Neurocomputing 455 (2021) 202–214

Contents lists available at ScienceDirect

Neurocomputing

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

A fast constrained state transition algorithm

Xiaojun Zhou [a], Jituo Tian [a], Jianpeng Long [a], Yaochu Jin [b], Guo Yu [b], Chunhua Yang [a,*]

[a] School of Automation, Central South University, Changsha 410083, China
[b] Department of Computer Science, University of Surrey, Guildford, Surrey GU2 7XH, UK

ARTICLE INFO

Article history:
Received 19 November 2020
Revised 31 March 2021
Accepted 17 May 2021
Available online 20 May 2021
Communicated by Zidong Wang

Keywords:
Constrained optimization
Fast state transition algorithm
Sequential quadratic programming
Constraint-handling technique

ABSTRACT

When solving constrained optimization problems in real industrial processes, both optimality and computational efficiency need to be considered. However, most existing meta-heuristic algorithms are slow to find the global optimum. The first reason is that the way to generate and select candidate solutions is time-consuming. The low probability to generate and select potential solutions in assisting the computational efficiency is another reason. In this paper, a simplified state transition algorithm (STA) and a novel constraint-handling technique are proposed to address the above issues for small size constrained optimization problems. Firstly, three out of four operators in basic STA to produce candidate solutions are selected and two operators are modified with adaptive parameter tuning, which have a large probability to generate potential solutions, but consumes less time. Secondly, the constraint-handling technique considers not only the objective function value and the constraint violation but also the difference among candidate solutions. Thirdly, the sequential quadratic programming embedded into the simplified STA can further speed up the convergence. Experiments are conducted on 22 well-known test functions from IEEE CEC2006 and 4 engineering constrained optimization problems, in comparison with state-of-the-art algorithms. The experimental results show that the proposed method is competitive in finding the optimum faster.

1. Introduction

In real-world applications, there are many constrained optimization problems (COPs), such as the process optimization [51,33,38] and system design [9,11,26,14]. Approaches for dealing with COPs can be classified into two categories: conventional deterministic optimization algorithms and meta-heuristic based intelligent algorithms. Usually, the conventional optimization algorithms are based on gradient information [6,32,3], such as the sequential quadratic programming (SQP) [16,34] and the method of Lagrange multiplier [1,22]. They have strong local search capability and fast convergency but weak global search capability [13]. To address these weaknesses, the meta-heuristic based intelligent algorithms, which have strong global search ability [2,15], have been developed for decades to deal with COPs.

Recently, increasing attention has been paid to developing meta-heuristic based intelligent algorithms for finding the global optimum of the COPs. Wang et al. [46] introduced an improved differential evolution (DE) to locate the optimum by introducing a new encoding mechanism. Karaboga et al. [25] adopted a modified artificial bee colony algorithm (ABC) and Deb's rules to find the optimum, while Brajevic et al. [8] used the cross-based ABC for solving COPs. Rao et al. [36,35] proposed a new approach, teaching learning-based optimization (TLBO), which can avoid dropping into the local optimum, for coping with COPs. Long et al. [29] proposed an improved grey wolf optimization for exploration and exploitation and a modified augmented Lagrangian multiplier method for handling constraints. Han et al. [17] designed a constrained state transition algorithm, where the state transition algorithm (STA) is the search engine and the constraint handling technique is the two-stage strategy, while the Zhou et al. [53] integrated the benefit of improved STA and a preference trade-off strategy for solving the power dispatch constrained optimization problem. Garg et al. [13] proposed a hybrid approach, PSO-GA, to deal with the COPs. This method incorporated the operators of GA in PSO to balance the global and local search abilities. However, the way in some approaches to generate and select solutions for the search of the global optimum is complex and time consuming. Therefore, efficiently and effectively locating the optima in dealing with the COPs remains an issue.

Due to the difficulty in the selection of the potential infeasible solutions, a variety of constraint-handling techniques have been designed and integrated into intelligent algorithms to cope with

* Corresponding author.
E-mail address: ychh@csu.edu.cn (C. Yang).

https://doi.org/10.1016/j.neucom.2021.05.053
0925-2312/

COPs. The constraint-handling techniques can be roughly divided into three categories. 1) Penalty based methods [23,24,28]: these studies use static, dynamic or self-adaptive penalty function methods to convert COPs into unconstrained optimization problems. However, it is challenging to appropriately tune the penalty factor to select the candidate solutions. 2) Feasibility preference methods [4,10,47]: these methods select the candidate solutions from one out of three solution sets, i.e., feasible solution set, feasible and infeasible solution set, or infeasible solution set, and they prefer feasible solutions over infeasible solutions. They are parameter-free but lack of exploration of the infeasible regions. 3) Hybrid methods: includes the stochastic ranking [37], two-stage strategy [17], adaptive trade-off model [45], ϵ-constraint method [40], etc. These methods may have better applicability in dealing with different constraints than the above techniques, probably because they leverage various strategies to select solutions in different optimization stages. Consequently, these methods are time-consuming. To sum up, the optimality of the solutions has the highest priority in the above techniques, and the diversity is sometimes considered, but the computational efficiency is easily neglected.

Inspired by the fast convergency of the conventional deterministic algorithms and the global search ability of the intelligent algorithms, this paper integrates the SQP into the STA for fast locating the global optimum. Besides, a new constraint handling technique is designed by taking the optimality, diversity and convergence rate into account to resolve the COPs. The main contributions of this paper are listed as follows.

- [1] A simplified STA embedded with SQP is proposed. In STA, three out of four operators are selected, which have a larger probability to generate candidate solutions to help finding the optimum. Besides, a self-adaptive mechanism is designed to tune the parameters in two operators for global and local search. Once a feasible solution with smallest function value has been found after a certain number of iterations, SQP is adopted to do exploitation around the feasible solution, which can further enhance the computational efficiency.
- [2] A novel constraint-handling technique, i.e., branch and screen strategy, is proposed, which takes consideration of the diversity, optimality, and computational efficiency of the solutions during the selection. The diversity can be seen from the separate selection of solutions among the feasible and infeasible solution sets by using a distance function, which can also reflect the optimality by choosing the best solutions among these two solution sets. This technique is relatively simpler than the hybrid techniques but can obtain more diverse solutions, so that the computational efficiency can be enhanced to much extent.

The rest of this paper is organized as follows. The related work is introduced in Section II. In Section III, the proposed algorithm is described in detail. The experimental studies and analysis are presented in Section IV. Finally, the conclusion of this paper is drawn in Section V.

2. Related work

In this section, some related definitions of COPs are firstly introduced. Then the basic state transition algorithm and the comparison algorithms in experiment are presented in detail.

2.1. Constrained optimization problems

Without loss of generality, the constrained optimization problems can be expressed as follows.

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & h_i(\boldsymbol{x}) = 0, \quad i = 1,2,\cdots,p \\ & g_j(\boldsymbol{x}) \leqslant 0, \quad j = 1,2,\cdots,q \\ & l_k \leqslant x_k \leqslant u_k, \quad k = 1,2,3,\ldots,n \end{aligned} \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function; $h_i(\boldsymbol{x})$ and $g_j(\boldsymbol{x})$ are the equality and inequality constraints respectively and their numbers are p and q respectively; the number of decision variables is n; the $\boldsymbol{x}$ can be described as: $\boldsymbol{x} = (x_1, x_2, x_3, \ldots, x_n)$; l_k and u_k are the lower and upper bounds of the kth decision variable.

The constraint violation is a common concept in COPs. It describes a degree that the constraints are violated in candidate solutions.

$$\begin{cases} G_j(\boldsymbol{x}) = \max(0, g_j(\boldsymbol{x}))^k, j = 1,2,\cdots,q \\ H_i(\boldsymbol{x}) = \max(0, |h_i(\boldsymbol{x})| - \xi)^k, i = 1,2,\cdots,p \end{cases}, \tag{2}$$

where, $G_j(\boldsymbol{x})$ and $H_i(\boldsymbol{x})$ are the violation of $g_j(\boldsymbol{x})$ and $h_i(\boldsymbol{x})$ respectively; ξ is a threshold value for equality constraints and $k$is normally 1 or 2.

2.2. Basic state transition algorithm

The basic STA [54] is a meta-heuristic algorithm, which was firstly proposed in 2012 for unconstrained optimization problems. The main idea of the STA is regarding a solution as a state, and the process of updating solution is considered as the state transition [56]. The search ability of STA has been demonstrated in many real engineering fields, such as nonlinear system identification [55,49], industrial process control and optimization [52,50,58,53,18], machine learning [42,57,21] and other fields [20,19]. In STA, there are four operators, i.e., expansion, rotation, axesion and translation transformation, to generate candidate solutions. The unified framework of basic STA is presented as follows.

$$\boldsymbol{s_{k+1}} = \boldsymbol{A_k s_k} + \boldsymbol{B_k u_k} \tag{3}$$

where, the $\boldsymbol{s_k}$ and $\boldsymbol{s_{k+1}}$ are the best current and next states, respectively; $\boldsymbol{u_k}$ is the function of historical states; A_k and B_k denote the matrices for state transition.

The expansion, rotation, axesion and translation transformation operators are detailed as follows.

(1) Expansion transformation:

$$\boldsymbol{s_{k+1}} = \boldsymbol{s_k} + \gamma \boldsymbol{R_e s_k}, \tag{4}$$

where γ is a positive constant, called the expansion factor; $R_e \in \mathbb{R}^{n\times n}$ is a random diagonal matrix with its entries obeying the Gaussian distribution.

(2) Rotation transformation:

$$\boldsymbol{s_{k+1}} = \boldsymbol{s_k} + \alpha \boldsymbol{R_r} \frac{\boldsymbol{1}}{\boldsymbol{n}\|\boldsymbol{s_k}\|_2} \boldsymbol{s_k}, \tag{5}$$

where α is a positive parameter, called the rotation factor; $R_r \in \mathbb{R}^{n\times n}$is a random matrix whose elements are distributed uniformly in the range of [-1,1]; the $\|\cdot\|_2$ is a L2-norm of a vector; the n denotes the dimension of the $\boldsymbol{x_k}$. The rotation transformation is designed for both local and global search in a hypersphere by adjusting the rotation factor α.

