

# 控制与决策

Control and Decision

## 求解电力系统经济调度问题的改进粒子群优化算法

梁静, 葛士磊, 瞿博阳, 于坤杰

引用本文:

梁静, 葛士磊, 瞿博阳, 等. 求解电力系统经济调度问题的改进粒子群优化算法[J]. *控制与决策*, 2020, 35(8): 1813–1822.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.1490>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### [基于改进邻域搜索策略的人工蜂群算法](#)

Artificial bee colony algorithm based on improved neighborhood search strategy

*控制与决策*. 2019, 34(5): 965–972 <https://doi.org/10.13195/j.kzyjc.2017.1506>

### [基于拓扑结构与粒子变异改进的粒子群优化算法](#)

Modified particle swarm optimization algorithms based on topology and particle mutation

*控制与决策*. 2019, 34(2): 419–428 <https://doi.org/10.13195/j.kzyjc.2017.1063>

### [基于双鱼群算法的电力系统无功优化](#)

Reactive power optimization of power system based on double fish–swarm algorithm

*控制与决策*. 2018, 33(10): 1886–1892 <https://doi.org/10.13195/j.kzyjc.2017.0552>

### [基于种群特征反馈的布谷鸟搜索算法](#)

Cuckoo search algorithm based on swarm feature feedback

*控制与决策*. 2016, 31(6): 969–975 <https://doi.org/10.13195/j.kzyjc.2015.0829>

### [基于DEPSO的模糊时间ZW多产品厂间歇调度](#)

A study of the DEPSO–based multiproduct plants batch scheduling under uncertainty with zero wait

*控制与决策*. 2015, 30(12): 2275–2279 <https://doi.org/10.13195/j.kzyjc.2014.1735>

# 求解电力系统经济调度问题的改进粒子群优化算法

梁 静<sup>1</sup>, 葛士磊<sup>2</sup>, 瞿博阳<sup>3</sup>, 于坤杰<sup>1†</sup>

(1. 郑州大学 电气工程学院, 郑州 450001; 2. 郑州大学 产业技术研究院, 郑州 450001;  
3. 中原工学院 电子信息学院, 郑州 450007)

**摘要:** 电力系统经济调度问题是电力系统中的一个重要的研究课题, 针对该问题, 提出一种改进粒子群优化 (ODPSO) 算法. 改进算法在搜索前期, 采用广义的反向学习策略, 使算法能够快速靠近较优的搜索区域, 从而提高收敛速度; 在搜索后期, 借鉴差分进化算法的进化机制设计改进的变异和交叉策略, 对当前种群的最优粒子进行更新, 从而提高种群的多样性, 进而协助算法获得全局最优解. 为了验证改进粒子群优化算法的有效性, 对 CEC2006 提出的 22 个基准约束测试函数进行仿真, 结果表明改进算法相比其他算法在寻优精度和稳定性上更具优势. 最后, 将改进算法应用于考虑机组爬坡速率约束、机组禁行区域约束以及电力平衡约束的两个电力系统经济调度问题, 取得了令人满意的结果.

**关键词:** 粒子群优化算法; 函数优化; 约束处理; 反向学习; 基准测试函数; 电力系统经济调度

中图分类号: TP18; TM734

文献标志码: A

## Improved particle swarm optimization algorithm for solving power system economic dispatch problem

LIANG Jing<sup>1</sup>, GE Shi-lei<sup>2</sup>, QU Bo-yang<sup>3</sup>, YU Kun-jie<sup>1†</sup>

(1. School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China; 2. Industrial Technology Research Institute, Zhengzhou University, Zhengzhou 450001, China; 3. School of Electric & Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China)

**Abstract:** The power system economic dispatch problem is an important research topic in power systems. To solve this problem, an improved particle swarm optimization (ODPSO) algorithm is proposed. In the early stage of the improved algorithm, the generalized opposition-based learning strategy is used to make the algorithm quickly close to the potential search area and improve the convergence speed. In the later stage of searching, inspired by evolutionary process of differential evolution, an improved mutation and crossover strategy is developed to update the optimal particle of the current population, thus improving the population diversity and assisting the algorithm to obtain global optimal solution. In order to validate the effectiveness of the improved algorithm, 22 constraint test functions presented in CEC2006 are simulated. Experimental results show that the improved algorithm is superior to other compared algorithms in terms of the accuracy and stability. Finally, the improved algorithm is applied to two economic dispatch problems of power systems, which takes into account the ramp rate limits of the generating units, prohibited operating zones and power balance constraint, and satisfying results are obtained.

**Keywords:** particle swarm optimization algorithm; function optimization; constraint handling; opposition-based learning; benchmark functions; power system economic dispatch problem

## 0 引言

经济调度问题是电力系统中较为重要的数学优化问题, 其目的是在满足电力系统运行约束的前提下最小化总发电成本, 这对电力系统的经济可靠运行具有重要意义<sup>[1-2]</sup>. 电力系统经济调度问题有多个约束

条件, 如电力平衡约束、运行约束、斜坡速率限制与禁止运行空间, 这些约束条件使得该问题的可行域空间变得十分复杂.

许多传统方法, 如 Lagrangian 法、投影梯度法和内点惩罚函数法等, 由于过于依赖初始值和梯度信

收稿日期: 2018-10-31; 修回日期: 2019-02-14.

基金项目: 国家自然科学基金项目 (61806179, 61876169, 61473266, 61673404); 中国博士后科学基金项目 (2017M622373).

责任编辑: 巩敦卫.

†通讯作者. E-mail: yukunjie@zzu.edu.cn.

息,已不适用于解决非线性且约束条件众多的优化问题.考虑到传统方法的一些缺陷,越来越多的学者采用群智能优化算法解决这一类问题.群智能优化算法在求解非凸、不连续和不可导的非线性优化问题时具有很强的优势,因此在处理可行域存在同样特征的电力系统经济调度问题时得到了广泛的应用<sup>[3]</sup>.如郑晓菁<sup>[4]</sup>使用人工蜂群优化算法解决多区域电力系统经济调度问题;吴亮红等<sup>[5]</sup>提出快速自适应差分进化算法解决复杂电力系统经济调度问题;王凌等<sup>[6]</sup>提出一种基于混合双种群的差分进化算法解决电力系统经济负荷分配;Adarsh等<sup>[7]</sup>提出一种混沌蝙蝠优化算法解决电力系统经济调度问题;针对含阀点加载效应和多燃料选择的经济调度问题,Modiri-Delshad等<sup>[8]</sup>提出一种回溯搜索算法解决此类问题;Zou等<sup>[9]</sup>提出一种改进的差分进化算法用于解决考虑阀点加载效应和不考虑阀点加载效应的经济负载调度问题;Yousefi<sup>[10]</sup>提出一种改进的粒子群优化算法,用于解决经济负荷分配问题.在这些群智能优化算法中,由于粒子群优化算法(particle swarm optimization, PSO)<sup>[11]</sup>具有编程简单、直观易实现等优点,在工程上得到了广泛的应用.然而,PSO算法在处理电力系统经济调度这种非凸、不连续的复杂约束优化问题时,易陷入局部最优.因此,许多学者提出一些改进的PSO算法解决电力系统经济调度问题.文献[12]提出一种改进的正交设计PSO算法解决单区域和多区域的经济负载调度问题,实验结果表明该算法能较有效地解决非凸、不连续的经济负载调度问题;文献[13]提出一种基于自适应社会学习因子的PSO算法,与基本的PSO算法相比,此算法对于IEEE 5、IEEE 14和IEEE 30总线系统,收敛速度更快,结果更准确;文献[14]将高斯随机变量引入PSO算法中的速度更新公式中,在4个不同的非凸经济调度问题中取得了良好的效果.虽然以上的研究工作取得了较令人满意的结果,但算法在寻优精度和稳定性上的表现还有待提高,尤其算法在搜索后期容易陷入局部最优的问题有待解决.

