Information Sciences 690 (2025) 121593

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A learning-based artificial bee colony algorithm for operation optimization in gas pipelines

Min Liu [a], Yundong Yuan [b], Aobo Xu [a], Tianhu Deng [c], Ling Jian [a,*]

[a] School of Economics and Management, China University of Petroleum, Qingdao, 266580, Shandong, China
[b] SINO-Pipeline International Company Limited, Beijing, 102206, China
[c] Department of Industrial Engineering, Tsinghua University, Beijing, 100084, China

ARTICLE INFO

Keywords:
Optimal control
Gas compressors
Artificial bee colony
Deep reinforcement learning
Multi-label classification

ABSTRACT

The operation optimization of compressors is crucial for powering natural gas transportation and minimizing the gas consumption of the compressors themselves. In the literature, continuous control variables are typically discretized to cope with the curse of dimensionality by traditional dynamic programming methods and meta-heuristics, such as genetic algorithms and ant colony optimization. To provide a more accurate prediction, we developed a learning-based artificial bee colony (ABC) algorithm by integrating deep reinforcement learning. The merits of this innovation lie in two folds: 1) introduces function approximation to address challenges posed by the continuous state associated with gas consumption; and 2) improves the basic ABC's search capacity and reduces the risk of converging into local optima. Furthermore, the technology of multi-label classification is employed in the function approximation method to support the simultaneous optimal control of compressors in all stations, which can significantly enhance decision efficiency. Computational studies on real data demonstrate that the proposed method outperforms existing methods in the literature in terms of gas consumption.

1. Introduction

Pipeline networks are widely used for the long-distance transmission of large volumes of gas. As gas flows through pipelines, pressure reduces due to the friction with the inner pipe walls and heat dissipation to the surroundings [1]. Thus, compressor stations are strategically installed at intervals along pipelines, equipped with multiple turbo-compressors to counteract pressure drops by compressing the gas. Simultaneously, compressors consume 3%∼5% of the transported gas to power their operations. The resulting transportation cost in large-scale pipeline systems, such as the Trans-Asia Gas Pipeline, which has an impressive annual design throughput of 30 billion cubic meters, is significant. In such a capital-intensive industry, pipeline companies can benefit greatly by optimizing the operation of gas compressors, as even minor improvements can yield substantial energy and cost savings. Thus, conducting research on optimal control for the cost-effective operation of gas compressors is imperative [2,3].

The long trip of gas from wellheads to residential and business areas involves gathering, transmission, and distribution systems. And our specific focus on the second segment. Operation optimization in gas pipelines is typical fuel cost minimization problems (FCMP), aiming to generate discharge pressures and feasible compressors' operation schemes that meet pipeline constraints, adhere to compressors' working domain, and fulfill gas demand loads. However, optimizing this process faces a primary challenge: developing

* Corresponding author.
E-mail address: bebetter@upc.edu.cn (L. Jian).

https://doi.org/10.1016/j.ins.2024.121593
Received 28 February 2024; Received in revised form 15 October 2024; Accepted 22 October 2024

Available online 28 October 2024

an efficient algorithm. This problem falls under the category of nonconvex mixed-integer nonlinear programming (MINLP) due to their inclusion of nonlinear objectives, nonconvex constraints associated with the physics of gas, pipelines, and compressors, as well as a combination of continuous and discrete decision variables, such as discharge pressure and the number of operating compressor units [4]. These factors all result in longer computation times and increased challenges in obtaining satisfactory solutions.

Various algorithms have been developed to reduce gas consumption by optimizing compressor operations, including dynamic programming (DP) and meta-heuristics (MHs). DP methods establish a recursive relationship among compressor stations using the pressure drop equation and sequentially conduct optimal control of compressors at each station [4,5]. While DP methods exhibit higher efficiency compared to exhaustive searches over the solution space, they face limitations in handling large, high-dimensional, or continuous control variables due to the "curse of dimensionality". In recent years, a variety of MHs, such as simulated annealing [6,7], tabu search [8], ant colony optimization [9–11], and genetic algorithm [12,13] have been applied to optimize the discharge pressure of compressors, striking a balance between solution quality and computational complexity. However, these methods often exhibit premature convergence and are implemented with the discretization of control variables, which inevitably introduces the discretization error. Motivated by the limitations of existing methods, we propose a learning-based artificial bee colony (ABC) algorithm capable of solving optimization problems with continuous control variables. This approach enhances the basic ABC algorithm by integrating a deep reinforcement learning method, improving its search capacity and mitigating the risk of converging to local optima.

The ABC algorithm shows great performance in various problems, but it also has some limitations when tackling complex problems. It employs one search strategy for both exploration and exploitation, resulting in two main constraints: (*i*) limited search efficiency because only one dimension can be updated at a time; (*ii*) insufficient exploitation due to collecting information from only a randomly chosen neighbor to generate new solutions. The first constraint can be addressed by setting a constant greater than 1 or employing predetermined adaptation methods to control the number of variables to be updated. But defining these control parameters appropriately for the complex optimization problem of compressor operation is challenging, because hyperparameter tuning is time-consuming and heavily relies on the designer's experience. Nowadays, integrating reinforcement learning techniques into MHs has shown promising results [14]. MHs generate substantial data during their iterative search process, including good and bad solutions, the sequence of search operators and local optima [15,16]. Reinforcement learning techniques can extract valuable knowledge from the generated data to guide them towards efficient search strategies, and improve solution quality and convergence rate. Therefore, reinforcement learning is a good choice to intelligent adjust the frequency of perturbation during the employed bee phase based on immediate rewards obtained from solution update outcomes. Considering the crucial factors impacting gas consumption are continuous, a deep reinforcement learning method, actor–critic, is used in this paper. For the second limitation, the current best solutions can benefit exploitation enhancement and are more likely to approach the globally optimal solution. Therefore, the global best solution is employed in the onlooker bee phase to enhance local search. Moreover, we validate our approach with a real operational dataset and demonstrate the superiority of the optimal control schemes derived from our method compared to other common meta-heuristics.

In this paper, we introduce a learning-based algorithm for optimizing operations in gas pipelines, integrating actor-critic and global best information into the basic ABC algorithm to enhance its search efficiency. The novel contributions of this paper are twofold. From the methodological front, we employ two enhanced search strategies in the employed bee phase and the onlooker bee phase to strike a balanced trade-off between exploration and exploitation. On one hand, actor-critic is utilized in the employed bee phase to dynamically adjust the updated dimensions, aiming to improve exploration. On the other hand, global best information is integrated into the onlooker bee phase to enhance exploitation. From the application front, we demonstrate the superiority of our proposed method over other common meta-heuristics on a real gas compressors operational data.

The remaining sections of the paper are organized as follows. In Section 2, we present the operational principles of gas compressors, the basic artificial bee colony algorithm, and our learning-based algorithm. Section 3 examines the optimal results. In Section 4, we summarize our research and discuss potential extensions and future studies.

2. Methodology

Gas consumption optimization takes place at the compressor stations, where controllable variables (e.g., discharge pressure, number of operating compressors) are determined based on hydraulic and thermal conditions (e.g., normalized flow rate, suction states) and the operating range of compressors. To effectively solve this issue, our paper proposes a reinforcement learning-aided ABC algorithm with enhanced search capability. We first provide a brief overview of the gas compressors' operational principles, and then introduce our methodology including the basic and improved ABC algorithm designed to minimize gas consumption.

2.1. Operation principles of gas compressors

Compressors are driven by gas turbines, where impellers work on the gas and convert kinetic energy into static pressure through discharging diffusers [17]. Gas pressure is thus boosted at a compression ratio of the discharge pressure to the suction pressure, accompanied by a portion of gas consumption [18]. The formula describing the gas consumption volume v_{nk} of compressor k in station n is [19]

$$v_{nk} = \frac{q_{nk} h_{nk}}{\eta_m \eta_d \eta_p \mathrm{LHV}}, \tag{1}$$

where q_{nk} is the volumetric flow rate at norm condition. The polytropic efficiency (η_m), driver efficiency (η_d), and mechanical efficiency (η_p) of the compressor can all be obtained by consulting the design documents. LHV is lower heating value of gas, which

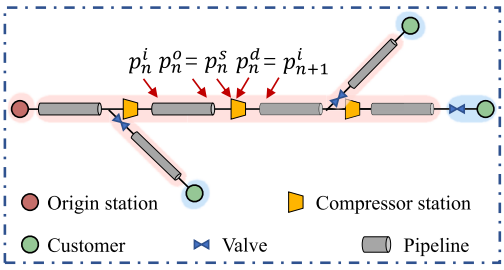

Fig. 1. Schematic of gas pipeline system.

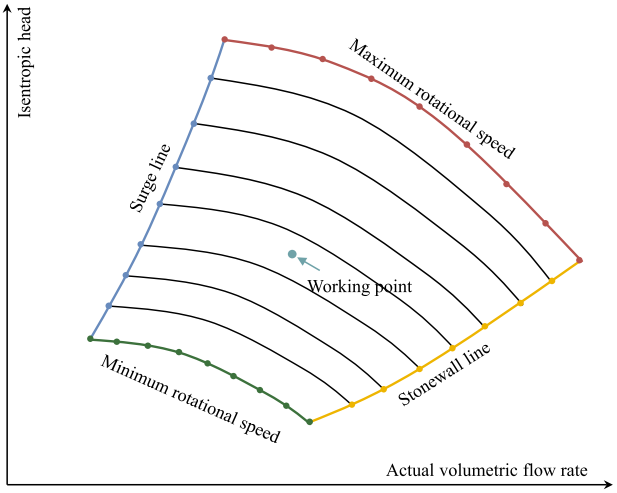

Fig. 2. Working domain of a compressor.

can be calculated by averaging the LHV of each component based on their respective mole fractions and molecular mass. And gas composition can be determined by gas chromatographs. h_{nk} is compressor's isentropic head, quantifying the amount of work required to isentropically compress the gas from suction pressure p_n^s to discharge pressure p_n^d [20]. As compressors within one station are installed in parallel, they all operate with the same isentropic head [21–23]:

$$h_{nk} = \frac{zRT_n^s}{(\kappa-1)/\kappa}\left[\left(\frac{p_n^d}{p_n^s}\right)^{(\kappa-1)/\kappa} - 1\right]. \tag{2}$$

Here, z, κ, R and T_n^s are gas compressible factor, volumetric adiabatic index, specific gas constant and temperature at the suction side of station n. Compressible factor z is used to describe the deviation of real gas from ideal gas and can effectively reflect the gas flow state. Its calculation involves numerous formulations, with the BWRS equation being widely regarded as the one of the most accurate approaches [24], which is shown in Appendix A-1 'BWRS equation of state'. And these parameters can be calculated once the discharge pressure and suction pressure are determined.

