chaotic state; r is the perturbation parameter of the chaotic system, a random number between 0 and 1.

The control parameters, including the population size (N_{pop}), maximum number of iterations (T), initial scaling factor (F), and

Table 1
Seven common mutation strategies.

Mutation strategy	Differential expression	Description
DE/rand/1	$\mathbf{v}_i^{(t)} = \mathbf{x}_{r1}^{(t)} + F\cdot\left(\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r3}^{(t)}\right)$	The basic mutation strategy, introducing randomness to enhance exploration ability.
DE/best/1	$\mathbf{v}_i^{(t)} = \mathbf{x}_{\text{best}}^{(t)} + F\cdot\left(\mathbf{x}_{r1}^{(t)} - \mathbf{x}_{r2}^{(t)}\right)$	Uses the current global best solution as the base vector, promoting fast convergence but with a risk of premature convergence.
DE/rand/2	$\mathbf{v}_i^{(t)} = \mathbf{x}_{r1}^{(t)} + F\cdot\left(\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r3}^{(t)}\right) + F\cdot\left(\mathbf{x}_{r4}^{(t)} - \mathbf{x}_{r5}^{(t)}\right)$	Introduces two differential vectors to enhance diversity, improving exploration in complex landscapes.
DE/best/2	$\mathbf{v}_i^{(t)} = \mathbf{x}_{\text{best}}^{(t)} + F\cdot\left(\mathbf{x}_{r1}^{(t)} - \mathbf{x}_{r2}^{(t)}\right) + F\cdot\left(\mathbf{x}_{r3}^{(t)} - \mathbf{x}_{r4}^{(t)}\right)$	Uses two differential vectors while referencing the best solution, improving convergence speed but increasing the risk of stagnation.
DE/rand-to-best/bin	$\mathbf{v}_i^{(t)} = \mathbf{x}_{r1}^{(t)} + F\cdot\left(\mathbf{x}_{\text{best}}^{(t)} - \mathbf{x}_{r1}^{(t)}\right) + F\cdot\left(\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r3}^{(t)}\right)$	A hybrid approach that leverages both the global best and multiple difference vectors to improve convergence stability.
DE/current-to-rand/bin	$\mathbf{v}_i^{(t)} = \mathbf{x}_i^{(t)} + F\cdot\left(\mathbf{x}_{r1}^{(t)} - \mathbf{x}_i^{(t)}\right) + F\cdot\left(\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r3}^{(t)}\right)$	Introduces stochasticity into the mutation process, helping maintain diversity and avoiding local optima.
DE/current-to-best/bin	$\mathbf{v}_i^{(t)} = \mathbf{x}_i^{(t)} + F\cdot\left(\mathbf{x}_{\text{best}}^{(t)} - \mathbf{x}_i^{(t)}\right) + F\cdot\left(\mathbf{x}_{r1}^{(t)} - \mathbf{x}_{r2}^{(t)}\right)$	Balances exploration and exploitation by guiding individuals toward the best solution while maintaining population diversity.

initial crossover probability (CR) are first specified. Subsequently, the SPM chaotic mapping is applied to generate the initial population, as follows:

$$\mathbf{x}_i^{(0)} = \mathbf{x}_{\text{lb}} + z_i\cdot(\mathbf{x}_{\text{ub}} - \mathbf{x}_{\text{lb}}) \tag{20}$$

where $\boldsymbol{i}$ is the initial population individual; $i = 1, 2, \ldots, N_{\text{pop}}$; $\mathbf{x}_{\text{lb}}$ and $\mathbf{x}_{\text{ub}}$ are the lower and upper bounds for each individual.

2.2.2. Adaptive enhanced mutation strategy

The mutation strategy plays a pivotal role in DE algorithm. Table 1 summarizes seven commonly used differential mutation strategies, together with their corresponding formulations and characteristics (Deng et al., 2022; Reyes-Davila et al., 2025). Although these strategies differ in their exploration behaviors, they share a common principle: the search direction is determined by a base vector guided by differential vectors, while the step size is controlled by a scaling factor. However, the dependence on specific base vector selection and individual differences may lead to an imbalance between global exploration and local exploitation. For example, mutation strategies favoring global exploration, such as DE/rand/1, often suffer from reduced local search accuracy, whereas strategies emphasizing local exploitation, such as DE/best/1, may easily lose global search capability. In high-dimensional optimization problems, traditional differential vector-based mutation strategies tend to focus on a limited dimension, which degrades search efficiency. Therefore, adaptive mechanisms, hybrid mutation strategies, and dynamic scaling schemes are required to improve the algorithm's convergence and stability.

To overcome the limitations associated with a fixed scaling factor F, an adaptive updating mechanism is introduced to dynamically adjust F according to the current iteration stage. This strategy enables the algorithm to satisfy the varying search requirements at different evolutionary stages. The nonlinear dynamic update formula for F is given by:

$$F(t) = F_{\max} - \frac{1}{1 + \mathrm{e}^{-\alpha(t/T-0.5)}}\cdot(F_{\max} - F_{\min}) \tag{21}$$

where $F_{\max}$ and $F_{\min}$ represent the initial and final values of F, respectively; t denotes the current iteration count, and T is the maximum number of iterations. The term $\frac{1}{1+\mathrm{e}^{-\alpha(t/T-0.5)}}$ represents a shifted and scaled Sigmoid function, which is widely used for parameter control due to its characteristic S-shaped profile (Yussif et al., 2024). This curve exhibits a smooth start, rapid transition, and stable convergence, making it particularly suited for balancing global exploration and local exploitation in optimization algorithms. The parameter α controls the steepness of the Sigmoid function, with positive values indicating increased steepness. In this work, α is set to 10. By regulating F through the shifted and scaled Sigmoid function, its variation becomes smooth and continuous, effectively avoiding abrupt transitions that could negatively impact convergence. As illustrated in Fig. 1, the curve exhibits a typical S-shaped trend, where the value of F varies slowly in the early stage, undergoes a smooth transition in the middle phase, and gradually stabilizes in the later stage. This adaptive adjustment mechanism is particularly suitable for the dynamic regulation of F in algorithm, as it promotes stronger

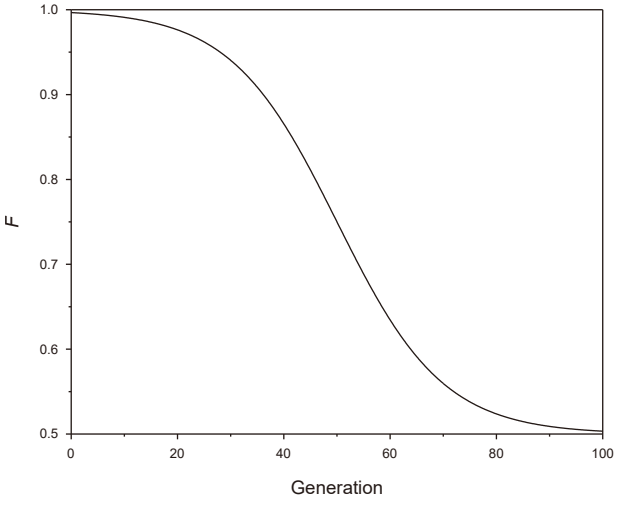

Fig. 1. Variation of the scaling factor over generations.

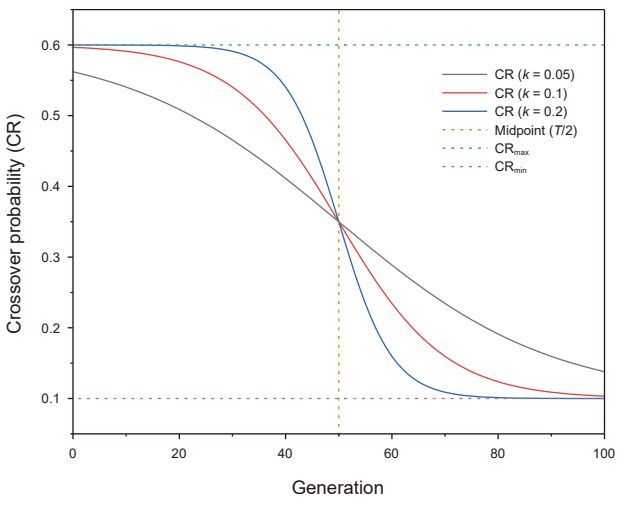

Fig. 2. Sigmoid-based adaptive crossover probability decay.

exploration in the early stage while prioritizing refined exploitation in the later stages.

Inspired by the follower position update strategy in SSA, a new basic differential vector update strategy is proposed by considering the positional relationship between current population individuals and global best solution. In this strategy, the global best individual serves as the reference vector, and a logarithmic spiral search mechanism is introduced to guide individuals to update along a spiral trajectory, thereby generating diverse search paths rather than being confined to traditional linear differential mutations. The inherent characteristics of the spiral curve enable extensive global exploration in the early stage and a gradual transition toward local regions for refined exploitation in the later stage. Additionally, a random perturbation matrix $\boldsymbol{A}$ with random values of ± 1, is introduced to disturb the mutation direction, guiding individuals to oscillate around the global best individual and preventing directional bias. Furthermore, local variance is used to adaptively control the mutation magnitude in each dimension, avoiding excessive oscillations while maintaining search stability and directionality. Based on these designs, the new basic differential vector is formulated as follows:

$$\mathbf{v}_{i\text{base}}^{(t)} = \mathbf{x}_{\text{best}}^{(t)} + \left| \mathbf{x}_i^{(t)} - \mathrm{e}^{bl} \cdot \cos(2\pi l) \cdot \mathbf{x}_{\text{best}}^{(t)} \right| \cdot \mathbf{A}^{+} \cdot \mathbf{L} \tag{22}$$

where $\mathbf{v}_{i\text{base}}^{(t)}$ denotes the basic mutation vector of the i-th population individual at the t-th generation, and t represents the current iteration number. The parameter b is the spiral shape factor that controls the shape of the spiral trajectory, while l is a random number, typically drawn from a uniform distribution, which modulates the angular variation of the spiral search. The vectors $\mathbf{x}_{\text{best}}^{(t)}$ and $\mathbf{x}_i^{(t)}$ denote the position vectors of the global best individual and the i-th individual at the t-th generation, respectively. The matrix $\mathbf{A}$ is a $1 \times d$ random matrix with elements randomly assigned ± 1, and it is primarily used to perturb the mutation direction, enabling the search to oscillate around $\mathbf{x}_{\text{best}}^{(t)}$. Additionally, $\mathbf{A}^{+} = \mathbf{A}^{\mathrm{T}} (\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}$ is employed to regulate the self-adaptability of the mutation step size, ensuring that the effective mutation amplitude decreases automatically with increasing dimensionality. This mechanism prevents excessive mutation steps in high-dimensional spaces, thereby avoiding large deviations from the global optimum and maintaining search stability. Moreover, $\mathbf{L}$ is a $1 \times d$ dimensional matrix whose elements are all equal to 1.

To improve the dynamic adaptability of the spiral search, a linear diversity feedback strategy is introduced to regulate the spiral coefficient b according to population diversity. Population diversity is quantified using Shannon entropy (Lin, 2002; Noorizadeh and Shakerzadeh, 2010), which characterizes the distribution uncertainty across dimensions. High entropy indicates diverse solutions, promoting exploration, while low entropy reflects convergence, favoring exploitation. Specifically, when individuals exhibit similar values across dimensions, the entropy approaches zero, whereas a more uniform distribution yields higher entropy value. Based on this diversity information, CR is adjusted accordingly to balance global and local search adaptively throughout the optimization process. For a population $\mathbf{p}$ in a D-dimensional search space, the Shannon entropy is calculated as follows:

$$h(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{23}$$

Here, p_i represents the probability of individuals in the population falling into the i-th discretized interval for each dimension, and n denotes the number of discretized intervals in each dimension of the population, which directly affects the accuracy of Shannon Entropy calculation. If n is too small, important distribution information loss may occur; if n is too large, data sparsity may lead to unstable entropy values. Generally, the number of intervals can be adjusted based on N_{pop} and D. In this study, the Freedman-Diaconis rule (Contreras-Reyes and Brito, 2022) is adopted to determine a reasonable number of discretized intervals. This rule is well suited for large-scale populations, as it effectively balances distribution resolution and data sparsity. The number of discrete intervals is computed as:

$$n = \left\lceil \frac{\max(\mathbf{p}) - \min(\mathbf{p})}{2 \times \mathrm{IQR} \times N_{\text{pop}}^{-1/3}} \right\rceil \tag{24}$$

where IQR is the interquartile range of the data distribution, and N_{pop} is the population size.

The coefficient b is linearly scaled between $b_{\min}$ and $b_{\max}$ based on the ratio of current population diversity to its theoretical maximum value, as expressed in Eq. (25):

$$b = b_{\min} + (b_{\max} - b_{\min}) \times \frac{h}{H} \tag{25}$$

where $H = \log_2(n)$ is the theoretical maximum Shannon entropy, which corresponds to the scenario where population individuals are uniformly distributed across the search space. Specifically, b is dynamically regulated by the population diversity.

Based on the constructed basic differential vector, a fitness-driven adaptive enhanced differential mutation strategy is further proposed. The improvement in the global best fitness between two consecutive generations is quantified by the following equation:

$$\Delta f = f_{\text{prev}} - f_{\text{best}} \tag{26}$$

where f_{prev} represents the global best fitness in the previous generation, and f_{best} denotes the global best fitness in the current generation. This fitness-driven adaptive mechanism dynamically regulates the mutation intensity based on the improvement of fitness, allowing the algorithm to enhance global exploration when evolutionary progress stagnates and to shift toward exploitation when convergence is evident.