本文提出一种改进的粒子群优化算法,将广义反向学习与差分进化的改进变异和交叉机制混合到粒子群优化算法中,并利用可行性规则的约束处理方法求解约束优化问题.其中:广义反向学习机制用于算法的搜索前期,旨在加快算法逼近较好的搜索区域,提高算法的收敛速度,并为后期的局部精细搜索提供条件;差分进化算法的改进变异和交叉策略用于搜索后期对当前种群的最优粒子位置进行更新,以增

强种群的多样性,提高算法在较好搜索区域的精细搜索能力,最终提高算法的寻优精度.改进策略的引入不但能够提高粒子群优化算法的寻优速度,还可以增强算法的局部精细化搜索,进而改善寻优精度和稳定性.为验证改进算法的性能,选用CEC2006中的22个约束测试函数进行测试,并与其他算法进行对比.结果表明,改进算法能够较好地解决约束优化问题,并且在寻优精度和稳定性方面具有较强的竞争力.最后将改进算法用于求解电力系统的经济调度问题,2个经典案例的仿真结果表明所提算法相对于其他算法能够在满足一系列约束条件的情况下获得较小的发电成本.

## 1 经济调度问题数学模型

### 1.1 目标函数

文献[15]详细叙述了经济调度问题的目标函数与约束条件,经济调度问题的主要目的是在考虑所有机组与系统的等式和不等式约束条件的基础上,使所花费的发电燃料成本最少.每组机组单元简化的燃料成本如 $F_i(P_i)$ 和 $n$ 组机组的总燃料成本 $F_t$ 如下所示:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2, \quad (1)$$

$$F_t = \sum_{i=1}^n F_i(P_i). \quad (2)$$

其中: $F_i$ 是第 $i$ 组机组单元的燃料成本, $a_i$ 、 $b_i$ 与 $c_i$ 对应的是第 $i$ 组单元的燃料成本系数, $P_i$ 是第 $i$ 组发电机产生的电能, $n$ 是发电机组的总个数.

### 1.2 约束条件

经济调度问题的约束条件包括电力平衡约束、运行约束、斜坡速率限制与禁止运行空间.电力平衡约束是指为了平衡电力,电力系统的总发电量应等于整个系统负载与传输损失之和,它可以用以下公式表示:

$$\sum_{i=1}^n P_i = P_{\text{load}} + P_{\text{loss}}. \quad (3)$$

其中: $P_{\text{load}}$ 是整个系统的负载; $P_{\text{loss}}$ 是线上的传输损失, $P_{\text{loss}}$ 用B系数法计算,可表示为

$$P_{\text{loss}} = \sum_{i=1}^n \sum_{j=1}^n P_i B_{ij} P_j + \sum_{i=1}^n B_{0i} P_i + B_{00}. \quad (4)$$

每个发电机组输出功率必须维持在系统稳定运行要求的范围之内,即

$$P_{i,\min} \leq P_i \leq P_{i,\max}, \quad (5)$$

其中 $P_{i,\max}$ 、 $P_{i,\min}$ ( $i = 1, 2, \dots, n$ )分别为第 $i$ 组发电机组的上下限.爬坡速率约束将发电机组的输出

功率限制到一定范围之内. 因此, 对于第  $i$  组机组, 它的输出功率应满足以下条件:

$$\begin{cases} P_i - P_i^0 \leq \text{UR}_i, \\ P_i^0 - P_i \leq \text{DR}_i, \end{cases} \quad (6)$$

其中  $\text{UR}_i$  与  $\text{DR}_i$  分别是上坡速率限制值与下坡速率限制值. 在实际运行中, 由于发电机的蒸汽阀和轴承振动等一些物理构造上的限制, 发电机的有功出力范围是不连续的, 如下所示:

$$P_i \in \begin{cases} P_{i,\min} \leq P_i \leq P_{i,1}^L, \\ P_{i,k-1}^U \leq P_i \leq P_{i,k}^L, \\ P_{i,\text{pz}_i}^U \leq P_i \leq P_{i,\max}, \end{cases} \quad (7)$$

$i = 1, 2, \dots, \text{npz}, k = 2, 3, \dots, \text{pz}_i.$

其中:  $P_{i,k}^L$  和  $P_{i,k}^U$  分别是第  $i$  组机组的第  $k$  个禁行区域的下限和上限,  $\text{npz}$  是所有机组的禁行区间总个数,  $\text{pz}_i$  是第  $i$  组机组的禁行区间个数.

## 2 粒子群优化算法

粒子群优化算法的主要思想是: 首先, 随机初始化种群规模为  $N$ , 维度为  $D$  的粒子群, 在第  $k$  次迭代时, 第  $i$  个粒子的位置为  $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ , 该粒子通过向它的个体极值和全局极值学习来更新它的位置. 然后, 通过“适者生存”法则更新粒子的个体极值和全局极值. 最后, 判断算法是否达到停止条件. 若已达到, 则输出最优解; 否则继续迭代, 更新粒子速度和位置. 通过不断迭代更新, 最终找到最优解. 粒子的更新过程主要依据如下两个公式:

$$v_{i,d}^{k+1} = wv_{i,d}^k + c_1r_1(\text{pbest}_{i,d}^k - x_{i,d}^k) + c_2r_2(\text{gbest}_d^k - x_{i,d}^k), \quad (8)$$

$$x_{i,d}^{k+1} = x_{i,d}^k + v_{i,d}^{k+1}. \quad (9)$$

其中: 式(8)是粒子的速度更新公式;  $w$  是惯性权重, 一般取  $[0.4, 0.9]$  之间的数, 使其依据自身速度进行惯性运动;  $c_1$  和  $c_2$  是学习因子, 表示粒子受自身及全局的影响程度, 调节向个体最优和全局最优方向飞行的最大移动步长;  $r_1$  和  $r_2$  是  $0$  到  $1$  之间的随机数;  $v_{i,d}^k$  是第  $k$  次迭代的第  $i$  个粒子第  $d$  维的速度,  $d = 1, 2, \dots, D$ ; 式(9)是粒子的位置更新公式.

