

Novel Control Vector Parameterization Method with Differential Evolution Algorithm and Its Application in Dynamic Optimization of Chemical Processes^{*}

SUN Fan (孙帆)¹, ZHONG Weimin (钟伟民)^{2,**}, CHENG Hui (程辉)¹ and QIAN Feng (钱锋)^{1,**}

¹ Key Laboratory of Advanced Control and Optimization for Chemical Processes (Ministry of Education), East China University of Science and Technology, Shanghai 200237, China

² Department of Automation, East China University of Science and Technology, Shanghai 200237, China

Abstract Two general approaches are adopted in solving dynamic optimization problems in chemical processes, namely, the analytical and numerical methods. The numerical method, which is based on heuristic algorithms, has been widely used. An approach that combines differential evolution (DE) algorithm and control vector parameterization (CVP) is proposed in this paper. In the proposed CVP, control variables are approximated with polynomials based on state variables and time in the entire time interval. Region reduction strategy is used in DE to reduce the width of the search region, which improves the computing efficiency. The results of the case studies demonstrate the feasibility and efficiency of the proposed methods.

Keywords control vector parameterization, differential evolution algorithm, dynamic optimization, chemical processes

1 INTRODUCTION

Chemical process is often modeled by a group of differential algebraic equations (DAEs). Dynamic optimization problems are encountered in many chemical engineering applications to optimize a predefined performance index such as profitability and productivity subject to safety or environmental specifications over a period by controlling operation variables [1–3]. A typical dynamic optimization problem for a continuous process is described as follows:

$$\min J(x, u) = \Phi[x(t_f), t_f] + \int_{t_0}^{t_f} \Psi[x(t), u(t), t] dt \quad (1)$$

where $x(t)$ is the state variable, $u(t)$ denotes the control variable, J is the performance index, t_0 is the initial time, t_f is the final time, Φ is a penalty function for the final state $x(t_f)$ at the final time t_f , and Ψ is a function to weight the state path traveled and control sequence used.

The objective function in Eq. (1) is essentially subject to the following equality constraints, which constitute the DAEs of the model representing the dynamics of the system to be optimized:

$$\begin{cases} \phi[x(t), u(t), t] = 0 \\ \frac{dx}{dt} = f[x(t), u(t), t] \end{cases} \quad \text{Obviously, the initial conditions}$$

for the state variables $x(t_0)$ should be given first in order to solve the model.

Dynamic optimization methods can be divided into dynamic programming (DP), indirect methods,

and direct methods [4–8]. DP solves dynamic optimization problems effectively, except the dimension curse, which impairs the applicability of the method. To overcome this drawback, iterative DP (IDP) has been proposed. IDP reduces the expansion of dimension and promotes reliability of the solution by iterative computation. For indirect methods, gradient based algorithms using the Hamiltonian function are reliable, such as conjugate gradient method and steepest descent method. However, the gradient information is not always available. Biegler *et al.* have done a lot of valuable work on simultaneous strategies for dynamic optimization [9–12]. This approach corresponds to a fully implicit Runge-Kutta method with high order accuracy and stability properties. Direct methods employ two different types of parameterization, namely, complete parameterization (CP) and control vector parameterization (CVP). CP discretizes both state and control variables, while CVP only discretizes control variables. CVP transforms the original problem into a non-linear programming (NLP) problem. However, it is not easy to find the global optimal solution by CVP method because the precision of the results is overly dependent on gradient information. When the gradient is difficult to calculate, the algorithm will get into trouble. Intelligent optimization algorithms, including genetic algorithm (GA), particle swarm optimization (PSO) and differential evolution (DE) algorithm, have been proved to be feasible methods when the gradient is difficult to compute [13–22].

In this study, the CVP method with DE algorithm is proposed for solving dynamic optimization problems. The CVP-DE method is applied to three case

Received 2012-07-30, accepted 2012-10-30.

^{*} Supported by the Major State Basic Research Development Program of China (2012CB720500), the National Natural Science Foundation of China (Key Program: U1162202), the National Science Fund for Outstanding Young Scholars (61222303), the National Natural Science Foundation of China (61174118, 21206037) and Shanghai Leading Academic Discipline Project (B504).

^{**} To whom correspondence should be addressed. E-mail: fqian@ecust.edu.cn; wmzhong@ecust.edu.cn

studies to demonstrate its effectiveness in dealing with dynamic optimization problems.

2 CVP WITH DE

A novel CVP method is proposed, in which the control variable is expressed by a polynomial function. Region reduction strategy is used in DE to search for the best coefficients of the polynomial.

2.1 CVP method

Referring to Eq. (1), the objective is to find the profile of the control variable $u(t)$, which optimizes the objective function. The “unknowns” in the optimization problem are the differential state variables $x(t)$ and control variables $u(t)$. There is usually a physical knowledge about systems, so it is not difficult to find functions to express the control variables $u(t)$. In the present work, when the form of the control variable discretization is given, it can be used to obtain the polynomial. Otherwise, the polynomial based on state variables and time to express the control variable is established according to the following rules: $u(t) = p_1x_1 + p_2x_2 + L + p_nx_n + p_{n+1}x_1^2 + L + p_{n+n}x_n^2 + p_{2n+1} + p_{2n+2}t$.

