Applied Mathematics and Computation 216 (2010) 3343–3354

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Apply a novel evolutionary algorithm to the solution of parameter selection problems

Daim-Yuang Sun *, Sheng-Chung Ni, Tzu-Chen Huang

Department of Chemical and Materials Engineering, National Chin-Yi University of Technology, Taichung County, 411 Taiwan, Republic of China

ARTICLE INFO

Keywords:
Parameter selection
Control parametrization
Nonlinear programming
Evolutionary algorithms
Line-up competition algorithm

ABSTRACT

In this study, a modified line-up competition algorithm (LCA) is used to solve parameter selection problems. The so-called parameter selection problems contain parameter identification problems and optimal control problems. Once the later problems are transformed by control parametrization, the parameters embedded in both problems are selected by the proposed method under the framework of integration approach. Two parameter identification problems and one optimal control problem are given to demonstrate the use of LCA. The results show that in addition to being insensitive to the initial conditions, LCA is very efficient in solving highly nonlinear parameter selection problems.

1. Introduction

In many fields of science and engineering, researchers use models to explore interaction between physical system and its surrounding. The so-called models are a set of differential and/or algebraic equations based on first principles. In order to properly demonstrate the dynamic and static behaviors of the described systems, the models require not only suitable mathematical formulations but also a set of correct parameters. Notably, some of these parameters can be referred to public or commercial data bank, while some of them need to be determined from experiments or measurements. Determining unknown parameters in a mathematical model via the input–output data generated by the described system is entitled parameter identification (or parameter estimation). With similar procedure to the above, curve-fitting is to determine the best coefficients in a presumed function so that the resultant curve may best fit a series of given data. In solving optimal control problems (OCPs), on the other hand, transformation techniques [1,2] are to approximate state and/or control variables by some trial function with unspecific parameters. The OCPs are thus converted into nonlinear programming problems with the decision variables derived from the parameters. Obviously, all the above problems are eventually converted into the problems of selecting a set of feasible parameters that may optimize (minimize or maximize) a predefined objective function. However, owing to the highly nonlinear, multi-modal and non-convex properties embedded in the problems, to acquire a set of global parameters is still a challenging task.

With very fast convergent rate, gradient-based methods have been applied in the selection of unknown parameters very successfully. However, these methods are very sensitive to the initial guesses, and thus often assailed by local convergence especially for non-convex and multi-modal systems. Contrasting to the above methods, evolutionary algorithms (EAs), including genetic algorithms, evolutionary strategies and evolutionary programming, directly generate decision vector via some specially-designed mechanism inspired by biological world. The basic differences between these EAs are that genetic algorithms focus on gene operations, whereas evolutionary strategies and evolutionary programming both emphasize the behavior change of individual. Unfortunately, all of them are difficult to maintain population diversity, and to balance local

* Corresponding author.
E-mail address: dysun@mail.ncut.edu.tw (D.-Y. Sun).

0096-3003/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2010.04.062

and global searches [3]. Additionally, they also face the problem of premature convergence. Thus, the EAs are still possible to acquire local solutions, even the numerical efforts are paid very considerably. To overcome the above drawbacks, a brand-new EA, entitled line-up competition algorithm (LCA) [3], is developed based on the concept of cooperation and competition among biological population. Successfully applied to different types of static and dynamic problems [4–8], LCA have been proved to be very advantageous over other evolutionary algorithms. In this study, LCA is applied to solve parameter selection (PS) problems under the framework of integration approach.

In the rest of this article, An brief overview for LCA is introduced in Section 2. The detailed computational steps are also included in the same section. The general descriptions for PS problems are presented in Section 3. The numerical illustrations are given in Section 4. The conclusions are made in the last section.

2. Line-up competition algorithm

The basic line-up competition algorithm is originally developed to minimize the following objective function:

$$\min_{\Phi} \quad \mathcal{I} \tag{1a}$$

subject to

$$\begin{aligned} &\mathbf{h}(\Phi) = 0, \\ &\mathbf{g}(\Phi) \leqslant 0, \\ &\mathbf{L}^0 \leqslant \Phi \leqslant \mathbf{U}^0, \end{aligned} \tag{1b}$$

where $\Phi \equiv [\varphi_1, \cdots, \varphi_{N_c}]'$ denotes the decision vector, which is bounded by the lower and upper bounds, $\mathbf{L}^0 \equiv [L_1^0, \ldots, L_c^0]'$ and $\mathbf{U}^0 \equiv [U_1^0, \ldots, U_c^0]'$. $\mathbf{h}$ and $\mathbf{g}$ are equality and inequality constraints.

To minimize Eq. (1), LCA uniformly generates N_f trial vectors, named *fathers*, over the entire searching space. Based on the corresponding objective values, these fathers are then queued in sequence, named *line-up*. Therein, the father with the best objective value is placed at the first position of this sequence, while the poorest is at the last. The first generation is completed so far. Obviously, the line-up is raked in ascending order for a minimization problem, or in descending order for a maximization problem. To promote the overall convergent rate, the size of search space is shrunk by a heuristic factor with a value smaller than 1.0, and is systematically partitioned into N_f sub-regions (or sub-spaces). Then, each father is given a sub-region to produce N_m trial vectors, named "*offsprings*". The reduced search space is now occupied by N_f *families*, each consisting $N_m + 1$ *members*. Fig. 1 shows a situation that a two-dimensional search space is occupied by four families $N_f = 4$, each having five members $N_m = 5$. Notably, the strategy of allocating sub-regions is to give a better father a smaller one. As a result, the father at the first position of the line-up acquires the smallest sub-region, and the father at the last position acquires the biggest one. The purpose behind such strategy is to give the better families more chances to refine objective values, and to give the poorer families more chances to escape from local solutions. Additionally, even having gathered around a local region during the searches, some father can be easily re-distributed under such allocation strategy. Thus, the local convergence can be effectively surpassed. After the above allocations, the best solution in each family is selected in order to compete with the new fathers from other families for the prior position in new line-up. The steps including *partition*, *selection* and *sequencing* never stop until the predefined generations. From the above descriptions, LCA intrinsically contains two kinds of competition: one is among family members, the other is among the families. In terms of optimization,

