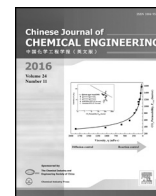




Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

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

Process Systems Engineering and Process Safety

Solving chemical dynamic optimization problems with ranking-based differential evolution algorithms☆

Xu Chen^{1,2}, Wenli Du^{1,*}, Feng Qian¹¹ Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China² School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China

ARTICLE INFO

Article history:

Received 9 June 2015

Received in revised form 1 September 2015

Accepted 2 December 2015

Available online 2 May 2016

Keywords:

Dynamic optimization

Differential evolution

Ranking-based mutation operator

Control vector parameterization

ABSTRACT

Dynamic optimization problems (DOPs) described by differential equations are often encountered in chemical engineering. Deterministic techniques based on mathematic programming become invalid when the models are non-differentiable or explicit mathematical descriptions do not exist. Recently, evolutionary algorithms are gaining popularity for DOPs as they can be used as robust alternatives when the deterministic techniques are invalid. In this article, a technology named ranking-based mutation operator (RMO) is presented to enhance the previous differential evolution (DE) algorithms to solve DOPs using control vector parameterization. In the RMO, better individuals have higher probabilities to produce offspring, which is helpful for the performance enhancement of DE algorithms. Three DE-RMO algorithms are designed by incorporating the RMO. The three DE-RMO algorithms and their three original DE algorithms are applied to solve four constrained DOPs from the literature. Our simulation results indicate that DE-RMO algorithms exhibit better performance than previous non-ranking DE algorithms and other four evolutionary algorithms.

© 2016 The Chemical Industry and Engineering Society of China, and Chemical Industry Press. All rights reserved.

1. Introduction

Dynamic optimization problems (DOPs) are often encountered in chemical engineering, as most industrial process models are time-dependent and described by differential equations. The solution of DOPs is usually very difficult because of their highly nonlinear and multidimensional nature, as well as the presence of constraints on state and control variables and implicit process discontinuities [1]. Given the profound importance of DOPs in industrial and engineering practices, developing efficient methods for DOPs has attracted great interest. Dynamic optimization methods can be roughly divided into three categories: dynamic programming (DP), indirect methods, and direct methods.

Classic DP method relies on Bellman's optimality [2]. DP is a successful method for solving DOPs, except for dimension curse. To overcome this drawback, Luus [3] proposed iterative dynamic programming method by use of coarse grid points and search region reduction strategies. However, its high computational cost for systems involving a large number of differential-algebraic equations has restricted its application to problems on a smaller scale [4].

With indirect methods, DOPs are solved by using Pontryagin's maximum principle [5]. It converts the original problem into a two-point boundary value problem, which rarely has an analytical solution and requires numerical alternative such as shooting method [6]. The two-point boundary value problem is always extremely difficult to solve, especially in the presence of point or path constraints on state variables. Therefore, indirect methods are extremely complicated to apply in practice.

Direct methods transform the original dynamic problem (which is infinite dimensional) into a finite dimensional non-linear programming (NLP) problem, either using complete parameterization (CP) [7] or control vector parameterization (CVP) [8,9]. CP method, also named simultaneous strategy, discretizes both state and control variables. This full discretization results in a large-scale NLP and specialized NLP solvers have to be used to solve the NLP efficiently. The CP method has been recently reviewed by Kameswaran and Biegler [7]. CVP method only discretizes control variables and transforms original DOP into an NLP where the system dynamics (differential equations) must be solved for each evaluation of the performance index. The dimension of the NLP problems in CVP is much smaller than that in CP. Therefore, CVP is relatively easier to implement, and a large number of optimization algorithms, including deterministic gradient-based algorithms [8] and stochastic optimization algorithms [10], have been combined with CVP to deal with DOPs.

The NLPs from the application of direct approaches (such as CVP) are frequently multimodal. Deterministic gradient-based algorithms may converge to local optima, especially if they are started far away from

☆ Supported by the National Natural Science Foundation of China (61333010, 61134007 and 21276078) and "Shu Guang" project of Shanghai Municipal Education Commission, the Research Talents Startup Foundation of Jiangsu University (15JDG139), and China Postdoctoral Science Foundation (2016M591783).

* Corresponding author.

E-mail address: wldu@ecust.edu.cn (W.L. Du).

the global solution [9]. In addition, explicit mathematical descriptions of industrial model sometimes do not exist, and methods based on gradient information may become incapable. To surmount these difficulties, stochastic optimization algorithms based on evolutionary computing can be used as robust alternatives. The use of evolutionary methods to optimize DOPs, including genetic algorithm [10–12], simulated annealing [13,14], particle swarm optimization [15–17], and scatter search [18], has received increasing interest.

Differential evolution (DE) is a population based stochastic optimization technique, invented by Storn and Price [19]. Since its inception in 1995, it has emerged as a very competitive form of evolutionary computing. The use of DE algorithms to solve DOPs also has drawn the attention of many researchers [20]. Chiou and Wang [20] developed a hybrid DE algorithm by embedding an acceleration phase and a migration phase into the original DE algorithm to solve DOPs in fed-batch fermentation process. Kapadi and Gudi [21] employed standard DE (SDE) to solve optimal control and parameter selection problems of fed-batch fermentation involving general constraints on state variables. To speed up DE and solve DOPs, Babu and Angira [22] introduced modified DE (MDE) that utilizes only one set of population compared with the two sets in the original DE at any given time point in a generation. Angira and Santosh [1] suggested the use of trigonometric DE (TDE) to solve DOPs encountered in chemical engineering. Fan *et al.* [23] proposed a hybrid DE algorithm called Alopex-DE by integrating Alopex to solve DOPs of chemical processes.

Das and Suganthan [24] pointed out that DE's weak selective pressure (due to unbiased selection of parents or target vectors) may result in inefficient exploitation. To overcome this weakness, Gong and Cai [25] presented a ranking-based mutation operator (RMO) for DE algorithms, in which better individuals have larger possibilities to be selected in mutation operator. This study deals with the utilization of RMO to enhance the performance of CVP-based DE algorithms for DOPs. The DOPs are first converted into NLP problems by CVP approach; then the RMO is incorporated into three DE algorithms, SDE [21], MDE [22], and TDE [1], to obtain three ranking-based differential evolution (DE-RMO) algorithms, *i.e.*, SDE-RMO, MDE-RMO, and TDE-RMO, to solve DOPs. Three DE-RMO algorithms and their non-ranking DE algorithms are applied to solve four constrained DOPs from previous studies. The simulation results indicate that the DE-RMO algorithms can provide better findings in terms of solution accuracy and convergence speed compared with previous non-ranking DE algorithms.

2. Formulation of Dynamic Optimization Problems

Dynamic optimization allows the computation of optimal operating policies to minimize (or maximize) a predefined performance index [18]. The objective function is formulated as

$$\min J = \psi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \phi(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (1)$$

where J is the objective function, $\psi(\mathbf{x}(t_f), t_f)$ is a Mayer item, and $\int_{t_0}^{t_f} \phi(\mathbf{x}(t), \mathbf{u}(t), t) dt$ is a Lagrangian term.