2.2 DE

DE is an efficient, effective and robust evolutionary optimization technique and is more likely to obtain the global solution as compared to several traditional optimization methods [23–25]. DE has three operations: mutation, crossover, and selection. The crucial idea behind DE is a scheme for generating trial vectors. Mutation and crossover are used to generate trial vectors, and selection then determines which of the vectors will survive into the next generation. Several variants to the DE algorithm have been suggested, a commonly used algorithm is as follows [26].

(1) Randomly create M individuals to form a population. The current generation number is denoted by Gen .

(2) Evaluate each individual.

(3) Create M individuals for a trial population as follows.

① Select three individuals at random, $F_{r_1}^{Gen} \neq F_{r_2}^{Gen} \neq F_{r_3}^{Gen}$, from the current population.

② Create a new trial vector $\mathbf{v}_j^{Gen+1}$ using the following mutation operator:

$$\mathbf{v}_j^{Gen+1} = F_{r_1}^{Gen} + P_F (F_{r_2}^{Gen} - F_{r_3}^{Gen}) \quad (2)$$

where indices r_1 , r_2 and r_3 are three distinct vectors randomly generated from the current generation and

$r_1, r_2, r_3 \in \{1, 2, \dots, M\}$, which are also different from index $j (j \in \{1, 2, \dots, M\})$, and P_F is the scale factor that is always in $(0, 1]$.

③ In the crossover, the target vector is mixed using the following scheme with the mutated vector, to yield the trial vector:

$$\mathbf{u}_j^{Gen+1} = \begin{cases} \mathbf{v}_j^{Gen+1}, & \text{rand}() \leq CR \\ F_j^{Gen} \text{ crossover with } \mathbf{v}_j^{Gen+1}, & \text{rand}() > CR \end{cases} \quad (3)$$

where $\text{rand}()$ is a random number and $\text{rand}() \in [0, 1]$, CR is the crossover probability and $CR \in [0, 1]$.

(4) For each individual F_j^{Gen} in the current population select the corresponding $\mathbf{u}_j^{Gen+1}$ in the trial population. With these two individuals, create offspring F_j^{Gen+1} by a greedy selection operator as follows:

$$F_j^{Gen+1} = \begin{cases} \mathbf{u}_j^{Gen+1}, & f(\mathbf{u}_j^{Gen+1}) < f(F_j^{Gen}) \\ F_j^{Gen}, & f(\mathbf{u}_j^{Gen+1}) \geq f(F_j^{Gen}) \end{cases} \quad (4)$$

If and only if, the trial vector $\mathbf{u}_j^{Gen+1}$ yields a better fitness function value than F_j^{Gen} , then $\mathbf{u}_j^{Gen+1}$ is set to F_j^{Gen+1} . Otherwise, the old value F_j^{Gen} is retained.

(5) Repeat steps (3) and (4) until a termination criterion is met.

2.3 Region reduction strategy in DE

Region reduction strategy is applied in DE to gradually reduce the width of the search region. At the k th step of region reduction, the searching region of parameters $p_i (i = 1, 2, \dots, n)$ is denoted by $[p_{i\min}^{(k)}, p_{i\max}^{(k)}]$ ($i = 1, 2, \dots, n$). At the beginning,

$$p_{i\min}^{(0)} = p_{i\min}, i = 1, 2, \dots, n, \quad (5)$$

$$p_{i\max}^{(0)} = p_{i\max}, i = 1, 2, \dots, n, \quad (6)$$

$$w_i^{(0)} = p_{i\max} - p_{i\min}, i = 1, 2, \dots, n, \quad (7)$$

where $p_{i\max}$ and $p_{i\min}$ are the initial estimated values, and w_i is the searching width.

Moving the baseline to the optimal value just obtained in the last region reduction step and using Eqs. (8)–(10) to make results more accuracy:

$$w_i^{(k)} = r w_i^{(k-1)}, i = 1, 2, \dots, n, \quad (8)$$

$$p_{i\min}^{(k)} = \frac{p_{ib} - w_i^{(k)}}{2}, i = 1, 2, \dots, n, \quad (9)$$

$$p_{i\max}^{(k)} = \frac{p_{ib} + w_i^{(k)}}{2}, i = 1, 2, \dots, n, \quad (10)$$

where r is reduction ratio, $0 < r < 1$, and p_{ib} is the optimal value obtained in the last region reduction step. Then return to run DE until $|J^{(k)} - J^{(k-1)}| < e$, where $J^{(k)}$ is the optimal objective function of the k th region reduction step.

2.4 Steps of the algorithm

The detailed steps of the proposed algorithm are given as follows.

Step 1 Generate randomly the original population of size M . The current generation number is denoted by Gen , and G is the maximum generations allowed. The region reduction step is denoted by k .

Step 2 Calculate objective function by the CVP method and evaluate each individual. The explicit Runge-Kutta method is used to solve the differential equations.

Step 3 Perform mutation and crossover operations to obtain the offspring. Perform mutation using Eq. (2) with a probability P_F . Perform crossover using Eq. (3) with a probability CR .

