

文章编号:1671-6833(2019)03-0036-06

经济负荷分配问题中的约束处理方法研究

吴 擎¹, 张春江², 高 亮³

(1.华中农业大学 工学院,湖北 武汉 430070; 2.南洋理工大学 电机与电子工程学院,新加坡 639798; 3.华中科技大学 机械科学与工程学院,湖北 武汉 430074)

摘 要:经济负荷分配是电力系统中重要的优化问题,该问题有若干约束,以往文献常采用罚函数法来处理约束,但罚函数法难以设置罚系数.可行性规则和 ε 约束法是两种高效且常用的约束处理技术,却难以直接应用于经济负荷分配问题.结合问题的特点,提出了一种将负荷平衡等式约束转化为边界不等式约束的方法:利用功率平衡约束,采用近似法和二次方程求根公式求出一个变量,并对该变量增加两个边界不等式约束,然后采用可行性规则或 ε 约束法来处理约束.实验部分采用两个经典的经济负荷分配问题对算法进行了测试.结果表明,与其他算法相比,该方法能求得更优的解.

关键词:经济负荷分配;约束处理方法;差分进化算法

中图分类号: TM71 **文献标志码:** A **doi:**10.13705/j.issn.1671-6833.2018.06.003

0 引言

经济负荷分配问题(economic load dispatch problem,EDP)的目标是在满足各种等式与不等式约束的情况下减少总燃料成本.求解EDP的优化方法大体可以分为两类:传统的数值优化方法和智能优化方法.传统的优化方法包括线性规划、二次规划和内点法等^[1-2].这类方法高效且精确,但对初始值很敏感,易提前收敛于局部最优解.智能优化方法包括遗传算法^[3-4]、进化策略^[5]、粒子群优化^[6-7]和差分进化算法(differential evolution,DE)^[8-9]等.此类方法对所求问题的特性没有要求,还能应用于复杂的大规模优化问题^[10],因而被越来越多的学者所采用.为了提高效率和精度,这些算法往往经过了复杂的改进,例如有些算法^[11]还需借助于二次序列规划的局部搜索能力,这些改进在很大程度上给使用者带来了麻烦.因此,笔者将采用最基本的差分进化算法^[12]来求解该问题.

本文的重点在于约束处理方法.前面提到的方法大部分采用罚函数法,该方法需要设置罚系数,易造成过惩罚或欠惩罚的问题.智能优化算法中最常用的约束处理方法有可行性规则(feasibility rules,FR)^[13]和 ε 约束法^[14],这两种方法简单高效且易于与进化算法结合,因而得到了非常广泛的应用^[15].但在EDP中,却鲜少发现这两种方法的存在.通过分析FR和 ε 约束法的特性以及所求问题的特点,发现了这两种方法不适合求解EDP问题的原因.在此基础上,结合EDP问题的特点,笔者提出了一种将负荷平衡约束转化为边界不等式约束的方法,转化后在进化算法中采用FR和 ε 约束法便十分有效了.实验结果表明,即便采用最简单的差分进化算法,仍然能求得非常有竞争力的解.

EDP的目标是在一定时间内在满足电力系统运行条件的情况下,通过对发电机组的发电功率进行优化使得系统的燃料成本最小.

1 EDP的数学模型

电力系统燃料成本FC是每个发电机组燃料成本的总和,如式(1)所示:

Min $FC = \text{Min} \sum_{i=1}^n FC_i(p_i)$, (1)

式中: n 为系统的发电机组数; p_i 为第*i*台发电机功率; $FC_i(p_i)$ 为第*i*个发电机组的燃料成本函数. $FC_i(p_i)$ 通常被拟合为一个二次多项式,如式(2)

所示:

$$FC_i(p_i) = a_i p_i^2 + b_i p_i + c_i, \quad i = 1, 2, \dots, n, \quad (2)$$

式中: a_i 、 b_i 、 c_i 为燃料成本的系数。

考虑以下4个约束。

(1) 功率平衡约束. 在考虑系统损耗的情况下, 电力系统的发电功率应等于实际功率需求 D 与系统的转运损耗 P_L 之和, 如式(3)所示:

$$\sum_{i=1}^n p_i = D + P_L. \quad (3)$$

系统的转运损耗由式(4)计算:

$$P_L = \sum_{i=1}^n \sum_{j=1}^n p_i B_{ij} p_j + \sum_{i=1}^n B_{i0} p_i + B_{00}, \quad (4)$$

