

文章编号: 1001-0920(2011)12-1813-05

基于差分蜂群算法的电力系统经济负荷分配

熊伟丽¹, 徐 迈², 徐保国^{1a}

(1. 江南大学 a. 轻工过程先进控制教育部重点实验室, b. 物联网工程学院, 江苏 无锡 214122; 2. 清华大学 电子工程系, 北京 100084)

摘要: 针对电力系统经济负荷分配问题, 提出一种有效的差分蜂群算法. 受差分进化算法的启发, 该算法基于差分进化操作改进了雇佣蜂的搜索方式, 提高了探索能力和收敛速度. 此外, 提出一种有效的修复机制以保证新个体的可行性. 该算法在带有阀点效应和多燃料特征的典型电力系统经济负荷分配问题上进行了测试. 仿真结果验证了所提算法的有效性.

关键词: 经济负荷分配; 差分蜂群算法; 人工蜂群算法; 差分进化

中图分类号: TP13

文献标识码: A

Differential bee colony algorithm for non-convex economic load dispatch

XIONG Wei-li¹, XU Mai², XU Bao-guo^{1a}

(1a. Key Laboratory of Advanced Process Control for Light Industry of Ministry of Education, 1b. School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China; 2. Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. Correspondent: XIONG Wei-li, E-mail: greenpre@163.com)

Abstract: A differential artificial bee colony algorithm is proposed for economic load dispatch problems of power systems. Inspired by the differential evolution algorithm, the search behaviors of employed bees are improved by using the differential operators, which enhances the exploration ability and accelerates the converge speed of the algorithm. Moreover, an effective repair procedure is proposed to guarantee the feasibility of the new individuals. Experiments are carried out on some typical problems with valve-point effects and multiple fuels. Simulation results show the effectiveness of the proposed algorithm.

Key words: economic load dispatch; differential artificial bee colony; artificial bee colony; differential evolution

1 引言

电力系统经济负荷分配(ELD)是在满足电力需求和电机特征等约束下, 寻找系统中不同电机的负荷分配, 从而使总体发电成本最小化. 该问题是电力系统中最常见的一种问题, 较明显地影响着电力系统的经济效益, 因此受到了国内外学者的广泛关注. 受阀点效应、多种燃料等因素的影响, 经济负荷分配问题往往具有强非线性和多极小等特征, 这使得经典基于梯度的数学算法难以直接应用. 尽管动态规划算法^[1-2]能避免梯度计算, 但对于高维问题容易造成“维数爆炸”, 很难推广应用. 由于智能优化算法对问题特征没有明确要求, 并具有很强的全局搜索能力, 近年来, 在经济负荷分配问题上得到了广泛应用, 如遗传算法^[3]、进化规划^[4]、粒子群算法^[5]、人工免疫算

法^[6]和差分进化算法^[7]等.

人工蜂群(ABC)算法^[8]是一种新兴的智能优化算法. 该算法受采蜜过程中蜜蜂行为的启发, 模仿雇佣蜂、旁观蜂和侦查蜂的行为, 提出了3种搜索方式. 搜索过程中3种方式相互协作, 共同引导算法的进化. 目前, 国内外对于ABC算法的研究刚刚开始, 有大量的领域值得探索.

本文针对ELD问题, 提出了一种改进的差分人工蜂群算法(DABC), 并应用于考虑阀点效应和多种燃料特征的经济负荷分配问题.

2 问题描述

ELD问题要在满足系统负荷需求和电机特征等约束条件下, 最优化总体发电成本. 一般情况下, 该问题可描述为如下形式:

收稿日期: 2010-08-19; 修回日期: 2010-12-24.

基金项目: 国家自然科学基金项目(30971689, 60805014); 中央高校基本科研业务费专项资金项目(JUSRP10927); 江苏省博士后基金项目(1101021B).

作者简介: 熊伟丽(1978-), 女, 副教授, 从事过程控制与优化算法的研究; 徐保国(1950-), 男, 教授, 博士生导师, 从事过程控制系统、无线传感器网络的研究.

$$\min f = \sum_{i=1}^n F_i(P_i); \quad (1)$$

$$\text{s.t. } \sum_{i=1}^n P_i - P_L - P_D = 0, \quad (2)$$

$$P_i^{\min} \leq P_i \leq P_i^{\max}. \quad (3)$$

其中: P_i 为第 i 个发电机的发电功率, $F_i(P_i)$ 为第 i 个发电机的发电成本, n 为发电机的数量, P_L 为系统的传输损耗, P_D 为系统的电力需求总量, P_i^{\min} 和 P_i^{\max} 分别为第 i 个发电机最小和最大输出功率.