Step 4 Evaluate and select the offspring individual using Eq. (4).

Step 5 If $|J^{(k)} - J^{(k-1)}| > e$, go to Step 6, else $k = k + 1$ and compute $w_i^{(k)}$, $p_{i\min}^{(k)}$ and $p_{i\max}^{(k)}$ according to Eqs. (8)–(10).

Step 6 If $Gen \neq G$, return to Step 2, else stop.

3 CASE STUDIES

In this section, the algorithm is applied to several cases. The algorithm is carried out 10 times for each case, in which the parameters of DE are $M = 30$, $CR = 0.5$, and $P_F = 0.3$. The max evolutionary generations G and the reduction ratio r vary with different cases. It is suggested the value of r be chosen between $[0.5, 0.8]$.

3.1 Case 1

Case 1 is the temperature control for consecutive reaction problem. The objective is to obtain the optimal temperature profile that maximizes the yield of the intermediate product B at the end of operation in a batch reactor. The mathematical model is described as

$$\begin{aligned} \max J(T) &= x_2(t_f), \\ \text{s.t. } \frac{dx_1}{dt} &= -4000 \exp\left(\frac{-2500}{T}\right) x_1^2, x_1(0) = 1, \\ \frac{dx_2}{dt} &= 4000 \exp\left(\frac{-2500}{T}\right) x_1^2 - 620000 \exp\left(\frac{-5000}{T}\right) x_2, \\ x_2(0) &= 0, \end{aligned} \quad (11)$$

with $298 \leq T \leq 398$, $t_f = 1$, where x_1 is the concentration of A, x_2 is the concentration of B, and T is

temperature (K).

The optimum value of 0.610775 has been reported for this case. Rajesh *et al.* [13] obtained a value of 0.61045. Zhang *et al.* [17] obtained the values of 0.6100 ($n = 10$) and 0.6104 ($n = 20$) using iterative ant colony algorithm (IACA). With simulated annealing (SA) algorithm, Faber *et al.* [20] obtained a value of 0.610708.

3.2 Case 2

Case 2 is also a temperature control for consecutive reaction problem. The objective is to obtain the optimal temperature profile that maximizes the selectivity of the target product B at the end of operation in a batch reactor where the reaction $A \rightarrow B \rightarrow C$ takes place. The mathematical model is described as

$$\begin{aligned} \max J(T) &= C_B(t_f) / [1 - C_A(t_f)] \\ \text{s.t. } \frac{dC_A}{dt} &= -k_1 C_A, C_A(0) = 1, \\ \frac{dC_B}{dt} &= k_1 C_A - k_2 C_B, C_B(0) = 0, \end{aligned} \quad (12)$$

where $t_f = 12.5$ s, $k_i = k_{0i} \exp\left(\frac{-E_i}{RT}\right)$, $i = 1, 2$, $k_{01} = 65.6 \text{ s}^{-1}$, $k_{02} = 1970 \text{ s}^{-1}$, $E_1 = 4.18 \times 10^4 \text{ J} \cdot \text{mol}^{-1}$, $E_2 = 6.688 \times 10^4 \text{ J} \cdot \text{mol}^{-1}$, C_A is the concentration of A, C_B is the concentration of B, k_1 and k_2 are rate constants.

Rajesh *et al.* [13] used the objective that maximizes the concentration of product B at the end of operation and this case is denoted as Case 2 (a). With the constraint $C_B(t_f) = 0.45$, this problem becomes a constrained optimization problem and is denoted as Case 2 (b). With the constraints $C_B(t_f) \geq 0.44$ and $700 \leq T \leq 1000$, this problem becomes a more complicated problem and is denoted as Case 2 (c).

3.3 Case 3

Foreign protein production using recombinant bacteria was modeled by Lee and Ramirez [27]. The nutrient and inducer are requisite for foreign protein production. In order to yield maximal economic benefit, it is required to control the glucose feed rate and the inducer feed rate. The model is described as

$$\begin{aligned} \frac{dx_1}{dt} &= u_1 + u_2, \quad \frac{dx_2}{dt} = \mu x_2 - \frac{u_1 + u_2}{x_1} x_2, \\ \frac{dx_3}{dt} &= \frac{u_1}{x_1} C_{nf} - \frac{u_1 + u_2}{x_1} x_3 - Y^{-1} \mu x_2, \\ \frac{dx_4}{dt} &= R_{fp} x_2 - \frac{u_1 + u_2}{x_1} x_4, \quad \frac{dx_5}{dt} = \frac{u_2}{x_1} C_{if} - \frac{u_1 + u_2}{x_1} x_5, \\ \frac{dx_6}{dt} &= -k_1 x_6, \quad \frac{dx_7}{dt} = k_2 (1 - x_7). \end{aligned}$$

The state variables ($x_1, x_2, \dots, x_7$) are: reaction volume (x_1 in L), cell density (x_2 in $\text{g}\cdot\text{L}^{-1}$), nutrient concentration (x_3 in $\text{g}\cdot\text{L}^{-1}$), foreign protein concentration (x_4 in $\text{g}\cdot\text{L}^{-1}$), inducer concentration (x_5 in $\text{g}\cdot\text{L}^{-1}$), inducer shock factor on the cell growth rate (x_6), and the inducer recovery factor on the cell growth rate (x_7). The two control variables u_1 and u_2 are the glucose feed rate ($\text{L}\cdot\text{h}^{-1}$) and the inducer feed rate ($\text{L}\cdot\text{h}^{-1}$), respectively. Additional variables are growth yield coefficient (Y), nutrient concentration in nutrient feed (C_{nf} in $\text{g}\cdot\text{L}^{-1}$), inducer concentration in inducer feed (C_{if} in $\text{g}\cdot\text{L}^{-1}$), specific growth rate (μ in h^{-1}), foreign protein production rate (R_{fp} in h^{-1}), shock and recovery parameters (k_1, k_2).