式中: B_{ij} 、 B_{i0} 、 B_{00} 为损耗系数。

(2) 电机发电功率范围约束. 单个发电机组的发电功率 p_i 必须满足最小与最大功率约束 p_{imin} 、 p_{imax} , 即

$$p_{imin} \leq p_i \leq p_{imax}, \quad i = 1, 2, \dots, n. \quad (5)$$

(3) 功率升降限制约束. 考虑输出功率的升降限制, 即较前一次的输出功率相比, 当前的输出功率要在一定的范围之内. 具体约束如下式所示:

$$p_i^0 - DR_i \leq p_i \leq p_i^0 + UR_i, \quad i = 1, 2, \dots, n, \quad (6)$$

式中: p_i^0 为前一次的输出功率; DR_i 为当前功率下降时下降范围的限度; UR_i 为功率上升时上升范围的限度。

(4) 禁止功率区域约束. 某些发电机组有一些禁止功率区域. 其约束如式(7)所示:

$$p_i \in \begin{cases} p_{i,1}^{\min} \leq p_i \leq p_{i,1}^l; \\ p_{i,k-1}^u \leq p_i \leq p_{i,k}^l; \quad k = 2, \dots, zo_i; \\ p_{i,zo_i}^u \leq p_i \leq p_i^{\max}, \end{cases} \quad (7)$$

式中: $p_{i,k}^l$ 和 $p_{i,k}^u$ 为发电机组 i 的第 k 个禁止区的下界和上界。

2 约束处理

当采用进化算法求解上述问题时, 式(5)表示的电机发电功率范围约束通常被处理为自变量的边界约束。

数学模型中有4个约束, 通常将发电机组功率范围约束和功率升降限制约束结合形成如下的自变量边界约束:

$$\max(p_i^{\min}, p_i^0 - DR_i) \leq p_i \leq \min(p_i^{\max}, p_i^0 + UR_i), \quad (8)$$

可对新解中的越界分量重新赋值来处理该约束. 对于功率平衡约束和禁止功率区域约束, 以往文献中通常采用罚函数法. 除罚函数法以外, 可行性规则(feasibility rules, FR)与 ε 约束处理法是进化

算法中两种常见的约束处理方法^[15], 都可以用来比较两个解的优劣。

FR 最初用于遗传算法中的锦标赛选择, 其具体表述如下: 当两个解都不可行时, 根据约束违反量的大小来比较优劣; 当两个解一个可行、另一个不可行时, 可行解优于不可行解; 当两个解都是可行解时, 根据目标函数的大小比较优劣。

ε 约束法最初由 Takahama 和 Sakai^[16] 提出, 与 FR 一样, ε 约束法也可用来比较两个解的优劣, 只是在比较的时候增加了一个参数 ε . 比较规则如下: 当两个解的约束违反量都大于 ε 时, 根据约束违反量的大小比较优劣; 当两个解的约束违反量一个小于 ε 、另一个大于 ε 时, 前者优于后者; 当两个解的约束违反量都小于 ε 时, 根据目标函数的大小来比较优劣. 实际上, 当 $\varepsilon = 0$ 时, ε 约束法就等同于 FR. 通常采用下式来控制 ε 值^[15]:

$$\begin{aligned} \varepsilon(0) &= \varphi(X_\theta); \\ \varepsilon(t) &= \begin{cases} \varepsilon(0) (1 - t/T_c)^{cp}, & 0 < t < T_c; \\ 0, & t \geq T_c, \end{cases} \end{aligned} \quad (9)$$

式中: ε 的初值 $\varepsilon(0)$ 被设为在初始种群中排第 θ 位的约束违反量 $\varphi(X_\theta)$; t 为当前迭代次数. 当迭代次数 t 小于 T_c 时, ε 值按指数形式下降, 下降速度由参数 cp 控制; 否则的话, 将 ε 取值为 0。