### 2.1 反向学习

反向学习是同时考虑当前位置的解及其相反位置的解, 从而能更快地逼近全局最优解的一种学习策略. 研究表明, 相比于随机产生的一个解, 反向解更有可能接近全局最优解<sup>[16]</sup>. 定义  $\mathbf{p} = (p_1, \dots, p_D)$  为  $D$  维空间的一个点,  $p_d \in [p_d^{\min}, p_d^{\max}]$ , 其中  $p_d^{\min}$ 、 $p_d^{\max}$  分

别为点  $\mathbf{p}$  第  $d$  维的最小边界值和最大边界值. 因此, 在  $D$  维空间中对应的相反的点可用如下公式定义:

$$\hat{p}_d = p_d^{\min} + p_d^{\max} - p_d, \quad d = 1, 2, \dots, D. \quad (10)$$

文献[17]提出一种广义的反向学习策略, 对于点  $\mathbf{p}$ , 通过广义反向学习得到的相反的点可用如下公式表示:

$$\tilde{p}_d = k(p_d^{\min} + p_d^{\max}) - p_d, \quad d = 1, 2, \dots, D. \quad (11)$$

其中  $k$  是  $[0, 1]$  之间的随机数.

在改进 PSO 算法中, 若算法处于搜索前期, 则对于当前种群的最优个体, 按如下公式实施广义的反向学习策略:

$$\mathbf{X}_d^{\text{new}} = k(x_d^{\min} + x_d^{\max}) - \text{gbest}_d, \quad d = 1, 2, \dots, D. \quad (12)$$

其中:  $k$  是  $[0, 1]$  之间的随机数;  $x_d^{\min}$  与  $x_d^{\max}$  分别是粒子第  $d$  维的最小、最大边界值;  $\mathbf{X}^{\text{new}}$  是当前种群最优个体  $\text{gbest}$  通过广义的反向学习策略产生的新解,  $\mathbf{X}^{\text{new}}$  与  $\text{gbest}$  通过“适者生存”的竞争法则选择较优个体, 作为当前种群的最优个体. 在改进 PSO 算法中实施广义反向学习策略后, 若粒子的位置超出范围, 本文中的做法是按如下公式重新初始化粒子的位置:

$$x_{i,d} = x_d^{\min} + \text{rand}(x_d^{\max} - x_d^{\min}). \quad (13)$$

### 2.2 变异和交叉策略

差分进化 (differential evolution, DE) 算法是一种基于群体演化的优化算法, 变异和交叉操作是该算法的关键步骤. 由于 DE 算法的全局优化性能以及数值稳定性都非常出色, 得到了广泛的应用. 许多学者提出将 PSO 与 DE 相混合, 以改进 PSO 算法. 文献[18]提出了一种混合方式, 即 PSO 算法先作用于整个种群, 然后由 DE 算法对所有粒子的个体最优位置进行改进, 其 DE 和 PSO 母体算法分别是 DE/best/2/bin 和基于 Gbest 模型的 PSO 算法. 主要操作包括变异和交叉. 变异可以表示为

$$z_{i,d} = \text{gbest}_d + F(\text{pbest}_{m_1,d} - \text{pbest}_{m_2,d} + \text{pbest}_{m_3,d} - \text{pbest}_{m_4,d}). \quad (14)$$

其中:  $\mathbf{z}_i$  是变异向量;  $F$  是缩放因子;  $m_1 \sim m_4$  是从  $1$  至种群规模  $N$  中随机选取的整数, 且  $m_1 \neq m_2 \neq m_3 \neq m_4$ .

交叉可以表示为

$$u_{i,d} = \begin{cases} z_{i,d}, & \text{if } \text{rand} \leq C_R \text{ or } d = \text{rn}_i; \\ \text{pbest}_{i,d}, & \text{otherwise.} \end{cases} \quad (15)$$

其中:  $u_{i,d}$  是试探向量  $\mathbf{u}_i$  的第  $d$  维,  $C_R$  是交叉概率,  $\text{rn}_i$  是从  $1$  至  $D$  中随机选取的整数. 在完成变异、交叉

过程之后,通过选择操作,按照适应度值选择较优的解.受到此文献的启发,本文借鉴差分进化算法的进化机制,设计改进的变异和交叉策略对当前种群的最优粒子进行更新,将当前种群最佳个体  $gbest$  作为基向量,随机选择两个不同的个体极值作为差分向量,变异过程可表示为

$$z_{i,d} = gbest_d + F(pbest_{m_1,d} - pbest_{m_2,d}). \quad (16)$$

在交叉阶段,按如下公式进行交叉操作:

$$u_{i,d} = \begin{cases} z_{i,d}, & \text{if } \text{rand} \leq C_R \text{ or } d = r_{m_i}; \\ gbest_d, & \text{otherwise.} \end{cases} \quad (17)$$

在满足交叉条件时,试探向量的第  $d$  维等于变异向量的第  $d$  维;不满足条件时,试探向量的第  $d$  维等于当前最优个体  $gbest$  的第  $d$  维.在处理无约束优化问题的最小化问题时,选择过程仍是选择最小的适应度值,但处理带约束条件的优化问题时,就需要约束处理机制.

### 2.3 约束处理机制

在处理约束优化问题时,当前最流行的约束处理方法分为4大类<sup>[19]</sup>: 1) 惩罚函数法; 2) 可行性规则; 3) 多目标优化方法; 4) 混合方法. 惩罚函数法通过惩罚因子将目标函数和约束条件相结合,构造惩罚函数,从而将约束优化问题转化为无约束优化问题进行求解.然而,惩罚因子的选取对结果影响较大,很难得到一个最合适的惩罚因子.可行性规则方法具有简单有效且易于实现的优点,得到了较广泛的应用.在解决最小化问题时,可行性规则主要思想可以描述为:在比较两个可行解时,优先选择目标函数值小的解;在比较两个不可行解时,优先选择约束违反度小的解;在比较可行解与不可行解时,优先选择可行解.多目标优化方法将约束优化问题转化为双目标优化问题或者多目标优化问题,从而对问题进行求解.另外,受到“无免费午餐”理论的启发,许多学者提出将一些约束处理方法相混合的方法,如文献[20]提出一种新的动态权重选择策略,且这种策略结合了可行性规则;文献[21]提出将多目标优化技术与惩罚函数相结合的方法解决约束优化问题.这些方法在处理约束优化问题时也得到了不错的效果.

考虑到所选择的约束处理方法的简便性,本文选用可行性规则方法执行解之间的比较选择过程,如文献[22]所描述的可行性规则方法.

