International Journal of Control, Automation and Systems 16(X) (2018) 1-13
http://dx.doi.org/10.1007/s12555-016-0338-6

ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

# A Two-stage State Transition Algorithm for Constrained Engineering Optimization Problems

**Jie Han, Chunhua Yang, Xiaojun Zhou*, and Weihua Gui**

**Abstract:** In this study, state transition algorithm (STA) is investigated into constrained engineering design optimization problems. After an analysis of the advantages and disadvantages of two well-known constraint-handling techniques, penalty function method and feasibility preference method, a two-stage strategy is incorporated into STA, in which, the feasibility preference method is adopted in the early stage of an iteration process whilst it is changed to the penalty function method in the later stage. Then, the proposed STA is used to solve three benchmark problems in engineering design and an optimization problem in power-dispatching control system for the electrochemical process of zinc. The experimental results have shown that the optimal solutions obtained by the proposed method are all superior to those by typical approaches in the literature in terms of both convergency and precision.

**Keywords:** Constrained engineering optimization, feasibility preference method, penalty function method, state transition algorithm.

## 1. INTRODUCTION

Many real-world problem arising from different fields, such as engineering design, structural optimization, controller design, very large scale integration (VLSI) design, economics, and location problems [1–5], can be regarded as constrained optimization problems (COPs) [6]. Without a loss of generality, a general constrained optimization problem (1) can be mathematically formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \le 0, i=1,\cdots,q, \\ & h_i(\boldsymbol{x}) = 0, i=q+1,\cdots,m, \\ & l_i \le x_i \le u_i, i=1,\cdots,n, \end{aligned} \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function, $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$ is the decision variable, $l_i$ and $u_i$ represent the lower bound and the upper bound of $x_i$, respectively. There are $q$ inequality constraints $g_i(\boldsymbol{x})(i=1,\cdots,q)$ and $m-q$ equality constraints $h_i(\boldsymbol{x})(i=q+1,\cdots,m)$. In general, the objective function is defined on a search space, $\mathbb{S}$, which is a n-dimensional rectangle in $\mathbb{R}^n$, and the proper domains of variables are defined by their lower and upper bounds. The set of solutions that satisfy all constraints is called feasible region, which is described as follows:

$$\mathbb{F} = \{\boldsymbol{x}\in\mathbb{R}^n | g_i(\boldsymbol{x}) \le 0, h_i(\boldsymbol{x}) = 0\},$$

and at any point $\boldsymbol{x}\in\mathbb{F}$, inequality constraints that satisfy $g_i(\boldsymbol{x}) = 0$ are called active constraints at $\boldsymbol{x}$. By extension, equality constraints are considered active at all points of $\mathbb{F}$.

In order to simplify the constrained problem, equality constraints are usually transformed into inequality constraints as follows:

$$|h_i(\boldsymbol{x})| - \varepsilon \le 0, \tag{2}$$

where $\varepsilon > 0$ is a given feasibility tolerance.

In order to verify the effectiveness and robustness of optimization algorithms, it is necessary to study various problems in complex COPs. In general, there are two major types of optimization problems: test functions and engineering design problems. The former type of problem is artificial problem which is useful to evaluate characteristics of optimization algorithms [7]. The engineering design optimization problems are all from practical industrial applications and each parameters have specific physical meanings. As one of the field which can be commonly encountered [8], constrained engineering optimization problem always has nonlinear objective functions and constraint functions, and the feasible region of such problems could be either one single bounded region or a collection of multiple disjointed region. Since the deterministic methods are mainly depended on the gradient information,

Manuscript received June 4, 2016; revised March 16, 2017; accepted July 24, 2017. Recommended by Associate Editor Ho Jae Lee under the direction of Editor Yoshito Ohta. This work was supported by the National Natural Science Foundation (NNSF) of China (61503416, 61533021), the 111 Project (B17048), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 61621062) the Innovation-Driven Plan in Central South University, and the Independent Exploration Innovation Program for Postgraduate Students of Central South University(Grant No. 2017zzts136).

Jie Han, Chunhua Yang, Xiaojun Zhou, and Weihua Gui are with the School of Information Science and Engineering, Central South University, Changsha 410083, P. R. China (e-mails: {hanjie, ychh, michael.x.zhou, gwh}@csu.edu.cn).
* Corresponding author.

they are easily to drop into local optimum when solving constrained engineering optimization problems. In order to deal with this type of problem, a huge number of meta-heuristic algorithms have been proposed in last decades. For instance, Coello [9] introduced a GA-based technique that used self-adaptive penalty approach to find the optimum of these constrained engineering optimization problems. Guedria [10] compared the performance of different algorithms, like HGA (hybrid genetic algorithm) [11], CPSO (co-evolutionary particle swarm optimization algorithm) [12], CAEP (cultural algorithms with evolutionary programming) [13], SC (society and civilization algorithm) [14], WCA (water cycle algorithm) [15] and MBA (mine blast algorithm) [16], which are mainly tested on engineering optimization problems. Although the benchmark problems in engineering optimization have the known global optimum and are hard to improve the solution precision, there are still enough space for more effective algorithms that cost less computational effort and obtain more accurate results.

In recent years, state transition algorithm (STA) [17–21] inspiring from the concepts of state and state transition has been proposed. Unlike most of the existing evolutionary algorithms, the basic STA is based on individual iterative method. In basic STA, four state transformation operators named rotation, translation, expansion and axesion are used for forming a regular neighborhood start from an current state, since there exists stochastic properties in the state transition matrices, and then a sampling technique is used to create a candidate state set. These four transformation operators have the ability of both local and global search, and in the mean while, they are alternative used in STA [22]. The effectiveness and efficiency of basic STA have been testified when compared with other state-of-art intelligent optimization methods, like particle swarm optimization (PSO) and genetic algorithm (GA) [20]. Thus, the constrained engineering optimization problems using STA are studied in this paper.

The remainder of the paper is organized as follows. Section 2 presents the detail of basic STA and in Section 3, the advantages and disadvantages of two well-known constraint-handling techniques are discussed and a two-stage strategy is proposed to solve constrained optimization problem. Then, engineering design optimization problems are given to evaluate the performance of STA in Section 4. Finally, the paper is summarized in Section 5.

## 2. BASIC STATE TRANSITION ALGORITHM

Basic state transition algorithm (STA) is a recently proposed optimization algorithm that is based on the concepts of state transition and state space representation in control theory. In the process of solving optimization problems by STA, every solution is regarded as a state, and the update of current solution is treated as a state transition. In general, the framework of state transition algorithm can be defined as follows:

$$\begin{cases} \boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k + B_k \boldsymbol{u}_k, \\ y_{k+1} = f(\boldsymbol{x}_{k+1}), \end{cases} \tag{3}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ represents a state, corresponding to a candidate solution; $A_k$ and $B_k$ are state transition matrices with appropriate dimensions, which usually stand for transformation operators; $\boldsymbol{u}_k$ is a function of $\boldsymbol{x}_k$ and historical states; $f$ is the objective function or evaluation function.

In order to generate candidate solution, four special state transformation operators are designed in basic STA.

1) Rotation transformation

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \frac{1}{n\|\boldsymbol{x}_k\|_2} R_r \boldsymbol{x}_k, \tag{4}$$

where $\alpha$ is a positive constant, called rotation factor; $R_r \in \mathbb{R}^{n\times n}$, is a random matrix with its elements belonging to the range of [-1, 1] and $\|\cdot\|_2$ is the 2-norm of a vector. The rotation transformation has the function of local search which means it can generate candidate solution in a domain of hypersphere with a given radius $\alpha$.