Pipeline connects two neighboring compressor stations and realizes the transition from one compressor station's discharge state to the next compressor station's suction state by the pressure drop equation [5,8,9]:

$$p_n^{i2} - p_n^{o2} = \frac{\lambda_n z \Delta T_n^{av} l_n}{c_0^2 d_n^5} q_n^2 \tag{3}$$

where p_n^i and p_n^o are pressures at the inlet and outlet of pipe n. $c_0 = 0.03848(\mathrm{m}^2 \cdot s \cdot K^{1/2} \cdot kg^{-1})$. $\Delta = \frac{m_g}{m_a}$ is the gas relative density, where m_g and m_a are the molecular weight of gas and air, respectively. l_n and d_n are the length and diameter of pipe n, which can be determined based on design documents. Friction coefficient λ_n can be computed based on the surface roughness, diameter and the given normalized flow rate q_n of pipe n, as shown in Appendix A-2 'Friction factor'. The average temperature T_n^{av} is related to the discharge pressure and suction pressure, as outlined in Appendix A-3 'Average temperature'. The relationships between $p_n^i, p_n^o, p_n^d, p_n^s$ are illustrated in Fig. 1. Eq. (1), (2) and (3) imply that we can regulate the gas consumption of each compressor by controlling the discharge pressure p_n^d, and further manage the total consumption of each station by controlling the discharge pressure p_n^d and the number of operating compressors o_n.

Compressor can only operate within its permissible region. Plotting a scatter diagram of feasible isentropic head h_{nk} and actual flow rate q^a_{nk} reveals that the operating envelope is enclosed by four curves (as shown in Fig. 2). The top and bottom sides of curves define the maximum and minimum allowable rotational speed limits, as described by:

$$\omega^{min}_{nk} \le \omega_{nk} \le \omega^{max}_{nk}. \tag{4}$$

The curve on the left side is the surge line constraint:

$$q^a_{nk} \ge \hat{c}_{1nk} + \hat{c}_{2nk}(\omega_{nk}) + \hat{c}_{3nk}(\omega_{nk})^2 - \hat{c}_{4nk}(\omega_{nk})^3; \tag{5}$$

while the curve on right side is the stonewall line:

$$q^a_{nk} \le \hat{c}_{5nk} + \hat{c}_{6nk}(\omega_{nk}) + \hat{c}_{7nk}(\omega_{nk})^2 + \hat{c}_{8nk}(\omega_{nk})^3, \tag{6}$$

where $\{\hat{c}_{jnk} | j = 1, ..., 8\}$ are constants associated with the nature of compressors. A decision variable p^d_n is feasible only if there exists a flow assignment of q_{nk} that ensures every $q^a_{nk} = (q_{nk} z R T^s_n)/p^s_n$ fall within the working domain of compressor k.

2.2. Standard ABC algorithm and its limitations

In this section, we outline the basic process of the standard ABC algorithm and its limitations, laying the conceptual groundwork for future enhancements to the algorithm.

(1) Standard ABC algorithm

The ABC algorithm solves optimization problems by simulating bee colonies foraging in nature, where the bees' food sources denote solutions to problems [25,26]. When solving problems, the ABC algorithm starts with an initialization phase and then iterates through three searching phases until the termination condition is satisfied (maximum iterations = T). And the three searching phases are: (i) the employed bee phase, which seeks new food sources within the entire search space and focuses on exploration or global search; (ii) the onlooker bee phase, which uses information from employed bees to further exploit promising food sources and concentrates more on exploitation or local search; (iii) the scout bee phase, which monitors all food sources to prevent stagnation. The ABC algorithm has an equal number of food sources, employed bees and onlooker bees, and at most, one scout bee. The details and implementation of the initialization phase and the three searching phases are described as follows:

Initialization phase. ABC algorithm is initialized within the search space by randomly generating M food sources using:

$$x_{i,j} = x_{min,j} + \text{rand}(0, 1) \times (x_{max,j} - x_{min,j}). \tag{7}$$

Here, $x_i = (x_{i,1}, x_{i,2}, .., x_{i,N})$ represents the i-th food source, where N is the dimension of the problem. Each $x_{i,j}$ corresponds to the value of i-th ($i = 1, ..., M$) food source' j-th ($j = 1, ..., N$) variable, with $x_{min,j}$ and $x_{max,j}$ as the lower and upper bounds of the jth dimension, respectively. M is also the number of employed bees and onlooker bees.

Employed bee phase. Based on the existing food source x_i, each employed bee generates a new one using the following search strategy:

$$v_{i,j} = x_{i,j} + \phi_{ij} \times (x_{k,j} - x_{i,j}) \tag{8}$$

where $k(\neq i)$ and j are randomly chosen from $\{1, ..., M\}$ and $\{1, ..., N\}$ respectively; ϕ_{ij} is a uniform random number in the range of $[-1, 1]$. Only one dimension, $x_{i,j}$ of x_i, is updated to produce v_i. If the objective function of v_i, $f(v_i)$, is better than that $f(x_i)$ of x_i, v_i displaces x_i and the counter c_i is reset to 0. Otherwise, x_i remains unchanged and the counter c_i increments by one.

Onlooker bee phase. Promising food sources are further exploited in this phase. So each onlooker bee selects a candidate solution by obeying probability value p_i, which is calculated from fitness function fit_i via roulette wheel selection as follows:

$$\begin{aligned} \text{fit}_i &= \begin{cases} \dfrac{1}{1 + f(x_i)} & \text{if} f(x_i) \ge 0 \\ 1 + |f(x_i)| & \text{otherwise} \end{cases} \\ p_i &= \frac{\text{fit}_i}{\sum_{j=1}^{M} \text{fit}_j} \end{aligned} \tag{9}$$

where fit_i and $f(x_i)$ are the fitness value and objective function of food source x_i. This ensures that better food sources are more likely to be chosen. Then the chosen food sources are updated using the same rule as in the employed bee phase.

Scout bee phase. To deal with the issue of local optima, if a food source has not been improved within a predefined period $limit$, it is replaced by a new one generated using Eq. (7). And it's counter c_i is reset to 0.

(2) Limitations of the ABC algorithm

The basic ABC algorithm contains a global search (exploration) in the employed bee phase and a local search (exploitation) in the onlooker bee phase. Both phases generate new solutions using the Eq. (8). Thus, the performance of ABC algorithm heavily depends on the effectiveness of Eq. (8). Actually, Eq. (8) exhibits apparent defects and limitations: (i) From the exploration front, Eq. (8) randomly selects only one dimension, $x_{i,j}$, of the food source x_i each time for updating to generate a new feasible solution, which hinders the algorithm's ability to explore the solution space effectively. (ii) From the exploitation front, Eq. (8) only collects information from a

randomly chosen neighbor, without making full use of the current best information in the iterative trajectories. However, the current best solutions can benefit exploitation enhancement and are more likely to approach the globally optimal solution. (*iii*) The basic ABC algorithm employs a single search strategy, Eq. (8), for both exploration and exploitation, which may pose challenges in striking the right balance between the two strategies. Therefore, to effectively address the complex gas consumption optimization problem using the ABC algorithm, it is essential to implement improvements to the basic ABC algorithm.

2.3. Learning-based ABC algorithm

To address the limitations mentioned in Section 2.2, we adopt a two-fold approach in which the search behavior of the onlooker bee differs from that of the employed bee for distinct purposes. On one hand, we use actor-critic method to intelligently adjust the frequency of perturbation during the employed bee phase based on immediate rewards obtained from solution update outcomes. The actor-critic architecture is a powerful RL framework that integrates the advantages of both value-based and policy-based techniques. It has demonstrated exceptional performance and superior stability across a range of large-scale problems with continuous state representations. During the training process, the actor-critic method perceives the state of each food source and then determines the dimensions to be updated in each employed bee phase. Then, the reward can be obtained by comparing the quality of the food sources before and after the update and a new food source is generated. The objective of this learning-based adaptive approach is to discover parameter configurations of actor-critic method that maximize the improvement of the initial solution's performance to prevent getting trapped in local optima and enhance global search ability. On the other hand, besides the randomly chosen neighbor, we incorporate the neighbor that carries information about the current optimal solution into the update equation, Eq. (8), during the onlooker bee phase. This modification is achieved by integrating the update equation of the particle swarm optimization algorithm (PSO). Therefore, it allows new individuals learn more information from their neighbors and guide the search towards a more promising direction.

Like any ABC algorithm, it is essential to begin by identifying the fundamental elements. Here, the discharge pressures of all compressor stations serve as the i-th food source $x_i = \{p^d_{i1}, ..., p^d_{in}, ..., p^d_{iN}\}$, with the number of compressor stations representing its dimension N. Then, for each station n in food source x_i, the number of running compressors o_{in} is determined by an enumeration method to achieve the objective of minimum gas consumption. The total consumption of all stations is used to measure the quality of food source x_i. The overall framework of the proposed approach is shown in Algorithm 1.