(3) Axesion transformation:

$$\boldsymbol{s_{k+1}} = \boldsymbol{s_k} + \delta \boldsymbol{R_a s_k}, \tag{6}$$

where $\delta$is a positive parameter, called the axesion factor; $R_a \in \mathbb{R}$is a random variable in [0,1]. The axesion transformation is also designed for both exploration and exploitation but search along the axes by adjusting the translation factor to change the range of the search.

(4) Translation transformation:

$$\boldsymbol{s_{k+1}} = \boldsymbol{s_k} + \beta \boldsymbol{R_t} \frac{\boldsymbol{s_k} - \boldsymbol{s_{k-1}}}{\|\boldsymbol{s_k} - \boldsymbol{s_{k-1}}\|_2}, \tag{7}$$

where β is the translation factor, which is a positive constant; the $R_t \in \mathbb{R}$ is a random variable in [0,1]. The translation transformation is applied for local search, which searches along the line from $\boldsymbol{s_{k-1}}$ to $\boldsymbol{s_k}$ with a fixed search range.

2.3. Related algorithms

Several comparison algorithms used in the experiment are introduced in this part, which are highly studied.

2.3.1. Teaching-learning-based optimization (TLBO)

The teaching–learning-based optimization was proposed by Rao [36] for mechanical design optimization problems in 2011. The main idea of the TLBO is to simulate the influence of a teacher on learners. The process of TLBO can be divided into two phases. The first phase consists of the "Teacher Phase", which means learning from the teacher. The second phase consists of the "Learner Phase", which means learning in the interaction among learners. One of the merit of TLBO is parameter-free, while another merit is robust to deal with the continuous non-linear COPs.

2.3.2. Constrained tree-seed algorithm with Deb's rules (CTSA)

The modified constrained tree-seed algorithm with Deb's rules is one of the population-based algorithm which was designed by Babalik [4] in 2018. It was inspired by the relations between the trees and seeds grown on the land. The Deb's rules was used for selecting the trees and seeds, which would survive in the next iteration. In the CTSA, there are three specific operators to detect and search different areas and enhance the convergence speed, respectively. Especially, the local search capability is controlled by the number of seeds during the optimization process.

2.3.3. Differential evolution algorithm with encoding mechanism (DEEM)

The differential evolution (DE) algorithm with encoding mechanism was proposed by Wang [46] for obtaining the optimal layout of the wind farm to maximize the power output. The DE algorithm as a search engine provides strong global search ability for solving the COPs in the DEEM. Moreover, a caching strategy is adopted to speed up the evaluation process. With the help of the proposed cashing strategy, DEEM is able to find the optima of the COPs by tuning few parameters.

3. The proposed fast constrained state transition algorithm

3.1. Motivation

In dealing with COPs, it is common that the existing methods are dedicated to locating the optimum by introducing complex strategies in generating and selecting candidate solutions, which may sacrifice a lot of execution time. Consequently, this paper takes the optimality and computational efficiency into account from the following aspects. The first is the fast solution generation. An operator is introduced to produce a number of solutions at a time rather than using an operator multiple times to generate the same number of solutions, so that much computation time can be saved by less frequently using the solution generator. The second is the solution selection in the constraint-handling technique. The selection takes account of the optimality, the differentiation and the computational efficiency, where the diverse solutions may assist in the search of the optimum and further improve the computational efficiency. The third is the convergence enhancement. The conventional SQP having fast convergency is introduced into the STA to quickly locate the optimum. Therefore, this research aims to enhance the computational efficiency without compromising the optimality to deal with the COPs.

3.2. Fast constrained state transition algorithm

This paper proposes a fast constrained state transition algorithm (FCSTA) to deal with COPs. The FCSTA includes three key components. The first is the simplified STA for fast solution generation, presented in subSection 3.2.1. The second part is SQP for convergence enhancement, as exhibited in subSection 3.2.2. The last part is the the proposed constraint-handling technique for solution selection, as shown in subSection 3.2.3.

3.2.1. Modified state transition algorithm

In this study, the modified STA is adopted to deal with COPs. But this research only investigates the rotation, axesion and translation transformation operators. The expansion and translation transformation operators are designed for global search and local search, respectively, and the rest two transformation operators aim to improve the diversity of the solutions generated by the expansion and translation transformation operators in different aspects. For example, the rotation operator samples solutions in a controllable hyper-sphere where the center is the solution with the incumbent best objective function value. The axesion operator generates solutions by modifying the values of decision variables of the selected solutions in different dimensions. Due to the controllable search ability of the rotation, axesion and translation transformation operators, these operators may be effective in the detection of the feasible regions, and further assist in the search of the optimum of the COPs. In contrast, the expansion operator generates solution in the whole search space while these solutions may be more likely to be sampled in the infeasible regions. Consequently, the expansion operator consumes many evaluations but fails to produce feasible solutions to solve COPs.

After selecting three out of four operators in basic STA, two of them need to be modified with adaptive parameter tuning, so in Eq. (5), and Eq. (6), there are two parameters (α and δ) to control the search scale. In this work, a simple self-adaptive parameter tuning strategy is designed to balance the global and local search in different stages of the optimization. In the initial stage, the tuning process is conducted as follows.

$$\alpha = \begin{cases} \alpha_{\max 2}, & \textit{if} \;\; \alpha \leqslant \alpha_{\max 1} \\ \alpha / fc, & \text{otherwise} \end{cases}, \tag{8}$$

$$\delta = \begin{cases} \delta_{\max 2}, & \textit{if} \;\; \delta \leqslant \delta_{\max 1} \\ \delta / fc, & \text{otherwise} \end{cases}, \tag{9}$$

where the fc is a constant, called lessening coefficient, which is applied to shrink the search range of rotation and axesion operators in terms of exponential function; the α is set in the range of $[\alpha_{\max 1}, \alpha_{\max 2}]$, which helps the rotation operator explore the search space in a controllable hyper-sphere. By contrast, δ is set in $[\delta_{\max 1}, \delta_{\max 2}]$ but axesion operator assists the search along the axis. Both of them are meant to globally detect the feasible regions.

After the initial stage, the tuning process is performed as follows.

$$\alpha = \begin{cases} \alpha_{\max 1}, & \textit{if} \;\; \alpha \leqslant \alpha_{\min} \\ \alpha / fc, & \text{otherwise} \end{cases}, \tag{10}$$

$$\delta = \begin{cases} \delta_{\max 1}, & \textit{if} \;\; \delta \leqslant \delta_{\min} \\ \delta / fc, & \text{otherwise} \end{cases}, \tag{11}$$

where the α is in $[\alpha_{\min}, \alpha_{\max 1}]$, and δ in $[\delta_{\min}, \delta_{\max 1}]$. In this stage, both rotation and axesion transformation operators are applied to do exploitation in the feasible regions.

3.2.2. SQP based local search

In comparison with the intelligent algorithms, the deterministic algorithms, like the SQP, have a stronger local search ability with a fast speed. Therefore, this study integrates the SQP into the search process when some feasible solutions are found after a number of iterations. The readers are referred to [7] for more details of SQP. With the assistance of the SQP, the search process will be sped up to find the region around the optimum, in which a feasible solution is specified to the SQP as the initial solution.

3.2.3. Branch and screen strategy

In constrained evolutionary optimization, the constraint handling techniques are applied for selecting potential solutions from the candidate solution set. Commonly, the optimality and complexity are taken into consideration in some constraint handling strategies, such as the penalty method [12] and feasibility preference method [5]. In the hybrid techniques, both the optimality and diversity are considered to select potential solutions to generate effective solutions in exploring the feasible regions, such as the multi-objective concepts [43,44] and the ensemble of constraint-handling techniques [30,41]. However, the hybrid techniques use different strategies to select solutions in different optimization stages so that they may cost much computing resources. Hence, the optimality, computational efficiency and diversity are considered in this work and a novel constraint handling technique is proposed, called branch and screen strategy. The branch and screen strategy is detailed in Fig. 1.

In Fig. 1, candidate solutions are divided into two categories (feasible and infeasible solution sets). Then the solutions in different categories will be screened or filtered in different ways according to the objective values and constraint violations.

In the selection of solutions among the feasible solution set, the objective function values of the solutions are sorted in an ascending order. The right branch of Fig. 1 shows that the feasible solution with the smallest objective value is selected with the highest priority. Then, other candidate solutions are screened according to Eq. (12).

$$S = \|\boldsymbol{x_k} - \boldsymbol{x_{best}}\|_2 - \xi, \tag{12}$$

where

$$\xi = \frac{c \cdot (U - L)}{\left(\frac{FEs}{Thre\_1} + 1\right)},$$

where c and $Thre\_1$ are two constants; U and L are the upper and lower bounds of the decision vector; FEs denotes the current number of function evaluations in an optimization process; $\boldsymbol{x_k}$ is the current candidate solution; $\boldsymbol{x_{best}}$ is a solution with best objective function value. If S is lager than zero, then $\boldsymbol{x_k}$ is selected. Eq. (12) describes the diversity of the solutions by computing the distance between the solution $\boldsymbol{x_k}$ to the $\boldsymbol{x_{best}}$, where ξ is used to describe the degree of the variance. Therefore, the screen strategy to select solutions among the feasible solution set will maintain diversity by selecting the solution with best objective value, avoiding repetitive search of a specific region by selecting diverse solutions in terms of Eq. (12).

An illustrative example of ξ over *FEs* is given in Fig. 2, where the upper and lower bounds are obtained according to a constrained engineering optimization problem (power dispatch) [17]. The plot shows that ξ descends with the increase of the number of evaluations. In the initial stage of the optimization, a large value is set to ξ to select the solutions assisting the global search but a small value assisting the local search in the later stage of optimization.

