

文章编号:1005-0523(2018)03-0134-9

电力系统动态环境经济调度模型研究

左丽霞,余 愿,孙惠娟

(华东交通大学电气与自动化工程学院,江西 南昌 330013)

摘要:电力系统动态环境经济调度在节能减排中具有举足轻重的地位,以经济成本和环境成本最小为优化目标建立模型,其经济成本计及了常规火电机组调点效应引起的能耗成本。在等式约束条件中考虑了网损,采用解方程的方法来处理模型中复杂的约束关系。针对微分进化算法多样性低导致的早熟问题,采用改进型微分进化算法求解模型,并在保证原有计算精度的前提下,利用 Matlab 并行计算的能力增强微分进化算法,提高算法的计算效率。经典 10 机组系统算例仿真结果表明,与传统微分进化算法相比,改进后的微分进化算法不仅具有更佳的全局搜索能力,还可以为决策者提供更快、更优的选择。

关键词:动态经济调度;改进型微分进化;并行计算

中图分类号: TM73

文献标志码: A

随着社会经济的不断发展以及电力系统的急剧增大,只考虑发电机组出力约束的静态经济调度(economic dispatch, ED)已不能满足电力系统安全稳定的要求。在 ED 基础上计及各时间段相互影响的动态经济调度(dynamic economic dispatch, DED)由此产生。DED 考虑了机组出力爬坡/滑坡速率,其调度策略更符合实际,但难度也更大。近年来,随着环境污染和大气污染的进一步加重,世界各国都制定了与节能减排相关的法律法规^[1],因此兼顾常规火电厂污染气体排放的动态环境经济调度(dynamic economic emission dispatch, DEED)应运而生。

DEED 既要考虑经济又要考虑环境,是一个具有非线性、强约束的多目标优化问题^[2]。目前,国内外学者对 DEED 的研究主要集中在多目标函数的求解,算法上虽然取得了一定的研究成果^[3-7],但总体存在以下问题:目标函数和约束条件不完备,目标函数中不计调点效应引起的成本,约束条件中等式约束不计网损,最优解集中个体密集、分布不均匀、计算速度慢等问题。本文将研究一种改进型微分进化算法,用于快速有效的求解 DEED 问题。

1 DEED 建模

DEED 既要考虑经济又要考虑环境,因此选择发电成本最小和环境成本最小作为 2 个相互竞争的目标函数,数学模型如下:

$$\begin{aligned} & \min\{f_1(x), f_2(x)\} \\ & \text{s.t. } g(x)=0, h(x)\leq 0 \end{aligned} \quad (1)$$

式中: $f_1(x)$ 为发电成本目标函数; $f_2(x)$ 为环境成本目标函数; $g(x)$ 为等式约束; $h(x)$ 为不等式约束。

1.1 目标函数

1) 机组发电成本函数。在满足负载和运行约束的前提下,合理地分配各发电机组的出力以使整个调度期

收稿日期:2017-09-14

基金项目:江西省教育厅科学技术研究项目(GJJ160526,GJJ160525);江西省自然科学基金项目(20171BAB202001);华东交通大学科研基金资助项目(15DQ05)

作者简介:左丽霞(1982—),女,讲师,硕士,研究方向为电力系统智能优化算法。

间内发电总成本最小^[8]。当计及汽轮机的阀点效应时,目标函数表达式为

$$\min f_1 = \sum_{t=1}^T \sum_{i=1}^H [F_i(P_{it}) + E_i(P_{it})] \quad (2)$$

$$F_i(P_{it}) = a_i + b_i P_{it} + c_i F_i P_{it}^2 \quad (3)$$

$$E_i(P_{it}) = |d_i \sin[e_i(P_i^{\min} - P_{it})]| \quad (4)$$

式中: $F_i(P_{it})$ 和 $E_i(P_{it})$ 分别为发电机组 i 在 t 时段的发电成本和由阀点效应引起的能耗成本; H 为常规火电机组的台数; T 为调度总的时段数; P_{it} 为发电机组 i 在 t 时段的有功出力, P_i^{\min} 为发电机组 i 的最小出力; a_i, b_i 和 c_i 为机组 i 的燃料费用系数; d_i, e_i 为机组 i 的阀点效应系数。

2) 污染气体排放量函数。整个调度期间合理地分配各发电机组的出力以使污染气体排放总量最小^[7],综合排污特性可表示为

$$\min f_2 = \sum_{t=1}^T \sum_{i=1}^H [\alpha_i + \beta_i P_{it} + \gamma_i P_{it}^2 + \eta_i \exp(\delta_i P_{it})] \quad (5)$$