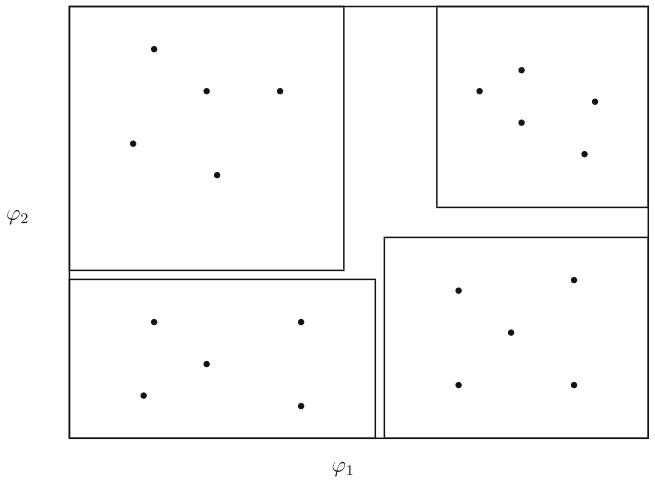

Fig. 1. A two-dimensional space is occupied by four families with five member.

the former competitions can be regarded as local searches with the function of thoroughly examining the possible solutions that spread over the entire search space. The later competition is analogous to global searches. Beside selecting the current best solution, such competition also provide the guide to rearrange the order of each family in the next line up. Thus, each family in LCA is endowed with synchronized mission to improve the current solution and to search for new optima. Hence, through the competitions and cooperations among the families, the global solution may be quickly approached.

As most direct random search algorithms, notably, basic LCA and other EAs use uniform sampling strategy to generate new decision vectors. Masri and Bekey [9] have proved that such sampling policy is tend to use a large number of decision vectors to converge solution especially for high-dimensional systems. That implies the methods using uniform sampling strategy require larger objective function calls, and thus have slow convergence. To improve the convergence of direct random search algorithms, Goulcher and Casares Long [10] have suggested sampling new trial vectors from the neighborhood of the current best solution via normal (or Gaussian) distribution. The authors introduce such kind of sampling policy into LCA to generate new offsprings in our previous studies [6–8]. The results show that such modification is very effective in largely accelerating the convergent rate of basic LCA even the solved problems are highly nonlinear. The detailed steps of this algorithm are shown in the next.

2.1. The detailed steps

1. Assign the numbers of generation, member and family, namely N_g, N_m and N_f.
2. Set the generation counter g as 1. Uniformly generate N_f decision vectors

$$\left\{\Phi^{(*f,g)}|f=1,\ldots,N_f\right\} \tag{2}$$

as the fathers of the first generation over the entire searched space. The fth decision vector in Eq. (2) consists of N_c decision variables

$$\Phi^{(*f,g)}=[\varphi_1^{(*f,g)}\cdots\varphi_c^{(*f,g)}\cdots\varphi_{N_c}^{(*f,g)}]'. \tag{3}$$

The value of $\varphi_c^{(*f,g)}$ can be initially assigned, or produced by an uniform random number generator as follows

$$\varphi_c^{(*f,g)}=L_c^0+\zeta\cdot\Delta_c^{(f,g)}, \tag{4}$$

where $\Delta_c^{(f,g)}=U_c^0-L_c^0$ means the size of the search space for the cth decision variable; ζ is an uniform random number ranging from 0 to 1.

3. Calculate the corresponding objective value for each father.

$$\left\{\left(\Phi^{(*f,g)},\mathcal{I}^{(*f,g)}\right)\Big|f=1,\ldots,N_f\right\}. \tag{5}$$

4. (**Competitions between families**) According to the corresponding objective values, line-up the fathers and the associated search spaces in ascending/descending order for minimization/maximization problems, namely

$$\begin{aligned}&\left\{(\Phi^{(*f,g)},\mathcal{I}^{(*f,g)},\Delta^{(f,g)})|f=1,\ldots,N_f)\right\}\\ &\qquad\qquad\downarrow \textit{ordering}\\ &\left\{\left(\widetilde{\Phi}^{(*f,g)},\widetilde{\mathcal{I}}^{(*f,g)},\widetilde{\Delta}^{(f,g)}\right)|f=1,\ldots,N_f)\right\},\end{aligned} \tag{6}$$

where $\Delta^{(f,g)}\equiv[\Delta_1^{(f,g)},\ldots,\Delta_{N_c}^{(f,g)}]'$ is the size vector of the fth decision vector, $\widetilde{\Delta}^{(f,g)}\equiv[\widetilde{\Delta}_1^{(f,g)},\ldots,\widetilde{\Delta}_{N_c}^{(f,g)}]'$ is the new size vector after ordering.

5. Reduce the size of the search region of the worst family $\widetilde{\Delta}^{(N_f,g)}$, which is just the search region of this generation, to act as the search region in the next generation

$$\Delta\leftarrow\beta\cdot\widetilde{\Delta}^{(N_f,g)}. \tag{7}$$

Notably, a small value of β is enough for simple cases. For high nonlinear cases, however, a larger β is required to globally approach the solution.

6. Add 1 to the counter g. Calculate the new lower and upper bounds in the next generation

$$\begin{aligned}\widetilde{L}_c^{(f,g)}&=\max\left\{\tilde{\varphi}_c^{(*f,g)}-\left(\frac{f\cdot\Delta}{N_f}\right),L_c^0\right\},\\ \widetilde{U}_c^{(f,g)}&=\min\left\{\tilde{\varphi}_c^{(*f,g)}+\left(\frac{f\cdot\Delta}{N_f}\right),U_c^0\right\}.\end{aligned} \tag{8}$$

The new searching space for each family is thus determined as

$$\Delta_c^{(f,g)}=\widetilde{U}_c^{(f,g)}-\tilde{L}_c^{(f,g)}\quad c=1,\ldots,N_c\quad f=1,\ldots,N_f. \tag{9}$$

7. Generate N_m decision vector for each family. The cth variable of the mth vector in the fth family is determined by

$$\varphi_c^{(m,f,g)} \leftarrow \widetilde{\varphi}_c^{(*,f,g-1)} + \lambda\gamma\sigma_c \quad m = 1,\ldots,N_m, \tag{10}$$

where σ_c is the distance from the current best variable $\widetilde{\varphi}_c^{(*,f,g-1)}$ to its nearest bound, namely

$$\min\{(\widetilde{U}_c^{(f,g)} - \widetilde{\varphi}_c^{(*,f,g-1)}),\quad (\widetilde{\varphi}_c^{(*,f,g-1)} - \widetilde{L}_c^{(f,g)})\} \tag{11}$$

γ is a pseudo-random number from a normal distribution of zero mean and one standard deviation; λ is the heuristic parameter with the value of 1/3.

