

# 求解约束优化问题的协同进化教与学优化算法

刘三阳<sup>1</sup> 靳安钊<sup>1</sup>

**摘要** 对约束优化问题, 为了避免罚因子和等式约束转化为不等式约束时引入的约束容忍度参数所带来的不便, 本文在基本教与学优化 (Teaching-learning-based optimization, TLBO) 算法中加入了自我学习过程并提出了一种求解约束优化问题的协同进化教与学优化算法, 使得罚因子和约束容忍度随种群的进化动态调整. 对 7 个常见测试函数的数值实验验证了算法求解带有等式和不等式约束优化问题的有效性.

**关键词** 教与学优化算法, 协同进化, 罚因子, 约束容忍度, 种群多样性

**引用格式** 刘三阳, 靳安钊. 求解约束优化问题的协同进化教与学优化算法. 自动化学报, 2018, 44(9): 1690–1697

**DOI** 10.16383/j.aas.2017.c170076

## A Co-evolutionary Teaching-learning-based Optimization Algorithm for Constrained Optimization Problems

LIU San-Yang<sup>1</sup> JIN An-Zhao<sup>1</sup>

**Abstract** In order to avoid the inconvenience of penalty factors and the tolerance amount during transforming equality constraints into inequality constraints, the self-learning process is combined with teaching-learning-based algorithm, and a co-evolutionary teaching-learning-based algorithm is thus proposed, which makes the penalty factors and tolerance amounts dynamically adjust along with the population evolution. Numerical experiments on seven common test functions verify the effectiveness of the algorithm to solve optimization problems with equality and inequality constraints.

**Key words** Teaching-learning-based optimization algorithm, co-evolutionary, penalty factor, tolerance amount, population diversity

**Citation** Liu San-Yang, Jin An-Zhao. A co-evolutionary teaching-learning-based optimization algorithm for constrained optimization problems. *Acta Automatica Sinica*, 2018, 44(9): 1690–1697

约束优化问题是科学研究和工程应用领域经常遇到的一类数学规划问题, 它广泛存在但又难以求解, 因此对求解约束优化问题有效方法的研究具有重要的理论和实际意义. 目前存在的约束优化算法大致可以分为两类: 传统优化算法和现代智能优化算法. 传统优化算法大多是基于梯度信息的寻优方法, 如可行方向法、投影梯度法和既约梯度法等<sup>[1]</sup>. 它们对于不可导, 可行域不连通, 甚至没有显式表达式的问题束手无策. 现代智能优化算法主要包括差分进化算法 (Differential evolutionary, DE)<sup>[2]</sup>、粒子群算法 (Particle swarm optimization, PSO)<sup>[3–5]</sup>、遗传算法 (Genetic algorithm, GA)<sup>[6]</sup>、蚁群算法 (Ant colony optimization, ACO)<sup>[7]</sup>、人工蜂群算法 (Artificial bee colony, ABC)<sup>[8]</sup> 等. 近年

来, 现代智能算法得到了广泛的研究, 在求解约束优化问题方面取得了显著的成效.

大多数现代智能优化算法都是一种无约束搜索技术, 将其与约束处理技术结合的过程中需要引入一些额外的参数, 这些参数通常由使用者决定, 选取合适的参数一般需要经过多次试验不断调节.

罚函数法是应用最广泛的约束处理技巧, 其主要思想就是通过对目标函数  $f(x)$  增加惩罚项  $p(x)$  来构造适应度值函数  $fitness(x)$ . 惩罚适应度值函数的构造过程会引入参数罚因子  $w$ , 罚因子选取太小则个体的适应度值主要由目标函数决定, 此时群体可能滞留在不可行区域, 很难找到最优解; 罚因子选取太大则种群快速进入可行域, 从而忽略了对不可行区域的搜索, 这样就不能充分利用不可行解提供的信息<sup>[9]</sup>, 从而很难找到最优解.

此外, 还有很多约束优化问题带有等式约束, 等式约束的出现使得约束优化问题的求解变得更加困难. 等式约束的出现极大地缩小了可行区域的范围, 对搜索最优解带来了极大的困难<sup>[10]</sup>. 一般的等式约束处理技术都是将等式约束转化为不等式约束, 即将等式约束  $h(x) = 0$  转化为

收稿日期 2017-02-14 录用日期 2017-06-22  
Manuscript received February 14, 2017; accepted June 22, 2017  
国家自然科学基金 (61373174) 资助  
Supported by National Natural Science Foundation of China (61373174)  
本文责任编辑 王占山  
Recommended by Associate Editor WANG Zhan-Shan  
1. 西安电子科技大学数学与统计学院 西安 710126  
1. Department of Mathematic Science, Xidian University, Xi'an 710126

$$|h(x)| - \delta \leq 0 \quad (1) \quad \text{更好:}$$

其中,  $\delta$  为等式约束的约束容忍度参数, 一般取较小的正数. 这一转化过程改变了可行域的拓扑结构,  $\delta$  的大小直接决定了这种拓扑结构改变程度, 也决定了可行区域的大小. 因此设置合适的约束容忍度参数  $\delta$  是十分必要的, 然而参数  $\delta$  的合理设置是一项十分困难的工作.

