

多目标粒子群算法的动态多燃料经济环境负荷分配

黄 松¹, 王 艳^{2†}, 纪志成²

(1. 常州工学院 电气与光电工程学院, 江苏 常州 213002; 2. 江南大学 物联网工程学院, 江苏 无锡 214122)

摘 要: 考虑动态的负荷需求和多种燃料资源, 以经济成本和环境成本为优化指标, 建立动态多燃料经济环境负荷分配的多目标优化模型, 并提出一种多目标粒子群优化算法求解该类优化模型. 模型采用动态负荷需求和多种燃料资源, 更有利于节约电能成本和提高能源利用效率, 但高维数、复杂非线性和多目标成为求解该优化模型的难点, 故在算法中引入多目标解集更新策略和变邻域搜索策略. 实验仿真结果表明, 该模型是有效的, 且采用所提算法求解这类模型时所获得的近似 Pareto 前端的精度明显优于其他算法.

关键词: 粒子群优化算法; 多目标优化; 经济环境负荷分配; 电力系统

中图分类号: TP18; TM85

文献标志码: A

Dynamic multiple-fuels economic environmental dispatch using multi-objective particle swarm optimization

HUANG Song¹, WANG Yan^{2†}, JI Zhi-cheng²

(1. School of Electrical and Optoelectronic Engineering, Changzhou Institute of Technology, Changzhou 213002, China; 2. School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

Abstract: Aiming at economic cost and environmental cost, and considering dynamic load demand and multiple fuels, a multi-objective dynamic multi-fuel economic environmental dispatch model is established and a multi-objective particle swarm optimization is proposed to solve the optimization model. Dynamic load and multiple fuels benefit energy cost saving and the improvement of energy utilization efficiency of the model. However, it is a hard task to solve it due to the high dimensions, complex nonlinearity and multiple objectives. In the algorithm, an update strategy of multi-objective set and variable neighborhood search are introduced. Simulation results show that the model is valid and the approximate Pareto front obtained by the proposed algorithm is superior to other algorithms in terms of the degree of approximation for solving the problem.

Keywords: particle swarm optimization; multi-objective optimization; economic environmental dispatch; power system

0 引 言

电能是当前生产和生活中获得能源的主要形式, 电能消费具有一定的实时性和季节性. 电力系统在发电时可能包含不同类型的发电机组, 各种发电机组的发电功率、可靠性、效率、SO₂ 和 CO₂ 排放量等属性各不相同. 因此, 需要合理地安排各发电机组的电力负荷, 使电力系统安全可靠运行, 同时减少经济成本和环境成本.

群智能优化方法在求解电力系统静态经济负荷分配 (Economic load dispatch, ELD) 这类具有不连续、不可导、非凸等特点的非线性优化问题时具有很强

的优势, 常常有学者研究采用群智能优化方法求解 ELD 问题的可行性和效率^[1-4]. 邹恩等^[1] 提出一种基于 Tent 映射的混沌变异粒子群优化算法以克服算法的早熟收敛, 然后采用该算法解决电力系统的 ELD 问题; Hosseinnezhad 等^[2] 提出一种基于相位角矢量的 θ 粒子群优化算法来求解带网络损耗和死区约束的 ELD 问题; Wang 等^[3] 将差分进化与和声搜索算法的混合算法应用于多种不同约束情况下具有多燃料的 ELD 问题; Mandal 等^[4] 将磷虾算法应用于求解带阀点效应、死区约束和多燃料的 ELD 问题.

由于火力发电存在环境污染, 减少污染物排放

收稿日期: 2017-07-18; 修回日期: 2017-11-23.

基金项目: 国家自然科学基金项目 (61572238); 江苏省杰出青年基金项目 (BK20160001).

责任编委: 柴利.

作者简介: 黄松 (1984—), 男, 博士, 从事群智能优化算法及其应用的研究; 王艳 (1978—), 女, 教授, 博士生导师, 从事网络化控制系统、制造系统能效优化等研究.

†通讯作者. E-mail: wangyan88@jiangnan.edu.cn

和环保节能成为电力工业调整的重要方向. 电力系统经济环境负荷分配问题 (Economic environmental dispatch, EED) 是多目标优化问题, 需要考虑环境成本, 这增加了问题的难度. Hota 等^[5] 提出了一种改进的模糊细菌觅食算法求解 EED 问题. 该算法采用全局最优自然选择的觅食策略和模糊决策方法获得最优的折衷解, 并应用于 6 机组 30 总线的测试系统和 40 机组的台湾电力系统; Wang 等^[6] 建立了电力系统的随机性模型和确定性模型, 提出了一种改进的粒子群优化算法求解具有随机性和确定性的 EED 问题; Basu^[7] 提出了一种多目标差分进化算法求解带阀点效应的 EED 问题.

电能消费具有波动性, 静态电能负荷规划会对电能资源造成浪费. 考虑到电能消费的实时性, Jiang 等^[8] 提出了一种改进的自适应多目标差分进化算法求解动态经济环境负荷分配问题 (Dynamic economic environmental dispatch, DEED); Guo 等^[9] 提出了一种具有多个生产者的群搜索优化算法 (GSOMP) 求解 DEED 问题, 并对该算法及 MOPSO 算法和 NSGA-II 算法在 30 总线和 118 总线算例上进行了仿真验证; Basu^[10] 采用一种快速非支配排序的遗传算法 II 求解 DEED 问题; Elaiw 等^[11] 提出了一种采用差分进化算法或粒子群优化算法进行全局搜索, 采用序列二次规划进行微调的混合算法求解考虑阀点效应的 DEED 问题; Zhang 等^[12] 提出了一种具有模拟退火技术的混合差分进化算法 (MOHDE-SAT) 求解 DEED 问题, 该算法通过模拟退火技术和熵多样性方法自适应地调节种群多样性, 并引入了启发式约束处理方法; Niknam 等^[13] 提出了一种引入小生境机制和模糊聚类方法的 θ 多目标教学优化算法求解 DEED 问题, 并通过求解 5 机组、10 机组、120 机组的算例验证了其性能.

由于不同燃料的燃烧效率、污染物排放和市场价格均不相同, 各种燃料的经济成本和污染成本也各不相同. 合理选择燃料能够更有效地减少不必要的经济成本和环境成本, 获得更好的经济效益和社会效益, 同时也使得决策更具灵活性. 受到物理特性的影响, 发电机组本身具有一定的运行死区. 综合上述多种因素, 本文建立考虑多燃料情况和动态负荷需求, 且具有死区约束等多种约束的经济环境负荷分配问题的模型, 提出一种基于 Pareto 和均匀分解的混合多目标粒子群优化算法. 该模型在 EED 模型的基础上增加多种燃料的负荷发电情况和基于时段的用电负荷需求, 同时考虑到电机的运行死区, 更加符合实际需求. 在算法上, 首先改进粒子速度的更新策略, 通过

定义个体最优集合和全局最优集合来保存多目标问题的 Pareto 非劣解, 并通过 Pareto 支配关系和基于分解的标量子问题的混合策略更新个体最优解集和全局最优解集, 然后对全局最优解集进行变邻域搜索来提高优化性能. 该算法能够有效地提高求解这类优化问题的收敛精度.