2) Translation transformation

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \beta R_t \frac{\boldsymbol{x}_k - \boldsymbol{x}_{k-1}}{\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|_2}, \tag{5}$$

where $\beta$ is a positive constant, called translation factor; $R_t \in \mathbb{R}$ is a random variable with its elements belonging to the range of [0,1]. The translation transformation is designed for a line search which is only performed when a better solution can be found by other transformation operators.

3) Expansion transformation

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \gamma R_e \boldsymbol{x}_k, \tag{6}$$

where $\gamma$ is a positive constant, called expansion factor; $R_e \in \mathbb{R}^{n\times n}$ is a random diagonal matrix with its elements obeying the Gaussian distribution. The expansion transformation is for global search which can search in the whole space with probability.

4) Axesion transformation

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \delta R_a \boldsymbol{x}_k, \tag{7}$$

where $\delta$ is a positive constant, called axesion factor; $R_a \in \mathbb{R}^{n\times n}$ is a random diagonal matrix with its elements obeying the Gaussian distribution and only one random position having nonzero value. The axesion transformation is proposed to strengthen the single dimensional search as well as global search.

Pseudo-code of basic STA for unconstrained optimization problems is given in Algorithm 1.

In Algorithm 1, *SE* is called search enforcement, representing the times of transformation by a certain operator.

**Algorithm 1:** Pseudo-code of basic STA for unconstrained optimizaiton problems.

**Input:**
*maxiter*: the maximum number of iterations
*SE*: search enforcement
*Best*: the initial solution
**Output:**
*Best*∗: the optimal solution

**for** $iter = 1$ **to** $maxiter$ **do**
  **if** $\alpha < \alpha_{\min}$ **then**
    $\alpha \leftarrow \alpha_{\max}$
  **end if**
  $Best \leftarrow$ expansion(funfcn,$Best, SE, \cdots$)
  $Best \leftarrow$ rotation(funfcn,$Best, SE, \cdots$)
  $Best \leftarrow$ axesion(funfcn,$Best, SE, \cdots$)
  $\alpha \leftarrow \frac{\alpha}{fc}$
**end for**
$Best* \leftarrow Best$

**Algorithm 2:** Pseudo-code of expansion transformation in Algorithm 1.

**Input:**
*oldBest*: the best solution in the last transformation
**Output:**
*Best*: the best solution

$fBest \leftarrow$ feval(funfcn,$oldBest$)
$State \leftarrow$ op_expand($Best, SE, \cdots$)
$newBest \leftarrow \min(State)$
$fGBest \leftarrow$ feval(funfcn,$newBest$)
**if** $fGBest < fBest$ **then**
  $fBest \leftarrow fGBest$
  $Best \leftarrow newBest$
  $State \leftarrow$ op_translate($oldBest, Best, SE, \cdots$)
  $newBest \leftarrow \min(State)$
  $fGBest \leftarrow$ feval(funfcn,$newBest$)
  **if** $fGBest < fBest$ **then**
    $fBest \leftarrow fGBest$
    $Best \leftarrow newBest$
  **end if**
**end if**

In STA, the rotation factor $\alpha$ is reducing periodically from a maximum value $\alpha_{\max}$ to a minimum value $\alpha_{\min}$ in an exponential way with base $fc$, which is called lessening coefficient. The changeable rotation factor can not only speed up the process of finding optimal solution but also increase the accuracy of the solution. For unconstrained optimization problems, the "greedy criterion" is used to accept a new best solution. Algorithm 2 illustrates the process of expansion function in Algorithm 1 which is similar to the process of rotation function and axesion function.

To achieve a better understanding of the detailed steps of state transition algorithm, the readers can download the STA toolbox via the following link: `http://www.mathworks.com/matlabcentral/fileexchange/52498-state-transition-algorithm`, and Zhou has shown how to use the Matlab toolbox for continuous state transition algorithm [23].

## 3. CONSTRAINT-HANDLING TECHNIQUES

Since the basic STA is essentially an unconstrained optimization procedure, it is necessary to find additional mechanisms to deal with the constraints. There are a large number of ways to cope with constraints, and readers can refer to and the references therein for details [24–27]. In this study, we focus on the following two well-known constraint-handling techniques widely used in COPs.

### 3.1. Penalty function method

The most common way to handle constraints is the penalty function method. The idea behind this method is to transform constrained optimization problems into unconstrained ones by adding certain terms to the objective function based on the amount of constraint violation[15]:

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{q} p_i G_i + \sum_{j=q+1}^{m} p_j H_j, \tag{8}$$

where $p_i, p_j$ are penalty factors,

$$G_i = \max\{0, g_i(\boldsymbol{x})\}^{\kappa}, H_j = \max\{0, |h_j(\boldsymbol{x})| - \varepsilon\}^{\kappa}, \tag{9}$$

where $\kappa$ is normally 1 or 2.

### 3.2. Feasibility preference method

Based on the preference of feasible solutions over infeasible solutions, Deb [28] proposed the feasibility preference method, in which, two solutions are compared according to the following criteria:

1) any feasible solution is preferred to any infeasible solution;
2) among two feasible solutions, the one having better objective function value is preferred;
3) among two infeasible solutions, the one having smaller constraint violation is preferred.

### 3.3. Advantages and disadvantages

The main limitation of the penalty function method is that it requires an appropriate selection of the penalty factors. If the penalty factors are large, it is more likely to obtain a feasible solution; however, in this case, the exploration ability is discouraged. If the penalty factors are as small as possible, they are ideal to get an optimum located at the boundary of the feasible region, but it may risk an infeasible solution in the end for other cases.

A big advantage of the feasibility preference method is that it is most likely to obtain a feasible solution since an infeasible solution can never be accepted once a feasible solution is captured. However, it neglects some valuable information contained in infeasible solutions and it becomes extremely difficult to find a better feasible solution than current feasible solution at the later stage of the iteration process.

### 3.4. Two-stage strategy for constrained STA

The above mentioned constraint-handling techniques, which can be considered as criteria to choose the "best" state, are incorporated into basic STA. Then, to draw on each other's strengths, we give a two-stage strategy for constrained STA, namely, in the early stage of an iteration process, the feasibility preference method is adopted to make the solution tend to be feasible. In the later stage, we change it to the penalty function method so as to make use of the information of all candidate solutions and obtain the best one.

To quantify the constraint violation ($G(x)$)in the feasibility preference method, the following formula is used

$$G(\boldsymbol{x}) = \sum_{i=1}^{q} \max\{0, g_i(\boldsymbol{x})\} + \sum_{j=q+1}^{m} \max\{0, |h_j(\boldsymbol{x})| - \varepsilon\}, \tag{10}$$

which can be considered as the distance of a solution $\boldsymbol{x}$ from the boundaries of the feasible space.

The main procedure of the constrained STA with a two-stage strategy is given in Algorithm 3.

In Algorithm 3, *maxiter* is the maximum number of iterations, $\rho \in [0,1]$ is used to control the ratio of feasibility preference and penalty function methods in an iteration process, and $flag$ = 1 means that the feasibility preference method is adopted while penalty function method is used when $flag$ = 0. The detailed selection mechanism is described in Algorithm 4, and the "operator" represents those four transformation operators in STA.

The two-stage strategy not only can get the optimal solution but also has better performance than single strategy [7]. Since the prominent search ability and outstanding solution precision of two-stage strategy, we use this method to deal with engineering design optimization problems.