基于以上的这些不足, 本文提出了一种求解约束优化问题的协同进化教与学优化算法 (Co-evolutionary teaching-learning-based optimization algorithm, CTLBO). 该方法在基本教与学优化算法过程中加入了一个随机变异搜索的自我学习机制, 增强了算法局部搜索能力. 文章首次提出将约束容忍度参数加入进化参数集中和罚因子一起自动进化, 避免了人工调节约束容忍度参数的困难, 同时也实现了约束容忍度参数自适应地进化, 有效地提高了求解约束优化问题的效率, 尤其是含有等式约束的优化问题. 同时在计算个体适应度时, 将已有协同进化算法适应度值公式中的可行个体均值项替换为可行个体的方差, 并根据进化时期的不同阶段, 采用不同的适应度值计算公式, 有效地保证了种群的多样性.

### 1 基本教与学优化算法

教与学优化算法 (Teaching-learning-based optimization, TLBO)<sup>[11-12]</sup> 由 Rao 等于 2010 年提出, 它模拟了教师给学员教学的过程. 教与学优化算法具有参数少、算法简单、易理解和求解精度高等特点. 自教与学优化算法提出以后, 研究人员对其进行了深入研究, 并提出了许多改进算法<sup>[13-15]</sup>, 这些算法都取得了较好的效果.

教与学优化算法中的教师和学员相当于进化算法中的个体. 教师是适应度值最好的个体, 学员学习的每一个科目对应于一个决策变量.

#### 1.1 教学阶段

学员通过两个方面提高自己的成绩, 向教师学习和向同学学习. 教学阶段是向教师学习的过程, 学习阶段是向其他学员学习的过程. 假设在第  $k$  次迭代的过程中,  $M_k$  是学员的平均成绩,  $T_k$  是教师. 此时学员将会根据全体学员平均成绩与教师之间的差距来进行学习, 差距由下式给出:

$$\text{Difference\_Mean}_k = r_k(T_k - T_F M_k) \quad (2)$$

其中,  $r_k$  是 0 到 1 之间的随机数,  $T_F$  是一个学习因子, 取值为 1 到 2 之间的随机数. 实验表明, 当  $T_F$  按照下式进行 1 或 2 的随机取值时算法取得的结果

$$T_F = \text{round}[1 + \text{rand}(0, 1)\{2 - 1\}] \quad (3)$$

在教学阶段, 学员按照下面式子更新自己的成绩:

$$X_{\text{new}} = X_{\text{old}} + \text{Difference\_Mean}_k \quad (4)$$

如果更新后  $X_{\text{new}}$  对应的函数值优于  $X_{\text{old}}$  对应的函数值, 则接受  $X_{\text{new}}$ , 否则拒绝  $X_{\text{new}}$ .

#### 1.2 学习阶段

在学习阶段, 学员按照下面方式进行更新成绩: 对于学员  $X_i$ , 随机选取一个学员  $X_j$  ( $i \neq j$ ), 如果  $f(X_i) < f(X_j)$ , 则

$$X_{\text{new},i} = X_{\text{old},i} + r_i(X_i - X_j), \quad i, j = 1, 2, \dots, NP \quad (5)$$

否则

$$X_{\text{new},i} = X_{\text{old},i} + r_i(X_j - X_i), \quad i, j = 1, 2, \dots, NP \quad (6)$$

如果更新后  $X_{\text{new},i}$  对应的函数值优于  $X_{\text{old},i}$  对应的函数值, 则接受  $X_{\text{new},i}$ ; 否则, 拒绝  $X_{\text{new},i}$ , 其中,  $i = 1, 2, \dots, NP$ .

## 2 协同进化教与学约束优化算法 (CTLBO)

### 2.1 算法模型

Coello<sup>[16]</sup> 提出了一种求解约束优化问题的协同进化遗传算法, He 等<sup>[17]</sup> 提出了求解约束优化问题的协同进化粒子群算法, Huang 等<sup>[18]</sup> 提出了求解约束优化问题的协同进化差分进化算法 (Co-evolutionary differential evolution, CDE). 这些算法在求解工程约束优化问题时取得了较好的结果. 本文基于这些算法提出一种协同进化教与学约束优化算法, 图 1 展示了算法模型.

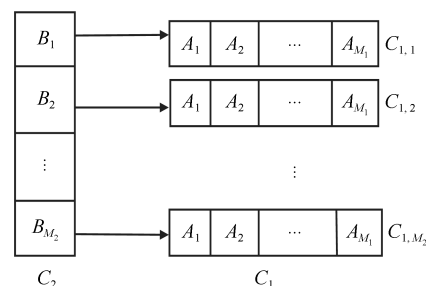


图 1 算法模型

Fig. 1 Algorithm model

在协同进化教与学约束优化算法中, 班级  $C_2$  包含  $M_2$  个学员, 班级  $C_1$  包含  $M_2$  个子班级, 每个子

班级包含  $M_1$  个学员. 班级  $C_2$  中的学员  $B_i$  的成绩代表着班级  $C_1$  中子班级  $C_{1,i}$  的学员进化过程中的罚因子  $\omega$  和约束容忍度  $\delta$ . 在协同进化的每一代, 班级  $C_1$  中的学员首先进行自我学习过程, 然后使用班级  $C_2$  提供的参数集用教与学优化算法进化  $G_1$  代. 当班级  $C_1$  完成  $G_1$  代进化之后, 班级  $C_2$  中每个学员的适应度值根据其所对应的  $C_1$  中的子班级学员位置分布情况进行更新, 然后对班级  $C_2$  用教与学优化算法进化一代. 重复上述进化过程, 直到达到最大进化代数  $G_2$ .