When $\Delta f < 0$, the basic differential evolution mutation strategy is activated to introduce significant perturbations to enhance population diversity and facilitate a broad search, thereby avoiding entrapment in local optima. The mutation update formula is given by:

$$\mathbf{v}_i^{(t)} = \mathbf{v}_{i\text{base}}^{(t)} + a \cdot F(t) \cdot \left[\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r1}^{(t)} \right] \tag{27}$$

where $\mathbf{v}_i^{(t)}$ is the mutated vector of the i-th individual at the t-th generation; $a = \alpha(2r_1 - 1)$; $\alpha = 1 - t/T$. The parameter a varies within the range $[-1,1]$, controlling the perturbation strength; α is a control parameter that linearly decreases from 1 to 0 as the iteration number t increases. r_1 is a random number within the interval $[0,1]$; $F(t)$ is an adaptive scaling factor that varies over time, controlling the mutation step size; and $\mathbf{x}_{r2}^{(t)}$ and $\mathbf{x}_{r1}^{(t)}$ denote the positions of two distinct randomly selected population individuals at the t-th generation.

When $\Delta f \geq 0$, indicating an improvement in global best fitness, an elite differential mutation strategy guided by a logarithmic spiral search is employed. In this strategy, individuals perform refined local searches around the current best solution along a spiral trajectory with a controlled search radius. The mutation update for each individual is formulated as follows:

$$\mathbf{v}_i^{(t)} = \mathbf{v}_{i\text{base}}^{(t)} + a \cdot F(t) \cdot \left[\mathrm{e}^{bl} \cdot \cos(2\pi l) \cdot \mathbf{x}_{\text{best}}^{(t)} - \mathbf{x}_{r1}^{(t)} \right] \tag{28}$$

The proposed method integrates multiple complementary mechanisms, including global best guidance, logarithmic spiral

trajectory search, random perturbation matrices, local variance control, and adaptive scaling factors. By dynamically evaluating the improvement of the current optimal solution, the algorithm adaptively switches between an elite differential mutation guided by spiral search and a basic differential evolution mutation. This adaptive switching mechanism promotes population diversity in the early stage to prevent premature convergence, while enhancing convergence rate and optimization accuracy in the later stage.

2.2.3. Adaptive enhanced crossover strategy

In the DE algorithm, crossover is a critical step performed after mutation to exchange information between mutant vectors and parent individuals, thereby generating new offspring, enhancing population diversity. This can be expressed as (Deng et al., 2022):

$$\mathbf{u}_{i,j}^{(t)} = \begin{cases} \mathbf{v}_{i,j}^{(t)}, \text{ if } \text{rand}_j \le \text{CR} \text{ or } j = j_{\text{rand}} \\ \mathbf{x}_{i,j,}^{(t)} \text{otherwise} \end{cases} \tag{29}$$

Here, $\mathbf{u}_{i,j}^{(t)}$ represents the j-th dimension of the offspring individual generated for the i-th population individual at the t-th generation; $\mathbf{v}_{i,j}^{(t)}$ is the j-th dimension of the mutant vector; $\mathbf{x}_{i,j,}^{(t)}$ is the j-th dimension of the parent individual. The variable rand_j is a uniformly distributed random number in the interval $[0, 1]$, which determines whether crossover is performed on the j-th dimension. The parameter CR is the crossover probability, controlling the proportion of information transferred between the parent and mutant individuals. The index j_{rand} is a randomly selected dimension that guarantees at least one component of the offspring is inherited from the mutant vector, thereby avoiding complete replication of the parent individual.

Dynamic adjustment of the crossover probability CR is a crucial improvement strategy for DE, particularly in high-dimensional and multimodal optimization problems. Conventional iteration-based CR scheduling fail to capture the evolving population state and may result in premature convergence or insufficient exploration. To overcome this limitation, a population-diversity-driven adaptive CR strategy is proposed, in which the normalized Shannon entropy is mapped through a shifted sigmoid function to dynamically regulate CR according to both iteration progress and diversity information:

$$\text{CR}(t) = \text{CR}_{\max} - \frac{1}{1 + \text{e}^{-\alpha(t/T - h/H)}} \cdot (\text{CR}_{\max} - \text{CR}_{\min}) \tag{30}$$

where, $\text{CR}_{\max}$ and $\text{CR}_{\min}$ represent the maximum and minimum crossover probabilities, respectively. By normalizing the entropy using H, consistent CR regulation across different optimization problems is ensured, enhancing its generalization capability. In the early evolutionary stage, when population diversity is high, the Sigmoid curve decays slowly and maintains a relatively large CR, promoting extensive crossover and strong global exploration. As iterations proceed, population diversity gradually decreases, leading to a reduced CR, and the search process naturally shifts toward refined local exploitation (see Fig. 2). This adaptive mechanism helps avoid unnecessary exploration while improving convergence speed and solution accuracy.

2.2.4. Adaptive population size adjustment strategy

An adaptive population size adjustment strategy based on fitness variation is proposed in this study. During the optimization process, the population size is dynamically regulated in response to changes in global best fitness. A larger population is maintained in the early stage to enhance global exploration, while in the later stage, the population size is gradually reduced to accelerate convergence. At each generation, the global best fitness is updated, and the convergence status is evaluated using the fitness variation Δf between two consecutive generations. When $\Delta f > \varepsilon$ (a predefined threshold), indicating a significant improvement, the population size is reduced to lower computational cost and enhance local exploitation. The detailed adjustment procedure is as follows.

(1) Determine the population size reduction

The population size is dynamically adjusted based on a scaling factor β, by which the number of individuals in the population is reduced as follows:

$$N_{\text{dec}} = \left\lfloor \beta \times N_{\text{pop}}^{(t)} \right\rfloor \tag{31}$$

where β is the scaling factor, typically selected between 0.1 and 0.3. $N_{\text{pop}}^{(t)}$ is the population size at the t-th generation, and N_{dec} is the number of individuals to be removed, which is determined according to the current population size and optimization requirements. Over-compressing the population may significantly restrict search diversity, and negatively affect the optimization performance.

(2) Deletion of poor-fitness individuals

To preserve solution quality and guide convergence, the global best individual is first identified and recorded. The mean fitness of the current population is then calculated, and N_{dec} individuals with fitness values below the mean are randomly removed. The remaining individuals are sorted by fitness, and the presence of the global best individual is verified. If absent, it is reinserted by randomly replacing one existing individual. This mechanism prevents the loss of the best solution, particularly when the algorithm stagnates near local optima. By eliminating low-quality individuals, the overall population fitness is improved, convergence is accelerated, and a reasonable level of diversity is maintained, thereby improving both exploration and exploitation capabilities of the algorithm.

(3) Updating and reconstructing the population

Based on the number of removed individuals and the remaining population, the population size is updated as $N = N - N_{\text{dec}}$, and the overall structure of the population subsequently reconstructed.

2.2.5. The E-ADE algorithm flowchart

The overall flowchart of the E-ADE algorithm is shown in Fig. 3. The iterative process continues until either convergence to the global optimum is achieved or the maximum number of iterations is reached. The proposed adaptive mechanisms substantially enhance the efficiency and robustness of the algorithm in solving complex optimization problems.

The pseudocode of the E-ADE algorithm is given in Table 2.

2.2.6. Benchmark test

To comprehensively evaluate the optimization performance of the E-ADE algorithm, nine classical benchmark functions are selected to examine its convergence accuracy, convergence speed, and global search capability. These benchmark functions are widely used in optimization research and are all multimodal, making them particularly suitable for evaluating global search

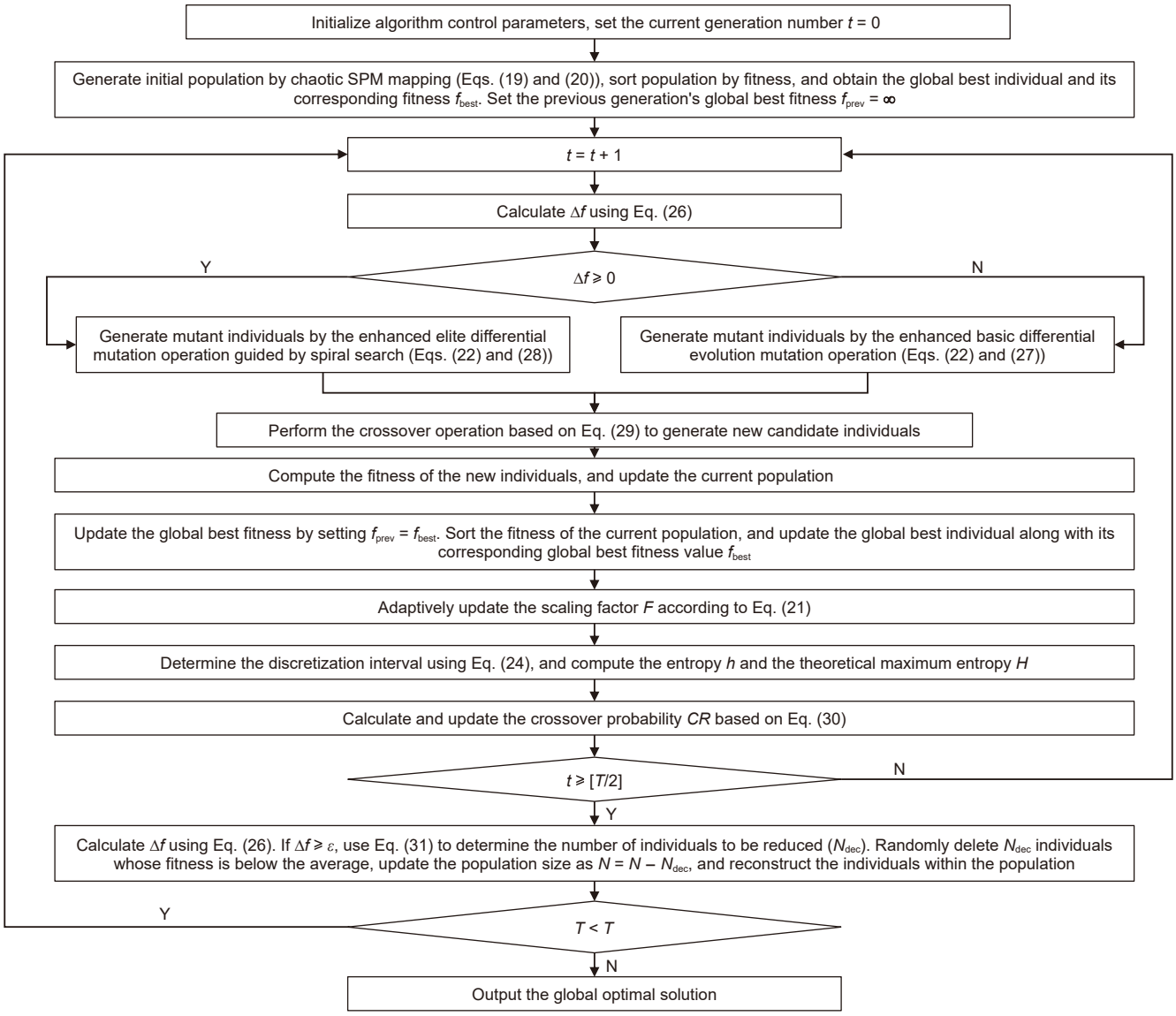

Fig. 3. The flowchart of the E-ADE algorithm.

efficiency. Their mathematical expressions, search domains, optimal values, and related attributes are summarized in Table 3.

For comparative evaluation, the proposed E-ADE algorithm was systematically benchmarked against several mainstream optimization methods, including DE, GA, SSA, WOA and L-SHADE. As one of the champion algorithms in the CEC2014 competition, L-SHADE incorporates a linear population size reduction mechanism, exhibiting excellent convergence performance and optimization robustness (Tanabe and Fukunaga, 2014). To ensure the scientific validity and fairness of the comparisons, the key parameters of all algorithms were set according to their standard recommended configurations as listed in Tables 4 and 5. Each algorithm was independently run 10 times on each benchmark function under the identical parameter settings to guarantee the reliability and statistical significance of the experimental results. Through this standardized testing framework, the performance of E-ADE in different scenarios is systematically evaluated and compared with that of the classical algorithms.

The experimental results are shown in Fig. 4, where the x-axis represents the iteration number and the y-axis represents the fitness value. Compared with other algorithms (DE, GA, SSA, WOA and L-SHADE), E-ADE consistently converges rapidly to or near the global optimal solution across all nine benchmark functions, demonstrating its superior global search capability and fast convergence performance.

The optimization performance of the different algorithms is evaluated using the best value (*Best*), mean value (*Mean*), and standard deviation (*Std*) of the obtained global optimal values. *Best* reflects the best solution achieved over multiple independent runs and serves as a direct indicator of the algorithm's global search capability, with smaller value indicating closer proximity to the theoretical optimum. *Mean* represents the overall performance across repeated runs and reflects the convergence accuracy and reliability of the algorithm; a lower *Mean* value indicates more consistent high-quality solutions. *Std* is used to measure the stability of the algorithm by quantifying the fluctuation of results among different runs, where a smaller *Std* implies stronger robustness against random factors and initial conditions. Based on these standardized evaluation criteria, the performance of different algorithms on nine benchmark functions is

Table 2
The pseudocode of the E-ADE algorithm.