式中: $\alpha_i, \beta_i, \gamma_i, \eta_i$ 和 δ_i 为机组 i 污染排放的系数。

1.2 约束条件

1) 系统的功率平衡等式约束

$$\sum_{i=1}^H P_{it} = P_{Dt} + P_{Lt} \quad t=1, 2, \dots, T \quad (6)$$

式中: P_{Dt} 和 P_{Lt} 分别为 t 时段的系统负载预测值和系统网损值。

系统网损 P_{Lt} 的近似值一般采用简化的网损系数矩阵 \mathbf{B} 来进行计算:

$$P_{Lt} = \sum_{i=1}^H \sum_{j=1}^H P_{it} B_{ij} P_{jt} \quad (7)$$

式中: B_{ij} 为矩阵 \mathbf{B} 第 i 行第 j 列分量。

2) 发电机组的出力不等式约束

$$P_i^{\min} \leq P_{it} \leq P_i^{\max} \quad i=1, 2, \dots, H; t=1, 2, \dots, T \quad (8)$$

式中: P_i^{\min} 和 P_i^{\max} 分别为机组 i 的最小出力和最大出力。

3) 发电机组的爬坡/滑坡速率不等式约束

$$-D_{Ri} \leq P_{it} - P_{i(t-1)} \leq U_{Ri} \quad i=1, 2, \dots, H; t=1, 2, \dots, T \quad (9)$$

式中: $P_{i(t-1)}$ 为发电机组 i 在 $t-1$ 时段的有功出力; U_{Ri} 和 D_{Ri} 为发电机组 i 的爬坡和滑坡速率。

2 DEED 复杂约束条件处理

DEED 的强约束性和非线性特性使得模型求解变得相当困难,高效快速求解非线性动态优化问题的核心是寻求相关方法处理约束条件。

2.1 等式约束处理

本模型中计及网损的等式约束式(6)处理起来相对比较困难,将采用求解方程的方法来处理等式约束。具体思路如下。

1) 将一台机组的出力作为未知量。例如将第一台机组的出力作为未知量 x ,在机组出力范围内随机产生 $N-1$ 台机组的出力,带入等式约束式(6)中,则 N 台机组的出力组合自然满足了功率等式约束条件。由于功率的等式约束计及网损,所以对求解的方程为一元二次方程。

$$Ax^2 + Bx + C = 0 \quad (10)$$

2) 整理方程。带入参数可以将等式约束式(6)化为以下形式:

$$x + \sum_{i=2}^N P_{i1} = P_{D1} + \left(\sum_{i=2}^N \sum_{j=2}^N P_{ij} B_{ij} P_{ji} + \sum_{j=2}^N x B_{ij} P_{ji} + x \sum_{i=2}^N x P_{ji} B_{ij} + x B_{11} x \right) \quad (11)$$

化简后可以求得一元二次方程的系数如下:

$$\left. \begin{aligned} A &= B_{11} \\ B &= 2 \sum_{i=2}^N P_{i1} B_{i1} - 1 \\ C &= P_{D1} + \sum_{i=2}^N \sum_{j=2}^N P_{ij} B_{ij} P_{ji} - \sum_{i=2}^N P_{i1} \end{aligned} \right\} \quad (12)$$

3) 求解方程。根据公式(12)中的系数,即可求解一元二次方程,当方程有实数解,且解在待求机组出力范围之内,才为可行解,否则即为不可行解,可以在目标函数中对应加上对应惩罚项^[9],淘汰掉不可行的解。

2.2 不等式约束处理

不等式约束式(8)和(9)处理起来比较简便,只需要令相关违反约束的变量变为边界值即可。针对不同时段,按式(13)设置常规机组的出力上下限,再按式(14)对机组出力进行调整。

$$P_{it}^{\min} = \begin{cases} P_i^{\min} & t=1 \\ \max(P_i^{\min}, P_{i,t-1} - D_{Ri}) & t>1 \end{cases} \quad (13)$$

$$P_{it}^{\min} = \begin{cases} P_i^{\min} & t=1 \\ \max(P_i^{\min}, P_{i,t-1} - U_{Ri}) & t>1 \end{cases}$$

$$P_{it} = \begin{cases} P_{it}^{\min} & P_{it} < P_{it}^{\min} \\ P_{it} & P_{it}^{\min} \leq P_{it} \leq P_{it}^{\max} \\ P_{it}^{\max} & P_{it} > P_{it}^{\max} \end{cases} \quad (14)$$