EDP 中功率平衡等式约束会使得约束优化的可行域变为一个超平面. 当问题的维度很大时, 很难找到这个超平面上的可行解. 另一方面, 这个可行域超平面还很大, 这意味着当偏好可行解的 FR 找到一个可行解后, 就很难搜寻到远处的最优解了. ε 约束法能增加可行域的大小, 在一定程度上能克服 FR 的缺点, 但这种对可行域的扩增是暂时的, 当 ε 下降为 0 时, 其本质上仍然是可行性规则, 如果此时还没有找到超平面上邻近最优解的可行解, 就很容易陷入超平面上的局部最小值. 为了降低求解难度, 通常的做法是将等式约束松弛为一个不等式约束, 如式(10)所示. 但这一方面难免影响到解的精确性, 另一方面松弛为不等式约束后, 可行域的范围还是很小。

$$\left| \sum_{i=1}^n p_i - D - P_L \right| - \sigma < 0, \quad (10)$$

式中: σ 为一个很小的常数, 如 $1e-3$ 。

这里采用以下的改进方法处理功率平衡约束: 采用 DE 来求解该问题时, 只对前 $n-1$ 个变量进行优化. 在计算目标函数时, 先根据功率平衡约束求解第 n 个未被优化的变量 p_n , 并增加 p_n 的两

个边界约束,如式(11)所示,然后采用可行性规则或 ε 约束法来处理这两个约束:

$$\begin{cases} p_{n\min} - p_n < 0; \\ p_n - p_{n\max} < 0. \end{cases} \tag{11}$$

功率平衡约束为一个二次式.对于二次式,可采用求解二次方程的方法来进行转换,将式(3)转化如下:

$$B_{nn}p_n^2 + \left(\sum_{i=1}^{n-1} p_i B_{in} + \sum_{j=1}^{n-1} B_{nj}p_j + B_{n0} - 1 \right) p_n + \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} p_i B_{ij}p_j + \sum_{i=1}^{n-1} (B_{i0} - 1)p_i + B_{00} + D = 0.$$

令 $a = B_{nn}; b = \sum_{i=1}^{n-1} p_i B_{in} + \sum_{j=1}^{n-1} B_{nj}p_j + B_{n0} - 1;$

$$c = \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} p_i B_{ij}p_j + \sum_{i=1}^{n-1} (B_{i0} - 1)p_i + B_{00} + D.$$

则根据一元二次方程的求根公式可得:当 $b^2 - 4ac \geq 0$ 时, $p_n = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. 在这种情况下, b

值通常接近 $-1, \sqrt{b^2 - 4ac} < b$, 因此,方程的两个根都是正值.但是两根中的较小值通常符合范围约束,而较大值却远远超出可行域的范围,因此,在这里取根为:

$$p_n = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \tag{12}$$

但并不是所有的情况下,都满足 $b^2 - 4ac \geq 0$. 尤其是在搜索过程的早期,通常 $b^2 - 4ac < 0$, 此时不能采用一元二次函数的求根公式求解 p_n 值.因此,对式(3)进行简单变形,如下式所示:

$$p_n = D + P_L - \sum_{i=1}^{n-1} p_i. \tag{13}$$

此处采用了上次迭代中的 p_n 值来计算 P_L ,能够如此简化的原因是,相对于 p_n, P_L 的值较小, p_n 对 P_L 的贡献值也很小,采用不精确的 p_n 对 P_L 的影响不大.通过上式可求得 p_n 的近似值,也能很快搜寻到可行域附近.当搜寻到可行域附近后,易满足 $b^2 - 4ac \geq 0$ 的条件,然后就可根据式(12)求得满足功率平衡约束的可行解.

该约束处理方式的优势有三点:最重要的是将可行域极小的等式约束转化为了可行域较大的不等式约束,易找到可行解.其二,缩减了问题的一个维度,把 n 维的问题缩减到了 $n-1$ 维.第三,不需要对等式约束进行松弛,能获得精确解.