Pseudo-code of enhanced adaptive differential evolution algorithm (E-ADE):

Input: N_{pop}, T, F_{max}, F_{min}, CR_{max}, CR_{min}, β, ε
Output: Global best solution $\mathbf{x}_{best}$ with fitness f_{best}
Initialize population size $N = N_{pop}$, generation counter $t = 0$
Initialize control parameters $F \in [F_{min}, F_{max}]$, $CR \in [CR_{min}, CR_{max}]$
Generate initial population $\mathbf{P}(0)$ by chaotic SPM mapping (Eqs. (19) and (20))
Evaluate fitness of individuals in $\mathbf{P}(0)$
Sort population by fitness, obtain $\mathbf{x}_{best}$ and f_{best}
Set $f_{prev} = +\infty$//previous generation best fitness
while $t < T$ do
  $t = t + 1$
  //Mutation strategy selection
    Compute $\Delta f = f_{prev} - f_{best}$ (Eq. (23))
  if $\Delta f \geq 0$ then
    Generate mutant individuals via enhanced elite DE guided by spiral search (Eqs. (22) and (25))
  else
    Generate mutant individuals via enhanced basic DE strategy (Eqs. (22) and (24))
  end if
  //Crossover
  Perform crossover to produce trial vectors (Eq. (26))
  //Selection
  Evaluate fitness of trial individuals
  Update population using greedy selection and simulated annealing replacement rule
  //Global best update
  $f_{prev} = f_{best}$
  Sort population fitness and update $\mathbf{x}_{best}$, f_{best}
  //Control parameter update
  Update scaling factor F (Eq. (21))
  Determine discretization interval (Eq. (28))
  Compute entropy h and theoretical max entropy H
  Update CR via Eq. (29)
  //Dynamic population adjustment
  if $t \geq \lfloor T/2 \rfloor$ then
    Compute Δf again
    if $\Delta f \geq \varepsilon$ then
      Determine reduction size N_{dec} (Eq. (30))
      Randomly remove N_{dec} individuals with fitness below the population mean
    $N = N - N_{dec}$
    Reconstruct and update population
    end if
  end if
end while
return $\mathbf{x}_{best}$, f_{best}

Table 3
Benchmark test functions (Brest et al., 2006; Xue and Shen, 2020; Zuo and Guo, 2022).

Test function	Function expression	Search domain	Optimal value
F1	$f(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	$\lvert x_i\rvert \leq 100$	0
F2	$f(x) = \sum_{i=1}^{n} i \cdot x_i^4 + \text{random}[0.1)$	$\lvert x_i\rvert \leq 1.28$	0
F3	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$\lvert x_i\rvert \leq 5.12$	0
F4	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{1}{\sqrt{i}}\right) + 1$	$\lvert x_i\rvert \leq 600$	0
F5	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$\lvert x_i\rvert \leq 32$ $\lvert x_i\rvert \leq 50$	0
F6	$f_5(x) = \frac{\pi}{n}\left\{\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 \cdot \left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$\lvert x_i\rvert \leq 50$	0
F7	$f(x) = 0.1\left[\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2 \cdot \left(1 + \sin^2(3\pi x_{i+1})\right) + (x_n - 1)^2 \cdot \left(1 + \sin^2(2\pi x_n)\right)\right] + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	$\lvert x_i\rvert \leq 500$	0
F8	$f(x) = 418.9829n - \sum_{i=1}^{n} x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	$\lvert x_i\rvert \leq 10$	0
F9	$f(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i\rvert$		

Table 4
Key parameters for algorithms.

Algorithm	Key parameters
E-ADE, DE, L-SHADE	$F = 0.85$, CR= 0.95
GA	$F = 0.1$, CR = 0.9
SSA	PD = 0.2, ST = 0.8
WOA	$B = 1$, $w = 0.9$–0.4

Table 5
Experimental parameter settings.

Symbol	Meaning	Value
N_{pop}	Population size	50
T	Maximum iterations	200
dim	Dimension	30
n	Number of runs	10

comprehensively compared. The statistical results are summarized in Table 6, where the best values for each metric are highlighted in bold to clearly illustrate the performance differences among the algorithms across different benchmark functions.

In terms of *Best* metric, E-ADE achieved or approached the theoretical optimum on all nine benchmark functions, demonstrating a strong global search capability. With respect to the *Mean* metric, E-ADE outperformed the other algorithms on eight test functions (F1, F2, F4, F5, F6, F7, F8, and F9). Although it slightly underperformed WOA on F3, its performance was still markedly superior to that of the remaining comparative algorithms, indicating high convergence accuracy and stable search behavior across multiple runs. In terms of the *Std* metric, E-ADE exhibited the smallest fluctuations on eight test functions (F1, F2, F3, F4, F5, F6, F7, and F9), reflecting excellent result consistency and strong robustness against stochastic effects. On F8, although the *Std* value of E-ADE was slightly higher than that of DE and GA, its *Best* and *Mean* values were significantly lower, highlighting its superior convergence accuracy and search efficiency. Overall, the consistently strong performance of E-ADE across all three metrics (*Best*, *Mean*, and *Std*) demonstrates its high stability, robustness, and reliability in solving diverse optimization problems.

A systematic comparative analysis was conducted between seven commonly used mutation strategies in the standard DE algorithm—namely, DE/rand/1, DE/rand/2, DE/best/2, DE/best/1,

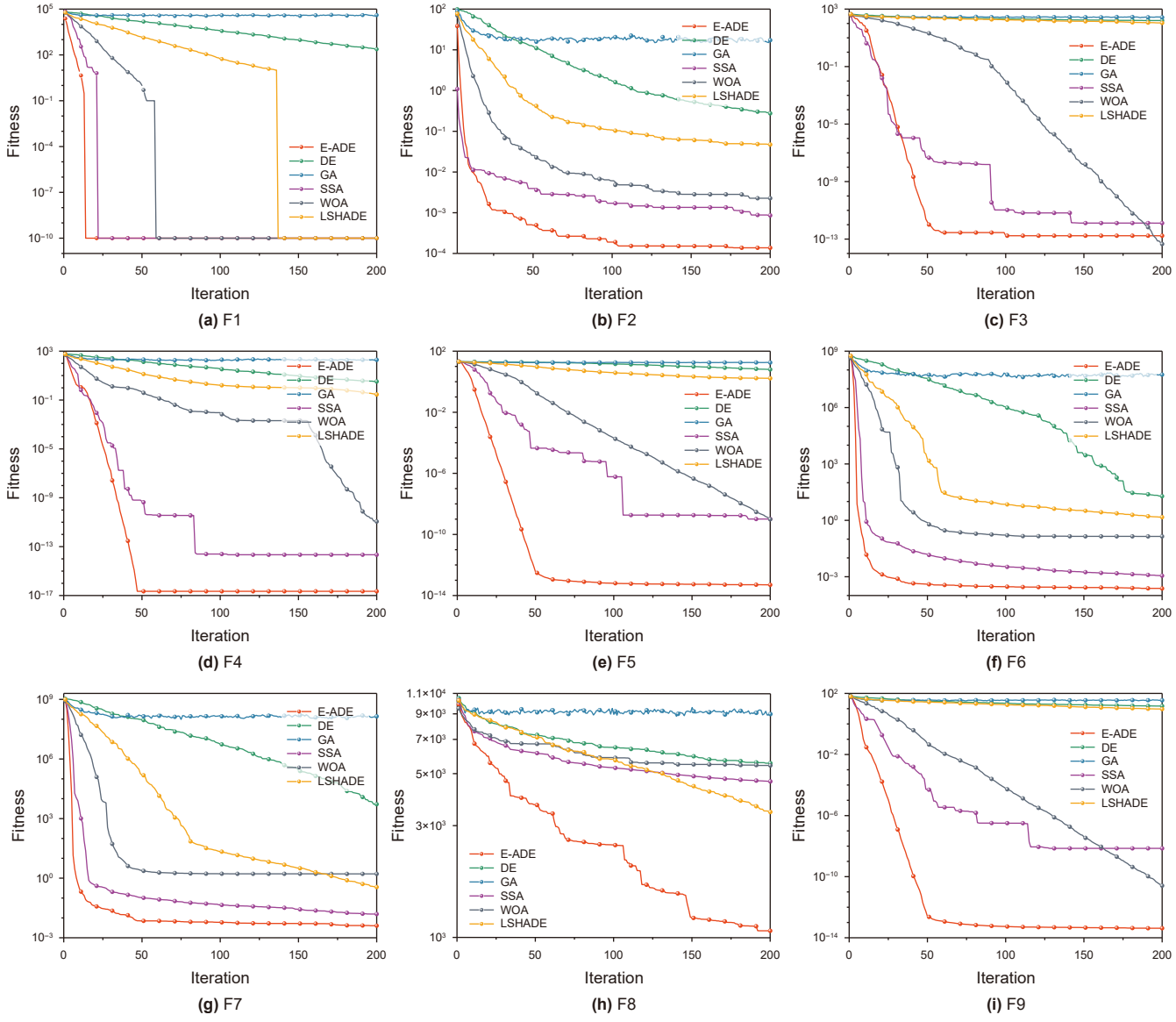

Fig. 4. Test results of different algorithms.

Table 6
Test results of different algorithms.

Test function	Test results	E-ADE	DE	GA	SSA	WOA	L-SHADE
F1	*Best*	**0.00E+00**	1.89E+02	3.23E+04	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	**0.00E+00**	2.26E+02	4.05E+04	**0.00E+00**	**0.00E+00**	2.88E+00
	Std	**0.00E+00**	2.95E+01	5.45E+03	**0.00E+00**	**0.00E+00**	2.10E+00
F2	*Best*	**1.19E−05**	2.13E−01	3.75E+01	1.46E−04	4.78E−04	1.48E−02
	Mean	**2.45E−04**	3.11E−01	5.44E+01	1.11E−03	2.47E−03	4.71E−02
	Std	**1.84E−04**	6.00E−02	1.05E+01	8.20E−04	1.34E−03	2.25E−02
F3	*Best*	**0.00E+00**	1.52E+02	2.30E+02	**0.00E+00**	**0.00E+00**	7.93E+01
	Mean	1.73E−13	1.60E+02	2.65E+02	1.29E−12	**4.55E−14**	1.09E+02
	Std	**5.78E−14**	8.85E+00	2.14E+01	3.94E−12	7.48E−14	1.39E+01
F4	*Best*	**0.00E+00**	2.66E+00	1.33E+02	**0.00E+00**	**0.00E+00**	7.47E−02
	Mean	**2.22E−17**	3.16E+00	1.95E+02	2.49E−14	1.14E−11	2.80E−01
	Std	**7.02E−17**	3.55E−01	3.32E+01	6.86E−14	3.60E−11	1.47E−01
F5	*Best*	3.95E−14	5.39E+00	1.71E+01	**4.44E−15**	1.77E−10	7.21E−01
	Mean	**5.02E−14**	6.32E+00	1.80E+01	1.01E−09	9.88E−10	1.67E+00
	Std	**8.70E−15**	7.63E−01	5.90E−01	2.75E−09	1.16E−09	4.56E−01
F6	*Best*	1.20E−04	1.37E+01	2.19E+07	**5.32E−05**	1.08E−01	6.31E−03
	Mean	**2.37E−04**	1.92E+01	5.62E+07	1.13E−03	1.39E−01	1.46E+00
	Std	**1.15E−04**	4.29E+00	2.76E+07	5.65E−04	2.79E−02	1.31E+00
F7	*Best*	**1.26E−03**	7.98E+01	9.37E+07	2.06E−03	1.37E+00	4.91E−02
	Mean	**4.05E−03**	5.28E+03	1.40E+08	1.50E−02	1.64E+00	3.51E−01
	Std	**1.76E−03**	6.25E+03	3.77E+07	1.25E−02	1.88E−01	3.52E−01
F8	*Best*	**1.15E−01**	4.99E+03	7.92E+03	1.64E+03	4.63E+03	1.85E+03
	Mean	**6.35E+02**	5.56E+03	8.95E+03	4.65E+03	5.44E+03	3.43E+03
	Std	4.43E+02	**2.60E+02**	4.36E+02	2.29E+03	4.45E+02	6.05E+02
F9	*Best*	1.88E−15	1.36E+01	3.05E+01	**2.86E−17**	1.25E−12	7.41E+00
	Mean	**4.19E−14**	1.48E+01	3.53E+01	3.18E−07	2.58E−11	9.37E+00
	Std	**5.63E−14**	7.65E−01	4.20E+00	8.72E−07	4.30E−11	1.59E+00

The bold entries represent the minimum values among all compared results.

DE/rand-to-best/1, DE/current-to-best/1, and DE/current-to-rand/1—and the enhanced mutation strategy proposed in this study (namely, DE/current-to-ebest/1). To ensure fairness and reliability, all mutation strategies were independently executed 10 times on nine benchmark functions under identical parameter configurations.

The comparative results are shown in Fig. 5. It can be observed that, compared with the seven classical DE mutation strategies, the proposed DE/current-to-ebest/1 strategy consistently exhibits superior performance on complex multimodal functions, demonstrating stronger global search capability, faster convergence speed, and higher stability. In the early iterations, the convergence curves of DE/current-to-ebest/1 decline rapidly and approach the global optimum within a short period, while exhibiting minimal fluctuations across repeated runs, indicating both rapid convergence and high robustness.

A further comparative evaluation of different mutation strategies was conducted based on *Best*, *Mean*, and *Std* metrics. The experimental results summarized in Table 7 clearly indicate that the proposed DE/current-to-ebest/1 strategy exhibits significant advantages in optimization performance. Across all nine benchmark functions, this enhanced mutation strategy consistently outperforms the traditional DE mutation strategies according to the evaluation of the *Best*, *Mean*, and *Std* metrics. These results confirm that DE/current-to-ebest/1 possesses stronger global search capability, higher convergence accuracy, and superior robustness, particularly for complex high-dimensional optimization problems.

To further assess the optimization performance of the enhanced mutation strategy, the Wilcoxon rank-sum test was employed to conduct a statistical comparison of the optimization results. The test outcomes for DE/current-to-ebest/1 against the other mutation strategies across nine benchmark functions are summarized in Table 8.

