Pseudo derivative evolutionary algorithm and convergence analysis
Yu, Yang; Wang, Zhongjie; Tianfield, Huaglory; Lu, Chengchao

Published in:
International Journal of Modeling, Simulation, and Scientific Computing

DOI:
10.1142/S1793962318500411

Publication date:
2018

Document Version
Peer reviewed version

Link to publication in ResearchOnline

Citation for published version (Harvard):
Pseudo Derivative Evolutionary Algorithm and Convergence Analysis

Yang Yu\textsuperscript{a,1}, Zhongjie Wang\textsuperscript{b,2,∗}, Huaglory Tianfield\textsuperscript{c,3}

\textsuperscript{a}Department of Control Science and Engineering, Tongji University, Shanghai 201804, China
\textsuperscript{b}Department of Control Science and Engineering, Tongji University, Shanghai 201804, China
\textsuperscript{c}Department of Computer, Communications and Interactive Systems, Glasgow Caledonian University, Glasgow G4 0BA, United Kingdom

Abstract

In this paper a novel evolutionary algorithm (EA), called pseudo derivative evolutionary algorithm (PDEA), is proposed. The basic idea of PDEA is to use pseudo derivative, which is obtained based on the information produced during the evolution, to help search the solution of optimization problem. The pseudo derivative drives the search process in a more informed direction. That makes PDEA different from the random optimization methods. The convergence of PDEA is first analyzed based on systems theory. The convergence condition of PDEA is then derived, though this condition is too strong to be satisfied. Next, this condition is relaxed based on entropy theory. Finally, performances of PDEA are evaluated on the benchmark functions and an adaptive liquid level control system of a surge tank. The numeric simulation results show that PDEA is capable of finding the solutions to the optimization problems with good accuracy, reliability, and speed.

Keywords: pseudo derivative evolutionary algorithm (PDEA), evolutionary algorithm (EA), convergence analysis, entropy theory

1. Introduction

Evolutionary algorithm (EA) is a kind of population algorithm to search for an optimal or near optimal solution to complex problems in polynomial time. Common variants of EA include genetic algorithm (GA) \cite{1–3}, genetic programming \cite{4} \cite{5}, differential evolution (DE) \cite{6} \cite{7}, particle swarm optimization (PSO) \cite{8–10}, bacterial foraging optimization (BFO) \cite{11–13}, artificial bee colony algorithm (ABC) \cite{14} etc.

EAs do not require gradient information. Therefore, they are capable of solving a wide variety of optimization problems, which may be linear, quadratic, unimodal, discontinuous, non-differentiable, strongly convex, etc \cite{15–18}. However, different from derivative based iterative methods for optimization, the convergence of EAs have not proved explicitly, or is proved under some very strong hypothesis.

To analyse the convergence of GA, a Markov chain model is proposed \cite{19–22}. The process of GA can be proved as a homogeneous Markov chain. Then GA is analysed based on the properties and theorems of the Markov chain. GA is proved to be convergent with the assumption that the number of generations is infinite. PSO can be modelled by dynamical equations \cite{23–26}. The evolution process of PSO can be divided into two different parts. The first part assumes each individual has an initial position and velocity. The second part is about how to produce the next generation individuals. The velocity is modified by two kinds of information, i.e., the best position in all preceding generations and the best position the individual ever obtained. With iterations of these two parts, PSO is able to find the solution to an optimization problem. By modelling the iteration process, the dynamic equations of PSO can be formulated. Thereafter, PSO is analysed based on the dynamic equations. However, the convergence condition of PSO is too strong that even the standard PSO is not convergent. BFO, modelling the individual and group behaviour of \textit{E.Coli} bacteria, is a distributed optimization process. Reference \cite{27} analyses BFO by formulating the mathematical model of the chemotactic movements in continuous time. Then the Lyapunov stability theorem is used to analyse the convergence of the dynamic model. It assumes the objective function is continuously differentiable. References \cite{28} \cite{29} analyse the one-dimensional DE based on a mathematical model. The model has been formulated based on probability theory and dynamic equation. It assumes the trial solutions are limited within a small region, and the fitness landscape has a moderate gradient.

This paper proposes a pseudo derivative EA (PDEA), which explicitly uses the pseudo derivative obtained based on the information produced during the evolution process to search for the optimal solution. First of all, the convergence of PDEA is studied based on systems theory. PDEA is proved to be convergent. The convergence condition is derived, though this condition is too strict and hardly satisfied. Then, we relax the convergence condition based on entropy theory.

