Neural Comput & Applic
DOI 10.1007/s00521-017-3232-0

ORIGINAL ARTICLE

Dynamic optimization based on state transition algorithm for copper removal process

Miao Huang[1] · Xiaojun Zhou[1] · Tingwen Huang[2] · Chunhua Yang[1] · Weihua Gui[1]

Received: 11 May 2017 / Accepted: 4 October 2017

Abstract The copper removal process (CRP) is an indispensable step for the purification of zinc sulfate solution by adding powdered zinc to a series of reactors in zinc hydrometallurgy. The selection of optimal amount of zinc powder is a complicated task because of the complex reaction mechanism, resulting in the fluctuation of copper ion concentration and the waste of zinc powder in the actual process. In this paper, we formulate a dynamic optimization problem (DOP) for the control of the zinc powder in CRP, aiming at reducing production costs and improving product quality simultaneously. A novel dynamic optimization method based on the state transition algorithm (STA) is investigated for solving this problem, and to improve the performance of STA, an adaptive strategy is adopted by its transformation operators. Simulation results from some classical DOPs show that the proposed method can optimize effectively and efficiently. The proposed approach is successfully applied to solve the DOP arising in CRP and the simulation results show that zinc powder consumption is considerably reduced under the assumption of an acceptable copper ion concentration.

Keywords Copper removal process · Dynamic optimization · State transition algorithm · Global optimization

✉ Xiaojun Zhou
michael.x.zhou@csu.edu.cn

[1] School of Information Science and Engineering, Central South University, Changsha 410083, China

[2] Texas A&M University at Qatar, Doha 23874, Qatar

1 Introduction

Zinc is an important nonferrous metal, widely used in electrical engineering, chemical engineering, automotive battery and other industries. More than 80% of zinc all over the world are produced by zinc hydrometallurgy, which is composed by five steps: roasting, calcine leaching, solution purification, electrowinning and melting [1, 2]. In the solution purification process, it is important to remove the impurities to an acceptable level before electrowinning, like copper, cobalt, nickel, and cadmium, which will lower the electrolytic efficiency and reduce the quality of the zinc ingot. Copper is the major impurity in leaching solution of zinc hydrometallurgy. Because of copper's special negative oxidation potential, the copper removal process (CRP) is commonly carried out by using zinc additives in the first stage of solution purification procedure. An appropriate zinc powder addition needs to be chosen precisely to ensure an acceptable copper ion concentration in CRP. Although an excess of amount of zinc powder addition can make a sharp decrease of copper ion concentration, it may cause huge waste of zinc powder; meanwhile, inadequate amount of copper ions can not provide enough activator to accelerate precipitation of cobalt subsequently. In the real-world CRP, due to the complex reaction mechanism and the unavailability of on-line ion concentration measurement, manual operation hardly assures the precision, reliability and stability, leading to problems such as unqualified outlet copper ion concentration, excessive zinc powder consumption, and then resulting in low electrolytic efficiency and fluctuation of the quality of zinc products. Therefore, it is critical to propose an effective method to improve productivity, reduce production costs, and optimize the zinc powder addition to ensure an qualified copper ion concentration with the least zinc powder consumption.

Published online: 16 October 2017

The optimization problem arising in CRP is a typical dynamic optimization problem (DOP), whose process variables and process values keep changing with dynamic systems which are formulated by mass–energy differential equations. Generally, DOPs aim to optimize a predefined performance index for reducing production costs, improving product quality, meeting safety requirements and environmental regulations over a time interval [3–5]. The basic characteristic of DOP is the value determination of a function at infinite points, namely all points in the control horizon. This type of problems are really difficult to solve due to their highly nonlinear, multidimensional, multimodal nature, and the presence of constraints on both state and control variables [6].

Several typical methods has been proposed over the last two decades. DOPs can be solved either by analytical methods or numerical methods [7]. Early analytical solution strategy, known as variational approach, is proposed to solve classical DOPs under special conditions, and guarantees a global solution. However such approach requires rigorous continuous-differential properties for system functions and variables, making it not easy to implement. Alternatively, the numerical methods apply discretization to convert the original infinite-dimensional problem into a finite-dimensional nonlinear programming (NLP) problem directly. Considering the degree of discretization, numerical methods generally can be categorized as complete discretization (i.e., it discretizes both sate and control variables of DOPs) or partial discretization [i.e., it only discretizes control variables, which is referred as control vector parameterization (CVP)]. The dynamic optimization method based on complete discretization inherently generates a large-scale nonlinear programming problem, and therefore it requires extra large-scale NLP solvers which may increase the optimization complexity [8]. On the other hand, CVP method has been widely used due to their easy implementation, good computational efficiency [9, 10]. In this paper, CVP method will be introduced detailedly to solve the DOPs.

Recently, evolutionary algorithms such as genetic algorithms (GA) [11], particle swarm optimization (PSO) [12], differential evolution (DE) [13] and other heuristic algorithms [14] have been introduced to solve various optimization problems, found to be robust and more likely to locate global optimum as compared to traditional optimization methods based on gradient. A stochastic global optimization algorithm named state transition algorithm (STA) [15–22] has been applied successfully in various complex problems and exhibited excellent performance for solving high-dimensional and nonlinear optimization problems compared with GA and PSO. The basic STA takes advantages of the space structure of the objective function and searches optimal solutions utilizing its special state transformation operators, whose search space is mainly determined by their transformation factors. A large transformation factor may facilitate the global search, while a small transformation factor can promote the local search. The strategy with a fixed transformation factor adopted in the basic STA is not beneficial to the solution accuracy and only increases redundant calculations in the later stage of optimization. Therefore, in order to reduce computational complexity and improve solution accuracy , it is necessary to develop adaptive transformation factors to balance global exploration and local exploitation in STA.

In what follows, the novelty and the contributions of this paper are summarized: (1) In order to obtain a qualified ion concentration with the least zinc powder consumption, a dynamic optimization problem is formulated for the control of the zinc powder in CRP. (2) An improved state transition algorithm with adaptive parameters is proposed, and then a novel dynamic optimization method, CVP–STA, is proposed for solving the DOPs. Simulation results from some classical DOPs have demonstrated that the proposed method can optimize effectively and efficiently and becomes a promising alternative method for solving DOPs. (3) The proposed approach is successfully applied to solve the DOP arising in CRP.

The remainder of this paper is organized as follows. In Sect. 2, the typical industrial CRP is described and analyzed in detail, and then a mathematical dynamic model for CRP is constructed based on the interactions among CSTR models. In Sect. 3, a dynamic optimization problem is formulated for CRP, where various limitations and specifications are considered. In Sect. 4, a novel dynamic optimization approach based on the improved STA and CVP method is proposed. Section 5 demonstrates the effectiveness of proposed method by several typical DOPs, and then the proposed approach is successfully applied to solve the DOP arising in CRP. Finally, the conclusion is drawn in Sect. 6.

2 Process analysis

2.1 Description of the copper removal process

In zinc hydrometallurgy, some impurities like copper, nickel, cobalt, and cadmium often coexist in the $ZnSO_4$ leaching solution. These impurity metal ions interfere with the electrowinning process and lead to energy waste and downgrade in product quality. Generally, purification process can be divided into three steps to purify them gradually.