The dynamic model of a chemical process is often described by differential equations as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad t \in [t_0, t_f] \quad (2)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (3)$$

where $\mathbf{x}(t) \in R^n$ denotes the vector of state variables, $\mathbf{u}(t) \in R^m$ denotes the vector of control variables (or operational variables), and $\mathbf{x}(t_0) = \mathbf{x}_0$ is the initial conditions.

Four kinds of constraints may exist in the DOPs. They are path inequality constraints, path equality constraints, terminal inequality

constraints, and terminal equality constraints. Path constraints should be satisfied in the entire time interval $[t_0, t_f]$, *i.e.*,

$$g_i^{\text{path}}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0, \quad i = 1, \dots, l \quad (4)$$

$$h_i^{\text{path}}(\mathbf{x}(t), \mathbf{u}(t), t) = 0, \quad i = l + 1, \dots, p. \quad (5)$$

Terminal constraints should be satisfied only at the end of the process, *i.e.*,

$$g_i^{\text{end}}(\mathbf{x}(t_f), \mathbf{u}(t_f), t_f) \leq 0, \quad i = 1, \dots, k \quad (6)$$

$$h_i^{\text{end}}(\mathbf{x}(t_f), \mathbf{u}(t_f), t_f) = 0, \quad i = k + 1, \dots, q. \quad (7)$$

Control variables may be subject to lower and upper bounds as follows:

$$\mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U. \quad (8)$$

3. Differential Evolution Algorithms

DE is a population-based stochastic optimizer in the continuous search domain, proposed by Storn and Price [19]. It is capable of handling non-differentiable, non-linear and multi-modal optimization problems. DE initializes a population of NP individuals and employs mutation, crossover, and selection operators at each generation to evolve its population toward the optimal direction. DE population initializes NP individuals (each individual is called a target vector) from the search space

$$\mathbf{X}_i = (x_{i1}^g, x_{i2}^g, \dots, x_{iD}^g), \quad i = 1, \dots, NP \quad (9)$$

where g denotes the generation counter and D defines the number of variables.

After initialization, the mutation operator is applied to generate mutant vector $\mathbf{V}_i^g$ for each target vector $\mathbf{X}_i^g$. Many mutation strategies are described in previous studies. A classical strategy is "DE/rand/1":

$$\mathbf{V}_i^g = \mathbf{X}_{r1}^g + F \cdot (\mathbf{X}_{r2}^g - \mathbf{X}_{r3}^g) \quad (10)$$

where F is a scale factor, r_1 , r_2 , and r_3 are three mutually different integers randomly generated from $[1, NP]$ and different from the target vector index i .

After mutation, DE employs a crossover operator to produce the trial vector $\mathbf{U}_i^g$ between $\mathbf{X}_i^g$ and $\mathbf{V}_i^g$. The crossover operator performed on each component is

$$u_{ij}^g = \begin{cases} v_{ij}^g, & \text{if } \text{rand} \leq CR \text{ or } j = j_{\text{rand}}, j = 1, 2, \dots, D \\ x_{ij}^g, & \text{otherwise} \end{cases} \quad (11)$$

where CR is the crossover rate and j_{rand} is a randomly generated integer in $\{1, D\}$.

A selection operator then adopts a one-to-one competition between $\mathbf{U}_i^g$ and $\mathbf{X}_i^g$:

$$\mathbf{X}_i^{g+1} = \begin{cases} \mathbf{U}_i^g, & \text{if } \mathbf{U}_i^g \text{ is better than } \mathbf{X}_i^g \\ \mathbf{X}_i^g, & \text{otherwise} \end{cases} \quad (12)$$

DE repeats these three operators until a termination criterion is satisfied. Due to its simple structure, ease of use and good performance, several DE algorithms have been proposed to deal with DOPs, such as SDE [21], MDE [22], and TDE [1].

MDE is proposed to reduce the computational time of original DE and optimize DOPs [22]. It utilizes only one set of population compared with the two sets in the original DE. If a generated trial vector $\mathbf{U}_i^g$ is

better than its corresponding target vector $\mathbf{X}_f^g$, the former is immediately copied into the current population and participates in the mutation. Such an improvement enhances the convergence speed, with less function evaluations as compared to DE. The detailed implementation of MDE can be found in [22].

TDE was proposed by Fan and Lampinen [26], in which a new local search operation, i.e., trigonometric mutation operation (TMO), is embedded into the original DE. Angira and Santosh [1] dealt with the application and evaluation of TDE to solve DOPs. In TDE, the TMO is performed according to the following equation

$$\mathbf{V}_i^g = (\mathbf{X}_{r_1}^g + \mathbf{X}_{r_2}^g + \mathbf{X}_{r_3}^g)/3 + (p_2 - p_1)(\mathbf{X}_{r_1}^g - \mathbf{X}_{r_2}^g) + (p_3 - p_2)(\mathbf{X}_{r_2}^g - \mathbf{X}_{r_3}^g) + (p_1 - p_3)(\mathbf{X}_{r_3}^g - \mathbf{X}_{r_1}^g) \quad (13)$$

where $p_1 = |f(\mathbf{X}_{r_1}^g)|/p'$, $p_2 = |f(\mathbf{X}_{r_2}^g)|/p'$, $p_3 = |f(\mathbf{X}_{r_3}^g)|/p'$, and $p' = |f(\mathbf{X}_{r_1}^g)| + |f(\mathbf{X}_{r_2}^g)| + |f(\mathbf{X}_{r_3}^g)|$.

TDE performs the TMO with a probability Mt and performs the “DE/rand/1” using Eq. (10), with a probability $1-Mt$. The detailed procedure of TDE can be found in literature [1].

4. Ranking-based Differential Evolution Algorithms for DOPs

In this section, we present the ranking-based DE algorithms for constrained DOPs. First, we state the CVP approach. Next, the RMO technology is described. Finally, we integrate RMO into previous DE algorithms to form three ranking-based differential evolution (DE-RMO) algorithms for constrained DOPs.

4.1. CVP approach

The original DOP is an infinite-dimensional optimization problem, as the control vector $\mathbf{u}(t)$ to be optimized is a continuous function of time t . Therefore, CVP [8] is required to transform original problem into a finite-dimensional NLP problem. In the present work, time interval $[t_0, t_f]$ is divided into N stages and the i th time stage is $[t_{i-1}, t_i]$ ($i = 1, 2, \dots, N$). Control variables are approximated by constant functions in the i th stage $[t_{i-1}, t_i]$. Hence, the coding of optimization variables are expressed as

$$\mathbf{X} = \underbrace{[u_{11}, \dots, u_{1N}]}_{u_1}, \underbrace{[u_{21}, \dots, u_{2N}]}_{u_2}, \dots, \underbrace{[u_{m1}, \dots, u_{mN}]}_{u_m} \quad (14)$$

where u_{ji} ($j = 1, 2, \dots, m; i = 1, 2, \dots, N$) is the approximation of the j th control variable u_j in the i th time stage. After discretization, an optimization algorithm can be applied to select the optimal values of $\mathbf{X}$. During optimization, the differential equations [i.e., Eqs. (2) and (3)] must be solved using a differential equations solver for each evaluation of the objective function value (OFV).