3 DEED 模型求解

3.1 微分进化算法

微分进化(differential evolution, DE)算法是 Rainer Storn 和 Kenneth Price 在 1995 年提出的,它运用实数编码的方式,进行选择、交叉和变异操作,具有寻优速度快、收敛精度高、全局搜索的功能好等优点。DE 算法寻优的特性要显著优于遗传算法、蚁群算法等其他的优化算法,而且算法简单、参数少、便于与其他算法结合,其最为常用的 DE/best 进化策略操作如下:

$$Y_i^{G+1} = X_{\text{best}}^G + F(X_{r_1}^G - X_{r_2}^G) \quad (15)$$

式中: Y_i^{G+1} 为进化操作所产生的中间个体; X_r^G 为第 G 代第 r 个个体向量, $i \neq r_1 \neq r_2 \neq \text{best}$, r_1, r_2 随机选取; F 为进化步长; X_{best}^G 为第 G 代最优个体。

DE 算法作为群智能进化算法的一种,存在的缺陷是:在解决复杂寻优问题时,随着进化后期个体多样性下降,式(15)中的差分项($X_{r_1}^G - X_{r_2}^G$)会趋于 0,从而导致进化停滞而陷入早熟。为解决群智能算法固有的早熟问题,更快、更准确地找出最优解集(Pareto 前沿),本文将改进 DE 算法。

3.2 改进型微分进化算法

利用化学计算中分子间相互作用的势能来改进 DE 算法,势能 V 的计算可用数学家 John Lennard-Jones 提出的 L-J 势能函数来描述,即

$$V = 4\varepsilon \left[\left(\frac{r_0}{r} \right)^{12} - \left(\frac{r_0}{r} \right)^6 \right] = 4\varepsilon \left(\frac{r_0}{r} \right)^{12} \left[1 - \left(\frac{r_0}{r} \right)^6 \right] \quad (16)$$

式中: r_0 是 $V=0$ 时的距离; ε 为势能阱深度系数。分子间距离 r 小于 r_0 时为斥力,且随着 r 的减小,势能 V 急剧增大,直至无穷大;而当 r 大于 r_0 时为引力,但 V 较小。

在进化过程中,当个体相似时可利用上述分子靠近斥力增大的原理自动维持种群多样性,从而避免寻优陷入早熟,实现算法的高效深度寻优。为将变异幅度控制在 r_{k0} 的合理范围内,设种群个体为 d 维向量, x_{ie} ,

x_i 为 x 的上下限值,参考式(16)将个体 X_i 与个体 X_j 的势能向量 $V_{i,j}$ 可描述为

$$\left. \begin{aligned} X_i &= [x_i^1, x_i^2, \dots, x_i^d], X_j = [x_j^1, x_j^2, \dots, x_j^d] \\ r_{i,j}^k &= \min(|x_{i,k} - x_{j,k}|, r_{k0}), r_{k0} = r_0(x_u^k - x_l^k) \\ v_{i,j}^k &= r_{k0} \cdot [1 - (r_{i,j}^k / r_{k0})^6], V_{i,j} = [v_{i,j}^1, v_{i,j}^2, \dots, v_{i,j}^d] \end{aligned} \right\} \quad (17)$$

式(17)中只考虑相似个体间的排斥关系,一旦个体间维差距 r^k 小于临界距离 r_{k0} ,排斥能 v^k 便迅速增大到 r_{k0} ,使个体间始终存在足量差异来保持种群多样性,从而有效提高算法的寻优深度。基于上述原理,将传统 DE/best 进化策略改进为如下改进型微分进化(modified differential evolution MDE)策略:

$$Y_i^{G+1} = X_{\text{best}}^G + F(X_{r1}^G - X_{r2}^G + V_{i,j}^G) \quad (18)$$

3.3 并行计算

利用群智能算法个体计算操作的相似性和计算机多核并行计算的优点,将改进型微分进化算法中的种群分解为多个子种群在计算单元(CPU核)同步并行计算,每个计算单元独立执行一个子程序。

在并程序的优化过程中,寻优效果的好坏主要取决于子程序间的数据通信力度(由通信间隔 gap 、个体通信规模 $pool$ 两方面描述),在保证并行程序寻优精度的前提下,如何平衡通信力度,提高数据的通信质量具有重要意义。