理想情况下, 一个发电机的发电成本可描述为

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i, \quad (4)$$

其中 a_i, b_i, c_i 为第 i 个发电机的相关常数. 在实际问题中, 需要考虑一些阀点效应的影响. 此时, 发电成本可描述为

$$\tilde{F}_i(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i \sin(f_i(P_i^{\min} - P_i))|, \quad (5)$$

其中 e_i, f_i 为与第 i 个发电机阀点效应的相关常数.

同时, 使用不同的燃料, 电机的输出功率和发电成本也会有明显的差别. 这种情况下, 一个发电机的发电成本可描述为

$$\tilde{F}_i(P_i) = \begin{cases} a_{i,1} P_i^2 + b_{i,1} P_i + c_{i,1}, & \text{if } P_i^{\min} \leq P_i < P_{i,1}; \\ a_{i,2} P_i^2 + b_{i,2} P_i + c_{i,2}, & \text{if } P_{i,1} \leq P_i < P_{i,2}; \\ \vdots \\ a_{i,k} P_i^2 + b_{i,k} P_i + c_{i,k}, & \text{if } P_{i,k-1} \leq P_i \leq P_i^{\max}. \end{cases} \quad (6)$$

对应的, 同时考虑多种燃料和阀点效应, 发电机的发电成本可以描述为

$$\tilde{F}_i(P_i) = \begin{cases} a_{i,1} P_i^2 + b_{i,1} P_i + c_{i,1} + \\ |d_{i,1} \sin(e_{i,1}(P_i^{\min} - P_i))|, & \text{if } P_i^{\min} \leq P_i < P_{i,1}; \\ a_{i,2} P_i^2 + b_{i,2} P_i + c_{i,2} + \\ |d_{i,2} \sin(e_{i,2}(P_i^{\min} - P_i))|, & \text{if } P_{i,1} \leq P_i < P_{i,2}; \\ \vdots \\ a_{i,k} P_i^2 + b_{i,k} P_i + c_{i,k} + \\ |d_{i,k} \sin(e_{i,k}(P_i^{\min} - P_i))|, & \text{if } P_{i,k-1} \leq P_i \leq P_i^{\max}. \end{cases} \quad (7)$$

从式(5)~(7)可以看出, ELD问题有很强的非线性, 难以使用传统数学方法求解. 本文尝试使用ABC算法求解考虑阀点效应和多燃料特性的ELD问题.

3 人工蜂群算法

ABC算法^[8]是受蜜蜂采蜜行为启发而提出的一种种群搜索算法. 该算法将问题求解看作蜜蜂寻找蜜源的过程, 问题解得好坏对应于蜜源的质量. 在搜索中, 该算法模仿3种蜜蜂的搜索行为: 雇佣蜂、旁观蜂和侦查蜂. 算法中, 每个雇佣蜂搜索一个蜜源并分享蜜

源信息给旁观蜂; 旁观蜂根据雇佣蜂提供的信息概率选择一个较好的蜜源进行探索; 侦查蜂随机地探索新的蜜源.

在搜索过程中, 算法首先随机生成一些蜜源 $\mathbf{X}^0 = [\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_N^0]$, 其中 N 为蜜源数目. 每个蜜源 $\mathbf{x}_i^0 = [x_{i,1}^0, x_{i,2}^0, \dots, x_{i,d}^0]$ 对应问题的一个解, 其中 d 为问题维数. 在搜索的每一代, 首先派出雇佣蜂对每个蜜源进行探索, 如果能得到更优的蜜源, 则替换当前蜜源, 否则保留原蜜源; 然后旁观蜂根据每个蜜源的备选概率 p_i 选择一个蜜源进行探索. 如果一个蜜源经过一定数目 limit 次探索后没有明显改进, 则派出侦查蜂随机寻找一个蜜源将其替换.