The system parameters for this bacteria system are given as below:

$$\mu = \frac{0.407x_3}{(0.108 + x_3 + x_3^2)/14814.8} \left(x_6 + \frac{0.22}{0.22 + x_5} x_7 \right),$$

$$k_1 = k_2 = 0.09x_5 / (0.034 + x_5),$$

$$R_{\text{fp}} = \frac{0.095x_3}{(0.108 + x_3 + x_3^2)/14814.8} \times \left(\frac{0.0005 + x_5}{0.022 + x_5} \right),$$

$$C_{\text{if}} = 4.0, C_{\text{nf}} = 100.0.$$

The objective is to maximize $J(u_1, u_2)$ by controlling the glucose feed rate (u_1) and the inducer feed rate (u_2):

$$J(u_1, u_2) = x_4(t_f) - Q \int_{t_0}^{t_f} u_2(t) dt, \quad (13)$$

where Q is the price ratio of inducer to product, and a value of 5 is chosen, $t_f = 10$ h. The initial conditions

for the system are $\mathbf{x}(0) = (1, 0.1, 40, 0, 0, 1, 0)^T$. The control variables (u_1, u_2) is restricted in the range of $[0, 10^{-2} \text{ L}\cdot\text{h}^{-1}]$.

For this problem, Roubos *et al.* [28] obtained a value of 0.8149 using GA after 25000 generation (population size is 40). Sarkar and Modak [29] used a hybrid artificial neural network/simulated annealing algorithm (ANN-SA) and obtained a value of 0.816713. Zhang *et al.* [17] obtained a value of 0.8158 using IACA after three iterations (1200 cycles in total, population size is 100). Lin *et al.* [30] used hybrid optimal artificial immune network (HOAI-Net) and obtained a value of 0.8165.

3.4 Results and discussion

3.4.1 Results of Case 1

In Case 1, control variable approximation function is assumed as $T = p_0[1 + p_1 \exp(-p_2 x_1)] \times [1 + p_3 \exp(-p_4 x_2)]$, with the following constraints:

$$T = \begin{cases} 298, u < 298 \\ u, 298 \leq u \leq 398 \\ 398, u > 398 \end{cases}.$$

The initial searching region of all the parameters is set to $[0, 500]$, and $G = 60$, $r = 0.6$. The region reduction processes are shown in Table 1.

The maximal value reached 0.61073. And the values of the coefficients are $p = \{208.52884, 84.37439, 68.04497, 104.98534, 112.49267\}$. The concentrations vs. time and optimal temperature profile are shown in Figure 1.

3.4.2 Results of Case 2

In Case 2, the control variable approximation func-

Table 1 Region reduction processes for Case 1

Iteration	Searching region of p_0-p_4
1	$\{[0, 500], [0, 500], [0, 500], [0, 500], [0, 500]\}$
2	$\{[99.21, 399.21], [-61.75, 238.25], [-99.44, 200.56], [-56.36, 243.64], [-53.79, 246.21]\}$
3	$\{[112.36, 292.36], [-16.75, 163.25], [-30.78, 149.22], [7.62, 187.62], [29.13, 209.13]\}$
4	$\{[167.21, 275.21], [33.65, 141.65], [28.39, 136.39], [51.31, 159.31], [62.18, 170.18]\}$

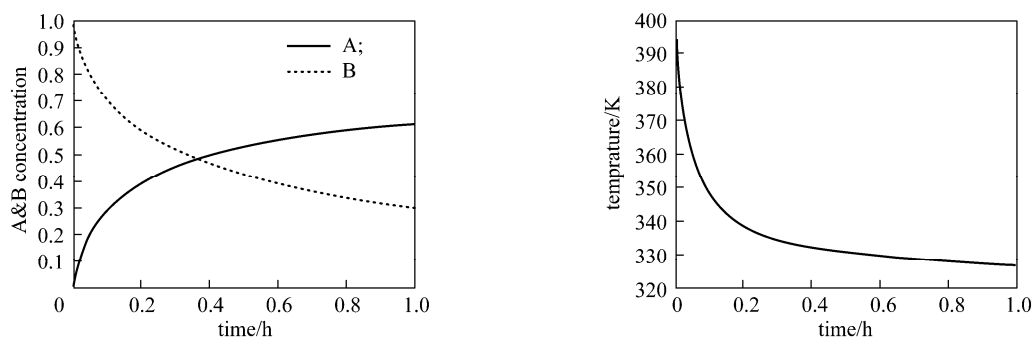


Figure 1 Concentrations and optimal temperature profile of Case 1

Table 2 Region reduction processes for Case 2 (a)