According to statistical standards, a p-value smaller than 0.05 indicates a statistically significant difference in optimization performance. Specifically, when the median performance of DE/current-to-ebest/1 is lower than that of a comparison strategy, the result is marked as "+", indicating superior performance; conversely, when the median is higher, the result is marked as "–", indicating inferior performance. A p-value greater than 0.05 is denoted as " = ", representing no significant difference. As shown in Table 8, DE/current-to-ebest/1 significantly outperforms all other mutation strategies across nine benchmark functions. These results further confirm that the proposed DE/current-to-ebest/1 strategy is both efficient and robust, providing strong statistical evidence for its effectiveness in solving complex optimization problems.

3. Examples and results

3.1. Case study

The PUNQ-S3 reservoir model is a widely recognized benchmark model in reservoir engineering and numerical simulation (Barker et al., 2001; Gu and Oliver, 2005; Zhang et al., 2017). Originally developed from the North Sea PUNQ-S3 project, this model represents a geologically complex and highly heterogeneous reservoir, making it an ideal test case for evaluating various reservoir management strategies. It has been widely utilized in optimization and decision-making studies.

The reservoir is a dome-shaped structure with a gas cap and an underlying oil zone, typical of a hydrocarbon-bearing formation, with dimensions of $19 \times 28 \times 5$ (a total of 2660 grid cells), as seen in Fig. 6. The model includes six production wells and no injection wells, with production starting in January 1967 and simulated until February 2008. As shown in Fig. 7, by the end of the simulation period, the reservoir recovery factor reached 44.1%, while the water cut has risen to 86.8%, indicating that the reservoir has entered a high water-cut and high recovery stage. At this stage, further optimization of reservoir management strategies becomes crucial for enhancing remaining oil recovery and extending the economic life of the reservoir.

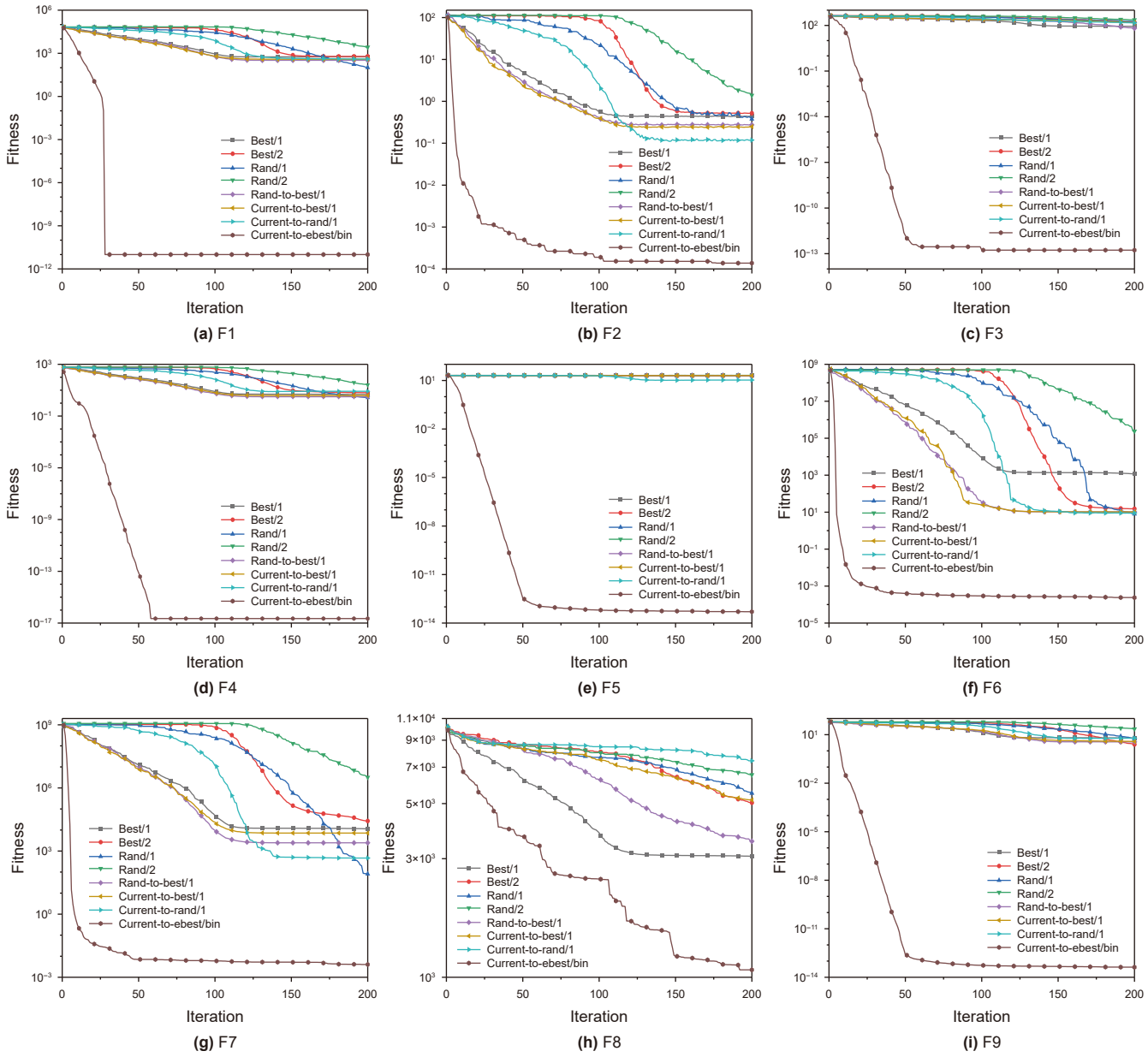

Fig. 5. Test results of different mutation strategies.

Taking the PUNQ-S3 model as an example, the newly established comprehensive intelligent optimization framework was applied to conduct a full-scale well pattern adjustment design. The algorithmic parameters were set as follows: population size of 20, maximum number of iterations of 30, initial mutation factor of 0.8, initial crossover probability of 0.95, maximum iteration stagnation of 10, maximum number of infill wells of 3, maximum number of conversion wells of 2, and no shut-in wells allowed. The minimum well spacing constraint was set to 200 m, the maximum horizontal well length to 700 m, and the production/injection rate for new wells to 300 m^3/d. Based on the benchmark test results, the E-ADE, SSA, and WOA algorithms, which exhibited superior global optimization performance, were selected for further evaluation under identical parameter settings. Each algorithm was independently executed ten times on the PUNQ-S3 model. During each optimization run, well operations and pattern adjustments were evaluated over a 10-year simulation period. Fig. 8 presents the average convergence curves obtained from ten independent optimization runs. It is evident that E-ADE exhibits significantly superior optimization capability and convergence stability compared with WOA and SSA. Specifically, the final objective function value obtained by E-ADE reaches approximately 9.21×10^6 m^3, which is markedly higher than those obtained by WOA ($\approx 8.48 \times 10^6$ m^3) and SSA ($\approx 8.75 \times 10^6$ m^3). Moreover, E-ADE shows rapid improvement in the early iteration stage and maintains continuous progress in the later stage, reflecting its strong global search ability and efficient exploration–exploitation balance. In contrast, WOA converges prematurely around the 10th iteration and becomes trapped in a local optimum, while SSA shows moderate improvement in the later stage but still yields inferior final solutions compared with E-ADE. Owing to its multi-stage updating mechanism, SSA performs multiple position updates and repeated fitness evaluations for the same individual within a single iteration, resulting in a substantially increased computational burden. In this study, the actual number of fitness

Table 7
Test results of different mutation strategies.

Test function	Test results	Rand/1	Rand/2	Best/1	Best/2	Rand-to-best/1	Current-to-best/1	Current-to-rand/1	Current-to-ebest/1
F1	*Best*	5.30E+01	1.70E+03	1.78E+02	2.68E+02	1.48E+02	1.36E+02	1.48E+02	**0.00E+00**
	Mean	1.02E+02	2.67E+03	5.66E+02	6.07E+02	3.26E+02	3.85E+02	4.01E+02	**0.00E+00**
	Std	4.06E+01	5.81E+02	3.17E+02	2.26E+02	1.33E+02	1.84E+02	1.86E+02	**0.00E+00**
F2	*Best*	3.79E−01	6.26E−01	1.96E−01	2.73E−01	1.26E−01	1.06E−01	3.63E−02	**1.19E−05**
	Mean	2.73E−01	1.39E+00	4.48E−01	5.21E−01	2.75E−01	2.49E−01	1.19E−01	**2.45E−04**
	Std	1.52E−01	5.15E−01	3.20E−01	3.29E−01	1.64E−01	1.03E−01	6.03E−02	**1.84E−04**
F3	*Best*	1.58E+02	2.00E+02	5.77E+01	1.47E+02	2.32E+01	1.09E+02	1.26E+02	**0.00E+00**
	Mean	1.77E+02	2.31E+02	8.76E+01	1.64E+02	6.39E+01	1.52E+02	1.46E+02	**1.73E−13**
	Std	1.14E+01	1.82E+01	1.97E+01	1.59E+01	3.18E+01	2.54E+01	1.15E+01	**5.78E−14**
F4	*Best*	2.21E+00	1.50E+01	2.34E+00	2.93E+00	1.25E+00	1.69E+00	5.81E+00	**0.00E+00**
	Mean	2.69E+00	2.59E+01	4.64E+00	6.37E+00	3.10E+00	3.87E+00	8.25E+00	**2.22E−17**
	Std	3.87E−01	4.97E+00	1.67E+00	2.39E+00	2.06E+00	1.50E+00	2.59E+00	**7.02E−17**
F5	*Best*	1.99E+01	2.06E+01	2.00E+01	2.00E+01	2.00E+01	2.00E+01	8.68E+00	**3.95E−14**
	Mean	2.05E+01	2.07E+01	2.00E+01	2.00E+01	2.00E+01	2.00E+01	1.07E+01	**5.02E−14**
	Std	2.68E−01	3.51E−02	1.71E−03	6.47E−02	3.11E−02	1.95E−02	1.13E+00	**8.70E−15**
F6	*Best*	4.20E+00	3.77E+03	1.39E+01	8.50E+00	3.80E+00	3.70E+00	3.93E+00	**1.20E−04**
	Mean	8.07E+00	2.40E+05	7.64E+02	1.52E+01	1.04E+01	1.03E+01	9.01E+00	**2.37E−04**
	Std	2.47E+00	3.18E+05	1.75E+03	5.86E+00	8.58E+00	6.24E+00	1.05E+01	**1.15E−04**
F7	*Best*	2.42E+01	1.35E+06	8.89E+01	3.00E+02	3.51E+01	4.49E+01	1.27E+01	**1.26E−03**
	Mean	8.20E+01	3.19E+06	1.14E+04	2.69E+04	2.47E+03	7.14E+03	4.64E+02	**4.05E−03**
	Std	1.25E+02	2.09E+06	1.38E+04	2.49E+04	2.95E+03	1.18E+04	1.07E+03	**1.76E−03**
F8	*Best*	4.11E+03	5.87E+03	2.45E+03	3.83E+03	2.61E+03	4.18E+03	6.74E+03	**1.15E−01**
	Mean	5.50E+03	6.53E+03	3.07E+03	5.04E+03	3.53E+03	5.17E+03	7.43E+03	**6.35E+02**
	Std	6.57E+02	4.34E+02	**2.93E+02**	5.29E+02	5.85E+02	7.00E+02	4.05E+02	4.43E+02
F9	*Best*	3.92E+00	1.81E+01	3.71E+00	5.07E−01	1.49E+00	2.41E+00	4.70E+00	**1.88E−15**
	Mean	5.57E+00	2.27E+01	6.24E+00	2.50E+00	3.60E+00	3.87E+00	6.09E+00	**4.19E−14**
	Std	1.07E+00	2.14E+00	2.38E+00	1.36E+00	1.13E+00	9.92E−01	1.11E+00	**5.63E−14**

The bold entries represent the minimum values among all compared results.

Table 8
Wilcoxon rank-sum test results of the DE/Current-to-ebest/1 mutation strategy compared with other strategies.

p-value	DE/best/1	DE/best/2	DE/rand/1	DE/rand/2	DE/rand-to-best/1	DE/current-to-best/1	DE/current -to-rand/1
F1	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
F2	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
F3	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
F4	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
F5	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
F6	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
F7	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
F8	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
F9	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
+/ = /-	9/0/0	9/0/0	9/0/0	9/0/0	9/0/0	9/0/0	9/0/0

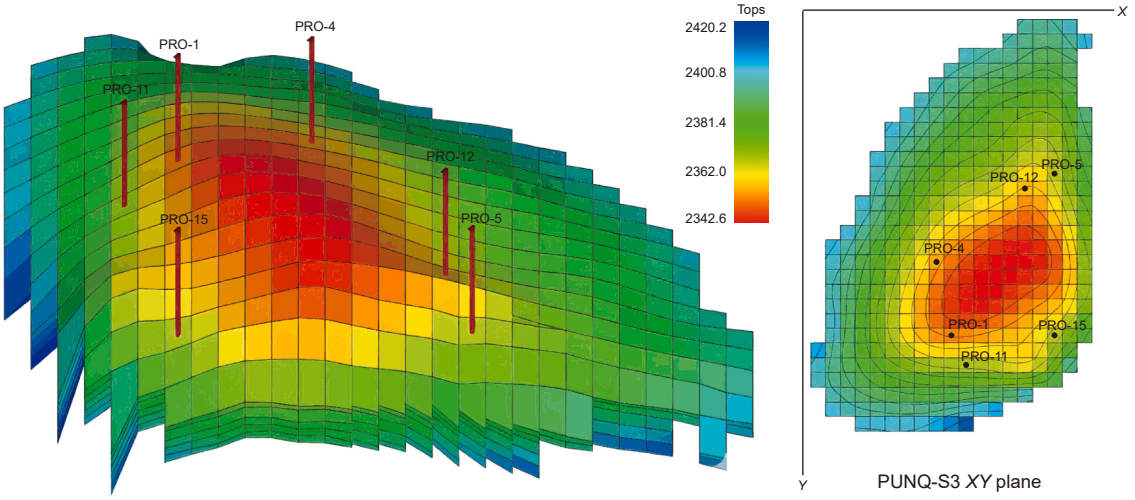

Fig. 6. Location of wells in top structure map.