The screen strategy to select solutions among the infeasible solution set is presented in the left branch of Fig. 1. Firstly, the objective function value of the infeasible solutions is compared with that of the best solution ($\boldsymbol{x_{best}}$) in the feasible solution set. Then, the infeasible solutions with smaller objective values will be retained for further screening. After the selection, the infeasible solution with smallest constraint violation is firstly selected. Then other infeasible solutions with smaller constraint violation are selected into the potential solution set when their corresponding S values are positive by computing Eq. (12).

The branch and screen strategy proposed in this study simultaneously considers the optimality and diversity of the feasible and infeasible solutions, so the solution of the COPs may quickly converge in different directions. On the one hand, the comparison in terms of the objective values may drive the search process toward the optimum of COPs, especially among the feasible solution set. On the other hand, the diversity of feasible solutions estimated by Eq. (12) can help the local search (like SQP and translation operator) to find the local optimum, while the diversity of infeasible solutions may guide the search towards the feasible regions where their optimal solutions have smaller objective values than the best one ($\boldsymbol{x_{best}}$).

The main procedure of branch and screen strategy is given in Algorithm 1.

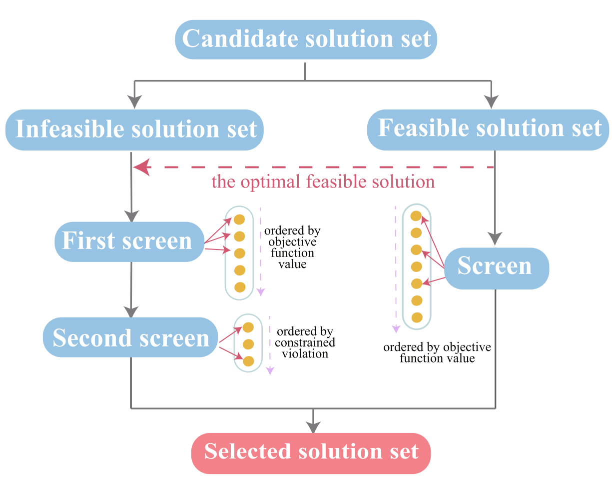

Fig. 1. An illustration of branch and screen strategy.

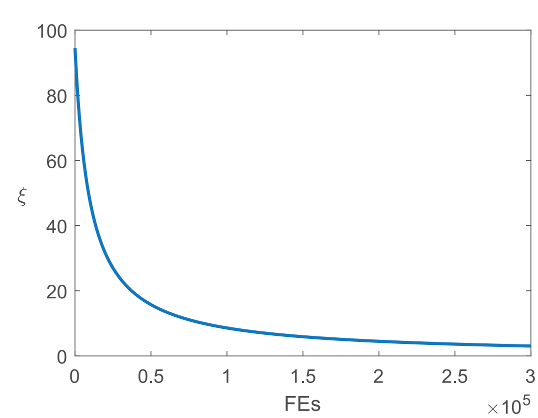

Fig. 2. Example for ξ over the number of evaluations *FEs* on the problem of power dispatch.

Algorithm 1: Pseudo-code of the branch and screen strategy

Input:
X: a solution set generated by the modified STA
F: the corresponding objective function values
G: the corresponding constrained violation values
Output:
X^*: the selected potential solutions
F^*: the objective values of selected potential solutions
G^*: the constraint violations of selected potential solutions
1. $[X\_feasi, F\_feasi, num\_feasi, X\_infeasi, F\_infeasi, G\_infeasi, num\_infeasi] \leftarrow$ **Branch** (X, F, G)
2. **if** $num\_feasi > 0$ **then**:
3. $[X^*\_feasi, F^*\_feasi] \leftarrow$ **Scr_Feasi**(X_feasi, F_feasi)
4. $G^*\_feasi \leftarrow 0$; $F\_feasiBest \leftarrow F^*\_feasi(1)$
5. **else**
6. $X^*\_feasi \leftarrow []$; $F^*\_feasi \leftarrow []$
7. $G^*\_feasi \leftarrow []$; $F\_feasiBest \leftarrow []$
8. **end if**
9. **if** $num\_infeasi > 0$ **then**:
10. $[X^*\_infeasi, F^*\_infeasi, G^*\_infeasi] \leftarrow$ **Scr_Infeasi** $(X\_infeasi, F\_infeasi, G\_infeasi, F\_feasiBest)$
11. **else**
12. $X^*\_infeasi \leftarrow []$; $F^*\_infeasi \leftarrow []$
13. $G^*\_infeasi \leftarrow []$
14. **end if**
15. $X^* \leftarrow [X^*\_feasi; X^*\_infeasi]$
16. $F^* \leftarrow [F^*\_feasi; F^*\_infeasi]$
17. $G^* \leftarrow [G^*\_feasi; G^*\_infeasi]$

In Algorithm 1, the **Branch** function in line 1 is designed to divide candidate solutions into feasible and infeasible solution sets. The output of the **Branch** function includes feasible solutions (*X_feasi*) and infeasible solutions (*X_infeasi*), objective function values of feasible solutions (*F_feasi*) and infeasible solutions (*F_infeasi*), constraint violation of infeasible solutions (*G_infeasi*), and the number of feasible solutions (*num_feasi*) and infeasible solutions (*num_infeasi*). Lines 2 to 9 are applied to deal with feasible solution set. The **Scr_Feasi** function is applied to screen the diverse feasible solutions with smaller objective function value. The *F_feasiBest* is the objective function value of the smallest feasible solution. Lines 10 to 16 are used to cope with the infeasible solution set. The **Scr_Infeasi** function is used to screen diverse infeasible solutions with smaller constraint violation and objective function value.

3.3. Framework of the proposed algorithm

The proposed fast constrained state transition algorithm consists of the modified STA, SQP and the branch and screen strategy. The main procedure of FCSTA is given in Algorithm 2.

Algorithm 2: Pseudo-code of the fast constrained state transition algorithm

Input:
max_FEs: the maximum number of evaluations
$[\alpha_{max1}, \alpha_{max2}]$: the range of rotation factor in initial stage
$[\alpha_{min}, \alpha_{max1}]$: the range of rotation factor in second stage
$[\delta_{max1}, \delta_{max2}]$: the range of axesion factor in initial stage
$[\delta_{min}, \delta_{max1}]$: the range of axesion factor in second stage
β: the factor of translation transformation operator
$r, Thre\_2$: two constants
SE: the number of samples
Output:
$Best^*$: the optimal solution

a (*continued*)

Algorithm 2: Pseudo-code of the fast constrained state transition algorithm

1. $FEs \leftarrow 0$; $Iter \leftarrow 0$
2. $\alpha \leftarrow \alpha_{max2}$; $\delta \leftarrow \delta_{max2}$
3. **while** $FEs < max\_FEs$ **do**
4. if $FEs \leqslant max\_FEs * r$
5. Do parameter tuning as Eqs. 8,9
6. **else**
7. Do paramter tuning as Eqs. 10,11
8. **end if**
9. $[X_r, F_r, G_r] \leftarrow$ **rotation** $(func, X, \alpha, SE, range)$
10. $[X, F, G] \leftarrow$ **branch_screen** (X_r, F_r, G_r)
11. $[X_a, F_a, G_a] \leftarrow$ **axesion** $(func, X, \delta, SE, range)$
12. $[X, F, G] \leftarrow$ **branch_screen** (X_a, F_a, G_a)
13. $[X_t, F_t, G_t] \leftarrow$ **translation** $(func, X, \beta, SE, range)$
14. $[X, F, G] \leftarrow$ **branch_screen** (X_t, F_t, G_t)
15. **if** $Iter$ mod $Thre\_2 == 0$ **and** $G(1) == 0$ **then**
16. $[X, F] \leftarrow$ **SQP** $(func, X(1), F(1), range)$
17. **end if**
18. $Iter \leftarrow Iter + 1$
19. **end while**
20. $Best^* \leftarrow X$

In Algorithm 2, lines 1 to 2 initialize several parameters, such as the current number of function evaluations (*FEs*, which is treated as a global variable and updated in each transformation operator) and iterations (*Iter*). Lines 4 to 8 implement simple self-adaptive parameter tuning strategy by Eqs. (8)–(11). In lines 9 to 17, the highlighted functions in bold are used to generate and select candidate solutions. The output of **rotation** function denotes the generated candidate solutions, the corresponding objective values and constraint violations, and the same as the **axesion** and **translation** functions. The **branch_screen** is used to select potential solutions (*X*) from the generated candidate solution set. The *X(1)* is the best feasible solution in *X*. The *F(1)* is the corresponding function value of *X(1)*. SQP is conducted in lines 15–17.

The flow chart of FCSTA is presented as the Fig. 3.

In initialization, the parameters of the modified STA and the branch and screen strategy are initialized. After the self-adjust parameter tuning, the rotation, axesion and translation transformation operators are introduced to generate candidate solution alternatively. After each transformation, the branch and screen strategy is performed. Once the condition is satisfied that a feasible solution has found under a certain number of iterations, SQP is conducted. The above steps will be repeated until the terminal condition is met.

4. Experimental studies and analysis

4.1. Benchmark test and parameter setting

In this work, 22 well-known benchmark test functions [28] are adopted to test the proposed method in comparison with three state-of-the-art algorithms to deal with the COPs, i.e, TLBO [36], CTSA [4], and DEEM [46]. The 22 well-known benchmark test functions are widely used for the constrained real-parameter optimization [27]. Various characteristics are synthetically taken into account on 22 test functions such as the combination of (i.e. linear/nonlinear) objective functions and (i.e. equality/inequality) constraints, and different dimensions of decision space. After that, four engineering optimization problems are applied to further investigate the performance of FCSTA, i.e, welded beam design [39], tension/compression spring design [11], pressure vessel design [31] and power dispatch [17]. The four problems are detailed in Eqs. (13)–(16).

The parameter setting of FCSTA is presented in Table 1 for the experiments on 22 test functions and four real applications. All the programs were executed in Matlab 2018b using the CPU with an Intel(R) Core(TM) i7-8550U @ 1.8 GHz and 8.00 GB RAM. The operation system was the Microsoft Windows 10.