The remainder of this paper is arranged as follows. The principle of PDEA is presented in Section 2. In Section 3, con-
verge analysis is conducted to PDEA based on systems theory and entropy theory. In Section 4, performance evaluations are conducted by applying PDEA to benchmark functions and an adaptive liquid level control system. Finally, conclusions are drawn in Section 5.

2. PDEA

2.1. Pseudo Derivative

Iterative methods for optimization, such as Newton’s method, gradient descent method, employ derivative of the objective function in searching the solution. By analogy, we introduce the concept of pseudo derivative to EAs. The basic idea of PDEA is to make use of a pseudo derivative information explicitly in searching the solution to the optimization problem.

The information produced during the evolution contains the individual position and fitness value in preceding and current generations, and the distance between two individuals. Therefore, we define a pseudo derivative as below.

\[
pdx = \frac{f(i) - f(j)}{∥pos(i) - pos(j)∥} \quad i \neq j
\]

where \(i, j\) are indices of individuals, \(f(i)\) and \(pos(i)\) denote the fitness value and position of individual \(i\) within the search space. \(∥pos(i) - pos(j)∥\) is the distance between individual \(i\) and individual \(j\). The right-hand side of Eq.1 means the average change of fitness value over \(∥pos(i) - pos(j)∥\). Actually, we can find that \(\lim_{∥pos(i) - pos(j)∥ \to 0} pdx\) is the derivative in the direction of vector \((pos(i) - pos(j))\). Therefore, \(pdx\) is called the pseudo derivative.

Using the best fitness values in the current generation and across all the preceding generations, we have

\[
pdc = \frac{fcbest - f(i)}{∥pcbest - pos(i)∥}
\]

\[
pdh = \frac{fhbest - f(i)}{∥phbest - pos(i)∥}
\]

where \(fcbest\) denotes the best fitness value in current generation, \(pcbest\) the corresponding position. \(fhbest\) and \(phbest\) denote the best fitness value and position across all preceding generations. \(pdc\) and \(pdh\) are the pseudo derivatives which represent the average fitness change in the direction of a individual towards the best individual. Compared with true derivative, \(pdc\) and \(pdh\) represent the approximations of fitness value changes during the search process in the directions towards \(pcbest\) and \(phbest\), respectively.

2.2. Algorithm

Now, we have two kinds of pseudo derivative informations \(pdc\) and \(pdh\) in PDEA. Different from true derivative, pseudo derivatives just reflect the approximation of the fitness value changes in two certain directions, but still, \(pdc\) and \(pdh\) can provide the information of objective function. In order to expand the search region and avoid premature convergence, we encourage the individual to explore its search region. For this purpose, we design the algorithm such that each individual has a random direction which is defined by \(pdr\) with a step to move within the neighbourhood, as illustrated in Fig.1.

The individual behaviour is driven by the three kinds of pseudo derivatives, i.e., \(pdc\), \(pdh\), and \(pdr\). Regardless of the driving strategies, the procedure of PDEA can be described as in Algorithm 1.

![Pseudo derivatives of individual](image)

**Algorithm 1 PDEA**

**Input:** maximum number of iterations \(Nmax\), and \(k = 0\)

1: generate the initial individuals and calculate the fitness values;
2: for \(k \leq Nmax\) do
3: obtain \(pdc\), \(pdh\), and \(pdr\) of every individual;
4: update individuals according to the driving strategy and form the new generation
5: new individual := \(S_k\) (individual, \(pdc\), \(pdh\), \(pdr\));
6: end for

**Output:** the best individual and its fitness value;

Different driving strategies defined by \(S(\cdot)\) will result in different paradigms of PDEA. The driving strategies \(S(\cdot)\) used to update individuals can take various forms. Without loss of generality, here we consider the strategy as follows.

\[
S_{k+1}(i) = \alpha \cdot u_1 \cdot (pcbest(k) - pos_k(i)) + \beta \cdot u_2 \cdot (phbest(k) - pos_k(i)) + \gamma \cdot u_3 \cdot S_k(i)
\]

\[
S_1(i) = d
\]

\[
pos_{k+1}(i) = pos_k(i) + S_{k+1}(i)
\]

where \(\alpha, \beta, \gamma\) are the weight coefficients in the direction of \(pdc\), \(pdh\) and \(S_k\). \(S_1\) is determined by \(pdr\). The value of \(\alpha, \beta, \gamma\) are determined by \(pdc\), \(pdh\), and \(pdr\). \(d\) is a random vector for the local search strategy with a given length. \(k\) is the index of iterations. \(pos_k(i)\) denotes the position of individual \(i\) in generation \(k\). \(pcbest(k)\), \(phbest(k)\) denote the position of the best individual in current generation \(k\) and all preceding
generations prior to \( k \). \( \mathbf{u} = (u_1, u_2, u_3) \) represents the probability weight coefficients. Each element of \( \mathbf{u} \) can be a probability distribution or a logistic variable for determining whether to use the corresponding pseudo derivative. For example, if \( \mathbf{u} = (0, 0, 1) \), \( S(i, pdc, pdh, pdr) \) it is the same as the mutation part of DE/rand/1 algorithm [30].