There are four kinds of constraints in the DOPs: path inequality constraints, path equality constraints, terminal inequality constraints, and terminal equality constraints. To deal with the path inequality constraints (i.e. Eq. (4)) and path equality constraints (i.e. Eq. (5)), p additional variables $x_{n+1}, \dots, x_{n+l}, x_{n+l+1}, \dots, x_{n+p}$, called state constraint variables [27], are introduced using the following relationship

$$\dot{x}_{n+i}(t) = \max\{g_i^{\text{path}}(\mathbf{x}(t), \mathbf{u}(t), t), 0\}, i = 1, \dots, l \quad (15)$$

$$\dot{x}_{n+i}(t) = |h_i^{\text{path}}(\mathbf{x}(t), \mathbf{u}(t), t)|, i = l+1, \dots, p \quad (16)$$

with the initial condition expressed as

$$x_{n+i}(0) = 0, \quad i = 1, \dots, p. \quad (17)$$

The final value $x_{n+i}(t_f)$ gives the total violation of i th path constraint integrated over the entire time interval.

For terminal inequality constraints (i.e. Eq. (6)) and terminal equality constraints (i.e. Eq. (7)), the constraint violations can be calculated as

$$\max\{g_i^{\text{end}}(\mathbf{x}(t_f), \mathbf{u}(t_f), t_f), 0\}, i = 1, \dots, k \quad (18)$$

and

$$\max|h_i^{\text{end}}(\mathbf{x}(t_f), \mathbf{u}(t_f), t_f)|, i = k+1, \dots, k+q. \quad (19)$$

The overall constraint violation (OCV) is the sum of all path and terminal constraint violations, calculated as follows.

$$\text{OCV} = \underbrace{\sum_{i=1}^p x_{n+i}(t_f)}_{\text{path constraint violation}} + \underbrace{\sum_{i=1}^k \max\{g_i^{\text{end}}, 0\} + \sum_{i=k+1}^q |h_i^{\text{end}}|}_{\text{terminal constraint violation}}. \quad (20)$$

After obtaining the objective values and overall constraint violations of all individuals in the DE algorithms, Deb's feasibility-based rule [28] is used to compare two solutions.

Given two individuals X_1 and X_2 , their corresponding objective function values and overall constraint violations are $(\text{OFV}_1, \text{OCV}_1)$ and $(\text{OFV}_2, \text{OCV}_2)$. X_1 is said to be better than X_2 , if any of the following three conditions is met:

- (1) both X_1 and X_2 are infeasible, and $\text{OCV}_1 < \text{OCV}_2$;
- (2) X_1 is feasible, but X_2 is infeasible;
- (3) both X_1 and X_2 are feasible, and $\text{OFV}_1 < \text{OFV}_2$.

Compared with the penalty function constraint-handling method, Deb's feasibility-based rule does not introduce any sensitive parameter. Interested readers can read the survey paper of constraint-handling methods by Mezura-Montes and Coello [29].

4.2. Ranking-based mutation operator

Previous studies indicate that the DE algorithms may suffer from low convergence speed, because of the unbiased selection of parents in mutation operators [24]. Recently, Gong and Cai [25] introduced the ranking-based mutation operator into DE. In the RMO, better individuals have more chance to be utilized to produce offspring, which is helpful for the performance enhancement of DE algorithms.

This paper uses RMO to enhance the performance of DE to solve constrained DOPs. Therefore, the RMO technology is integrated into CVP-based DE algorithms. A feasibility-based ranking technique is used to sort the DE population from best to worst. The feasibility-based ranking technique is based on Deb's feasibility-based rule [28] described in subsection 4.1. It is described as follows:

- (1) feasible individuals are sorted before infeasible individuals;
- (2) feasible individuals are sorted on the basis of their OFVs;
- (3) infeasible individuals are sorted on the basis of their OCVs.

Gong et al. [30,31] have proposed some adaptive ranking mutation technique for constrained DE, but our feasibility-based ranking technique is much simpler compared with the adaptive ranking mutation technique, which may be more suitable for chemical DOPs.

To realize the feasibility-based ranking technique, we first calculate the objective function values and overall constraint violations of each

individual X_i by CVP approach. Then, a quick sort algorithm can be used to sort the whole population from best to worst.

Subsequently, the ranking R_i of i th vector is assigned as

$$R_i = NP - i, \quad i = 1, 2, \dots, NP \quad (21)$$

where NP is the population size. According to Eq. (21), the best vector in the current population will give the highest ranking.

The selection probability of i th individual is calculated as

$$p_i = \frac{R_i}{NP}, \quad i = 1, \dots, NP. \quad (22)$$

The individuals in the RMO are finally selected according to the selection probabilities. The ranking-based individual selection in RMO is illustrated in Fig. 1. The individuals with higher rankings (or selection probabilities) are more likely to be chosen in the mutation operator. This is beneficial for the performance enhancement of DE algorithms.

Input: The target vector index i

Output: The selected vector indexes r_1, r_2 , and r_3 in mutation “DE/rand/1”.

```

1: Randomly select  $r_1 \in \{1, NP\}$  {base vector index}
2: While  $\text{rand} > p_{r_1}$  or  $r_1 = i$ 
3:   Randomly select  $r_1 \in \{1, NP\}$ 
4: End
5: Randomly select  $r_2 \in \{1, NP\}$  {terminal vector index}
6: While  $\text{rand} > p_{r_2}$  or  $r_2 = r_1$  or  $r_2 = i$ 
7:   Randomly select  $r_2 \in \{1, NP\}$ 
8: End
9: Randomly select  $r_3 \in \{1, NP\}$  {starting vector index}
10: While  $r_3 = r_2$  or  $r_3 = r_1$  or  $r_3 = i$ 
11:   Randomly select  $r_3 \in \{1, NP\}$ 
12: End

```

Fig. 1. Ranking-based individual selection in RMO.

4.3. Implementations of DE-RMO algorithms

In this work, the RMO is incorporated into SDE [21], MDE [22], and TDE [1] to form three DE-RMO algorithms, namely SDE-RMO, MDE-RMO, and TDE-RMO, to solve constrained DOPs. We choose these three DE algorithms because all of them can be realized easily, which is beneficial for chemical DOPs. Here we only describe TDE-RMO in detail. The main procedure is given as follows.

Step 1: Initial a population with NP individuals. Set scale factor F , crossover rate CR , and trigonometric mutation probability Mt .

Step 2: Calculate the OFV and OCV for each individual using the CVP approach. The explicit Runge–Kutta method is used to solve the differential equations.

Step 3: Sort the population according to the feasibility-based ranking technique.

Step 4: For each individual, do Steps 5–8.

Step 5: If $\text{rand} < Mt$, perform the TMO according to Eq. (13); else, perform the RMO according to Fig. 1 and Eq. (10).