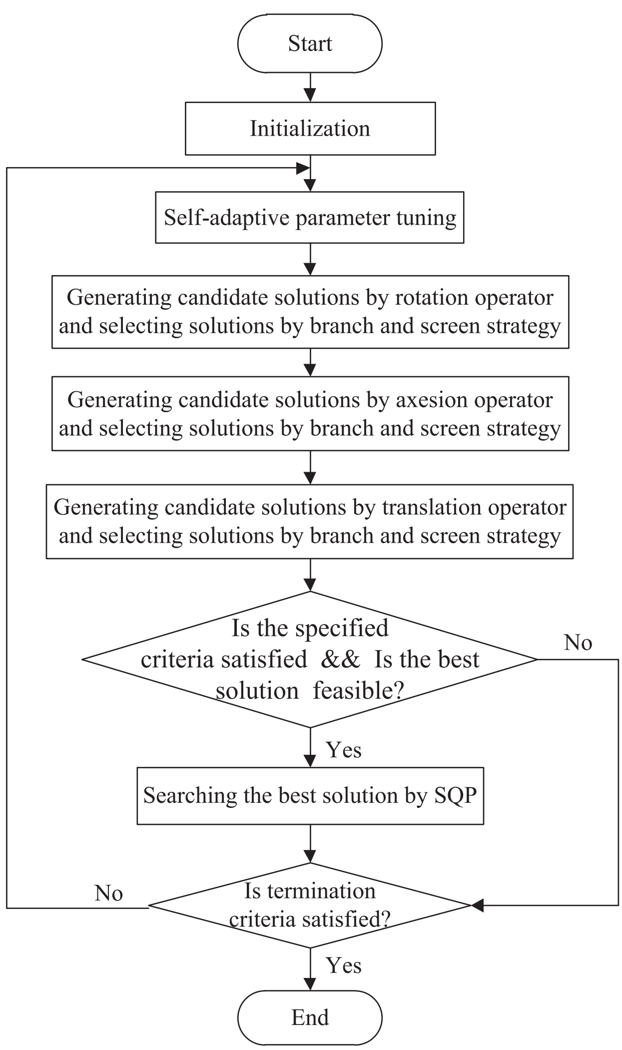

Fig. 3. Flow chart of FCSTA.

4.2. Experiments on 22 well-known test functions

The experiments on 22 test functions are presented in Table 2, where characteristics are best objective function values (*Best*), standard deviation (*Std.dev*) and average time (*T_ave(s)*).

The results are based on 50 independent runs where 300000 function evaluations are set for all methods per each independent run. In the table, *Best* reflects the optimality of four different methods. *Std.dev* indicates the robustness of algorithms. In this study, the *T_ave(s)* describes how fast the algorithm can find the optimum of the COPs. Notably, the equality constraints in the above test functions are converted into inequality constraint by introducing the variable reduction strategy in [48].

In Table 2, the experimental results show that the FCSTA outperforms the compared algorithms with respect to the computational efficiency, optimality, and robustness on most instances of 22 test functions. Specifically, the *T_ave(s)* value of FCSTA on all instances is much less than that obtained by other three algorithms. Especially, the *T_ave(s)* of FCSTA on G21 and G23 are 0.66s and 0.74s, respectively, and FCSTA achieves around 10 times faster than other methods to find the optimum on these two problems. The computational efficiency of FCSTA benefits from the way of the solution generations, because the proposed algorithm saves much time to sample solutions by three transformation operators. According to the (*Best*) value, FCSTA performs the same in comparison with DEEM on most instances and slightly worse on G05 G14, G21, and G23, but outperforms CTSA and TLBO on 9 and 10 out of 22, respectively and performs the same on the rest instances. Additionally, the CTSA and TLBO fail to find the feasible solution of G21. By contrast, TLBO fails to find the feasible solution of G17, G21 and G23 in some dependent runs, therefore, their corresponding standard deviations on these problems cannot be computed and are denoted by ('-'). In terms of the standard deviation, FCSTA shows better performance than CTSA and TLBO on 10 and 13 out of 22 instances, respectively, but slightly worse than DEEM on 5 instances. On the rest instances, FCSTA performs the same as the three methods. In summary, the proposed FCSTA has a good balance between the optimality, computational efficiency and robustness.

Table 1
Parameter settings of FCSTA

Parameter	Value	Parameter	Value	Parameter	Value
max_FEs	300000	c	0.01	r	0.03
Thre_1	1e4	δ_{max2}	30	α_{max2}	20
Thre_2	10	δ_{max1}	3	α_{max1}	2
SE	40	δ_{min}	1e-4	α_{min}	1e-4
fc	2	β	1		

Table 2
Experimental results of FCSTA and other three methods over 50 independent runs on 22 test functions.

Func	FCSTA			TLBO [36]			CTSA [4]			DEEM [46]		
	Best	*Std.dev*	*T_ave(s)*	*Best*	*Std.dev*	*T_ave(s)*	*Best*	*Std.dev*	*T_ave(s)*	*Best*	*Std.dev*	*T_ave(s)*
G01	-15	0	**0.93**	-15	0	25.62	-15.00	0.00	13.66	-15	0	7.71
G02	-0.80	0.01	**1.17**	-0.76	0.08	32.66	-0.80	0.00	11.15	-0.80	0.00	8.72
G03	-1.00	0.00	**0.40**	-1.00	0.07	22.55	-1.00	0.00	9.18	-1.00	0.00	6.75
G04	-30665.54	0.00	**0.62**	-30665.54	0.00	21.41	-30665.54	0.00	10.86	-30665.54	0.00	8.97
G05	5126.52	0.09	**0.83**	5582.01	0.00	22.33	5126.89	56.50	12.12	5126.50	0.00	6.65
G06	-6961.81	0.00	**0.81**	-6961.81	0.00	20.46	-6961.81	0.00	10.41	-6961.81	0.00	7.32
G07	24.31	0.00	**0.76**	24.32	0.06	24.74	24.40	0.07	11.52	24.31	0.00	5.65
G08	-0.10	0.00	**0.64**	-0.10	0.00	21.23	-0.10	0.00	10.64	-0.10	0.00	6.23
G09	680.63	0.00	**0.81**	680.63	0.00	28.97	680.63	0.00	11.24	680.63	0.00	8.77
G10	7049.25	11.95	**0.77**	7077.02	96.89	27.56	7057.36	45.27	14.39	7049.25	0.00	6.98
G11	0.75	0.00	**0.72**	0.75	0.00	18.83	0.75	0.00	10.36	0.75	0.00	4.75
G12	-1	0	**2.28**	-1	0	99.44	-1	0	84.50	-1	0	7.73
G13	0.05	0.00	**0.97**	0.44	0.20	22.65	0.50	0.14	11.46	0.05	0.00	6.93
G14	-47.72	0.01	**0.51**	-45.40	1.12	26.43	-45.07	1.11	12.98	-47.76	0.00	8.66
G15	961.72	0.00	**0.74**	961.71	1.35	19.51	962.06	2.31	11.14	961.72	0.00	6.78
G16	-1.90	0.00	**0.85**	-1.90	0.00	27.09	1.90	0.00	15.85	-1.91	0.00	7.72
G17	8853.53	0.00	**0.79**	9025.71	-	24.57	8865.40	0.00	13.13	8853.53	0.00	7.11
G18	-0.87	0.09	**0.79**	-0.86	0.04	29.49	-0.86	0.08	12.47	-0.87	0.00	7.70
G19	32.66	0.00	**0.98**	32.74	0.45	30.85	37.07	0.93	11.13	32.66	0.00	11.05
G21	193.79	0.01	**0.66**	-	-	26.25	-	-	13.47	193.72	0.23	8.16
G23	-400.00	0.00	**0.73**	0	-	24.84	-400.00	0.03	11.06	-400.06	0.00	9.21
G24	-5.51	0.00	**0.70**	-5.51	0.00	20.95	-5.51	0.00	10.65	-5.51	0.00	6.33

4.3. Engineering optimization problems

The engineering optimization problems are from the real-world applications. They can reflect the real industrial production processes. Four engineering optimization problems are introduced to test the performance of the proposed method, including the welded beam design [39], tension/compression spring design [11], pressure vessel design [31] and power dispatch [17].

A. Welded beam design

$$\begin{aligned} \min \quad & f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14+x_2), \\ \text{s.t.} \quad & g_1(X) = \tau(X) - \tau_{max} \leqslant 0, \\ & g_2(X) = \sigma(X) - \sigma_{max} \leqslant 0, \\ & g_3(X) = x_1 - x_4 \leqslant 0, \\ & g_4(X) = 0.125 - x_1 \leqslant 0, \\ & g_5(X) = \delta(X) - 0.25 \leqslant 0, \\ & g_6(X) = P - P_c(X) \leqslant 0, \\ & g_7(X) = 0.10471x_1^2 + 0.04811x_3x_4(14+x_2) - 5 \leqslant 0. \end{aligned} \tag{13}$$

where

$$\begin{aligned} & 0.1 \leqslant x_1 \leqslant 2, 0.1 \leqslant x_2 \leqslant 10, \\ & 0.1 \leqslant x_3 \leqslant 10, 0.1 \leqslant x_4 \leqslant 2, \\ & \tau(X) = \sqrt{\tau_1^2 + 2\tau_1\tau_2(\tfrac{x_2}{2R}) + \tau_2^2}, \\ & \tau_1 = \tfrac{P}{\sqrt{2}x_1x_2}, \tau_2 = \tfrac{MR}{J}, \\ & M = P(L + \tfrac{x_2}{2}), J(X) = 2\{\sqrt{2}x_1x_2[\tfrac{x_2^2}{4} + (\tfrac{x_1+x_3}{2})^2]\}, \\ & R = \sqrt{\tfrac{x_2^2}{4} + (\tfrac{x_1+x_3}{2})^2}, \sigma(X) = \tfrac{6PL}{x_4x_3^2}, \delta(X) = \tfrac{6PL^3}{Ex_3^3x_4}, \\ & P_c(X) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}(1 - \tfrac{x_3}{2L}\sqrt{\tfrac{E}{4G}}), \\ & G = 12 \times 10^6 ps, E = 30 \times 10^6 ps, \\ & P = 6000lb, L = 14in. \end{aligned}$$