## 2.2 自我学习过程

用教与学优化算法进化班级  $C_1$  时, 在基本教与学优化算法中加入了一个增加算法搜索能力的随机变异自我学习机制. 对班级  $C_{1,j}$  的每一个学员  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$ , 随机选取其第  $j$  维变量  $x_{i,j}$  构造  $X_{\text{new1}}$  如下:

$$x_{\text{new1},i,k} = \begin{cases} x_{i,k}, & k \neq j \\ x_{i,j} + \text{rand}(x_{\max}(j) - x_{i,j}), & k = j \end{cases}$$

如果  $\text{fitness}(X_{\text{new1}}) < \text{fitness}(X_i)$ , 则  $X_i = X_{\text{new1}}$ ; 否则构造  $X_{\text{new2}}$  如下:

$$x_{\text{new2},i,k} = \begin{cases} x_{i,k}, & k \neq j \\ x_{i,j} + \text{rand}(x_{i,j} - x_{\min}(j)), & k = j \end{cases}$$

如果  $\text{fitness}(X_{\text{new2}}) < \text{fitness}(X_i)$ , 则  $X_i = X_{\text{new2}}$ . 此随机变异过程与高斯随机扰动相比, 操作较为简单且容易理解, 避免了人工设置高斯扰动的参数均值和方差.

## 2.3 进化班级 $C_1$

对于下面的约束优化问题:

$$\begin{aligned} \min & f(x) \\ \text{s. t.} & g_i(x) \leq 0, \quad i = 1, 2, \dots, n \\ & h_j(x) = 0, \quad j = 1, 2, \dots, p \end{aligned} \quad (7)$$

根据 Richardson 等<sup>[19]</sup> 提出的罚函数构造准则, 考虑每个解的约束违反量和约束违反数目, 构造下面适应度值函数:

$$\text{fitness}(x) = f(x) + \omega_1 \times \text{sum\_viol}(x) + \omega_2 \times \text{num\_viol}(x) \quad (8)$$

其中,  $\text{sum\_viol}(x)$  代表解  $x$  所对应的约束违反量,  $\text{num\_viol}(x)$  代表解  $x$  所对应的约束违反数目,  $\omega_1, \omega_2$  分别是它们对应的罚因子.

对于约束违反量, 按下面方式得到:

$$\text{sum\_viol}(x) = \sum_{i=1}^N g_i(x), \forall g_i(x) > 0 \quad (9)$$

其中,  $N = n + p$ , 此处假设所有的等式约束已经按式 (1) 转化为了不等式约束.

对班级  $C_1$  的每个学员按式 (8) 确定的适应度函数值使用 TLBO 算法进化一代.

## 2.4 进化班级 $C_2$

班级  $C_2$  中的每一个学员代表一个参数集 ( $\omega_1, \omega_2$  和  $\delta$ ). 当班级  $C_1$  中的子班级  $C_{1,j}$  进化  $G_1$  代之后, 班级  $C_2$  中学员  $B_j$  的适应度值按下面方式确定:

1) 当子班级  $C_{1,j}$  中至少含有一个可行解时,  $B_j$  称为有效学员, 适应度值可以分两个阶段计算: 进化初期阶段和进化后期阶段. 进化初期阶段, 算法目标是增加可行解数量的同时增加种群多样性, 此时  $B_j$  的适应度值函数可以表示为

$$\text{fitness}(B_j) = \frac{\sum (\text{objective}_{j,i} - \text{mean\_objective}_j)^2}{\text{num\_feasible}} \quad (10)$$

进化后期阶段, 算法目标是尽快收敛到最优解, 此时  $B_j$  的适应度值函数可以表示为

$$\text{fitness}(B_j) = \frac{\sum (\text{objective}_{j,i} - \text{mean\_objective}_j)^2}{\text{num\_feasible}} \quad (11)$$

其中,  $\text{objective}_{j,i}$  表示子班级  $C_{1,j}$  中可行解  $i$  的目标函数值,  $\text{mean\_objective}_j$  表示子班级  $C_{1,j}$  中可行解目标函数值的均值,  $\text{num\_feasible}$  表示子班级  $C_{1,j}$  中可行解的数目. 由概率统计知识可知, 方差比均值更能反映种群的波动情况. 与已有协同进化算法 [16–18] 中的适应度值函数

$$\text{fitness}(B_j) = -\frac{\sum \text{feasible}}{\text{num\_feasible}} - \text{num\_feasible}$$