The copper removal process (CRP) is commonly carried out in the first stage of purification, not only because copper is the major impurity in leaching solution of zinc

hydrometallurgy, but also because it is more negative oxidation potential than the other impurities. A simplified schematic of CRP is shown in Fig. 1. In practice, the $ZnSO_4$ leaching solution needs to flow through two copper precipitation reactors consecutively, where zinc powder is delivered individually by weight belts. And then, the overflowed purified solution is sent to the thickener to separate clean solution from the precipitate. The supernatant flows to the following purification stage, while part of the precipitate is returned to the 1# reactor to accelerate precipitation reaction.

The copper ion is deposited through two ways in CRP [23]: The majority of copper ions continuously reacts with zinc and precipitate as metallic copper according to the chemical reaction (1):

$$CuSO_4 + Zn \rightarrow ZnSO_4 + Cu \downarrow \tag{1}$$

and part of the copper ions reacts with metallic copper and precipitate in the form of cuprous oxide in accordance with the reaction (2):

$$CuSO_4 + Cu + H_2O \rightarrow Cu_2O \downarrow + H_2SO_4 \tag{2}$$

Owning to the complex reaction mechanism and rigorous requirement of precise control, it is essential to build a dynamic model to develop a visual and quantitative analysis for CRP.

2.2 Dynamic model of the CRP

Considering the reactors are very large and the zinc solution in the reactors is stirred continuously, the copper precipitation reactors can be regarded as s CSTR system. Based on the material and mass balance principles, the dynamic CSTR model can described as follows [24]:

$$V\dot{C}_{Cu^{2+},1} = QC^0_{Cu^{2+},1} - (1+q)QC_{Cu^{2+},1} - Vr_{Cu^{2+},1} \tag{3}$$

$$\begin{aligned} V\dot{C}_{Cu^{2+},2} = {} & (1+q)QC^0_{Cu^{2+},2} \\ & - (1+q)QC_{Cu^{2+},2} - Vr_{Cu^{2+},2}. \end{aligned} \tag{4}$$

Here $C_{Cu^{2+},i}$ and $C^0_{Cu^{2+},i}$, $i = 1, 2$ are the outlet concentrations and feed concentrations of copper ions of the ith reactor, and $\dot{C}_{Cu^{2+},i}$ and $r_{Cu^{2+},i}$, $i = 1, 2$ are the change rate and reaction rate of copper ions concentration of the ith reactor. V is the solution volume in each reactor , Q and q is the inlet flow rate and returned underflow rate of the first reactor, respectively. Noting that, the outlet impure solution of the #1 reactor flows into the #2 reaction directly, namely $C^0_{Cu^{2+},2} = C_{Cu^{2+},1}$.

According to the kinetic modeling of the competitive–consecutive reaction system presented in [25], the reaction rate of copper ions concentration in CRP can be described as the following simplified equation:

$$r_{Cu^{2+},1} = (k_1 G_{Zn,1} + k_2) V^{-1} C_{Cu^{2+},1} \tag{5}$$

$$r_{Cu^{2+},2} = (k_1 G_{Zn,2} + k_3) V^{-1} C_{Cu^{2+},2}, \tag{6}$$

where $k_i, i = 1, 2, 3$ are system parameters, $G_{Zn,i}, i = 1, 2$ are the zinc powder addition rate of the ith reactor.

In the context of CRP, the choice of the zinc powder addition of two reactors is critical. Therefore, in the remainder of this paper, we denote the zinc powder addition rate of two reactors as the control variable, namely $\boldsymbol{u} = [G_{Zn,1}, G_{Zn,2}]$, and the outlet copper ions concentration as state variable, which is described as

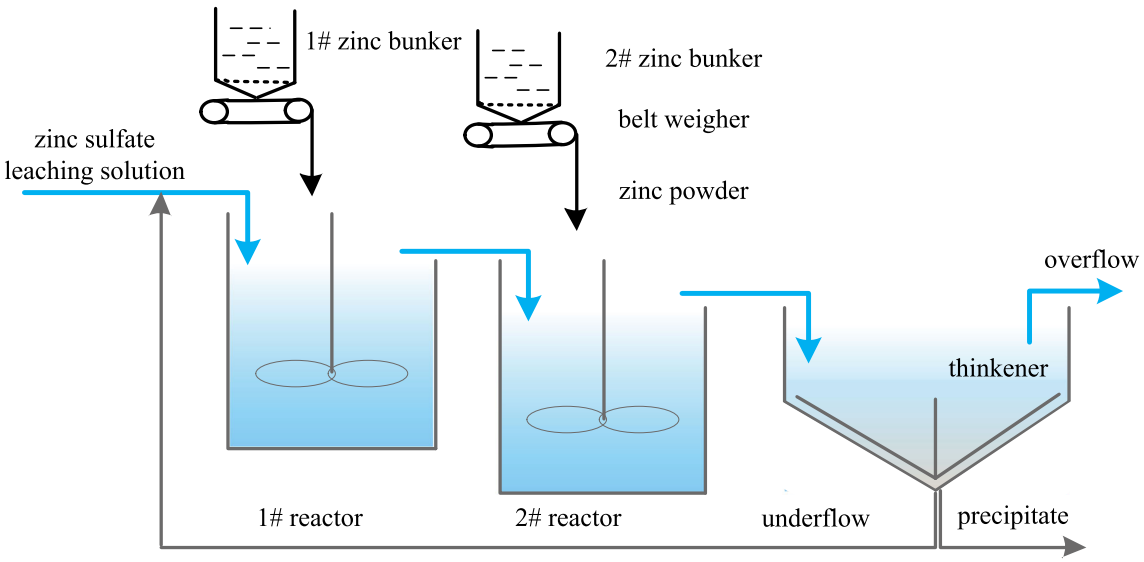

Fig. 1 A simplified schematic of CRP

$\boldsymbol{x} = [C_{Cu^{2+},1}, C_{Cu^{2+},2}]$. As mentioned above, the mathematical dynamic model of the CRP is rewritten as:

$$\begin{cases} \dot{x}_1 = V^{-1}[Qx_0 - (Q+q)x_1 - (k_1u_1 + k_2)x_1] \\ \dot{x}_2 = V^{-1}[(Q+q)x_1 - (Q+q)x_2 - (k_1u_2 + k_3)x_2] \end{cases} \tag{7}$$

3 Dynamic optimization problem of the CRP

The dynamic optimization of zinc powder addition is essential to guarantee a qualified copper ion concentration and to minimize the zinc powder consumption in the meanwhile. In this section, we introduce the general formulation of dynamic optimization problem and formulate a DOP for CRP.

3.1 General formulation of dynamic optimization problem

Most of the industrial optimization problems are dynamic in nature. One basic characteristic of such problem is that the system is modeled by differential–algebraic equation (DAE) system deriving from the mass and energy balance.

$$\begin{cases} \dot{\boldsymbol{x}} = f(\boldsymbol{u}(t), \boldsymbol{x}(t), t), \\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \end{cases} \tag{8}$$

where $\boldsymbol{x}(t) \in \Re^n$ is a state profile vector, with initial condition $\boldsymbol{x}_0$ and $\boldsymbol{u}(t) \in \Re^m$ is a control profile vector.