B. Tension/compression spring design

$$\begin{aligned} \min \quad & f(X) = (x_3+2)x_2x_1^2, \\ \text{s.t.} \quad & g_1(X) = 1 - \tfrac{x_2^3x_3}{71785x_1^4} \leqslant 0, \\ & g_2(X) = \tfrac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \tfrac{1}{5108x_1^2} - 1 \leqslant 0, \\ & g_3(X) = 1 - \tfrac{140.45x_1}{x_2^2x_3} \leqslant 0, \\ & g_4(X) = \tfrac{x_1+x_2}{1.5} - 1 \leqslant 0. \end{aligned} \tag{14}$$

where

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

C. Pressure vessel design

$$\begin{aligned} \min \quad & f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + +3.1661x_1^2x_4 + 19.84x_1^2x_3 \\ \text{s.t.} \quad & g_1(X) = -x_1 + 0.0193x_3 \leqslant 0, \\ & g_2(X) = -x_2 + 0.00954x_3 \leqslant 0, \\ & g_3(X) = -\pi x_3^2x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \leqslant 0, \\ & g_4(X) = x_4 - 240 \leqslant 0. \end{aligned} \tag{15}$$

where

$$0.0625 \leqslant x_1, x_2 \leq 6.1875,\ 10 \leqslant x_3, x_4 \leqslant 200.$$

D. Power dispatch

$$\begin{aligned} \min \quad & F_C(X) = \sum_{i=1}^{N_t} PW_i \times T_i \times P_i + F_{C0}, \\ \text{s.t.} \quad & h(X) = \sum_{i=1}^{N_t}\sum_{j=1}^{N_e} q \times I_{ij} \times Nc_j \times E_{ij} \times T_i = G. \end{aligned} \tag{16}$$

where

$$\begin{aligned} & PW_i = \sum_{j=1}^{N_e} V_{ij} \times I_{ij} \times Nc_j, \\ & I_{ij} = Np_j \times S \times X_{ij}, \\ & V_{ij} = a_0 + a_1 \times X_{ij}, \\ & E_{ij} = b_0 + b_1 \times X_{ij} + b_2 \times X_{ij}^2 + b_3 \times X_{ij}^3 + b_4 \times X_{ij}^4, \\ & Nc = [240, 240, 246, 192, 208, 208, 208], \\ & Np = [34, 46, 54, 56, 56, 57, 57], \\ & j = [1, 2, 3, 4, 5, 6, 7], i = [1, 2, 3] \end{aligned}$$

$$P = [1.6, 1, 0.7] \times 0.5627, b_0 = 0.785037,$$
$$b_1 = 5.855e - 4, b_2 = 2e - 6, b_3 = 3.2094e - 9,$$
$$b_4 - 1.9052e - 12, a_0 = 2.76284, a_1 = 0.00093,$$
$$F_{C0} = 164000, G = 960, S = 1.13,$$
$$200 \leqslant X_{ij} \leqslant 650.$$

In this section, two experiments are designed to fully investigate the performance of the proposed method to efficiently locate the optimum of these real applications in comparison with the three methods. Firstly, the termination condition is set to a fixed number of evaluations which is equal to 300000. The corresponding experimental results are shown in Table 3 and Fig. 4. Secondly, the termination condition is set to a fixed time which is equal to 0.5s. The experimental results are illustrated in Table 4 and Fig. 5.

All the results are based on 50 independent runs on these four problems and presented in Table 3 and Table 4. The best, mean, worse and standard deviation of the objective function values, average time (*T_ave*) or evaluations (*FEs_ave*) are applied as the indicators to evaluate the results. The experimental settings are given in Table 1.

Table 3
Experimental results of FCSTA and three methods over 50 independent runs with a fixed number of evaluations on four engineering optimization problems.

Problem	Algorithm	*Best*	*Mean*	*Worst*	*Std.dev*	*T_ave(s)*
Welded beam design	FCSTA	1.70	1.70	1.70	0.00	**0.82**
	TLBO [36]	1.70	1.70	1.70	0.00	24.78
	CTSA [4]	1.70	1.70	1.70	0.00	15.74
	DEEM [46]	1.70	1.70	1.70	0.00	6.37
Tension/Compression spring design	FCSTA	0.01	0.01	0.01	0.00	**0.86**
	TLBO [36]	0.01	0.01	0.01	0.00	20.45
	CTSA [4]	0.01	0.01	0.01	0.00	13.44
	DEEM [46]	0.01	0.01	0.01	0.00	6.17
Pressure vessel design	FCSTA	5885.33	5885.33	5885.33	0.00	**0.80**
	TLBO [36]	5885.53	5885.55	5885.92	0.10	21.12
	CTSA [4]	5897.08	5945.64	6074.05	38.61	13.42
	DEEM [46]	5885.33	5885.33	5885.33	0.00	6.71
Power dispatch	FCSTA	1777658.14	1777658.14	1777658.14	0.00	**1.73**
	TLBO [36]	1777665.94	1777692.46	1778001.33	57.45	41.43
	CTSA [4]	1780729.95	1781652.88	1783022.44	513.42	25.38
	DEEM [46]	1777658.15	1777658.17	1777685.28	0.03	12.49

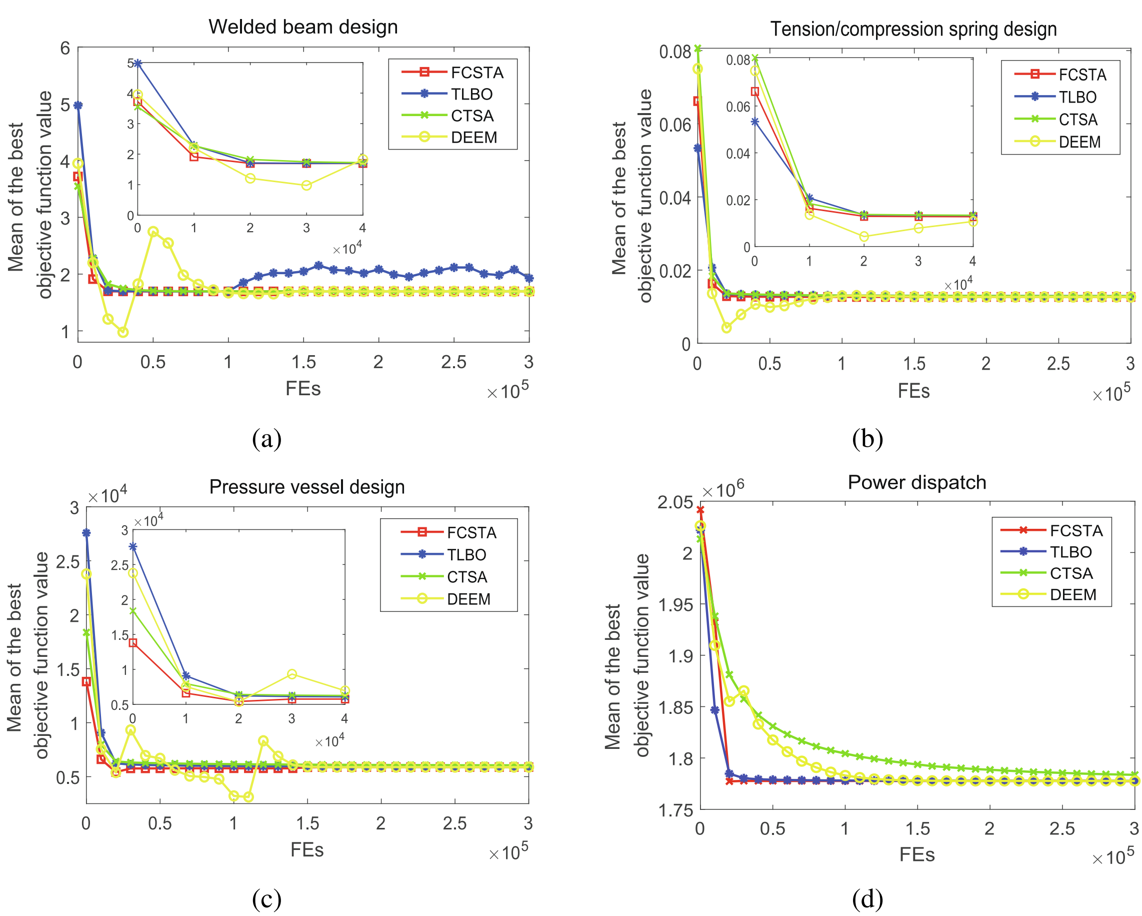

Fig. 4. Convergence graphs of FCSTA and three methods in 50 independent runs with a fixed number of evaluations on four engineering optimization problems..

Table 4
Experimental results of FCSTA and other three selected methods over 50 independent runs with a fixed time on four engineering optimization problems.