## 4. APPLICATIONS IN ENGINEERING CONSTRAINED PROBLEMS

In this paper, the STA with two-stage strategy is used to solve constrained engineering optimization problems. With the purpose of evaluating the performance of this algorithm, we chose 3 benchmark problems in engineering design and an optimization problem in power-dispatching control system for the electrochemical process of zinc.

**Algorithm 3:** Pseudo-code of constrained STA with a two-stage strategy.

**Input:**
*maxiter*: the maximum number of iterations
*SE*: search enforcement
$\rho$: the ratio of two methods
*Best*: the initial solution
**Output:**
*Best*∗: the optimal solution

```
Best ← generate a initial solution
for iter = 1 to maxiter do
    if α < α_min then
        α ← α_max
    end if
    if iter < ρ × maxiter then
        flag=1
    else flag=0
    end if
    Best ← expansion(funfcn,Best,SE,flag···)
    Best ← rotation(funfcn,Best,SE,flag···)
    Best ← axesion(funfcn,Best,SE,flag···)
    α ← α/fc
end for
Best∗ ← Best
```

**Algorithm 4:** Pseudo-code of two-stage selection mechanism in Algorithm 3.

```
State ← operator(Best,SE,flag,···)
if flag = 1 then
    Best ← slection1(State)
else
    Best ← slection2(State)
end if
```

These problems consist of objective functions and constraints with various types and nature, such as quadratic, cubic, polynomial and nonlinear.

In the same time, several well-known optimizers are used for comparison. It is worth pointing that the results calculated by built-in GA in MATLAB (which is based on globally convergent augmented lagrangian barrier technique) are obtained during this study. All of the methods are run under the MATLAB (Version R2010b) software platform. The parameter settings of STA are analyzed in [20] based on numerical experiments, which are given as follows: $\alpha_{\max} = 1$, $\alpha_{\min} =$ 1e-4, $\beta = 1$, $\gamma = 1$, $\delta = 1$, $fc = 2$, $SE = 30$. For the parameters in handling constraints, $\kappa$ = 2 is used, and the penalty factors of three benchmark problems are fixed at *realmax* (the largest double precision floating point number in MATLAB) to restrict an infeasible solution as heavily as possible, while in power-dispatching control system, we set the penalty factor as 1e6 to satisfy practical application. The ratio

$\rho = 0.5$ is used in this study so that in the half later stage the penalty function method is adopted. The feasibility tolerance $\varepsilon$ = 1e-4 is used in this study. The maximum number of iterations is 2000. For fairness, the population size is 30 and the maximum number of generations is 6000 in GA. All methods are run independently for 20 trails. To evaluate the performance of these methods, the experimental results include the best solution, the best, mean, standard deviation (st.dev) and worst objective function values in 20 runs.

### 4.1. Welded beam design

The welded beam design is a classic test problem for constrained optimization which is aimed to minimize the fabricating cost of welded beam ($f$) and satisfy constraints about shear stress ($\tau$), bending stress in the beam ($\theta$), buckling load on bar ($P_c$), end deflection of the beam ($\delta$) and side constraints [30]. The variables associated with this problem are: $x_1$ = thickness of the weld ($h$), $x_2$ = length of the welded joint ($l$), $x_3$ = width of the beam ($t$), $x_4$ = thickness of the beam ($b$). Fig. 1 shows the welded beam structure.

The mathematical formulation of this problem is as follows:

$$\min f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14+x_2) \quad (11)$$
$$\text{s.t. } g_1(X) = \tau(X) - \tau_{max} \le 0,$$
$$g_2(X) = \sigma(X) - \sigma_{max} \le 0,$$
$$g_3(X) = x_1 - x_4 \le 0,$$
$$g_4(X) = 0.125 - x_1 \le 0,$$
$$g_5(X) = \delta(X) - 0.25 \le 0,$$
$$g_6(X) = P - P_c(X) \le 0,$$
$$g_7(X) = 0.10471x_1^2 + 0.04811x_3x_4(14x_2) - 5 \le 0,$$

where

$$0.1 \le x_1 \le 2, 0.1 \le x_2 \le 10,$$
$$0.1 \le x_3 \le 10, 0.1 \le x_4 \le 2,$$
$$\tau(X) = \sqrt{\tau_1^2 + 2\tau_1\tau_2(\frac{x_2}{2R}) + \tau_2^2},$$
$$\tau_1 = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau_2 = \frac{MR}{J},$$
$$M = P(L + \frac{x_2}{2}),$$
$$J(X) = 2\{\sqrt{2}x_1x_2[\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2]\},$$
$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2},$$
$$\sigma(X) = \frac{6PL}{x_4x_3^2}, \quad \delta(X) = \frac{6PL^3}{Ex_3^3x_4},$$
$$P_c(X) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}),$$

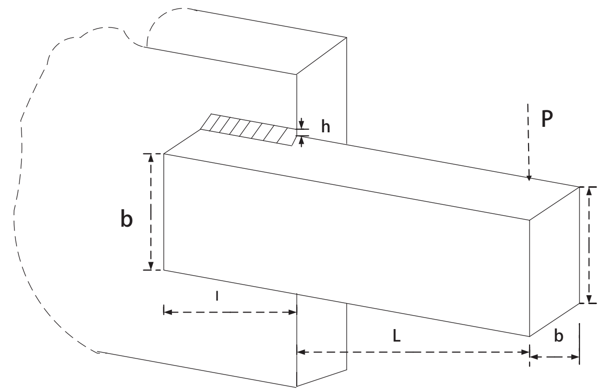


Fig. 1. The design of welded beam,

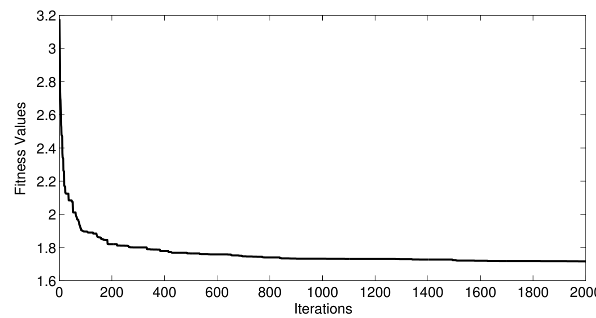


Fig. 2. The iterative curve of welded beam design.

$$G = 12 \times 10^6 \text{ ps}, \quad E = 30 \times 10^6 \text{ ps},$$
$$P = 6000 \text{ lb}, \quad L = 14 \text{ in}.$$

There are many algorithms used to optimize this problem, such as: HGA, CPSO, hybrid Nelder-Mead simplex search and particle swarm optimization (NM-PSO) [31], CAEP, SC, WCA, MBA, improved accelerated PSO (IAPSO) [10] and so on. The optimal solution obtained by STA is $X$ = [0.20532536, 3.26035648, 9.03664424, 0.20572991] with corresponding function value equal to $f(X)$ = 1.69563970 and constraints $[g_1(X), g_2(X), \dots, g_7(X)]$ = [-0.10520197, -0.17417862, -4.04330102, -3.45179021, -0.08032536, -0.22831066, -0.0339737].

The results of the experiments are shown in Table 1, Table 2 and Fig. 2. Table 1 shows the comparison of the best solution of STA and other previously reported studies. Table 2 shows the statistical optimization results of these algorithms. The iterative curve of average results obtained by STA is given in Fig. 2.