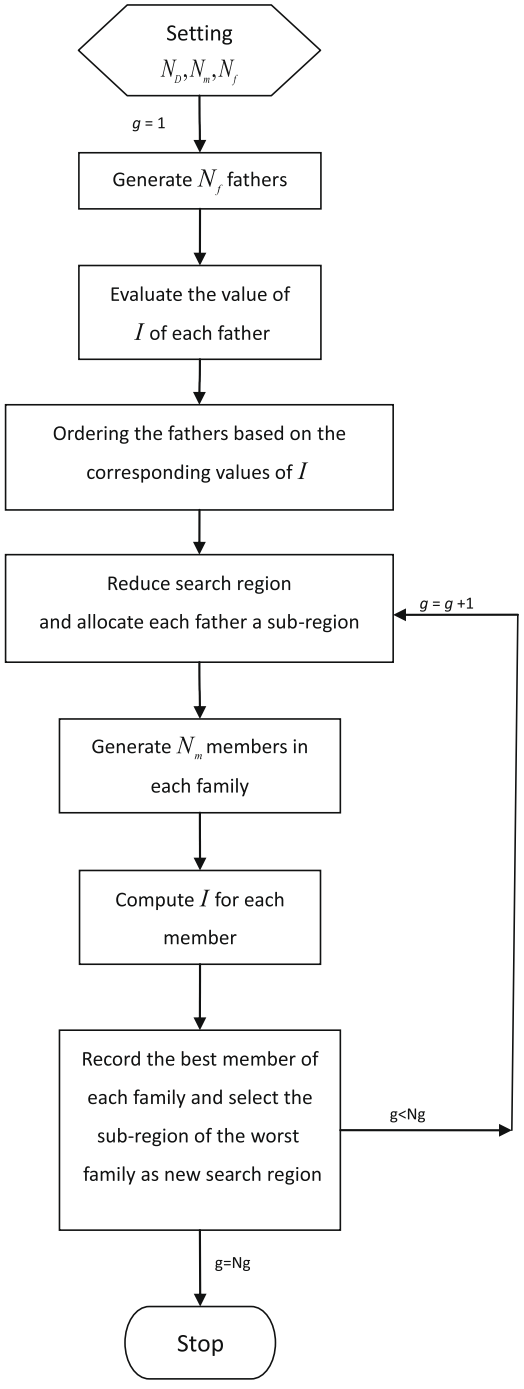

Fig. 2. The flow diagram for LCA.

8. (**Competitions between members**) Select the member with the best objective value from the fth family, namely

$$\{(\Phi^{(m,f,g)}, \mathcal{I}^{(m,f,g)}) \Big| f = 1, \ldots, N_f\} \underset{Select}{\rightarrow} \left(\Phi^{(*,f,g)}, \mathcal{I}^{(*,f,g)}, \Lambda^{(f,g)}\right). \tag{12}$$

9. If $g < N_g$, Go back to Step 4, otherwise stop.

Noted that the above descriptions have been diagramed in Fig. 2.

3. Problem statements

A general parameter selection problem can be posed as

$$\min_{\Phi} \mathcal{I} \tag{13a}$$

subject to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \Phi, t), \tag{13b}$$

$$\mathbf{x}(0) = \mathbf{x}_0, \tag{13c}$$

$$\mathbf{h}(\mathbf{x}(t), \Phi, t) = \mathbf{0}, \tag{13d}$$

$$\mathbf{L}^0 \leqslant \Phi \leqslant \mathbf{U}^0, \tag{13e}$$

where $\mathbf{x}$ is $(N \times 1)$ state vector; $\Phi \equiv [\varphi_1, \ldots, \varphi_{N_c}]'$ denotes the parameter vector to be identified. Derived from first principles, Eqs. (13b) and (13c) represent the ordinary differential equations (ODEs) and the corresponding initial conditions. L equality constraints, Eq. (13d), provide the algebraic relation among the unknown parameters or other considerations. $\mathbf{L}^0$ and $\mathbf{U}^0$ denote the lower and upper bounds on the parameters. Their values are in general determined by the physical intuition of scientists or engineers. Notably, once there have no the time derivative terms, Eq. (13) become the descriptions for a pure curve-fitting problem. In parameter identification problems and curve-fitting problems, the objective function, Eq. (13a), is often cast as minimizing sum of least square errors, namely

$$\mathcal{I} = \sum_{i=1}^{n} \sum_{j=1}^{m} (\tilde{x}_{ij} - x_{ij})^2, \tag{14}$$

where $\tilde{x}_{ij}$ and x_{ij} is the jth prediction for the ith state and x_{ij} the corresponding measured value; n and m are the numbers of the state variables to be sampled and the total sampling numbers, respectively. Notably, n may be smaller than or equal to N in most cases. In optimal control problems, Eq. (13a) can be one of the following formulations:

- *Bolza* type

$$\mathcal{I} = \Theta[t_f] + \int_0^{t_f} \Omega(\mathbf{x}, \Phi, t)\, dt. \tag{15}$$

- *Lagrange* type:

$$\mathcal{I} = \int_0^{t_f} \Lambda(\mathbf{x}, \Phi, t)\, dt. \tag{16}$$

- *Mayer* type:

$$\mathcal{I} = \Lambda[t_f], \tag{17}$$

where Φ means the parametrized control input (s); t_f represents the fixed ending time. Because Eqs. (15) and (16) both can be converted into Eq. (17) by adding a new state variable, the Mayer-type objective function, Eq. (17), is adopted in further discussions. To follow the above formulations, the objective functions need to change theirs sign in maximization problems.