(1) Leaning-based global search

The solution search equation, Eq. (8), in the employed bee phase limits the search efficiency because only one dimension can be updated at a time. This constraint can be easily addressed by setting a constant greater than 1 or employing predetermined adaptation methods to control the number of variables to be updated. However, defining these control parameters appropriately for the complex optimization problem of compressor operation is challenging because hyperparameter tuning is time-consuming and heavily relies on the designer's experience. Reinforcement learning is known for its ability to learn from experience and adapt to changing conditions. Therefore, in this paper, we integrate reinforcement learning into the ABC algorithm to determine which dimensions $d_{i,up}$ to be updated for each food source x_i in the employed bee phase by leveraging the feedback of iterative information. This enables flexible adjustment of $d_{i,up}$ at different stages of the search process, thereby enhancing exploration of the solution space and improving the efficiency and robustness of the ABC algorithm. Prior to our work, [27] incorporated Q-learning to dynamically select the number of changing dimensions nb_{up} in Eq. (8) during the employed bee phase. However, they randomly selected the nb_{up} variables to be modified from the N available variables. In the optimal control of gas compressors, adjusting the discharge pressure of different stations (updating different dimensions in the food source) can significantly impact the total gas consumption. Furthermore, Q-learning struggles to handle high-dimensional state spaces caused by continuous variables like discharge pressure. Therefore, we integrate actor-critic into the ABC algorithm to solve the operational optimization problem in gas pipelines.

The basic ABC algorithm begins with an initialization phase and then iterates through the employed bee phase, the onlooker bee phase, and the scout bee phase T times. Therefore, an episode in reinforcement learning corresponds to the maximum number of iterations T in ABC. At time step 0, M food sources $\{x_{0,1}, ..., x_{0,M}\}$ are generated in the initialization phase, and each food source $x_{0,i}$ has its initial state $s_{0,i}$ (further details will be introduced later). At each time step t, the actor-critic method with an actor network θ_a and a critic network θ_c is employed to sequentially determine $d_{i,up}$ (i.e., action $a_{t,i}$) for each food source $x_{t-1,i}$ among the M food sources based on their respective state $s_{t-1,i}$, rather than setting the same value for the entire population. The reward $r_{t,i}$ is assigned based on the impact of the action $a_{t,i}$ on food source $x_{t-1,i}$, and a new candidate solution $x_{t,i}$ with the updated state $s_{t,i}$ is generated, as depicted in Fig. 3. Thus, M samples $\{(s_{t-1,1}, a_{t,1}, r_{t,1}, s_{t,1}), ..., (s_{t-1,M}, a_{t,M}, r_{t,M}, s_{t,M})\}$ are obtained at time step t to update the parameters (θ_a, θ_c) using batch gradient update.

At each time step t, the actor-critic method sequentially operates on M food sources. And the Markov decision process for determining which dimensions $d_{i,up}$ to update for each food source $x_{t,i}$ is defined in detail as follows:

State. The state representation should capture the key characteristics of each food source $x_{t,i}$ and the critical factors affecting gas consumption. Therefore, state representation $s_{t,i}$ of food source $x_{t,i}$ is defined as discharge pressures, pressure ratios (the ratio of discharge pressure to suction pressure), and whether the newly generated solution better than the old one. For food source $x_{t,i}$ with N dimensions, its state $s_{t,i}$ has $2N + 1$ dimensions. The states of M food sources are accordingly denoted as $\{s_{t,1}, ..., s_{t,M}\}$. And discharge pressure and pressure ratio in the state representation are continuous. When the state space is continuous, evaluating a function at every possible state or state–action pair becomes challenging or even impossible [28,29]. To address this issue, deep reinforcement learning can be employed to estimate the value or action value function with a parametric or nonparametric approach. Therefore, actor-critic is used in this paper.

Algorithm 1. Pseudo code of learning-based ABC.

Input: number of solutions M, $limit$ parameter value, maximum iterations T, actor network parameters θ_a and critic network parameters θ_c;

```
// Initialization phase;
for i = 1 to M do
    Generate initial solution x_{0,i} with Eq. (7), evaluate it;
    Store the best solution so far
    Initialize state s_{0,i};
    c_{0,i} = 0;
end
for t = 1 to maximum iteration T do
    // Employed bee phase;
    for i = 1 to M do
        Choose an action a_{t,i} via actor-critic approach or random policy;
        Perform action a_{t,i} on adjusting d_{i,up} and generate a candidate solution v_{t,i} with equation shown in Fig. 3;
        if f(v_{t,i}) < f(x_{t-1,i}) then
            Assign v_{t,i} to x_{t,i}; c_{t,i} = 0;
            s_{t-1,i} → s_{t,i}; reward r_{t,i} = 1;
        else
            Assign x_{t-1,i} to x_{t,i};
            c_{t,i} = c_{t-1,i} + 1;
            s_{t-1,i} → s_{t,i}; reward r_{t,i} = −1;
        end
    end
    Update actor network parameters θ_a with Eq. (17) critic network parameters θ_c with Eq. (14);
    Calculate fitness values of solutions and assign probability values by Eq. (9);
    // Onlooker bee phase;
    for i = 1 to M do
        Choose a solution x_{t-1,i} based on probability p_i;
        Update x_{t-1,i} to v_{t,i} via Eq. (18) and evaluate v_{t,i};
        if f(v_{t,i}) < f(x_{t-1,i}) then
            Assign v_{t,i} to x_{t,i}; c_{t,i} = 0;
        else
            Assign x_{t-1,i} to x_{t,i};
            c_{t,i} = c_{t-1,i} + 1;
        end
    end
    // Scout bee phase;
    for i = 1 to M do
        if c_{t,i} > limit then
            Generate a new solution x_{t,i} with Eq. (7); c_{t,i} = 0;
        end
    end
    Store the best solution so far.
end
```

$$
\begin{bmatrix} v_{t,i,1} \\ v_{t,i,2} \\ \vdots \\ v_{t,i,j} \\ \vdots \\ v_{t,i,N} \end{bmatrix}
=
\begin{bmatrix} x_{t-1,i,1} \\ x_{t-1,i,2} \\ \vdots \\ x_{t-1,i,j} \\ \vdots \\ x_{t-1,i,N} \end{bmatrix}
+
\begin{bmatrix} a_{t,i,1} \\ a_{t,i,2} \\ \vdots \\ a_{t,i,j} \\ \vdots \\ a_{t,i,N} \end{bmatrix}
\times
\begin{bmatrix} \phi_{t,i,1} \\ \phi_{t,i,2} \\ \vdots \\ \phi_{t,i,j} \\ \vdots \\ \phi_{t,i,N} \end{bmatrix}
\times
\begin{bmatrix} x_{t-1,k,1} - x_{t-1,i,1} \\ x_{t-1,k,2} - x_{t-1,i,2} \\ \vdots \\ x_{t-1,k,j} - x_{t-1,i,j} \\ \vdots \\ x_{t-1,k,N} - x_{t-1,i,N} \end{bmatrix}
$$

Fig. 3. The new updating rule in the employed bee phase.

Action. At each time step t, the actor-critic method with parameters (θ_a, θ_c) executes actions a_{1t} on food source $x_{t-1,1}$, $a_{t,2}$ on food source $x_{t-1,2}$, and so forth, until $a_{t,M}$ on food source $x_{t-1,M}$, as illustrated in Fig. 4. More specifically, for each food source $x_{t,i}$, the agent is tasked with making a binary decision for each of the N dimensions. Thus, the action $a_{t,i}$ is defined as an N-dimensional vector, where 1 corresponds to updating the dimension using Eq. (8), and 0 corresponds to keeping the dimension unchanged. This decision-making process is framed as a sequential binary classification task, where a series of binary decisions are made for each of the M food sources. Therefore, the action space can be represented as $A=\{a_1, a_2, ..., a_j, ..., a_{2^N}\}$, where a_j is described as in Table 1

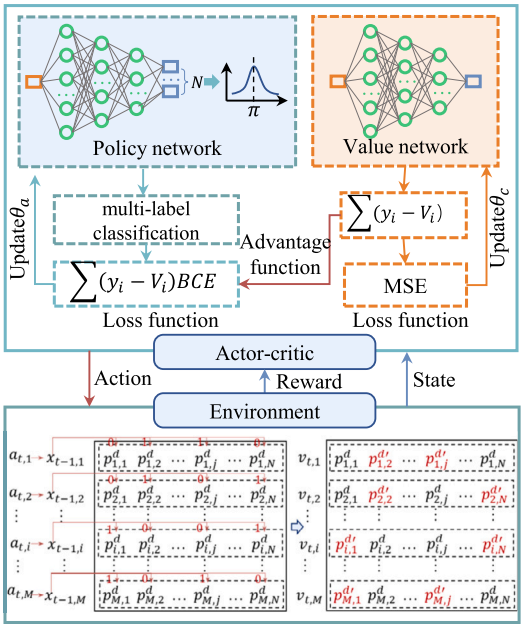

Fig. 4. The architecture of the actor-critic approach.

Table 1
Action representation.

a_1	a_2	...	a_{2^N-1}	a_{2^N}
$[0,0,...,0]_N$	$[0,0,...,1]_N$	...	$[1,1,...,0]_N$	$[1,1,...,1]_N$

and $a_{j,1},...,a_{j,N} \in \{0,1\}$. Subsequently,the new candidate solution for food source $x_{t,i}$ is generated using the updated rule as shown in Fig. 3.

Policy. Actor-critic method [30] employs an actor (policy) network to select actions and a critic (value) network to evaluate the chosen actions and provide guidance for updating the actor network. The actor network, implemented as a deep neural network with two fully connected layers, extracts features from the state $s_{t,i}$ and outputs the probability of updating each dimension in food source $x_{t,i}$. However, sequentially outputting decisions for each of the N dimension is inefficient. Motivated by [31], we design the actor network to make simultaneous decisions for all of the N dimensions in food source $x_{t,i}$ by employing the activation layer with the same number of units as the dimensions of food source.