From Table 1, STA offers better results compared to any other earlier solutions reported in the literature. And the best results obtained by STA are all in the inner space of feasible region instead of the boundary of feasible region, which shows the robustness of STA. And in Table 2, the mean solution detected by STA is better than other solutions found by other techniques. Fig. 2 illustrates the fitness values with respect to the number of iterations for the

Table 1. Comparison of the best solution of welded beam design problem.

| DV[1] | He [12] | Coello [13] | Yuan [11] | Zahara [31] | Eskandar [15] | Sadollah [16] | Guedria [10] | STA |
|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0.202369 | 0.205700 | 0.205700 | 0.205830 | 0.205728 | 0.205729 | 0.205730 | **0.20532536** |
| $x_2$ | 3.544214 | 3.470500 | 3.470500 | 3.468338 | 3.470522 | 3.470493 | 3.470489 | **3.26035648** |
| $x_3$ | 9.048210 | 9.036600 | 9.036600 | 9.036624 | 9.036620 | 9.036626 | 9.036624 | **9.03664424** |
| $x_4$ | 0.205723 | 0.205700 | 0.205700 | 0.205730 | 0.205729 | 0.205729 | 0.205730 | **0.20572991** |
| $g_1(x)$ | -13.655547 | -0.000472 | -769.3403 | -0.025250 | -0.034128 | -0.001614 | -1.05e-10 | **-0.10520197** |
| $g_2(x)$ | -78.814077 | -0.001561 | 4.481548 | -0.053122 | -3.49e-05 | -0.016911 | -6.91e-10 | **-0.17417862** |
| $g_3(x)$ | -3.35e-03 | 0.000000 | 0.000000 | 0.000100 | -1.19e-06 | -2.40e-07 | -7.66e-15 | **-4.04330102** |
| $g_4(x)$ | -3.424572 | -3.432984 | -3.433213 | -3.433168 | -3.432980 | -3.432982 | -3.432984 | **-3.45179021** |
| $g_5(x)$ | -0.077369 | -0.080730 | -0.080700 | -0.080830 | -0.080728 | -0.080729 | -0.080730 | **-0.08032536** |
| $g_6(x)$ | -0.235595 | -0.235540 | -0.235538 | -0.235540 | -0.235540 | -0.235540 | -0.235540 | **-0.22831066** |
| $g_7(x)$ | -4.472858 | -0.000779 | 2.603347 | -0.031555 | -0.013503 | -0.001464 | -5.80e-10 | **-0.03397937** |
| $f(x)$ | 1.728024 | 1.724852 | 1.724852 | 1.724717 | 1.724856 | 1.724853 | 1.724852 | **1.69563970** |

[1]DV: stands for design variables.

Table 2. Comparison of the statistical results of welded beam design problem.

| Method | Best | Mean | Worst | SD |
|---|---|---|---|---|
| Coello [13] | 1.724852 | 1.971809 | 3.179709 | 4.43e-01 |
| He [12] | 1.728024 | 1.748831 | 1.782143 | 1.29e-02 |
| Liu [32] | 1.724852 | 1.724852 | 1.724852 | 6.70e-16 |
| Zahara [31] | 1.724717 | 1.726373 | 1.733393 | 3.50e-03 |
| Lampinen [33] | 1.733461 | 1.768158 | 1.824105 | 2.21e-02 |
| Eskandar [15] | 1.724856 | 1.726427 | 1.744697 | 4.29e-03 |
| Kashan [34] | 1.724852 | 1.724852 | 1.724852 | 7.11e-15 |
| Sadollah [16] | 1.724853 | 1.724853 | 1.724853 | 6.94e-19 |
| Guedria [10] | 1.724852 | 1.724853 | 1.724862 | 2.02e-06 |
| **STA** | **1.6956397** | **1.7160908** | **1.7530472** | **1.83e-02** |

welded beam design problem. Note that STA shows better converging behavior due to the efficiency of its searching mechanism and handling technique.

## 4.2. Pressure vessel design

This problem is aimed to design a cylindrical vessel which is capped at both ends by hemispherical heads, as shown in Fig. 3 [35]. The total costs including the cost of material, forming and welding are optimized to find the minimum value. There are four design variables: $x_1$ = thickness of the pressure vessel ($T_s$), $x_2$ = thickness of the head ($T_h$), $x_3$ = inner radius of the vessel ($R$), $x_4$ = length of the vessel without heads ($L$).

The mathematical formulation of this problem is as follows:

$$\begin{aligned} \min f(X) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\ &+3.1661x_1^2x_4 + 19.84x_1^2x_3, \qquad (12)\\ \text{s.t. } g_1(X) &= -x_1 + 0.0193x_3 \le 0, \\ g_2(X) &= -x_2 + 0.00954x_3 \le 0, \\ g_3(X) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0, \end{aligned}$$

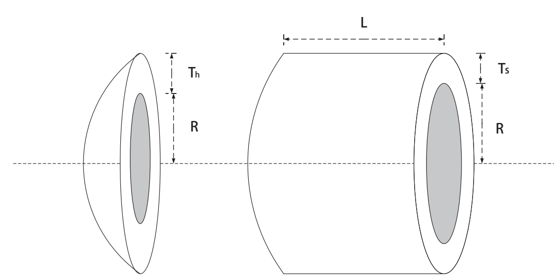


Fig. 3. The design of pressure vessel.

$$g_4(X) = x_4 - 240 \le 0,$$

where $1 \times 0.0625 \le x_1, x_2 \le 99 \times 0.0625$, $10 \le x_3, x_4 \le 200$.

Similarly, the pressure vessel design problem has been studied previously using co-evolutionary differential evolution (CDE) [41], CPSO, hybrid particle swarm optimization (HPSO) [37], NM-PSO, WCA, MBA and so on. The best results obtained by STA is $f(X) = 5886.45436$ corresponding to $X$ = [0.77854, 0.38484, 40.3389, 199.775349] and constraints [$g_1(X)$, $g_2(X)$,

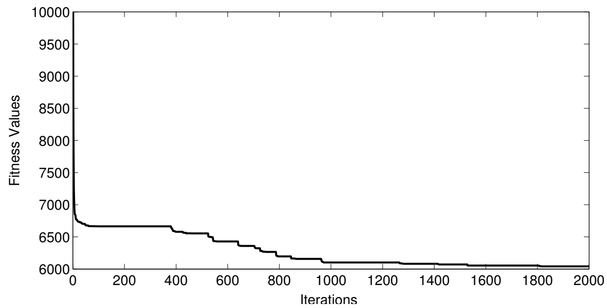


Fig. 4. The iterative curve of pressure vessel design.

$\ldots$, $g_4(X)$] = [-2.1123e-07, -6.2039e-06, -110.9246, -40.2465].

The comparison of these algorithms and STA are given in Table 3 and Table 4. It worth to note that the feasibility rate of GA is only 40 percent, which means there are only 8 feasible solutions in 20 runs. However, the feasibility rate of STA can reach 100 percent, which indicates STA has high reliability. As can be seen from Table 3 and Table 4, STA converge once again to a new solution better than others. Moreover, the best solution obtained by Zahara violates the first and the second constraints and then can not be compared to other best solutions. Fig. 4 depicts the reduction of fitness values versus the number of iterations of STA. In terms of the information of Fig. 4, we can find that STA has the strong ability of exploration and exploitation which can get rid of the stagnation point and obtain the global optimal point.

### 4.3. Tension/Compression spring design

This problem is aimed to minimize the weight of a tension/compression spring [41, 42], as shown in Fig. 5. In this design, the constraints include minimum deflection, shear stress, surge frequency, limits on outside diameter and some variables. There are three design variables: $x_1$ = mean coil diameter, $x_2$ = the wire diameter, and $x_3$ = the number of active coil.