Iteration	Searching region of p_0-p_4
1	$\{[0, 1000], [-5, 5], [-50, 50], [-5, 5], [-50, 50]\}$
2	$\{[498.21, 998.21], [-2.61, 2.39], [-19.34, 30.66], [-2.38, 2.62], [-19.15, 30.85]\}$
3	$\{[611.32, 861.32], [-1.62, 0.88], [-6.81, 18.19], [-0.75, 1.75], [-4.22, 20.76]\}$
4	$\{[677.21, 802.21], [-0.99, 0.26], [-0.14, 12.36], [-0.13, 1.12], [2.33, 14.83]\}$

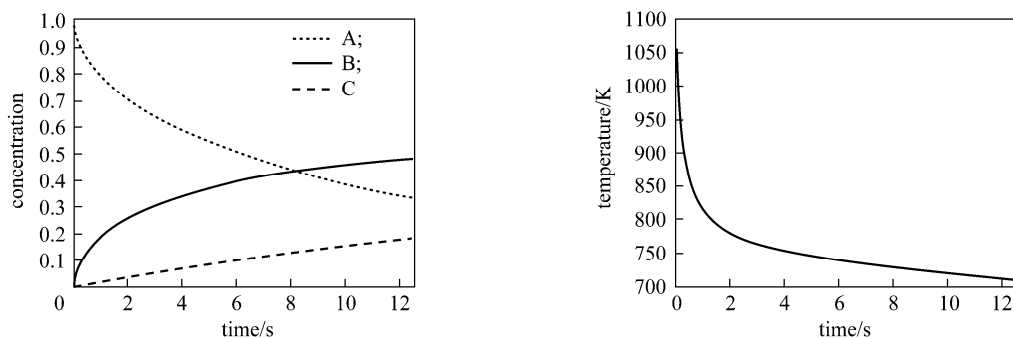


Figure 2 Concentrations and optimal temperature profile for Case 2 (a)

Table 3 Region reduction processes of Case 2 (b)

Iteration	Searching region of p_1-p_6
1	$\{[-500, 1500], [-500, 1500], [-500, 1500], [-500, 1500], [-200, 300], [-200, 300]\}$
2	$\{[381, 1381], [-625, 375], [-620, 380], [199, 1199], [-22, 228], [-109, 141]\}$
3	$\{[671, 1171], [-388, 112], [-366, 134], [494, 994], [36, 161], [-50, 75]\}$
4	$\{[777, 1027], [-258, -8], [-235, 15], [584, 834], [68, 131], [-21, 42]\}$

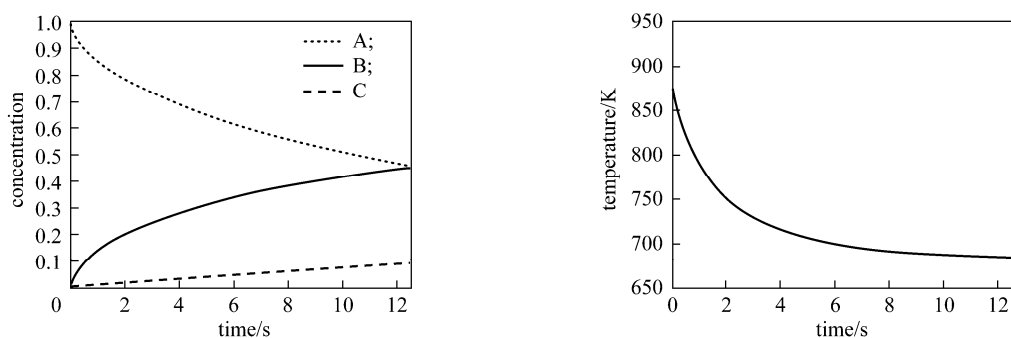


Figure 3 Concentrations and optimal temperature profile for Case 2 (b)

tion is $T = p_0[1 + p_1 \exp(-p_2 C_A)][1 + p_3 \exp(-p_4 C_B)]$.

For Case 2 (a), $G = 60$ and $r = 0.5$. The region reduction processes are shown in Table 2. The optimal concentration of the target product B obtained is 0.480047. The values of the coefficients are $p = \{740.44471, -0.36149, 5.88172, 0.50283, 8.57771\}$. The concentrations vs. time and the optimal temperature profile are shown in Fig. 2.

For Case 2 (b), $G = 100$ and $r = 0.5$. The region reduction processes are shown in Table 3. The values of the coefficients are $p = \{75.613, -0.36608, 2.79374, 0.40929, 13.73314\}$. The concentrations of the products

at the exit of the reactor are $C_A(t_f) = 0.459456$, $C_B(t_f) = 0.4500000$, and $C_C(t_f) = 0.090544$. The yield coefficient of the target product reached 0.8325. The concentrations vs. time and temperature vs. time results are shown in Fig. 3.