3. Convergence Analysis of PDEA

3.1. Proof of PDEA convergence

Considering Eqs. (4) (5) (6), for individual \( i \) in generation \( k \), let’s use notations omitting the individual’s index, denote \( x(k) = S(i), y(k) = \text{pos}(i) \). Then we have

\[
x(k + 1) = \varphi_1(p\text{best}(k) - y(k)) + \varphi_2(p\text{best}(k) - y(k)) + \varphi_3 x(k)
\]

\[
y(k + 1) = y(k) + x(k + 1)
\]

\[
\varphi_1 = \alpha \cdot u_1
\]

\[
\varphi_2 = \beta \cdot u_2
\]

\[
\varphi_3 = \gamma \cdot u_3
\]

Definition 3.1. The discrete-time state space system of Eq. (12) is said controllable if for any initial state \( \mathbf{z}(0) = \mathbf{z}_0 \) and any target state \( \mathbf{z}_t \), there exists an input series that transfers \( \mathbf{z}_0 \) to \( \mathbf{z}_t \). Otherwise, the system is said uncontrollable.

Lema 3.1. The discrete-time state space system of Eq. 12 is controllable, if the controllability matrix

\[
\mathbf{C} = [\mathbf{B} \mathbf{AB} \mathbf{A}^2 \mathbf{B} \ldots \mathbf{A}^{n-1} \mathbf{B}]
\]

has a full row rank of \( n \), where \( n \) is the dimension of matrix \( \mathbf{A} \).

Therefore, we have

\[
\mathbf{C}_\zeta = \begin{bmatrix} \zeta & \varphi_3 \zeta - \zeta^2 \\ \varphi_3 \zeta - \zeta^2 + \zeta \end{bmatrix}
\]

It is clear that matrix \( \mathbf{C}_\zeta \) is of full rank when \( \zeta \neq 0 \). So, system of Eq. (12) is controllable, that is, it is convergent.

3.2. Convergence condition of PDEA

Consider the autonomous system

\[
\mathbf{z}(k + 1) = \mathbf{A}\mathbf{z}(k)
\]

Definition 3.2. System \( \mathbf{z}(k + 1) = \mathbf{A}\mathbf{z}(k) \) is marginally stable or stable in the sense of Lyapunov if every finite state \( \mathbf{z}_0 \) excites a bounded response. It is asymptotically stable if every finite initial state excites a bounded response, which approaches 0 as \( t \to \infty \).

Lema 3.2. System \( \mathbf{z}(k + 1) = \mathbf{A}\mathbf{z}(k) \) is marginally stable iff all eigenvalues of \( \mathbf{A} \) have magnitudes less than or equal to 1 and those equal to 1 are simple roots of the minimal polynomial of \( \mathbf{A} \). It is asymptotically stable iff all eigenvalues of \( \mathbf{A} \) have magnitudes less than 1.

The eigenvalues of \( \mathbf{A} \) are as follows

\[
\begin{align*}
\lambda_1 &= e^{\varphi_1 t} - \varphi_2 \\
\lambda_2 &= e^{\varphi_1 t} - \varphi_3 \\
\lambda_3 &= e^{\varphi_1 t} - \varphi_1 - \varphi_2 - \varphi_3
\end{align*}
\]

When \( \Lambda < 0 \), \( e_1 \) and \( e_2 \) is a pair of conjugate complex. \( ||e_1|| = ||e_2|| = \sqrt{\varphi_1} \). If \( 0 \leq \sqrt{\varphi_1} < 1 \), the system is convergent. When \( \Lambda \geq 0 \), if both ||\( e_1 \)|| and ||\( e_2 \)|| are less than 1, the system is convergent. So, the convergence condition is the union of the two cases above, that is.

\[
\mathcal{R}_c = \{ \Lambda < 0 \text{ and } \varphi_3 \in [0, 1] \} \cup \{ \Lambda \geq 0 \text{ and } ||e_{1,2}|| < 1 \}
\]

\( \mathcal{R}_c \) is termed the convergence region. \( ||e_{1,2}|| < 1 \) represents both \( ||e_1|| \) and \( ||e_2|| \) are less than 1.