相比较, 等式 (10) 和 (11) 中的可行解方差项  $\frac{\sum (\text{objective}_{j,i} - \text{mean\_objective}_j)^2}{\text{num\_feasible}}$  比已有算法中的可行解均值项  $\frac{\sum \text{feasible}}{\text{num\_feasible}}$  更能反映可行解的波动情况, 更能体现种群多样性. 在等式 (10) 中, 当种群可行解数目  $\text{num\_feasible}$  越大, 种群可行解多样性越高, 即  $\frac{\sum (\text{objective}_{j,i} - \text{mean\_objective}_j)^2}{\text{num\_feasible}}$  越大时, 目标函数适应度值越小, 因此在种群选择阶段能够偏向更大可行解数量和更高种群多样性的种群. 在等式 (11) 中, 即进化后期阶段, 可行解个数越多, 可行解分布越集中即可行解方差越小, 适应度函数值越小. 由于此时种群中的大多数解已经进化为可行解, 且分布较为

集中, 因此这样的适应度函数能够在搜索过程中保证算法快速地收敛到最优解.

2) 当子班级  $C_{1,j}$  中不存在可行解时, 则称  $B_j$  为无效学员, 此时  $B_j$  适应度值函数如下

$$\text{fitness}(B_j) = \max(P_{\text{valid}}) + \frac{\sum \text{sum\_viol}}{\sum \text{num\_viol}} + \sum \text{num\_viol} \quad (12)$$

其中,  $\max(P_{\text{valid}})$  表示有效学员的最大目标函数值,  $\sum \text{sum\_viol}$  表示  $C_{1,j}$  中学员成绩的约束违反量之和,  $\sum \text{num\_viol}$  表示  $C_{1,j}$  中学员成绩的约束违反数目之和. 此时种群不存在可行解, 算法的首要目标是使种群搜索到可行解, 因此具有较小约束违反度的个体优先被选择.  $\max(P_{\text{valid}})$  项的存在保证了最优可行个体始终优于不可行个体, 从而使种群在选择过程中优先选择可行个体, 进而使算法在搜索过程中偏向可行区域.  $\sum \text{num\_viol}$  项的存在可以使约束违反度较小的个体优先被选择. 因此式 (12) 可以保证种群朝向可行区域搜索, 从而加快算法的寻优速度.

### 3 算法流程图

通过以上对协同进化教与学约束优化算法的描述, 算法流程如图 2 所示.

### 4 仿真分析

为了验证算法求解含有等式和不含等式约束优化问题的性能, 本文对 13 个标准测试函数 (测试函数见文献 [20]) 中带有等式约束问题  $g05, g11, g13$  和不带等式约束问题  $g06$  以及三个工程问题进行了求解. 此处仅给出三个工程优化问题的数学表示.

**问题 1.** (Welded beam design problem)

问题 1 取自 Coello<sup>[16]</sup> 的焊接梁设计最小费用问题, 模型如图 3 所示.

它含有 4 个决策变量  $h, l, t$  和  $b$ ,  $x_1$  代表  $h$ ,  $x_2$  代表  $l$ ,  $x_3$  代表  $t$ ,  $x_4$  代表  $b$ , 问题可以用下面优化问题表示:

$$\begin{aligned} \min f(x) &= 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \\ \text{s. t. } g_1(x) &= \tau(x) - \tau_{\max} \leq 0 \\ g_2(x) &= \sigma(x) - \sigma_{\max} \leq 0 \\ g_3(x) &= x_1 - x_4 \leq 0 \\ g_4(x) &= 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \end{aligned}$$

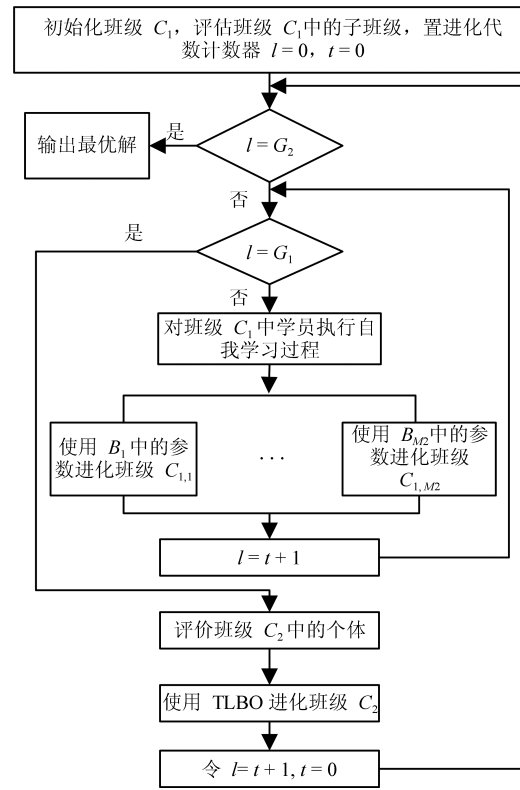


图 2 算法流程

Fig. 2 Flow chart of CTLBO

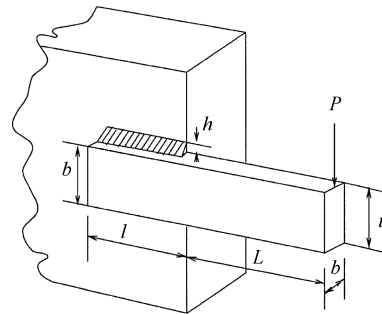


图 3 焊接梁模型图

Fig. 3 Model of welded beam

$$\begin{aligned} g_5(x) &= 0.125 - x_1 \leq 0 \\ g_6(x) &= \delta(x) - \delta_{\max} \leq 0 \\ g_7(x) &= P - P_c(x) \leq 0 \end{aligned}$$