Step 6: Perform the crossover operator according to Eq. (15).

Step 7: Calculate the OFV and OCV for each individual using the CVP approach.

Step 8: Perform the selection operator Eq. (11) based on Deb's feasibility-based rule.

Step 9: Return to Step 3 until the termination condition is met.

Step 10: Output the best solutions.

Fig. 2. illustrates the flowchart of TDE and TDE-RMO for DOPs. From Fig. 2, we can see that the differences between TDE and TDE-RMO are: (1) TDE-RMO needs to sort the DE population before the mutation operators; (2) TDE-RMO employs RMO, while TDE employs the classic “DE/rand/1”. However, both modifications can be easily realized. Therefore, TDE-RMO does not significantly increase the complexity of TDE algorithm.

5. Case Studies

In this section, the DE-RMO algorithms are applied to four constrained DOPs by CVP. First, we state the simulation setup for this study. Subsequently, simulations are conducted to compare the DE-RMO algorithms with their non-ranking DE algorithms. Finally, TDE-RMO is compared with some other stochastic optimization algorithms.

5.1. Simulation setup

Four constrained DOPs from the literature are used to evaluate the performance of the DE-RMO algorithms. The parameter settings of these DE algorithms are in Table 1.

Three evaluation criteria are adopted to measure the performance of the algorithms.

BOFV (best objective function value): it records the best objective function values when the maximal number of function evaluation $maxNFES$ is reached. The best, mean, and worst BOFV, as well as the standard deviations (std), are presented. BOFV measures the solution accuracy of an algorithm.

SR (success rate): It is equal to the number of successful runs over total runs. A success run means that within $maxNFES$, the algorithm finds a solution with satisfactory precision J_s .

ANFES: It is used to record the average number of function evaluations for an algorithm to find a solution with satisfactory precision J_s . ANFES measures the convergence speed of an algorithm.

5.2. Simulation results

The DE-RMO algorithms and their non-ranking DE algorithms are used to solve four constrained DOPs. All algorithms are coded in matlab. The built-in routine “ode45” is chosen as differential equation solver.

5.2.1. Problem 1 — constrained van der Pol oscillator

This problem is extracted from a previous study [32] and described as follows.

$$\begin{aligned} \min \quad & J = x_3(t_f) \\ \text{s.t.} \quad & \begin{cases} \dot{x}_1 = (1 - x_2^2)x_1 - x_2 + u \\ \dot{x}_2 = x_1 \\ \dot{x}_3 = x_1^2 + x_2^2 + u^2 \end{cases} \\ & x(0) = [0, 1, 1], -0.3 \leq u \leq 1.0, t_f = 5 \end{aligned} \quad (23)$$

The following path constraint is imposed on $x_1(t)$, i.e.,

$$x_1(t) \geq -0.4. \quad (24)$$

For problem 1, the time interval is divided into $N = 20$ stages in CVP approach. BOFV is recorded when $maxNFES$ reaches 30000, and the satisfactory solution precision is set as $J_s = 2.97500$. The values of $maxNFES$ and J_s are set based on the numeric experiments, because we cannot know the best objective function values in advance for a real-world DOP.

Table 2 shows the results of these DE algorithms. The boldface means that DE-RMO algorithms are better than their corresponding non-ranking DE algorithms. Thus our proposed DE-RMO algorithms

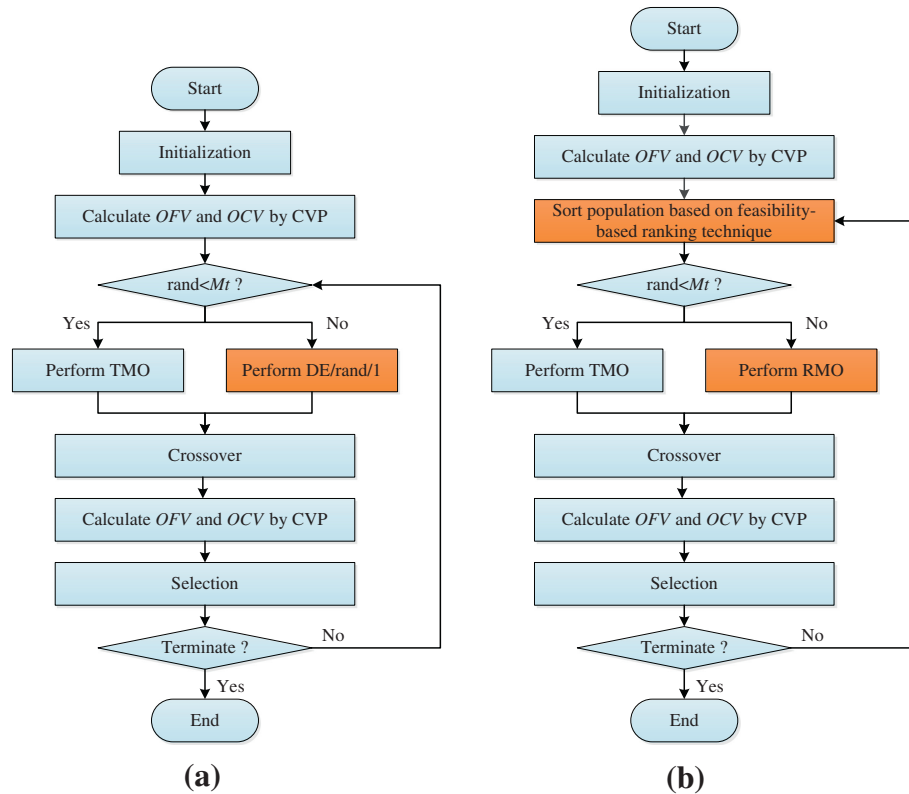


Fig. 2. Flowcharts (a) TDE for DOPs (b) TDE-RMO for DOPs.

provide consistently more accurate solutions than their corresponding non-ranking DE algorithms with respect to BOFV. Considering the std. of BOFV, we find that all of the DE-RMO algorithms give smaller std. values than their corresponding non-ranking DE algorithms. Thus the RMO is able to enhance the robustness of the previous DE algorithms. All three DE-RMO algorithms succeed in finding satisfactory solutions for each of 20 runs. In terms of ANFES, DE-RMO algorithms get less ANFES compared with their corresponding non-ranking DE algorithms, so that DE-RMO algorithms converge faster. Considering the overall performance, TDE-RMO ranks the first, followed by MDE-RMO, TDE, SDE-RMO, MDE, and SDE.