根据改进型微分进化算法的寻优特点,优化前期种群多样性高要保障足够的搜索广度,通信力度要小;而优化后期,应提高精度寻优效率及时分享优秀个体,此时通信力度反而要大。因此,随着寻优代数 i 的增加,数据通信力度不断提高,即通信间隔 gap 逐渐减小,个体通信 $pool$ 规模逐渐增大。为此,本文依据式(19)和式(20)对 gap 和 $pool$ 进行动态调整。其中, $I_iter \max$ 为总优化代数, x_NP 为每个核的种群规模, $round(\)$ 为四舍五入函数。 D_i 为随 i 的增大而降低的动态调整参数, gap 的最小间隔取 $I_iter \max / 100$, $pool$ 的最小通信规模取 $x_NP / 10$ 。

$$\left. \begin{aligned} D_i &= I_iter \max / i \\ gap &= round(D_i \cdot I_iter \max / 100) \end{aligned} \right\} \quad (19)$$

$$pool = round(x_NP / (10 \cdot D_i)) \quad (20)$$

并行改进型微分进化(parallel molecule differential evolution PMDE)算法的流程图如图1所示,假设计算机的核心数为 n (对应计算单元 $lab_1 \sim lab_n$),将规模为 I_NP 的种群按核心数 n 均分,分配给每个核的总种群规模为 x_NP 。下面以计算单元 lab_1 为例,具体实现步骤如下:

步骤1:总种群1非劣排序

对 lab_1 上分配的总种群1按目标函数值大小进行非劣排序,并计算拥挤距离。

步骤2:优选出母种群1

进化开始,采用锦标赛方法,并根据交配池的大小,从总种群1中优选出母种群1。

步骤3:改进型微分进化

母种群1经过改进型微分进化操作得到进化后的子种群1,再将总种群1和子种群1合并生成混合过渡种群1。

步骤4:动态调整通信间隔,并判断是否通信

根据式(19)计算第 i 代的通信间隔 gap ,并判断各 lab 间是否通信,若通信则启动 MATLAB 并行工具箱中的路障函数将各子程序都同步到第 i 代,执行步骤5;若不通信,则直接执行步骤6。

步骤5:计算通信池规模

当第 i 代通信时,根据式(20)计算此时的动态通信池规模 $pool$,并按锦标赛选择操作优选的优秀个体 x_{el} ,然后通过首尾循环通信方式进行各 lab 间的数据传递,完成通信。