其中,  $\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{R} + (\tau'')^2}$ ,  $\tau' = \frac{P}{\sqrt{2}x_1x_2}$ ,  $\tau'' = \frac{MR}{J}$ ,  $M = P(L + \frac{x_2}{2})$ ,  $\sigma(x) = \frac{6PL}{x_4x_3^2}$ ,  $L = 14 \text{ in}$ ,  $R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2}$ ,  $\delta(x) = \frac{4PL^3}{E x_3^3 x_4}$ ,  $P = 6000 \text{ lb}$ ,  $J = 2\{\sqrt{2}x_1x_2[\frac{x_2^2}{12} + (\frac{x_1+x_3}{2})^2]\}$ ,  $E = 30 \times 10^6 \text{ psi}$ ,  $P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}})$ ,  $G = 12 \times 10^6 \text{ psi}$ ,  $\tau_{\max} = 13600 \text{ psi}$ ,  $\sigma_{\max} = 30000 \text{ psi}$ ,

$\delta_{\max} = 0.25 \text{ in}$ ,  $0.1 \leq x_1 \leq 2$ ,  $0.1 \leq x_2 \leq 10$ ,  $0.1 \leq x_3 \leq 10$ ,  $0.1 \leq x_4 \leq 2$ .

**问题 2.** (Pressure vessel problem)

问题 2 取自于工程上的压力容器设计最小费用问题, 它的数学表示由 Kannan<sup>[21]</sup> 给出, 模型如图 4 所示.

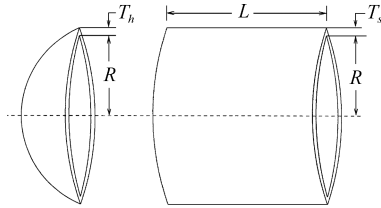


图 4 压力容器模型

Fig. 4 Model of pressure vessel

其中,  $x_1$  表示  $T_s$ ,  $x_2$  表示  $T_h$ ,  $x_3$  表示  $R$ ,  $x_4$  表示  $L$ ,  $x_1$  和  $x_2$  的值取为整数然后乘以 0.0625, 问题可以用下面优化问题表示:

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

其中,  $1 \leq x_1 \leq 99$ ,  $1 \leq x_2 \leq 99$ ,  $10 \leq x_3 \leq 200$ ,  $10 \leq x_4 \leq 200$ .

**问题 3.** (A tension string design problem)

问题 3 是工程上的最小化张力弦的重量问题, 模型如图 5 所示.

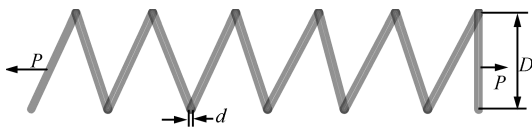


图 5 最小化张力弦模型

Fig. 5 Model of tension string

它由 Arora<sup>[22]</sup> 和 Belegundu 等<sup>[23]</sup> 给出了数学表示, 其中  $x_1$  代表  $d$ ,  $x_2$  代表  $D$ ,  $x_3$  代表  $P$ , 问题可以用下面的优化问题表示:

$$\begin{aligned} \min f(x) &= (x_3 + 2)x_2x_1^2 \\ \text{s. t. } g_1(x) &= 1 - \frac{x_3^3x_2}{71785x_1^4} \leq 0 \\ g_2(x) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\ g_3(x) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\ g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0 \end{aligned}$$

其中,  $0.05 \leq x_1 \leq 2$ ,  $0.25 \leq x_2 \leq 1.3$ ,  $2 \leq x_3 \leq 15$ . 本文算法参数设置如下:  $M_1 = 50$ ,  $M_2 = 10$ ,  $G_1 = 30$ ,  $G_2 = 8$ , 对于问题 1 和问题 2 中取  $\omega_{1,\min} = \omega_{2,\min} = 0$ ,  $\delta_{\min} = 10^{-6}$ ,  $\omega_{1,\max} = \omega_{2,\max} = 1000$ ,  $\delta_{\max} = 10^{-2}$ . 问题 3 中取  $\omega_{1,\min} = \omega_{2,\min} = 5000$ ,  $\delta_{\min} = 10^{-6}$ ,  $\omega_{1,\max} = \omega_{2,\max} = 25000$ ,  $\delta_{\max} = 10^{-2}$ . 问题 g05 中取  $\omega_{1,\min} = \omega_{2,\min} = 1000$ ,  $\delta_{\min} = 10^{-6}$ ,  $\omega_{1,\max} = \omega_{2,\max} = 3000$ ,  $\delta_{\max} = 10^{-2}$ . 问题 g06 中取  $\omega_{1,\min} = \omega_{2,\min} = 0$ ,  $\delta_{\min} = 10^{-6}$ ,  $\omega_{1,\max} = \omega_{2,\max} = 10000$ ,  $\delta_{\max} = 10^{-2}$ . 问题 g11 中取  $\omega_{1,\min} = \omega_{2,\min} = 0$ ,  $\delta_{\min} = 10^{-10}$ ,  $\omega_{1,\max} = \omega_{2,\max} = 10$ ,  $\delta_{\max} = 10^{-5}$ . 问题 g13 中取  $\omega_{1,\min} = \omega_{2,\min} = 0$ ,  $\delta_{\min} = 10^{-8}$ ,  $\omega_{1,\max} = 1$ ,  $\omega_{2,\max} = 0.1$ ,  $\delta_{\max} = 10^{-3}$ .