1 多目标经济环境负荷分配问题

1.1 目标函数

电力系统经济环境负荷分配问题是指在等式和不等式约束下, 如何优化电力系统的资源和分配负荷, 使得电力系统的经济成本和环境成本最小的最优化问题. 由于不同时段对负荷的需求不同, 考虑最小化 H 时段动态负荷分配的经济成本和环境成本的数学公式如下:

$$\begin{cases} \min F = \sum_{t=1}^H \sum_{i=1}^M F_i^{(t)}, \\ \min E = \sum_{t=1}^H \sum_{i=1}^M E_i^{(t)}. \end{cases} \quad (1)$$

其中: M 为发电机的数量, H 为时段数, $F_i^{(t)}$ 为第 i 台发电机 t 时段的经济成本, $E_i^{(t)}$ 为第 i 台发电机 t 时段的环境成本, F 为总经济成本, E 为总环境成本.

每台发电机的经济成本可以拟合二次函数, 环境成本为非线性函数. 第 i 台发电机的经济成本和环境成本的具体数学公式如下:

$$\begin{cases} F_i^{(t)} = a_i(P_i^{(t)})^2 + b_i P_i^{(t)} + c_i, \\ E_i^{(t)} = \alpha_i(P_i^{(t)})^2 + \beta_i P_i^{(t)} + \gamma_i + \eta_i \exp(\delta_i P_i^{(t)}). \end{cases} \quad (2)$$

其中: $P_i^{(t)}$ 为第 i 台第 t 时段发电机的有功功率, a_i 、 b_i 和 c_i 分别为第 i 台发电机的经济成本系数, α_i 、 β_i 、 γ_i 、 η_i 和 δ_i 分别为第 i 台发电机的环境成本系数.

发电机启动过程中会产生阀点效应, 导致发电功率具有起伏现象. 具体地, 阀点效应的公式如下:

$$e_i^{(t)} = |g_i \sin[h_i(P_i^{\min} - P_i^{(t)})]|. \quad (3)$$

其中: g_i 和 h_i 为第 i 个发电机组的阀点效应系数, P_i^{\min} 为第 i 台发电机发电功率的下限. 考虑阀点效应的影响, 第 i 台发电机经济成本的数学公式如下:

$$F_i^{(t)} = a_i(P_i^{(t)})^2 + b_i P_i^{(t)} + c_i + |g_i \sin[h_i(P_i^{\min} - P_i^{(t)})]|. \quad (4)$$

由于发电机存在使用多种燃料的情况, 且多种燃料的经济成本和环境成本的曲线各不相同, 考虑多种燃料的经济成本和环境成本可描述为

$$\bar{F}_i^{(t)} =$$

$$\begin{cases}
 a_{i1}(P_i^{(t)})^2 + b_{i1}P_i^{(t)} + c_{i1} + & P_i^{\min} \leq P_i^{(t)} \leq P_i^{\max}, \\
 |g_{i1} \sin[h_{i1}(P_i^{\min} - P_i^{(t)})]|, & \\
 P_i^{\min} \leq P_i^{(t)} \leq P_{i1}, \text{ 燃料1;} & \\
 a_{i2}(P_i^{(t)})^2 + b_{i2}P_i^{(t)} + c_{i2} + & \\
 |g_{i2} \sin[h_{i2}(P_i^{\min} - P_i^{(t)})]|, & \\
 P_{i1} < P_i^{(t)} \leq P_{i2}, \text{ 燃料2;} & \\
 \vdots & \\
 a_{ik}(P_i^{(t)})^2 + b_{ik}P_i^{(t)} + c_{ik} + & \\
 |g_{ik} \sin[h_{ik}(P_i^{\min} - P_i^{(t)})]|, & \\
 P_{i(k-1)} < P_i^{(t)} \leq P_i^{\max}, \text{ 燃料}k. &
 \end{cases} \quad (5)$$

$$\begin{cases}
 \alpha_{i1}(P_i^{(t)})^2 + \beta_{i1}P_i^{(t)} + \gamma_{i1} + \eta_{i1} \exp(\delta_{i1}P_i^{(t)}), & \\
 P_i^{\min} \leq P_i^{(t)} \leq P_{i1}, \text{ 燃料1;} & \\
 \alpha_{i2}(P_i^{(t)})^2 + \beta_{i2}P_i^{(t)} + \gamma_{i2} + \eta_{i2} \exp(\delta_{i2}P_i^{(t)}), & \\
 P_{i1} < P_i^{(t)} \leq P_{i2}, \text{ 燃料2;} & \\
 \vdots & \\
 \alpha_{ik}(P_i^{(t)})^2 + \beta_{ik}P_i^{(t)} + \gamma_{ik} + \eta_{ik} \exp(\delta_{ik}P_i^{(t)}), & \\
 P_{i(k-1)} < P_i^{(t)} \leq P_i^{\max}, \text{ 燃料}k. &
 \end{cases} \quad (6)$$

其中: $\bar{F}_i^{(t)}$ 为第 i 台发电机 t 时段的多燃料经济成本, $\bar{E}_i^{(t)}$ 为第 i 台发电机 t 时段的多燃料环境成本, a_{ik} 、 b_{ik} 和 c_{ik} 分别为第 k 种燃料在第 i 台发电机上的经济成本系数, α_{ik} 、 β_{ik} 、 γ_{ik} 、 η_{ik} 和 δ_{ik} 分别为第 k 种燃料在第 i 台发电机上的环境成本系数。

综上可得, 考虑最小化多种燃料和动态负荷的经济成本和环境成本的数学公式如下:

$$\begin{cases}
 \min F = \sum_{t=1}^H \sum_{i=1}^M \bar{F}_i^{(t)}, \\
 \min E = \sum_{t=1}^H \sum_{i=1}^M \bar{E}_i^{(t)}.
 \end{cases} \quad (7)$$

1.2 约束条件

功率平衡约束属于等式约束. 为了保证安全稳定供电, 在任意时段均需要保证功率平衡, 其公式为

$$\sum_{i=1}^M P_i^{(t)} = P_L^{(t)} + P_S^{(t)}. \quad (8)$$

其中: $P_L^{(t)}$ 为 t 时段电力系统的总负荷, $P_S^{(t)}$ 为 t 时段电力系统的网络损耗.