Generally, dynamic optimization in those applications aims to optimize a predefined performance index like profitability, product quality, and productivity subject over a time interval. Therefore, a typical dynamic optimization problem can be stated as follows:

$$\min_{\boldsymbol{u}(t)} J(\boldsymbol{u}(t)) = \Psi_0(\boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} L_0(\boldsymbol{u}(t), \boldsymbol{x}(t), t)\mathrm{d}t$$
$$\text{s.t.} \begin{cases} \dot{\boldsymbol{x}} = f(\boldsymbol{u}(t), \boldsymbol{x}(t), t) \\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\ \boldsymbol{h}(t_f, \boldsymbol{x}(t \mid \boldsymbol{u}), \boldsymbol{u}(t)) \le 0 \\ \boldsymbol{g}(t, \boldsymbol{x}(t \mid \boldsymbol{u}), \boldsymbol{u}(t)) \le 0 \\ \boldsymbol{u}^l \le \boldsymbol{u}(t) \le \boldsymbol{u}^u \\ t \in [t_0, t_f] \end{cases}, \tag{9}$$

where $J(\boldsymbol{u}(t))$ is the predefined performance index, namely objective function, basically composed of two parts: The evaluation function at the final time t_f, denoted by Ψ_0; and the cost function, L_0, weighting the state path travelled and the control sequence used. The control variables $\boldsymbol{u}(t)$ must satisfy their bound constraints $\boldsymbol{u}^l$ and $\boldsymbol{u}^u$, resulting from system's physical limitations. Besides, the boundary conditions and path constraints are denoted by **h** and **g**, respectively. In summary, such dynamic optimization problem is to find the input $\boldsymbol{u}^*(t), t \in [t_0, t_f]$ that drives the plant along the trajectory $\boldsymbol{x}^*(t), t \in [t_0, t_f]$ such that the aforementioned cost function is minimized where the final time t_f is fixed.

3.2 Formulation of the dynamic optimization problem in CRP

The dynamic optimization in CRP arises a typical dynamic optimization problem to determine every control point for a dynamic CSTR system in the control horizon to ensure a qualified outlet copper ion concentration without incurring a large amount of control effort.

3.2.1 Objective function

In order to minimize the total zinc powder additive amount, the dynamic optimization problem in CRP has the following objective:

$$\min_{\boldsymbol{u}(t)} J(\boldsymbol{u}(t)) = \int_{t_0}^{t_f} u_1(t) + u_2(t)\mathrm{d}t, \tag{10}$$

3.2.2 Constraints

By adding powdered zinc to a series of reactors, most copper ions will be precipitated in CRP, while 0.2–0.4 g/L copper ions need to remain to inhibit precipitation of cadmium and to accelerate precipitation of cobalt. To ensure a qualified outlet copper ion concentration, a terminal state constraint must be satisfied; on the other hand, because of the technical and equipment limitation in an actual CRP, the path constraint of control must be taken into consideration.

Therefore, the constraints of the dynamic optimization problems can formulated as following:

$$\begin{cases} C_{Cu^{2+},\min} \le x_2(t_f) \le C_{Cu^{2+},\max} \\ u^l_{\text{zinc}} \le \boldsymbol{u}(t) \le u^u_{\text{zinc}} \end{cases} \tag{11}$$

What is more, there exists implicit constraints imposed by dynamic model. During the whole process, the control variable and process value keep changing in accordance with the dynamic CSTR system, which is formulated by mass–energy differential Eq. (7).

3.2.3 The resulting problem formulation

In summary, the dynamic optimization problem of the CRP can be formulated as follows:

$$\min_{\boldsymbol{u}(t)} J(\boldsymbol{u}(t)) = \int_{t_0}^{t_f} u_1(t) + u_2(t)\mathrm{d}t,$$
$$\text{s.t.} \begin{cases} \text{Dynamic equation (7)} \\ \boldsymbol{x}(t_0) = [x_1(0), x_2(0)] \\ C_{Cu^{2+},\min} \le x_2(t_f) \le C_{Cu^{2+},\max} \\ u^l_{\text{zinc}} \le \boldsymbol{u}(t) \le u^u_{\text{zinc}} \\ t \in [t_0, t_f], \end{cases} \tag{12}$$

where $x(t_0)$ is the initial condition for the dynamic system and t_f means the effective residence time of solution in reactors.

Now, such a typical nonlinear dynamic optimization problem above is really challenging to solve, not only due to their highly nonlinear and multidimensional nature, but also due to the presence of terminal and path constraints on both state and control variables.

4 Proposed dynamic optimization approach

4.1 Control vector parameterization

Control vector parameterization (CVP) method is widely used for solving dynamic optimization problems involving differential equation systems or transient processes. Instead of discreting both state and control variables, the idea of CVP is to only approximate the control variables by a linear combination of basis functions so that an infinite-dimensional control problem can be transformed into a finite-dimensional nonlinear programming problem, which subsequently can be solved by a suitable numerical optimization algorithm.

In this section, we introduce a general uniform subinterval approximation scheme. Applying this approximation scheme, the time horizon was evenly partitioned by prefixed time knot $t_k, k = 0, \ldots, n$, where $n \ge 1$ is the number of the discrete time intervals, with the knot points satisfying

$$t_0 < t_1 < t_2 < \cdots < t_{n-1} < t_n = t_f, \tag{13}$$

Besides, the distance of each time knot is equal, so the subinterval solely determined by parameter n:

$$t_k = t_0 + k \times (t_f - t_0)/n \cdot \quad k = 0, \ldots, n \tag{14}$$

In terms of the kth subinterval, with the piecewise basis function, the control trajectory over the entire time span is approximated as follows:

$$u(t) = \sum_{k=1}^{n} \delta_k(t) u^k(t), t \in [t_0, t_f]$$
$$\delta_k(t) = \begin{cases} 1, & t \in [t_{k-1}, t_k] \\ 0, & \text{else} \end{cases}, \quad k = 0, \ldots, n \tag{15}$$

where $[t_{k-1}, t_k]$ is the kth control subinterval, and $u^k(t)$ is the approximating control function defined on the kth subinterval. Note that, the control variables at each time stage are represented with common basis functions, such as piecewise-constant function, linear function, quadratic function or polynomial function [26]. This approximation scheme is illustrated in Fig. 2.

So far, applying this approximation scheme yields a finite-dimensional approximation of the original optimal control problem, and a new cost function has been yield as below:

$$\min J = \Psi_0(\boldsymbol{x}(t_f), t_f) + \sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} L_0\big(u^k(t), \boldsymbol{x}(t), t\big)\mathrm{d}t$$
$$t \in [t_0, t_f], \tag{16}$$

where control heights and fixed switching knots need to be chosen optimally.

By designing special algorithms for computing the gradients of the cost and constraint function, the approximate problem can be solved using standard gradient-based optimization techniques. However,the gradient-based optimization methods are dependent on the gradient information and easy to get trapped into local minima. Therefore, a global optimization method based on STA will be introduced to overcome this limitation.

4.2 Proposed improved STA

4.2.1 The basic STA

State transition algorithm (STA) is a global stochastic optimization algorithm proposed by Zhou [15], and it has already shown appealing features as a efficient algorithm for solving various optimization problems [27–29]. In STA, a solution to an optimization problem is considered as a state, and an update of a solution can be treated as a state transition. As time goes by, the solution will be transferred to the optimal state gradually by its special transformation operators. A detailed algorithm procedure of the basic STA can be seen in [15].