对于问题 1~3, 算法独立运行 30 次, 它们的最优解未知, 只有目前求得的最优解作为参考. 运行结果如表 1~6 所示 (NA 表示原文没有提供具体数据).

对问题 g05, g06, g11 和 g13 独立运行 30 次, 运行结果和其他算法比较如表 7 所示.

通过对表 1~6 的实验结果进行对比, 协同进化教与学约束优化算法对三个工程问题求得了更好的解, 并且在求解精度和稳定性上都要优于表中列出的算法, 其标准差明显小于文章列出的比较算法. 通过表 7 可以看出, 算法在对 4 个标准测试函数 g05, g06, g11 和 g13 的求解过程中也表现出了良好的性能. 对函数 g06 和 g11, 本文算法和对比算法达到了同样好的效果, 对其他两个函数, 本文算法的求解结果要优于其他对比算法.

**5 结论**

本文在基本教与学优化算法中加入了一种自我学习机制, 然后对已有协同进化算法的适应度值函数进行改进, 提出了一种可以增加种群多样性的适应度值函数, 并首次提出将等式约束对应的约束容忍度参数加入到协同进化参数集来实现参数自适应进化, 避免了人工调整参数的困难. 在此基础上提出了一种求解约束优化问题的协同进化教与学约束优化算法 (CTLBO), 该算法结合了协同进化算法和教与学优化算法的优良特性, 在算法过程中没有引入额外的参数, 保持了教与学优化算法参数较少这一特点. 对三个等式约束优化问题和 4 个不等式约束优化问题进行了仿真分析, 并与其他一些算法进行比较, 结果表明本文所提出的算法在求解带有等式约束和不等式约束的优化问题时具有较好的性能, 其求解精确度和稳定性较文中列出的算法也有所提高.

表1 不同方法求得问题1最优解的比较

Table 1 Comparison of the best solution for Example 1 found by different methods

变量	TLBO <sup>[12]</sup>	CDE <sup>[18]</sup>	UABC <sup>[24]</sup>	ITLBO <sup>[25]</sup>	CTLBO
$x_1(h)$	0.205730	0.203137	0.205730	0.205730	0.205730
$x_2(l)$	3.470489	3.542998	3.470489	3.470489	3.470489
$x_3(t)$	9.036624	9.033498	9.036624	9.036624	9.036626
$x_4(b)$	0.205730	0.206179	0.205730	0.205730	0.205730
$g_1(x)$	-0.000001	-44.578568	-0.000028	-0.000000	-0.000002
$g_2(x)$	-0.000001	-44.663534	-0.000025	-0.000000	-0.009189
$g_3(x)$	-0.000000	-0.003042	-0.000000	-0.000000	-0.000000
$g_4(x)$	-3.432984	-3.423726	-3.432984	-3.432984	-3.432984
$g_5(x)$	-0.080730	-0.078137	-0.080730	-0.080730	-0.080730
$g_6(x)$	-0.235540	-0.235557	-0.235540	-0.235540	-0.235540
$g_7(x)$	-0.000000	-38.028268	-0.000050	-0.000000	-0.000004
$f(x)$	1.724852	1.733462	1.724852	1.724852	1.724852

表2 不同方法求得问题1结果统计表

Table 2 Statistical results of different methods for Example 1

方法	最优解	均值	最差解	标准差
TLBO <sup>[12]</sup>	1.7248523	1.7248525	1.7248569	8.53E-07
CDE <sup>[8]</sup>	1.733461	1.768158	1.824105	2.2E-02
UABC <sup>[24]</sup>	1.724852	1.724853	NA	1.70E-06
ITLBO <sup>[25]</sup>	1.7248523	1.7248523	1.7248523	6.77E-16
COMDE <sup>[26]</sup>	1.7248523	1.7248523	1.7248523	1.60E-12
CTLBO	1.724852	1.724852	1.724853	0.00E-00

表3 不同方法求得问题2最优解的比较

Table 3 Comparison of the best solution for Example 2 found by different methods

变量	TLBO <sup>[12]</sup>	CDE <sup>[18]</sup>	UABC <sup>[24]</sup>	ITLBO <sup>[25]</sup>	CTLBO
$x_1(T_s)$	0.8125	0.812500	0.8125	0.8125	0.812500
$x_2(T_h)$	0.4375	0.437500	0.4375	0.4375	0.437500
$x_3(R)$	42.098446	42.09841	42.098446	42.098446	42.098400
$x_4(L)$	176.636596	176.63769	176.636596	176.636596	176.636596
$g_1(x)$	-0.000000	-6.677E-7	-0.000000	-0.000000	-0.000000
$g_2(x)$	-0.035881	-0.035881	-0.035881	-0.035881	-0.035880
$g_3(x)$	-0.000000	-3.6831	-0.000000	-0.000000	-7.0E-10
$g_4(x)$	-63.363404	-63.3623	-63.363404	-63.363404	-63.363404
$f(x)$	6 059.714335	6 059.7340	6 059.714335	6 059.714335	6 059.714335