在探索一个蜜源 i 时, 雇佣蜂和旁观蜂采用相同的探索方式: 随意选择一个附近的蜜源 k , 并按下式探索当前蜜源的邻域:

$$v_{i,j}^{t+1} = \begin{cases} x_{i,s}^t + u(x_{k,s}^t - x_{i,s}^t), & \text{if } j = s; \\ x_{i,j}^t, & \text{otherwise.} \end{cases} \quad (8)$$

其中: v_i^{t+1} 为探索蜜源, x_k^t 为邻域中的任一蜜源, s 为任意选择的一个维度, u 为 -1 和 1 之间均匀分布的随机数. 如果 v_i^{t+1} 的质量优于 x_i^t , 则替换当前蜜源; 否则, 保留原蜜源.

旁观蜂根据雇佣蜂提供的蜜源信息, 以轮盘赌的方式选择一个较好的蜜源进行探索. 每个蜜源被选择的概率为

$$p_i = \text{fit}_i / \sum_{i=1}^N \text{fit}_i. \quad (9)$$

其中: fit_i 为第 i 个蜜源的适应值, N 为蜜源数目.

如果一个蜜源经过一定次数 limit 的探索后没有得到改进, 则认为对应的蜜源已经枯竭, 对应于问题局部极小. 此时, 派出侦查蜂在搜索空间随机生成一个蜜源替换当前蜜源, 即

$$x_{i,j}^{t+1} = x_j^{\min} + \text{rand}(\) \times (x_j^{\max} - x_j^{\min}). \quad (10)$$

其中: x_j^{\min} 和 x_j^{\max} 分别表示问题第 j 维的最小值和最大值, $\text{rand}(\)$ 表示 0 和 1 之间均匀分布的随机数.

ABC算法的整体流程如下:

Step 1: 按照式(10)生成 N 个初始蜜源.

Step 2: 评价每个蜜源, 计算蜜源质量.

Step 3: 雇佣蜂按照式(8)探索每个蜜源.

Step 4: 按照式(9)计算每个蜜源的备选概率.

Step 5: 旁观蜂以轮盘赌的方式选择一个蜜源并按照式(8)进行探索.

Step 6: 如果存在 limit 次探索未能改进的蜜源, 则派出侦查蜂对其重新初始化.

Step 7: 如果满足终止条件, 则输出已知最优解; 否则, 返回 Step 3.

4 差分蜂群算法

ABC 算法主要模仿 3 种搜索方式: 雇佣蜂、旁观蜂和侦查蜂。在算法中, 雇佣蜂对每个蜜源都进行探索, 负责全局搜索, 寻找有潜力的区域; 旁观蜂选择较好的蜜源进行探索, 负责局部搜索, 使得优秀的蜜源有更多的被探索机会; 侦查蜂对长期不能改进的蜜源进行初始化, 帮助算法跳出局部极小。这 3 种搜索方式相互协作, 有利于平衡算法的全局探索和局部搜索能力, 以便更好地找到最优解。

在文献 [8] 中, 雇佣蜂和旁观蜂都采用式 (8) 进行探索, 每次选择附近的一个蜜源改变当前蜜源的一个维度。旁观蜂采用这种方式可以有效地探索当前蜜源的邻域, 但雇佣蜂采用这种方式则不利于进行全局搜索。同时, 由于每次只改变蜜源的一个维度, 进化的速度很慢, 特别是在处理高维问题时效率特别低。此外, 由于雇佣蜂和旁观蜂的探索方式相同, 很容易产生重复搜索, 造成计算的浪费。在差分进化算法 [9] 中, 将另外两个个体的差分确定下一步搜索的位置。这种方式一方面能将步长和种群分布相适应, 提高步长的自适应性; 另一方面, 对于规模为 N 的种群, 能确定 $N * (N - 1)$ 种搜索方向, 有效地提高搜索的多样性。此外, 差分进化算法中的交叉操作能在选择几个维度的同时进行变异, 从而能更广泛地搜索解空间, 并避免计算量的浪费。为此, 本文采用差分进化的操作改进雇佣蜂的搜索方式, 即

$$v_{i,j}^{t+1} = \begin{cases} x_{i,j}^t, & \text{if rand}() \leq \text{cr}, j \neq \text{sn}; \\ x_{r_1,j}^t + F(x_{r_2,j}^t - x_{r_3,j}^t), & \text{otherwise.} \end{cases} \quad (11)$$