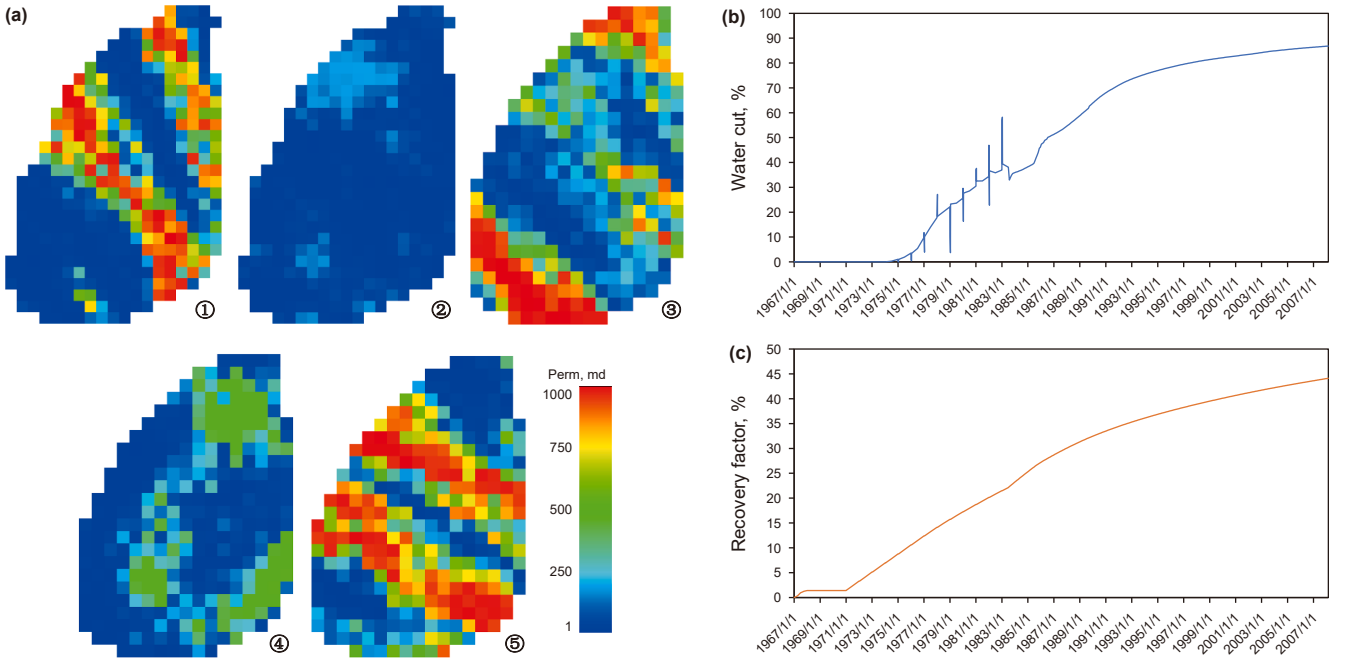

Fig. 7. PUNQ-S3 reservoir model. Panel **(a)** illustrates the permeability distribution across five different layers. Panels **(b)** and **(c)** respectively present the simulation results for water cut and recovery factor.

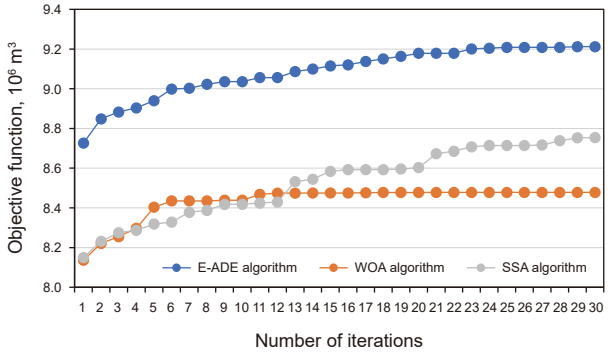

Fig. 8. Convergence curves of the E-ADE, WOA, and SSA algorithms over ten optimization runs.

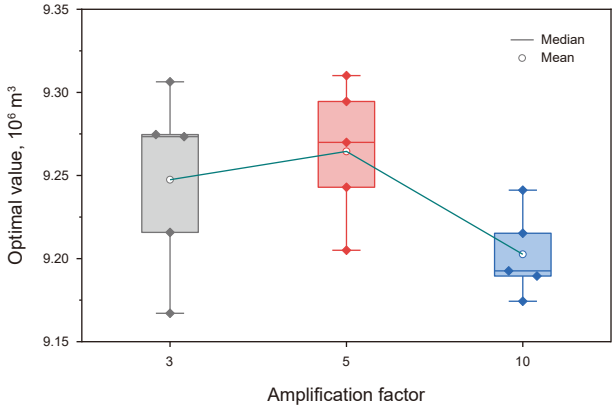

Fig. 9. Sensitivity analysis of the penalty amplification factor.

evaluations required by SSA (1180) is nearly twice its theoretical value (600), which significantly limits its practicality for field-scale optimization applications.

To investigate the influence of the penalty amplification factor ω on the optimization results and to assess the robustness of the proposed method with respect to this parameter, sensitivity tests were conducted with $\omega = 3$, 5, and 10. For each case, five independent optimization runs were performed under identical conditions, and the resulting objective function values were compared, as shown in Fig. 9. The results indicate that when $\omega = 3$, the optimization outcomes exhibit large dispersion, reflected by a broad range of objective values. When $\omega = 10$, the variability decreases considerably; however, the overall objective values also become lower. In contrast, the case with $\omega = 5$ yields the highest mean objective value while maintaining favorable stability across the five runs. These trends can be explained as follows: when ω is too small, the penalty is insufficient to effectively suppress low-productivity wells, resulting in an excessively large search space and increased solution variability. Conversely, when ω is excessively large, the penalty dominates the objective function and induces conservative, penalty-driven convergence, causing the algorithm to favor feasible but suboptimal solutions with lower overall production. Therefore, provided that the minimum production constraint for individual wells is satisfied, a relatively small yet adequately strong penalty factor (e.g., $\omega = 5$) is recommended to balance constraint enforcement, global exploration capability, and solution robustness.

The results of 10 independent optimization runs using E-ADE are shown in Fig. 10. The black line in Fig. 10 (mean curve) denotes the mean objective function value at each iteration, reflecting the average convergence path across all runs. The mean curve exhibits a rapid increase in the early stage followed by a gradual stabilization in the later stage, indicating smooth and stable convergence without significant oscillations and demonstrating the favorable convergence performance of the algorithm. The shaded region in the figure (error band) represents the region within the mean curve ± standard deviation, which quantifies the stability and uncertainty of the optimization results across multiple runs. As shown in Fig. 10, the shaded band is relatively narrow, and

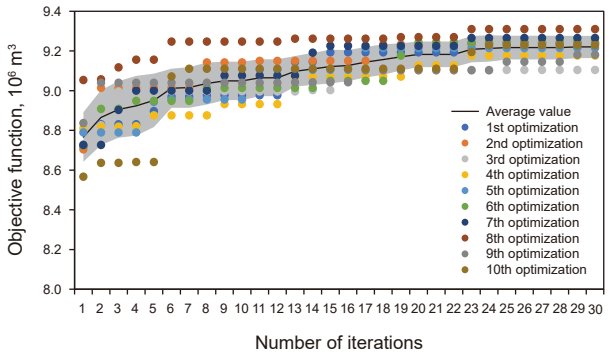

Fig. 10. Convergence curves for 10 optimization runs.

becomes progressively narrower with increasing iterations, indicating that the variability among different runs continuously decreases and the convergence paths become increasingly consistent. These results confirm the high stability, robustness, and reproducibility of the E-ADE algorithm.

Fig. 11 illustrates the relationship among optimizing runtime, total evaluations, feasible solutions, and optimal value. As shown in Fig. 11, the runtimes of the 10 independent optimization runs mainly falls within 2200–2700 s, with total evaluations ranging from 546 to 592, and feasible solutions typically between 400 and 492. Given the population size of 20 and the maximum iterations of 30, the theoretical upper limit of simulator calls per run is 600. The statistical results indicate that the average number of feasible evaluations is approximately 440, and about 160 simulator calls are avoided per run, corresponding to a reduction of invalid evaluations exceeding 26%. This demonstrates the improved computational efficiency achieved by the proposed method. A clear positive correlation is observed between optimization runtime and both total numbers of evaluations and feasible solutions, indicating that as the total evaluations and feasible solutions increase, the optimizing runtime also increases. The optimal objective values obtained from different runs are highly consistent, ranging from 9.10×10^6 m^3 to 9.31×10^6 m^3, further demonstrating the algorithm's strong stability and robustness. Moreover, no significant positive correlation is observed between the optimal value and optimizing runtime, indicating that near-optimal solutions can be achieved within relatively short computational time and that further increases in evaluations yield limited improvement. these results further confirm that the proposed algorithm exhibits superior performance in search efficiency and convergence performance, as it can achieve optimal results while minimizing the computational cost.

Among the 10 optimization scenarios, the optimal scheme was selected for comparison with the unadjusted scheme. The results

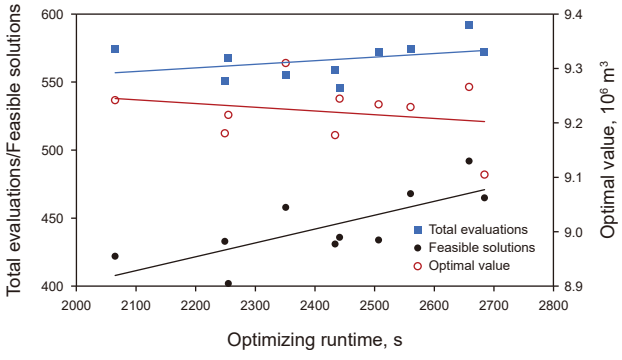

Fig. 11. Relationship curves among optimizing runtime, total evaluations, feasible solutions and optimal value.

indicate that, owing to the generally low water cut of the existing wells, no inefficient wells were shut in. Additionally, due to the adequate edge water energy in the block, no conversion of production wells to injection wells was required. The optimized scheme only involved the infilling of three new production wells deployed around the PRO-15 well. The spatial distribution of the adjusted well pattern is shown in Fig. 12.

Fig. 13 presents the remaining oil saturation distribution after 10 years of production under the unadjusted well pattern and the comprehensively optimized well pattern. It is evident that the optimized infilling of three production wells effectively mobilized the remaining oil around the PRO-15 well that was previously untapped. In particular, the optimized well trajectory of the infilled well P_IN1 penetrates five reservoir layers, significantly improving the vertical utilization. A pronounced reduction in remaining oil saturation is observed in all layers, with the most significant decrease occurring in the bottom three layers.

Fig. 14 further presents a performance comparison for the three infilled production wells designed. As can be seen from Fig. 14, over the predicted 10-year production period, the cumulative oil production (OPT) of these three wells ranges from 3.33×10^5 to 4.75×10^5 m^3, which is significantly higher than the threshold of 5×10^4 m^3 per infill well. These results demonstrate that the optimized infill wells not only substantially enhance the overall productivity of the block but also ensure the economic viability of each well, thereby fully achieving the expected optimization objectives.

3.2. Field application

3.2.1. Reservoir description

The W12-2 Oilfield is a geologically complex fault-block oilfield located in the South China Sea, and was formed through long-term inherited faulting activities. The fault system is predominantly northward-dipping and exhibits a stepped structural configuration descending from south to north, creating a typical domino-like strip fault-block pattern. The reservoir mainly composed of deltaic underwater distributary channel deposits, characterized by small oil-bearing areas and thin individual sand bodies. The primary producing intervals are the LIIa, LIIb, and LIIc oil groups, with relatively thin formation thickness.

The S4 well area is located at the distal end of the sediment supply and was developed at the terminal of underwater distributary channels. The effective thickness is mainly concentrated in the LIIa and LIIb oil groups. The LIIc oil group is dominated by sheet sands and estuarine bars, with relatively thin sand bodies. To avoid degassing during development, water injection is synchronized, and the well pattern adopts a triangular internal water injection

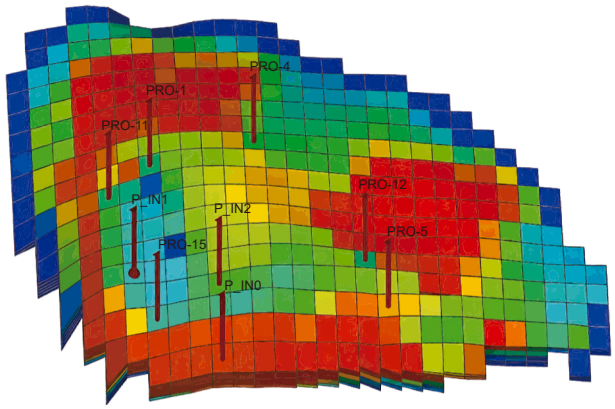

Fig. 12. Well location map after well pattern adjustment optimization.