表4 不同方法求得问题2结果统计表

Table 4 Statistical results of different methods for Example 2

方法	最优解	均值	最差解	标准差
TLBO <sup>[12]</sup>	6 059.714335	6 059.714335	6 059.714335	1.85E-12
CDE <sup>[18]</sup>	6 059.7340	6 085.2303	6 371.0455	4.3E+02
UABC <sup>[24]</sup>	6 059.714335	6 192.116211	NA	2.04E+02
ITLBO <sup>[25]</sup>	6 059.714335	6 059.714335	6 059.714335	1.85E-12
COMDE <sup>[26]</sup>	6 059.714335	6 059.714335	6 059.714335	3.62E-10
CTLBO	6 059.714335	6 059.714335	6 059.714335	0.00E-00

表 5 不同方法求得问题 3 最优解的比较

Table 5 Comparison of the best solution for Example 3 found by different methods

变量	TLBO <sup>[12]</sup>	ETLBO <sup>[13]</sup>	FETLBO <sup>[15]</sup>	UABC <sup>[24]</sup>	ITLBO <sup>[25]</sup>	CTLBO
$x_1(d)$	0.051506	0.051565	0.051691	0.051691	0.051698	0.051664
$x_2(D)$	0.35327	0.353713	0.356758	0.356769	0.356723	0.356112
$x_3(P)$	11.555900	11.468954	11.286578	11.285988	11.288662	11.313513
$g_1(x)$	-0.000388	-0.001029	0.000953	-0.000000	-0.000000	-1.50E-08
$g_2(x)$	-0.000020	-0.000062	-0.000014	-0.000000	-0.000000	-5.60E-08
$g_3(x)$	-4.042961	-4.047205	-4.053903	-4.053886	-4.053796	-4.057519
$g_4(x)$	-0.730778	-0.729815	-0.727701	-0.727694	-0.727725	-0.728149
$f(x)$	0.012671	0.0126674	0.0126652	0.012665	0.012665	0.0126547

表 6 不同方法求得问题 3 结果统计表

Table 6 Statistical results of different methods for Example 3

方法	最优解	均值	最差解	标准差
TLBO <sup>[12]</sup>	0.0126717	0.0127407	0.0127977	2.88E-05
ETLBO <sup>[13]</sup>	0.0126674	NA	NA	NA
FETLBO <sup>[15]</sup>	0.0126652	NA	NA	NA
CDE <sup>[18]</sup>	0.0126702	0.012703	0.012790	2.7E-05
UABC <sup>[24]</sup>	0.012665	0.012683	NA	3.31E-05
ITLBO <sup>[25]</sup>	0.0126652	0.0126662	0.0126735	2.12E-06
CTLBO	0.0126547	0.01265493	0.01265693	5.50E-07

表 7 不同方法求得问题  $g05, g06, g11$  和  $g13$  最优解的比较

Table 7 Comparison of the best solution for example  $g05, g06, g11$  and  $g13$  found by different methods

函数/最优值	统计项	方法				
		TLBO <sup>[12]</sup>	ETLBO <sup>[13]</sup>	HTS <sup>[27]</sup>	Jaya <sup>[28]</sup>	CTLBO
$g05/5\ 126.498$	最好解	5 126.486	5 126.484	5 126.486	5 126.486	5 126.517
	均值	5 126.6184	5 168.7149	5 126.6831	5 126.635	5 126.605
	最差解	5 127.714	5 261.826	5 126.5152	5 126.5061	5 126.759
$g06/-6\ 961.814$	最好解	-6 961.814	-6 961.814	-6 961.814	-6 961.814	-6 961.814
	均值	-6 961.814	-6 961.814	-6 961.814	-6 961.814	-6 961.814
	最差解	-6 961.814	-6 961.814	-6 961.814	-6 961.814	-6 961.814
$g11/0.750$	最好解	0.7499	0.750	0.7499	0.7499	0.750
	均值	0.7499	0.750	0.7499	0.7499	0.750
	最差解	0.7499	0.750	0.7499	0.7499	0.750
$g13/0.0539498$	最好解	0.44015	0.13314	0.37319	0.003625	0.053991
	均值	0.69055	0.83851	0.79751	0.003631	0.058190
	最差解	0.95605	0.99979	0.66948	0.003627	0.069077

References

1 Chen Bao-Lin. *Optimization Theory and Algorithm*. Beijing: Tsinghua University Press, 1989.  
(陈宝林. 最优化理论与算法. 北京: 清华大学出版社, 1989.)