功率上下限约束是发电机组安全稳定运行的保障, 发电机组运行时必须满足功率上下限约束, 具体的数学公式如下:

$$P_i^{\min} \leq P_i^{(t)} \leq P_i^{\max}, \quad (9)$$

其中 P_i^{\max} 和 P_i^{\min} 分别为第 i 台发电机安全运行的有功功率的上下限.

电力系统网络结构比较分散时不能忽略网络损耗的作用. 目前普遍采用 B 系数法计算网络损耗, 即

$$P_S^{(t)} = (P_i^{(t)})^T \mathbf{B} P_i^{(t)} + (P_i^{(t)})^T \mathbf{B}_0 + B_{00}. \quad (10)$$

其中: \mathbf{B} 为 $n \times n$ 维对称方阵, \mathbf{B}_0 为 n 维矢量, B_{00} 为常数. 上标 T 为矩阵转秩运算.

由于受到发电机组本身的物理特性的影响, 发电机存在着一些运行死区. 为了确保发电机平稳运行, 必须运行在死区之外, 死区约束为

$$\begin{cases}
 P_i^{\min} \leq P_i^{(t)} \leq P_{i1}^L, \\
 P_{i(j-1)}^U \leq P_i^{(t)} \leq P_{ij}^L, \\
 P_{iZ}^U \leq P_i^{(t)} \leq P_i^{\max}.
 \end{cases} \quad (11)$$

其中: P_{ij}^L 和 $P_{i(j-1)}^U$ 为第 j 个死区的有功功率的上下限, Z 为死区的个数.

此外, 发电机还有爬坡速率约束, 其公式为

$$\begin{cases}
 P_i^{(t)} \geq \max (P_i^{\min}, P_i^{(t-1)} - DR_i), \\
 P_i^{(t)} \leq \min (P_i^{\max}, P_i^{(t-1)} + UR_i),
 \end{cases} \quad (12)$$

其中 UR_i 和 DR_i 为爬坡速率的上下限值.

目前, 等式约束和不等式约束的处理方法有拒绝法、修复法、多目标法、罚函数法等. 对于 DEED 问题中存在的等式和不等式约束, 本文采用罚函数法处理, 具体数学描述如下:

$$\begin{cases}
 f_1(\mathbf{x}) = \\
 F(\mathbf{x}) + \sigma_1 \sum_{i=1}^l |h_i(\mathbf{x})| + \sigma_2 \sum_{j=1}^u \max \{0, g_j(\mathbf{x})\}, \\
 f_2(\mathbf{x}) = \\
 E(\mathbf{x}) + \sigma_1 \sum_{i=1}^l |h_i(\mathbf{x})| + \sigma_2 \sum_{j=1}^u \max \{0, g_j(\mathbf{x})\}.
 \end{cases} \quad (13)$$

其中: σ_1 和 σ_2 为惩罚因子, \mathbf{x} 为决策变量, $f_1(\mathbf{x})$ 和 $f_2(\mathbf{x})$ 分别为经济成本和环境成本的适应度, $h_i(\mathbf{x})$ 为等式约束, $g_j(\mathbf{x})$ 为不等式约束, $F(\mathbf{x})$ 为总经济成本, $E(\mathbf{x})$ 为总环境成本, l 和 u 分别为等式约束和不等式约束的数目.

2 混合多目标粒子群优化算法

2.1 基本粒子群优化算法

Kennedy 和 Eberhart 在 1995 年提出了粒子群优化算法^[14]. 粒子群优化算法的种群包含有限个粒子, 每个粒子具有速度和位置, 通过模拟鸟群被捕食行为改变其速度和位置来搜索可行域. 设 N 个 D 维粒子

的速度和位置分别为 \mathbf{v} 和 \mathbf{x} ,则 \mathbf{v} 和 \mathbf{x} 的数学描述为

$$\mathbf{v} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1D} \\ v_{21} & v_{22} & \cdots & v_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \cdots & v_{ND} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}.$$

每个粒子均有一个个体最优位置,整个种群具有一个全局最优位置.个体最优位置和全局最优位置分别代表着个体经验和社会经验.设个体最优位置和全局最优位置分别为 \mathbf{p}_{best} 和 \mathbf{g}_{best} ,则 \mathbf{p}_{best} 和 \mathbf{g}_{best} 的数学描述如下:

$$\mathbf{p}_{\text{best}} = \begin{bmatrix} p_{\text{best},11} & p_{\text{best},12} & \cdots & p_{\text{best},1D} \\ p_{\text{best},21} & p_{\text{best},22} & \cdots & p_{\text{best},2D} \\ \vdots & \vdots & \ddots & \vdots \\ p_{\text{best},N1} & p_{\text{best},N2} & \cdots & p_{\text{best},ND} \end{bmatrix},$$

$$\mathbf{g}_{\text{best}} = [g_{\text{best},1}, g_{\text{best},2}, \cdots, g_{\text{best},D}].$$

在采用粒子群优化算法时,需要对粒子速度和位置进行随机初始化,同时对个体最优位置和全局最优位置进行赋值.随机初始化的数学描述为

$$\begin{cases} v_{ij}^t = v_{\min,j} + (v_{\max,j} - v_{\min,j}) \times \text{rand}, \\ x_{ij}^t = x_{\min,j} + (x_{\max,j} - x_{\min,j}) \times \text{rand}. \end{cases} \quad (14)$$

其中: $i = 1, 2, \cdots, N$; $\text{rand} \in U(0, 1)$, $U(0, 1)$ 为均匀分布的随机数; $v_{\max,j}$ 和 $v_{\min,j}$ 为粒子速度第 j 维的上下限; $x_{\max,j}$ 和 $x_{\min,j}$ 为粒子位置第 j 维的上下限.

个体最优位置和全局最优位置的赋值过程为

$$\mathbf{p}_{\text{best},i}^t = \mathbf{x}_i^t, \quad \mathbf{g}_{\text{best}}^t = \text{argmin}\{f(\mathbf{p}_{\text{best},i}^t)\}. \quad (15)$$

在初始化后,粒子通过个体最优位置和全局最优位置学习个体经验和社会经验,从而更新粒子的速度和位置.基本粒子群算法的速度和位置更新公式为

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + c_1 r_1 (\mathbf{p}_{\text{best},i}^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{g}_{\text{best}}^t - \mathbf{x}_i^t), \quad (16)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}. \quad (17)$$

其中: ω 为惯性权重系数, c_1 和 c_2 为学习因子, r_1 和 r_2 为[0 1]上均匀分布的随机数.