Problem	Algorithm	*Best*	*Mean*	*Worst*	*Std.dev*
Welded beam design	FCSTA	**1.70**	**1.70**	**1.70**	**0.00**
	TLBO [36]	1.77	1.96	2.32	0.12
	CTSA [4]	1.73	1.84	2.04	0.07
	DEEM [46]	1.78	1.90	2.14	0.08
Tension/Compression spring design	FCSTA	0.01	0.01	0.01	0.00
	TLBO [36]	0.01	0.01	0.01	0.00
	CTSA [4]	0.01	0.01	0.01	0.00
	DEEM [46]	0.01	0.01	0.02	0.00
Pressure vessel design	FCSTA	**5885.33**	**5885.33**	**5885.33**	**0.00**
	TLBO [36]	6048.00	6382.85	7201.90	275.67
	CTSA [4]	6195.50	6574.91	7153.19	233.96
	DEEM [46]	6419.55	6911.49	7593.50	316.99
Power dispatch	FCSTA	**1777658.14**	**1777658.14**	**1777658.15**	**0.00**
	TLBO [36]	1799534.19	1851628.49	1901026.57	21897.65
	CTSA [4]	1888285.69	1904731.56	1945983.77	14584.41
	DEEM [46]	1822438.35	1854913.84	1915420.01	21156.19

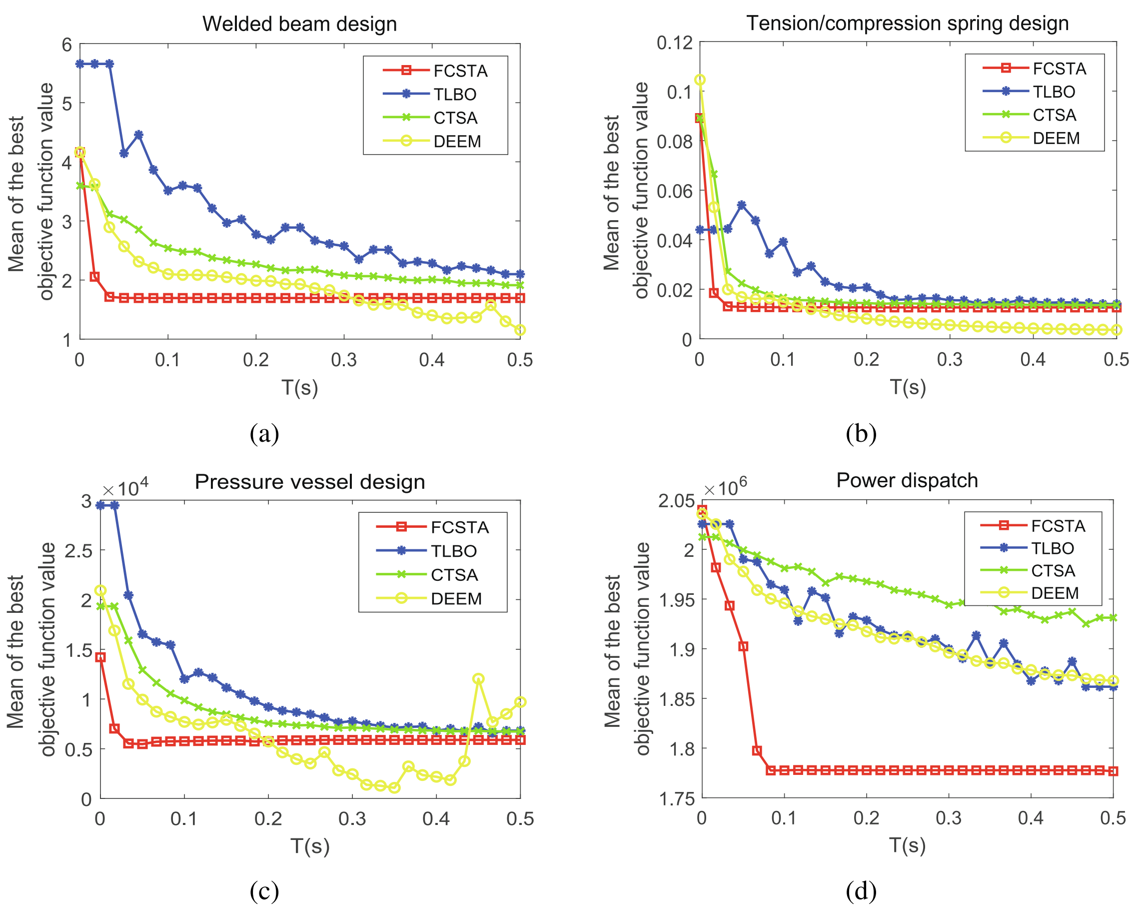

Fig. 5. Convergence graphs of FCSTA and other three methods in 50 independent runs with a fixed time on four engineering optimization problems..

The first set of experiment is performed with the fixed number of evaluations (300000), and the results are shown in Table 3 and Fig. 4.

- In Table 3, the experimental results are given in terms of the five indicators, including the best, mean, worst objective values, standard deviation, and average time. The experimental results show that FCSTA performs better than other algorithms with respect of the optimality, robustness and the computational efficiency on most engineering optimization problems. The best and worst indicators reflect the optimality and the smaller the values are, the better the performance achieves. The *Best* and *Worst* of FCSTA are smaller than that of TLBO and CTSA on pressure vessel design and power dispatch, while the optimality of FCSTA is the same as that of the other methods on the other two problems. The robustness is evaluated by means of the *Mean* and *Std.dev*. The results on the robustness performance of FCSTA are consistent with the results on the optimality. According to the average time (*T_ave(s)*), FCSTA outperforms TLBO, CTSA and DEEM and achieves over seven times faster

than these algorithms to find the optimal solutions on all tested applications. Accordingly, FCSTA performs better than other methods on these four engineering optimization problems.
- Fig. 4 presents the convergence curves of the four algorithms on four engineering problems. From the plots, the FCSTA has the fastest computational efficiency on four engineering optimization problems. FCSTA also shows good search ability to find the optimum while the computational efficiency of DEEM and TLBO fluctuates during the optimization.

The second set of experiment is conducted with the same fixed execution time (0.5s). The results are shown in Table 4 and Fig. 5.

- Table 4 shows that FCSTA outperforms other three methods on all indicator values. Firstly, the *Best* and the *Worst* values indicate that FCSTA achieves better performance on optimality than other methods on welded beam design, pressure vessel design and power dispatch, while it has the same performance on tension/compression spring design. Secondly, the advantage of FCSTA on the robustness is also apparent by comparing the *Mean* and *Std.dev* values of the best object values. Overall, the proposed method shows outstanding performance on these applications with respect of the optimality, robustness and computational efficiency.
- The convergence plots on these four applications are given in Fig. 5, where the execution time is set to 0.5s. The curves show that FCSTA has the higher computational efficiency compared with other methods, especially on welded beam design, pressure vessel design, and power dispatch. The curves of FCSTA are more stable than that of other methods and converge fast to the steady minimum value, which indicates the proposed method has better search ability to find the optimum on these problems. Combining the results in Fig. 4 and in Fig. 5, we find that there are many fluctuations in the convergence curves of DEEM and TLBO during the optimization process, as shown in Fig. 5, but the convergence curves of these two methods seem to be stable in Fig. 4. The phenomenon indicates that these two algorithms have to spend a large amount of time to reach the steady minimum values on different applications.

4.4. Effectiveness of the SQP and the modified constrained STA on FCSTA

In this section, an investigation is performed on the effectiveness of the SQP and modified STA on the performance of FCSTA, and the results are presented in Table 5 and Fig. 6. Here, CSTA is a constrained optimization algorithm based on STA and a two-stage constraint handling technique [17].'ICSTA_1' and 'ICSTA_2' are two variants of FCSTA without SQP operator. 'ICSTA_2' uses a novel constraint-handling technique compared to CSTA, and 'ICSTA_1' can be considered as a simplified 'ICSTA_2' which uses the simplified STA. The terminal condition is the fixed execution time and it is set to 0.2s.

- In Table 5, FCSTA outperforms 'ICSTA_1' and 'ICSTA_2' without SQP on pressure vessel design and power dispatch, and no worse than them on all tested applications. Notably, it means that FCSTA with SQP has better performance on optimality according to the *Best* and *Worst* values. Moreover, the robustness is also improved when the SQP is applied in FCSTA. But the *FEs_ave* value of FCSTA is less than that of 'ICSTA_1' and 'ICSTA_2', which means that the SQP takes more time to improve the convergence rate compared to CSTA. The simplified STA in 'ICSTA_1' can produce more solutions than 'ICSTA_2' with four transformations, which proves that the expansion transformation takes longer time to generate solutions than other three operators. Therefore, the simplified STA can take less time to generate the same number of candidate solutions. 'ICSTA_2' outperforms CSTA on Welded beam design, pressure vessel design and power dispatch, and no worse than them on all tested applications,which means that the novel constraint-handling technique is relatively simpler than the hybrid techniques but can obtain more diverse solutions, so that the convergence rate can be improved.
- In Fig. 6, the convergence curves of FCSTA are similar to that of 'ICSTA_1' and 'ICSTA_2' on welded beam design, tension/compression spring design and power dispatch. However, FCSTA is able to find smaller steady minimum values on tension/compression spring design and pressure vessel design, especially on power dispatch. FCSTA and its variants have better computational efficiency than CSTA on all applications except tension/-compression spring design problems. Overall, the SQP and simplified STA contribute significantly to FCSTA to find the optimum of these applications.

To sum up, it is found that the proposed FCSTA outperforms other compared approaches for the majority of all tested problems. There are three important components in FCSTA that can explain why it is superior to other approaches: Firstly, a simplified STA with adaptive parameter tuning is put forward, which can not only generate more potential candidate solutions but also consume less time. Secondly, a novel constraint-handling technique called branch and screen

Table 5
Experimental results of the FCSTA, the variants of FCSTA and CSTA in 50 independent runs with a fixed time on four engineering optimization problems.