The transformation operators serve to generate a candidate solution set, $\boldsymbol{x}_{k+1}$, based on the current state $\boldsymbol{x}_k$ and historical states $\boldsymbol{x}_{k-1}$. Generally, four special state transformation operators are designed [30].

1. Rotation transformation (RT)

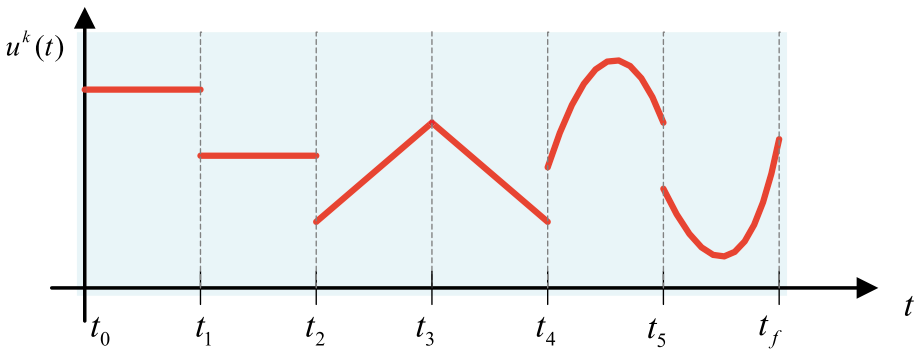

Fig. 2 Piecewise control approximation with equidistant knot points

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \frac{1}{n\|\boldsymbol{x}_k\|_2} R_r \boldsymbol{x}_k, \tag{17}$$

where α is a positive constant, called the rotation factor; $R_r \in \mathbb{R}^{n\times n}$, is a random matrix with its entries being uniformly distributed random variables defined on the interval $[-1, 1]$, and $\|\cdot\|_2$ is the 2-norm of a vector.

2. Translation transformation (TT)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \beta R_t \frac{\boldsymbol{x}_k - \boldsymbol{x}_{k-1}}{\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|_2}, \tag{18}$$

where β is a positive constant, called the translation factor; $R_t \in \mathbb{R}$ is an uniformly distributed random variable defined on the interval [0,1].

3. Expansion transformation (ET)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \gamma R_e \boldsymbol{x}_k, \tag{19}$$

where γ is a positive constant, called the expansion factor; $R_e \in \mathbb{R}^{n\times n}$ is a random diagonal matrix with its entries obeying the Gaussian distribution.

4. Axesion transformation (AT)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \delta R_a \boldsymbol{x}_k, \tag{20}$$

where δ is a positive constant, called the axesion factor; $R_a \in \mathbb{R}^{n\times n}$ is a random diagonal matrix with its entries obeying the Gaussian distribution and only one random position having nonzero value.

4.2.2 The improved STA

State transformation operator is a core body of techniques that enable STA to make good use of the historical states' information, to generate satisfactory candidate solutions and to make better state updates. The transformation factor in aforementioned operators is critical for the performance of STA. A large transformation factor enables operators to search in unexplored areas and facilitate the global search. At the same time, a small transformation factor promotes the fine-tuning of a solution in a local region and benefits local search. To address this issue, an adaptive strategy for transformation factors is suggested to balance the global and local search in the proposed STA.

In the basic STA, the rotation factor α is decreasing only according to the iterations and translation factor β even remains constant during the whole optimization procedure. While, in the proposed STA, the transformation factors will be changed adaptively according to the relative improvement between the last two best objective function values.

An adaptive strategy is adopted as follows: first, a counter c is defined to count the relative improvements:

$$c = \begin{cases} \max(0, c-1), & \text{if} \quad f(\text{Best}_k) - f(\text{Best}_{k-1}) > \tau \\ \max(0, c+1), & \text{else} \end{cases} \tag{21}$$

where τ is a specified tolerance suggested to be 10^{-4} and $f(\text{Best}_k)$ is the objective function value in the kth iteration. And then α and β are updated according to the following formula:

$$\alpha(\beta) = \begin{cases} fc \times \alpha(\beta), & \text{if} \quad c < C_1 \\ \frac{1}{fc} \times \alpha(\beta), & \text{if} \quad c > C_2 \\ \alpha(\beta), & \text{else} \end{cases}, \tag{22}$$

where C_1 and C_2 denote the lower and upper threshold values of the counter, and fc is suggested to be 2. If the counter is smaller than C_1, the factor will be multiplied by fc to promote global search, and if the counter is larger than C_2, the factor will be divided by fc to promote local search.

In addition to adopt an adaptive parameter strategy, we remove axesion operation to reduce the computational complexity of the proposed STA.

The procedures of the proposed method can be described as follows:

Step 1: Initialization: Randomly generate an initial solution Best_0 in the feasible solution space. Set the algorithm parameters, including $\alpha = \alpha_{\max}$, $\alpha_{\min}$, $\beta = \beta_{\max}$, $\beta_{\min}$, γ, fc and SE; let $k = 0$, $c = 0$.

Step 2: Expansion: Generate *SE* candidate solutions utilizing the ET operator based on the current best solution Best_k, then update the current best solution, and if Best_k has improved, then implement the TT operator and update Best_k as described above; otherwise not.

Step 3: Rotation: Generate *SE* candidate solutions utilizing the RT operator based on the current best solution Best_k, then update the current best solution, and if Best_k has improved, then implement the TT operator and update Best_k as described above; otherwise not.

Step 4: Parameter update: The rotation factor α and translation factor β are updated by Eqs. (21) and (22).

Step 5: Termination: If α and β exceed the minimum or maximum values, they are reset to the lower or upper bound values, accordingly. Let $k = k + 1$, go to step 2 until the specified termination criterion is met, and Best_k is considered a global minimizer of the problem.

4.3 Proposed dynamic optimization approach

CVP method constructs an approximated control function and allows us to estimate the fitness functions in a more computationally efficient way, and the global optimization method, STA, provide a global optimal solution of the resulting NLP avoiding getting trapped in the local optima. In this section, by integrating CVP with STA, a dynamic optimization approach named CVP–STA is formed.

Note that a piecewise-constant approximation scheme is the most commonly used in practice, owning to its simplicity, easy to implement, strong convergence properties and its versatility at handing non-standard optimal control problem [10]. In this paper, without loss of generality, we use piecewise-constant functions to approximate control variables.

With piecewise-constant basis functions, the control variables corresponding to the discrete time interval is approximated as follows:

$$\begin{aligned} &u(t) \approx u^k(t) = u_k, \\ &t \in [t_{k-1}, t_k], k = 1, \ldots, n, \end{aligned} \tag{23}$$

where $[t_{k-1}, t_k]$ is the kth control subinterval, and u_k is a constant value for the kth control stage.

According to the approximation scheme presented above, the approximate problem only involving a finite number of decision variables has been proposed, with its optimization variable vector

$$\chi = [u_1, \ldots, u_n], \tag{24}$$

and only the control parameters need to be chosen optimally. Furthermore, the control intervals number n can also be taken as decision variables to be optimized. In this paper, for simplicity, we focus solely on the typical control parameterizations method in which the number of control intervals n are fixed.

Based on the specific details of the uniform subinterval approximation scheme, a flowchart of the proposed method is given in Fig. 3.