对于功率禁止区域约束,如果 p_i 位于第 k 个禁止区域中: $p_{i,k}^l \leq p_i \leq p_{i,k}^u$, 采用式(14)计算其

约束违反量的值.对每一项求和就可得到功率禁止区域约束的约束违反量值.

$$\phi(p_{i,k}) = \min(p_i - p_{i,k}^l, p_{i,k}^u - p_i). \tag{14}$$

3 实验结果与分析

3.1 约束差分进化算法

这里采用差分进化算法作为优化方法.差分进化算法(DE)最初由 Storn & Price^[12]在1997年提出,它具有简单、高效、参数少、易于编程实现等优点,DE是近年来得到关注最广泛的进化算法之一.它有4个基本步骤:初始化;变异;交叉;选择.根据变异策略及交叉方式的不同,DE算法有很多变种,这里采用最基本的DE/rand/1/exp.

结合可行性规则的DE的伪代码如图1所示.结合 ε 约束法的DE的伪代码与之类似.

```
DE/rand/1/exp-FR
{
    P(0): 从可行域随机生成N个个体{xi}为初始种群;
    for (t=1; t ≤ Tmax; t++) {
        for (i=1; i ≤ N; i++) {
            从 [1, M] 中随机选择 p1, p2 和 p3
            s.t. p1 ≠ p2 ≠ p3 ≠ i;
            xnew = xi ∈ P(t-1);
            从 [1, n] 中随机选择 j;
            k=1;
            do { //交叉
                xjnew = xjn + F(xjp2 - xjp3) //变异
                if (xjnew < Lk || xjnew > Uk) //边界处理
                    xjnew = xji
                j = (j+1) % n;
                k++;
            } while (k ≤ n && u(0,1) < CR);
            采用可行性规则比较 xnew 和 xi
            if (xnew 优于 xi) //选择
                zi = xnew;
            else
                zi = xi;
        }
        P(t) = {zi, i=1, 2, ..., N}
    }
}
```

图 1 结合可行性规则的 DE/rand/1/exp 的伪代码

Fig.1 Pseudo code of DE/rand/1/exp with Feasibility rules

3.2 实例描述及参数设置

选用两个经典问题^[7]来进行测试,将其分别命名为 EDP1 和 EDP2.EDP1 包含 6 个发电单元, 26 根总线和 46 根传输线,总功率需求是 1 263 MW,6 个单元的特性及损耗系数 B 值见文献[7].

EDP2 包含 15 个发电单元,总功率需求是 2 630 MW,15 个单元的特性及 B 值见文献[7], 约束差分进化算法参数设置如表 1 所示.

表 1 算法参数设置

Tab.1 Parameter setting of algorithm

参数符号	参数含义	参数值
N	种群数量	20
$MaxFES$	最大评估次数	5 000(EDP1), 20 000(EDP2)
F	缩放系数	0.95
CR	交叉概率	0.98
σ	约束容许量	0.000 1
θ	ε 约束参数 1	0.8 N
T_c	ε 约束参数 2	0.4 T_{max}
cp	ε 约束参数 3	4

3.3 实验结果

3.3.1 4 种方案比较

将 FR 与 DE 结合并采用式 (10) 处理功率平衡约束称为方案 1;将 ε 约束法与 DE 结合并

采用式 (10) 处理功率平衡约束称为方案 2;将 FR 与 DE 结合并采用改进方法处理功率平衡约束称为方案 3;将 ε 约束法与 DE 结合并采用改进方法处理功率平衡约束求称为方案 4.

4 种方案在 EDP1 和 EDP2 上独立运行 30 次的统计结果如表 2 所示.从表 2 可见,对于 EDP1 和 EDP2,方案 3 和方案 4 的结果明显优于方案 1 和方案 2.对于 EDP1,方案 2 的结果也明显优于方案 1,可见对于该问题 ε 约束法是有效的.方案 3 和方案 4 没有显著性差异,这是因为采用改进方法处理功率平衡约束后,算法的效率很高,方案 3 和方案 4 都能找到很好的解.对于 EDP2,方案 1 的最优值和最差值要优于方案 2,但方案 2 的平均值要优于方案 1,而方案 4 的结果优于方案 3,可见 ε 约束法表现较好.