其中: $j = 1, 2, \dots, n$ 代表解的每一维; r_1, r_2, r_3 为随机选择的 3 个蜜源并且与 i 不同; F 为缩放因子, 取 0 和 1 之间的实数; cr 为选择概率; sn 为任意选择的一个维度, 用于保证至少有一维被改变。

由于经济负荷分配问题是一个约束优化问题, 使用智能优化算法求解需要配合适当的约束处理方式, 本文提出一种有效的修复机制保证解的可行性。对于新生成的蜜源 $v_i^{t+1} = [v_{i,1}^{t+1}, v_{i,2}^{t+1}, \dots, v_{i,d}^{t+1}]$, 以及式 (3) 的边界约束, 采用以下方式:

$$v_{i,j}^{t+1} = \begin{cases} 2x_j^{\max} - v_{i,j}^{t+1}, & \text{if } v_{i,j}^{t+1} > x_j^{\max}; \\ 2x_j^{\min} - v_{i,j}^{t+1}, & \text{if } v_{i,j}^{t+1} < x_j^{\min}; \\ v_{i,j}^{t+1}, & \text{otherwise.} \end{cases} \quad (12)$$

其中: $v_{i,j}^{t+1} (j = 1, 2, \dots, d)$ 为蜜源的任意一维。对式 (2) 描述的等式约束, 本文的修复操作如下:

Step 1: 按照下式计算误差 ΔP :

$$\Delta P = P_L + P_D - \sum_{i=1}^n P_i. \quad (13)$$

Step 2: 如果 $\Delta P = 0$, 则修复完成; 否则, 进入下一步。

Step 3: 按照下式任选一维 $v_{i,s}^{t+1}$:

$$v_{i,s}^{t+1} = x_{i,s}^{t+1} + \Delta P. \quad (14)$$

Step 4: 按下式检查边界并修正:

$$\begin{cases} v_{i,s}^{t+1} = x_s^{\max}, \Delta P = v_{i,s}^{t+1} - x_s^{\max}, & \text{if } v_{i,s}^{t+1} > x_s^{\max}; \\ v_{i,s}^{t+1} = x_s^{\min}, \Delta P = v_{i,s}^{t+1} - x_s^{\min}, & \text{if } v_{i,s}^{t+1} < x_s^{\min}; \\ \Delta P = 0, & \text{otherwise.} \end{cases} \quad (15)$$

其中 s 为任意选择的一个维度。

Step 5: 转入 Step 2。

基于以上操作, DABC 算法的流程如下:

Step 1: 按照式 (10) 生成 N 个初始蜜源。

Step 2: 修复每个初始蜜源并评价。

Step 3: 雇佣蜂以式 (11), (12) 探索每个蜜源。

Step 4: 按照式 (9) 计算每个蜜源被选择概率。

Step 5: 旁观蜂以轮盘赌选择一个蜜源并按照式 (8) 进行探索。

Step 6: 如果存在 limit 次探索未能改进的蜜源, 则派出侦查蜂对其重新初始化。

Step 7: 如果满足终止条件, 则输出已知最优解; 否则返回 Step 3。

与人工蜂群算法相比, 本文算法中的雇佣蜂采用差分变异和交叉的方式进行搜索, 能同时在多个维度进行更多多样性的搜索, 提高了搜索的多样性和搜索效率。此外, 雇佣蜂和旁观蜂采用不同的行为方式, 提高了搜索形式的多样性。和差分进化算法相比, 本文算法采用了 3 种不同的搜索机制, 提高了搜索的多样性; 同时, 轮盘赌机制能使优良的个体获得更多的进化机会, 从而提高计算的效率。为了验证该算法的性能, 本文将差分蜂群算法用于一些典型的电力系统负荷分配问题, 并和国际权威期刊的算法进行了比较。