2.2 Pareto支配关系

多目标问题需找到一系列相互折衷的解,使其尽可能接近多目标问题的Pareto前端.在多目标优化问题中,Pareto支配关系是非常重要的概念,Pareto最优集和Pareto前端均是由Pareto支配关系确定的.

定义1 (Pareto支配关系) 如果一个解 $\mathbf{s} = (s_1, s_2, \cdots, s_D)$ 支配另一个解 $\mathbf{u} = (u_1, u_2, \cdots, u_D)$,当且仅当 $\forall j \in \{1, 2, \cdots, m\}, f_j(\mathbf{s}) \leq f_j(\mathbf{u}), \exists j \in \{1, 2, \cdots, m\}, f_j(\mathbf{s}) < f_j(\mathbf{u})$,记作 $\mathbf{s} \succ \mathbf{u}$.其中: D 为决

策变量的维数, m 为优化目标的数目.

对于两个解 \mathbf{s} 和 \mathbf{u} ,如果支配关系 $\mathbf{s} \succ \mathbf{u}$ 和 $\mathbf{u} \succ \mathbf{s}$ 均不存在,则 \mathbf{s} 和 \mathbf{u} 为非支配关系.对于多目标问题的解 \mathbf{s} ,如果在搜索空间中不存在另外一个解 $\mathbf{u} \succ \mathbf{s}$,则 \mathbf{s} 称为Pareto最优解.所有Pareto最优解的集合称为Pareto最优集(Pareto set, PS).所有Pareto最优解在目标空间的集合称之为Pareto前端(Pareto front, PF)^[15-16].

2.3 混合多目标粒子群优化算法(MultiPSO)

2.3.1 粒子速度和位置的更新

这里介绍本文提出的基于Pareto和均匀分解的混合多目标粒子群优化算法(MultiPSO).首先,采用个体最优位置和全局最优位置同时更新粒子速度,不利于调整粒子多样性.为了更好地保持粒子多样性,本文采用个体最优位置和全局最优位置分别以一定的调节概率对粒子速度进行更新,这里可以通过调整调节概率大小来改变粒子多样性,更有利于避免求解多目标优化问题时的早熟问题.因此,粒子的速度和位置更新公式为

$$\mathbf{v}_i^{t+1} = \begin{cases} \omega \mathbf{v}_i^t + c_1 r_1 (\mathbf{p}_{\text{best},i}^t - \mathbf{x}_i^t), & \text{rand} < \varphi; \\ \omega \mathbf{v}_i^t + c_2 r_2 (\mathbf{g}_{\text{best}}^t - \mathbf{x}_i^t), & \text{others}. \end{cases} \quad (18)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}. \quad (19)$$

其中: φ 为调节概率,可以调节个体最优位置和全局最优位置的引导强度.

如果粒子速度和位置超过上下限,则对粒子速度和位置进行边界处理.第 i 个粒子第 j 维速度和位置的边界处理方法如下:

$$v_{ij}^{t+1} = v_{\min,j} + (v_{\max,j} - v_{\min,j}) \times \text{rand}. \quad (20)$$

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + (x_{\min,j} - x_{ij}^t) \times \text{rand}, & x_{ij}^{t+1} < x_{\min,j}; \\ x_{ij}^t + (x_{\max,j} - x_{ij}^t) \times \text{rand}, & x_{ij}^{t+1} > x_{\max,j}. \end{cases} \quad (21)$$

粒子群优化算法在迭代的过程中需要对个体最优位置和全局最优位置进行更新,下面主要通过定义个体最优解集和全局最优解集来更新个体最优位置和全局最优位置.

2.3.2 个体最优解集和全局最优解集的定义

由于多目标优化问题的两个解之间存在非支配关系,为了在个体最优和全局最优中同时保存多个Pareto非支配解的信息,本节定义个体最优解集和全局最优解集来存储每个粒子获得的非支配解和整个种群获得的非支配解.

定义2 (个体最优解集) 每个粒子定义一个集

合,用来保存每个粒子在搜索过程中所获得的非支配解和选择粒子的个体最优位置. 这个集合称为个体最优解集.

定义3(全局最优解集) 定义这样一个集合,用来保存整个种群在搜索过程中所获得的非支配解和选择全局最优位置. 这个集合称为全局最优解集.

2.3.3 个体最优解集和全局最优解集的更新

由于可能存在较多的Pareto非支配解,在每次迭代时,需要选择合适的标准对个体最优解集和全局最优解集进行更新,然后在更新后的个体最优解集和全局最优解集中选择个体最优位置和全局最优位置.

通过产生权重向量以及加权和的方法可以将多目标问题处理成标量子问题. 基于分解的进化算法(MOEA/D)通过产生的权重向量将多目标分解为一系列的标量子问题,在求解多目标问题时取得了不错的效果^[17]. 在本文中,由于每个粒子均存在一个个体最优解集,每个个体最优解集采用均匀设计方法产生的权重向量构成标量子问题,然后以Pareto支配关系和标量子问题作为这些解集的更新标准.

均匀设计方法是由方开泰教授和王元院士在1978年提出的一种实验设计方法^[18]. 均匀设计方法能安排较少且均匀分布的实验点并获得较好的实验结果. 设粒子群算法有 N 个粒子,优化目标 m 个,则需要产生 N 个 m 维的权重向量. 最常用的权重向量设计均匀性测度是中心化 L_2 -偏差,即 CD_2 -值^[19]. N 水平 m 因素的均匀分布表的 CD_2 -值计算如下:

$$CD_2 = \left(\frac{13}{12}\right)^m - \frac{2}{N} \sum_{k=1}^N \sum_{i=1}^m \left(1 + \frac{1}{2} \left| \lambda_{ki} - \frac{1}{2} \right| - \frac{1}{2} \left| \lambda_{ki} - \frac{1}{2} \right|^2\right) + \frac{2}{N^2} \sum_{k=1}^N \sum_{j=1}^N \prod_{i=1}^m \left(1 + \frac{1}{2} \left| \lambda_{ki} - \frac{1}{2} \right| + \frac{1}{2} \left| \lambda_{ji} - \frac{1}{2} \right| - \frac{1}{2} \left| \lambda_{ki} - \lambda_{ji} \right|\right). \quad (22)$$

均匀设计的权重向量:首先采用好格子法产生均匀设计表矩阵 $U_{N \times k} = (u_{ij})_{N \times k}$,其中 $k > m$,通过 $\lambda_{ij} = u_{ij}/N$ 将其转化为权重矩阵 $W'_{N \times k}$;通过随机选择 m 列构造一系列权重矩阵 $W_{N \times m}$;计算每个权重矩阵 $W_{N \times m}$ 的 CD_2 -值;选择具有最小 CD_2 -值的权重矩阵 $W_{N \times m}$ 的每一行作为 N 个粒子的权重向量.