Problem	Algorithm	*Best*	*Mean*	*Worst*	*Std.dev*	*FEs_ave*
Welded beam design	FCSTA	1.70	1.70	1.70	0.00	83674
	ICSTA_1	1.70	1.70	1.70	0.00	161931
	ICSTA_2	1.70	1.70	1.70	0.00	135492
	CSTA [17]	1.70	1.76	1.87	0.05	68867
Tension/Compression spring design	FCSTA	0.01	0.01	0.01	0.00	84940
	ICSTA_1	0.01	0.01	0.01	0.00	141867
	ICSTA_2	0.01	0.01	0.01	0.00	135171
	CSTA [17]	0.01	0.01	0.01	0.00	67716
Pressure vessel design	FCSTA	**5885.33**	**5885.33**	**5885.33**	**0**	82639
	ICSTA_1	5886.04	5968.84	6277.59	164.71	174879
	ICSTA_2	5885.67	6012.29	6795.55	210.78	148713
	CSTA [17]	5910.11	6497.81	7290.45	388.95	79364
Power dispatch	FCSTA	**1777658.14**	**1777658.24**	**1777659.11**	**0.20**	40061
	ICSTA_1	1787699.88	1818217.59	1875929.38	16258.64	51198
	ICSTA_2	1807761.01	1837694.28	1915711.45	27164.98	45227
	CSTA [17]	1817964.89	1868525.62	1969381.26	32573.87	20763

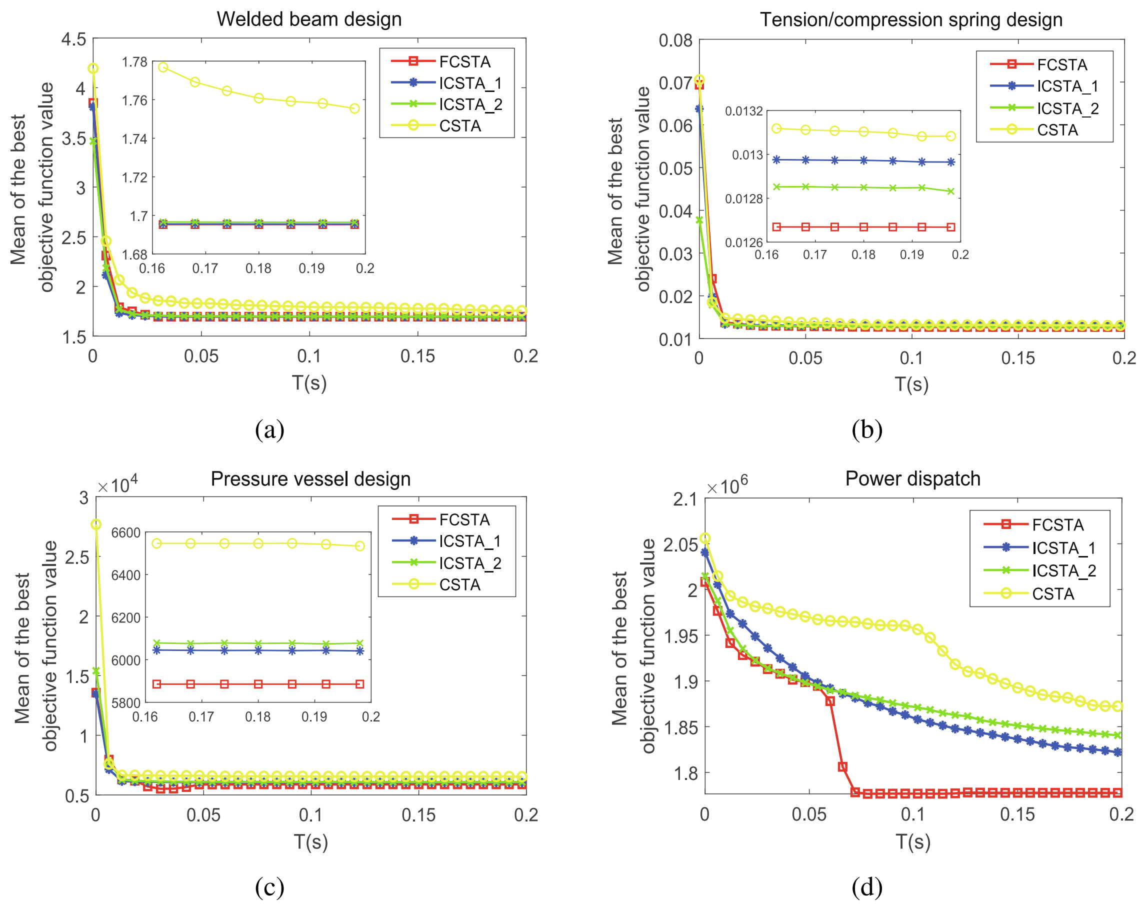

Fig. 6. Convergence graphs of the FCSTA, the variants of FCSTA and CSTA in 50 independent runs with a fixed time on four engineering optimization problems..

strategy is proposed to maintain diverse candidates including both feasible and infeasible solutions. Thirdly, a local search procedure based on SQP is used to further speed up the convergence.

5. Conclusion and future work

Recently, there are many methods proposed to deal with the constrained optimization problems. However, the majority of existing methods are slow to find the optima of the COPs due to time-consuming solution generation and selection during the optimization. To resolve this problem, this paper proposed a competitive constrained optimization algorithm which embeds the proposed branch and screen strategy into a simplified state transition algorithm. The branch and screen strategy considers the optimality, diversity, and computational efficiency simultaneously of the solutions during the constraint handling process, while the simplified state transition algorithm has strong exploration and exploitation abilities during the optimization. Once a feasible solution is found under a certain number of iterations, sequential quadratic programming is utilized to enhance the computational efficiency. A series of experiments have been conducted to test the performance of the proposed method, including 22 benchmark functions and 4 real industrial applications. The results demonstrated that the proposed method is competitive in finding the optima of the problems with a good balance between the robustness and fast convergency. In the future, we will continue to work on large scale constrained optimization problems based on decomposition method and constrained multi-objective optimization problems.

CRediT authorship contribution statement

Xiaojun Zhou: Writing - review & editing, Conceptualization, Investigation. **Jituo Tian:** Writing - review & editing, Software. **Jianpeng Long:** Writing - review & editing, Validation. **Yaochu Jin:** Writing - review & editing, Supervision. **Guo Yu:** Writing - review & editing, Supervision. **Chunhua Yang:** Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This study was funded by the National Natural Science Foundation of China (Grant Nos. 61873285, 72088101, 61860206014), the Research Foundation of Science and Technology of Hunan Province, China, under Grant 2019RS1003.

References

[1] Paul Armand, Riadh Omheni, A globally and quadratically convergent primal–dual augmented lagrangian algorithm for equality constrained optimization, Optim. Methods Softw. 32 (1) (2017) 1–21.
[2] Omar Abu Arqub and Zaer Abo-Hammour, Zaer Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inform. Sci. 279 (2014) 396–415.