2 Das S, Mullick S S, Suganthan P N. Recent advances in differential evolution-An updated survey. *Swarm & Evolutionary Computation*, 2016, **27**: 1-30

3 Duan H B, Li P, Yu Y X. A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory. *IEEE/CAA Journal of Automatica Sinica*, 2015, **2**(1): 11-18

4 Pan Feng, Chen Jie, Gan Ming-Gang, Cai Tao, Tu Xu-Yan. Model analysis of particle swarm optimizer. *Acta Automatica Sinica*, 2006, **32**(3): 368-377  
(潘峰, 陈杰, 甘明刚, 蔡涛, 涂序彦. 粒子群优化算法模型分析. 自动化学报, 2006, **32**(3): 368-377)

5 Jin Xin-Lei, Ma Long-Hua, Wu Tie-Jun, Qian Ji-Xin. Convergence analysis of the particle swarm optimization based on stochastic processes. *Acta Automatica Sinica*, 2007,

- 33(12): 1263–1268  
(金欣磊, 马龙华, 吴铁军, 钱积新. 基于随机过程的 PSO 收敛性分析. 自动化学报, 2007, 33(12): 1263–1268)
- 6 Long Q, Wu C Z. A hybrid method combining genetic algorithm and Hooke-Jeeves method for constrained global optimization. *Journal of Industrial & Management Optimization*, 2017, 10(4): 1279–1296
  - 7 Prakasam A, Savarimuthu N. Metaheuristic algorithms and probabilistic behaviour: a comprehensive analysis of Ant Colony Optimization and its variants. *Artificial Intelligence Review*, 2016, 45(1): 97–130
  - 8 Ma L B, Zhu Y L, Zhang D Y, Niu B. A hybrid approach to artificial bee colony algorithm. *Neural Computing and Applications*, 2016, 27(2): 387–409
  - 9 Coello C A C, Carlos A. Constraint-handling Techniques Used with Evolutionary Algorithms. In: Proceedings of the 10th Annual Conference Companion on Genetic and Evolutionary Computation. Atlanta, GA, USA: ACM, 2010. 2445–2466
  - 10 Wu G H, Pedrycz W, Suganthan P N, Mallipeddi P. A variable reduction strategy for evolutionary algorithms handling equality constraints. *Applied Soft Computing*, 2015, 37: 774–786
  - 11 Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 2011, 43(3): 303–315
  - 12 Venkata Rao R. Teaching-learning-based optimization and its engineering applications. *Teaching Learning Based Optimization Algorithm: and Its Engineering Applications*. Berlin, Heidelberg: Springer Publishing Company, 2015.
  - 13 Venkata Rao R, Patel V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations*, 2012, 3(4): 535–560
  - 14 Patel V K. *Teaching Learning Based Optimization Algorithm*. Berlin, Heidelberg: Springer International Publishing, 2016.
  - 15 Yu Kun-Jie, Wang Xin, Wang Zhen-Lei. Elitist teaching-learning-based optimization algorithm based on feedback. *Acta Automatica Sinica*, 2014, 40(9): 1976–1983  
(于坤杰, 王昕, 王振雷. 基于反馈的精英教学优化算法. 自动化学报, 2014, 40(9): 1976–1983)
  - 16 Coello C A C. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 2000, 41(2): 113–127
  - 17 He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 2007, 20(1): 89–99
  - 18 Huang F Z, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics & Computation*, 2007, 186(1): 340–356
  - 19 Richardson J T, Palmer M R, Liepins G E, Hilliard M. Some guidelines for genetic algorithms with penalty functions. In: Proceedings of the 3rd International Conference on Genetic Algorithms. George Mason University, USA: Morgan Kaufmann Publishers Inc., 1989. 191–197
  - 20 Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 2000, 4(3): 284–294
  - 21 Kannan B K, Kramer S N. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, 1994, 116(2): 405–411
  - 22 Arora J S. *Introduction to Optimal Design* New York: McGraw-Hill, 1989. 413–432
  - 23 Belegundu A D, Arora J S. A study of mathematical programming methods for structural optimization. Part I: theory. *International Journal for Numerical Methods in Engineering*, 1985, 21(9): 1583–1599
  - 24 Brajevic I, Tuba M. An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems. *Journal of Intelligent Manufacturing*, 2013, 24(4): 729–740
  - 25 Yu K J, Wang X, Wang Z L. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. *Journal of Intelligent Manufacturing*, 2016, 27(4): 831–843
  - 26 Mohamed A W, Sabry H Z. Constrained optimization based on modified differential evolution algorithm. *Information Sciences*, 2012, 194: 171–208
  - 27 Patel V K, Savsani V J. Heat transfer search (HTS): a novel optimization algorithm. *Information Sciences*, 2015, 324: 217–246
  - 28 Venkata R R. Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 2016, 7(1): 19–34



刘三阳 西安电子科技大学教授. 主要研究方向为应用数学, 最优化, 运筹学.  
E-mail: liusanyang@126.com  
(LIU San-Yang Professor at Xidian University. His research interest covers applied mathematics, especially optimization, and operations research.)



靳安钊 西安电子科技大学硕士研究生. 主要研究方向为最优化算法与应用. 本文通信作者.  
E-mail: jinanzhao@stu.xidian.edu.cn  
(JIN An-Zhao Master student at Xidian University. His research interest covers optimization algorithms and their application. Corresponding author of this paper.)