The mathematical formulation of this problem is as follows:

$$\min f(X) = (x_3+2)x_2x_1^2 \tag{13}$$

$$\text{s.t. } g_1(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0,$$

$$g_2(X) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0,$$

$$g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0,$$

$$g_4(X) = \frac{x_1+x_2}{1.5} - 1 \leq 0,$$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, $2 \leq x_3 \leq 15$.

There are many researchers using different optimization algorithms to solve this problem. These algorithms include genetic algorithm based co-evolution model (GA1) [9], CPSO, SC, improved harmony search (IHS) [43], gaussian quantum-behaved particle swarm optimization (G-QPSO) [38] and so on. The best results obtained by STA is $f(X) = 0.01266534$ corresponding to decision variable $X$ = [ 0.0516800, 0.3565001, 11.3018335] and constraints $[g_1(X), g_2(X), \ldots, g_4(X)]$ = [-6.218e-06, -1.691e-06, -4.0533150, -0.7278799].

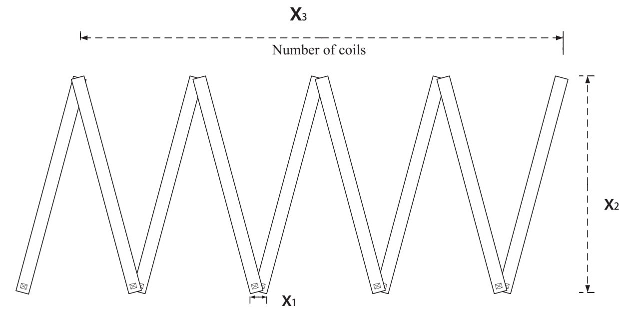


Fig. 5. The design of tension/compression spring.

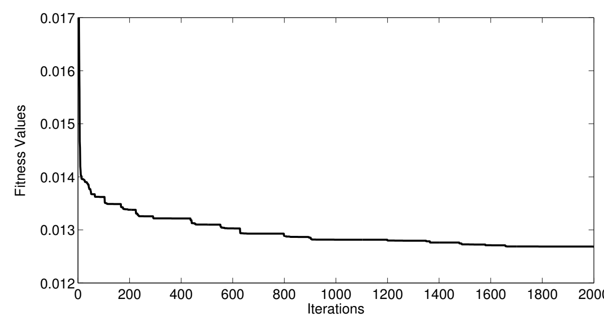


Fig. 6. The iterative curve of tension/compression spring design

The results of the experiments are shown in Table 5, Table 6 and Fig. 6. The STA still have better solution than other algorithms. According to the statistical results in Table 5, STA is the most stable algorithm with the smallest value of the standard deviation. Fig. 6 also shows the same results which the STA has both good whole astringency and fast convergence speed. It is worth pointing that the fitness value of STA is still decreasing in the later iterations and this phenomenon shows that STA can improve the precision of the results and find better solutions of this engineering problem.

### 4.4. Power-dispatching control system in the electrochemical process of zinc

In metallurgical industry, the electrochemical process of zinc (EPZ) is a large power-consuming process that accounts for 80% of the total electrical energy consumption of hydrometallurgy process. According to time-sharing price counting policy of electric power, if the electrochemical process of zinc runs with low current density in the

Table 3. Comparison of the best solution of pressure vessel design problem.

| DV | Huang [41] | He [12] | Zahara [31] | Coelho [38] | Sadollah [16] | Guedria [10] | GA[1] | STA |
|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0.8125 | 0.8125 | 0.8036 | 0.8125 | 0.7802 | 0.8125 | 0.8099 | **0.77854** |
| $x_2$ | 0.4375 | 0.4375 | 0.3972 | 0.4375 | 0.3856 | 0.4375 | 0.4004 | **0.38484** |
| $x_3$ | 42.0984 | 42.0913 | 41.6392 | 42.0984 | 40.4292 | 42.0984 | 41.9616 | **40.3389** |
| $x_4$ | 176.6376 | 176.7465 | 182.4120 | 176.6372 | 198.4964 | 176.6366 | 178.3500 | **199.775349** |
| $g_1(x)$ | -6.67e-07 | -1.37e-06 | 3.65e-05 | -8.79e-07 | 0 | -4.09e-13 | -2.09e-05 | **-2.1123e-07** |
| $g_2(x)$ | -3.58e-02 | -3.59e-04 | 3.79e-05 | -3.58e-02 | 0 | -3.58e-2 | -5.39e-05 | **-6.2039e-06** |
| $g_3(x)$ | -3.705123 | -118.7687 | -1.5914 | -0.2179 | -86.3645 | -1.39e-07 | -54.8632 | **-110.9246** |
| $g_4(x)$ | -63.3623 | -63.2535 | -57.5879 | -63.3628 | -41.5035 | -63.3634 | -61.65 | **-40.2465** |
| $f(x)$ | 6059.7340 | 6061.0777 | 5930.3137 | 6059.7208 | 5889.3216 | 6059.7143 | 5942.2806 | **5886.45436** |

[1]Numerical results obtained by GA are carried out during this study.

Table 4. Comparison of the statistical results of pressure vessel design problem.

| Method | Best | Mean | Worst | SD |
|---|---|---|---|---|
| He [12] | 6061.0777 | 6147.1332 | 6363.8041 | 86.4500 |
| Zahara [31] | 5930.3137 | 5946.7901 | 5960.0557 | 9.1610 |
| Coelho [38] | 6059.7208 | 6440.3786 | 7544.4925 | 448.4711 |
| Huang [41] | 6059.7340 | 6085.2303 | 6371.0455 | 43.0130 |
| Eskandar [15] | 6059.8553 | 6070.5884 | 6090.6114 | 11.3753 |
| Sadollah [16] | 5889.3216 | 6200.6477 | 6392.5062 | 160.34 |
| Guedria [10] | 6059.7143 | 6068.7539 | 6090.5314 | 14.0057 |
| GA | 5942.2806 | 6095.8824 | 6344.7175 | 126.6900 |
| **STA** | **5886.45436** | **6042.30370** | **6377.87825** | **150.8838** |

Table 5. Comparison of the best solution of tension/compression spring design problem.

| DV | Belegundu [40] | Arora [39] | Coello [9] | Saini [44] | Ray [14] | Mahdavi [43] | He [12] | STA |
|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0.05 | 0.053396 | 0.051480 | 0.050417 | 0.052160 | 0.051154 | 0.051728 | **0.0516800** |
| $x_2$ | 0.315900 | 0.399180 | 0.351661 | 0.321532 | 0.368159 | 0.349871 | 0.357644 | **0.3565001** |
| $x_3$ | 14.25 | 9.185400 | 11.632201 | 13.979915 | 10.648442 | 12.076432 | 11.244543 | **11.3018335** |
| $g_1(x)$ | -0.001267 | -0.001234 | -0.003337 | -0.001926 | -7.45e-09 | 0.000000 | -0.000845 | **-6.218e-06** |
| $g_2(x)$ | -0.003782 | -0.000018 | -0.000110 | -0.012944 | -3.68e-09 | -0.000007 | -1.26e-05 | **-1.691e-06** |
| $g_3(x)$ | -3.938302 | -4.123832 | -4.026318 | -3.899433 | -4.075805 | -4.027840 | -4.051300 | **-4.0533150** |
| $g_4(x)$ | -0.756067 | -0.698283 | -0.731239 | -0.752034 | -0.719787 | -0.736572 | -0.727090 | **-0.7278799** |
| $f(x)$ | 0.0128334 | 0.0127303 | 0.0127048 | 0.0130603 | 0.0126692 | 0.0126706 | 0.0126747 | **0.01266534** |