通过均匀设计原理构造 N 个 m 维的均匀分布权重向量 λ ,再采用加权和的方法构造均匀分布的标量子问题. 设第 i 个粒子的权重向量为 $\lambda_i = (\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{im})$,对于所有的 $j = 1, 2, \dots, m$,有 $\sum_{i=1}^m \lambda_{ij} = 1$ 且 $\lambda_{ij} \in [0, 1]$. $f = (f_1, f_2, \dots, f_m)$ 为优化目标

或适应值,则产生标量子问题的公式如下:

$$SC_i = \sum_{j=1}^m \lambda_{ij} f_j. \quad (23)$$

当每个粒子的个体最优解集均拥有一个标量子问题后,可以对个体最优解集和全局最优解集进行更新. 设个体最优解集和全局最优解集保存Pareto非支配解上限分别为 N_a 和 N_g . 当个体最优解集和全局最优解集中Pareto非支配解的个数没有超过上限时,采用Pareto支配关系删除被支配的解;超过上限时,个体最优解集通过自己的标量子问题进行更新,全局最优解集则采用随机权重的标量子问题进行更新.

在更新全局最优解集和个体最优解集后,需要选择个体最优位置和全局最优位置. 通过选择个体最优解集中具有最好标量子问题的位置作为每个粒子的个体最优位置;通过选择全局最优解集中具有最好随机权重标量子问题的位置作为全局最优位置.

粒子 i 的个体最优解集为 Ω_i 和权重向量 λ_i ,全局最优解集为 Ω_g , $f = (f_1, f_2, \dots, f_m)$ 为适应值,个体最优解集 Ω_i 和个体最优位置 $p_{best,i}$ 的更新如表1所示.

表1 个体最优解集的更新

1:	for $i = 1 \rightarrow N$ do
2:	$\Omega \leftarrow$ 选择 $[\Omega_i, x_i]$ 中的非支配解;
3:	Count \leftarrow 计算 Ω_i 中解的个数;
4:	if Count $> N_a$ then
5:	对于所有的 $P_a \in \Omega_i$,计算其标量子问题SC;
6:	%进行升序排序;
7:	[ord, ind]=Sort(SC', Ascend');
8:	%选择前 N_a 个非支配解;
9:	$\Omega_i \leftarrow \Omega_i(\text{ind}(1:N_a), :)$;
10:	end if
11:	$p_{best,i} \leftarrow \Omega_j(1, :)$.
12:	end for

全局最优解集 Ω_g 和全局最优位置 g_{best} 的更新如表2所示.

表2 全局最优解集的更新

1:	% Ω_x 为 $t + 1$ 次迭代时种群的非支配解集.
2:	$\Omega_g \leftarrow$ 选择 $[\Omega_g, \Omega_x]$ 中的非支配解;
3:	Count \leftarrow 计算 Ω_g 中解的个数;
4:	if Count $> N_g$ then
5:	产生一个随机权重向量 ω . $\sum_{j=1}^m \omega_{ij} = 1, \omega_{ij} \in [0, 1]$.
6:	对于所有的 $P_a \in \Omega_g$,计算其标量子问题SC;
7:	%进行升序排序
8:	[ord, ind]=Sort(SC', Ascend');
9:	%选择前 N_g 个非支配解;
10:	$\Omega_g \leftarrow \Omega_g(\text{ind}(1:N_g), :)$;
11:	end if
12:	$g_{best} \leftarrow \Omega_g(1, :)$.

2.3.4 变邻域搜索

变邻域搜索具有很强的局部开发能力. 本文通过对全局最优解集进行变邻域搜索以增强算法的局部搜索能力. 首先需要定义邻域操作. 对于全局最优解集的解 \mathbf{x}_i , 这里定义3个邻域操作.

差分邻域 ξ_1 : 随机从全局最优解集中选取另外两个非支配解, $\mathbf{x}_a = (x_{a1}, x_{a2}, \dots, x_{an})$ 和 $\mathbf{x}_b = (x_{b1}, x_{b2}, \dots, x_{bn})$, 取 $\Delta_1 = \min\{\text{abs}(\mathbf{x}_{ak} - \mathbf{x}_{bk}) | k = 1, 2, \dots, n\}$, 随机选取 \mathbf{x}_i 的一维 j , 则 \mathbf{x}_i 的邻域为 $x_{ij} = x_{ij} + \Delta_1$.

随机邻域 ξ_2 : 从 \mathbf{x}_i 中随机选取一维 j , 则 \mathbf{x}_i 的随机邻域为 $x_{ij} = x_{\min,j} + (x_{\max,j} - x_{\min,j}) \times \text{rand}$.

交换邻域 ξ_3 : 从 \mathbf{x}_i 中随机选取一维 j , 从全局最优解集中随机选取一个非支配解 $\mathbf{x}_a = (x_{a1}, x_{a2}, \dots, x_{an})$, 则 \mathbf{x}_i 的交换邻域为 $x_{ij} = x_{aj}$.

由此可得变邻域搜索如表3所示. 在变邻域搜索中, 需要进行局部搜索, 具体伪代码如表4所示.

表3 变邻域搜索

1:	for $i = 1 \rightarrow \text{Count}$ do
2:	$k = 1$;
3:	while $k \leq 3$ do
4:	选择 $\mathbf{x}_i \in \Omega_g$, 从 \mathbf{x}_i 的邻域结构 ξ_k
5:	随机产生一个邻域解 s'
6:	对 s' 进行局部搜索得到邻域解 s''
7:	if $s'' \succ s'$ then
8:	$s' \leftarrow s''$; $k = 1$;
9:	else
10:	$k = k + 1$;
11:	end if
12:	end while
13:	if $s' \succ x_i$ then
14:	$x_i \leftarrow s'$;
15:	end if
16:	end for

表4 局部搜索

1:	%K 为局部搜索的个数;
2:	for $i = 1 \rightarrow K$ do
3:	随机选择一个邻域结构 ξ_k ;
4:	从 s' 的邻域结构 ξ_k 产生一个邻域解 s'' ;
5:	if $s'' \succ s'$ then
6:	$s' \succ s''$.
7:	end if
8:	end for

为尽可能多地保存 Pareto 非劣解集, 定义一个外部档案集 Arch, 更新方法为: 合并全局最优解集和外部档案集 $\text{Arch}^{t+1} = \text{Arch}^t \cup \Omega_g$, 删除其中所有被支配的解, 用剩下的非支配解更新外部档案集 Arch.