In our paper, we utilize sigmoid as the activation function for the final layer of the actor network. Unlike the commonly-used softmax function, sigmoid guarantees that each output unit independently represents an updating probability for the corresponding dimension of food source $x_{t,i}$, without the constraint of the sum of outputs equaling 1. The application of the sigmoid function transforms the decision-making problem from a sequential binary classification task to a more efficient multi-label classification task. This means that the number of updating dimensions in each food source is uncertain, and the decisions for the N dimensions can be sampled separately according to the probability p. The process of the actor network is presented as shown below:

$$x = FC(s_{it}); \tag{10}$$

$$x = FC(x); \tag{11}$$

$$p = sigmoid(x) = 1/(1+e^{-x}). \tag{12}$$

And the structure of the critic network follows the same design as presented in [30].

In addition, as a common strategy to promote exploration, the ϵ-greedy strategy is adopted, where the agent has a 5% probability of choosing a random action.

Reward. Since the objective is to minimize the gas consumption, so the reward function focuses on the intermediate impact of the new solution generated using the updating rule shown in Fig. 3. Thus, the reward r_{it} is defined based on the comparison between the original solution and the newly generated one [32]. Specifically, the reward is +1 if $f(v_{t,i}) < f(x_{t-1,i})$, i.e., new equation improves the current solution and -1 otherwise.

Loss function. The actor-critic approach comprises two parts: actor and critic, each with a corresponding loss function. At each time step t, the actor-critic method with parameters (θ_a, θ_c) respectively performs actions $a_{t,1},...,a_{t,i},...,$ $a_{t,M}$ on food sources $x_{t-1,1},...,x_{t-1,i},...,x_{t-1,M}$ based on their respective states $s_{t-1,1},...,s_{t-1,i},...,s_{t-1,M}$. Subsequently, the correspond-

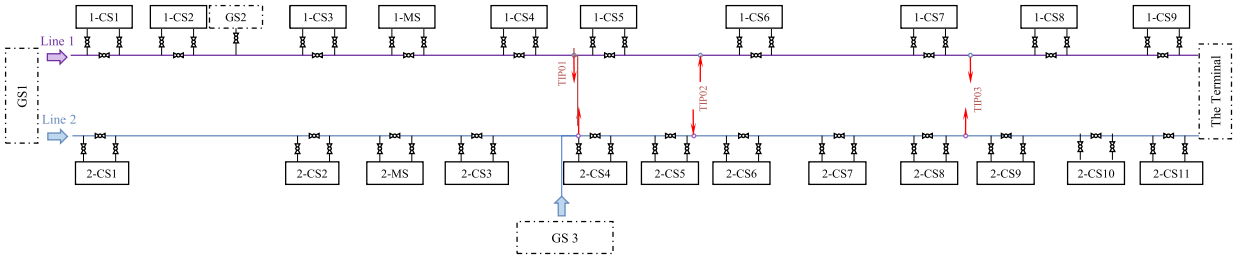

Fig. 5. Schematic of pipeline system.

ing rewards $r_{t,1},...,r_{t,i},...,r_{t,M}$ are assigned, and new candidate solutions $x_{t,1},...,x_{t,i},...,x_{t,M}$ with the updated states $s_{t,1},...,s_{t,i},...,s_{t,M}$ are obtained. These M samples $\{(s_{t-1,1},a_{t,1},r_{t,1},s_{t,1}),...,(s_{t-1,M},a_{t,M},r_{t,M},s_{t,M})\}$ are collectively utilized to update the parameters (θ_a,θ_c) to (θ'_a,θ'_c) through batch gradient update.

The critic can be viewed as a function approximator to estimate the value function, which tries to make the predict value $V_i = V\left(s_{t-1,i}\right)$ as close as possible to the target value $y_i = r+\gamma V_{(i+1)}$ for each food source $x_{t-1,i}$. Therefore, we aim to minimize the mean squared error (MSE), calculated as:

$$L_{critic} = \mathbb{E}(y-V)^2 = \frac{1}{M}\sum_{i=1}^{M}(y_i - V_i)^2. \tag{13}$$

And the parameters θ_c of the critic network can be updated using gradient descent as follows:

$$\theta'_c = \theta_c - \frac{1}{2}\alpha\nabla_{\theta_c}\mathbb{E}(y-V,\theta_c)^2. \tag{14}$$

The actor network simultaneously determines whether to update any of the N dimensions in each food source $x_{t-1,i}$. As mentioned earlier, the decision process is a multi-label classification task, for which binary cross-entropy loss (BCE loss) [33] is preferred, shown as follows:

$$\mathrm{BCE}_i = -\frac{1}{N}\sum_{j=1}^{N}\left(t_j\log(o_j)+(1-t_j)\log(1-o_j)\right), \tag{15}$$

where o_j, t_j represent the actor's outputs and targets for the food source. To reduce the variance arising from the use of sampled trajectories, a value-based baseline is incorporated into the policy-based methods. The actor loss function can be computed as follows:

$$L_{actor} = \mathbb{E}\left((y-V)\mathrm{BCE}\right) = \frac{1}{M}\sum_{i=1}^{M}(y_i - V_i)\mathrm{BCE}_i. \tag{16}$$

The actor network parameters θ_a will be updated through policy gradient, i.e.,

$$\theta'_a = \theta_a - \alpha\nabla_{\theta_a}\mathbb{E}\left((y-V)\mathrm{BCE},\theta_a\right). \tag{17}$$

(2) Optimal-trending local search

The primary purpose of the onlooker bee phase in the basic ABC algorithm is to explore promising solutions more effectively. However, according to Eq. (8), the updating of promising solutions can only proceed in the direction of a randomly chosen neighbor, which may have an uncertain impact. Therefore, many improved ABC variants incorporate the information of the global best solution to modify the solution search equation Eq. (8) [34,35]. For example, [36] added a difference vector $x_{best}-x_i$ to the Eq. (8). [37] employed various strategies based on x_{best}. Drawing inspiration from this, we revise the updating equation by incorporating the PSO as follows:

$$v_{i,j} = x_{i,j} + \xi_1 v_1(x_{k,j} - x_{i,j}) + \xi_2 v_2(x_{best,j} - x_{i,j}) \tag{18}$$

where $k(\neq i)$ and j are randomly chosen from $\{1,...,M\}$ and $\{1,...,N\}$ respectively; x_{best} represents the historical optimal solution of the entire population; ξ_1 is the individual cognitive acceleration coefficient, ξ_2 is the social acceleration coefficient. v_1 and v_2 are random numbers selected from the uniform distribution $[0,1]$.

3. Case studies

Basic data. In this section, we validate the proposed learning-based ABC algorithm using the case of real natural gas pipelines. The schematic diagram of the pipeline system, which includes two parallel buried pipelines labeled as Line1 and Line2, is depicted in Fig. 5. The total length of the pipelines is 1837 km, inner diameter is 1067 mm, and the average ambient temperature is 278 K. Line1 and Line2 are spaced 30 meters apart. They share three gas sources labeled as GS1, GS2, and GS3, and their gas composition can be determined using gas chromatographs. Along Line1' route, there are 9 compressor stations {1-CS1,..,1-CS9} and 3 metering stations

Table 2
Pipeline parameters.

Pipeline	Station	Case 1	Case 2	Case 3
Line 1	GS1	6607	7926	7697
	GS2	0	1047	894
	GS3	0	0	0
	TIP01	0	0	659
	TIP02	0	0	0
	TIP03	0	1188	1053
	P^{s}_{first}	6.50	6.70	6.70
	P^{min}_{end}	6.50	8.00	8.00
Pipeline	Station	Case 4	Case 5	Case 6
Line 2	GS1	1856	2052	2006
	GS2	1696	0	0
	GS3	2376	2188	2305
	TIP01	583.7	0	0
	TIP02	0	544	726
	TIP03	1008	0	0
	P^{s}_{first}	6.25	6.40	6.40
	P^{min}_{end}	6.50	8.00	8.00

Table 3
Number of running compressors under the actual and calculated operation schemes.

Stations	Case 1		Case 2		Case 3	
	Actual	Calculated	Actual	Calculated	Actual	Calculated
1-CS1	2	2	2	2	2	2
1-CS2	0	0	2	2	2	2
1-CS3	2	0	2	2	2	2
1-CS4	2	2	2	2	2	2
1-CS5	0	0	2	2	0	0
1-CS6	2	2	2	2	2	2
1-CS7	0	2	2	2	2	0
1-CS8	2	0	2	2	2	2
1-CS9	0	0	0	0	0	0
Total	10	8	16	16	14	12
Stations	Case 4		Case 5		Case 6	
	Actual	Calculated	Actual	Calculated	Actual	Calculated
2-CS1	1	1	1	1	1	1
2-CS2	1	0	0	0	0	0
2-CS3	0	0	0	0	0	0
2-CS4	1	1	1	1	1	1
2-CS5	0	0	0	0	0	0
2-CS6	1	1	1	1	1	1
2-CS7	1	0	0	0	0	0
2-CS8	2	1	1	1	1	1
2-CS9	0	0	0	0	0	0
2-CS10	1	0	1	1	1	1
2-CS11	0	0	0	0	0	0
Total	8	4	5	5	5	5

{1-MS1, 1-CS1, 1-CS9}. Along Line2' route, there are 11 compressor stations {2-CS1,..,2-CS11} and 3 metering stations {2-CS1, 2-MS1, 2-CS11}. Each compressor station is equipped with multiple compressor units to elevate gas pressure, while each metering station is equipped with a flow sensor to measure gas flow. The operating envelope of gas compressors can be estimated through the least squares analysis using initial data points provided by vendors. Furthermore, TIP01 and TIP03 function as intermediate gas downloading points, and TIP02 operates as a seasonal gas uploading point. To validate the feasibility of our proposed method, we formulate six distinct operational cases. Three of these cases involve varying gas flow from gas sources, upload, and download points in Line 1, while the remaining three feature varying gas flow from gas sources, upload, and download points in Line 2. And the actual operation schemes in Case1 and Case4 were implemented before the company purchased a optimization system, while those in Case2, Case3, Case5, and Case6 were implemented after the company acquired a optimization system. Crucial parameters for these cases, including volumetric flow for gas sources, upload and download points (Nm^3/d), the first station's suction pressure (P^{s}_{first}, MP), and the terminal station's minimum required pressure (P^{min}_{end}, MP), are presented in Table 2.