表 2 4 种方案在 EDP1 和 EDP2 上的统计结果

Tab.2 Statistical results for four tests on EDP1 and EDP2

美元

方案	EDP1				EDP2			
	最优值	平均值	最差值	标准差	最优值	平均值	最差值	标准差
方案 1	15 444.13	15 465.46	15 499.12	17.402 2	32 955.35	33 107.18	33 239.68	75.594 12
方案 2	15 444.10	15 459.46	15 493.81	10.654 0	32 972.38	33 079.71	33 251.82	75.081 19
方案 3	15 443.57	15 443.57	15 443.57	3.82E-07	32 698.79	32 705.12	32 751.61	14.841 84
方案 4	15 443.57	15 443.57	15 443.57	2.79E-07	32 698.38	32 700.12	32 702.80	1.144 96

4 种方案在两个问题上的收敛过程如图 2 和图 3 所示.在搜索前期,方案 1 出现了目标函数值的波动,这是因为在可行性规则下,方案 1 力求找到可行解,只优化约束违反量.而方案 2 在 ε 约束的作用下,可以先优化目标函数,因而刚开始时方案 2 获得了很小的目标函数值,但随着 ε 的减小,目标函数值越来越大;当 ε 快变为 0 时,收敛曲线在最优解附近停留了一段时间,但最终丢失了该解,陷入了局部最优.这是因为在功率平衡等式约束下,可行域太小,而可行性规则和 ε 约束法偏

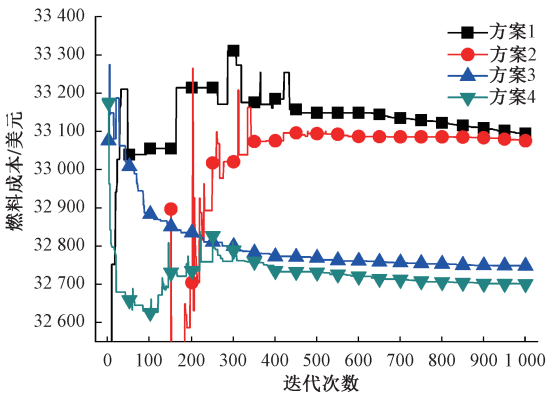


图 3 4 种方案在 EDP2 上目标函数的收敛图

Fig.3 Convergence plots of four tests on EDP2

好可行解,难以找到全局最优.而方案 3 和方案 4,对唯一的等式约束进行了处理,使算法中的种群都是可行解,能直接优化目标函数.因此,方案 3 和方案 4 都极快地收敛到了全局最优解.

对于 EDP2,方案 1 和方案 2 的表现与 EDP1 中类似.但因为该问题维度较高,方案 1 和方案 2 的解与方案 3 和方案 4 相差较大.方案 3 和方案 4 相比,很显然, ε 约束法在这里起了很大作用.方案 4 中,目标函数值先迅速下降,随着 ε 的减小,略有上升,然后再缓慢下降.与方案 2 不同的是,

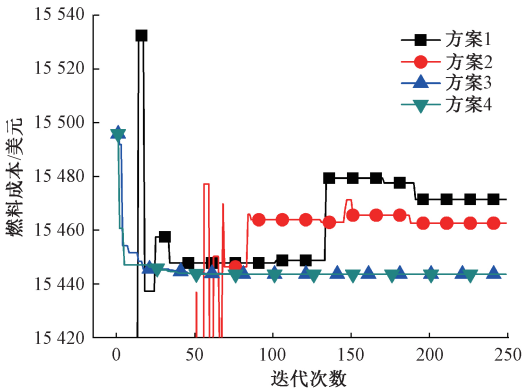


图 2 4 种方案在 EDP1 上目标函数的收敛图

Fig.2 Convergence plots of four tests on EDP1

方案 4 中的目标函数值变化比较缓慢,不至于丢失近优解.