The methods commonly used in the solution of Eq. (13) can be classified as collocation approach and integration approach. Using some trial function to approximate state variables, collocation approaches [11,12] convert Eq. (13) into a strict algebraic form. Then, the model parameters and the coefficients can be simultaneously determined by a well-established optimization technique. Generally speaking, the collocation approaches may converge the solutions very rapidly, especially when the initial guesses are close to the optimum. For highly nonlinear systems, however, largely increasing the order of trial function and the numbers of collocation elements becomes required in order for reducing the approximation errors. As a result, the local convergence might become a very serious problem in further optimization. The integration approaches contain outer and inner loops in solution procedure. Therein, the outer loop is responsible for the selection and production of trial parameters by the use of deterministic [13,14] or stochastic [15–20] optimizers. The new parameters are then passed to the inner loop to estimate the objective value. Such a solution procedure is conceptually very straightforward in use. Regarding the optimizers in outer loop, the deterministic optimizers which use the gradient to update the parameters in general have fast convergent rate. However, such kind of methods may fail to reach the global solution once the systems are nonlinear, multi-modal and/or discontinuous. In stochastic optimizers, on the other hand, the parameters are directly

given via a specially-designed mechanism. Whether the objective value is improved or not is the only consideration in selecting parameters. Thus, the stochastic methods are often more applicable to different kinds of problems. Since no deterministic techniques may guarantee global convergence in solving PS problems [14], this work applies the line-up competition algorithm in the outer loop of integration approach to select unknown parameters. Furthermore, Eqs. (13b)–(13d) consist of the so-called differential-algebraic equations (DAEs). Numerically, such kind of system is very stiff to integrate. Currently, DASSL-like solvers [21] are recognized as the standard integrator in solving DAEs. Once Eq. (13d) is absent, Eq. (13) consist of pure ordinary differential equations. DVERK [22] is the recommended tool for numerical integration.

4. Numerical illustrations

In this section, some examples are provided to illustrate the use of the line-up computation algorithm, which is coded by Compaq Visual Fortran 6.6a under Windows XP operating system. All the following computations are executed on the computer with Intel Core2 Quad Q6600 CPU and 2 Giga of random access memory (RAM).

4.1. Case 1: enzyme effusion problem

The first example to be discussed is described by

$$\begin{aligned}\frac{dx_1}{dt} &= p_1(27.8 - x_1) + \frac{p_4}{2.6}(x_2 - x_1) + \frac{4991}{t\sqrt{2\pi}}\exp\left(-0.5\left(\frac{\ln(t) - p_2}{p_3}\right)^2\right),\\ \frac{dx_2}{dt} &= \frac{p_4}{2.7}(x_1 - x_2),\end{aligned} \tag{18}$$

where the parameters to be identified are p_1, p_2, p_3 and p_4. The measured data of this problem may be referred from Table 1. Notably, the initial conditions of the original problem are unknown, thus, the state values of x_1 and x_2 are considered as the added parameters, p_5 and p_6, to search. As a result, the aim of this problem is to find a set of parameters, p_1, p_2, p_3, p_4, p_5 and p_6, to minimize the following objective function:

$$\mathcal{I} = \sum_{j=1}^{28}(\tilde{x}_{1j} - x_{1j})^2. \tag{19}$$

Using basic genetic algorithm with 500 generations, Nyarko and Scitovski [18] report the best parameters to be 0.284, 2.671, 0.392 and 0.161 with $\mathcal{I} = 4068.38$. Khalik et al. [20] determine the best parameters to be 0.2619, 2.6336, 0.3524 and 0.2575 with $\mathcal{I} = 4136.73$ by using the proposed real-coded GA with 300 generations. Table 2 shows the results by LCA using various N_f, N_m and β after 50 generations. The total members $N_f \times N_m$ in one generation are 200. Besides the results for $N_f = 1$, the other objective values are obviously smaller than the above-reported values. Therein, the best identified parameters are 0.2747, 2.6558, 0.3667 and 0.1998 with $\mathcal{I} = 3935.4$. Fig. 3 shows two trajectories of x_1: one is acquired by the measured data, while the other is determined by the model with the best parameters. Besides the spike point, most solutions can properly fit the measured data. Such phenomenon might result from the model error. Notably, the LCA can be regarded as a special kind of direct random search algorithm as $N_f = 1$. The corresponding objective value also shown in Table 2 is obviously unsatisfactory. Such outcomes seem to imply the policy of using multiply families in LCA is very effective in improving the solutions.

Table 1
Measured data for enzyme effusion problem.

t	x_1	t	x_1	t	x_1	t	x_1
0.1	27.8	21.3	331.9	42.4	62.3	81.1	23.5
2.5	20	22.9	243.5	44.4	58.7	91.1	24.8
3.8	23.5	24.9	212	47.9	41.9	101.9	26.1
7	63.6	26.8	164.1	53.1	40.2	115.4	33.3
10.9	267.5	30.1	112.7	59	31.3	138.7	17.8
15	427.8	34.1	88.1	65.1	30	163.2	16.8
18.2	339.7	37.8	76.2	73.1	30.6	186.7	16.8

Table 2
The identified parameters and the objective values for Case 1 by LCA.

p_1	p_2	p_3	p_4	$\mathcal{I}$	N_f	N_m	β
0.3277	2.7050	0.4060	0.0201	5799.9	1	200	0.9
0.2820	2.6655	0.3782	0.1998	4027.6	5	40	0.9
0.2740	2.6563	0.3645	0.2000	3943.5	10	20	0.85
0.2747	2.6558	0.3667	0.1998	3935.4	20	10	0.85
0.2723	2.6517	0.3651	0.1996	3960.6	25	8	0.85