For Case 2 (c), $G = 80$ and $r = 0.5$. The region reduction processes are shown in Table 4. The values of the coefficients are $p = \{701.2786, -0.47160, 10.89443, 0.27727, 13.39296\}$. The concentrations of the products at the exit of the reactor are $C_A(t_f) = 0.47788$, $C_B(t_f) = 0.4400000$, and $C_C(t_f) = 0.08212$. The yield coefficient of the target product is 0.8427.

Table 4 Region reduction processes of Case 2 (c)

Iteration	Searching region of p_0 - p_4
1	$\{[0, 1000], [-5, 5], [-50, 50], [-5, 5], [-50, 50]\}$
2	$\{[449.13, 949.13], [-2.98, 2.02], [-14.06, 35.94], [-2.21, 2.79], [-10.75, 39.25]\}$
3	$\{[577.98, 827.98], [-1.71, 0.79], [-1.51, 23.49], [-0.98, 1.52], [0.91, 25.91]\}$
4	$\{[638.18, 763.18], [-1.10, 0.15], [4.65, 17.15], [-0.35, 0.90], [7.15, 19.65]\}$

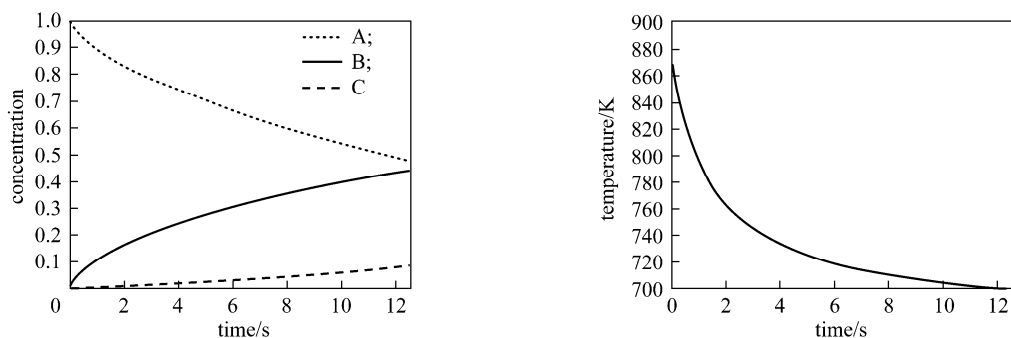


Figure 4 Concentrations and optimal temperature profile for Case 2 (c)

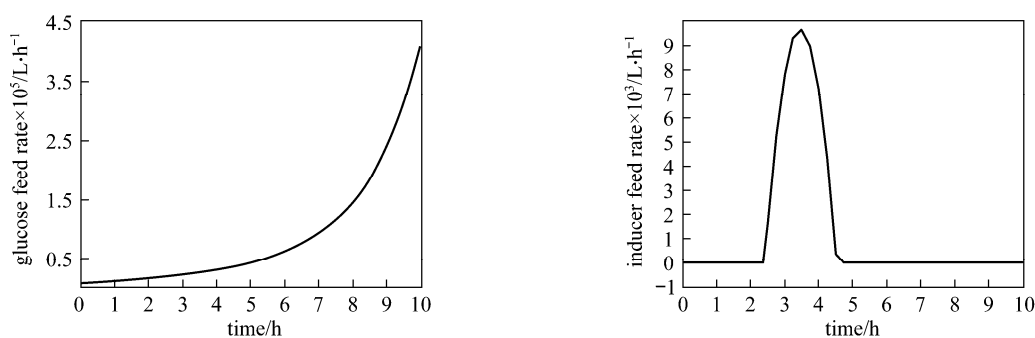


Figure 5 Optimal profiles of glucose feed rate and inducer feed rate

The concentrations vs. time and temperature vs. time results are shown in Fig. 4.

3.4.3 Results of case 3

For Case 3, the control variable approximation functions are given by

$$u_1 = p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 + p_6x_6 + p_7x_7 + p_8x_1^2 + p_9x_2^2 + p_{10}x_3^2 + p_{11}x_4^2 + p_{12}x_5^2 + p_{13}x_6^2 + p_{14}x_7^2 + p_{15} + p_{16}t,$$

$$u_2 = p_{17}x_1 + p_{18}x_2 + p_{19}x_3 + p_{20}x_4 + p_{21}x_5 + p_{22}x_6 + p_{23}x_7 + p_{24}x_1^2 + p_{25}x_2^2 + p_{26}x_3^2 + p_{27}x_4^2 + p_{28}x_5^2 + p_{29}x_6^2 + p_{30}x_7^2 + p_{31} + p_{32}t,$$

with the following constraints:

$$u_1 = \begin{cases} 0, u_1 < 0 \\ u_1, 0 \leq u_1 \leq 0.01 \\ 0.01, u_1 > 0.01 \end{cases}, \quad u_2 = \begin{cases} 0, u_2 < 0 \\ u_2, 0 \leq u_2 \leq 0.01 \\ 0.01, u_2 > 0.01 \end{cases}.$$

For this case, $G = 400$ and $r = 0.5$. The maximal value obtained by CVP-DE reached 0.81691. The optimal glucose feed rate and the optimal inducer feed rate are shown in Fig. 5. This case shows that the proposed method could also solve a problem with multiple state variables and more than one control variables.