3.3.2 与文献方法的比较

文献中有很多算法对该问题进行了求解,如模拟退火算法(simulated annealing, SA)^[17]、遗传算法(genetic algorithm, GA)^[6]、粒子群算法(particle swarm optimization, PSO)^[6-7,18]、人工免疫系统(artificial immune system, AIS)^[19].这些算法均采用罚函数方法来处理约束,并且大多在基础算法上进行了改进.

各种算法的统计结果如表 3 和表 4 所示.表中展示了各算法在多次运行下的最优值、平均值、最差值和标准差.某些算法的标准差未知,对于函数评估次数,PSO 和 GA 均为 20 000 次;CPSO(charotic particle swarm optimization)算法在文献[7]中未给出局部搜索的参数,但能推算出函数评估次数要大于 14 400;对于 SOH_PSO(self-organizing hierarchical particle swarm optimization)^[18],在 EDP2 上的种群数量达到了 400 和 500,即使其函数迭代次数只有 125,其函数评估次数也分别为 50 000 和 62 500,远超方案 3 和方案 4 在 EDP2 上的最大函数评估次数 20 000.通过这些数据可知,方案 3 和方案 4 在 EDP1 和 EDP2 上的最大函数评估次数(分别为 5 000 和 20 000)是比较少的.

表 3 在 EDP1 上各算法的统计结果

Tab.3 Statistical results of some algorithms on EDP1

美元				
算法	最优值	平均值	最差值	标准差
SA	15 461.10	15 488.98	15 545.50	28.37
GA	15 459.00	15 469.00	15 524.00	—
PSO	15 450.00	15 454.00	15 492.00	—
CPSO	15 446.00	15 449.00	15 490.00	—
SOH_PSO	15 446.02	15 497.35	15 609.64	—
AIS	15 448.00	15 459.70	15 472.00	6.25
方案 3 算法	15 443.57	15 443.57	15 443.57	3.82E-07
方案 4 算法	15 443.57	15 443.57	15 443.57	2.79E-07

表 4 在 EDP2 上各算法的统计结果

Tab.4 Statistical results of some algorithms on EDP2

美元				
算法	最优值	平均值	最差值	标准差
SA	32 786.40	32 869.51	33 028.95	112.32
GA	33 113.00	33 337.00	33 228.00	—
PSO	32 858.00	33 331.00	33 105.00	—
CPSO	32 834.00	33 021.00	33 318.00	—
SOH_PSO	32 751.00	32 878.00	32 945.00	—
AIS	32 854.00	32 873.25	32 892.00	10.81
方案 3 算法	32 698.79	32 705.12	32 751.61	14.84
方案 4 算法	32 698.38	32 700.12	32 702.80	1.14

从表 3 可见,对于 EDP1,方案 3 和方案 4 的结果要全面优于所比较的算法.其原因就在于等式约束的存在,而采用罚函数的方法容易造成过惩罚,使算法偏好可行解,容易收敛到局部最优.

从表 4 可见,对于 EDP2,方案 4 在所有指标中都是最优的;而方案 3 的结果也远优于其他算法.由此可见,对于维度较大的问题,采用罚函数法来处理约束的效果更差.采用改进方法处理功率平衡约束后,FR 和 ε 约束法均能求得较好的结果,特别是 ε 约束法的效果更佳.

4 结语

可行性规则与 ε 约束法是进化算法中两种常用的约束处理方法,在以往的文献中,虽然有很多的进化算法被用来求解 EDP 问题,但大部分约束处理方法都是传统的罚函数法.笔者分析了 EDP 问题中不适合采用可行性规则和 ε 约束法的原因,并提出了一种将 EDP 中的功率平衡约束转换为不等式约束的方法.实验结果表明,采用该方法时,即便是采用最基本的差分进化算法,也能求得非常有竞争力的解.

参考文献:

[1] ADLER R B, FISCHL R. Security constrained economic dispatch with participation factors based on worst case bus load variations[J]. IEEE transactions on power apparatus and systems, 1977, 96(2): 347-356.

[2] IRISARRI G, KIMBALL L M, CLEMENTS K A, et al. Economic dispatch with network and ramping constraints via interior point methods [J]. IEEE transactions on power systems, 1998, 13(1): 236-242.