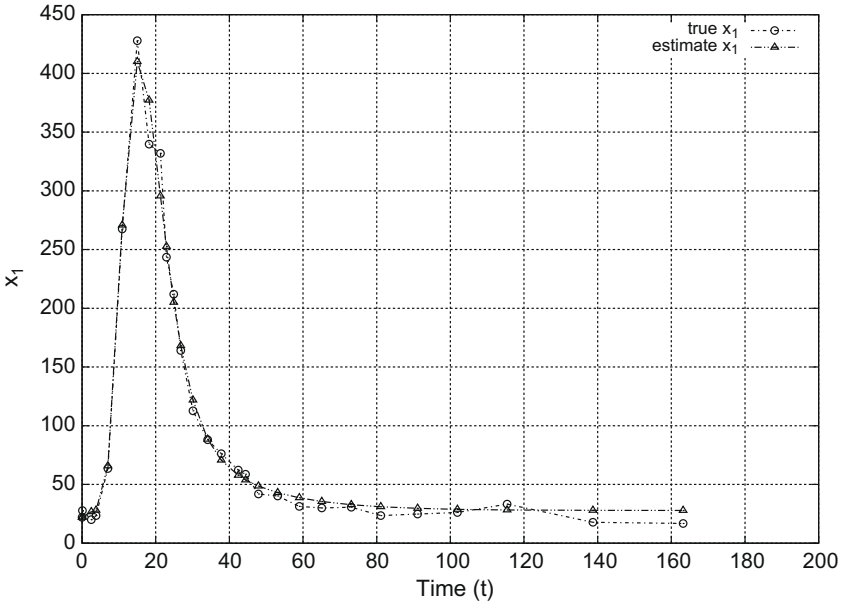

Fig. 3. The trajectories of x_1 for Case 1 from the measured data and the model.

4.2. Case 2: glucose conversion problem

Proposed by Rai and Constantinides [23], this model is used to formulate the conversion of glucose to gluconic acid by simple oxidation of sugar, which can be summarized as

$$\begin{aligned}
\frac{dx_1}{dt} &= b_1 x_1 \left(1. - \frac{x_1}{b_2}\right), \\
\frac{dx_2}{dt} &= \frac{b_3 x_1 x_4}{b_4 + x_4} - 0.9082 b_5 x_2, \\
\frac{dx_3}{dt} &= b_5 x_2, \\
\frac{dx_4}{dt} &= -1.011 \left(\frac{b_3 x_1 x_4}{b_4 + x_4}\right)
\end{aligned} \tag{20}$$

with the initial conditions

$$\mathbf{x}(0) = [0.5, 0., 0., 50.]'. \tag{21}$$

The true values for b_1, b_2, b_3, b_4 and b_5 are 0.949, 3.439, 18.72, 37.51 and 1.169, respectively. In this case, we first use the above model coupled with the true parameters to generate 50 sets of nominal data. Then, these data are given to LCA to identify the parameters. The objective function to be minimized is thus defined as

$$\mathcal{I} = \sum_{i=1}^{4} \sum_{j=1}^{50} (\tilde{x}_{ij} - x_{ij})^2. \tag{22}$$

Determined by LCA with 200 total members in one generation, the solutions of this problem are shown in Table 3. The best identified parameters are 0.9713, 3.4231, 18.7445, 38.6261 and 1.1663 by LCA with 100 generations. The corresponding objective value can be reduced to 0.182. The corresponding trajectories for x_1, x_2, x_3 and x_4 acquired by the model with the true and the identified parameters are shown in Fig. 4. The data generated by the model with identified parameters almost comply with the nominal ones. As the previous example, the objective value acquired by LCA with $N_f = 1$ is very bad.

Moreover, we also add ±5% of errors to the generated data to simulate measured noise. The parameters used in LCA are set the same as that shown in Table 3. The best parameters for such modification are determined to be 0.9595, 3.4788, 18.2332, 36.7789, 1.1742 with $\mathcal{I} = 23.972$ as shown in Table 4. The corresponding trajectories for x_1, x_2, x_3 and x_4 are shown in Fig. 5. Obviously, the model with the identified parameters still respond the trend of the dynamics of this case under the measured noise.

Table 3
The identified parameters and the corresponding objective values for Case 2 by LCA.

b_1	b_2	b_3	b_4	b_5	$\mathcal{I}$	N_f	N_m	β
0.9728	3.6490	20.3508	46.2900	1.1462	3.160	1	200	0.97
0.9688	3.5963	20.0145	44.6754	1.1722	1.505	5	40	0.97
0.9713	3.4231	18.7445	38.6261	1.1663	0.182	10	20	0.97
0.9668	3.4901	19.0927	40.1180	1.1677	0.257	20	10	0.90
0.9022	3.5458	19.7913	39.3689	1.1539	0.830	25	8	0.90

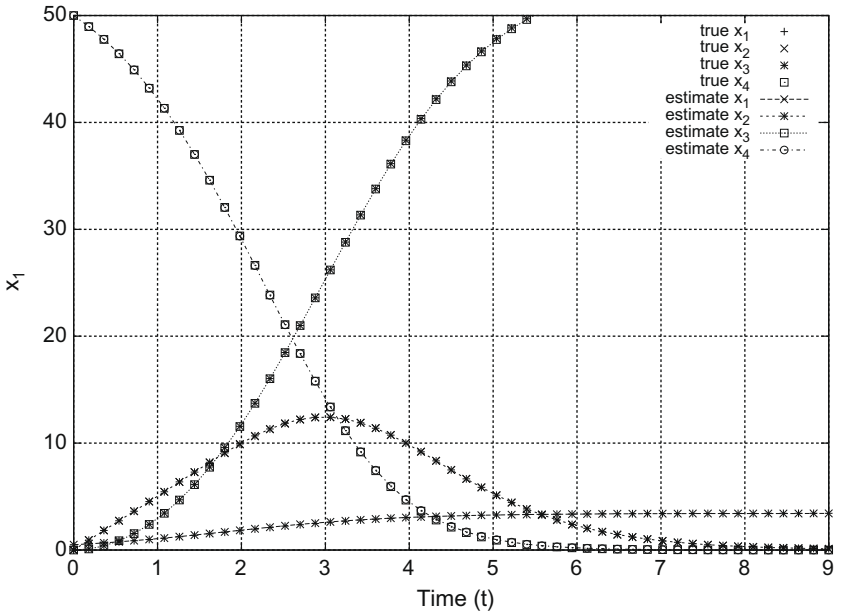

Fig. 4. The trajectories from the true model and the estimated model for Case 2 by LCA with $N_f = 10$ and $N_m = 20$.