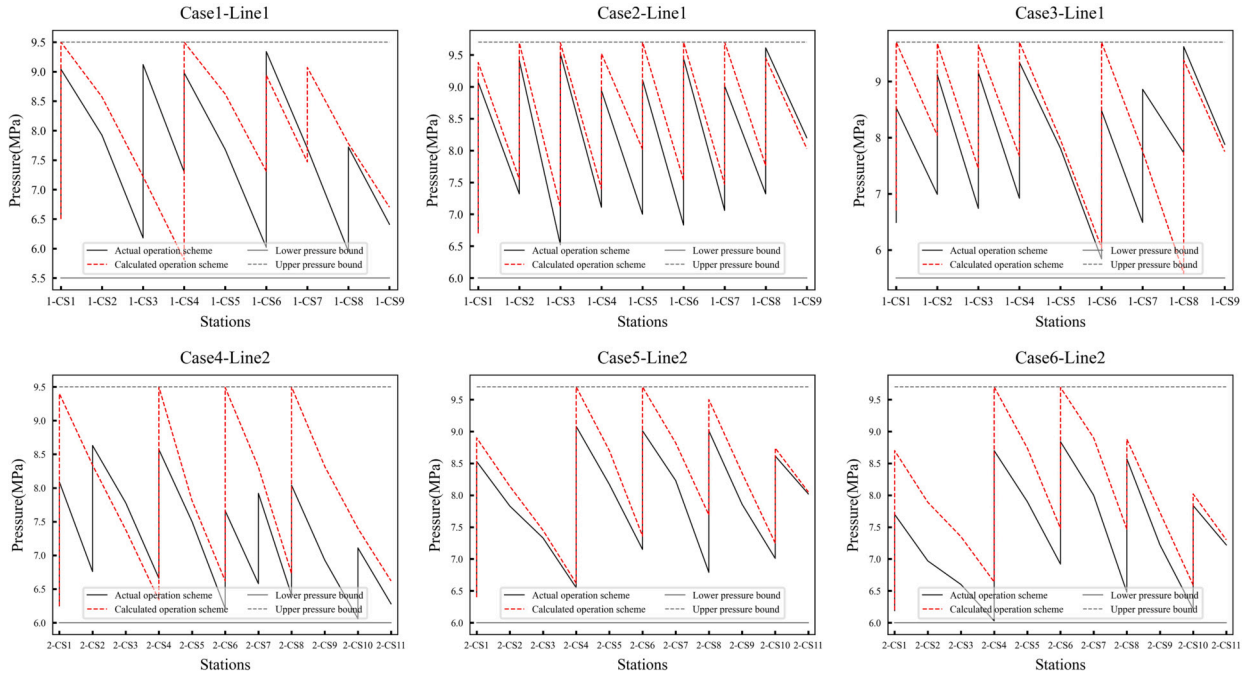

Fig. 6. Pressure distributions in the whole pipeline system under the actual and optimal operation schemes.

Operation schemes. The objective of this optimization problem is to minimize compressors' gas consumption by adjusting the discharge pressure and the number of operating compressors, while ensuring the normal transport of gas and meeting various constraints. According to the methodology proposed in Sections 2.3, the operation schemes of gas compressors can be intelligently determined. And we can analyze the characteristics of the operation scheme with the lowest gas consumption to gain some management insights. Fig. 6 illustrates the suction and discharge pressure distribution lines for both Line1 and Line2 under the actual and calculated operation schemes, all of which display serrated patterns. At each station point, if gas flows with compression operation, the vertical upward trend indicates an immediate pressure increase from the suction pressure to the discharge pressure; if gas flows without compression operation, a single point represents that the discharge and suction pressures are equal. Between two neighboring station points, the sloped downward curve indicates a gradual pressure loss along the pipeline due to factors like gas friction with the inner pipe wall. The two parallel gray lines represent the upper and lower bounds of pipeline pressure. The values may vary for different pipelines, and even for the same pipeline at different times. Table 3 shows the corresponding number of running compressors in each station.

When comparing the pressure distributions between the actual and calculated schemes in Fig. 6, it becomes evident that: (*i*) The pressure distributions of the actual and calculated optimization schemes reveal significant differences in Case 1 and Case 4. However, in Cases 2, 3, 5, and 6, the pressure distributions exhibit similar trends despite some variations, likely attributable to minor operational differences at several compressor stations. Notably, the actual optimization schemes in Case 1 and Case 4 were implemented before the company acquired an optimization system, while those in Cases 2, 3, 5, and 6 were implemented afterward. This indicates that the operation schemes generated by our proposed algorithm closely align with real-world applications after optimization, further demonstrating the effectiveness of our proposed approach. (*ii*) Except for the terminal compressor station, the discharge pressures of other stations in the calculated operation schemes are higher than those in the actual schemes, approaching the upper pressure bound set by design. This phenomenon occurs because, according to the mechanisms outlined in Appendices A-1 and A-2, for a given mass of gas in a pipeline segment, higher pressure values lead to a higher gas compressibility factor and a reduced gas flow velocity. Therefore, the contact area between the gas and the pipe wall decreases, leading to a reduction in pressure loss due to friction. And the gas consumed by the compressors to compensate for this pressure loss will also decrease. (*iii*) The suction pressure at the terminal station typically approaches the minimum contractual pressure requirement set by downstream companies or customers, as indicated by the minimum required pressure (P_{end}^{min}) in Table 2. As a result, this causes gas to exit the pipeline at a lower pressure. This situation occurs because excessive suction pressure can result in energy wastage due to throttling in the valves. In Table 3, it is evident that the number of active compressors in Line 1 surpasses that in Line 2, which is attributed to the higher gas flow in Line 1 compared to Line 2. Overall, the majority of compressor stations along Line 1 have running compressors, whereas approximately half of the compressor stations along Line 2 are in operation. Integrating Table 3 and Fig. 6 reveals a clear trend: "In both Case 1 and Case 4, the actual operational schemes exhibit more active stations and running compressors compared to their corresponding calculated operational schemes. Consequently, the discharge pressure in the calculated control schemes is higher, aiming to offset the differences in the number of stations and compressors. In Case 2, Case 3, Case 5, and Case 6, the actual control schemes exhibit an roughly equal number of active stations and the same total count of running compressors. Thus, their discharge pressures are nearly identical.

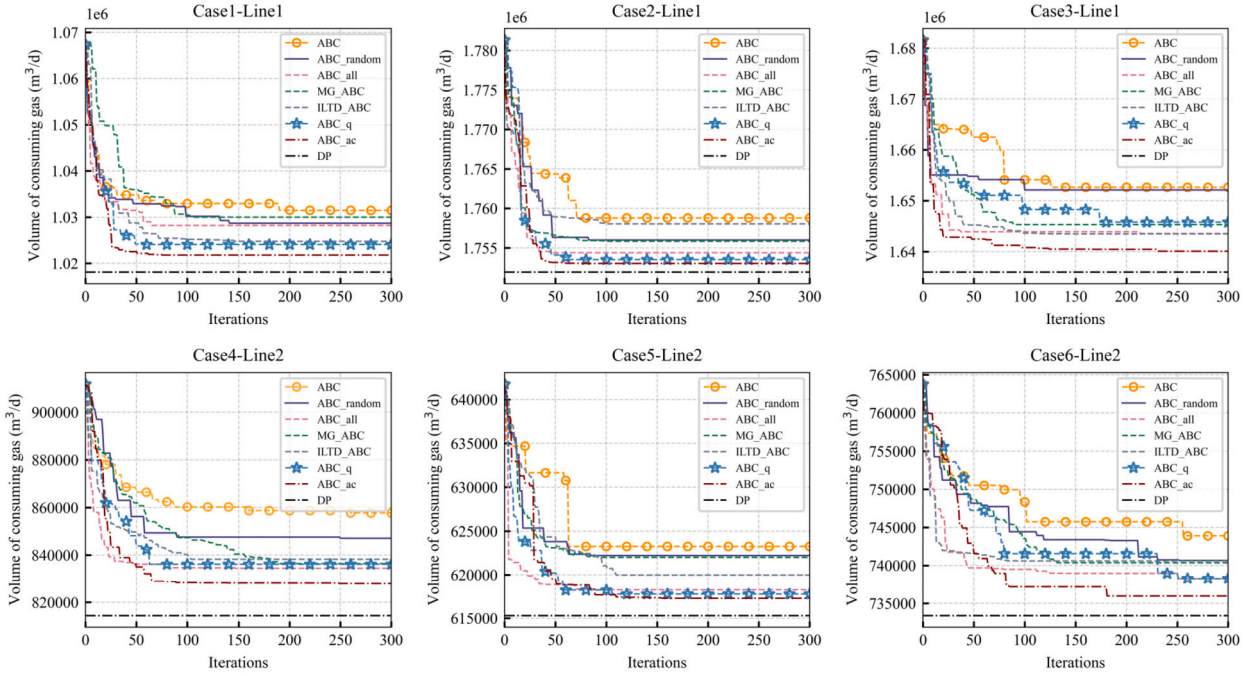

Fig. 7. The convergence performance of compared ABC algorithms.