The reported best result 2.95436 is obtained by Vassiliadis *et al.* [32], with a gradient-based algorithm. Our best result 2.97228 is within 0.61%

Table 1
Parameter settings for the DE algorithms

Algorithm	Parameter settings
SDE, SDE-RMO	$NP = 5D$, $F = 0.5$, $CR = 0.9$;
MDE, MDE-RMO	$NP = 5D$, $F = 0.5$, $CR = 0.9$;
TDE, TDE-RMO	$NP = 5D$, $F = 0.5$, $CR = 0.9$, $Mt = 0.1$

Table 2
Numerical results for problem 1

Algorithm	Best	Mean	Worst	Std	SR	ANFES	Rank
SDE	2.97381	2.97508	2.97713	7.94×10^{-4}	9/20	^①	6
SDE-RMO	2.97231	2.97233	2.97238	1.82×10^{-5}	20/20	15500	4
MDE	2.97340	2.97390	2.97544	4.64×10^{-4}	19/20	28331	5
MDE-RMO	2.97229	2.97230	2.97235	1.23×10^{-5}	20/20	13450	2
TDE	2.97229	2.97255	2.97347	3.43×10^{-4}	20/20	15425	3
TDE-RMO	2.97228 ^②	2.97228	2.97232	8.67×10^{-6}	20/20	9105	1

Boldface mean the results of DE-RMO algorithms are better than their non-ranking DE algorithms.

^① ANFES is not recorded because of the low SR value

^② Means the best value obtained in this work.

of the reported best value. Fig. 3 illustrates the optimal control profile and the path constraint trajectory obtained by the TDE-RMO. The optimal control profile is in agreement with that in [32]. The path constraint $x_1(t) \geq -0.4$ is satisfied, and it is active in the interval [0.494, 1.650].

5.2.2. Problem 2 – mathematical system with nonlinear inequality constraint

The mathematical model of this problem is presented as [27]

$$\begin{aligned} \min J &= x_3(t_f) \\ \text{s.t.} \quad &\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_2 + u \\ \dot{x}_3 = x_1^2 + x_2^2 + 0.005u^2 \\ \mathbf{x}(0) = [0, -1, 0], -20 \leq u \leq 20, t_f = 1 \end{cases} \end{aligned} \quad (25)$$

The system should satisfy an inequality path constraint:

$$h(x, t) = x_2 - 8(t - 0.5)^2 + 0.5 \leq 0. \quad (26)$$

For problem 2, the time interval is divided into $N = 20$ stages, and other parameters are: $D = 20$, $\max NFES = 60000$, and $J_s = 0.17500$. Table 3 shows the results of these DE algorithms. The DE-RMO algorithms consistently give better results than their corresponding non-ranking DE algorithms in terms of BOFV, SR and ANFES. Hence, the RMO is able to enhance the solution accuracy, success rate, and convergence speed of DE algorithms. TDE-RMO gives the best performance in this problem, followed by MDE-RMO, TDE, SDE-RMO, MDE, and SDE. Mekarapiruk and Luus [27] divided the interval into $N = 20$ stages and achieved a value of 0.17266 with the iterative dynamic programming algorithm. Our best result 0.17272 is very close to their result. The optimal control profile and the path constraint trajectory obtained by the TDE-RMO are plotted in Fig. 4. The optimal control profile is in agreement with that in [27], and the path constraint $h(x, t) \leq 0$ is active in the interval [0.290, 0.706].

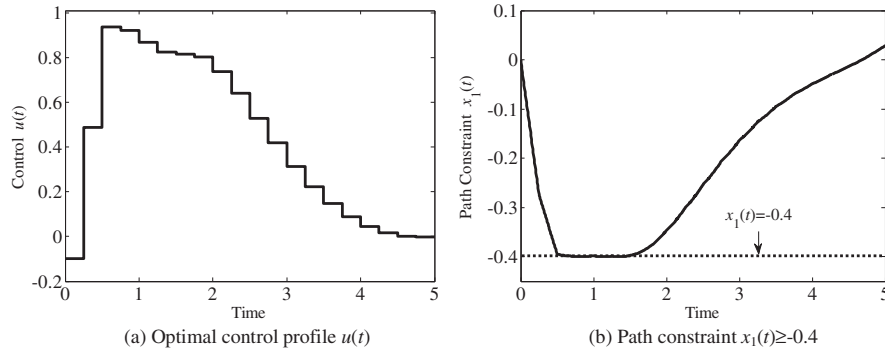


Fig. 3. The optimal control profile and the path constraint trajectory for problem 1.

Table 3
Numerical results for problem 2

Algorithm	Best	Mean	Worst	Std	SR	ANFES	Rank
SDE	0.17431	0.17546	0.17762	8.09×10^{-4}	7/20	\ ^①	6
SDE-RMO	0.17276	0.17284	0.17309	8.33×10^{-5}	20/20	32140	4
MDE	0.17376	0.17453	0.17596	5.71×10^{-4}	15/20	75500	5
MDE-RMO	0.17274	0.17279	0.17288	4.42×10^{-5}	20/20	29400	2
TDE	0.17273	0.17334	0.17469	5.74×10^{-4}	20/20	29760	3
TDE-RMO	0.17272 ^②	0.17281	0.17305	7.22×10^{-5}	20/20	17550	1

Boldface mean the results of DE-RMO algorithms are better than their non-ranking DE algorithms.

^① ANFES is not recorded because of the low SR value.

^② Means the best value obtained in this work.

5.2.3. Problem 3 – optimal operation for a batch reactor with a cooling jacket

This problem formulates a first-order consecutive exothermic reaction, $A \rightarrow P \rightarrow S$, occurring in a batch reactor with a cooling jacket [33]. The objective is to determine the optimal coolant flow rate u so that product P is maximized. The problem is described as

$$\begin{aligned}
 \max \quad & J = x_2(t_f) \\
 \text{s.t.} \quad & \begin{cases} \dot{x}_1 = -k_1 x_1 \\ \dot{x}_2 = k_1 x_1 - k_2 x_2 \\ \dot{x}_3 = k_2 x_2 \\ \dot{x}_4 = 193.4524k_1 x_1 + 35.7143k_2 x_2 - 8.8923(x_4 - x_5) \\ \dot{x}_5 = 33.1978(x_4 - x_5) - 38.7940(x_5 - x_6) \\ \dot{x}_6 = \frac{u}{0.53}(298 - x_6) + 19.2925(x_5 - x_6) \\ \mathbf{x}(0) = [0.975, 0.025, 0, 350, 373, 300] \\ 0 \leq u \leq 9, t_f = 3.5 \\ k_1 = 4.38 \times 10^4 \exp\left(\frac{-3.49 \times 10^7}{8314x_4}\right) \\ k_2 = 3.94 \times 10^5 \exp\left(\frac{-4.65 \times 10^7}{8314x_4}\right) \end{cases} \quad (27)
 \end{aligned}$$

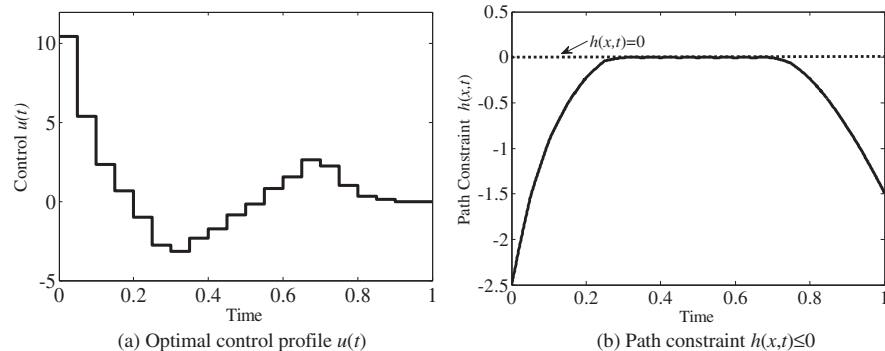


Fig. 4. The optimal control profile and the path constraint trajectory for problem 2.

where x_1 , x_2 , and x_3 denote the concentrations of A, P and S, respectively; x_4 , x_5 , and x_6 are the temperatures (in Kelvin) of contents, C, and jacket, respectively; u is the coolant flow rate ($\text{m}^3 \cdot \text{h}^{-1}$); k_1 and k_2 are the reaction rate constants. The first three differential equations describe the material balances, and other three differential equations describe the energy balances.