Table 4
The identified parameters and the objective values for Case 2 with ±5% measured errors.

b_1	b_2	b_3	b_4	b_5	$\mathcal{I}$	N_f	N_m	β
1.0264	3.4732	16.9582	35.0996	1.1871	24.963	5	40	0.97
0.9668	3.6936	21.3644	49.9978	1.1769	28.353	10	20	0.97
0.9767	3.4453	17.6394	35.0240	1.1760	23.992	20	10	0.90
0.9595	3.4788	18.2332	36.7789	1.1742	23.972	25	8	0.90

4.3. Case 3: photochemical reaction in continuous stirred tank reactor

The third example to be discussed is a three-input optimal control problem described by eight differential equations

$$\begin{aligned}
\frac{dx_1}{dt} &= 6.0 - qx_1 - 17.6x_1x_2 - 23.0x_1x_6x_3,\\
\frac{dx_2}{dt} &= u_1 - qx_2 - 17.6x_1x_2 - 146.0x_2x_3,\\
\frac{dx_3}{dt} &= u_2 - qx_3 - 73.0x_2x_3,\\
\frac{dx_4}{dt} &= -qx_4 + 35.2x_1x_2 + 51.3x_4x_5,\\
\frac{dx_5}{dt} &= -qx_5 + 219.0x_2x_3 - 51.3x_4x_5,\\
\frac{dx_6}{dt} &= -qx_6 + 102.6x_4x_5 - 23.0x_1x_6u_3,\\
\frac{dx_7}{dt} &= -qx_7 + 46.0x_1x_6u_3,\\
\frac{dx_8}{dt} &= 5.8(qx_1 - 6) - 3.7u_1 - 4.1u_2 - 5.u_3^2 + q(23x_4 + 11x_5 + 28x_6 + 35x_7),\\
q &= 6.0 + u_1 + u_2
\end{aligned} \tag{23a}$$

with the initial conditions

$$\mathbf{x}(0) = [0.1883, 0.2507, 0.0467, 0.0899, 0.1804, 0.1394, 0.1046, 0]'. \tag{23b}$$

The control inputs are bounded by

$$\begin{aligned} &0 \leqslant u_1 \leqslant 20, \\ &0 \leqslant u_2 \leqslant 6, \\ &0 \leqslant u_3 \leqslant 4. \end{aligned} \tag{23c}$$

The aim of the problem is to find the continuous inputs, $u_1(t)$, $u_2(t)$ and $u_3(t)$, that satisfy Eqs. (23a)–(23c) to maximize the objective function.

$$\mathcal{I} = x_8(t_f), \tag{23d}$$

where $t_f = 0.2$. Using iterative dynamic programming with 10 time stages to solve this problem, Luus [24] reports the best objective value to be 20.09. Smith [25] uses evolutionary programming to solve this problem by discretizing the problems into 30 or 50 consecutive subproblems. The acquired best and worst solutions are 20.0914 and 20.0512. Although the solution are very close to the previously reported value, it is obvious that the size of the discretized problem is too big.

To solve this problem under the frame work of integration approach coupled with LCA, the time horizon ranging from t_0 to t_f is evenly divided into P sections. Consequently, the time length of the jth section is

$$\Delta T = \frac{t_f}{P} = \omega_{j+1} - \omega_j \quad j = 1, \ldots, P, \tag{24}$$

where ω_j and ω_{j+1} denote the grid points at the starting and ending time of the jth section. Obviously, there are $P + 1$ time grids evenly spreading over the entire time horizon, and ω_1 and ω_{P+1} are corresponding to t_0 and t_f. If $\phi_{i,j}$ denotes the value of $u_i(t)$ at $t = \omega_j$. The continuous input u_i is thus approximated by $\{\phi_{i,j} | j = 1,\ldots,P + 1\}$. By the above parametrization, Eq. (23) can be re-posed as

$$\min_{\{\phi_{i,j} | i=1,\ldots,3, j=1,\ldots,P+1\}} \mathcal{I} = x_8(t_f) \tag{25a}$$

subject to

$$\begin{aligned} \frac{dx_1}{dt} &= 6.0 - qx_1 - 17.6x_1x_2 - 23.0x_1x_6x_3, \\ \frac{dx_2}{dt} &= u_{1,j} - qx_2 - 17.6x_1x_2 - 146.0x_2x_3, \\ \frac{dx_3}{dt} &= u_{2,j} - qx_3 - 73.0x_2x_3, \\ \frac{dx_4}{dt} &= -qx_4 + 35.2x_1x_2 + 51.3x_4x_5, \\ \frac{dx_5}{dt} &= -qx_5 + 219.0x_2x_3 - 51.3x_4x_5, \\ \frac{dx_6}{dt} &= -qx_6 + 102.6x_4x_5 - 23.0x_1x_6u_{3,j}, \\ \frac{dx_7}{dt} &= -qx_7 + 46.0x_1x_6u_{3,j}, \\ \frac{dx_8}{dt} &= 5.8(qx_1 - 6) - 3.7u_{1,j} - 4.1u_{2,j} - 5.u_{3,j}^2 + q(23x_4 + 11x_5 + 28x_6 + 35x_7), \\ q &= 6.0 + u_{1,j} + u_{2,j}, \\ 0 &\leqslant u_{1,j}(t) \leqslant 20, \\ 0 &\leqslant u_{2,j}(t) \leqslant 6, \\ 0 &\leqslant u_{3,j}(t) \leqslant 4, \\ \forall t &\in [\omega_j, \omega_{j+1}) \quad i = 1, \ldots, P, \end{aligned} \tag{25b}$$

where $u_{i,j}(t)$ denote the ith control applied in the duration of $[\omega_j, \omega_{j+1})$. Notably, when the objective value is estimated in inner loop, the value of $u_{i,j}(t)$ can be interpolated by piecewise constant (PC) function

$$u_{i,j}(t) = \phi_{i,j} \tag{26}$$

or piecewise ramp (PR) function

$$u_{i,j}(t) = \frac{\phi_{i,j+1} - \phi_{i,j}}{\omega_{j+1} - \omega_j}(t - \omega_j) + \phi_{i,j}. \tag{27}$$