5 实验仿真

5.1 考虑阀点效应的电力负荷分配

本节将 DABC 算法用于优化考虑阀点效应的电力负荷分配问题。首先考虑 13 维 ELD 问题。该系统的总电量需求为 1800 MW, 每个电机的参数如文献 [4] 所示。对于此问题, 设置 ABC 算法的参数为: 种群大小为 $\text{NP} = 40$, 最大进化代数为 $G_{\max} = 600$, 最大有效探索次数 $\text{limit} = 104$; DABC 算法采用和 ABC 算法相同的基本参数, 并设置 $F = 1$, $\text{cr} = 0.8$, 算法独立运行 100 次。统计算法的最优解 (Best), 平均解 (Mean), 最差解 (Worst), 标准差 (Std) 与函数评价次数 (NFE) 的结果如表 1 所示, 进化过程曲线如图 1 所示。表 2 列出了 DABC 算法找到的最优解。

表 1 13 维 ELD 问题的统计结果

算法	Best	Mean	Worst	Std	NFE
DABC	17960.366 1	17960.853 5	17969.348 6	1.61	2.4E4
ABC	17961.638 5	17967.888 0	17977.190 8	3.71	2.4E4

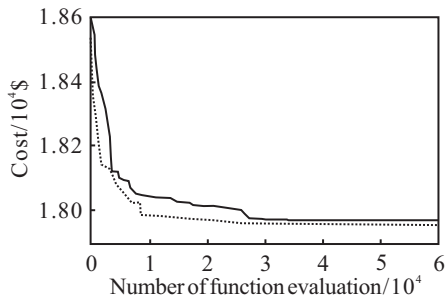


图 1 13 维 ELD 问题的搜索过程比较

表 2 13 维 ELD 问题的最优解

序号	功率	序号	功率	序号	功率
1	628.318 5	6	109.866 6	11	40.000 0
2	149.599 7	7	109.866 6	12	55.000 0
3	222.749 1	8	109.866 6	13	55.000 0
4	109.866 6	9	109.866 6	$\sum P_i$	1 800.000 0
5	60.000 0	10	40.000 0	f	17 960.366 1

由表 1 可以看出, 无论是最优解、平均解、最差解还是标准差, DABC 算法都明显优于 ABC 算法. 由图 1 可知, DABC 算法的进化速度都明显快于 ABC 算法. 这些表明, 与 ABC 算法相比, DABC 算法具有更强的全局搜索能力和更快的收敛速度.

表 3 对比了 DABC 算法和国内外期刊上的最近提出的几种算法, 包括混合 SQP 的基于完全学习的粒子群 (CPSO) 算法^[5]、双种群混合差分进化 (DHDE) 算法^[7]、差分进化 (DE) 算法^[10]、改进的差分进化 (IDE) 算法^[11]、量子粒子群 (QPSO) 算法^[12]、协调聚集粒子群 (CAPSO) 算法^[13]和混合细菌觅食 (HBF) 算法^[14]. 由表 2 易知, 在最优解、平均解、最差解和标准差方面, DABC 算法是最好的. 同时, 该算法的计算量也比较小, 这表明该算法更能快速有效地求解问题.

表 3 求解 13 维 ELD 问题的算法比较

方法	Best	Mean	Worst	Std	NFE
DHDE ^[7]	17 979	18 131	18 239	62.56	2.0E4
CPSO ^[5]	17 973.12	18 005.05	18 069.35	23.81	2.4E4
QPSO ^[12]	17 969.01	17 075.11	NA	NA	4.0E5
IDE ^[11]	17 961.944 0	17 974.686 9	18 061.411 0	20.31	2.5E4
DE ^[10]	17 963.83	17 965.48	17 975.36	NA	9.4E4
HBF ^[14]	17 960.499 8	17 969.856 9	NA	2.05	-
CAPSO ^[13]	17 960.37	17 967.94	17 978.14	NA	-
DABC	17 960.366 1	17 962.853 5	17 969.348 6	1.61	2.4E4

为了进一步检验算法在高维问题上的性能, 本节使用 40 维 ELD 问题进行测试. 该问题总电量需求为 10 500 MW, 每个机组的参数如文献 [4] 所示. 对于此

问题, 设置 ABC 算法的参数为 $NP=50, G_{max}=2400, limit=240$; DABC 算法采用和 ABC 算法相同的参数设置, 并设置 $F=0.3, cr=0.9$. 算法独立运行 100 次的统计结果如表 4 所示, 进化过程曲线如图 2 所示.