Remark 1 Basically, a heuristic stopping criterion is formulated based on checking the relative improvement of the best objective function if

$$\left| \frac{J^k - J^{k-\Delta}}{J^k} \right| \leq \varepsilon, \tag{25}$$

where ε is a sufficiently small positive number as the tolerance precision for terminating the search process of problem and Δ is an appropriate value to designate comparison interval.

However, in order to avoid sinking into the iterations for too long, another stopping criterion is proposed if

$$iter \geq MaxIter, \tag{26}$$

where *MaxIter* is an appropriate large number which usually given in advance.

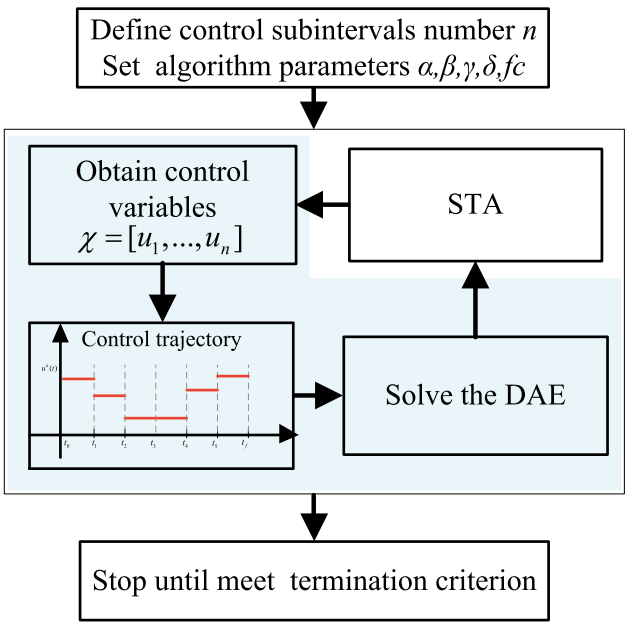

Fig. 3 The flowchart of CVP–STA

5 Simulation results and discussion

5.1 Example validation

In this section, we present several numerical simulations. Two typical real-word dynamic optimization problems are applied to demonstrate the effectiveness and efficiency of the approach formed for DOPs. For comparison, experiments adopt several recent evolution algorithms such as real-coded genetic algorithm (GA) [31], artificial bee colony (ABC) algorithm [32], adaptive particle swarm optimization (PSO) [33] and the basic state transition algorithm (STA) to solve the DOPs. For the sake of fairness and normalization, all of these calculations are carried on MATLAB (Version R2016b) software platform using 3.4 GHz Intel i7 PC with 8 G RAM. The built-in routine "ode45" is chosen as ODE integrator, and the same termination criteria is adopted for all test algorithms as shown in Remark 1 with $\Delta = 20$, $\epsilon = 10^{-6}$ and $MaxIter = 100 * Dim$, where Dim is the number of variables. The basic parameters of algorithms are shown in Table 1, which were recommended in their corresponding references.

Table 1 Basic parameters setting

Algorithm	Parameter	Value
Improved STA	SE	10
	Rotation factor range	[1e−4, 1]
	Translation factor range	[0.1, 10]
	Expansion factor	1
	Lower threshold	4
	Upper threshold	9
Basic STA [15]	SE	30
	Rotation factor range	[1e−4, 1]
	Translation factor	1
	Expansion factor	1
	Axesion factor	1
GA [31]	Population size	20
	Crossover rate	0.95
	Tournament size	2
	Mutation rate	0.05, 0.1, 0.2
ABC [32]	Colony size	30
	Food sources number	0.5 × colony size
	Limit	100
PSO [33]	Swarm size	30
	Inertia range	[0.1, 1.1]
	Self adjustment weight	1.49
	Social adjustment weight	1.49
	Minimum neighborhood size	0.25 × swarm size

During the test, the control variable is discretized into n intervals, and each algorithm for different control stages is run 30 times independently in order to ensure that the random initial solution does not generate any influence on the quality of the results obtained. Several common statistical indexes are employed to evaluate the performance of these method. Note that time is the average of 30 different runs.

Remark 2 The time complexity for all algorithms is presented here in brief. The time complexity of the evolutionary algorithm mainly consists of two parts: the search operators for generating candidate solutions and the calculation of the fitness, and it is generally a function of population size, problem type and the number of generations. For GA, the time complexity is $O(G \times D \times m)$ [34]. Here D is the dimension of a problem, G is the maximum number of iterations, and m is the size of population. Similarly, for PSO, ABC, the basic STA and the improved STA, the time complexity are the same, denoted as $O(G \times D \times m)$, where G, D, and m implies the same as those in GA.

5.1.1 Case 1: Plug flow reactor catalyst blend problem

In this problem, the objective is to obtain the optimal catalyst concentration profile that maximizes the yield of the intermediate product C for a fixed reactor length where the reaction $A \leftrightarrow B \rightarrow C$ occurs. The model equations describing the system dynamics of the reactant and product concentrations are given in [35]. Denoting x_A, x_B and J as the mole fractions of the substance A, B and C, the dynamic optimization problem is described as:

$$\max J = 1 - x_A(t_f) - x_B(t_f)$$
$$\text{s.t.} \begin{cases} \dot{x}_A = u(k_2 x_B - k_1 x_B) \\ \dot{x}_B = u(k_1 x_A - k_2 x_B) - (1-u)k_3 x_B \\ k_1 = 1, k_2 = 10, k_3 = 1 \\ x_A(0) = 1, x_B(0) = 0 \\ 0 \le u(t) \le 1 \\ t_f = 12, \end{cases} \tag{27}$$

where u is the fraction of catalyst contained in the mixture need to be chosen optimally.

During the test, the control variable is discretized into $n = 5, 10, 15$ intervals, and the results of 30 independent runs of each algorithm for different control stages are showed in Table 2, and the optimal control trajectory and state profiles of this problem obtained by improved STA are illustrated in Fig. 4.

Table 2 lists a comparison of results for algorithms using different number of control slots, and the bold symbol denotes better results. It is obvious that the