[3] LING S H, LEUNG F H F. An improved genetic algorithm with average-bound crossover and wavelet mutation operations[J]. Soft computing, 2007, 11(1): 7-31.

[4] 杨胡萍, 李威仁, 左士伟, 等. 基于改进遗传算法的电力系统无功优化[J]. 郑州大学学报(工学版), 2015, 36(6): 66-69.

[5] PEREIRA-NETO A, UNSIHUAY C, SAAVEDRA O R. Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch problem with generator constraints [J]. IEEE proceedings-generation, transmission and distribution, 2005, 152(5): 653-660.

[6] GAING Z L. Particle swarm optimization to solving the economic dispatch considering the generator constraints [J]. IEEE transactions on power systems, 2003, 18(3): 1187-1195.

- [7] CAI J J, MA X Q, LI L X, et al. Chaotic particle swarm optimization for economic dispatch considering the generator constraints[J]. Energy conversion and management, 2007, 48(2): 645-653.
- [8] COELHO L S, MARIANI V C. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect[J]. IEEE transactions on power systems, 2006, 21(2): 989-996.
- [9] 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.
- [10] 梁静, 刘睿, 瞿博阳, 等. 进化算法在大规模优化问题中的应用综述[J]. 郑州大学学报(工学版), 2018, 39(3): 15-21.
- [11] VICTOIRE T A A, JEYAKUMAR A E. Hybrid PSO-SQP for economic dispatch with valve-point effect[J]. Electric power systems research, 2004, 71(1): 51-59.
- [12] STORN R, PRICE K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces[J]. J Global Optim, 1997, 11(4): 341-359.
- [13] DEB K. An efficient constraint handling method for genetic algorithms[J]. Computer methods in applied mechanics and engineering, 2000, 186(2/4): 311-338.
- [14] TAKAHAMA T, SAKAI S. Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites[C]//2006 IEEE International Conference on Evolutionary Computation (CEC). Vancouver, BC: IEEE, 2006: 1-8.
- [15] MEZURA-MONTES E, COELLO C A C. Constraint-handling in nature-inspired numerical optimization: past, present and future[J]. Swarm and evolutionary computation, 2011, 1(4): 173-194.
- [16] TAKAHAMA T, SAKAI S, IWANE N. Constrained optimization by the ε constrained hybrid algorithm of particle swarm optimization and genetic algorithm[C]//Proceeding of AI 2005: Advances in Artificial Intelligence. Berlin: Springer, 2005: 389-400.
- [17] POTHIA S, NGAMROO I, KONGPRAWECHNON W. Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints[J]. Energy conversion and management, 2008, 49(4): 506-516.
- [18] CHATURVEDI K T, PANDIT M, SRIVASTAVA L. Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch[J]. IEEE transactions on power systems, 2008, 23(3): 1079-1087.
- [19] PANIGRAHI B K, YADAV S R, AGRAWAL S, et al. A clonal algorithm to solve economic load dispatch[J]. Electric power systems research, 2007, 77(10): 1381-1389.

Study of Constraint Handling Methods in Economic Load Dispatch Problem

WU Qing¹, ZHANG Chunjiang², GAO Liang³

(1.School of Engineering, Huazhong Agricultural University, Wuhan 430070, China; 2.School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore; 3.School of Mechanical Science & Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Economic load dispatch was an important problem in power system. Some constraints existed in this problem. In the literature, penalty function methods were usually used. However, it was difficult to set proper penalty coefficients. Feasibility rules and ε constraint method were two popular constraint handling techniques. However, they were not efficient for economic load dispatch problem. A constraint handling method that could transform the load balance constraint to two boundary constraints was proposed in this paper. Taking advantage of the load balance constraint, variable was solved by using approximation method and the root of quadratic equation. Therefore, two boundary inequality constraints were added for the variable. Feasibility rules and ε constraint method were used to handle the constraints. Two classic instances were tested in the numerical experiment part. The results showed the proposed algorithm could obtain better solutions than other compared algorithms in the literature.

Key words: economic load dispatch problem; constraint handling method; differential evolution