Table 6. Comparison of the statistical results of tension/compression spring design problem.

| Method | Best | Mean | Worst | SD |
|---|---|---|---|---|
| Coello [9] | 0.01270478 | 0.01276920 | 0.01282208 | 3.9390e-05 |
| Coello [45] | 0.0126810 | 0.012742 | 0.012973 | 5.900e-05 |
| Ray [14]] | 0.0126692 | 0.0129227 | 0.0167172 | 5.1985e-05 |
| He [12] | 0.0126747 | 0.012730 | 0.012924 | 5.1985e-05 |
| Monts [46] | 0.012698 | 0.13461 | 0.164850 | 9.6600e-04 |
| coelho [38] | 0.012669 | 0.013854 | 0.018127 | 1.341e-03 |
| Lampinen [33] | 0.012670 | 0.012703 | 0.012790 | 2.700e-05 |
| **STA** | **0.01266534** | **0.01268592** | **0.01272968** | **2.1672e-05** |

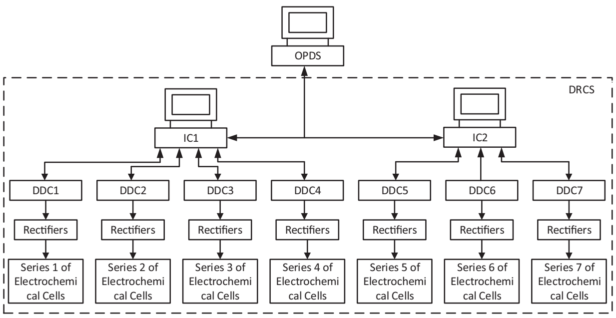


Fig. 7. The distributed architecture of optimal power-dispatching control system.

period of high price, and with high current density in the period of low price, then the cost of power consumption will be decreased. However, if the current density is too high or too low with respect to the technological requirements, it will lead to high power consumption and low current-efficiency. So, it is necessary to seek for optimal power-dispatching control system (OPDCS) in different pricing periods [47,48].

In this paper, we take example of the electrolytic zinc process in Zhuzhou Smeltery, located in Hunan Province, China. This problem aims to minimize the cost of power consumption and satisfies the constraints about daily output and technic parameters [49]. The distributed architecture of OPDCS is shown in Fig. 7. The OPDCS consists of an optimal power-dispatching system (OPDS) and a distributed rectifier control system (DRCS). The DRCS is composed of two industrial computers (IC1 and IC2), seven direct digit controllers (DDC, including DDC1, DDC2,..., and DDC7), rectifiers and so on. Each DDC controls the converting process in each series of cells [48].

In this problem, there are four pricing periods everyday and each period has seven plants to implement electrochemical processes of zinc. So, there are 28 design variables : $Dk_{ij}$ = current density (A/m$^2$), $i = 1,2,3,4$ = the number of period, $j = 1,2,3,\cdots,7$ = the number of plant. The mathematical formulation of this problem at present is as follows:

$$\min J(Dk) = \sum_{i=1}^{4} PW_i \times T_i \times P_i + J_0 \quad (14)$$

$$\text{s.t. } h(Dk) = \sum_{i=1}^{4}\sum_{j=1}^{7} q \times Dl_{ij} \times \eta_{ij} \times T_i = G_0,$$

where

$$PW_i = \sum_{j=1}^{7} U_{ij} \times Dl_{ij} \times N_j, Dl_{ij} = Dk_{ij} \times B_j \times S_0,$$

$$U_{ij} = a_0 + a_1 \times Dk_{ij},$$

$$\eta_{ij} = b_0 + b_1 \times Dk_{ij} + b_2 \times Dk_{ij}^2 + b_3 \times Dk_{ij}^3 + b_4 \times Dk_{ij}^4,$$

$$T_i = \begin{bmatrix} 5 & 4 & 7 & 8 \end{bmatrix},$$

$$P_i = 0.3059 \times \begin{bmatrix} 1 & 1.6 & 1.4 & 0.35 \end{bmatrix}, G_0 = 960.$$

$$N_j = \begin{bmatrix} 240 & 240 & 246 & 192 & 208 & 208 & 208 \end{bmatrix},$$

$$B_j = \begin{bmatrix} 34 & 46 & 54 & 56 & 56 & 57 & 57 \end{bmatrix},$$

$$S_0 = 1.13, q = 1.2202, a_0 = 2.76284, a_1 = 0.00093,$$

$$b_0 = 0.785037, b_1 = 5.855 \times 10^{-4}, b_2 = 2 \times 10^{-6},$$

$$b_3 = 3.2094 \times 10^{-9}, b_4 = -1.9052 \times 10^{-12},$$

$$Dk_{\min} \le Dk \le Dk_{\max}, Dk_{\min} = 200, Dk_{\max} = 650.$$

In these formulations,

$PW_i$ – the electrical load ($kW$) in the $i$th period.

$T_i$ – the hours ($h$) of the $i$th period.

$P_i$ – the electricity price (¥$/kWh$) of the $i$th period.

$U_{ij}$ – the cell voltage ($V$) of the $j$th plant in the $i$th period, $a_0$ and $a_1$ are obtained by recursive least squares method.

$Dl_{ij}$ – the magnitude of the current ($A$) of the electrolysis process of the $j$th plant in the $i$th period.

$N_j$ – the number of cells in the $j$th plant.

$B_j$ – the number of plates in a cell in the $j$th plant.

$S_0$ – the area ($m^2$) of a negative plate.

$G_0$ – the expected daily output of zinc ($t$).

$\eta_{ij}$ – the current-efficiency of the $j$th plant in the $i$th period, $b_0$, $b_1$, $b_2$, $b_3$ and $b_4$ are also obtained by recursive least squares method.

$q$ – the electrochemical equivalent of zinc ($g/Ah$).

$Dk_{\min}$ – the allowed minimum current density ($A/m^2$) of the electrochemical process of zinc to avoid dissolving zinc deposited on cathodes at too low current density.

$Dk_{\max}$ – the allowed maximum current density ($A/m^2$) of the electrochemical process of zinc, which depends on the capacity of equipment and power supply.

$J_0$ – the basic tariff charge of electrochemical process of zinc. There are two types of $J_0$: (i) considering the capacity of transformer, $J_0 = k_c \times R$, where $k_c$ is the fixed parameter, $R$ is the total capacity of transformers; (ii) considering the maximum demand of electrical load, $J_0 = k_d \times PW_{\max}$, where $k_d$ is the fixed parameter, $PW_{\max}$ is the maximum electrical load in 4 periods.

The basic tariff charge in this paper is considered the capacity of transformer ($k_c = 2/3$, $R = 246000$). This problem only be studied by several researches and its mathematical formulations in these paper are all different, so we compare the performance of STA with built-in GA in MATLAB which is carried out during this study. It is worth pointing out that the constraint in this problem is an equality constraint, and when using STA solve this problem, the penalty factor in the second stage of constraint handling technique is chosen 10e6 based on many tests.

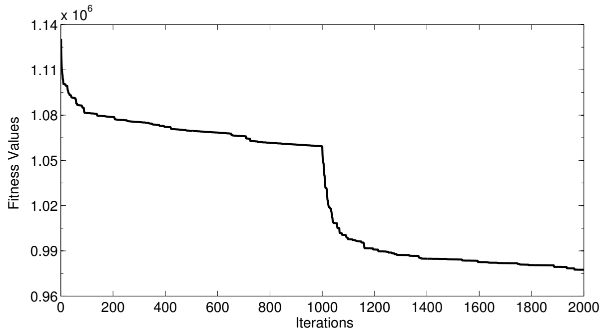


Fig. 8. The iterative curve of power-dispatching control system for the electrochemical process of zinc.