Table 2 Numerical simulation results for test problems

	n	Algorithm	Best	Mean	Worst	SD	Time (s)
Problem 1 (max)	5	Improved STA	0.472599	0.472599	0.472599	**1.206484e−09**	**5.099405**
		Basic STA [15]	0.472599	0.472599	0.472599	1.149066e−**07**	7.917762
		GA [31]	0.472599	0.472594	0.472530	1.252269e−05	11.116829
		ABC [32]	0.472599	0.472599	0.472599	1.117669e−07	5.501882
		PSO [33]	0.472599	0.472599	0.472599	8.557214e−09	5.763425
	10	Improved STA	0.473630	0.473630	0.473630	**9.716912e−08**	**15.329338**
		Basic STA [15]	0.473630	0.473628	0.473625	1.294347e−06	17.288272
		GA [31]	0.473630	0.473578	0.473441	4.568647e−05	26.030803
		ABC [32]	0.473630	0.473630	0.473627	7.013694e−07	19.016008
		PSO [33]	0.473630	0.473630	0.473630	1.211241e−07	16.582363
	15	Improved STA	0.474531	0.474531	0.474530	**1.029255e−07**	**33.564853**
		Basic STA [15]	0.474530	0.474528	0.474521	1.712338e−06	38.032976
		GA [31]	0.474531	0.474356	0.473995	1.176201e−04	40.849704
		ABC [32]	0.474531	0.474530	0.474529	2.843568e−06	46.333104
		PSO [33]	0.474531	0.474531	0.474530	1.827999e−07	36.512235
Problem 2 (min)	10	Improved STA	0.137258	**0.137258**	**0.137261**	**8.909569e−07**	**20.365621**
		Basic STA [15]	0.137258	0.137290	0.137385	2.990238e−05	24.974895
		GA [31]	0.137259	0.137318	0.137620	1.018687e−04	53.317429
		ABC [32]	0.137291	0.137838	0.140144	6.220728e−04	45.113475
		PSO [33]	0.137258	0.137332	0.137994	2.244053e−04	20.716706
	15	Improved STA	0.134969	**0.134970**	**0.134971**	**4.285284e−07**	**44.395781**
		Basic STA [15]	0.134971	0.135002	0.135079	2.440593e−05	50.962329
		GA [31]	0.134969	0.135015	0.135279	8.640548e−05	105.488990
		ABC [32]	0.135173	0.136301	0.140907	1.172842e−03	95.618774
		PSO [33]	0.134969	0.134995	0.135376	8.205750e−05	49.241551
	20	Improved STA	0.134156	**0.134156**	**0.134157**	**4.002515e−07**	**75.504284**
		Basic STA [15]	0.134160	0.134188	0.134299	3.192759e−05	85.426558
		GA [31]	0.134157	0.134674	0.138198	9.767036e−04	120.017122
		ABC [32]	0.134363	0.135418	0.138694	8.861394e−04	142.202570
		PSO [33]	0.134155	0.134219	0.135289	2.532343e−04	81.210973

execution time increases with the number of control slots. For problem 1, all algorithms can find the global optimum with high precision, while the execution time of the improved STA is less than the remainders for all control slots, which indicates the high efficiency of improved STA. And owning to the adaptive parameters, the improved STA exhibits excellent global exploration and local exploitation ability with good mean, best, worst values and the smallest standard deviation values.

5.1.2 *Case 2: Optimization of continuous stirred tank reactor (CSTR)*

Optimization of continuous stirred tank reactor (CSTR) is a multimodal optimal control problem which has been considered by Cruz [13]. The nonlinear differential equations that result from a heat and mass balance governing the behavior of the process. The dynamic optimization problem of this problem can be described as follow

$$\min J = x_3(t_{\mathrm{f}})$$
$$\text{s.t.}\begin{cases} \dot{x}_1 = -(2+u)(x_1+0.25) + (x_2+0.5)e^{\frac{25x_1}{x_1+2}} \\ \dot{x}_2 = 0.5 - x_2 - (x_2+0.5)e^{\frac{25x_1}{x_1+2}} \\ \dot{x}_3 = x_1^2 + x_2^2 + 0.1u^2 \\ \boldsymbol{x}(0) = [0.09, 0.09, 0] \\ 0 \le u(t) \le 5 \\ t_{\mathrm{f}} = 0.78, \end{cases} \tag{28}$$

where u represents the manipulation of coolant feed flow need to be chosen optimally. x_1 represents the deviation

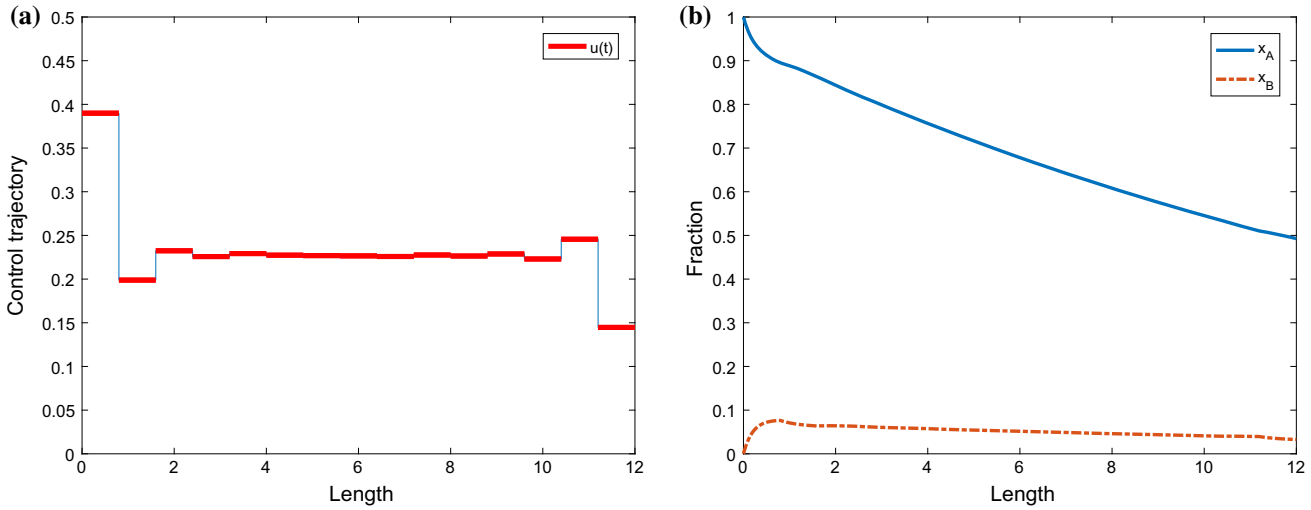

Fig. 4 Optimal results for problem 1 obtained by the improved STA. **a** Optimal control trajectory (case 1, $n = 15$), **b** optimal state profiles (case 1, $n = 15$)

from dimensionless steady-state temperature and x_2 stands for the deviation from the dimensionless steady-state concentration. In this problem, the objective is to obtain the optimal cooling fluid control profile $u(t)$ that minimizes the cumulative change in temperature x_1 and concentration x_2 without a large amount of coolant usage u.

In this case, the control variable is discretized into $n = 10, 15, 20$ intervals, and the results of 30 independent runs of each algorithm for different control stages are showed in Table 2, and the optimal control trajectory and state profiles of this problem obtained by improved STA is illustrated in Fig. 5.

From Table 2, we can see that the improved STA exhibits excellent global exploration and local exploitation ability for problem 2. In terms of the average execution time, it is clear that for all control slots the improved STA converges faster than PSO, GA and ABC algorithms, and at the same time, the improved STA has more accurate and stable solutions, which are quantified by the standard deviation and the worst results. Moreover, the mean results obtained by improved STA are always better than that of others, which signifies its superior stability and convergence performance. Therefore, adaptive transformation factors in the improved STA can not only promote the global exploration, but also improve the solution accuracy and the stability of the algorithm.