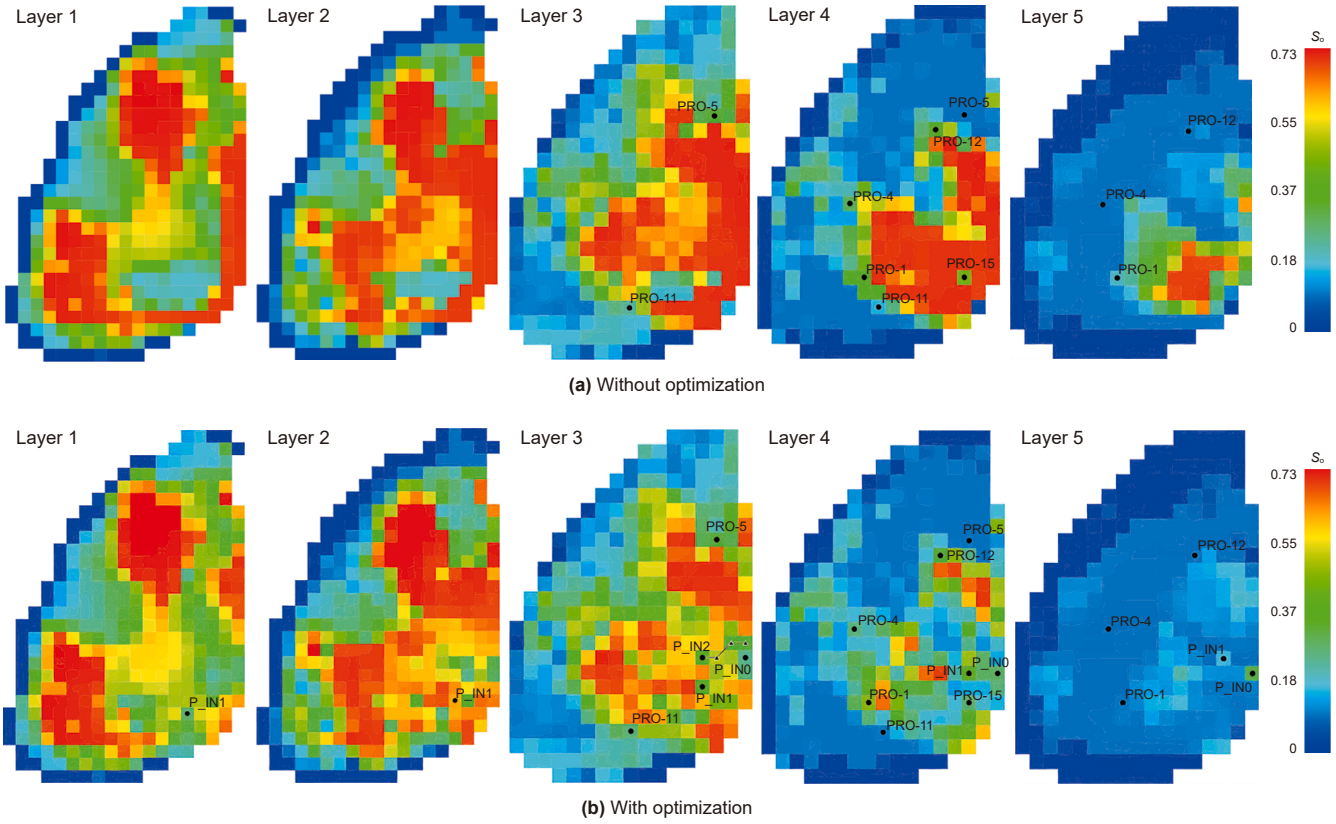

Fig. 13. Remaining oil distribution with or without well pattern adjustment optimization.

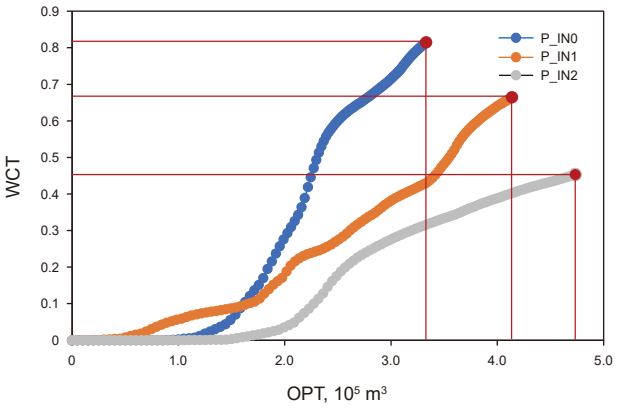

Fig. 14. Cumulative oil production (OPT) and water cut (WCT) relationship curves of three infilled pro-duction wells.

well pattern is adopted. A total of nine wells have been drilled in the block, all of which have been put into production. Currently, three oil wells and one water injection well remain operational, while five wells have been shut in. Fig. 15 shows the permeability and NTG (Net-to-Gross) distribution of the model. Due to the planar distribution of sand bodies, fault blocking, and strong reservoir heterogeneity, the formation pressure exhibits significant differences in the S4 well area, leading to uneven injection-production responses, along with significant variation in water cut. The waterflooding sweep efficiency is only 51%, indicating that comprehensive well pattern adjustment is urgently required to improve development performance.

For the S4 well area of the W12-2 Oilfield, there are currently 3 production wells and 1 injection well in operation. Based on the actual engineering requirements of this area, the optimization plan aims to reactivate 4 shut-in wells, including 2 production wells (B36 and B37) and 2 injection wells (B27 and B35). In the optimization scheme, the maximum numbers of infill wells, conversion wells, and thinning wells (shut-in low-efficiency wells) are set to 1, 2, and 2, respectively. A minimum well spacing constraint of 200 m and a maximum infill well segment length of 200 m are imposed. A five-year production forecast period is adopted as the evaluation horizon, and the proposed comprehensive optimization framework is applied to systematically optimize the well pattern. In terms of algorithm configuration, the population size is set to 20, the maximum number of iterations to 40, the initial mutation factor to 0.85, and the initial crossover probability to 0.95. This optimization covers multiple critical aspects, including infill well placement and trajectory design, low-efficiency well shut-in, and well conversion. Overall, the proposed optimization strategy aims to significantly improve the development performance, optimize the well pattern distribution, and enhance recovery efficiency within the forecast period.

3.2.2. Results and discussion

Fig. 16 presents the convergence curve of the well pattern optimization for the S4 well area of the W12-2 oilfield, illustrating the variation of the objective value (OPT) during the optimization process. In the early stage, the objective value exhibits only minor variation, indicating that the algorithm is primarily in the global exploratory phase and broadly sampling the solution space. With increasing iterations, the objective value gradually improves, reflecting the effectiveness of the optimization strategy and the progressive refinement of solutions. Several significant jumps are observed during the iteration process, suggesting that the superior well pattern adjustment schemes have been identified, resulting in

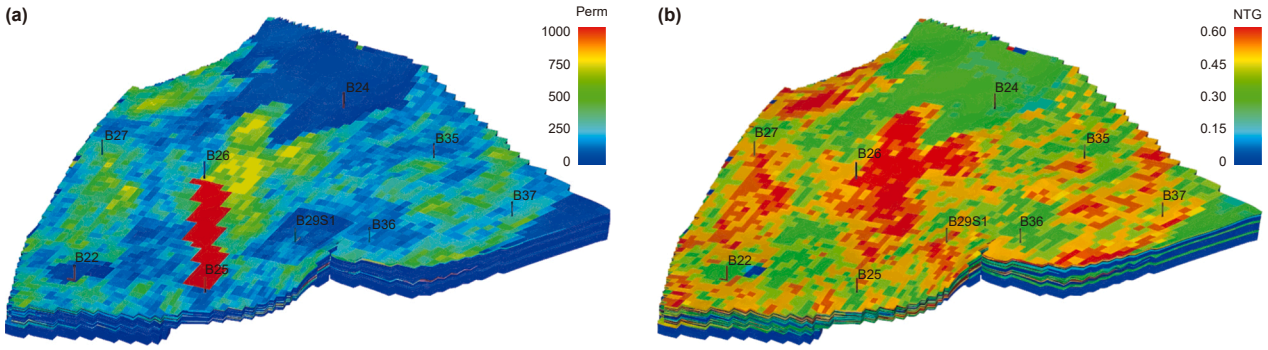

Fig. 15. The permeability and NTG distribution map of the S4 well area: **(a)** Permeability (Perm) and **(b)** NTG.

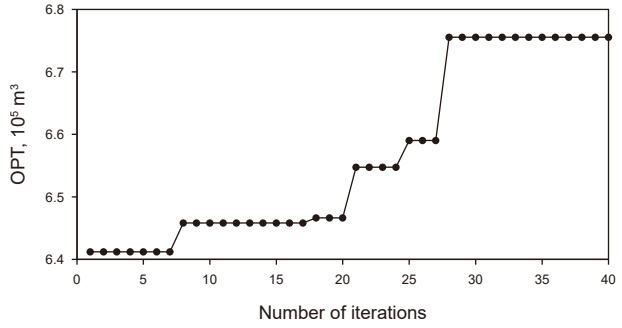

Fig. 16. Convergence curve of well pattern optimization for the S4 well area.

notable performance improvements. After approximately 28 iterations, the objective value stabilizes at 6.756×10^4 m^3, signifying the effective convergence of the optimization process. This confirms that the well pattern optimization for the S4 well area has been successfully realized.

The entire optimization process took approximately 11.5 h. Although the algorithm was originally designed for a total of 800 optimization calculations, the actual number of executed calculations was reduced to 724 due to the adoption of the adaptive population size adjustment strategy. Among these, the reservoir simulator was directly invoked 590 times, while 130 evaluations were handled through the optimization result-caching strategy, and only four invalid solutions were generated. As a result, a total of 210 direct reservoir simulation calls were avoided, saving the optimization time of approximately 26.25%, which significantly improves the overall optimization efficiency. It is worth noting that despite the substantial improvement in computational efficiency, the absolute runtime remains relatively high. This highlights that computational cost is still a major challenge for intelligent optimization algorithms in large-scale reservoir engineering applications, especially in scenarios involving uncertainty quantification or full-field simulation models. In future work, more advanced surrogate modeling techniques (such as neural-network-based proxies and hybrid response surface models) will be introduced, together with parallel and distributed computing frameworks and multi-fidelity simulation strategies, to further accelerate optimization process and improve its scalability.

The final the comprehensive well pattern optimization result for the S4 well area consists of one newly infilled production well (P_IN0) and one production well (B37) converted into an injection well. The optimized well pattern configuration is shown in Fig. 17. Through infill and conversion operations, the optimized scheme better matches the development needs, effectively improving injection-production balance and enhancing oil recovery.

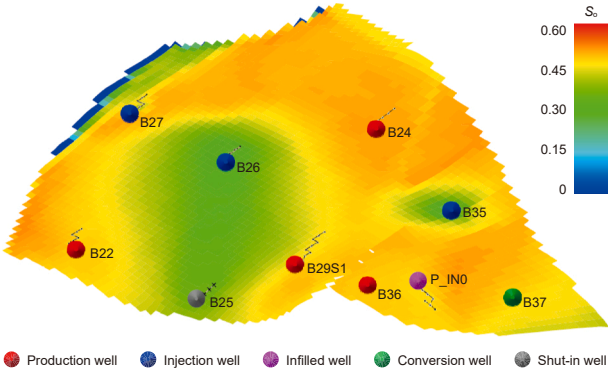

Fig. 17. Optimized well pattern layout of the S4 well area with oil saturation distribution as the base map.

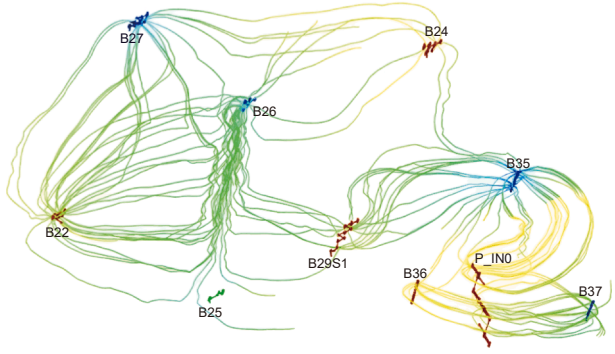

Fig. 18. Streamline field diagram of the well area after comprehensive well pattern optimization.

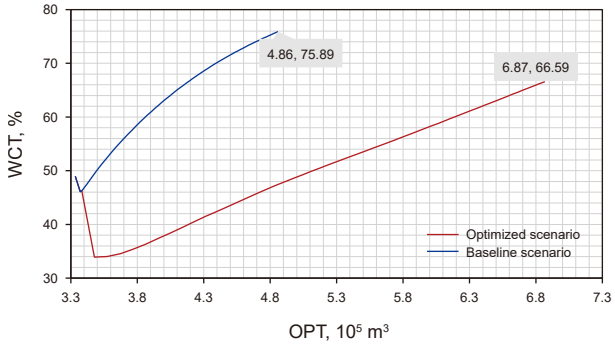

Fig. 19. Development indexes comparison between the baseline and optimized scenarios.