表 4 40 维 ELD 问题的统计结果

算法	Best	Mean	Worst	Std	NFE
DABC	121 403.535 8	121 422.835 8	121 452.674 9	18.78	1.2E5
ABC	121 411.282 4	121 475.619 4	121 561.029 7	46.52	1.2E5

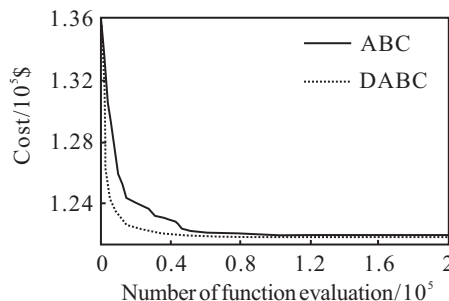


图 2 40 维 ELD 问题的搜索过程比较

从表 4 可以看出, 在相同的参数设置下, DABC 算法的统计性能明显优于 ABC 算法. 同样由图 2 可知, DABC 算法的收敛速度明显快于 ABC 算法. 这进一步表明 DABC 算法与 ABC 算法相比, 具有更强的全局搜索能力、更快的收敛速度和更好的鲁棒性.

表 5 对比了 DABC 算法和国内外期刊最近提出的一些算法, 包括 CPSO^[6]、DHDE^[7]、DE^[10]、IDE^[11]、QPSO^[12]、CAPSO^[13]、HBF^[14]和改进的粒子群 (IPSO) 算法^[15]. 由表 5 可以看出, DABC 算法能找到所有算法中的最优解. 就平均解而言, DABC 算法和 DE 算法性能相同, 好于其他所有算法. 在最差解方面, DABC 算法比 DE 算法稍差, 但也好于其他几种算法. 和 DE 算法、IPSO 算法相比, DABC 算法计算量较小. 这表明 DABC 算法是求解该问题的有效算法.

表 5 求解 40 维 ELD 问题的算法比较

算法	Best	Mean	Worst	Std	NFE
DHDE ^[7]	122 177	122 702	123 102	228.02	2.0E4
CPSO ^[5]	121 515.8	121 677.2	121 820.4	62.80	-
HBF ^[14]	121 423.639 7	121 814.946 5	NA	124.88	-
QPSO ^[12]	121 448.21	122 225.07	NA	NA	4.0E5
IDE ^[11]	121 423.401 3	121 526.733 0	121 696.986 8	54.86	2.5E4
DE ^[10]	121 416.29	121 422.72	121 431.47	NA	2.4E5
CAPSO ^[13]	121 413.20	121 428.14	121 453.56	NA	-
IPSO ^[15]	121 403.536 2	121 445.326 9	121 525.493 4	32.49	3.0E5
DABC	121 403.535 8	121 422.835 8	121 452.674 9	18.78	1.2E5

5.2 同时考虑阀点效应和多燃料的经济负荷分配

本节使用同时考虑阀点效应和多燃料情况下的 ELD 问题, 采用文献 [18] 中的实例. 此问题中含有 10 个电机, 总电量需求为 2 700 MW. 对此, 设置 ABC 算法的参数为: $NP=20, G_{max}=1000, limit=20$; DABC 算法采用和 ABC 算法相同的参数, 并设置 $F=$

0.5, $cr=0.3$. 算法独立运行 100 次的统计比较结果如表 6 所示, 算法进化过程比较如图 3 所示. 表 7 列出了 DABC 算法找到的最优解.

表 6 考虑阀点效应和多燃料的 ELD 问题统计比较

方法	Best	Mean	Worst	Std	Time
DABC	623.8266	623.8266	623.8267	1.97E-5	2.0E4
ABC	623.8271	623.8330	623.8395	2.87E-3	2.0E4

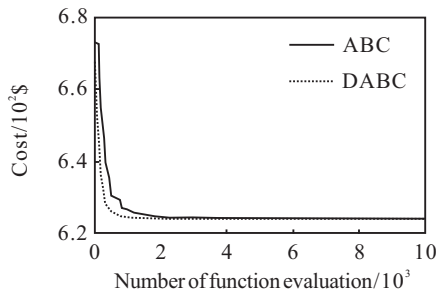


图 3 考虑阀点效应和多燃料的 ELD 问题的搜索过程

表 7 考虑阀点效应和多燃料的 ELD 问题的最优解

序号	功率	序号	功率	序号	功率
1	218.5940	5	279.9345	9	426.5883
2	211.7117	6	239.6394	10	275.8686
3	280.6571	7	287.7275	$\sum P_i$	2700.0000
4	239.6394	8	239.6394	f	623.8266

从表 6 可以看出, DABC 算法能找到比 ABC 算法更好的最优解, 恒定地找到最优解附近, 这样算法的方差非常小, 甚至 DABC 算法的最差解比 ABC 算法的最优解还好. 从图 3 可以看出, DABC 算法比 ABC 算法收敛速度更快.