3.4.4 Summary

All results for different cases are given in Table 5. There are no optimal solutions ever reported for Cases 2 (b), 2 (c) and 3, so the best values obtained by CVP-DE are considered as the optimal solutions. And “—” means “not reported”. In the table,

$$BE = \left| \frac{B_M - B_O}{B_O} \right|, \quad (14)$$

$$CE = C_t S_p \quad (15)$$

where “BE” means “relative error of best solution”, “CE” means “computational evaluation”, B_M is the best value obtained by the method, B_O is the optimal solution, C_t is the total cycles of the algorithm, and

Table 5 Summary of results

Case	Optimum	Method	Best	Mean	Worst	BE/%	CE
Case 1	0.610775	CACA ^① [13]	0.61045	—	—	0.053	—
		IACA [17]	0.6104	—	0.6100	0.061	30000–88000
		SA [20]	0.610708	—	0.610446	0.011	5597–335998
		CVP-DE	0.61073	0.61072	0.61070	0.007	7200
Case 2 (a)	0.480047	CACA [13]	0.4793	—	—	0.156	—
		IGA ^② [15]	0.4797	—	0.4791	0.072	10000–40000
		CVP-DE	0.480047	0.480045	0.480041	0	7200
Case 2 (b)	—	PSO [31]	0.7986	—	—	4.072	300000
		GOS ^③ [21]	0.8315	—	—	0.120	7500
		CVP-DE	0.83250	0.83244	0.83235	0	12000
Case 2 (c)	—	ICSA ^④ [30]	0.83699	0.83592	0.83213	0.680	4159.3
		CVP-DE	0.84270	0.84257	0.84229	0	9600
Case 3	—	GA [28]	0.8149	—	—	0.242	1000000
		ANNSA [29]	0.816713	—	0.816620	0.020	2120801
		IACA [17]	0.8158	—	—	0.132	120000
		HOpt-AiNet [30]	0.8165	0.8164	0.8162	0.047	11745
		CVP-DE	0.81691	0.81683	0.81671	0	80000

① CACA: complicate ant colony algorithm. ② IGA: iterative genetic algorithm. ③ GOS: graded optimization strategy. ④ ICSA: improved clonal selection algorithm.

S_p is the population size.

4 CONCLUSIONS

From the case studies, it is concluded that the proposed methods in this paper can be used to solve dynamic optimization problems efficiently and accurately. In the proposed CVP method, the control profile is approximated parametrically as a function of states and time. With CVP, the dynamic optimization problem is converted into a NLP problem in which unknown parameters are determined using DE. In addition, the estimation of this function is not difficult and can be performed simply especially when the physical knowledge of the system is available. In order to quickly reduce the range of the coefficients, region reduction strategy is used in DE to gradually reduce the width of the search region. By the region reduction strategy, the search range is reduced significantly and the probability of getting the optimal solution has greatly increased. This method is compared with CACA, IDP and IACA by solving three dynamic optimization cases. It is shown that the proposed algorithm is easy to implement with satisfactory accuracy.

NOMENCLATURE

B_M	the best value obtained from different algorithms
B_O	the optimal solution of cases
C	concentration
CR	crossover probability

C_t	total cycles
e	constant
f	fitness of individuals
G	maximum generations
Gen	current evolutionary generation
i	number
J	performance index
k	iteration times
M	number of individuals in the population
P_F	mutation probability
p_i	polynomial coefficients
r	reduction ratio
S_p	population size of different algorithms
T	temperature, K
t_f	final time, s
t_0	initial time, s
u	control variables
w	searching width
x	state variables
Φ	penalty function
ψ	function

REFERENCES

- 1 Srinivasan, B., Palanki, S., Bonvin, D., "Dynamic optimization of batch processes (I) Characterization of the nominal solution", *Comput. Chem. Eng.*, **27**, 1–26 (2003).
- 2 Roberto, I., "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**, 5663–5681 (2005).
- 3 Zhang, B., Yu, H.J., Chen, D.Z., "Sequential optimization of chemical dynamic problems by ant-colony algorithm", *J. Chem. Eng. Chin. Univ.*, **20** (1), 120–125 (2006). (in Chinese)
- 4 Rakesh, A., Alladwar, S., "Optimization of dynamic systems: A trigonometric differential evolution approach", *Comput. Chem. Eng.*,