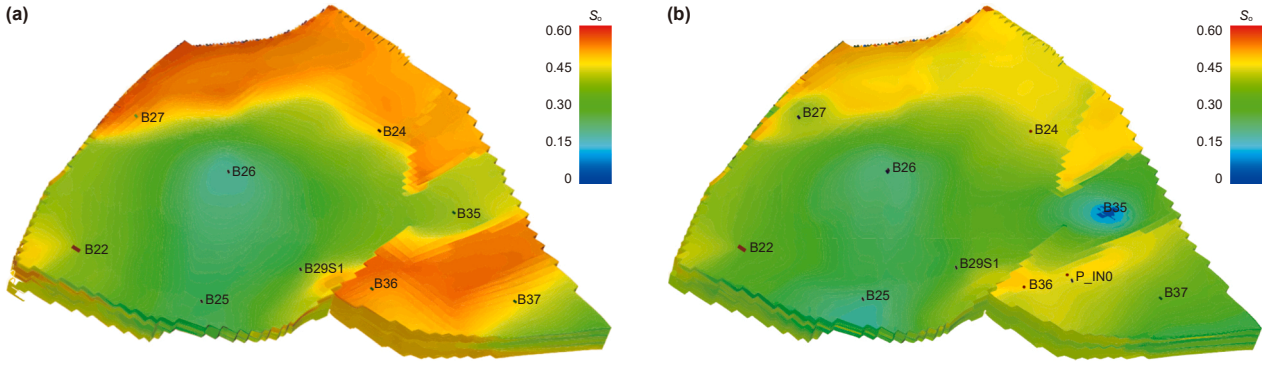

Fig. 20. Distribution map of remaining oil saturation after five-year production forecast for the baseline and optimized scenarios: **(a)** baseline scenario and **(b)** optimized scenario.

As shown in Fig. 17, the optimized infill well P_IN0 is deployed in a high remaining oil zone. The three surrounding wells (B35, B36, and B37), which were previously shut in, were reactivated under the optimized scheme. Among them, well B37 is converted into an injection well to effectively supplement the energy to the local reservoir region.

The flow field distribution shown in Fig. 18 indicates that, after the conversion of well B37, a favorable injection-production relationship is established between B37 and wells B36 and P_IN0. Moreover, a larger proportion of injected water from well B35 flows toward wells B36 and P_IN0, further optimizing the regional injection-production structure and improving the displacement efficiency.

To evaluate the optimization effectiveness, the original operating states of all wells prior to adjustment was maintained, and a five-year production forecast was conducted as the baseline scenario. Fig. 19 compares the cumulative oil production and water cut trends between the baseline and optimized schemes over a five-year production forecast period. As shown in Fig. 19, the optimized scheme exhibits significantly improved development performance compared with the baseline case, characterized by a notable reduction in water cut. By the end of the five-year forecast period, the cumulative oil production of the optimized scheme increases from 4.86×10^5 m^3 under the baseline scheme to 6.87×10^5 m^3, yielding an incremental oil production of 2.01×10^5 m^3.

Fig. 20 presents the remaining oil distribution after five years of production simulation for the baseline and optimized scenarios. Compared with the baseline case, the optimized well pattern significantly improved the displacement of previously immobile remaining oil, leading to a significant improvement in the overall sweep efficiency. The optimized layout not only strengthens the injection-production efficiency but also improves the waterflooding performance, thereby providing a solid foundation for the long-term efficient development of the S4 well area.

4. Conclusions

This study proposes an integrated and scalable optimization framework for well pattern adjustment in large-scale waterflooding reservoirs by jointly considering infill drilling (location, well type, and trajectory) and operational adjustments of existing wells (well type conversion and shut-in decisions). In model construction, multiple adjustment strategies such as infill drilling, shut-in of low-efficiency wells, and injector–producer conversions are comprehensively incorporated. Distinct decision variables and engineering constraints are defined for both new infill wells and existing wells, enabling the formulation of a high-dimensional collaborative optimization model that simultaneously addresses global development efficiency and individual well performance constraints. The proposed framework effectively alleviates the adverse impact of low-efficiency wells and enabling adaptive regulation of production well states.

To solve this complex optimization problem, a novel enhanced adaptive differential evolution (E-ADE) algorithm was developed. The algorithm introduces adaptive control of the scale factor and crossover probability and integrates the follower position update mechanism of SSA with the logarithmic spiral search strategy of WOA. Through an adaptive switching mechanism between spiral-guided elite differential mutation and standard differential mutation, together with dynamic adjustment of differential vectors and population size, E-ADE achieves an efficient balance between global exploration and local exploitation. This design significantly improving its capability for solving complex, multimodal, and high-dimensional optimization problems.

The performance of E-ADE was comprehensively validated using nine classical multimodal benchmark functions. Statistical indicators, including best value, mean, and standard deviation, confirmed that E-ADE consistently outperforms mainstream algorithms such as DE, GA, PSO, WOA, and SSA in convergence speed, optimization accuracy, and robustness. In particular, the proposed DE/current-to-ebest/1 mutation strategy exhibits stronger global search capability, faster convergence, and higher stability compared with seven commonly used DE variants.

Field-scale applications to the PUNQ-S3 model and the S4 block of the W12-2 oilfield further demonstrates the effectiveness of the proposed method in simultaneously optimizing new well deployment and adjust existing wells. The results confirm that collaborative optimization can provide technically feasible and operationally robust development scenarios under stringent offshore engineering constraints, while achieving a robust balance between overall field productivity and single-well performance.

Nevertheless, the current reliance on full-physics reservoir simulation still poses a significant computational burden, which limits large-scale industrial deployment, particularly for uncertainty-driven optimization that requires a large number of realizations. Future work will focus on developing efficient proxy models, exploiting large-scale parallel and distributed computing, and integrating multi-fidelity acceleration strategies to enable full-field, uncertainty-aware optimization within practical computational budgets.

CRediT authorship contribution statement

Xian-Min Zhang: Writing – review & editing, Supervision, Methodology, Conceptualization. **Jian-Gang Yang:** Writing – original draft, Visualization, Investigation. **Qi-Hong Feng:** Writing

– review & editing, Supervision, Funding acquisition. **Ya-Wei Hou:** Writing – review & editing, Validation. **Lei Zhang:** Visualization, Validation.

Data availability

The data used to support the findings are cited within the article. Also, the datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declaration of generative AI in scientific writing

During the preparation of this work the author(s) used ChatGPT and Deep seek in order to improve language and readability, with caution. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

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 Oil & Gas Major Project of China (2025ZD1402901).

List of abbreviations and units

$f(\mathbf{X})$	Objective function, m^3
$\mathbf{X}$	The vector of optimization variables
$C_{o,total}$	Total cumulative oil production of the oilfield, m^3
$C_{o,threshold}$	Cumulative oil production threshold for infill wells, m^3
$C_{o,n}$	Cumulative oil production of the n-th infill well, m^3
N	Maximum number of infill wells
$Q_{threshold}$	Predetermined minimum threshold, m^3/d
$f_{penalty}$	Penalty coefficient
c_i	The well type of the i-th infill well
$x_i^{heel}, y_i^{heel}, z_i^{heel}$	The x, y, and z coordinates of the heel position for the i-th infill well
$\mathbf{\Omega}_i$	The densification boundaries of the i-th infill well
$\theta_{min}, \theta_{max}$	The allowable range for the azimuth angle of the i-th infill well
d_{min}	The minimum allowable distance between wells
C_{oj}^{exist}	Binary variable indicating the shut-in status of the j-th existing well
M	The total number of existing wells
$\mathbf{X}_o^{exist}$	Measure vector
$\lfloor x \rfloor$	The floor function
s_1	The predetermined maximum number of wells allowed to be shut in
$1-\lvert C_{oj}^{exist} \rvert$	Indicator function that identifies wells undergoing well thinning
$\left\lfloor \frac{1-C_{oj}^{exist}}{2} \right\rfloor$	The status of the j-th existing well in the optimization model
r	The perturbation parameter of the chaotic system, a random number between 0 and 1
$\mathbf{x}_i^{(0)}$	The initial population individual
$\mathbf{x}_{lb}, \mathbf{x}_{ub}$	The lower and upper bounds for each individual
$\mathbf{v}_i^{(t)}$	The mutation vector of the i-th individual at the t-th generation
$\mathbf{x}_{best}^{(t)}$	The global best individual at the t-th generation
$\mathbf{x}_{r1}^{(t)}, \mathbf{x}_{r2}^{(t)}, \mathbf{x}_{r3}^{(t)}, \mathbf{x}_{r4}^{(t)}, \mathbf{x}_{r5}^{(t)}$	Individuals randomly selected from current population at the t-th generation
$\mathbf{x}_i^{(t)}$	The position vector of the i-th individual at the t-th generation
F	The scaling factor
F_{max}, F_{min}	The initial and final values of the scaling factor F
T	The current iteration count
$\mathbf{v}_{ibase}^{(t)}$	The basic mutation vector of the i-th individual at the t-th generation
b	The spiral parameter that controls the shape of the spiral trajectory
l	A random number typically drawn from a uniform distribution
L	A $1 \times d$ dimensional matrix with each element being 1
f_{prev}	The global best fitness value from the previous generation
f_{best}	The global best fitness value of the current generation
$F(t)$	An adaptive scaling factor that varies over time
α	A control parameter that linearly decreases from 1 to 0
r_1	A random number within the interval [0,1]
$\mathbf{U}_{i,j}^{(t)}$	The j-th dimension of the offspring individual generated by crossover for the i-th population individual in the t-th generation
$\mathbf{V}_{i,j}^{(t)}$	The j-th dimension of the mutant vector for the i-th individual in the t-th generation
$\mathbf{X}_{i,j}^{(t)}$	The j-th dimension of the i-th individual in the t-th generation
$rand_j$	A uniformly distributed random number in the range [0, 1]
CR	The crossover probability
j_{rand}	A randomly selected dimension index
p_i	The probability of individuals in the population falling into the i-th discretized interval for each dimension
n	The number of discretized intervals
N_{pop}	The population size
D	The problem dimension
IQR	The interquartile range of the data distribution
CR_{max}, CR_{min}	The maximum and minimum crossover probabilities, respectively
H	The theoretical maximum entropy
β	The scaling factor, typically set between 0.1 and 0.3
$N_{pop}^{(t)}$	The population size at generation t
N_{dec}	The number of individuals to be removed
Best	Best value
Mean	Mean value
Std	Standard deviation
DE	Differential Evolution
GA	Genetic Algorithm
SSA	Sparrow Search Algorithm
WOA	Whale Optimization Algorithm
PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization
SaDE	Self-Adaptive Differential Evolution
E-ADE	Enhanced Adaptive Differential Evolution
NPV	Net Present Value
OPT	Cumulative oil production
WCT	Water cut
NTG	Net-to-Gross

References

Ahmad, M.F., Isa, N.A.M., Lim, W.H., et al., 2022. Differential evolution: A recent review based on state-of-the-art works. Alex. Eng. J. 61 (5), 3831–3872. https://doi.org/10.1016/j.aej.2021.09.013.

Alghareeb, Z.M., Walton, S.P., Williams, J.R., 2014. Well placement optimization under constraints using modified cuckoo search. In: SPE Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition, SPE–172841–MS. https://doi.org/10.2118/172841-MS.

Ali, D.H., Al-Jawad, M.S., Van Kirk, C.W., 2015. Distribution of new horizontal wells by the use of artificial neural network algorithm. In: SPE Middle East Oil and Gas Show and Conference, SPE–172599–MS. https://doi.org/10.2118/172599-MS.

Arouri, Y., Sayyafzadeh, M., 2022. An adaptive moment estimation framework for well placement optimization. Comput. Geosci. 26 (4), 957–973. https://doi.org/10.1007/s10596-022-10135-9.

Awad, N.H., Ali, M.Z., Suganthan, P.N., et al., 2017. CADE: A hybridization of cultural algorithm and differential evolution for numerical optimization. Inf. Sci. 378, 215–241. https://doi.org/10.1016/j.ins.2016.10.039.

Ban, D., Lv, X., Wang, X.Y., 2020. Efficient image encryption algorithm based on 1D chaotic map. Comput. Sci 47 (4), 278–284. https://doi.org/10.11896/jsjkx.190600059.

Barker, J.W., Cuypers, M., Holden, L., 2001. Quantifying uncertainty in production forecasts: Another look at the PUNQ-S3 problem. SPE J. 6 (4), 433–441. https://doi.org/10.2118/74707-PA.

Brest, J., Greiner, S., Boskovic, B., et al., 2006. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. 10 (6), 646–657. https://doi.org/10.1109/TEVC.2006.872133.

Cao, J., Sui, D., 2022. Well path design and optimization using composite cubic Bezier curves. SPE J. 27 (6), 3254–3270. https://doi.org/10.2118/209830-PA.

Chaudhary, D., Tailor, A.K., Sharma, V.P., et al., 2019. HyGADE: Hybrid of genetic algorithm and differential evolution algorithm. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). https://doi.org/10.1109/ICCCNT45670.2019.8944822.

Chen, G.D., Li, Y., Zhang, K., et al., 2021. Efficient hierarchical surrogate-assisted differential evolution for high-dimensional expensive optimization. Inf. Sci. 542, 228–246. https://doi.org/10.1016/j.ins.2020.06.045.

Chen, H.W., Feng, Q.H., Zhang, X.M., et al., 2018. A meta-optimized hybrid global and local algorithm for well placement optimization. Comput. Chem. Eng. 117, 209–220. https://doi.org/10.1016/j.compchemeng.2018.06.013.

Chen, H.W., Feng, Q.H., Zhang, X.M., et al., 2017. Well placement optimization using an analytical formula-based objective function and cat swarm optimization algorithm. J. Pet. Sci. Eng. 157, 1067–1083. https://doi.org/10.1016/j.petrol.2017.08.024.

Contreras-Reyes, J.E., Brito, A., 2022. Refined cross-sample entropy based on Freedman-Diaconis rule: Application to foreign exchange time series. J. Appl. Comput. Mech. 8 (3), 1005–1013. https://doi.org/10.22055/jacm.2022.39470.3412.

Demir, F.B., Tuncer, T., Kocamaz, A.F., 2020. A chaotic optimization method based on logistic-sine map for numerical function optimization. Neural Comput. Appl. 32 (17), 14227–14239. https://doi.org/10.1007/s00521-020-04815-9.