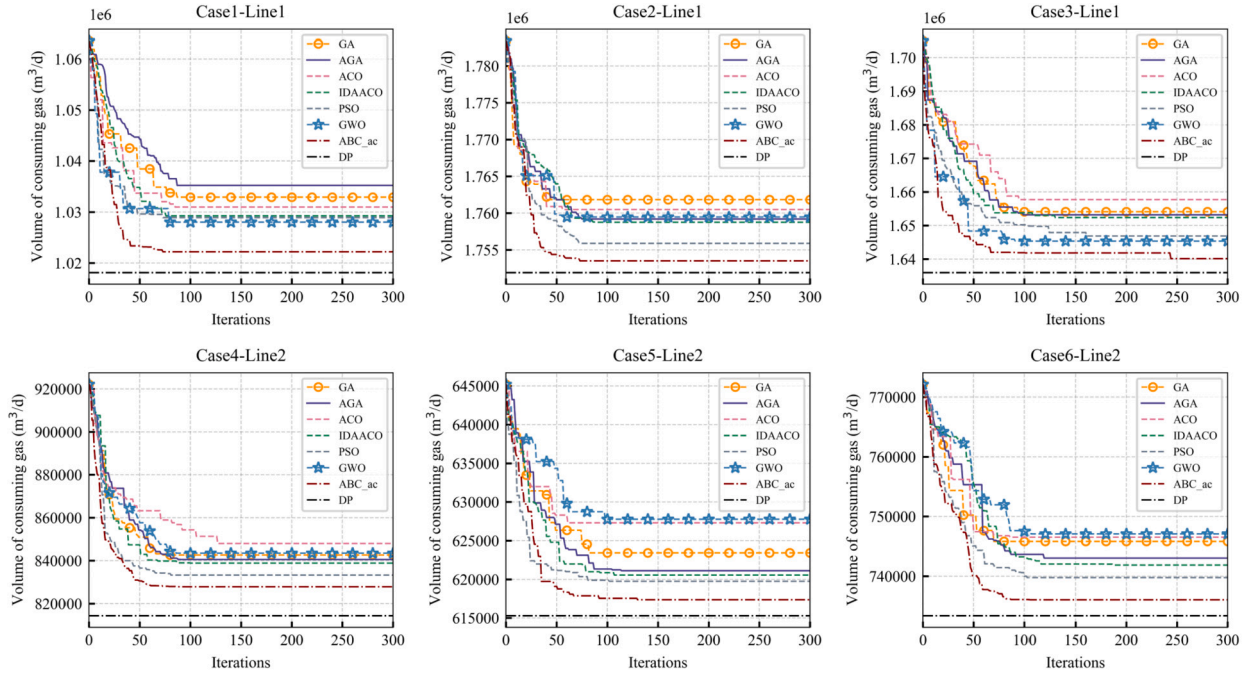

Fig. 8. The convergence performance of other compared algorithms.

Overall, the number of running compressors in the calculated operation schemes is lower than in the actual operation schemes. This is beneficial from the perspective of reducing compressor usage frequency and minimizing compressor wear. To sum up, three rules can be extracted for energy saving in gas pipelines: First, increase the discharge pressure upstream as much as possible; Second, maintain the suction pressure of the last station as close to the contract pressure as possible; Third, minimize the use of compressors.

Comparison experiments. The parameter d_{up} regulates control the frequency of perturbation in the proposed ABC. To evaluate the effectiveness of the learning-based ABC algorithm, ABC_ac, we compare it with five ABC variants and the standard ABC method on the six cases mentioned earlier. The five variants include: (1) ABC_random: this approach differs from the standard ABC by randomly

Table 4
Parameter settings of the proposed and compared algorithms.

Algorithm	Parameter setting
ABC [26]	$limit = 30$;
ABC_random	$limit = 30$;
ABC_all	$limit = 30$;
MG_ABC [35]	$limit = 30$; $q = 0.1$; $MR = 0.5$; $p = 0.1$;
ILTD_ABC [38]	$limit = 30$;
ABC_q [27]	$limit = 30$; $\alpha = 0.6$; $\beta = 0.4$; $\epsilon = 0.3$;
ABC_ac	$limit = 30$; $\xi_1 = 1.5$; $\xi_2 = 1.5$;
TS [39]	$T = 30$;
SA [40]	$\alpha = 0.99$; $\rho = 0.2$;
GA [41]	$p_c = 0.9$; $p_u = 0.1$;
AGA [47]	$\omega = 20$;
ACO [42]	$\alpha = 1$; $\beta = 5$; $\rho = 0.5$;
IDAACO [45]	$\tau_0 = 1$; $\alpha = 1$; $\beta = 4$; $\rho_0 = 0.5$; $Q_0 = 1$; $\gamma_0 = 1$; $w_1 = 0.4$; $w_2 = 1$; $w_3 = 0.2$;
PSO [43]	$\omega = 0.5$; $c_1 = 1.5$; $c_2 = 1.5$;
GWO [44]	$\vec{a} = [2, 0]$.

Table 5
Comparison with ABC and other ABC variants.

Algorithms	Case 1		Case 2		Case 3		Case 4		Case 5		Case 6	
	Gas usage	Time	Gas usage	Time	Gas usage	Time	Gas usage	Time	Gas usage	Time	Gas usage	Time
ABC	1031469	9.16	1758786	12.66	1652649	13.06	857743	10.01	623244	11.19	743909	12.04
ABC_random	1028688	10.91	1755988	14.33	1652118	14.36	847094	11.12	622207	13.35	740682	13.06
ABC_all	1028200	10.21	1754375	13.94	1643563	14.10	834476	10.74	618248	12.54	738279	12.97
MG_ABC	1030000	12.24	1755834	16.43	1645319	15.75	836462	12.56	622000	14.47	740368	14.22
ILTD_ABC	1024771	11.53	1758032	15.17	1643484	16.27	838254	11.74	619976	15.03	740612	14.76
ABC_q	1024095	11.32	1753504	14.59	1645800	16.00	836179	12.09	617767	14.02	738279	13.98
ABC_ac	**1021794**	11.40	**1753018**	14.88	**1640084**	15.85	**827902**	12.23	**617277**	14.25	**735957**	14.37

selecting which dimensions to update in Eq. (8); (2) ABC_all: this method deviates from the standard ABC by updating all dimensions in Eq. (8); (3) ABC_q: this variant, based on the strategy from [27], uses Q-learning to determine the number of dimensions to change in Eq. (8) without specifying which ones; (4) ABC algorithm with multi-elite guidance (MG_ABC): this method guides the search in Eq. (8) using elite individuals randomly chosen from three elite groups [35], and (5) ABC algorithm with an intelligent learning strategy and a turbulent operator (ILTD_ABC): this method incorporates experiences from the entire population and the current global best solutions into Eq. (8) and uses a turbulent distribution operator to adjust their relative weights [38]. To adapt to the continuous state settings in this paper, we substitute Q-learning in ABC_q with deep Q-network (DQN). All these algorithms undergo 300 iterations and have 50 food sources, i.e., $X = \{x_1, ..., x_i, ..., x_{50}\}$, where $x_i = \{p^d_{i1}, p^d_{i2}, ..., p^d_{iN}\}$ and the number of compressor stations represents its dimension N. Thus, $N = 8$ in Case 1-3, and $N = 11$ in Case 4-6. The control parameters of those algorithms are also presented in Table 4. Fig. 7 shows the performance and convergence of the compared ABC strategies. For actor-critic method, we configure the discount factor as 0.99, the learning rates for actor and critic as 0.001. In this comparison, the search effectiveness of ABC_ac is demonstrated through the solution of the operation optimization problems with different pipelines and flow rates. Simultaneously, it is observed that the variations among different algorithms are more pronounced in Line 2 than in Line 1. This phenomenon could be attributed to the following factors: (i) in Line 1, the large gas flow necessitates almost all compressor stations to be operational, resulting in a relatively smaller search space for the algorithm; (ii) whereas in Line 2, the lower gas flow allows for more diverse combinations of compressor station operations, thereby expanding the algorithm's solution space. Furthermore, when comparing Case 1-3 and Case 4-6 horizontally, a clear visual trend emerges: as the gas flow rates increase, there is a corresponding rise in the consumed gas volume.

We also conduct a series of comparative analyses of our approach against other widely used meta-heuristic algorithms, including tabu search (TS) [39], simulated annealing (SA) [40], genetic algorithm (GA) [41], ant colony optimization (ACO) [42], particle swarm optimization (PSO) [43], and grey wolf optimization (GWO) [44]. Additionally, we compare our method with algorithms designed for problems similar to gas consumption optimization, such as improved dynamic adaptive ACO (IDAACO), which has been applied to pipe routing design problems [45], and adaptive genetic algorithm (AGA), which is used for optimizing oil pipeline systems [46,47]. IDAACO enhances standard ACO by incorporating four key mechanisms: a heuristic strategy with directional information, an adaptive pseudorandom transfer strategy, and improved local and global pheromone updates. Similarly, AGA improves GA with adaptive local search, crossover, and mutation operators. Similar to all ABC algorithms, these methods also undergo 300 iterations and have a population size of 50 individuals. The control parameters of those algorithms are also presented in Table 4. Fig. 8 shows the performance and convergence of these meta-heuristic algorithms. Table 5 and 6 show that the proposed algorithm achieves the best performance in terms of gas consumption among these methods. However, this higher solution quality comes with the trade-off of longer computational time compared to some of the other methods. The extended computation time can be attributed to two main

Table 6
Comparison with frequently used meta-heuristics.