To further examine the convergence, we can derive system Eq. (12) as follows

\[
\begin{align*}
\mathbf{x}(k + 2) - (\varphi_3 - \zeta + 1)\mathbf{x}(k + 1) + \varphi_3 \mathbf{x}(k) &= 0 \\
y(k) &= \Gamma - \frac{x(k + 1) - \varphi_3 \mathbf{x}(k)}{\zeta}
\end{align*}
\]

So, we obtain

\[
\begin{align*}
x(k) &= c_1 e_1^{k} + c_2 e_2^{k} \\
y(k) &= \Gamma - \frac{c_1 e_1^{k} (e_1 - \varphi_3) + c_2 e_2^{k} (e_2 - \varphi_3)}{\zeta}
\end{align*}
\]

\( c_1 \) and \( c_2 \) are defined by initial state \( \mathbf{z}_0 \). Since both ||\( e_1 || \) and ||\( e_2 || \) are less than 1 under the convergence condition, \( \lim_{k \to \infty} x(k) = 0, \lim_{k \to \infty} y(k) = \Gamma \).

In order to investigate the convergence process, we set \( \varphi_3 = 0, \zeta = 1.5 \) to satisfy \( \mathcal{R}_c \) and test in the 2 dimensional Rastigin function. It’s shown in Fig.2.
3.3. Convergence condition relaxation

However, the convergence condition $R_\zeta$ is too strong to be satisfied. Here we will relax it. Let’s consider the main procedure of PDEA. At the beginning, the initial generation is created randomly. Each individual appears with the same probability $p$ in the solution space. Then, the individual is driven by the strategy consisted of $pdc$, $pdh$, and $pdr$ to convergence point. The closer the individual to the convergence point, the greater the probability that it appears in the area near the convergence point.

From the point of view of entropy theory, we define the entropy of the individual as $\log(1/p(i))$, where $p(i)$ denotes the probability of individual $i$ appearing in a certain area of the solution space. $\log(1/p(i))$ represents the uncertainty of individual $i$. Therefore, the entropy of generation $k$ can be defined as follows.

$$H(k) = E[\log(1/p(i))] = \sum_{i=1}^{n} p(i) \cdot \log(1/p(i))$$  \hfill (28)

Then, the behaviour of the individual is analyzed by $H(k)$.

From the convergence condition, Eqs. (26) and (27) derived of system of Eq. (12), it shows that all the individuals converge to the point determined by $\Gamma$. To inspect the system, during the iterations, $x(k)$ trends to 0 and $y(k)$ trends to $\Gamma$. That reflects the variation tendency of individuals that their positions are getting closer to $\Gamma$. The area where individuals appear is becoming more and more certain. Therefore, $H(k)$ is decreasing to 0, when $k \to \infty$.

However, the convergence condition that all the eigenvalues of $A$ have magnitudes less than 1 is too strong. Consider the general situation that $u = (u_1, u_2, u_3)$ is some probability distributions. According to the analysis based on systems theory, PDEA is convergent if and only if all the range of the distribution falls fully within $R_\zeta$. However, in terms of the analysis

Based on entropy theory, the convergence condition is relaxed. Let’s illustrate it by an example.

Consider the system of Eq. (12), the solution is

$$\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} c_1 \\ -\zeta c_1 \end{bmatrix} e^{c_1 k} + \begin{bmatrix} c_2 \\ -\zeta c_2 \end{bmatrix} e^{c_2 k} + \begin{bmatrix} 0 \\ \Gamma \end{bmatrix}$$  \hfill (29)

To inspect the convergence condition relaxation, we assume $\varphi_3 = 0$ and $\zeta \sim N(2, 2)$, where $N(2, 2)$ denotes the Gaussian distribution with the means and variance both equal to 2. Therefore, when $\zeta \geq 2$, $\|e_1\| = 0$ and $\|e_2\| \geq 1$. When $\zeta < 2$, $\|e_1\| < 1$ and $\|e_2\| < 1$. So, according to systems theory, if $\zeta < 2$, the system is convergent. If $\zeta \geq 2$, the system is not convergent. However, from the point of view of entropy theory, when $\zeta \geq 2$, $H(k)$ does not decrease. When $\zeta < 2$, $H(k)$ decreases. Because $E(\zeta < 2) = E(\zeta \geq 2) = 0.5$. So, $\lim_{k \to \infty} H(k) = 0.5H(1)$. The convergence region is illustrated in Fig. 3.