步骤6:混合过渡种群1非劣排序

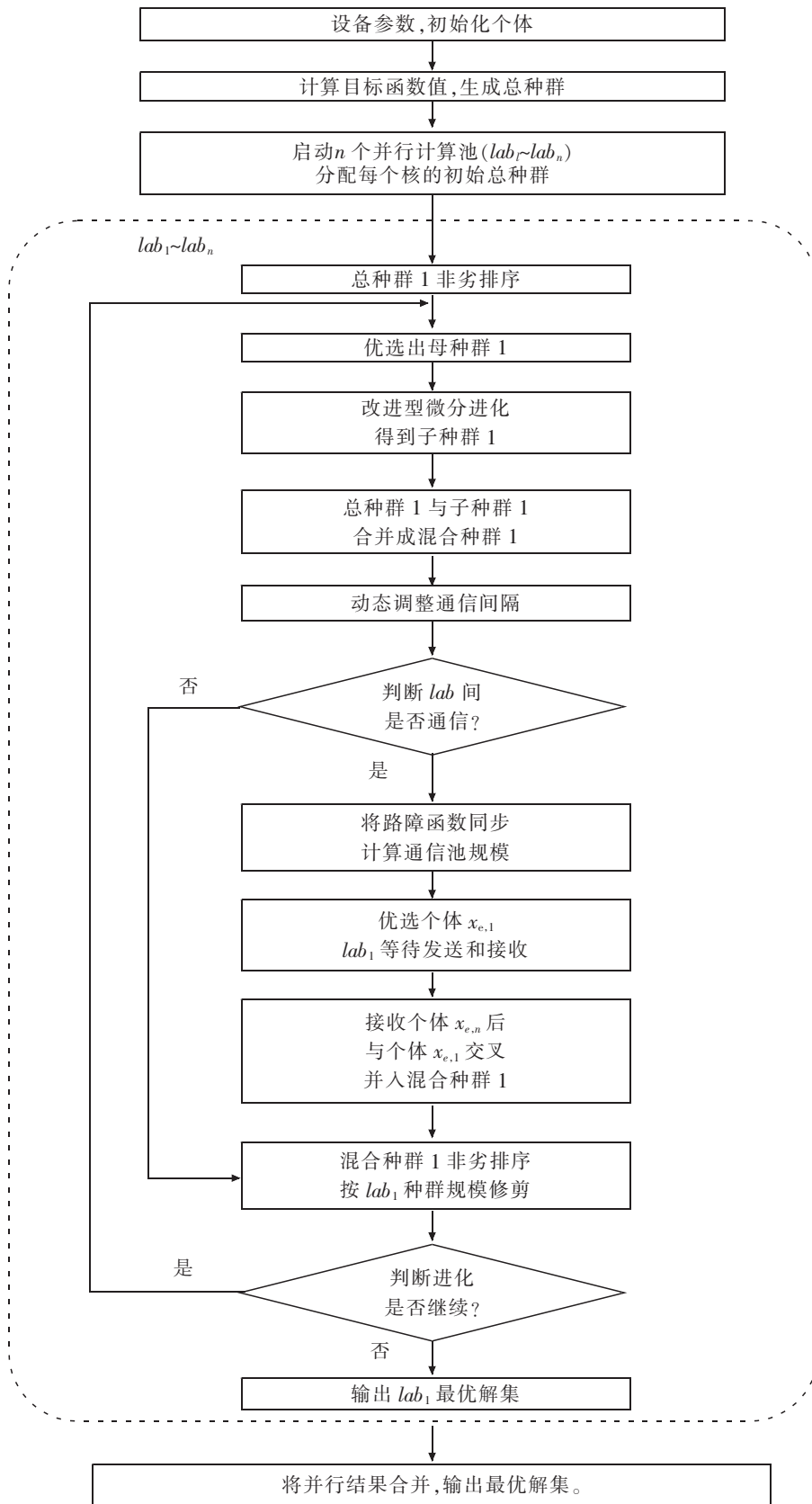


图1 PMDE 算法流程图
Fig.1 Flow chart of PMDE algorithm

对混合过渡种群 1 进行非劣排序操作,计算拥挤距离,并按 lab_1 总种群 1 的规模 x_{NP} 进行修剪。

步骤 7:判断进化是否继续

对迭代次数进行判断,若 $i \leq I_{iter\ max}$,则跳转至步骤 2,开始新的一次进化迭代;若判断条件不满足,则进化结束,输出 lab_1 的最优解集。将并行计算单元 $lab_1 \sim lab_n$ 的最优解集进行合并,消除重复解后即可输出 Pareto 最优解集。

加速比和并行效率是衡量并行算法性能的重要指标,其中加速比用来衡量并行算法对于串行算法的加速倍数,其定义为同一个任务在串行单核处理器运行时间(t_s)和并行多核处理器运行时间(t_p)的比率^[10]。

$$S_p = t_s / t_p \tag{21}$$

并行效率用来衡量单处理器计算能力被有效利用的比率^[11],理想情况下的加速 S_p 比应该等于处理器个数 P ,并行效率的数值等于 1,但实际情况 S_p 通常小于 P ,故并行效率为 0~1 之间取值,并行效率的表达式为

$$P_E = S_p / P \tag{22}$$

4 算例分析

4.1 系统描述及参数设置

采用 10 机组电力系统算例验证本文研究 PMDE 算法的有效性,并与 DE 算法、MDE 算法的结果进行比较。模型综合考虑了常规火电机组的阀点效应和系统的网络损耗。机组参数和负荷数据详见文献[7],调度时长为 24 h,即 24 个时段。仿真试验均在 Intel 酷睿 i7 四核 6700HQ 处理器 (2.60 GHz)、8 G 内存 64 位 Win10 操作系统上采用 MATLAB 编程实现,具体参数设置如下。

微分进化算法的参数:最大迭代次数 $I_{iter\ max}=5\ 000$,种群大小 $I_{NP}=480$,进化步长 $F=0.85$,改进型微进化的临界距离 $r_{k0}=0.001$ 。并行计算参数设置如下:计算机的核心数 $n=4$,可设置串行、双核并行、三核并行、四核并行 4 种模式。