Two constraints are path and terminal constraints.

$$\begin{aligned}
 x_4(t) &\leq 370\text{K} \\
 x_4(t_f) &\leq 320\text{K}
 \end{aligned} \quad (28)$$

For problem 3, the time interval is divided into $N = 20$ stages, and the other parameters are: $D = 20$, $\text{maxNFES} = 60000$, and $J_s = 0.64500$. The results of these DE algorithms are presented in Table 4. This problem is more difficult than problems 1 and 2, as DE and MDE cannot find satisfactory solutions within maxNFES . By contrast, all three DE-RMO algorithms give SR of 20/20. Therefore, DE-RMO algorithms are more reliable than their non-ranking DE algorithms. Overall, the results indicate that our DE-RMO algorithms perform better than their corresponding non-ranking DE algorithms in terms of BOFV, SR and ANFES. For this problem,

Table 4
Numerical results for problem 3

Algorithm	Best	Mean	Worst	Std	SR	ANFES	Rank
SDE	0.64490	0.64443	0.64366	3.29×10^{-4}	0/20	\ ^①	6
SDE-RMO	0.64580	0.64564	0.64521	1.35×10^{-4}	20/20	37425	3
MDE	0.64502	0.64456	0.64391	2.65×10^{-4}	1/20	\ ^①	5
MDE-RMO	0.64580	0.64570	0.64546	9.08×10^{-5}	20/20	33805	2
TDE	0.64569	0.64532	0.64468	2.77×10^{-4}	18/20	50816	4
TDE-RMO	0.64586 ^②	0.64569	0.64535	1.56×10^{-4}	20/20	29125	1

Boldface mean the results of DE-RMO algorithms are better than their non-ranking DE algorithms.

^① ANFES is not recorded because of the low SR value.

^② Means the best value obtained in this work.

TDE-RMO still ranks the first, followed by MDE-RMO, SDE-RMO, TDE, MDE, and SDE.

Sun *et al.* [33] acquired a value of 0.6446, with the line-up competition algorithm. Our best result is 0.64586, better than that in [33]. The optimal control profile and the constraint trajectory obtained by the TDE-RMO are plotted in Fig. 5. The path constraint $x_4(t) \leq 370\text{K}$ is active in the interval [1.151, 2.848], and terminal constraint $x_4(t_f) \leq 320\text{K}$ is active at the final time.

5.2.4. Problem 4 – optimal monoclonal antibody production

This problem considers the optimal monoclonal antibody production in a hybridoma fed-batch reactor [34]. The problem is described as

$$\begin{aligned} \max J &= \int_0^{t_f} L dt, L = q_{MAB} X_v(t) V(t) \\ \text{s.t.} \quad &\begin{cases} \frac{dX_v}{dt} = (\mu - k_d) X_v - \frac{(F_1 + F_2)}{V} X_v \\ \frac{dGlc}{dt} = \frac{F_1}{V} Glc_{in} - \frac{(F_1 + F_2)}{V} Glc - q_{Glc} X_v \\ \frac{dGln}{dt} = \frac{F_2}{V} Gln_{in} - \frac{(F_1 + F_2)}{V} Gln - q_{Gln} X_v \\ \frac{dLac}{dt} = q_{Lac} X_v - \frac{(F_1 + F_2)}{V} Lac \\ \frac{dAmm}{dt} = q_{Amm} X_v - \frac{(F_1 + F_2)}{V} Amm \\ \frac{dMAB}{dt} = q_{MAB} X_v - \frac{(F_1 + F_2)}{V} MAB \\ \frac{dV}{dt} = F_1 + F_2 \end{cases} \\ &x(0) = [2 \times 10^8 \quad 25 \quad 4 \quad 0 \quad 0 \quad 0.8], 0.0 \leq F_1 \leq 0.5, 0.0 \leq F_2 \leq 0.5, t_f = 10 \end{aligned} \quad (29)$$

where X_v , Glc , Gln , Lac , Amm , and MAB are the concentrations in viable cells, glucose, glutamine, lactate, ammonia, and monoclonal antibodies, respectively; V is the fermenter volume; Glc_m and Gln_m are the concentrations of glucose and glutamine in the feed stream, respectively; the control variables F_1 and F_2 are the volumetric feed rate of glucose and glutamine, respectively. The mathematical expressions of the specific rates μ , k_d , q_{Gln} , q_{Glc} , q_{Lac} , q_{Amm} , q_{MAB} , and the system parameters are provided in [34].

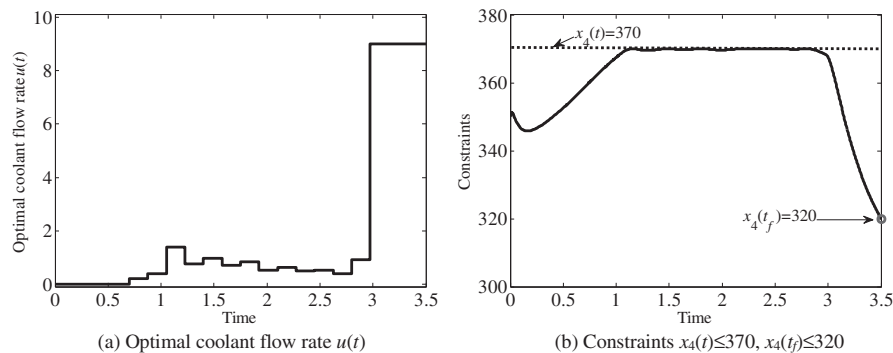


Fig. 5. The optimal coolant flow rate and the constraint trajectory for problem 3.

Table 5
Numerical results for problem 4

Algorithm	Best	Mean	Worst	Std	SR	ANFES	Rank
SDE	331.65980	330.46431	328.82462	9.54×10^{-1}	0/20	^①	6
SDE-RMO	334.02530	333.67352	333.28913	2.24×10^{-1}	15/20	178800	4
MDE	332.05848	330.83603	329.81007	5.25×10^{-1}	0/20	^①	5
MDE-RMO	334.07852	333.84554	333.48348	1.76×10^{-1}	18/20	139900	3
TDE	334.31220	333.95562	333.03898	3.36×10^{-1}	18/20	128366	2
TDE-RMO	334.63338^②	334.55889	334.20336	9.05×10^{-2}	20/20	64010	1

Boldface mean the results of DE-RMO algorithms are better than their non-ranking DE algorithms.