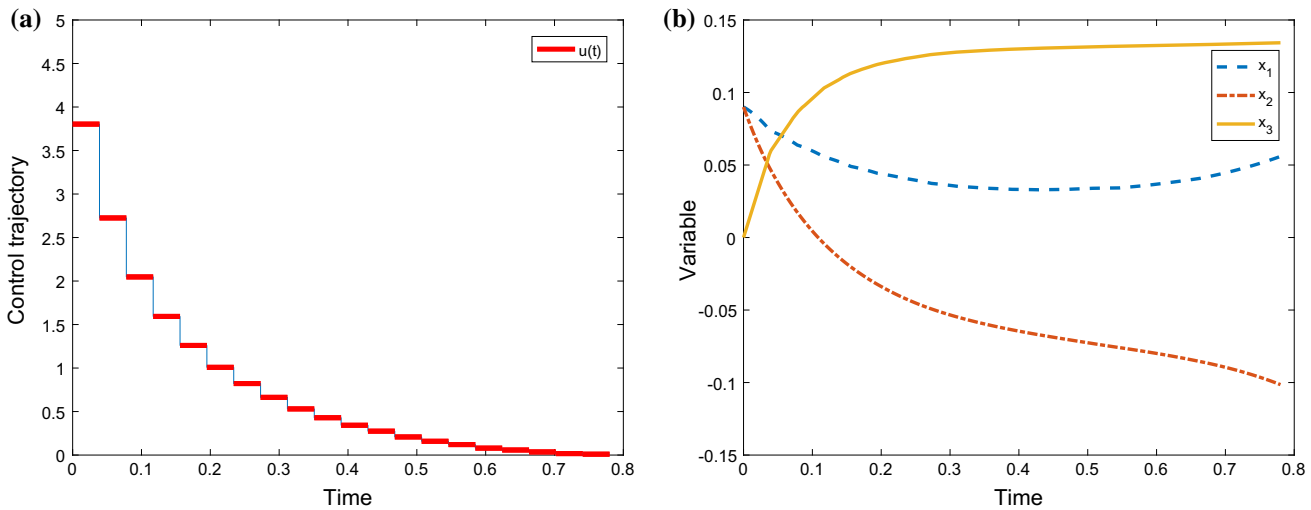

Fig. 5 Optimal results for problem 2 obtained by the improved STA. **a** Optimal control trajectory (case 2, $n = 20$), **b** optimal state profiles (case 2, $n = 20$)

Table 3 Numerical simulation results for CRP

	Algorithm	n	Best	Mean	Worst	SD	Time (s)
CRP (min)	Improved STA	5	556.8070	601.1436	615.2580	11.4147	56.6795
	Basic STA [15]	5	557.6382	611.9231	638.0023	19.4188	59.6975
	GA [31]	5	608.4483	614.4963	620.2041	3.1161	69.1452
	ABC [32]	5	604.7624	613.8727	624.0807	4.0373	87.3417
	PSO [33]	5	611.2100	615.9910	628.4048	3.8831	69.1380

5.2 Industrial numerical experiments

In this section, we use industry data of actual CRP to solve the dynamic optimization problem with the CVP–STA so as to verify its validity.

Normally, the solution volume V and solution flow rates Q and q, having no obvious fluctuation therefore are considered constant, and the detailed parameters can be found in [25]. In order to solve such a multi-variable dynamic optimization problem in CRP, the same control interval is adopted by two control variables and a penalty function is used to form their end-point state constrains. The choice of the number of piecewise interval n becomes an important issue. A sufficient large n makes the approximate optimal control trajectory converge to the true optima, while resulting in the curse of dimensionality and increasing optimization difficulty. In this paper, n is selected to be 5 to obtain a satisfactory optima while avoid heavy numerical calculation.

We first present the results for the case that the final time t_f is taken as 120 min (2 h). Based on the same uniform piecewise control approximation scheme, detailed comparisons of optimization results obtained by the improved STA, the basic STA, ABC, PSO, and GA have been listed in Table 3, and the experimental parameters are set the same with above experiments.

From the results, it is clear that the improved STA exhibits excellent performance in global search ability and stability, because the best (556.8070) and mean (601.1436) results of the improved STA are much better than that of others. Besides, we can see that the improved STA can converge to the optima faster as indicated by the average execution time (56.6795s). Although the standard deviation of the results obtained by the improved STA is a little big, the improved STA still has wonderful stability because the worst result (615.2580) is almost equal to the mean results obtained by GA, PSO and ABC algorithms. What's more, the standard deviation result (11.4147) of the improved STA is much smaller than that of the basic STA (19.4188), which indicates its improved local search ability. From these comparisons, it can be concluded that the improved STA gets very competitive performance with GA, PSO and ABC algorithm in solving the DOP of the CRP. The optimal control trajectory and state profiles of this problem obtained by the improved STA are illustrated in Fig. 6.

From Table 3 and Fig. 6, we can see that the total amount (601.1436 ± 11.4147 kg) of the zinc powder in 2 h under the dynamic optimization is much lower than the average amount (737.15 kg) of practical zinc powder used in real factory. At the same time, by applying the proposed approach, the oscillation range of the 2# outlet copper ion

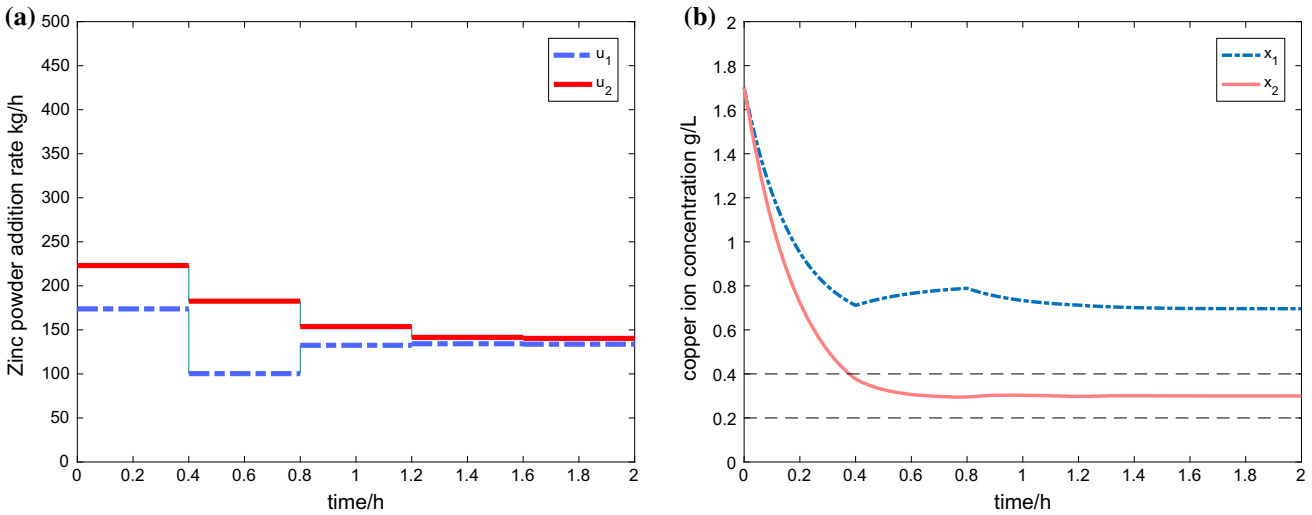

Fig. 6 Optimal results for DOP of the CRP obtained by the improved STA. **a** Optimal control trajectory ($n = 5$), **b** optimal state profiles ($n = 5$)

concentration can be stable near the center of the production limitation (0.2–0.4 g/L) without much fluctuation.