Algorithms	Case 1		Case 2		Case 3		Case 4		Case 5		Case 6	
	Gas usage	Time	Gas usage	Time	Gas usage	Time	Gas usage	Time	Gas usage	Time	Gas usage	Time
TS	1037652	6.52	1777511	7.21	1682541	7.54	850125	6.73	631133	7.24	753743	7.63
SA	1036660	7.11	1767893	7.98	1663929	8.21	849763	6.99	628768	7.64	752046	8.03
GA	1032918	9.58	1761843	13.02	1654110	12.15	842653	10.98	623419	12.15	745821	13.87
AGA	1035220	10.25	1759201	13.34	1653177	13.92	840548	12.09	621117	13.27	743029	14.24
ACO	1030968	10.11	1760487	12.53	1657724	12.97	848011	9.99	627304	12.26	746531	12.87
IDAACO	1029244	12.14	1758762	15.37	1652375	15.71	838882	13.29	620554	14.66	741861	15.43
PSO	1028900	8.12	1755889	9.78	1646870	10.33	833298	8.94	619754	10.98	739761	11.65
GWO	1028010	8.05	1759481	9.61	1645351	10.53	843525	9.15	627777	11.03	747082	11.22
ABC_ac	**1022190**	10.98	**1753520**	14.25	**1640136**	15.09	**827878**	12.00	**617360**	14.09	**736041**	14.98

factors: First, the ABC_ac algorithm tackles optimization problems by simulating the foraging behavior of a bee colony. Each iteration involves the operations of three types of bees: employed bees, onlooker bees, and scout bees. Since each bee type has specific tasks and procedures, this results in longer computation times per iteration. In contrast, algorithms like PSO, TS, and SA conduct only a single search in each iteration, which leads to shorter computation times. Second, the ABC_ac algorithm requires the collection of state information during each iteration and relies on a neural network to select actions. This additional process also contributes to the increased computational time. In general, this optimization problem serves oil and gas companies' daily gas operation optimization or long-term scheduling plans. Therefore, relatively speaking, although the proposed algorithm has higher time costs, this difference is not significant. Additionally, when the pipeline flow is high, the discharge pressure at almost all stations approaches the upper pressure limit to meet interstation pressure constraints, making the differences between algorithms less pronounced than in cases with lower flow rates. However, considering the price of natural gas and the long operational periods, even small gas savings can significantly reduce costs. Moreover, we found that the majority of the runtime is spent on solving the gas compressibility factor and friction coefficient using the Newton method and implicit iterative method. If these two factors can be precomputed and tabulated, the runtime could be reduced to approximately 4 minutes. In addition, we also solve these six cases using dynamic programming. In fact, the operation schemes of compressors between stations influence each other, which means that gas consumption optimization does not satisfy the non-aftereffect property required for dynamic programming. As a result, after discretizing the discharge pressure, it becomes necessary to compute all feasible operation schemes, leading the dynamic programming algorithm to take over 30 hours to execute. Therefore, dynamic programming id unsuitable for oil and gas companies' to make daily operation schemes. To sum up, considering both computation time and optimization performance, our proposed algorithm offers clear advantages.

4. Discussion and conclusions

In this research, we propose a learning-based artificial bee colony algorithm to conduct the optimal control of gas compressors in gas pipeline networks. The crucial factors affecting gas consumption (discharge pressure and the number of operating compressors) are used as the control variables. To avoid the discretization of continuous control variables in traditional algorithm and mitigate the risk of falling into local minima, we integrate the reinforcement learning with function approximation into the basic artificial bee colony algorithm. On one hand, reinforcement learning achieves the intelligent adjustment of the frequency of perturbation during the employed bee phase based on immediate rewards obtained from solution update outcomes. On the other hand, function approximation method can address challenges posed by the continuous state associated with gas consumption. The concept of multi-label classification is further used in the function approximation method to support the simultaneous optimal control of compressors in all stations. The evaluation model demonstrates the superiority of the optimal control policy obtained by our learning-based algorithm. Upon comparing the actual and optimal operation schemes, it becomes apparent that maintaining the pipeline system at a higher pressure level by setting a higher discharge pressure and ensuring perfect alignment between the downstream pressure and the minimum pressure requirement specified by the customers can effectively lead to a reduction in fuel consumption.

Delving into gas transmission optimization under transient states, which captures the dynamic fluctuations of the system over time, presents a promising avenue for future research. Furthermore, the integration of more authentic operational data into algorithms could offer valuable insights for guiding forthcoming operational strategies.

CRediT authorship contribution statement

Min Liu: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Yundong Yuan:** Resources, Formal analysis, Data curation. **Aobo Xu:** Writing – original draft, Software. **Tianhu Deng:** Writing – review & editing, Validation. **Ling Jian:** Writing – review & editing, Validation, Resources, Project administration, Funding acquisition, Data curation.

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.

Acknowledgement

This work was supported by the National Key Research and Development Program of China under Grant Nos.2021YFA1000104 and 2021YFA1000100, Laboratory Project of Higher Education Institutions in Shandong Province-Energy System Intelligent Management and Policy Simulation Laboratory at China University of Petroleum, and Youth Innovation Team of Higher Education Institutions in Shandong Province-Data Intelligence Innovation Team at China University of Petroleum.

Appendix A-1. BWRS equation of state

The state of natural gas is calculated by the BWRS equation as follows:

$$\rho_0 RT + \left(B_0 RT - A_0 - \frac{C_0}{T^2} + \frac{D_0}{T^3} - \frac{E_0}{T^4}\right)\rho_0^2 + \left(bRT - a - \frac{d}{T}\right)\rho_0^3 + \alpha\left(a + \frac{d}{T}\right)\rho_0^6 + \frac{c\rho_0^3}{T^2}\left(1 + \gamma\rho_0^2\right)e^{(-\gamma\rho_0^2)} - 0.001p = 0 \tag{A1}$$

where $A_0, B_0, C_0, D_0, E_0, a, b, c, d, \alpha, \gamma$ are the parameters related to the physical properties of gas composition, including the acentric factor, critical density, and critical temperature. And the gas composition in the pipeline may vary due to the gas injection at the sources and the gas mixing at the pipeline junctions. Given that Eq. (A1) is a non-linear implicit equation, determining ρ_0 involves employing the fixed-point method or a variant of the Newton-Raphson method. After calculating ρ_0, we can obtain the compressible factor z using [23]:

$$z = \frac{p}{1000R\rho_0 T}. \tag{A2}$$

Obviously, compressible factor z has a complex dependence on temperature T, pressure p, and gas composition. Note that the result ρ_0 from BWRS is the gas molar density, which can be converted to gas density ρ_g, using $\rho_g = \rho_0 \cdot m_g$, where m_g represents the gas molar mass.

Appendix A-2. Friction factor

The friction factor, λ, is a dimensionless number defined as four times the ratio of the local shear stress to the local flow kinetic energy density, i.e., $\lambda = 4\tau/(\rho\frac{u^2}{2})$ [21]. Depending on the Reynolds number R_{en}, gas exhibits either laminar or turbulent flow, requiring different empirical equations for calculating the friction factor [48]. Due to the high flow rate of gas in pipeline transmission, the flow within the pipes is considered fully turbulent, i.e., the Reynolds number values are much greater than 4000. For turbulent regime, one of the widely accepted formulations is the Colebrook-White (C-W) [49]:

$$\frac{1}{\sqrt{\lambda}} = -2lg\left(\frac{K_e}{3.71d_n} + \frac{2.51}{R_e\sqrt{\lambda}}\right). \tag{A3}$$

Similar to BWRS, Eq. (A3) is also a non-linear implicit equation involving the friction factor λ. And it requires the use of the fixed-point method or a variant of the Newton method for iterative solutions. The surface roughness K_e lacks a widely accepted definition and is difficult to measure. Accurate values for this parameter often require experimental measurement of pressure loss, followed by roughness calculation. Moreover, the Reynolds number is the ratio between inertial and viscous forces, defined by [50]:

$$R_e = \frac{\rho_g w d_n}{\vartheta_n} = \frac{4\Delta\rho_a q_n}{\pi d_n \vartheta} \tag{A4}$$

where ϑ is the dynamic viscosity of the gas and ρ_a is air density. Viscosity models are commonly described using an exponential relationship with temperature and often complex (see [22] for more details).

Appendix A-3. Average temperature

Based on the environment temperature T_{env} and the inlet temperature T_n^i, the outlet temperature T_n^o and average temperature T_n^{av} of the steady gas flow can be calculated by [23]:

$$T_n^o = T_{env} + (T_n^i - T_{env})e^{-al_n} - D\frac{p_n^o - p_n^i}{al_n}\left(1 - e^{-al_n}\right) \tag{A5}$$

$$T_n^{av} = T_{env} + (T_n^i - T_{env})\frac{1 - e^{-al_n}}{al_n} - D\frac{p_n^o - p_n^i}{al_n}\left(1 - \frac{1 - e^{-al_n}}{al_n}\right) \tag{A6}$$

where a is an intermediate parameter and $a = \frac{\pi d_n K_{ht}}{q_n^m c_p}$.

Data availability

Data will be made available on request.

References

[1] S. Zhang, S. Liu, T. Deng, Z.J.M. Shen, Transient-state natural gas transmission in gunbarrel pipeline networks, INFORMS J. Comput. 32 (2020) 697–713, https://doi.org/10.1287/ijoc.2019.0904.

[2] T.W.K. Mak, P.V. Hentenryck, A. Zlotnik, R. Bent, Dynamic compressor optimization in natural gas pipeline systems, INFORMS J. Comput. 31 (2019) 40–65, https://doi.org/10.1287/ijoc.2018.0821.

[3] J. Han, Y. Xu, D. Liu, Y. Zhao, Z. Zhao, S. Zhou, T. Deng, M. Xue, J. Ye, Z.J.M. Shen, Operations research enables better planning of natural gas pipelines, INFORMS J. Appl. Anal. 49 (2019) 23–39, https://doi.org/10.1287/inte.2018.0974.

[4] R.Z. Ríos-Mercado, S. Kim, E.A. Boyd, Efficient operation of natural gas transmission systems: a network-based heuristic for cyclic structures, Comput. Oper. Res. 33 (2006) 2323–2351, https://doi.org/10.1016/j.cor.2005.02.003.

[5] C. Borraz-Sánchez, D. Haugland, Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization, Comput. Ind. Eng. 61 (2011) 364–372, https://doi.org/10.1016/j.cie.2010.07.012.

[6] D. Mahlke, A. Martin, S. Moritz, A simulated annealing algorithm for transient optimization in gas networks, Math. Methods Oper. Res. 66 (2007) 99–115, https://doi.org/10.1007/s00186-006-0142-9.