本文中,处理可行解和不可行解的方法如下:

1) 在解的比较过程中,若两解均是可行解,则比较目标函数值的大小,此准则主要针对目标,使搜索

方向指向目标值好的区域.

2) 若一个为可行解,另一个为不可行解,则选择可行解,此准则旨在使搜索方向指向可行域.

3) 若两个解均为不可行解,则比较两个解的约束违反度值,选择约束违反度较小的解,此准则是为了使搜索方向向可行域方向靠近.

### 2.4 改进的粒子群优化算法步骤

本文提出的改进粒子群优化算法的主要思想如下:在PSO进行个体更新与群体最优个体更新之后,在搜索前期进行广义反向学习策略,后期采用差分进化的改进变异和交叉策略.

在本文中,定义当前函数评价次数为CFE,最大函数评价次数为FFEmax.当  $1 - CFE/FFEmax \geq Q$  时,为算法搜索前期,对当前种群最优个体实施广义的反向学习策略;否则,为算法搜索后期,对当前种群最优个体进行改进的变异、交叉与选择操作.其中,  $Q$  为算法的分割点,为了设置一个合理的分割点,在文中的实验部分会有详细阐述.改进的粒子群优化算法流程如图1所示.

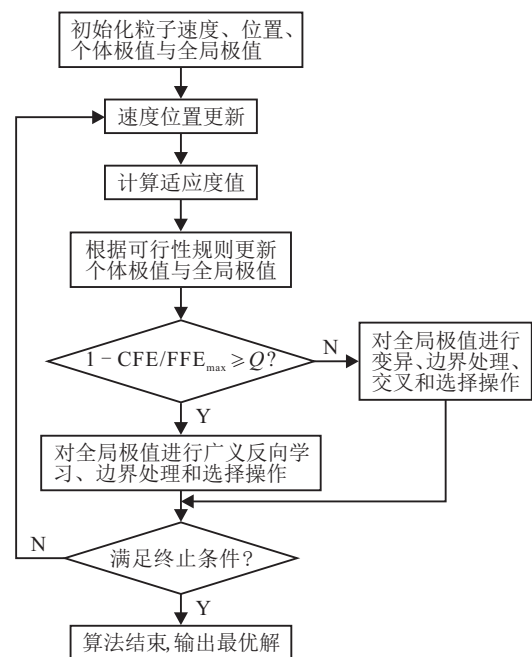


图1 改进粒子群优化算法流程

改进粒子群优化算法的详细步骤如下.

step 1: 初始化粒子速度与位置,对每个粒子进行评估,确定种群的个体极值和全局极值.

step 2: 分别按式(8)和(9)更新每一个粒子的速度和位置.

step 3: 对超出边界的粒子按式(13)进行边界处理.

step 4: 利用可行性规则更新个体极值与全局极

值.

step 5: 若是搜索前期, 则对当前最优个体实施广义的反向学习策略, 否则实施变异、交叉和选择操作.

step 6: 判断是否达到终止条件, 若已达到终止条件, 则算法结束, 输出最优解, 否则返回 step 2 继续迭代循环.

### 3 数值仿真及分析

#### 3.1 分割点对改进粒子群算法的性能影响

为了设置一个相对合理的分割点, 同时验证不同分割点对算法的性能影响, 针对 CEC2006 的 22 个测试函数采用不同分割点  $Q = (0, 0.3, 0.5, 0.7, 0.9, 1)$  各运行 25 次 (分割点为  $Q = 0$  代表算法只调用广义反向学习策略, 分割点为  $Q = 1$  代表算法只调用改进的差分操作), 不同分割点对 CEC2006 测试函数运行 25 次得到的均值结果如表 1 所示. 根据表 1 中各分割点对 CEC2006 的 22 个测试函数的均值, 用 Friedman 检验统计<sup>[23]</sup> 方法对不同分割点的性能进行排序, 结

果如图 2 所示. 通过图 2 可以看出, 分割点取不同值时对算法的性能有较大影响, 分割点取较小值时, 广义反向学习策略被调用的次数较多, 但算法的综合性能差; 当分割点逐渐取较大值时, 改进的差分操作策略被调用的次数逐渐增多, 算法的综合性能在总体上是逐渐递增的. 若分割点设置为  $Q = 1$ , 即舍去广义反向学习策略时, 算法的综合性能反而有稍微降低; 而当分割点设置为  $Q = 0.9$  时, 算法性能相对其他分割点综合性能最优, 故在本文中分割点设置为  $Q = 0.9$ .

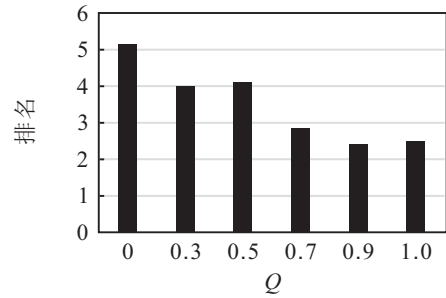


图 2 不同分割点在 CEC2006 上的 Friedman 检验排名

表 1 ODPSO 不同分割点对 CEC2006 测试结果对比

函数	Q = 0	Q = 0.3	Q = 0.5	Q = 0.7	Q = 0.9	Q = 1
g01	-15	-14.9998	-15	-15	-15	-15
g02	-0.65107	-0.6974	-0.69871	-0.68057	-0.74048	-0.75538
g03	-0.71857	-0.99831	-0.65793	-0.99954	-0.99714	-0.99749
g04	-30665.5	-30665.5	-30665.5	-30665.5	-30665.5	-30665.5
g05	5496.258	5313.921	5390.094	5175.245	5181.393	5176.443
g06	-6961.81	-6961.81	-6961.81	-6961.81	-6961.81	-6961.81
g07	27.15379	24.81824	25.00482	24.415	24.33488	24.31821
g08	-0.09583	-0.09583	-0.09583	-0.09583	-0.09583	-0.09583
g09	680.8764	680.68	680.642	680.63	680.6301	680.6301
g10	7592.999	7310.957	7304.999	7097.699	7053.512	7053.311
g11	0.7499	0.7499	0.7499	0.7499	0.7499	0.7499
g12	-1	-1	-1	-1	-1	-1
g13	1.517845	0.547815	0.608667	0.49986	0.340434	0.420729
g14	-43.6945	-46.3067	-45.2325	-47.3272	-47.7578	-47.3915
g15	963.8672	962.3466	963.7106	961.8903	961.7465	961.7189
g16	-1.90477	-1.90498	-1.90516	-1.9052	-1.90516	-1.90516
g17	8998.034	9046.472	8970.746	8941.768	8923.935	8945.863
g18	-0.78511	-0.82555	-0.79203	-0.79958	-0.82779	-0.85074
g19	39.36236	34.83525	35.78631	33.4349	33.21663	32.93995
g21	561.1299	228.2035	247.2843	235.8676	235.8598	288.9714
g23	251.0641	-60.1308	154.4853	-170.398	-311.738	-306.232
g24	-5.50801	-5.50801	-5.50801	-5.508	-5.50801	-5.50801

#### 3.2 ODPSO 性能测试

为了验证改进的粒子群优化算法在处理约束优化问题时的性能, 本文采用 CEC2006 中的 22 个标准测试函数与 PSO<sup>[11]</sup>、DE<sup>[24]</sup>、BLPSO<sup>[25]</sup>、ABC<sup>[26]</sup>、M-ABC<sup>[27]</sup>、ATMES<sup>[28]</sup> 进行比较. 为了比较的公平, 所选择算法的最大函数评价次数均设置为 240000.