^① ANFES is not recorded because of the low SR value.

^② Means the best value obtained in this work.

In this problem, the value of $V(t)$ is constrained by

$$V(t) \leq V_{\max}, V_{\max} = 2. \quad (30)$$

For this problem, the time interval is divided into $N = 20$ stages, and the other parameters are: $D = 20$, $\max NFES = 150000$, and $J_s = 333.50000$. Table 5 presents the results of the six DE algorithms. Our DE-RMO algorithms perform better than their non-ranking DE algorithms. The optimal control profile and the constraint trajectory are plotted in Fig. 6. The path constraint is active at the final time.

According to the simulation results, it can be concluded that:

- DE-RMO algorithms perform better than the non-ranking DE algorithms in terms of solution accuracy, success rate, and convergence speed for DOPs. The better performance of DE-RMO algorithms should be attributed to the introduction of RMO.
- The path constraints and terminal constraints are active at the optimal solutions in the four DOPs.
- TDE-RMO provides the best performance among all the used DE algorithms. It is suggested to be used as an efficient optimizer to solve DOPs in the future.

5.3. Further comparison with other evolutionary algorithms

In the previous subsection, the performance of DE-RMO algorithms verified by solving four DOPs. TDE-RMO provides the best results among three DE-RMO algorithms. To provide additional comparison for reference, TDE-RMO is compared with four other evolutionary algorithms, adaptive particle swarm optimization (APSO) [35], real-code genetic algorithm (RCGA) [36], artificial bee colony (ABC) [37], and teaching-learning-based optimization (TLBO) [38]. The parameters of these four algorithms are set as recommended in their original literature. All the algorithms use the same constraint-handling method as that in TDE-RMO. Table 6 shows TDE-RMO is better than these four algorithms on all problems. BOFV of TDE-RMO are more accurate and better than those of other algorithms. TDE-RMO also provides the

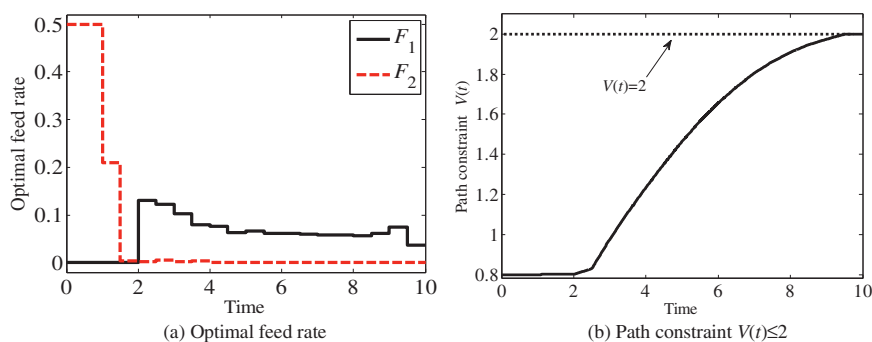


Fig. 6. The optimal coolant flow rate and the path constraint trajectory for problem 4.

Table 6

Comparison between TDE-RMO and other evolutionary algorithms

	Algorithm	Best	Mean	Worst	Std	ANFES	SR	Rank
Problem 1 (min)	TDE-RMO	2.97228	2.97228	2.97232	8.67×10^{-6}	9105	20/20	1
	APSO	2.97286	2.97445	2.97728	1.36×10^{-3}	22986	14/20	3
	RCGA	2.9882	3.01574	3.11665	2.82×10^{-2}	$\sqrt{\textcircled{1}}$	0/20	4
	ABC	3.00186	3.05708	3.14474	4.28×10^{-2}	$\sqrt{\textcircled{1}}$	0/20	5
	TLBO	2.97241	2.97361	2.9757	9.33×10^{-4}	19984	18/20	2
Problem 2 (min)	TDE-RMO	0.17272	0.17281	0.17305	7.22×10^{-5}	17550	20/20	1
	APSO	0.17454	0.18093	0.21948	1.51×10^{-2}	$\sqrt{\textcircled{1}}$	4/20	3
	RCGA	0.22994	0.28288	0.32948	2.42×10^{-2}	$\sqrt{\textcircled{1}}$	0/20	4
	ABC	0.43229	0.59199	0.77423	9.90×10^{-2}	$\sqrt{\textcircled{1}}$	0/20	5
	TLBO	0.17327	0.17424	0.17655	1.06×10^{-3}	38824	16/20	2
Problem 3 (max)	TDE-RMO	0.64586	0.64569	0.64535	1.56×10^{-4}	29125	20/20	1
	APSO	0.64335	0.62989	0.5964	1.55×10^{-2}	$\sqrt{\textcircled{1}}$	0/20	4
	RCGA	0.63917	0.63663	0.63464	1.30×10^{-3}	$\sqrt{\textcircled{1}}$	0/20	3
	ABC	0.59538	0.58408	0.56226	9.19×10^{-3}	$\sqrt{\textcircled{1}}$	0/20	5
	TLBO	0.64533	0.64389	0.64058	1.55×10^{-3}	$\sqrt{\textcircled{1}}$	4/20	2
Problem 4 (max)	TDE-RMO	334.63338	334.55889	334.20336	9.05×10^{-2}	64010	20/20	1
	APSO	329.25381	321.37311	281.31247	11.1	$\sqrt{\textcircled{1}}$	0/20	4
	RCGA	325.14648	323.41749	321.47966	1.19	$\sqrt{\textcircled{1}}$	0/20	3
	ABC	300.78203	295.12948	283.72424	4.82	$\sqrt{\textcircled{1}}$	0/20	5
	TLBO	334.60868	334.3452	333.67363	3.01×10^{-1}	58268	20/20	2

Boldface mean the best results among TDE-RMO, APSO, RCGA, ABC, and TLBO.

 $\sqrt{\textcircled{1}}$ ANFES is not recorded because of the low SR value.

smallest std. of BOFV over 20 run. Considering the SR, TDE-RMO is the most reliable. In terms of ANFES, TDE-RMO gets less ANFES compared with their corresponding non-ranking DE algorithms, which means that DE-RMO algorithms converge faster. The ranks of five algorithms for the four DOPs are TDE-RMO, TLBO, APSO, RCGA, and ABC.

6. Conclusions

In this study, the RMO technology has been proposed to enhance the DE algorithms to solve DOPs. Three DE-RMO algorithms have been designed by incorporating the RMO. The DE-RMO algorithms and their non-ranking DE algorithms are evaluated by solving four constrained DOPs from the literature. The simulation results demonstrate that the DE-RMO algorithms perform better than previous non-ranking DE algorithms in terms of solution accuracy, success rate and convergence speed. Hence, the DE-RMO algorithms can be used as promising alternatives to solve DOPs in the future.