4.2 仿真结果分析

1) DE 和 MDE 算法优化结果的对比。算法输出优化结果的对比如图 2 所示,其中图 2(a)和图 2(b)分别对应 DE 算法和 MDE 算法优化得到的 Pareto 最优解集。从图中可以看出,无论从 Pareto 最优前沿的完整性、非劣解分布的均匀性以及收敛特性来看,MDE 算法比 DE 算法都有明显的改进。

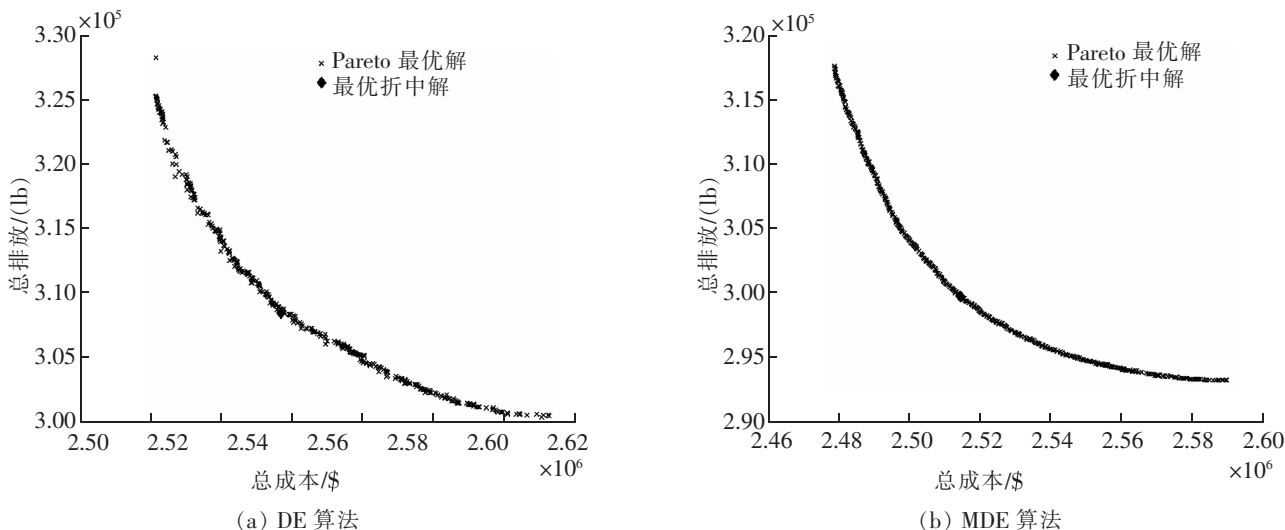


图 2 Pareto 最优解集对比
Fig.2 Comparison of Pareto optimal set

得到 Pareto 最优前沿后,采用基于模糊集理论的多属性决策技术,即可得出满意度函数值最大的最优折中解。两种算法的优化结果如表 1 所示,不论是对比极端解还是最优折中解,可以看出 MDE 比 DE 算法的经济性和环境性都要优,而且运行时间也短。可见,MDE 可以为决策者提供更优、更快的决策。

表 1 优化结果的对比

Tab.1 Comparison of optimization results

算法	运行时间/s	目标	总燃料费用/(10 ⁶ \$)	总排放量/(10 ¹⁰ lb)
DE	8 618.1	经济最优	2.518 621	3.282 410
		环境最优	2.613 938	3.004 130
		最优折中解	2.548 917	3.084 189
MDE	8 290.6	经济最优	2.478 724	3.175 918
		环境最优	2.589 846	2.931 437
		最优折中解	2.514 424	2.996 616