The best results obtained by STA is $f(X)$ = 9.5825e+05 corresponding to decision variable $X$ = [629, 200, 276, 646, 649, 200, 200, 649, 649 200, 200, 649, 647, 200, 200, 650, 624, 200, 267, 642, 324, 200, 200, 649, 537, 200, 200, 650] and constraint $h(X) = 960.0785$. The results of the experiments are shown in Table 7, Table 8 and Fig7.

As shown in Table 7, the STA can obtain better solution than GA which means STA can produce similar number of zinc with less money. And through analyzing the best solution of STA, we can find that in the period of high price, the current density is low and in the period of low price, the current density is high. This phenomenon is same as previously mentioned. Table 8 shows that STA has better results than GA in terms of best, mean and worst solutions. Fig. 8 also show that the STA has outstanding performance.

Based on the experimental results, the optimal solutions obtained by proposed method are all superior to other typical approaches. This is due to the fact that when solving constrained engineering optimization problems, it is easy to fall into the local optimal solution in the search process, and the state transition algorithm, as one of the global optimization method, not only has a special global search operator (expansion operator) to prevent premature convergence but also designs a local search operator (rotation operator) to improve the precision of the solution. At the same time, in the constraint-handling techniques, the two-stage strategy can obtain the candidate solutions belonging to feasible domain in the first stage and then use the feasible solution as the initial solution to find the optimal solution. Thus, the STA with two-stage strategy has better performance for constrained engineering optimization problems.

## 5. CONCLUSIONS

This paper presents the STA with two-stage strategy to solve various engineering optimization problems, which include welded beam design, pressure vessel design, tension/compression spring design and power-dispatching control system in the electrochemical process of zinc. In these problems, the objective functions and constraint functions are all nonlinear and nonconvex. The two-stage strategy of STA means that in the early stage of an iterative process, the feasible preference method is used to select a "best" solution, whilst it is changed to the penalty function method in the later stage. Application results have shown that STA can obtain better solution than other algorithms in literature in terms of both effectiveness and solution precision. Thus, STA can be considered as an alternative global search algorithm that can be applied to various engineering optimization problems.

However, since there are still some space to improve the precision of the final results obtained by STA, in the future, we will incorporate the gradient information into rotation operator to improve the local search ability of STA, and several deterministic optimization methods, such as `fmincon`, `GloptiPoly`, can also be used to accelerate the convergence speed of STA in the later iteration process. Besides, Zhao *et al*. [50] have proposed a new stepwise and piecewise approach to optimize the design of $CO_2$ transportation which is worth learning to improve the performance of STA. Thus, with the experience gained in this field, the STA for dealing with constrained optimization problem should be further studied in the future.

## REFERENCES

[1] S. Mohammadloo, M. H. Alizadeh, and M. Jafari, "Multivariable autopilot design for sounding rockets using intelligent eigenstructure assignment technique," *International Journal of Control Automation and Systems*, vol. 12, no. 1, pp. 208-219, 2014. [click]

[2] Z. Mohamed, M. Kitani, S. Kaneko, and G. Capi. "Humanoid robot arm performance optimization using multi objective evolutionary algorithm," *International Journal of Control, Automation and Systems*, vol. 12, no. 1, pp. 870-877, 2014. [click]

[3] Y. Minami, "Design of model following control systems with discrete-valued signal constraints," *International Journal of Control, Automation and Systems*, vol. 14, no. 1, pp. 331-339, 2016. [click]

[4] R. Madiouni, S. Bouallègue, J. Haggège, and P. Siarry. "Robust RST control design based on multi-objective particle swsarm optimization approach," *International Journal of Control, Automation and Systems*, vol. 14, no. 6, pp. 1607-1617, 2016. [click]

[5] Q. Zhang, Q. Wang, and G. Li. "Switched system identification based on the constrained multi-objective optimization problem with application to the servo turntable," *International Journal of Control, Automation and Systems*, vol. 14, no. 5, pp. 1153-1159, 2016. [click]

[6] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method for constrained optimization problems," *Proceedings of the Euro-International Symposium*

Table 7. Comparison of the best solution of power-dispatching control system for the electrochemical process of zinc.

| DV | GA | | | | STA | | | |
|---|---|---|---|---|---|---|---|---|
| $Dk_{11}, Dk_{21} \dots Dk_{41}$ | 452 | 237 | 342 | 634 | **629** | **200** | **276** | **646** |
| $Dk_{12}, Dk_{22} \dots Dk_{42}$ | 378 | 200 | 268 | 650 | **649** | **200** | **200** | **649** |
| $Dk_{13}, Dk_{23} \dots Dk_{43}$ | 598 | 248 | 205 | 650 | **649** | **200** | **200** | **649** |
| $Dk_{14}, Dk_{24} \dots Dk_{44}$ | 464 | 336 | 310 | 650 | **647** | **200** | **200** | **650** |
| $Dk_{15}, Dk_{25} \dots Dk_{45}$ | 349 | 329 | 307 | 650 | **624** | **200** | **267** | **642** |
| $Dk_{16}, Dk_{26} \dots Dk_{46}$ | 551 | 200 | 268 | 650 | **324** | **200** | **200** | **649** |
| $Dk_{17}, Dk_{27} \dots Dk_{47}$ | 275 | 202 | 268 | 650 | **537** | **200** | **200** | **650** |
| $h(Dk) = 960$ | 960.0179 | | | | **960.0785** | | | |
| $f(x)$ | 9.7961e+005 | | | | **9.5825e+005** | | | |

Table 8. Comparison of the statistical results of power-dispatching control system for the electrochemical process of zinc.

| Method | Best | Mean | Worst | SD |
|---|---|---|---|---|
| GA | 9.7961e+005 | 9.9669e+005 | 1.0279e+006 | 1.2516e+004 |
| **STA** | **9.5825e+005** | **9.7749e+005** | **9.8988e+005** | **9.8426e+003** |


*on Computational Intelligence 2002*, vol. 76, no. 1, pp. 214-220, 2002.

[7] J. Han, T. X. Dong, X. J. Zhou, C. H. Yang, and W. H. Gui, "State transition algorithm for constrained optimization problems," *Proceedings of the 33rd Chinese Control Conference*, pp. 7543-7548, 2014.

[8] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," *Computers & Structures*, vol. 89, no. 23-24, pp. 2325-2336, Dec. 2011. [click]

[9] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, pp. 113-127, 2000.

[10] N. B. Guedria, "Improved accelerated PSO algorithm for mechanical engineering optimization problems," *Applied Soft Computing*, vol. 40, pp. 455-467, 2016. [click]

[11] Q. Yuan and F. Qian, "A hybrid genetic algorithm for twice continuously differentiable NLP problems," *Comput. Chem. Eng.*, vol. 34, pp. 36-41, 2010. [click]

[12] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Eng. Appl. Artif. Intell.*, vol. 20, pp. 89-99, 2007. [click]

[13] C. A. C. Coello and R. L. Becerra, "Efficient evolutionary optimization through theuse of a cultural algorithm," *Eng. Optim.*, vol. 36, pp. 219-236, 2004. [click]

[14] T. Ray and K. M. Liew, "Society and civilization: an optimization algorithm based on the simulation of social behavior," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 386-396, 2003. [click]

[15] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm: a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Comput. Struct.*, vol. 110-111, pp. 151-166, 2012. [click]

[16] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput. J.*, vol. 13, pp. 2592-2612, 2013. [click]

[17] J. Han, C. H. Yang, X. J. Zhou, and W. H. Gui. "A new multi-threshold image segmentation approach using state transition algorithm," *Applied Mathematical Modelling*, vol. 4, pp. 588-601, 2017. [click]

[18] J. Han, C. H. Yang, X. J. Zhou, and W. H. Gui, "Dynamic multi-objective optimization arising in ironprecipitation of zinc hydrometallurgy," *Hydrometallurgy*, vol. 173, pp. 134-148, 2017.