References

- [1] R. Angira, A. Santosh, Optimization of dynamic systems: A trigonometric differential evolution approach, *Comput. Chem. Eng.* 31 (9) (2007) 1055–1063.
- [2] R. Bellman, R.E. Kalaba, *Dynamic Programming and Modern Control Theory*, Academic Press, New York, 1965.
- [3] R. Luus, Optimal control by dynamic programming using systematic reduction in grid size, *Int. J. Control.* 51 (5) (1990) 995–1013.
- [4] J.E. Jiménez-Hornero, I.M. Santos-Dueñas, I. García-García, Optimization of biotechnological processes. The acetic acid fermentation. Part III: Dynamic optimization, *Biochem. Eng. J.* 45 (1) (2009) 22–29.
- [5] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, E. Mishchenko, *The Mathematical Theory of Optimal Processes*, Interscience, New York, 1962.
- [6] J. Lee, W.F. Ramirez, Optimal fed-batch control of induced foreign protein production by recombinant bacteria, *AIChE J.* 40 (5) (1994) 899–907.
- [7] S. Kameswaran, L.T. Biegler, Simultaneous dynamic optimization strategies: Recent advances and challenges, *Comput. Chem. Eng.* 30 (10) (2006) 1560–1575.
- [8] E. Balsa-Canto, J.R. Banga, A.A. Alonso, V.S. Vassiliadis, Restricted second order information for the solution of optimal control problems using control vector parameterization, *J. Process Control* 12 (2) (2002) 243–255.
- [9] J.R. Banga, E. Balsa-Canto, C.G. Moles, A.A. Alonso, Dynamic optimization of bioprocesses: Efficient and robust numerical strategies, *J. Biotechnol.* 117 (4) (2005) 407–419.
- [10] D. Sarkar, M.J. Modak, Optimisation of fed-batch bioreactors using genetic algorithms, *Chem. Eng. Sci.* 58 (11) (2003) 2283–2296.
- [11] A.R. Mandli, J.M. Modak, Evolutionary algorithm for the determination of optimal mode of bioreactor operation, *Ind. Eng. Chem. Res.* 51 (4) (2011) 1796–1808.
- [12] Q.T. Pham, Using fuzzy logic to tune an evolutionary algorithm for dynamic optimization of chemical processes, *Comput. Chem. Eng.* 37 (2012) 136–142.
- [13] R. Faber, T. Jockenhövel, G. Tsatsaronis, Dynamic optimization with simulated annealing, *Comput. Chem. Eng.* 29 (2) (2005) 273–290.
- [14] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, Multicanonical jump walk annealing assisted by tabu for dynamic optimization of chemical engineering processes, *Eur. J. Oper. Res.* 185 (3) (2008) 1213–1229.
- [15] X. Chen, W. Du, R. Qi, F. Qian, H. Tianfield, Hybrid gradient particle swarm optimization for dynamic optimization problems of chemical processes, *Asia Pac. J. Chem. Eng.* 8 (5) (2013) 708–720.

- [16] X. Chen, W. Du, H. Tianfield, R. Qi, W. He, F. Qian, Dynamic optimization of industrial processes with nonuniform discretization-based control vector parameterization, *IEEE Trans. Autom. Sci. Eng.* 11 (4) (2014) 1289–1299.
- [17] Y. Zhou, X. Liu, Control parameterization-based adaptive particle swarm approach for solving chemical dynamic optimization problems[J], *Chem. Eng. Technol.* 37 (4) (2014) 692–702.
- [18] J.A. Egea, E. Balsa-Canto, M.S.G. Garcia, J.R. Banga, Dynamic optimization of nonlinear processes with an enhanced scatter search method, *Ind. Eng. Chem. Res.* 48 (9) (2009) 4388–4401.
- [19] R. Storn, K. Price, Differential evolution — a simple and efficient adaptive scheme for global optimization over continuous spaces, A Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, March 1995.
- [20] J.P. Chiou, F.S. Wang, Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process, *Comput. Chem. Eng.* 23 (9) (1999) 1277–1291.
- [21] M.D. Kapadi, R.D. Gudi, Optimal control of fed-batch fermentation involving multiple feeds using differential evolution, *Process Biochem.* 39 (11) (2004) 1709–1721.
- [22] B.V. Babu, R. Angira, Modified differential evolution (MDE) for optimization of nonlinear chemical processes, *Comput. Chem. Eng.* 30 (6) (2006) 989–1002.
- [23] Q.Q. Fan, Z.M. Lv, X.F. Yan, M.J. Guo, Chemical process dynamic optimization based on hybrid differential evolution algorithm integrated with Alopex, *J. Cent. South Univ.* 20 (2013) 950–959.
- [24] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evol. Comput.* 15 (1) (2011) 4–31.
- [25] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, *IEEE Trans. Cybern.* 43 (6) (2013) 2066–2081.
- [26] H.Y. Fan, J. Lampinen, A trigonometric mutation operation to differential evolution, *J. Glob. Optim.* 27 (1) (2003) 105–129.
- [27] W. Mekarapiruk, R. Luus, Optimal control of inequality state constrained systems, *Ind. Eng. Chem. Res.* 36 (5) (1997) 1686–1694.
- [28] K. Deb, An efficient constraint handling method for genetic algorithms, *Comput. Methods Appl. Mech. Eng.* 186 (2) (2000) 311–338.
- [29] E. Mezura-Montes, C.A.C. Coello, Constraint-handling in nature-inspired numerical optimization: past, present and future, *Swarm Evol. Comput.* 1 (4) (2011) 173–194.
- [30] W. Gong, Z. Cai, D. Liang, Engineering optimization by means of an improved constrained differential evolution, *Comput. Methods Appl. Mech. Eng.* 268 (2014) 884–904.
- [31] W. Gong, Z. Cai, D. Liang, Adaptive ranking mutation operator based differential evolution for constrained optimization, *IEEE Trans. Cybern.* 45 (4) (2015) 716–727.
- [32] V.S. Vassiliadis, R.W.H. Sargent, C.C. Pantelides, Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints, *Ind. Eng. Chem. Res.* 33 (9) (1994) 2123–2133.
- [33] D.Y. Sun, P.M. Lin, S.P. Lin, Integrating controlled random search into the line-up competition algorithm to solve unsteady operation problems, *Ind. Eng. Chem. Res.* 47 (22) (2008) 8869–8887.
- [34] J.A. Roubos, G. Van Straten, A.J.B. Van Boxtel, An evolutionary strategy for fed-batch bioreactor optimization; concepts and performance, *J. Biotechnol.* 67 (2) (1999) 173–187.
- [35] Z.H. Zhan, J. Zhang, Y. Li, H.H. Chung, Adaptive particle swarm optimization, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 39 (6) (2009) 1362–1381.
- [36] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, *Complex Syst.* 9 (3) (1994) 1–15.
- [37] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *J. Glob. Optim.* 39 (3) (2007) 459–471.
- [38] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems, *Inf. Sci.* 183 (1) (2012) 1–15.