当 MDE 获得最优折中解时,各机组的动态有功出力如表 2 所示,24 时段系统的功率误差几乎都为 0,很好地满足了 DEED 功率平衡的等式约束。

表 2 最优折中解

Tab.2 Optimal compromise solution

时段	机组出力/MW										总出力/ MW	负荷/ MW	网损/ MW	功率偏 差/MW
	机组 1	机组 2	机组 3	机组 4	机组 5	机组 6	机组 7	机组 8	机组 9	机组 10				
1	152.209	136.424	113.480	116.321	122.329	117.464	88.026	119.867	46.620	42.845	1055.584	1036	19.584	8.492E-05
2	152.155	137.485	93.671	119.564	172.228	120.013	93.241	119.524	76.118	48.512	1132.511	1110	22.511	2.319E-05
3	150.295	135.286	154.589	121.857	191.341	159.075	122.758	118.829	79.790	52.728	1286.548	1258	28.548	3.482E-05
4	150.130	150.677	193.702	167.515	240.601	156.902	129.457	119.303	79.215	54.159	1441.661	1406	35.661	7.411E-06
5	150.065	219.115	193.639	179.072	240.043	158.347	128.300	118.678	78.823	54.009	1520.092	1480	40.092	8.757E-06
6	227.414	221.877	213.196	229.028	242.965	158.740	129.999	119.999	79.750	54.320	1677.288	1628	49.288	1.549E-05
7	229.096	223.118	264.188	251.934	242.608	159.979	129.997	119.997	79.997	54.969	1755.881	1702	53.881	4.122E-05
8	231.080	242.225	286.965	286.828	243.000	159.990	130.000	119.998	79.938	54.911	1834.936	1776	58.936	3.324E-05
9	300.433	302.759	304.110	299.814	243.000	160.000	130.000	119.929	80.000	54.965	1995.010	1924	71.010	1.362E-05
10	329.623	344.643	339.931	300.000	243.000	159.917	130.000	120.000	79.768	54.680	2101.563	2022	79.563	1.559E-05
11	379.569	386.424	340.000	300.000	243.000	160.000	130.000	120.000	80.000	54.894	2193.887	2106	87.887	1.660E-05
12	407.124	407.700	340.000	300.000	243.000	159.997	130.000	120.000	80.000	54.671	2242.492	2150	92.492	2.911E-05
13	358.371	374.102	338.637	300.000	242.991	159.744	130.000	119.521	78.802	54.339	2156.508	2072	84.508	3.780E-05
14	292.614	308.915	306.415	299.973	242.994	159.679	129.822	119.777	79.920	54.869	1994.978	1924	70.978	3.195E-05
15	226.549	280.487	278.799	262.226	242.985	159.740	129.809	119.998	79.938	54.789	1835.320	1776	59.320	2.703E-05
16	158.825	222.905	215.533	215.138	242.106	159.848	129.709	119.861	79.463	54.854	1598.242	1554	44.242	1.986E-05
17	150.092	218.361	188.179	186.925	233.649	158.594	129.936	119.847	79.993	54.484	1520.059	1480	40.059	1.022E-05
18	222.161	219.796	214.255	235.276	242.412	159.355	129.399	119.712	79.946	54.886	1677.197	1628	49.197	1.932E-05
19	227.407	249.494	293.017	277.508	242.842	159.852	129.996	119.997	79.899	54.968	1834.978	1776	58.978	3.344E-05
20	302.502	321.195	335.268	299.965	243.000	160.000	130.000	120.000	80.000	54.993	2046.924	1972	74.924	1.030E-05
21	296.087	299.837	313.085	299.998	241.959	159.975	129.971	119.966	79.054	54.984	1994.917	1924	70.917	2.487E-05
22	220.734	220.126	233.598	250.706	220.712	159.460	128.319	117.525	75.498	50.487	1677.163	1628	49.163	1.465E-05
23	150.177	140.598	162.882	203.109	176.923	158.529	125.396	114.350	77.961	53.944	1363.871	1332	31.871	9.411E-06
24	153.206	138.421	83.963	158.711	172.312	121.584	129.874	119.488	79.411	52.566	1209.537	1184	25.537	1.259E-05

2) MDE 和 PMDE 算法运行效率的对比。分别运行串行的 MDE 算法和并行的 PMDE 算法的程序,运行时间和优化结果的对比如表 3 所示,很明显并行算法的运行时间比串行明显大大减少。并行计算的核心数越多,每个核分的种群规模越小,通信次数是随总的迭代次数动态调整,而每次通信的个体数就是随着每个核分的种群规模来动态调整的。而且并行计算的核心数越多,CPU 利用率越高,在并行计算优化效果不变的前提下,算法运算的时间也对应大大减少,优化计算的效率,包括加速比和计算效率都显著提高。

表 3 算法运行情况的对比
Tab.3 Comparison of algorithm operation

运行方式	计算核心数/个	每核的种群个体数/个	通信次数/次	每次通信的个体数/个	CPU 利用率/%	运行时间/s	最优折中解总燃料成本/\$	最优折中解总排放量/(lb)	加速比	计算效率
单核串行 MDE	1	480	-	-	19	8 104.6	2 514 423.911 2	299 661.645 7	-	-
两核并行 PMDE	2	240	52	12~119	34	4 210.9	2 511 151.716 6	306 819.454 4	1.925	0.962 5
三核并行 PMDE	3	160	52	8~79	48	2 744.3	2 512 848.809 0	305 784.234 8	2.953	0.984 3
四核并行 PMDE	4	120	52	6~59	63	2 028.7	2 513 923.151 9	304 166.333 6	3.995	0.998 8