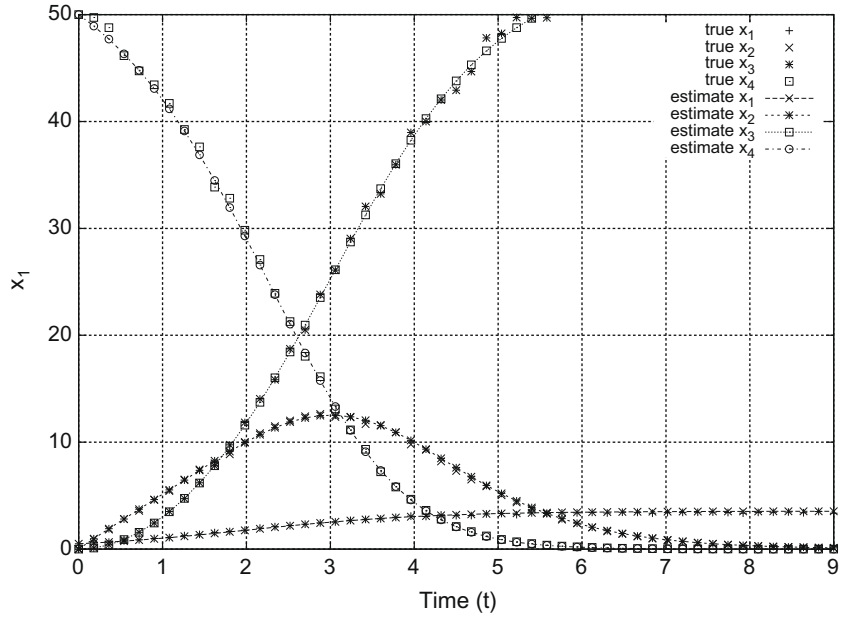

Fig. 5. The trajectories from the true model and estimated model with 5% measured noise.

Table 5
The objective values for Case 3 by LCA with PC-type and PR-type control inputs.

N_f	N_m	$\mathcal{I}$ by PC function	$\mathcal{I}$ by PR function
8	25	20.0631	20.0657
10	20	20.0537	20.0728
20	10	20.0832	20.0779
25	8	20.0716	20.0477

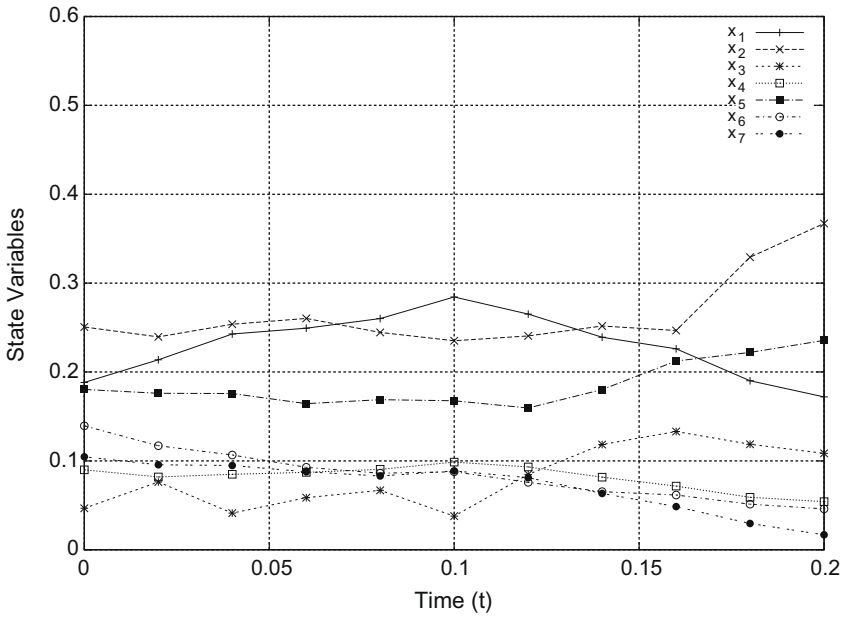

Fig. 6. The best state trajectories in Example 3, where $\mathcal{I} = 20.0832$.

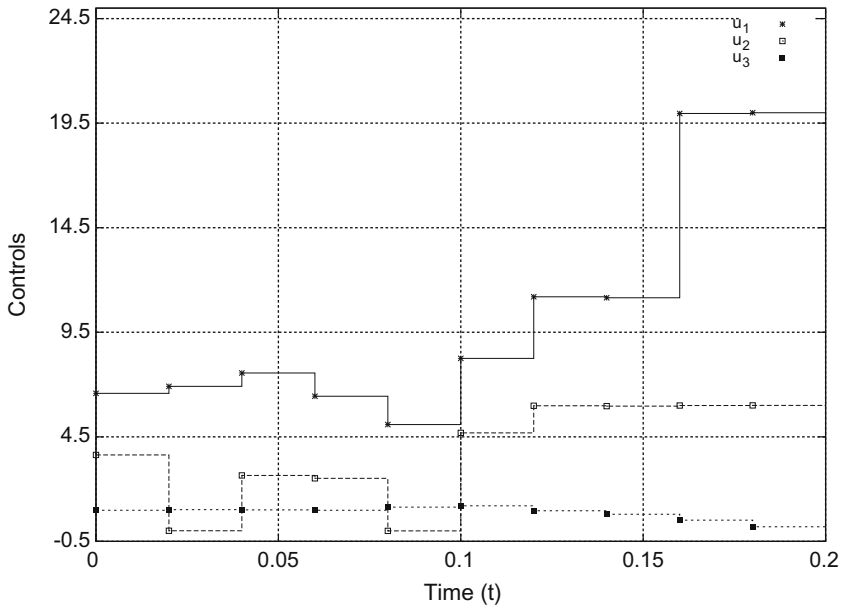

Fig. 7. The best control profiles for u_1, u_2 and u_3 in Example 3, where $\mathcal{I} = 20.0832$.