改进的粒子群优化算法的参数设置为: 种群规模为 100, 惯性权重  $w$  随着迭代次数从 0.9 至 0.4 线性递减, 学习因子  $c_1$  与  $c_2$  均为 1.49445, 缩放因子  $F$  为 0.9, 交叉概率  $C_R$  为 0.9. 每个算法独立运行 25 次, 对结果进行统计分析, 表 2 给出了每个算法的均值和方差. 值得说明的是, 算法 PSO、ABC、ATMES、M-ABC、

DE的运行结果取自文献[29], BLPSO的运行结果由其源代码运行所获得. 表2中: “-”表示结果未获取,

“NF”表示在25次独立运行中未找到可行解, 加粗显示的为最优结果.

表2 ODPSO与其他算法对CEC2006测试函数结果对比

函数	标准	ODPSO	BLPSO	M-ABC	ATMES	PSO	DE	ABC
g01	均值	<b>-15</b>	-9.748 34	<b>-15</b>	<b>-15</b>	-14.71	-14.555	-15
	方差	3.63e-16	7.50e-01	<b>0.00e+00</b>	<b>0.00e+00</b>	-	-	-
g02	均值	-0.740 480 694	-0.292 729 3	<b>-0.799 336</b>	-0.787 6	-0.419 96	-0.665	-0.792 4
	方差	4.34e-02	2.74e-02	<b>6.84e-03</b>	0.018	-	-	-
g03	均值	-0.997 140 653	-0.174 627 3	<b>-1</b>	-0.999 9	0.764 813	<b>-1</b>	<b>-1</b>
	方差	1.29e-02	1.45e-01	<b>4.68e-05</b>	1.02e-04	-	-	-
g04	均值	<b>-30 665.538 67</b>	-30 654.967	<b>-30 665.538 7</b>	<b>-30 665.538 7</b>	<b>-30 665.539</b>	<b>-30 665.539</b>	<b>-30 665.539</b>
	方差	9.89e-06	7.78e+00	2.22e-11	<b>7.43e-12</b>	-	-	-
g05	均值	5 181.393 285	5 749.447 32	5 178.139	<b>5 127.732 1</b>	5 135.973	5 264.27	5 185.714
	方差	7.47e+01	4.27e+02	5.61e+01	<b>2.15e+00</b>	-	-	-
g06	均值	<b>-6 961.813 876</b>	-6 471.693 72	<b>-6 961.813 9</b>	<b>-6 961.813 9</b>	<b>-6 961.814</b>	-6 954.434	-6 961.813
	方差	<b>0.00e+00</b>	2.82e+02	<b>0.00e+00</b>	3.71e-12	-	-	-
g07	均值	24.334 879 41	155.794 5	24.415	<b>24.314 56</b>	32.407	24.31	24.473
	方差	5.86e-02	5.62e+01	1.24e-01	<b>1.42e-02</b>	-	-	-
g08	均值	<b>-0.095 825 041</b>	<b>-0.095 8</b>	<b>-0.095 825</b>	<b>-0.095 825</b>	<b>-0.095 825</b>	<b>-0.095 825</b>	<b>-0.095 825</b>
	方差	<b>5.67e-18</b>	2.94e-05	4.23e-17	6.12e-17	-	-	-
g09	均值	<b>680.630 057 4</b>	715.216 1	680.65	680.64	<b>680.63</b>	<b>680.63</b>	680.64
	方差	<b>9.27e-13</b>	1.53e+01	1.55e-02	1.26e-02	-	-	-
g10	均值	<b>7 053.512 497</b>	NF	7 233.882	7 277.47	7 205.5	7 147.334	7 224.407
	方差	<b>1.19e+01</b>	NF	1.10e+02	1.97e+02	-	-	-
g11	均值	<b>0.749 9</b>	<b>0.75</b>	<b>0.75</b>	<b>0.75</b>	<b>0.749</b>	0.901	<b>0.75</b>
	方差	<b>1.13e-16</b>	1.95e-03	2.30e-05	2.82e-04	-	-	-
g12	均值	<b>-1</b>	<b>-1</b>	<b>-1</b>	<b>-1</b>	-0.998 875	<b>-1</b>	<b>-1</b>
	方差	0.00e+00	6.53e-06	<b>0.00e+00</b>	<b>0.00e+00</b>	-	-	-
g13	均值	0.340 434 355	1.175 367	0.158 552	<b>0.053 959</b>	0.569 358	0.872	0.968
	方差	2.36e-01	1.30e+00	1.72e-01	<b>1.06e-05</b>	-	-	-
g14	均值	<b>-47.757 801 8</b>	-39.566 39	-47.271	-47.727 8	-40.871	-29.218 7	-40.107 1
	方差	<b>1.25e-02</b>	3.08e+00	2.46e-01	5.05e-02	2.29e+00	1.36e+01	7.14e+00
g15	均值	961.746 474 5	964.540 6	961.719	<b>961.715 3</b>	965.515 4	961.753 7	966.286 8
	方差	6.41e-02	3.69e+00	1.42e-02	<b>2.69e-04</b>	3.72e+00	1.22e-01	3.12e+00
g16	均值	<b>-1.905 155 259</b>	-1.785 031	<b>-1.905 155</b>	-1.902 816	<b>-1.905 2</b>	<b>-1.905 2</b>	<b>-1.905 2</b>
	方差	7.81e-16	5.39e-02	4.52e-16	8.41e-04	<b>2.34e-16</b>	<b>2.34e-16</b>	<b>2.34e-16</b>
g17	均值	8 923.934 753	8 964.631 87	8 987.459	<b>8 896.400 8</b>	8 899.472 1	8 932.044 4	8 941.924 5
	方差	7.26e+01	2.23e+02	9.57e+01	<b>3.27e+01</b>	3.79e+01	4.68e+01	4.26e+01
g18	均值	-0.827 792 699	-0.257 151 7	-0.795 018 7	-0.843 026	-0.827 6	-0.861 65	<b>-0.865 87</b>
	方差	7.80e-02	7.83e-02	9.39e-02	6.35e-02	1.11e-01	3.67e-03	<b>3.37e-04</b>
g19	均值	33.216 633 82	82.731 769 8	34.267	33.379 68	36.617 2	<b>32.768</b>	36.007 8
	方差	1.19e+00	2.27e+01	6.31e-01	3.52e-01	2.04e+00	<b>6.28e-02</b>	1.83e+00
g21	均值	<b>235.859 785 5</b>	NF	306.609	NF	345.659 5	366.919 3	275.543 6
	方差	6.27e+01	NF	<b>1.98e+01</b>	NF	6.63e+01	9.13e+01	6.05e+01
g23	均值	<b>-311.737 659</b>	NF	-35.272	NF	-25.917 9	-7.264 2	-4.325 4
	方差	1.21e+02	NF	8.28e+01	NF	4.30e+01	2.30e+01	<b>1.37e+01</b>
g24	均值	<b>-5.508 013 272</b>	-5.502 11	<b>-5.508 013</b>	<b>-5.508 013</b>	<b>-5.508</b>	<b>-5.508</b>	<b>-5.508</b>
	方差	9.06e-16	3.92e-03	2.71e-15	<b>0.00e+00</b>	9.36e-16	9.36e-16	9.36e-16