[3] Omar Abu Arqub, Mohammed Al-Smadi, Fuzzy conformable fractional differential equations: novel extended approach and new numerical solutions, Soft. Comput. (2020) 1–22.
[4] Ahmet Babalik, Ahmet Cevahir Cinar, Mustafa Servet Kiran, A modification of tree-seed algorithm using Deb's rules for constrained optimization, Appl. Soft Computing 63 (2018) 289–305.
[5] Ricardo Landa Becerra, Carlos A. Coello, Cultured differential evolution for constrained optimization, Comput. Methods Appl. Mech. Eng. 195 (33–36) (2006) 4303–4322.
[6] Dimitri P Bertsekas, Nonlinear programming, Taylor & Francis, 1997.
[7] Paul T Boggs, Jon W Tolle, Sequential quadratic programming, Acta Numerica 4 (1995) 1–51.
[8] Ivona Brajevic, Crossover-based artificial bee colony algorithm for constrained optimization problems, Neural Comput. Appl. 26 (7) (2015) 1587–1601.
[9] ViníCius V. De Melo, Grazieli L.C. Carosio, Investigating multi-view differential evolution for solving constrained engineering design problems, Expert Syst. Appl. 40 (9) (2013) 3370–3377.
[10] Kalyanmoy Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2–4) (2000) 311–338.
[11] Leandro dos Santos Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, Expert Syst. Appl. 37 (2) (2010) 1676–1683.
[12] Raziyeh Farmani, Jonathan A. Wright, Self-adaptive fitness formulation for constrained optimization, IEEE Trans. Evol. Comput. 7 (5) (2003) 445–455.
[13] Harish Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016) 292–305.
[14] Noradin Ghadimi, Risk-assessment of photovoltaic-wind-battery-grid based large industrial consumer using information gap decision theory, Sol. Energy 169 (2018) 343–352.
[15] Noradin Ghadimi, Adel Akbarimajd, Hossein Shayeghi, Oveis Abedinia, Two stage forecast engine with feature selection technique and improved meta-heuristic algorithm for electricity load forecasting, Energy 161 (2018) 130–142.
[16] Abolfazl Gharaei, Seyed Hamid Reza Pasandideh, Four-echelon integrated supply chain model with stochastic constraints under shortage condition: sequential quadratic programming, Ind. Eng. Manage. Syst. 16 (3) (2017) 316–329.
[17] Jie Han, Chunhua Yang, Xiaojun Zhou, Weihua Gui, A two-stage state transition algorithm for constrained engineering optimization problems, Int. J. Control Autom. Syst. 16 (2) (2018) 522–534.
[18] Miao Huang, Xiaojun Zhou, Tingwen Huang, Chunhua Yang, Weihua Gui, Dynamic optimization based on state transition algorithm for copper removal process, Neural Comput. Appl. 31 (7) (2019) 2827–2839.
[19] Zhaoke Huang, Chunhua Yang, Xiaofang Chen, Keke Huang, Yongfang Xie, Adaptive over-sampling method for classification with application to imbalanced datasets in aluminum electrolysis, Neural Comput. Appl. (2019) 1–17.
[20] Zhaoke Huang, Chunhua Yang, Xiaojun Zhou, Weihua Gui, A novel cognitively inspired state transition algorithm for solving the linear bi-level programming problem, Cognitive Comput. 10 (5) (2018) 816–826.
[21] Zhaoke Huang, Chunhua Yang, Xiaojun Zhou, Tingwen Huang, A hybrid feature selection method based on binary state transition algorithm and ReliefF, IEEE J. Biomed. Health Inform. 23 (5) (2019) 1888–1898.
[22] Kazufumi Ito, Karl Kunisch, Lagrange Multiplier Approach to Variational Problems and Applications, Soc. Ind. Appl. Math. (2008), https://doi.org/10.1137/1.9780898718614.
[23] Suhong Jiang, Jin Zhang, Caihua Chen, Guihua Lin, Smoothing partial exact penalty splitting method for mathematical programs with equilibrium constraints, J. Global Optim. 70 (1) (2018) 223–236.
[24] Christian Kanzow, Daniel Steck, Augmented lagrangian and exact penalty methods for quasi-variational inequalities, Comput. Optim. Appl. 69 (3) (2018) 801–824.
[25] Dervis Karaboga, Bahriye Akay, A modified artificial bee colony (ABC) algorithm for constrained optimization problems, Appl. Soft Comput. 11 (3) (2011) 3021–3031.
[26] Hossein Khodaei, Mahdi Hajiali, Ayda Darvishan, Mohammad Sepehr, Noradin Ghadimi, Fuzzy-based heat and power hub models for cost-emission operation of an industrial consumer using compromise programming, Appl. Thermal Eng. (2018), https://doi.org/10.1016/j.applthermaleng.2018.04.008.
[27] J.J. Liang, Thomas Philip Runarsson, Efren Mezura-Montes, Maurice Clerc, Ponnuthurai Nagaratnam Suganthan, C.A. Coello Coello, Kalyanmoy Deb, Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization, J. Appl. Mech. 41 (8) (2006) 8–31.
[28] Jianjun Liu, Kok Lay Teo, Xiangyu Wang, Changzhi Wu, An exact penalty function-based differential search algorithm for constrained global optimization, Soft. Comput. 20 (4) (2016) 1305–1313.
[29] Wen Long, Ximing Liang, Shaohong Cai, Jianjun Jiao, Wenzhuan Zhang, A modified augmented lagrangian with improved grey wolf optimization to constrained optimization problems, Neural Comput. Appl. 28 (1) (2017) 421–438.
[30] Rammohan Mallipeddi, Ponnuthurai N. Suganthan, Ensemble of constraint handling techniques, IEEE Trans. Evol. Comput. 14 (4) (2010) 561–579.
[31] Seyedali Mirjalili, Andrew Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[32] Jorge Nocedal, Stephen Wright, Numerical optimization, Springer Science & Business Media, 2006.
[33] Antonin Ponsich, C.A. Coello CoelloCoello, Differential evolution performances for the solution of mixed-integer constrained process engineering problems, Appl. Soft Computing 11 (1) (2011) 399–409.
[34] Liqun Qi, Yu Fei Yang, A globally and superlinearly convergent sqp algorithm for nonlinear constrained optimization, J. Global Optim. 21 (2) (2001) 157–184.
[35] R. Rao, Vivek Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, Int. J. Ind. Eng. Comput. 3 (4) (2012) 535–560.
[36] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.
[37] Thomas Philip Runarsson, Xin Yao, Search biases in constrained evolutionary optimization, IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.) 35 (2) (2005) 233–243.
[38] Mohammadhossein Saeedi, Mahdi Moradi, Meysam Hosseini, Armin Emamifar, Noradin Ghadimi, Robust optimization based optimal chiller loading under cooling demand uncertainty, Appl. Therm. Eng. 148 (2019) 1081–1091.
[39] S. Shadravan, H.R. Naji, Vahid Khatibi Bardsiri, The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, Eng. Appl. Artif. Intell. 80 (2019) 20–34.
[40] Tetsuyuki Takahama, Setsuko Sakai, Constrained optimization by the <texmath type="inline">varepsilon </texmath> constrained differential evolution with gradient-based mutation and feasible elites, in: 2006 IEEE International Conference on Evolutionary Computation, IEEE, 2006, pp. 1–8.
[41] M. Fatih Tasgetiren, P. Nagaratnam Suganthan, Quan-Ke Pan, Rammohan Mallipeddi, Sedat Sarman. An ensemble of differential evolution algorithms for constrained function optimization. In IEEE Congress on Evolutionary Computation, pages 1–8. IEEE, 2010..
[42] Qi'an Wang, Miao Huang, Xiaojun Zhou, Feature selection in froth flotation for production condition recognition, IFAC-PapersOnLine 51 (21) (2018) 123–128.
[43] Yong Wang, Zixing Cai, Guanqi Guo, Yuren Zhou, Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems, IEEE Trans. Syst., Man, Cybern., Part B (Cybern.) 37 (3) (2007) 560–575.
[44] Yong Wang, Zixing Cai, Yuren Zhou, Zhun Fan, Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique, Struct. Multidisciplinary Optim. 37 (4) (2009) 395–413.
[45] Yong Wang, Zixing Cai, Yuren Zhou, Wei Zeng, An adaptive tradeoff model for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 12 (1) (2008) 80–92.
[46] Yong Wang, Hao Liu, Huan Long, Zijun Zhang, Shengxiang Yang, Differential evolution with a new encoding mechanism for optimizing wind farm layout, IEEE Trans. Industr. Inf. 14 (3) (2018) 1040–1054.
[47] Yong Wang, Bingchuan Wang, Hanxiong Li, Gary G Yen, Incorporating objective function information into the feasibility rule for constrained evolutionary optimization, IEEE Trans. Cybern. 46 (12) (2015) 2938–2952.
[48] Wu. Guohua, Witold Pedrycz, Ponnuthurai N Suganthan, Rammohan Mallipeddi, A variable reduction strategy for evolutionary algorithms handling equality constraints, Appl. Soft Computing 37 (2015) 774–786.
[49] Yongfang Xie, Simi Wei, Xiaoli Wang, Sen Xie, Chunhua Yang, A new prediction model based on the leaching rate kinetics in the alumina digestion process, Hydrometallurgy 164 (2016) 7–14.
[50] Chunhua Yang, Shijun Deng, Yonggang Li, Hongqiu Zhu, Fanbiao Li, Optimal control for zinc electrowinning process with current switching, IEEE Access 5 (2017) 24688–24697.
[51] Yu. Yang, Improved benders decomposition and feasibility validation for two-stage chance-constrained programs in process optimization, Ind. Eng. Chem. Res. 58 (12) (2019) 4853–4865.
[52] Xiaojun Zhou, Miao Huang, Tingwen Huang, Chunhua Yang, Weihua Gui, Dynamic optimization for copper removal process with continuous production constraints, IEEE Trans. Industr. Inf. (2019), https://doi.org/10.1109/TII.2019.2943500.
[53] Xiaojun Zhou, Jianpeng Long, Xu. Chongchong, Guanbo Jia, An external archive-based constrained state transition algorithm for optimal power dispatch, Complexity (2019), https://doi.org/10.1155/2019/4727168.
[54] Xiaojun Zhou, Chunhua Yang, Weihua Gui, State transition algorithm, J. Ind. Manage. Optim. 8 (4) (2012) 1039–1056.
[55] Xiaojun Zhou, Chunhua Yang, Weihua Gui, Nonlinear system identification and control using state transition algorithm, Appl. Math. Comput. 226 (2014) 169–179.
[56] Xiaojun Zhou, Chunhua Yang, Weihua Gui, A statistical study on parameter selection of operators in continuous state transition algorithm, IEEE Trans. Cybern. 49 (10) (2018) 3722–3730.
[57] Xiaojun Zhou, Ke Yang, Yongfang Xie, Chunhua Yang, Tingwen Huang, A novel modularity-based discrete state transition algorithm for community detection in networks, Neurocomputing 334 (2019) 89–99.
[58] Xiaojun Zhou, Jiajia Zhou, Chunhua Yang, Weihua Gui, Set-point tracking and multi-objective optimization-based PID control for the goethite process, IEEE Access 6 (2018) 36683–36698.

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 system, artificial intelligence and machine learning, optimization theory and algorithms, duality theory and global optimization with applications. E-mail: michael.x.zhou@csu.edu.cn.

Jituo Tian received his Bachelor's degree in Automation in 2020 from Central South University, Changsha, China. He is currently a master student at Central South University, Changsha, China. His main interests include constrained optimization, bi-level optimization, and optimal control of complex industrial process. E-mail: jituo.tian@csu.edu.cn.

Jianpeng Long received her Bachelor's degree in Automation in 2017 from Hunan University of Technology, Changsha, China and received her M.Eng. in Control Science and Engineering from Central South University, China in 2020. Her main interests include constrained optimization, data-driven optimization and state transition algorithm. E-mail: vikilong@csu.edu.cn.

Yaochu Jin received the B.Sc., M.Sc., and Ph.D. degrees in automatic control from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. Degree from Ruhr-University Bochum, Bochum, Germany, in 2001. He is a Distinguished Chair Professor of computational intelligence with the Department of Computer Science, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He was a Finland Distinguished Professor funded by the Finnish Funding Agency for Innovation (Tekes) and a Changjiang Distinguished Visiting Professor appointed by the Ministry of Education, China. His main research interests include data-driven surrogate-assisted evolutionary optimization, secure machine learning, multiobjective evolutionary deep learning, swarm robotics, and evolutionary developmental systems. Dr. Jin is the recipient of the 2018 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, the 2015, 2017, and 2020 IEEE Computational Intelligence Magazine Outstanding Paper Award, and the Best Paper Award of the CIBCB 2010. He is recognized as a Highly Cited Researcher 2019 by the Web of Science Group. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Complex & Intelligent Systems. He was an IEEE Distinguished Lecturer, and the Vice President for Technical Activities of the IEEE Computational Intelligence Society. He was the General Co-Chair of the 2016 IEEE Symposium Series on Computational Intelligence and the 2020 IEEE Congress on Evolutionary Computation. E-mail: yaochu.jin@surrey.ac.uk.

Guo Yu received the B.S. degree in information and computing science and the M.Eng. degree in computer technology from Xiangtan University, Xiangtan, China, in 2012 and 2015, respectively, and the Ph.D. degree in computer science from the University of Surrey, Guildford, U.K., in 2020. His current research interests include preference learning, evolutionary optimization, and machine learning. Dr. Yu is a Regular Reviewer of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, Artificial Intelligence, IEEE Computational Intelligence Magazine, IEEE ACCESS, Complex & Intelligent Systems, Applied Soft Computing, and Soft Computing. E-mail: guo.yu@surrey.ac.uk.

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. E-mail: ychh@csu.edu.cn.