The convergence process based on entropy theory is plotted in Fig. 4. It is clear that even though individuals do not converge to a certain point, they converge to a certain area. Over the convergence process, $H(k)$ decreases to a certain value.

Using entropy theory, we derive the relaxed convergence
condition. If \( H(k) < H(1) \) as \( k \to \infty \), the system is convergent. This means that PDEA is convergent, if \( \mathcal{R}_c \cap \mathcal{R}_e \neq \emptyset \).

However, in practice, to guarantee the performance of PDEA, we suggest keeping \( \lim_{k \to \infty} H(k) \leq 0.5H(1) \).

4. Comparison with other EAs

Convergence analysis of EAs is an important and challenging issue. Markov chain model is proposed to study the convergence of GA. However, the convergence analysis is based on the assumption of infinite population and generation. PSO, BFO, and DE are analysed based on dynamic equations. However, the convergence condition of PSO is too strong, and the standard PSO is not convergent. The convergence analyses of BFO and DE are both based on the assumption that the objective function is differentiable.

Unlike the above analysis, PDEA convergence analysis does not require such strong assumptions. First, from the point of view of systems theory, we consider PDEA as a linear discrete-time state space system and prove that it is controllable. For PDEA, controllability is convergence. We derive the convergence condition of PDEA according to systems theory. Then we relax the convergence condition from the point of view of entropy theory. That can also be used to relax the convergence condition of DE, PSO, and BFO.

5. Numeric Evaluations

In this section, PDEA is applied to solve 16 benchmark functions [31] for numeric evaluations. These functions are selected from commonly used optimization test functions. Furthermore, PDEA is applied to an adaptive liquid level control system of a surge tank.

5.1. Evaluations with benchmark functions

The properties of these test functions are summarized in Table 1, where \( f(x^*) \) denotes the optimum of the benchmark functions.

GA, as a popular EA, is used as the comparison reference for PDEA. The crossover, mutation probabilities and encode accuracy of GA are set as 0.9, 0.1, and \( 10^{-4} \), respectively. The parameters of PDEA are set as below. \( \alpha \) and \( \beta \) are determined by \( pdc \) and \( pdh \). We set \( \alpha/\beta = pdc/pdh \), \( \alpha+\beta = 4 \), \( \max(\alpha/\beta) = 3 \), \( \min(\alpha/\beta) = 1/3 \), and \( \gamma = 1 \).

The simulation environment is Matlab version 2010b, run in Operating System of MS Windows 10, 64-bit, Processor of Intel Core I7-4790, Memory of 8GB DDR3 RAM.

To make the tests complete, 50 independent tests are carried out for each benchmark function. The population size is set as 100, maximum iteration is set 200. The performance metrics, including computation time, mean error, and variance, are listed in Table 1.

From the numeric simulation results, it is clear that PDEA is capable of finding the optimal or near optimal solutions despite of the diversity of benchmark functions, with GA as the comparison reference. In terms of the mean value, variance value, and computation time, the accuracy, reliability and computation speed of PDEA, are acceptable.

5.2. Adaptive liquid level control

Optimization methods can be used to estimate models and design controllers in adaptive control. Therefore, in order to evaluate the engineering performance, PDEA is applied to an adaptive liquid level control of a surge tank.

The purpose of applying PDEA in Fig. 5 is to learn the plant model during the operation of the indirect adaptive control system. To evaluate the learning capability of PDEA, the error between the model output and the plant output is the objective function as defined. During each interval, PDEA searches in the objective function space to find the plant model of minimum identification error. Then, the best model is selected and applied to the standard certainty-equivalence control law.

We consider the surge tank liquid level control problem as in [13]. The model is below

\[
\frac{dh(t)}{dt} = -\frac{d}{A(h(t))} + \frac{\bar{c}}{A(h(t))}u(t) \tag{30}
\]

where \( h(t) \) is the liquid level, \( u(t) \) is the input. \( A(h(t)) = ||\bar{a}h(t) + \bar{b}|| \) represents the tank cross-sectional area; \( \bar{a}, \bar{b}, \bar{c} \), and \( d \) are constants. However, the tank cross-sectional area is not known. We have to estimate the plant dynamics so that control can be designed for compensation.