由表2中的数据可知,改进的粒子群优化算法ODPSO在13个测试函数(g01、g04、g06、g08、g09、g10、g11、g12、g14、g16、g21、g23和g24)上获得了最优的结果,比较明显地优于PSO、DE、ABC、ATMES、M-ABC和BLPSO这些对比算法.对于测试函数g08、g09、g10、g11和g14,ODPSO不但获得了最优的平均值,且25次运行结果的方差同样是最小的.在大部分测试函数上的优秀表现说明了ODPSO在处理约束优化问题上的精度和稳定性更具优势.根据表2中对比算法的平均值,用Friedman检验统计方法对不同算法的性能进行排序,结果如图3所示.从图3可以看出,ODPSO在7种算法中取得了排名第1的结果,ATMES在7种算法中排名第2.另外,与BLPSO相比,ODPSO有非常明显的性能改进.此外,为了比较对比算法之间的差异显著性,利用Wilcoxon符号秩和检验<sup>[30]</sup>在5%的显著性水平下进行多问题检验分析,检验结果如表3所示.表3中:“+”表示ODPSO明显优于对比算法,“≈”表示两种算法之间没有显著性差异.由表3可知:ODPSO明显优于BLPSO、ABC、PSO和DE;而与ATMES和M-ABC相比,ODPSO没有显著性差异.

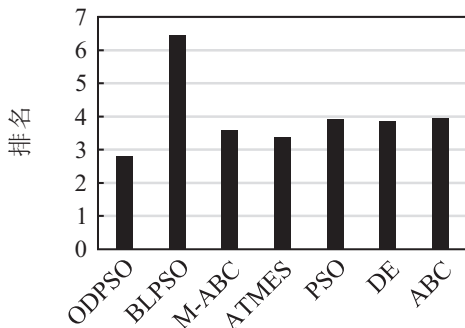


图3 对比算法在CEC2006上的Friedman检验排名

表3 ODPSO与其他算法对CEC2006测试函数结果对比

算法	R+	R-	decision
BLPSO	231	0	+
M-ABC	172.5	80.5	≈
ATMES	125.5	127.5	≈
PSO	189	64	+
DE	183	48	+
ABC	207.5	45.5	+

为了进一步说明所提出算法的稳健性,选择CEC2006的22个测试函数中的4个代表性的测试函数g09(多项式项类型)、g11(二次项类型)、g14(非线性类型)和g24(线性类型).ODPSO与其他对比算法关于各测试函数的箱线图如图4~图7所示.

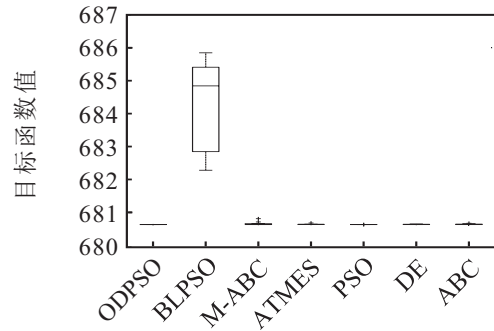


图4 ODPSO与对比算法关于g09的箱线图

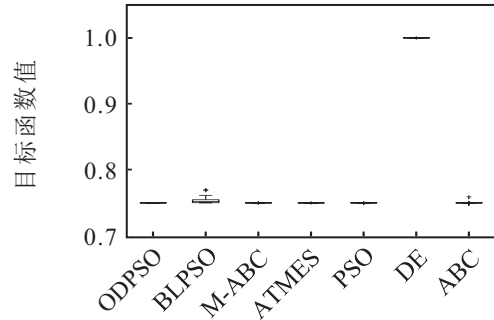


图5 ODPSO与对比算法关于g11的箱线图

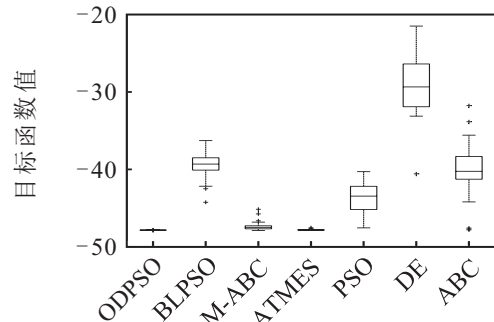


图6 ODPSO与对比算法关于g14的箱线图

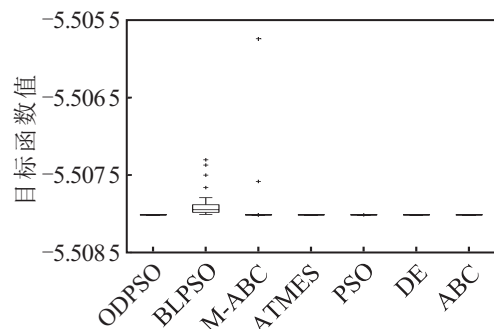


图7 ODPSO与对比算法关于g24的箱线图

图4~图7显示了不同算法针对4种测试函数独立运行25次得到的结果分布情况.通过箱线图的比较也可看出,所提出的ODPSO算法在准确性和可靠性方面与对比算法相比有较为出色的表现.

基于以上对比分析结果,在解决约束优化问题时,与PSO、DE、ABC、ATMES、M-ABC和BLPSO算法相比,ODPSO在寻优精度和稳定性方面更具有竞争力.



## 4 求解电力系统经济调度问题

### 4.1 测试系统1

首先,将改进的粒子群优化算法应用到一个包含6单元发电机组的测试系统,该系统负载为1 263 MW,在此系统中考虑传输损失与禁区区间等约束条件.这个测试系统的所有数据取自文献[31]与文献[32],机组参数可取自文献[33].