[7] D.A. Rodríguez, P.P. Oteiza, N.B. Brignole, Simulated annealing optimization for hydrocarbon pipeline networks, Ind. Eng. Chem. Res. 52 (2013) 8579–8588, https://doi.org/10.1021/ie400022g.

[8] C. Borraz-Sánchez, R.Z. Ríos-Mercado, Improving the operation of pipeline systems on cyclic structures by tabu search, Comput. Chem. Eng. 33 (2009) 58–64, https://doi.org/10.1016/j.compchemeng.2008.07.009.

[9] A. Chebouba, F. Yalaoui, A. Smati, L. Amodeo, K. Younsi, A. Tairi, Optimization of natural gas pipeline transportation using ant colony optimization, Comput. Oper. Res. 36 (2009) 1916–1923, https://doi.org/10.1016/j.cor.2008.06.005.

[10] A.K. Arya, S. Honwad, Multiobjective optimization of a gas pipeline network: an ant colony approach, J. Pet. Explor. Prod. Technol. 8 (2017) 1389–1400, https://doi.org/10.1007/s13202-017-0410-7.

[11] A.K. Arya, Optimal operation of a multi-distribution natural gas pipeline grid: an ant colony approach, J. Pet. Explor. Prod. Technol. 11 (2021) 3859–3878, https://doi.org/10.1007/s13202-021-01266-3.

[12] S. Sanaye, J. Mahmoudimehr, Minimization of fuel consumption in cyclic and non-cyclic natural gas transmission networks: assessment of genetic algorithm optimization method as an alternative to non-sequential dynamic programing, J. Taiwan Inst. Chem. Eng. 43 (2012) 904–917, https://doi.org/10.1016/j.jtice.2012.04.010.

[13] A. Demissie, W. Zhu, C.T. Belachew, A multi-objective optimization model for gas pipeline operations, Comput. Chem. Eng. 100 (2017) 94–103, https://doi.org/10.1016/j.compchemeng.2017.02.017.

[14] R. Qi, J.q. Li, J. Wang, H. Jin, Y.y. Han, QMOEA: a Q-learning-based multiobjective evolutionary algorithm for solving time-dependent green vehicle routing problems with time windows, Inf. Sci. 608 (2022) 178–201, https://doi.org/10.1016/j.ins.2022.06.056.

[15] M. Karimi-Mamaghan, M. Mohammadi, P. Meyer, A.M. Karimi-Mamaghan, E.G. Talbi, Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: a state-of-the-art, Eur. J. Oper. Res. 296 (2022) 393–422, https://doi.org/10.1016/j.ejor.2021.04.032.

[16] Z. Wei, W. Gao, M. Gong, G.G. Yen, A bi-objective evolutionary algorithm for multimodal multi-objective optimization, IEEE Trans. Evol. Comput. 1–1doi (2022), https://doi.org/10.1109/TEVC.2022.3217258.

[17] Y. Zhao, X. Xu, M. Qadrdan, J. Wu, Optimal operation of compressor units in gas networks to provide flexibility to power systems, Appl. Energy 290 (2021) 116740, https://doi.org/10.1016/j.apenergy.2021.116740.

[18] C. Borraz-Sánchez, R.Z. Ríos-Mercado, A hybrid meta-heuristic approach for natural gas pipeline network optimization, in: M.J. Blesa, C. Blum, A. Roli, M. Sampels (Eds.), Hybrid Metaheuristics, Berlin, Heidelberg, 2005, pp. 54–65.

[19] A.H. Alinia Kashani, R. Molaei, Techno-economical and environmental optimization of natural gas network operation, Chem. Eng. Res. Des. 92 (2014) 2106–2122, https://doi.org/10.1016/j.cherd.2014.02.006.

[20] T. Deng, Y. Liang, S. Zhang, J. Ren, S. Zheng, A dynamic programming approach to power consumption minimization in gunbarrel natural gas networks with nonidentical compressor units, INFORMS J. Comput. 31 (2019) 593–611, https://doi.org/10.1287/ijoc.2018.0833.

[21] A. López-Benito, F.J. Elorza Tenreiro, L.C. Gutiérrez-Pérez, Steady-state non-isothermal flow model for natural gas transmission in pipes, Appl. Math. Model. 40 (2016) 10020–10037, https://doi.org/10.1016/j.apm.2016.06.057.

[22] Z. Marfatia, X. Li, On steady state modelling for optimization of natural gas pipeline networks, Chem. Eng. Sci. 255 (2022) 117636, https://doi.org/10.1016/j.ces.2022.117636.

[23] Q. Chen, L. Zuo, C. Wu, Y. Li, K. Hua, M. Mehrtash, Y. Cao, Optimization of compressor standby schemes for gas transmission pipeline systems based on gas delivery reliability, Reliab. Eng. Syst. Saf. 221 (2022) 108351, https://doi.org/10.1016/j.ress.2022.108351.

[24] X. Zhang, C. Wu, L. Zuo, Minimizing fuel consumption of a gas pipeline in transient states by dynamic programming, J. Nat. Gas Sci. Eng. 28 (2016) 193–203, https://doi.org/10.1016/j.jngse.2015.11.035.

[25] J.A. Concha-Carrasco, M.A. Vega-Rodríguez, C.J. Pérez, A multi-objective artificial bee colony approach for profit-aware recommender systems, Inf. Sci. 625 (2023) 476–488, https://doi.org/10.1016/j.ins.2023.01.050.

[26] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (2007) 459–471, https://doi.org/10.1007/s10898-007-9149-x.

[27] Y. Cui, W. Hu, A. Rahmani, A reinforcement learning based artificial bee colony algorithm with application in robot path planning, Expert Syst. Appl. 203 (2022) 117389, https://doi.org/10.1016/j.eswa.2022.117389.

[28] S. Peng, Q. Feng, Data-driven optimal control of wind turbines using reinforcement learning with function approximation, Comput. Ind. Eng. 176 (2023) 108934, https://doi.org/10.1016/j.cie.2022.108934.

[29] W. Gao, Y. Li, Solving a new test set of nonlinear equation systems by evolutionary algorithm, IEEE Trans. Cybern. 53 (2023) 406–415, https://doi.org/10.1109/TCYB.2021.3108563.

[30] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 1928–1937.

[31] A. Xu, L. Jian, A deep news headline generation model with REINFORCE filter, in: 2023 International Joint Conference on Neural Networks (IJCNN), 2023, pp. 1–7.

[32] H. Lu, X. Zhang, S. Yang, A learning-based iterative method for solving vehicle routing problems, in: International Conference on Learning Representations, 2019.

[33] T. Liu, H. Wang, Y. Wang, X. Wang, L. Su, J. Gao, SimFair: a unified framework for fairness-aware multi-label classification, in: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, 2023, pp. 14338–14346.

[34] B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inf. Sci. 192 (2012) 120–142, https://doi.org/10.1016/j.ins.2010.07.015.
[35] X. Zhou, J. Lu, J. Huang, M. Zhong, M. Wang, Enhancing artificial bee colony algorithm with multi-elite guidance, Inf. Sci. 543 (2021) 242–258, https://doi.org/10.1016/j.ins.2020.07.037.
[36] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (2010) 3166–3173, https://doi.org/10.1016/j.amc.2010.08.049.
[37] Y. Xue, J. Jiang, B. Zhao, T. Ma, A self-adaptive artificial bee colony algorithm based on global best for global optimization, Soft Comput. 22 (2018) 2935–2952, https://doi.org/10.1007/s00500-017-2547-1.
[38] H. Gao, Y. Shi, C.M. Pun, S. Kwong, An improved artificial bee colony algorithm with its application, IEEE Trans. Ind. Inform. 15 (2019) 1853–1865, https://doi.org/10.1109/TII.2018.2857198.
[39] F. Glover, Tabu search—part I, ORSA J. Comput. (1989), https://doi.org/10.1287/ijoc.1.3.190.
[40] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680, https://doi.org/10.1126/science.220.4598.671.
[41] T. Rooker, Genetic Algorithms in Search, Optimization, and Machine Learning, AI Mag., vol. 12, 1991, pp. 102–103.
[42] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 26 (1996) 29–41, https://doi.org/10.1109/3477.484436.
[43] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[44] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, https://doi.org/10.1016/j.advengsoft.2013.12.007.
[45] C. Liu, L. Wu, X. Huang, W. Xiao, Improved dynamic adaptive ant colony optimization algorithm to solve pipe routing design, Knowl.-Based Syst. 237 (2022) 107846, https://doi.org/10.1016/j.knosys.2021.107846.
[46] Q. Yuan, Y. Gao, Y. Luo, Y. Chen, B. Wang, J. Wei, B. Yu, Study on the optimal operation scheme of a heated oil pipeline system under complex industrial conditions, Energy 272 (2023) 127139, https://doi.org/10.1016/j.energy.2023.127139.
[47] J. Li, R. Liu, R. Wang, Handling dynamic capacitated vehicle routing problems based on adaptive genetic algorithm with elastic strategy, Swarm Evol. Comput. 86 (2024) 101529, https://doi.org/10.1016/j.swevo.2024.101529.
[48] K. Liu, L.T. Biegler, B. Zhang, Q. Chen, Dynamic optimization of natural gas pipeline networks with demand and composition uncertainty, Chem. Eng. Sci. 215 (2020) 115449, https://doi.org/10.1016/j.ces.2019.115449.
[49] C.F. Colebrook, Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws, J. Inst. Civ. Eng. 11 (1939) 133–156, https://doi.org/10.1680/ijoti.1939.13150.
[50] Y. Ruan, Q. Liu, W. Zhou, B. Batty, W. Gao, J. Ren, T. Watanabe, A procedure to design the mainline system in natural gas networks, Appl. Math. Model. 33 (2009) 3040–3051, https://doi.org/10.1016/j.apm.2008.10.008.