[19] X. J. Zhou, P. Shi, C. C. Lim, C. H. Yang, and W. H. Gui, "A dynamic state transition algorithm with application to sensor network localization," *Neurocomputing*, vol. 273, pp. 237-250, 2018.

[20] X. J. Zhou, C. H. Yang, and W. H. Gui, "State transition algorithm," *Journal of Industrial and Management Optimization*, vol. 8, no. 3, pp. 1039-1056, 2012.

[21] X. J. Zhou, C. H. Yang, and W. H. Gui, "Nonlinear system identification and control using state transition algorithm," *Applied Mathematics & Computation*, vol. 226, no. 1, pp. 169-179, 2012.

[22] F. X. Zhang, X. J. Zhou, C. H. Yang, and W. H. Gui, "Fractional-order PID controller tuning using continuous state transition algorithm," *Neural Computing and Applications*, pp. 1-10, 2016.

[23] X. J. Zhou, C. H. Yang, and W. H. Gui, "A matlab toolbox for continuous state transition algorithm," *Proceedings of the 35th Chinese Control Conference*, pp. 9172-9177, 2016.

[24] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11, pp. 1245-1287, 2002.

[25] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284-294, 2000. [click]

[26] Z. X. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 658-675, 2006. [click]

[27] Y. Wang and Z. X. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 117-134, 2012. [click]

[28] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311-338, 2000. [click]

[29] A. R. Conn, N. I. M. Gould, and P. L. Toint, "A globally convergent augmented lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds," *Mathematics of Computation*, vol. 66, no. 217, pp. 261-288, 1997.

[30] Ï. Karen, N. Kaya, and F. Öztürk, "Intelligent die design optimization using enhanced differential evolution and response surface methodology," *Journal of Intelligent Manufacturing*, vol. 26, no. 5, pp. 1027-1038, 2015. [click]

[31] E. Zahara and Y. T. Kao, "Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems," *Expert Syst. Appl.*, vol. 36, pp. 3880-3886, 2009. [click]

[32] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Appl. Soft Comput. J.*, vol. 10, pp. 629-640, 2010. [click]

[33] J. Lampinen, "A constraint handling approach for the differential evolution algorithm," *IEEE Proc. 2002 Congr. Evol. Comput. CEC'02 (Cat. No.02TH8600)*, pp. 1468-1473, 2002.

[34] A. Husseinzadeh Kashan, "An efficient algorithm for constrained global optimization and application to mechanical engineering design: league championship algorithm (LCA)," *Comput. Des.*, vol. 43, pp. 1769-1792, 2011. [click]

[35] W. C. Yi, Y. Z. Zhou, L. Gao, X. Y. Gao, and C. J. Zhang, "Engineering design optimization using an improved local search based epsilon differential evolution algorithm," *Journal of Intelligent Manufacturing*, pp. 1-22, 2016.

[36] F. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Appl. Math. Comput.*, vol. 186, pp. 340-356, 2007. [click]

[37] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Appl. Math. Comput.*, vol. 186, pp. 1407-1422, 2007. [click]

[38] L. D. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Syst. Appl.*, vol. 37, pp. 1676-1683, 2010. [click]

[39] J. S. Arora, *Introduction to Optimum Design*, McGraw-Hill, New York, 1989.

[40] A. D. Belegundu, *A Study of Mathematical Programming Methods for Structural Optimization*, PhD thesis, Department of Civil and Environmental Engineering, University of Iowa, Iowa, USA, 1982.

[41] J. D. Huang, L. Gao, and X.Y. Li, "A teaching clearning-based cuckoo search for constrained engineering design problems," *Advances in Global Optimization*, Springer International Publishing, pp. 375-386, 2015.

[42] P. C. Ye, G. Pan, Q. G. Huang, and Y. Shi, "A new sequential approximate optimization approach using radial basis functions for engineering optimization," *Intelligent Robotics and Applications*, Springer International Publishing, pp. 83-93, 2015. [click]

[43] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, pp. 1567-1579, 2007. [click]

[44] T. Ray and P. Saini, "Engineering design optimization using a swarm with an intelligent information sharing among individuals," *Engineering Optimization*, vol. 33, pp. 735-748, 2001. [click]

[45] C. A. C. Coello and E. M. Montes, "Constraint- handling in genetic algorithms through the use of dominance-based tournament selection," *Advanced Engineering Informatics*, vol. 16, pp. 193-203, 2002. [click]

[46] E. M. Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *International Journal of General Systems*, vol. 37, pp. 443-473, 2008. [click]

[47] X. J. Zhou, C. H. Yang, and W. H. Gui, "Modeling and control of nonferrous metallurgical processes on the perspective of global optimization," *Control Theory & Applications*, vol. 9, no. 004, pp. 1158-1169, 2015.

[48] C. H. Yang, G. Deconinck, and W. H. Gui, "An optimal power-dispatching control system for the electrochemical process of zinc based on backpropagation and hopfield neural networks," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 5, pp. 953-961, 2003.

[49] C. H. Yang, G. Deconinck, W. H. Gui, and Y. G. Li, "An optimal power-dispatching system using neural networks for the electrochemical process of zinc depending on varying prices of electricity," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 229-236, 2002.

[50] D. Y. Zhao, Q. H. Tian, Z. M. Li, and Q. M. Zhu, " A new stepwise and piecewise optimization approach for CO2 pipeline," *International Journal of Greenhouse Gas Control*, vol. 49, pp. 192-200, 2016. [click]

**Jie Han** received her Bachelor's degree in Automation in 2014 from Central South University, Changsha, China and she is currently a Ph.D. student at Central South University, Changsha, China. Her main interests include optimization theory and algorithms, state transition algorithm, optimization and control of complex industrial process.

**Chunhua Yang** received her M.Eng. in Automatic Control Engineering and her Ph.D. in Control Science and Engineering from Central South University, China, in 1988 and 2002, respectively, and was with the Electrical Engineering Department, Katholieke Universiteit Leuven, Belgium from 1999 to 2001. She is currently a full professor in the School of Information Science & Engineering, Central South University. Her research interests include modeling and optimal control of complex industrial process, intelligent control system, and fault-tolerant computing of real-time systems.

**Xiaojun Zhou** received his Bachelor's degree in Automation in 2009 from Central South University, Changsha, China and received the PhD degree in Applied Mathematics in 2014 from Federation University Australia. He is currently an Associate Professor at Central South University, Changsha, China. His main interests include modeling, optimization and control of complex industrial process, optimization theory and algorithms, state transition algorithm, duality theory and their applications.

**Weihua Gui** received the degree of the B.Eng.(Automatic Control Engineering) and the M.Eng. (Control Science and Engineering) from Central South University, Changsha, China, in 1976 and 1981, respectively. From 1986 to 1988 he was a visiting scholar at Universitat-GH-Duisburg, Germany. He is a member of the Chinese Academy of Engineering and has been a full professor in the School of Information Science & Engineering, Central South University, Changsha, China, since 1991. His main research interests are in modeling and optimal control of complex industrial process, distributed robust control, and fault diagnoses.