将改进的算法独立运行50次取其平均值,与PSO、DE、BLPSO、M-ABC、ATMES和ABC独立运行50次取得的平均值进行比较.各个算法的参数设置均取自相应文献的推荐值,最大评估次数均设置为240 000.表4给出了各个算法的平均结果,可以看出ODPSO获得的平均燃料成本低于其他对比算法.

表4 在测试系统1时,各算法的结果对比

算法	平均燃料成本/(\$/h)
ODPSO	<b>15 457.395 5</b>
PSO	15 494.396 69
M-ABC	15 473.105 91
ATMES	15 461.498 9
DE	15 498.466 11
BLPSO	15 486.873 55
ABC	15 493.244 83

图8显示了ODPSO与对比算法对测试系统1的收敛曲线对比.从图8可以看出,每个算法的图形都会存在上下振荡,这是因为算法在迭代寻优过程中,解的选择是优先选择约束违反度值较小的解.当解的约束违反度值为零之后,基于目标函数值进行解的选择,收敛曲线才会逐渐递减.从图8可以看出,相对多数对比算法,ODPSO找到较优的可行解所经历的评估次数较少,收敛速度较快.在最终的收敛精度方面,ODPSO同样也表现出优秀的性能.

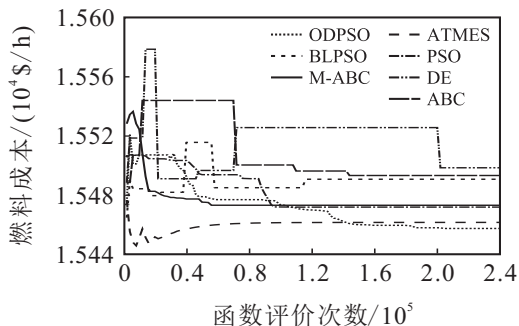


图8 ODPSO与对比算法对测试系统1的收敛曲线

### 4.2 测试系统2

测试系统2有15个发电机组,整个系统的负载需求为2 630 MW.发电机组2、5、6与12有禁止运行区间,这些数据可以从文献[34]中获得,系统的其他数

据从文献[33]中获得,机组参数从文献[33]获得.将改进的算法独立运行50次取其平均值,与PSO、DE、BLPSO、M-ABC、ATMES和ABC独立运行50次取得的平均值进行比较.各个算法的参数设置均取自相应文献,最大评估次数均设置为240 000.表5给出了各个算法的平均结果.从表5可以看出,虽然ODPSO得到的平均燃料成本要稍微高于ATMES,但是相对于其他5种对比算法,ODPSO获得的平均燃料成本均较低,结果表明ODPSO算法性能相比于大多数对比算法具有竞争力.

表5 在测试系统2时,各算法的结果对比

算法	平均燃料成本/(\$/h)
ODPSO	<b>33 060.993 81</b>
PSO	33 181.307 9
M-ABC	33 129.945 31
ATMES	32 997.194 28
DE	33 203.230 82
BLPSO	33 227.510 1
ABC	33 175.495 8

图9显示了ODPSO与对比算法对测试系统2的收敛曲线对比.从图9可以看出,相比于BLPSO、DE、M-ABC、PSO和ABC,ODPSO能够以较少的评估次数收敛到较优区域,且最终得到的解的质量较高.相比于ATMES,虽然ODPSO最终得到的解的质量较差,但该算法在前50 000次函数评估中,就找到了相对较优的可行解.

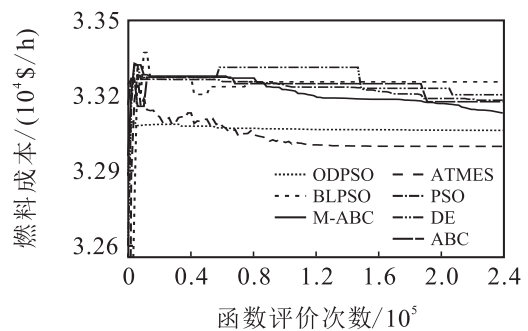


图9 ODPSO与对比算法对测试系统2的收敛曲线

上述这两个案例结果的对比表明,相对于对比算法,ODPSO在收敛速度和精度方面均表现出了较优或相当的综合性能.以上结果表明,本文设计的改进粒子群优化算法在求解电力系统经济调度问题时,同样表明出了优秀的求解性能.

## 5 结论

本文提出了一种改进的粒子群优化算法,并应用于求解电力系统的经济调度问题.通过在搜索前期和搜索后期分别引入广义反向学习策略和差分进化

的改进变异交叉机制,从而在改进算法的搜索前期加速逼近较优的搜索区域,提高收敛速度;搜索后期在较优搜索区域实施局部精细搜索,提高解的质量.将改进的算法在CEC2006中的22个测试函数上进行仿真测试,并应用到电力系统经济调度的两个实际案例中.分析和对比结果表明,与其他算法相比,改进的粒子群优化算法能够较好地解决复杂约束优化问题,在寻优精度和稳定性方面具有较好的优势.

在未来工作中,可以将改进粒子群算法应用到更多的实际优化问题中.另外,改进算法在约束处理方法上还有改进空间,如文献[35]中采用的无参数惩罚约束方法在处理约束优化问题时就得到了不错的效果.在以后的工作中,会考虑借鉴此约束处理方法,进一步改善所提算法.