The parameters of PDEA are set as follows. \( u = (u_1, u_2, u_3) \) is uniform distribution in \((0, 1)\). \( \alpha = \beta = 1 \), and \( \gamma = 0.5 \), which means we set \( pdc \) and \( pdh \) with the same weight. \( pdr \) has a half weight. Population size \( Np = 10 \). The maximum number of iteration \( Nmax = 10 \).

Each individual corresponds to the model parameter. The objective function is defined as the sum of squares of the 50 past identification error values. For parameter adjustment, PDEA searches the objective function space to find the best plant parameters. The tracking performance and the best objective function value are plotted in Fig.6. The reference input of liquid
Table 1: PDEA performances on benchmark functions

<table>
<thead>
<tr>
<th>Benchmark Function</th>
<th>Property</th>
<th>$f(x^*)$</th>
<th>GA</th>
<th>PDEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Variance</td>
<td>Time (s)</td>
<td>Mean</td>
</tr>
<tr>
<td>Ackley</td>
<td>0</td>
<td>0.8167</td>
<td>1.4899</td>
<td>0.3860</td>
</tr>
<tr>
<td>Griewank</td>
<td>0</td>
<td>0.0700</td>
<td>0.0060</td>
<td>0.3470</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>0</td>
<td>0.3419</td>
<td>0.2230</td>
<td>0.2910</td>
</tr>
<tr>
<td>Drop-Wave</td>
<td>-1</td>
<td>-0.9372</td>
<td>0.0020</td>
<td>0.3310</td>
</tr>
<tr>
<td>Bohachevsky</td>
<td>Bowl Shaped</td>
<td>0</td>
<td>0.8386</td>
<td>3.0724</td>
</tr>
<tr>
<td>Sphere</td>
<td>0</td>
<td>8.6769E-4</td>
<td>1.4583E-5</td>
<td>0.3060</td>
</tr>
<tr>
<td>Power Sum</td>
<td>Plate Shaped</td>
<td>0</td>
<td>0.0205</td>
<td>0.0048</td>
</tr>
<tr>
<td>Easom</td>
<td>Steep Drops</td>
<td>-1</td>
<td>-0.0730</td>
<td>0.0012</td>
</tr>
<tr>
<td>Michalewicz</td>
<td>-1.8013</td>
<td>3.6263E-5</td>
<td>3.7151E-5</td>
<td>0.3100</td>
</tr>
<tr>
<td>Six Hump</td>
<td>-1.0316</td>
<td>1.8835E-5</td>
<td>0.3190</td>
<td>-1.0316</td>
</tr>
<tr>
<td>Dixon &amp; Price</td>
<td>Valley Shaped</td>
<td>0</td>
<td>0.0292</td>
<td>0.0041</td>
</tr>
<tr>
<td>Rosenbrock’s</td>
<td>0</td>
<td>0.2101</td>
<td>0.1591</td>
<td>0.3240</td>
</tr>
<tr>
<td>Beale</td>
<td>0</td>
<td>0.0127</td>
<td>0.0120</td>
<td>0.3130</td>
</tr>
<tr>
<td>Branin</td>
<td>Others</td>
<td>0.3978</td>
<td>4.5087E-4</td>
<td>0.3080</td>
</tr>
<tr>
<td>Goldstein&amp;Price</td>
<td>3</td>
<td>3.0962</td>
<td>0.0460</td>
<td>0.2950</td>
</tr>
</tbody>
</table>

Figure 6: The response of adaptive controller.

level is a periodic square wave. For each square input, only at the first sampling interval, the tracking error is a little big. The maximum error value is 23.15%. But after 2 or 3 intervals, the tracking liquid level approaches the reference input quickly. The tracking error is almost zero.

6. Conclusions

In this paper, we have proposed a novel evolutionary algorithm called pseudo derivative evolutionary algorithm (PDEA). PDEA combines derivative with evolutionary algorithms. The advantage of PDEA is utilizing pseudo derivative to drive the optimization process. The convergence of PDEA is first analyzed based on systems theory and the convergence condition is derived. Next, this condition, which is too strong, is relaxed based on entropy theory. Different from common convergence analysis with other EAs, the convergence of PDEA does not require strong assumptions, such as differentiability and infinite population or generation. And the strict convergence region is also extended. Finally, we apply PDEA to both benchmark functions and adaptive control problem for evaluating the numeric and engineering performances. For the benchmark functions, PDEA shows strong ability in finding the optimal solution of different kinds of benchmark functions. For the adaptive control problem, PDEA is capable of identifying the plant model and tracking the reference input well.

References