- 31, 1055–1063 (2007).
- 5 Eva, B.C., Vassiliou, S.V., Julio, R.B., “Dynamic optimization of single- and multi-stage systems using a hybrid stochastic-deterministic method”, *Ind. Eng. Chem. Res.*, **44**, 1514–1523 (2005).
- 6 Julio, R.B., Eva, B.C., Carmen, G.M., Antonio, A.A., “Dynamic optimization of bioprocesses: Efficient and robust numerical strategies”, *J. Biotechnol.*, **117**, 407–419 (2005).
- 7 Sihem, T., Didier, D., Ghizlane, H., “Open-loop optimization and trajectory tracking of a fed-batch bioreactor”, *Chem. Eng. Process.*, **47**, 1933–1941 (2008).
- 8 Jose, A.E., Eva, B.C., Maria, S.G., Julio, R.B., “Dynamic optimization of nonlinear processes with an enhanced scatter search method”, *Ind. Eng. Chem. Res.*, **48**, 4388–4401 (2009).
- 9 Biegler, L.T., Cervantes, A.M., Wachter, A., “Advances in simultaneous strategies for dynamic process optimization”, *Chem. Eng. Sci.*, **57**, 575–593 (2002).
- 10 Biegler, L.T., “An overview of simultaneous strategies for dynamic optimization”, *Chem. Eng. Process.*, **46** (11), 1043–1053 (2007).
- 11 Biegler, L.T., Zavala, V.M., “Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization”, *Comput. Chem. Eng.*, **33**, 575–582 (2009).
- 12 Antonio, F.T., Biegler, L.T., “Simultaneous mixed-integer dynamic optimization for integrated design and control”, *Comput. Chem. Eng.*, **31**, 588–600 (2007).
- 13 Rajesh, J., Kapil, G., Hari, S.K., Jayaraman, V.K., Kulkarni, B.D., “Dynamic optimization of chemical processes using ant colony framework”, *Comput. Chem.*, **25**, 583–595 (2001).
- 14 Debasis, S., Jayant, M.M., “Optimisation of fed-batch bioreactors using genetic algorithms”, *Chem. Eng. Sci.*, **58**, 2283–2296 (2003).
- 15 Zhang, B., Chen, D.Z., “Iterative genetic algorithm and its application to policies optimization for bioreactor”, *J. Chem. Ind. Eng. (China)*, **56** (1), 100–104 (2005). (in Chinese)
- 16 Hong, W.R., Tan, P.C., Wang, S., Pu, L., “Comparison of arithmetic operations for dynamic process optimization approach”, *Chin. J. Chem. Eng.*, **18** (1), 80–85 (2010).
- 17 Zhang, B., Chen, D.Z., Zhao, W., “Iterative ant-colony algorithm and its application to dynamic optimization of chemical process”, *Comput. Chem. Eng.*, **29**, 2078–2086 (2005).
- 18 Walid, T., Patrick, S., “A new charged ant colony algorithm for continuous dynamic optimization”, *Appl. Math. Comput.*, **197**, 604–613 (2008).
- 19 Shelokar, P.S., Jayaraman, V.K., Kulkarni, B.D., “Multicanonical jump walk annealing assisted by tabu for dynamic optimization of chemical engineering processes”, *Euro. J. Opera. Res.*, **185**, 1213–1229 (2008).
- 20 Faber, R., Tobias, J., George, T., “Dynamic optimization with simulated annealing”, *Comput. Chem. Eng.*, **29**, 273–290 (2005).
- 21 Zhang, B., Chen, D.Z., Wu, X., “Graded optimization strategy and its application to chemical dynamic optimization with fixed boundary”, *J. Chem. Ind. Eng. (China)*, **56** (7), 1276–1280 (2005). (in Chinese)
- 22 Lin, K.H., He, Y.J., Chen, D.Z., “Improved clonal selection algorithm for constrained dynamic optimization problems”, *J. Chem. Eng. Chin. Univ.*, **23** (5), 858–863 (2009). (in Chinese)
- 23 Fan, H. Y., Lampinen, J., “A trigonometric mutation operation to differential evolution”, *J. Glob. Opt.*, **27**, 105–129 (2003).
- 24 Angira, R., Babu, B.V., “Optimization of process synthesis and design problems: A modified differential evolution approach”, *Chem. Eng. Sci.*, **61**, 4707–4721 (2006).
- 25 Babu, B.V., Angira, R., “Modified differential evolution (MDE) for optimization of nonlinear chemical processes”, *Comput. Chem. Eng.*, **30**, 989–1002 (2006).
- 26 Mathys, C.P., Andries, P.E., “Using competitive population evaluation in a differential evolution algorithm for dynamic environments”, *Euro. J. Oper. Res.*, **1**, 1–14 (2011).
- 27 Lee, J., Ramirez, W.F., “Optimal fed-batch control of induced foreign protein production by recombinant bacteria”, *AIChE J.*, **40**, 899–907 (1994).
- 28 Roubos, J.A., Straten, G., Boxtel, A.J.B., “An evolutionary strategy for fed-batch bioreactor optimization: Concepts and performance”, *J. Biotechnol.*, **67**, 173–187 (1999).
- 29 Sarkar, D., Modak, J.M., “ANNSA: A hybrid artificial neural network/simulated annealing algorithm for optimal control problems”, *Chem. Eng. Sci.*, **58**, 3131–3142 (2003).
- 30 Lin, K.H., He, Y.J., Chen, D.Z., “Hybrid optimal artificial immune network and its application to dynamic process optimization”, *J. Zhejiang. Univ. (Eng. Sci.)*, **42** (12), 2181–2186 (2008). (in Chinese)
- 31 Mo, Y.B., Hu, S.X., Chen, D.Z., “Chaos particle swarm optimization algorithm and its application in biochemical process dynamic optimization”, *J. Chem. Ind. Eng. (China)*, **57** (9), 2123–2127 (2006). (in Chinese)