#### 参考文献(References)

- [1] Al-Betar M A, Awadallah M A, Khader A T, et al. Economic load dispatch problems with valve-point loading using natural updated harmony search[J]. *Neural Computing and Applications*, 2018, 29(10): 767-781.
- [2] Mehmood K, Ahmad A. Improved grey wolf optimization for economic load dispatch problem considering valve point loading effect and prohibited operating zones[J]. *The Nucleus*, 2018, 54(4): 250-257.
- [3] 黄松, 王艳, 纪志成. 多目标粒子群算法的动态多燃料经济环境负荷分配[J]. *控制与决策*, 2018, 33(7): 1255-1263.  
(Huang S, Wang Y, Ji Z C. Dynamic multiple-fuels economic environmental dispatch using multi-objective particle swarm optimization[J]. *Control and Decision*, 2018, 33(7): 1255-1263.)
- [4] 郑晓菁. 基于人工蜂群优化法的多区域电力系统经济调度[J]. *计算机工程与科学*, 2015, 37(8): 1533-1539.  
(Zheng X J. Multi-area economic dispatch of power system based on artificial bee colony optimization[J]. *Computer Engineering and Science*, 2015, 37(8): 1533-1539.)
- [5] 吴亮红, 王耀南, 袁小芳, 等. 基于快速自适应差分进化算法的电力系统经济负荷分配[J]. *控制与决策*, 2013, 28(4): 557-562.  
(Wu L H, Wang Y N, Yuan X F, et al. Fast self-adaptive differential evolution algorithm for power economic load dispatch[J]. *Control and Decision*, 2013, 28(4): 557-562.)
- [6] 王凌, 黄付卓, 李灵坡. 基于混合双种群差分进化的电力系统经济负荷分配[J]. *控制与决策*, 2009, 24(8): 1156-1160.  
(Wang L, Huang F Z, Li L P. Economic distribution of power systems based on hybrid differential evolution with double populations[J]. *Control and Decision*, 2009, 24(8): 1156-1160.)
- [7] Adarsh B R, Raghunathan T, Jayabarathi T, et al. Economic dispatch using chaotic bat algorithm[J]. *Energy*, 2016, 96: 666-675.
- [8] Modiri-Delshad M, Aghay K S H, Taslimi-Renani E, et al. Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options[J]. *Energy*, 2016, 116: 637-649.
- [9] Zou D, Li S, Wang G G, et al. An improved differential evolution algorithm for the economic load dispatch problems with or without valve-point effects[J]. *Applied Energy*, 2016, 181: 375-390.
- [10] Yousefi N. Solving nonconvex economic load dispatch problem using particle swarm optimization with time varying acceleration coefficients[J]. *Complexity*, 2016, 21(6): 299-308.
- [11] Kennedy J, Eberhart R. Particle swarm optimization[J]. *Proceedings of IEEE International Conference On Neural Networks*, 1995, 4(8): 1942-1948.
- [12] Qin Q, Cheng S, Chu X, et al. Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization[J]. *Applied Soft Computing*, 2017, 59: 229-242.
- [13] Jain N K, Nangia U, Jain J. Economic load dispatch using adaptive social acceleration constant based particle swarm optimization[J]. *Journal of the Institution of Engineers*, 2018, 99(5): 431-439.
- [14] Basu M. Modified particle swarm optimization for nonconvex economic dispatch problems[J]. *International Journal of Electrical Power and Energy Systems*, 2015, 69(7): 304-312.
- [15] Ghorbani N, Babaei E. Exchange market algorithm for economic load dispatch[J]. *International Journal of Electrical Power and Energy Systems*, 2016, 75(2): 19-27.
- [16] Mahdavi S, Rahnamayan S, Deb K. Opposition based learning: A literature review[J]. *Swarm and Evolutionary Computation*, 2018, 39: 1-23.
- [17] Wang H, Wu Z, Rahnamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning[J]. *Information Sciences*, 2011, 181(20): 4699-4714.
- [18] 辛斌, 陈杰. 粒子群优化与差分进化混合算法的综述与分类[J]. *系统科学与数学*, 2011, 31(9): 1130-1150.  
(Xin B, Chen J. Overview and classification of particle swarm optimization and differential evolution hybrid algorithms[J]. *Journal of System Science and Mathematical*, 2011, 31(9): 1130-1150.)
- [19] 王勇, 蔡自兴, 周育人, 等. 约束优化进化算法[J]. *软件*

- 学报, 2009, 20(1): 11-29.  
(Wang Y, Cai Z X, Zhou Y R, et al. Constrained optimization evolutionary algorithms[J]. Journal of Software, 2009, 20(1): 11-29.)
- [20] Peng C, Liu H L, Gu F. A novel constraint-handling technique based on dynamic weights for constrained optimization problems[J]. *Soft Computing*, 2018, 22(12): 3919-3935.
- [21] Datta R, Deb K, Segev A. A bi-objective hybrid constrained optimization (HyCon) method using a multi-objective and penalty function approach[C]. 2017 IEEE Congress on Evolutionary Computation (CEC). San Sebastián: IEEE, 2017: 317-324.
- [22] 李智勇, 黄滔, 陈少淼, 等. 约束优化进化算法综述[J]. 软件学报, 2017, 28(6): 1529-1546.  
(Li Z Y, Huang T, Chen S M, et al. Overview of constrained optimization evolutionary algorithms[J]. Journal of Software, 2017, 28(6): 1529-1546.)
- [23] Richardson A. Nonparametric statistics for non-statisticians: A step-by-step approach by Gregory W. Corder, Dale I. Foreman[J]. *International Statistical Review*, 2010, 78(3): 451-452.
- [24] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341-359.
- [25] Chen X, Tianfield H, Mei C, et al. Biogeography-based learning particle swarm optimization[J]. *Soft Computing*, 2016, 21(24): 1-23.
- [26] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm[J]. *Journal of Global Optimization*, 2007, 39(3): 459-471.
- [27] Mezura-Montes E, Cetina-Domínguez O. Empirical analysis of a modified artificial bee colony for constrained numerical optimization[J]. *Applied Mathematics and Computation*, 2012, 218(22): 10943-10973.
- [28] Wang Y, Cai Z, Zhou Y, et al. An adaptive tradeoff model for constrained evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(1): 80-92.
- [29] Yu K, Wang X, Wang Z. Constrained optimization based on improved teaching-learning-based optimization algorithm[J]. *Information Sciences*, 2016, 352: 61-78.
- [30] Rey D, Neuhäuser M. Wilcoxon-signed-rank test[M]. Berlin: Springer, 2011: 1658-1659.
- [31] Mohammadi-Ivatloo B, Rabiee A, Soroudi A, et al. Iteration PSO with time varying acceleration coefficients for solving non-convex economic dispatch problems[J]. *International Journal of Electrical Power and Energy Systems*, 2012, 42(1): 508-516.
- [32] Saber A Y. Economic dispatch using particle swarm optimization with bacterial foraging effect[J]. *International Journal of Electrical Power and Energy Systems*, 2012, 34(1): 38-46.
- [33] Cai J, Ma X, Li L, et al. Chaotic particle swarm optimization for economic dispatch considering the generator constraints[J]. *Energy Conversion and Management*, 2007, 48(2): 645-653.
- [34] Jayabarathi T, Raghunathan T, Adarsh B R, et al. Economic dispatch using hybrid grey wolf optimizer[J]. *Energy*, 2016, 111: 630-641.
- [35] 陈美蓉, 郭一楠, 巩敦卫, 等. 一类新型动态多目标鲁棒进化优化方法[J]. 自动化学报, 2017, 43(11): 2014-2032.  
(Chen M R, Guo Y N, Gong D W, et al. A novel dynamic multi-objective robust evolutionary optimization method[J]. *Acta Automatica Sinica*, 2017, 43(11): 2014-2032.)

### 作者简介

梁静 (1981—), 女, 教授, 博士生导师, 从事计算智能及其应用等研究, E-mail: liangjing@zzu.edu.cn;

葛士磊 (1994—), 男, 硕士生, 从事计算智能及其应用的研究, E-mail: shileige1001@163.com;

瞿博阳 (1984—), 男, 教授, 博士, 从事计算智能及其应用等研究, E-mail: qby1984@hotmail.com;

于坤杰 (1990—), 男, 副教授, 博士, 从事进化计算及其应用的研究, E-mail: yukunjie@zzu.edu.cn.

(责任编辑: 齐 霖)