2.3.5 MultiPSO 算法求解多燃料的 DEED 问题

MultiPSO 算法在求解 DEED 问题时, 首先要确定决策变量. 在本文中, H 时段 n 组发电机的分配

负荷即为决策变量, 维数为 $H \times n$, 设 $P_n^{(t)}$ 为第 n 组发电机第 t 时段的负荷, 则 $\mathbf{x} = [P_1^{(1)}, \dots, P_1^{(H)}, P_2^{(1)}, \dots, P_2^{(H)}, \dots, P_n^{(1)}, \dots, P_n^{(H)}]$ 即为粒子位置.

MultiPSO 算法的具体实现步骤如下.

Step 1: 初始化粒子群算法参数, 确定各个决策变量的上下限, 按照式(14)和(15)初始化粒子速度和位置.

Step 2: 将粒子初始位置作为个体最优解集, 将种群的非支配解作为全局最优解集, 选出个体最优位置和全局最优位置.

Step 3: 判断算法是否满足停止条件. 是, 则执行 Step 9; 否则执行 Step 4 ~ Step 8.

Step 4: 根据式(18)和(19)更新粒子速度和位置, 按式(20)和(21)进行边界处理.

Step 5: 通过处理等式约束和不等式约束计算经济成本和环境成本, 并通过罚函数法(即式(13))计算所有粒子的多目标适应值.

Step 6: 根据 2.3.3 节产生均匀权重向量和标量子问题, 根据表1和表2更新个体最优解集和全局最优解集, 选择个体最优位置 \mathbf{p}_{best} 和全局最优位置 \mathbf{g}_{best} .

Step 7: 按照表3和表4对全局最优解集进行变邻域搜索.

Step 8: 通过全局最优解集 Ω_g 更新外部档案集 Arch, 返回 Step 3.

Step 9: 输出 Arch, 算法运行结束.

3 仿真实验与结果分析

3.1 参数设置和算例

下面将通过仿真实验说明建立动态多燃料经济环境负荷分配模型的必要性, 并验证 MultiPSO 算法在求解该问题上的性能. 本实验将在 Windows 7 操作系统的 Matlab 2011b 软件下进行. 硬件配置为: PC 机, Intel(R) core(TM) i3-3 240 处理器, 主频 3.4 GHz, 内存 4 GB. 本文选择具有两种燃料 5 机组的 DED1 算例进行仿真实验, 具体系数和相关参数如表5和表6所示. 在这里, $\sigma_1 = \sigma_2 = 10$, $H = 24$, 通过以前数据预测, 24 小时的动态负荷需求如表7所示, 其中 DED1 算例的 B 系数数据如下:

$$B = \begin{bmatrix} 4.9 & 1.4 & 1.5 & 1.5 & 2.0 \\ 1.4 & 4.5 & 1.6 & 2.0 & 1.8 \\ 1.5 & 1.6 & 3.9 & 1.0 & 1.2 \\ 1.5 & 2.0 & 1.0 & 4.0 & 1.4 \\ 2.0 & 1.8 & 1.2 & 1.4 & 3.5 \end{bmatrix} \times 10^{-5},$$

$$B_0 = [0, 0, 0, 0, 0],$$

$$B_{00} = 0.$$

表5 DED1算例的约束条件数据

发电机	P^{\min}/MW	P^{\max}/MW	DR/(MW/h)	UR/(MW/h)	PZ1/MW	PZ2/MW	功率范围	
							燃料1	燃料2
G_1	10	175	30	30	[20, 30]	[50, 60]	[10, 75]	[75, 175]
G_2	20	185	40	40	[30, 40]	[75, 95]	[70, 185]	[20, 70]
G_3	30	190	55	55	[40, 50]	[105, 125]	[110, 190]	[30, 110]
G_4	40	250	85	85	[50, 60]	[165, 185]	[40, 150]	[150, 250]
G_5	50	300	105	105	[60, 70]	[205, 235]	[50, 185]	[185, 300]

表6 DED1算例的经济成本系数和环境成本系数

发电机	燃料	a_i	b_i	c_i	g_i	h_i	α_i	β_i	γ_i	η_i	δ_i
G_1	燃料1	0.015 0	-2.0	225	110	0.042	290	-2.22	0.014 5	0.503	0.020
	燃料2	0.066 0	-8.10	850	125	0.035	360	-1.11	0.018 4	0.993	0.040
G_2	燃料1	0.012 5	-1.8	260	40	0.040	285	-2.22	0.014 5	0.503	0.020
	燃料2	0.046 0	-8.10	609	50	0.042	360	-1.11	0.014 2	0.993	0.040
G_3	燃料1	0.012 5	-3.6	300	50	0.038	295	-2.26	0.013 8	0.503	0.020
	燃料2	0.036 0	-8.10	607	50	0.042	150	-1.89	0.015 0	0.993	0.040
G_4	燃料1	0.010 5	-2.0	320	60	0.037	310	-2.26	0.013 8	0.503	0.020
	燃料2	0.032 4	-7.74	450	65	0.063	180	-2.08	0.012 1	0.914	0.045
G_5	燃料1	0.011 5	-1.8	240	80	0.035	310	-2.42	0.013 2	0.503	0.020
	燃料2	0.021 4	-7.74	540	65	0.063	180	-2.08	0.012 1	0.914	0.045

表7 DED1算例的24时段需求负荷

时段	P_L	时段	P_L	时段	P_L	时段	P_L
1	310	7	585	13	650	19	620
2	365	8	604	14	620	20	625
3	405	9	620	15	595	21	615
4	440	10	615	16	550	22	565
5	488	11	625	17	528	23	480
6	525	12	640	18	580	24	383

为了验证 MultiPSO 算法的性能,这里选择的对比算法为 NSGA-II^[20]、SPEAII^[21]和 MOCLPSO^[22],参数设置见表8.对于所有算法,独立运行20次,种群大小为50,迭代次数为300.

表8 对比算法参数设置

算法	参数设置
MultiPSO	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2, \varphi = 0.9, N_a = 5, N_g = 30;$
NSGA-II	$P_c = 0.9, P_m = 1/N, \eta_c = 20, \eta_m = 20$
SPEAII	$P_c = 0.9, P_m = 1/D, N_e = 50$
MOCLPSO	$P_c = 0.1, P_m = 0.4, c_1 = c_2 = 2$

由于 DED1 算例的真实 Pareto 前端尚不清楚,本文将采用 C 测度^[23](Coverage of two sets, C -metric)和超体积测度^[24](Hypervolume, HV-metric)评价算法获得的近似 Pareto 前端的好坏.