表 8 对比了 DABC 算法和国际期刊近期提出的几种算法的比较. 从表 8 可以看出, DABC 算法是 6 种算法中性能最好的. 虽然 IPSO 算法能得到和 DABC 算法一样的最优解, 但其平均解和最差解相对较差. 其他 4 种算法的结果无论是最优解、平均解还是最差解都明显不如 DABC 算法. 综上所述, DABC 算法能快速、有效地求解该问题.

表 8 求解考虑阀点效应和多燃料的 ELD 问题的算法比较

方法	Best	Mean	Worst	Std	NFE
IGA ^[18]	624.5178	625.8692	630.8705	NA	4.5E5
NPSO ^[19]	624.1273	624.9985	626.9981	NA	-
APSO ^[20]	624.0145	624.8185	627.3049	NA	-
HPSO ^[21]	623.9488	624.0816	624.2930	NA	9.0E4
IPSO ^[15]	623.8266	623.8273	623.8291	5.0E-4	3.0E5
DABC	623.8266	623.8266	623.8267	1.2E-5	2.0E4

6 结 论

针对电力系统经济负荷分配问题, 本文提出了一种差分蜂群算法. 该算法通过引入差分进化操作提高了其全局搜索能力和收敛速度, 并通过一种修复操作保证了新解的可行性. 该算法在带有阀点效应和多燃料等特征的电力负荷分配问题上进行了测试. 与国际

权威期刊的相关算法相比, 本文算法能获得更好的结果. 这表明, 本文算法是求解电力系统经济负荷分配问题的一种有效算法. 进一步的工作包括考虑动态和多目标的电力负荷分配问题, 以及将本文算法推广到其他工程约束优化问题.

参考文献(References)

- [1] Wood J, Wollenberg B F. Power generation, operation, and control[M]. 2nd ed. New Jersey: Wiley press, 1984.
- [2] 王承民, 郭志忠, 于尔铿. 电力市场中一种基于动态规划法经济负荷分配算法[J]. 电力系统自动化, 2000, 24(21): 19-22, 26.
(Wang C M, Guo Z Z, Yu E K. An economic generation scheduling method based on dynamic programming for power market[J]. Automation of Electric Power Systems, 2000, 24(21): 19-22, 26.)
- [3] 何大阔, 王福利, 毛志忠. 基于改进遗传算法的电力系统经济负荷分配[J]. 控制与决策, 2007, 22(2): 230-232, 237.
(He D K, Wang F L, Mao Z Z. Economic dispatch of power systems based on an improved genetic algorithm[J]. Control and Decision, 2007, 22(2): 230-232, 237.)
- [4] Sinha N, Chakrabarti R, Chattopadhyay P K. Evolutionary programming techniques for economic load dispatch[J]. IEEE Trans on Evolutionary Computation, 2003, 7(1): 83-94.
- [5] 王瑜, 李斌, 袁博. 混合 SQP 的基于完全学习的粒子群优化算法在电力系统中经济分配问题的应用[J]. 中国科学, 2010, 40(3): 403-411.
(Wang Y, Li B, Yuan B. Hybrid of comprehensive learning particle swarm optimization and SQP algorithm for large scale economic load dispatch optimization of power system[J]. Science China, 2010, 40(3): 403-411.)
- [6] 蒙文川, 邱家驹, 卞晓猛. 电力系统经济负荷分配的人工免疫混沌优化算法[J]. 电网技术, 2006, 30(23): 41-44.
(Meng W C, Qiu J J, Bian X M. Artificial immune algorithm integrated with chaotic optimization for economic dispatch of power system[J]. Power System Technology, 2006, 30(23): 41-44.)
- [7] 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 population[J]. Control and Decision, 2009, 24(8): 1156-1160.)
- [8] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony(ABC) algorithm[J]. J of Global Optimization, 2007, 39(3): 459-471.