6 Conclusion

To further improve productivity and reduce production costs, the dynamic optimization of zinc powder addition in the CRP was studied in this paper. Based on a comprehensive analysis of CRP, a dynamic optimization problem is formulated to obtain a qualified ion concentration with the least zinc powder consumption. As for the complex DOPs, a dynamic optimization method, named CVP–STA, has been proposed. In this method, the original infinite-dimensional DOPs can be successively transformed into a finite-dimensional NLP by means of CVP method, and a global meta-heuristic algorithm, STA, is introduced to solve the resulting NLP. An adaptive parameter strategy is proposed to improve the performance of STA. Simulation results on two classical DOPs and industrial applications have demonstrated that the proposed method can optimize effectively and efficiently and can be a promising alternative method for solving engineering DOPs.

Acknowledgements Authors thank the National Natural Science Foundation of China (Grant Nos. 61503416, 61533020, 61533021, 61590921), the 111 Project (Grant No. B17048), National Priority Research Project NPRP 8-274-2-107, funded by Qatar National Research Fund and the Fundamental Research Funds for the Central Universities of Central South University (Grant No. 2017zzts487) for the funding support.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Balarini JC, de Oliveira Polli L, Miranda TLS, de Castro RMZ, Salum A (2008) Importance of roasted sulphide concentrates characterization in the hydrometallurgical extraction of zinc. Miner Eng 21(1):100–110
2. Laatikainen K, Lahtinen M, Laatikainen M, Paatero E (2010) Copper removal by chelating adsorption in solution purification of hydrometallurgical zinc production. Hydrometallurgy 104(1):14–19
3. Srinivasan B, Palanki S, Bonvin D (2003) Dynamic optimization of batch processes: I. Characterization of the nominal solution. Comput Chem Eng 27(1):1–26
4. Açıkmeşe B, Blackmore L (2011) Lossless convexification of a class of optimal control problems with non-convex control constraints. Automatica 47(2):341–347
5. Wongrat W, Younes A, Elkamel A, Douglas PL, Lohi A (2011) Control vector optimization and genetic algorithms for mixed-integer dynamic optimization in the synthesis of rice drying processes. J Frankl Inst 348(7):1318–1338
6. Angira R, Santosh A (2007) Optimization of dynamic systems: a trigonometric differential evolution approach. Comput Chem Eng 31(9):1055–1063
7. Biegler LT, Grossmann IE (2004) Retrospective on optimization. Comput Chem Eng 28(8):1169–1192
8. Biegler LT (2007) An overview of simultaneous strategies for dynamic optimization. Chem Eng Process 46(11):1043–1053
9. Schlegel M, Stockmann K, Binder T, Marquardt W (2005) Dynamic optimization using adaptive control vector parameterization. Comput Chem Eng 29(8):1731–1751
10. Lin Q, Loxton R, Teo KL (2014) The control parameterization method for nonlinear optimal control: a survey. J Ind Manag Optim 10(1):275–309
11. Sarkar D, Modak JM (2003) Optimisation of fed-batch bioreactors using genetic algorithms. Chem Eng Sci 58(11):2283–2296
12. Chen X, Du W, Tianfield H, Qi R, He W, Qian F (2014) Dynamic optimization of industrial processes with nonuniform discretization-based control vector parameterization. IEEE Trans Autom Sci Eng 11(4):1289–1299
13. Cruz IL, Van Willigenburg L, Van Straten G (2003) Efficient differential evolution algorithms for multimodal optimal control problems. Appl Soft Comput 3(2):97–122
14. Zhang H, Cao X, Ho JK, Chow TW (2017) Object-level video advertising: an optimization framework. IEEE Trans Ind Inf 13(2):520–531
15. Zhou X, Yang C, Gui W (2012) State transition algorithm. J Ind Manag Optim 8(4):1039–1056
16. Zhou X, Shi P, Lim CC, Yang C, Gui W (2017) A dynamic state transition algorithm with application to sensor network localization. Neurocomputing. doi:10.1016/j.neucom.2017.08.010
17. Han J, Yang C, Zhou X, Gui W (2017) A new multi-threshold image segmentation approach using state transition algorithm. Appl Math Model 44:588–601
18. Han J, Yang C, Zhou X, Gui W (2017) Dynamic multi-objective optimization arising in iron precipitation of zinc hydrometallurgy. Hydrometallurgy 173:134–148
19. Han J, Yang C, Zhou X, Gui W (2017) A two-stage state transition algorithm for constrained engineering optimization problems. Int J Control Autom Syst (**in press**)
20. Zhou X, Gao DY, Simpson AR (2016) Optimal design of water distribution networks by a discrete state transition algorithm. Eng Optim 48(4):603–628
21. Zhou X, Gao DY, Yang C, Gui W (2016) Discrete state transition algorithm for unconstrained integer optimization problems. Neurocomputing 173:864–874
22. Zhang F, Yang C, Zhou X, Gui W (2016) Fractional-order PID controller tuning using continuous state transition algorithm. Neural Comput Appl. doi:10.1007/s00521-016-2605-0
23. Näsi J (2004) Statistical analysis of cobalt removal from zinc electrolyte using the arsenic-activated process. Hydrometallurgy 73(1):123–132
24. Li YG, Gui WH, Teo KL, Zhu HQ, Chai QQ (2012) Optimal control for zinc solution purification based on interacting CSTR models. J Process Control 22(10):1878–1889
25. Zhang B, Yang C, Zhu H, Li Y, Gui W (2013) Kinetic modeling and parameter estimation for competing reactions in copper removal process from zinc sulfate solution. Ind Eng Chem Res 52(48):17074–17086
26. Irizarry R (2005) A generalized framework for solving dynamic optimization problems using the artificial chemical process paradigm: applications to particulate processes and discrete dynamic systems. Chem Eng Sci 60(21):5663–5681
27. Han J, Dong T, Zhou X, Yang C, Gui W (2014) State transition algorithm for constrained optimization problems. In: the 33rd Chinese control conference (CCC). IEEE, pp 7543–7548

28. Zhou X, Hanoun S, Gao DY, Nahavandi S (2015) A multiobjective state transition algorithm for single machine scheduling. In: Gao D, Ruan N, Xing W (eds) Advances in global optimization, vol 95. Springer, Cham, pp 79–88
29. Zhou X, Yang C, Gui W (2014) Nonlinear system identification and control using state transition algorithm. Appl Math Comput 226:169–179
30. Zhou X, Yang C, Gui W (2016) A matlab toolbox for continuous state transition algorithm. In: 2016 35th Chinese control conference (CCC). IEEE, pp 9172–9177
31. Tran T-D, Jin G-G (2010) Real-coded genetic algorithm benchmarked on noiseless black-box optimization testbed. In: Proceedings of the 12th annual conference companion on genetic and evolutionary computation. ACM, pp 1731–1738
32. Karaboga D, Ozturk C (2011) A novel clustering approach: artificial bee colony (abc) algorithm. Appl Soft Comput 11(1):652–657
33. Iadevaia S, Lu Y, Morales FC, Mills GB, Ram PT (2010) Identification of optimal drug combinations targeting cellular networks: integrating phospho-proteomics and computational network analysis. Cancer Res 70(17):6704–6714
34. Senthilnath J, Das V, Omkar SN, Mani V (2013) Clustering using levy flight cuckoo search. In: Bansal J, Singh P, Deep K, Pant M, Nagar A (eds) Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), vol 202. Springer, India
35. Dadebo S, McAuley K (1995) Dynamic optimization of constrained chemical engineering problems using dynamic programming. Comput Chem Eng 19(5):513–525