C -metric 的计算公式如下:

$$C(A, B) = \frac{|\{u \in B | \exists v \in A : v \succ u\}|}{|B|}$$

其中: A 和 B 分别为两种算法获得的近似 Pareto 前端. $|\cdot|$ 代表 Pareto 前端中非支配解的个数.

超体积测度是参考点与 Pareto 非支配解所框定的面积或体积的累加,能够综合反映出算法的收敛性

和多样性. HV-metric 的计算公式如下:

$$HV(A, \mathbf{r}) = \{\cup \text{volume}(s, \mathbf{r}) | s \in A, \mathbf{r} \in \mathbf{r}\}.$$

其中: A 为算法获得的近似 Pareto 前端, \mathbf{r} 为所选的参考点.

3.2 仿真结果与分析

下面分别取两组权重的实验结果说明同时考虑具有多种燃料选择的经济成本和环境成本模型的必要性.取两组固定权重,即 (0.5, 0.5) 和 (0.99, 0.01),则 MultiPSO 算法获得 Pareto 前端的最好解分别为 $(2.95, 3.09) \times 10^4$ 和 $(2.49, 5.87) \times 10^4$.对比两组最好解,相对于权重 (0.99, 0.01),当同时关注经济成本和环境成本时(即权重为 (0.5, 0.5)),经济成本上升了 18.5%,环境成本下降了 47.4%.相对于权重 (0.5, 0.5),关注经济效益(即权重为 (0.99, 0.01))使得经济成本下降了 15.6%,而环境成本却上升了 90%.因此,该模型的经济成本与环境成本相互具有较大影响,同时优化具有必要性.

图1和图2给出了两组固定权重时最好解的负荷分配图.

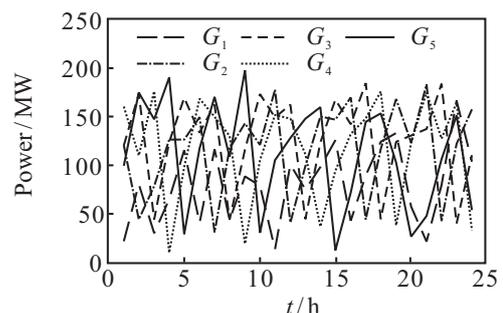


图1 最好解为 $(2.95, 3.09) \times 10^4$ 的负荷分配

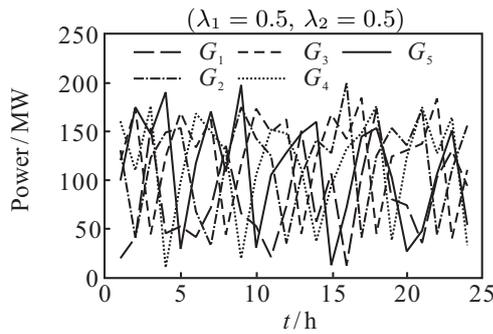


图2 最好解为 $(2.49, 5.87) \times 10^4$ 的负荷分配 $(\lambda_1 = 0.99, \lambda_2 = 0.01)$

通过图1和图2可以看出,对比两种方案, G_1 在时段3、5、7、8、9、10、14更换了燃料种类, G_2 在时段6更换了燃料种类, 而发电机组 G_3, G_4, G_5 在所有时段的燃料种类保持不变. 整个方案有 $24 \times 5 = 120$ 时段选择燃料种类, 而这里只有8个时段更换了燃料, 两种方案的变更较小, 未更换燃料的时段负荷变动也相对较小, 保持了更换方案的相对稳定性. 综上所述, 对部分发电机组的燃料种类进行更换能够对环境成本产生较大影响, 这说明考虑多种燃料能够控制环境成本, 增加了决策的灵活性. 因此, 考虑多种燃料的 DEED 问题是可行和必要的, 对实际应用具有一定的参考价值.

下面以4种算法的对比实验结果说明 MultiPSO 算法求解具有多种燃料的动态经济环境负荷分配模型的有效性. 表9为4种算法求解 DED1 的 C -metric 指标和 HV-metric 指标的实验结果数据.

表9 4种算法求解 DED1 的 C -metric 和 HV-metric

适应值	MultiPSO	NSGA-II	SPEAII	MOCLPSO
MultiPSO	0/0.00	0/0.00	0/0.00	0/0.00
NSGA-II	49/1.00	0/0.00	0/0.00	0/0.00
SPEAII	18/1.00	18/1.00	0/0.00	0/0.00
MOCLPSO	9/1.00	9/1.00	9/1.00	0/0.00
平均值	1.583e+012	3.593e+012	1.348e+012	1.125e+011
中值	1.563e+012	3.606e+012	1.305e+012	1.125e+011
最好值	1.978e+012	3.606e+012	1.708e+012	1.125e+011
最差值	1.293e+012	3.481e+012	1.255e+012	1.125e+011
方差	1.311e+011	3.930e+010	2.140e+011	1.608e-005
t -value	—	-4.133e+000	4.133e+000	4.221e+000
CPU	8.377e+001	2.709e+001	5.945e+001	2.720e+001

在表9前4行的数据中: ‘/’ 左侧数据代表第一行算法获得的非支配解支配第一列算法非支配解的个数, 右侧数据代表第一行算法获得 Pareto 前端与第一列算法获得 Pareto 前端的 C -metric 值; 后7行分别列出了4种算法20次运行获得近似 Pareto 前端的 HV-metric 指标的最好值、平均值、最差值、方差、 t -value 和平均 CPU 时间.

在表9中, 第一列的 C -metric 值均为1, 这说明 MultiPSO 算法获得的非支配解支配了 NSGA-II、

SPEAII、MOCLPSO 获得的所有非支配解. 由于 C -metric 反映 Pareto 前端的收敛性, 说明 MultiPSO 算法所得的近似 Pareto 前端的收敛性明显优于 NSGA-II、SPEAII 算法和 MOCLPSO 算法. MultiPSO 算法通过引入变邻域搜索, 增强了算法的局部寻优能力, 提高了算法的搜索精度, 使得该算法所得近似 Pareto 前端的收敛性明显提高.