5 结束语

建立了电力系统动态环境经济调度的模型,以经济成本和环境成本最小为优化目标。其中,经济成本计及了常规火电机组阀点效应引起的能耗成本,等式约束条件中也考虑了网损,并采用解方程的方法很好地处理了模型中的等式约束条件。本文采用一种改进型微分进化 MDE 算法进行求解,很好地保证了种群的多样性,避免陷入早熟。还利用了 Matlab 并行计算的能力对算法进行增强,并行改进型微分进化 PMDE 算法可以进一步提高计算的速度和效率。通过仿真结果表明,与传统的微分进化算法相比,PMDE 算法具有更好的全局搜索能力,Pareto 最优解集分布也更加均匀,采用并行计算后,可为决策者提供更快、更优的选择。随着新能源在电力系统中的比重越来越大,以及能源互联网越来越深入的研究,计及可再生能源入网的动态环境经济调度必然成为新时期新阶段需要重点研究的工作。

参考文献:

- [1] 孟安波,李专. 采用多目标纵横交叉算法的电力系统动态环境经济调度[J]. 继电器,2016,44(2):109-115.
- [2] 江兴稳,周建中,王浩,等. 电力系统动态环境经济调度建模与求解[J]. 电网技术,2013,37(2):385-391.
- [3] 邓方,崔静,方浩,等. 基于改进粒子群优化的弹道并行求解算法[J]. 北京理工大学学报,2015,35(4):391-396.
- [4] BASU M. Fuel constrained economic emission dispatch using nondominated sorting genetic algorithm-II[J]. Energy,2014,78(4):649-664.
- [5] 刘刚,彭春华,相龙阳. 采用改进型多目标粒子群算法的电力系统环境经济调度[J]. 电网技术,2011,35(7):139-144.
- [6] 刘国静,韩学山,杨明,等. 电力系统运行协同的经济调度[J]. 中国电机工程学报,2014,34(16):2668-2675.
- [7] 杨柳青,林舜江,刘明波,等. 考虑风电接入的大型电力系统多目标动态优化调度[J]. 电工技术学报,2014,29(10):286-295.
- [8] 朱志键,王杰. 基于改进 NSGA-II 的电力系统动态环境经济调度[J]. 电力自动化设备,2017,37(2):176-183.
- [9] 傅军栋,邹欢,康水华. PSO-SVM 算法在智能建筑环境监控系统中的应用[J]. 华东交通大学学报,2016,33(1):121-127.
- [10] 孟晓丽,唐巍,刘永梅,等. 大规模复杂配电网三相不平衡潮流并行计算方法[J]. 继电器,2015,43(13):45-51.
- [11] 刘方. 基于大系统分解协调和多核集群并行计算的流域梯级水电中长期调度[J]. 中国电机工程学报,2017,37(9):2479-2490.

Research on Dynamic Economic Emission Dispatch Model of Power System

Zuo Lixia, Yu Yuan, Sun Huijuan

(School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: Dynamic economic emission dispatch of power system plays an important role in energy saving and emission reduction. In this study, the model is established by taking economic cost and environmental cost minimum as the optimization objectives. The energy cost caused by the valve point effect of the conventional thermal power unit is included in the economic cost. The network loss is considered in the equality constraint, and the solution equation is used to deal with the complex constraint relation in this model. In view of the premature diversity caused by the low diversity of differential evolution algorithm, the modified differential evolution algorithm is adopted to solve the model. Under the premise of ensuring the accuracy of the original calculation, the ability of Matlab parallel computing is used to enhance the differential evolution algorithm and improve the computational efficiency of the algorithm. The simulative results of the classical 10-unit test system show that compared with the traditional differential evolution algorithm, the parallel modified differential evolution algorithm not only has better global search ability, but also can provide a faster and better choice for decision-makers.

Key words: dynamic economic emission dispatch; modified differential evolution; parallel computing

勘误:2018年第2期张翀等人的论文《山区小城市非混行道路行程时间修正模型研究》中第68页公式(2)更正为

$$t(t)=t_0[1+\alpha_1[Q_{es}(t)/C(t)]^{\beta_1}+\alpha_2 Q_b(t)^{\beta_2}] \quad (2)$$