Deng, W., Ni, H.C., Liu, Y., et al., 2022. An adaptive differential evolution algorithm based on belief space and generalized opposition-based learning for resource allocation. Appl. Soft Comput. 127, 109419. https://doi.org/10.1016/j.asoc.2022.109419.

Ding, S.W., Lu, R.R., Xi, Y., et al., 2020. Efficient well placement optimization coupling hybrid objective function with particle swarm optimization algorithm. Appl. Soft Comput. 95, 106511. https://doi.org/10.1016/j.asoc.2020.106511.

Forouzanfar, F., Reynolds, A.C., Li, G.M., 2012. Optimization of the well locations and completions for vertical and horizontal wells using a derivative-free optimization algorithm. J. Pet. Sci. Eng. 86, 272–288. https://doi.org/10.1016/j.petrol.2012.03.014.

Gharehchopogh, F.S., Namazi, M., Ebrahimi, L., et al., 2023. Advances in sparrow search algorithm: a comprehensive survey. Arch. Comput. Methods Eng. 30 (1), 427–455. https://doi.org/10.1007/s11831-022-09804-w.

Gu, Y., Oliver, D.S., 2005. History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter. SPE J. 10 (2), 217–224. https://doi.org/10.2118/89942-PA.

Guo, Z.Y., Reynolds, A.C., 2018. Robust life-cycle production optimization with a support-vector-regression proxy. SPE J. 23 (6), 2409–2427. https://doi.org/10.2118/191378-PA.

Güyagüler, B., Horne, R.N., 2004. Uncertainty assessment of well-placement optimization. SPE Reservoir Eval. Eng. 7 (1), 24–32. https://doi.org/10.2118/87663-PA.

Hamid, O., Rahim, Z., Alshakhs, M., et al., 2015. Completion optimization for unconventional reservoir. In: SPE Middle East Oil and Gas Show and Conference, SPE–172772–MS. https://doi.org/10.2118/172772-MS.

Houssein, E.H., Gad, A.G., Hussain, K., et al., 2021. Major advances in particle swarm optimization: Theory, analysis, and application. Swarm Evol. Comput. 63, 100868. https://doi.org/10.1016/j.swevo.2021.100868.

Humphries, T.D., Haynes, R.D., James, L.A., 2014. Simultaneous and sequential approaches to joint optimization of well placement and control. Comput. Geosci. 18 (3), 433–448. https://doi.org/10.1007/s10596-013-9375-x.

Jayabarathi, T., Chalasani, S., Shaik, Z.Z., et al., 2007. Hybrid differential evolution and particle swarm optimization based solutions to short term hydro thermal scheduling. WSEAS Trans. Power Syst. 11 (2), 245–254.

Jesmani, M., Bellout, M.C., Hanea, R., et al., 2016. Well placement optimization subject to realistic field development constraints. Comput. Geosci. 20 (6), 1185–1209. https://doi.org/10.1007/s10596-016-9584-1.

Katoch, S., Chauhan, S.S., Kumar, V., 2021. A review on genetic algorithm: Past, present, and future. Multim. Tools Appl. 80 (5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6.

Khasanov, M., Babin, V., Melchaeva, O., et al., 2014. Application of mathematical optimization techniques for well pattern selection. In: SPE Russian Petroleum Technology Conference, SPE–171163–MS. https://doi.org/10.2118/171163-MS.

Kwon, S., Park, G., Jang, Y., et al., 2021. Determination of oil well placement using convolutional neural network coupled with robust optimization under geological uncertainty. J. Pet. Sci. Eng. 201, 108118. https://doi.org/10.1016/j.petrol.2020.108118.

Li, L.L., Jafarpour, B., 2012. A variable-control well placement optimization for improved reservoir development. Comput. Geosci. 16 (4), 871–889. https://doi.org/10.1007/s10596-012-9292-4.

Li, S.S., Feng, Q.H., Zhang, X.M., et al., 2024. Integrated optimization of well placement and perforation layer selection using a modified dung beetle algorithm. Geoenergy Sci. Eng. 240, 213059. https://doi.org/10.1016/j.geoen.2024.213059.

Li, Y., Wang, D.P., Li, C.L., 2006. Vectorial well arrangement in anisotropic reservoirs. Pet. Explor. Dev. 33 (2), 225–227 and 245. https://doi.org/10.3321/j.issn:1000-0747.2006.02.022.

Li, Y.T., Han, T., Tang, S.Q., et al., 2023. An improved differential evolution by hybridizing with estimation-of-distribution algorithm. Inf. Sci. 619, 439–456. https://doi.org/10.1016/j.ins.2022.11.029.

Lin, J.H., 2002. Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theor. 37 (1), 145–151. https://doi.org/10.1109/18.61115.

Lu, R., Reynolds, A., 2020. Joint optimization of well locations, types, drilling order, and controls given a set of potential drilling paths. SPE J. 25 (3), 1285–1306. https://doi.org/10.2118/193885-PA.

Malallah, A., Alashwak, A., Nashawi, I.S., 2021. Infill well placement optimization in two-dimensional heterogeneous reservoirs under waterflooding using upscaling wavelet transform. J. Pet. Sci. Eng. 201, 108439. https://doi.org/10.1016/j.petrol.2021.108439.

Mishra, P., Pooja Tripathi, S.P., 2024. Optimizing constrained engineering problem nH-WDEOA: Using hybrid nature-inspired algorithm. Int. J. Inf. Technol. 16 (3), 1899–1907. https://doi.org/10.1007/s41870-023-01654-4.

Mohammadi, B., Guan, Y.Q., Moazenzadeh, R., et al., 2021. Implementation of hybrid particle swarm optimization-differential evolution algorithms coupled with multi-layer perceptron for suspended sediment load estimation. Catena 198, 105024. https://doi.org/10.1016/j.catena.2020.105024.

Mousavi, S.M., Bakhtiarimanesh, P., Enzmann, F., et al., 2024. Machine-learned surrogate models for efficient oil well placement under operational reservoir constraints. SPE J. 29 (1), 518–537. https://doi.org/10.2118/217467-PA.

Nadimi-Shahraki, M.H., Zamani, H., Asghari Varzaneh, Z., et al., 2023. A systematic review of the whale optimization algorithm: Theoretical foundation, improvements, and hybridizations. Arch. Comput. Methods Eng. 30 (7), 4113–4159. https://doi.org/10.1007/s11831-023-09928-7.

Nasir, Y., Volkov, O., Durlofsky, L.J., 2022. A two-stage optimization strategy for large-scale oil field development. Optim. Eng. 23 (1), 361–395. https://doi.org/10.1007/s11081-020-09591-y.

Noorizadeh, S., Shakerzadeh, E., 2010. Shannon entropy as a new measure of aromaticity, Shannon aromaticity. Phys. Chem. Chem. Phys. 12 (18), 4742–4749. https://doi.org/10.1039/B916509F.

Onwunalu, J.E., Durlofsky, L.J., 2011. A new well-pattern-optimization procedure for large-scale field development. SPE J. 16 (3), 594–607. https://doi.org/10.2118/124364-PA.

Park, H.-Y., Yang, C.D., Al-Aruri, A.D., et al., 2017. Improved decision making with new efficient workflows for well placement optimization. J. Pet. Sci. Eng. 152, 81–90. https://doi.org/10.1016/j.petrol.2017.02.011.

Piotrowski, A.P., Napiorkowski, J.J., Piotrowska, A.E., 2023. Particle swarm optimization or differential evolution—A comparison. Eng. Appl. Artif. Intell. 121 (106008). https://doi.org/10.1016/j.engappai.2023.106008.

Prasun, S., Wojtanowicz, A., 2018. Determination and implication of ultimate water cut in well-spacing design for developed reservoirs with water coning. J. Energy Resour. Technol. 140 (8), 082902. https://doi.org/10.1115/1.4039743.

Priyadarshi, R., Kumar, R.R., 2025. Evolution of swarm intelligence: A systematic review of particle swarm and ant colony optimization approaches in modern research. Arch. Comput. Methods Eng. 1–42. https://doi.org/10.1007/s11831-025-10247-2.

Qin, A.K., Suganthan, P.N., 2005. Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE Congress on Evolutionary Computation, pp. 1785–1791. https://doi.org/10.1109/CEC.2005.1554904.

Raji, S., Dehnamaki, A., Somee, B., et al., 2022. A new approach in well placement optimization using metaheuristic algorithms. J. Pet. Sci. Eng. 215, 110640. https://doi.org/10.1016/j.petrol.2022.110640.

Reyes-Davila, E., Haro, E.H., Casas-Ordaz, A., et al., 2025. Differential evolution: A survey on their operators and variants. Arch. Comput. Methods Eng. 32 (1), 83–112. https://doi.org/10.1007/s11831-024-10136-0.

Rosić, M.B., Simić, M.I., Pejović, P.V., 2021. An improved adaptive hybrid firefly differential evolution algorithm for passive target localization. Soft Comput. 25 (7), 5559–5585. https://doi.org/10.1007/s00500-020-05554-8.

Sayyafzadeh, M., Pourafshary, P., Rashidi, F., 2010. Increasing ultimate oil recovery by infill drilling and converting weak production wells to injection wells using streamline simulation. In: SPE International Oil and Gas Conference and Exhibition in China, SPE–132125–MS. https://doi.org/10.2118/132125-MS.

Schuetter, J., Mishra, S., Zhong, M., et al., 2015. Data analytics for production optimization in unconventional reservoirs. In: Unconventional Resources Technology Conference, pp. 249–269. https://doi.org/10.15530/URTEC-2015-2167005.

Semnani, A., Ostadhassan, M., Xu, Y.G., et al., 2021. Joint optimization of constrained well placement and control parameters using teaching-learning based optimization and an inter-distance algorithm. J. Pet. Sci. Eng. 203, 108652. https://doi.org/10.1016/j.petrol.2021.108652.

Tanabe, R., Fukunaga, A.S., 2013. Success-history based parameter adaptation for differential evolution. In: 2013 IEEE Congress on Evolutionary Computation, pp. 71–78. https://doi.org/10.1109/CEC.2013.6557555.

Tanabe, R., Fukunaga, A.S., 2014. Improving the search performance of SHADE using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1658–1665. https://doi.org/10.1109/CEC.2014.6900380.

Tian, L., Yang, D.Y., Zheng, S.X., et al., 2018. Parametric optimization of vector well patterns for hydraulically fractured horizontal wells in tight sandstone reservoirs. J. Pet. Sci. Eng. 162, 469–479. https://doi.org/10.1016/j.petrol.2017.12.054.

Wang, X., Haynes, R.D., Feng, Q.H., 2016. A multilevel coordinate search algorithm for well placement, control and joint optimization. Comput. Chem. Eng. 95, 75–96. https://doi.org/10.1016/j.compchemeng.2016.09.006.

Wang, Y., Li, T., Liu, X.J., et al., 2022. An adaptive clonal selection algorithm with multiple differential evolution strategies. Inf. Sci. 604, 142–169. https://doi.org/10.1016/j.ins.2022.04.043.

Wen, T., Thiele, M.R., Ciaurri, D.E., et al., 2014. Waterflood management using two-stage optimization with streamline simulation. Comput. Geosci. 18 (3), 483–504. https://doi.org/10.1007/s10596-014-9404-4.

Xu, A.Z., Mu, L.X., Zhao, L., et al., 2012. A new strategy of well pattern design and adjustment to enhance production of oil fields with vertical multiple series of reservoirs. In: SPE Kuwait International Petroleum Conference and Exhibition. https://doi.org/10.2118/163297-ms.

Xue, J., Shen, B., 2020. A novel swarm intelligence optimization approach: Sparrow search algorithm. Syst. Sci. Control Eng. 8 (1), 22–34. https://doi.org/10.1080/21642583.2019.1708830.

Yang, D.X., Liu, Z.J., Zhou, J.L., 2014. Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization. Commun. Nonlinear Sci. Numer. Simul. 19 (4), 1229–1246. https://doi.org/10.1016/j.cnsns.2013.08.017.

Yang, W.B., Xia, K.W., Fan, S.R., et al., 2022. A multi-strategy whale optimization algorithm and its application. Eng. Appl. Artif. Intell. 108 (104558). https://doi.org/10.1016/j.engappai.2021.104558.

Yussif, A.S., Adjei, S., Wilson, B., 2024. Sigmoid-function-based adaptive pelican optimization algorithm for global optimization. Int. J. Inf. Electr. Eng. Appl. Sci. 7 (2). https://doi.org/10.54554/ijeeas.2024.7.02.003.

Zhang, J.Q., Sanderson, A.C., 2009. JADE: Adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. 13 (5), 945–958. https://doi.org/10.1109/TEVC.2009.2014613.

Zhang, K., Chen, Y.X., Zhang, L.M., et al., 2015. Well pattern optimization using NEWUOA algorithm. J. Pet. Sci. Eng. 134, 257–272. https://doi.org/10.1016/j.petrol.2015.02.017.

Zhang, K., Zhang, H., Zhang, L.M., et al., 2017. A new method for the construction and optimization of quadrangular adaptive well pattern. Comput. Geosci. 21 (3), 499–518. https://doi.org/10.1007/s10596-017-9626-3.

Zhang, Y., Fan, Z., Yang, D., et al., 2017. Simultaneous estimation of relative permeability and capillary pressure for PUNQ-S3 model with a damped iterative-ensemble-kalman-filter technique. SPE J. 22 (3), 971–984. https://doi.org/10.2118/177846-PA.

Zuo, M.C., Guo, C.F., 2022. DE/current-to-better/1: A new mutation operator to keep population diversity. Intell. Syst. Appl. 14, 200063. https://doi.org/10.1016/j.iswa.2022.200063.