By setting $P = 10$, we solve this problem by LCA with various N_f, N_m and β. The value of β is given as 0.99. The total trial members are given as 200. Table 5 shows the objective values acquired by PC and PR functions after 300 generations. Most objective values can be close to the known best value within 1%. Since the solutions are acquired by the system with 30 decision variables, it implies LCA is still very efficient in solving high-dimensional problem. The state trajectories and control profiles for the case with $\mathcal{I} = 20.0832$ are, respectively, shown in Figs. 6 and 7.

5. Conclusions

In this study, the modified line-up competition algorithm is applied to solve nonlinear parameter selection problems. Three typical problems, including two nonlinear parameter identification problems and one multi-input optimal control problem, are used as demonstration for the proposed method. In order to convert into a PS problem, the inputs in the optimal control problem are parametrized. The unspecified parameters embedded in the above problems are selected by LCA under the framework of integration approach. The results show that in addition to being very efficient to converge to the global solutions, the method is also very robust to the initial conditions of optimization. Compared with other EAs, the solution policies used in LCA is simple and direct.

Acknowledgement

The authors thank the anonymous reviewers for their constructive comments and suggestions.

References

[1] J.E. Cuthrell, L.T. Biegler, On the optimization of differential-algebraic process systems, AIChE Journal 33 (1987) 1257–1270.
[2] C.J. Goh, K.L. Teo, Control parametrization: a unified approach to optimal control problems with general constraints, Automatica 24 (1988) 3–18.
[3] L.X. Yan, D.X. Ma, Global optimization of nonconvex nonlinear programs using line-up competition algorithm, Computer and Chemical Engineering 25 (2001) 1601–1610.
[4] L.X. Yan, D.X. Ma, Solving combinatorial optimization problems with line-up competition algorithm, Computer and Chemical Engineering 27 (2003) 251–258.
[5] L.X. Yan, K. Shen, S.H. Hu, Solving mixed integer nonlinear programming problems with line-up competition algorithm, Computer and Chemical Engineering 28 (2004) 2647–2657.
[6] D.Y. Sun, P.M. Lin, S.P. Lin, Integrating controlled random search into line-up competition algorithm to solve the unsteady operation problems, Industrial and Engineering Chemistry Research 47 (2008) 8869–8887.
[7] D.Y. Sun, T.C. Huang, The solutions of time-delayed optimal control problems by the use of modified line-up competition algorithm, Journal of the Taiwan Institute of Chemical Engineers 41 (2010) 54–64.
[8] D.Y. Sun, The solution of singular optimal control problems using the modified line-up competition algorithm with region-relaxing strategy, ISA Transactions 49 (2010) 106–113.
[9] S.F. Masri, G.A. Bekey, A global optimization algorithm using adaptive random search, Applied Mathematics and Computation 7 (1980) 353–375.
[10] R. Goulcher, J.J. Casares Long, The solution of steady-state chemical engineering optimisation problems using random-search algorithm, Computer and Chemical Engineering 2 (1978) 33–36.

[11] J. Villadsen, M.L. Michelsen, Solutions of Differential Equations by Polynominal Approximation, Prentice-Hall, Inc., Englewood Cliffs, USA, 1978.
[12] I.B. Tjao, L.T. Biegler, Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems, Industrial and Engineering Chemistry Research 30 (1991) 358–376.
[13] C.A. Schnepper, M.A. Stadtherr, Robust process simulation using interval methods, Computer and Chemical Engineering 20 (1996) 187–199.
[14] W.R. Esposito, C.A. Floudas, Global optimization for the parameter estimation of differential-algebraic systems, Industrial and Engineering Chemistry Research 39 (2000) 1291–1310.
[15] Z. Belohlav, P. Zamostny, P. Kluson, J. Volf, Application of random search algorithm for regression analysis of catalytic hydrogenation, The Canadian Journal of Chemical Engineering 75 (1997) 735–742.
[16] T.Y. Park, G.F. Froment, A hybrid genetic algorithm for the estimation of parameters in detailed kinetic models, Computer and Chemical Engineering 22 (1998) S103–S110.
[17] A. Eftaixias, J. Font, A. Fortuny, A. Fabregat, F. Stüber, Nonlinear kinetic parameter estimation using simulated annealing, Computer and Chemical Engineering 26 (2002) 1725–1733.
[18] E.K. Nyarko, R. Scitovski, Solving the parameter identification problem of mathematical models using genetic algorithm, Applied Mathematics and Computation 153 (2004) 651–658.
[19] L.L. Li, L. Wang, L.H. Liu, An efficient hybrid PSOSA strategy for optimization, its application to parameter estimation, Applied Mathematics and Computation 179 (2006) 135–146.
[20] M.A. Khalik, M. Sherif, S. Saraya, F. Areed, Parameter identification problem: real-coded GA approach, Applied Mathematics and Computation 187 (2007) 1495–1501.
[21] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical Solution of Initial Value Problems in Differential-Algebraic Equations, North-Holland, New York, USA, 1989.
[22] T.E. Hull, W.D. Enright, K.R. Jackson, User Guide to DVERK – A Subroutine for Solving Nonstiff ODE's, Department of Computer Science, University of Toronto, Canada, vol. 100, 1976.
[23] V.R. Rai, A. Constantinides, Mathematical modeling and optimization of the gluconic acid fermentation, AIChE Symposium Series 69 (1973) 114–122.
[24] R. Luus, Application of dynamic programming to high-dimensional non-lionear optimal control problems, International Journal of Control 52 (1990) 239–250.
[25] S. Smith, An program for a class of continuous optimal control problems, IEEE International Conferences on Evolutionary Computation 1 (1995) 418–422.