从 HV-metric 指标的最好值、最差值、平均值和中值可以看出, MultiPSO 算法获得的结果要明显优于 MOCLPSO 算法, 与 SPEAII 算法相差不大, 劣于 NSGA-II, 但在同一数量级上. 就 HV-metric 指标的方差而言, MultiPSO 算法优于 SPEAII, 劣于 NSGA-II, 而 MOCLPSO 的方差几乎为0, 即10次获得的 HV-metric 指标没有变化, 已经陷入了局部最优值. 因此, MultiPSO 算法在求解该模型时具有不错的稳定性. 在表9中, t -value 均小于2.05, 说明 MultiPSO 算法获得的 HV-metric 指标在统计上要优于 SPEAII 和 MOCLPSO 算法, 劣于 NSGA-II 算法. 由于 HV-metric 指标能够反映算法多样性, MultiPSO 算法在收敛性和多样性上具有一定的折中, 获得了更好的收敛性. 就平均 CPU 时间而言, MultiPSO 算法的结果要劣于3种对比算法. 这是由于 MultiPSO 算法中加入了变邻域搜索, 在一定程度上增加了算法的时间消耗, 但是, 这样使得算法获得的 Pareto 前端的精度明显提高, 有益于获得更好的经济成本和环境成本.

图3为4种算法独立运行20次获得 DED1 的近似 Pareto 前端图.

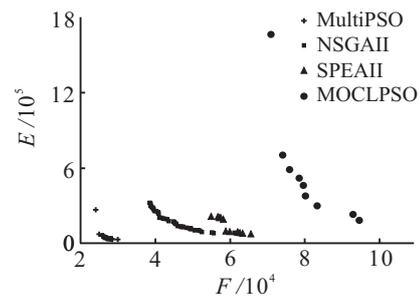


图3 4种算法的近似 Pareto 前端

从图3可以看出, MultiPSO 算法获得 Pareto 前端的收敛精度要明显优于其他3种算法. 就空间分布均匀性而言, MultiPSO 算法仅差于 NSGA-II 算法, 但优于 SPEAII 算法和 MOCLPSO 算法.

综上所述, MultiPSO 算法能够在不明显降低近似 Pareto 前端的分布均匀性的同时获得更高的质量和精确度, 收敛性更好, 有利于获得更低的经济成本和环境成本. 就降低经济成本以及环境成本而言, MultiPSO 算法求解动态多燃料经济环境负荷分配模型获得了更好的效果.

4 结论

电力系统经济负荷分配是一类典型的高维非线性优化问题. 本文建立了具有多种燃料和动态负荷需求的经济环境负荷分配问题的数学模型, 提出了一种基于 Pareto 和均匀分解的混合多目标粒子群优化算法. 这种具有多种燃料的 DEED 模型增加了决策灵活性, 有利于平衡经济成本和环境成本. 仿真实验验证了该模型的必要性, 同时验证了 MultiPSO 算法在求解该问题时具有更高的质量和精确度, 对于减少经济成本和环境成本有着重要的指导作用. 下一步需要建立更加实用的动态模型和采集实际电厂数据, 并进行进一步的验证.

参考文献(References)

- [1] 邹恩, 林兰, 黄浩扬, 等. 基于经济负荷分配的混沌迭代粒子群算法[J]. 桂林理工大学学报, 2014, 34(3): 579-583.
(Zou E, Lin L, Huang H Y, et al. Chaotic iteration particle swarm optimization algorithm based on economic load dispatch[J]. J of Guilin University of Technology, 2014, 34(3): 579-583.)
- [2] Hosseinnazhad V, Babaei E. Economic load dispatch using θ -PSO[J]. Int J of Electrical Power & Energy Systems, 2013, 49(7): 160-169.
- [3] Wang L, Li L. An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems[J]. Int J of Electrical Power & Energy Systems, 2013, 44(1): 832-843.
- [4] Mandal B, Roy P K, Mandal S. Economic load dispatch using krill herd algorithm[J]. Int J of Electrical Power & Energy Systems, 2014, 57(5): 1-10.
- [5] Hota P K, Barisal A K, Chakrabarti R. Economic emission load dispatch through fuzzy based bacterial foraging algorithm[J]. Int J of Electrical Power & Energy Systems, 2010, 32(7): 794-803.
- [6] Wang L, Singh C. Stochastic economic emission load dispatch through a modified particle swarm optimization algorithm[J]. Electric Power Systems Research, 2008, 78(8): 1466-1476.
- [7] Basu M. Economic environmental dispatch using multi-objective differential evolution[J]. Applied Soft Computing, 2011, 11(2): 2845-2853.
- [8] Jiang X, Zhou J, Wang H, et al. Dynamic environmental economic dispatch using multiobjective differential evolution algorithm with expanded double selection and adaptive random restart[J]. Int J of Electrical Power & Energy Systems, 2013, 49(1): 399-407.
- [9] Guo C X, Zhan J P, Wu Q H. Dynamic economic emission dispatch based on group search optimizer with multiple producers[J]. Electric Power Systems Research, 2012, 86(4): 8-16.
- [10] Basu M. Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II[J]. Int J of Electrical Power & Energy Systems, 2008, 30(2): 140-149.
- [11] Elaiw A M, Xia X, Shehata A M. Hybrid DE-SQP and hybrid PSO-SQP methods for solving dynamic economic emission dispatch problem with valve-point effects[J]. Electric Power Systems Research, 2013, 103(8): 192-200.
- [12] Zhang H, Yue D, Xie X, et al. Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch[J]. Applied Soft Computing, 2015, 34(C): 312-323.
- [13] Niknam T, Golestaneh F, Sadeghi M S. θ -Multiobjective teaching-learning-based optimization for dynamic economic emission dispatch[J]. IEEE Systems J, 2012, 6(2): 341-352.
- [14] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of the 6th Int Symposium on Micro Machine and Human Science. Nagoya: IEEE, 1995: 39-43.
- [15] Fonseca C M, Fleming P J. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization[C]. ICGA. Nagoya: Morgan Kaufmann Publishers Inc, 1993: 416-423.
- [16] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms[J]. Evolutionary Computation, 1994, 2(3): 221-248.
- [17] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Trans on Evolutionary Computation, 2007, 11(6): 712-731.
- [18] Fang K T. The uniform design: Application of number-theoretic methods in experimental design[J]. Acta Mathematicae Applicatae Sinica, 1980, 3(4): 363-372.
- [19] Fang K T, Lin D K J. Uniform experimental designs and their applications in industry[J]. Handbook of Statistics, 2003, 22(3): 131-170.
- [20] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [21] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization[J]. Evolutionary Methods for Design, Optimization, and Control, 2002: 95-100.
- [22] Huang V L, Suganthan P N, Liang J J. Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems[J]. Int J of Intelligent Systems, 2006, 21(2): 209-226.
- [23] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Trans on Evolutionary Computation, 1999, 3(4): 257-271.
- [24] Zitzler E, Thiele L. Multiobjective optimization using evolutionary algorithms—A comparative case study[C]. Int Conf on Parallel Problem Solving from Nature, Springer. Berlin: Heidelberg, 1998: 292-301.