

文章编号: 1001-0920(2009)08-1156-05

基于混合双种群差分进化的电力系统经济负荷分配

王 凌, 黄付卓, 李灵坡

(清华大学 a. 自动化系, b. 信息科学与技术国家实验室, 北京 100084)

摘 要: 针对电力系统经济负荷分配本质上的非线性约束优化问题, 提出一种双种群混合差分进化算法. 采用两个种群且以较小的计算量实现目标函数的寻优并保持解的可行性, 同时引入单纯型法来提高算法的局部搜索能力. 基于典型算例对该算法的进化行为进行测试, 并通过仿真和比较验证了所提出算法的有效性.

关键词: 经济负荷分配; 约束处理; 差分进化; 双种群

中图分类号: TP18 **文献标识码:** A

Economic distribution of power systems based on hybrid differential evolution with double populations

WANG Ling, HUANG Fu-zhuo, LI Ling-po

(a. Department of Automation, b. National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China. Correspondent: WANG Ling, E-mail: wangling@tsinghua.edu.cn)

Abstract: Economic distribution of power systems can be regarded as a non-linear constrained optimization problem. In this paper, a hybrid double-population differential evolution is proposed, which uses two populations to optimize objective function and to maintain feasibility of solutions with a little computational effort. In addition, simplex method is embedded to enhance the exploitation ability. The evolving behavior of the algorithm is tested based on a typical instance. Simulation and comparison show the effectiveness of the proposed algorithm.

Key words: Economic dispatch; Constraint handling; Differential evolution; Double populations

1 引 言

电力系统经济负荷分配(ED)^[1]是在满足负载需求的约束条件下,对发电机组单元进行调配,最小化发电成本的约束优化问题.除约束性外,考虑阀点效应,发电机的耗量特性不是一个处处可导的函数,且问题存在强非线性和多极小性,给求解带来了很大困难,已成为智能算法求解的经典问题.典型的电力分配问题通常有 3 维、13 维和 40 维 3 类.文献[2]提出一种动态规划方法,但要求目标函数可导,忽略了阀点效应.[3]采用人工免疫混沌优化算法进行求解,有效地解决了 10 维问题,但未能解决 40 维的 ED 问题.[4]给出一种改进的进化规划算法,显示了良好的优化质量.

差分进化(DE)算法^[5]是一种新兴的进化计算技术,最初用于解决切比雪夫多项式逼近问题,后来发展成为解决复杂优化问题的有效技术.DE 是基于群体智能理论的优化算法,通过群体内个体间的

合作与竞争产生群体智能,用于指导优化搜索.该算法在函数优化方面表现出较强的优化能力^[6].

本文针对约束优化问题,提出一种基于双种群的混合 DE 算法(HDEDP),并将该算法应用于 ED 问题.仿真结果验证了所提出算法的鲁棒性和有效性.

2 问题描述

ED 问题是电力系统中最常见的一类优化问题.电力系统在满足操作约束的条件下,要求最小化发电机的燃料成本.该问题通常可描述为如下包含一个目标函数和两类约束的形式:

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

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

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

其中: P_i 为第 i 台发电机的功率, $F_i(P_i)$ 为第 i 台发

收稿日期: 2008-09-15; 修回日期: 2008-12-08.

基金项目: 国家自然科学基金项目(60834004,70871065,60774082).

作者简介: 王凌(1972—),男,江苏武进人,教授,博士生导师,从事智能优化、调度理论与方法等研究;李灵坡(1984—),男,河南洛阳人,硕士生,从事智能优化算法的研究.

电机的燃料成本, n 为发电机的数量. 第 1 个约束来自能量守恒定律, 即所有发电机功率之和等于电网线路损耗 P_L 与系统内电力需求 P_D 之和; 第 2 个约束表示各发电机所能达到的功率范围.

由于传统数学规划方法要求目标函数可导, $F_i(P_i)$ 通常采用下式描述的近似模型:

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

其中 a_i, b_i 和 c_i 是第 i 台发电机的相关常数.

实际上, 当考虑发电机的阀点效应时, 发电机的燃料成本可用下式更精确地描述:

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

其中 e_i 和 f_i 是与第 i 台发电机阀点效应有关的常数.

由于模型不可导, 无法用传统的非线性规划方法求解, 这为 DE 算法提供了展示的空间.

3 差分进化算法

DE 具有记忆个体向量最优解的能力和种群内信息共享的机制. 即通过种群间个体向量的合作与竞争, 实现优化问题的求解. DE 算法基于实数编码, 在问题的可行解空间随机初始化种群

$$X^0 = [x_1^0, x_2^0, \dots, x_{np}^0],$$

其中: 个体向量 $x_i^0 = [x_{i,1}^0, x_{i,2}^0, \dots, x_{i,d}^0]$ 用于表征问题解, d 为问题维数, np 为种群规模.

在每代进化过程中, 首先对当前种群分别进行变异和交叉操作, 以产生新种群; 然后利用基于贪婪思想的选择操作, 对这两个种群进行一对一的选择, 从而产生最终的新种群. 在第 t 代进化时, 种群对各个体向量 x_i^t 实施变异操作, 得到与其相对应的变异个体 v_i^{t+1} . 即

$$v_i^{t+1} = x_{r1}^t + F(x_{r2}^t - x_{r3}^t). \quad (6)$$

其中: $r1, r2$ 和 $r3$ 是随机选取的互不相同且与 i 不同的整数; x_{r1}^t 为父代基向量; $(x_{r2}^t - x_{r3}^t)$ 为父代差分向量; F 为缩放因子.

变异操作是算法实现全局探索的关键, 它利用原种群的个体向量信息生成新解. 变异操作完成后, 利用下式:

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & \text{rand}(j) < p_c \text{ or } j = \text{rnbr}(i); \\ x_{i,j}^t, & \text{else.} \end{cases} \quad (7)$$

对 x_i^t 和生成的变异向量 v_i^{t+1} 实施交叉操作, 生成实验向量 u_i^{t+1} . 其中: $\text{rand}(j)$ 为 $[0, 1]$ 上均匀分布的随机数; $\text{rnbr}(i)$ 为 $\{1, 2, \dots, d\}$ 之间的随机整数, p_c 为交叉概率. 显然, 交叉概率越小, u_i^{t+1} 保留原个体的信息越多. 交叉操作使实验向量 u_i^{t+1} 同时继承了原向量和变异向量的信息.

最后, 对实验向量 u_i^{t+1} 和 x_i^t 的目标值进行比较. 对于最小化问题, 选择目标值低的个体作为新种群的个体 x_i^{t+1} . 即

$$x_i^{t+1} = \begin{cases} u_i^{t+1}, & f(u_i^{t+1}) < f(x_i^t); \\ x_i^t, & \text{else.} \end{cases} \quad (8)$$

DE 的选择操作采用贪婪的原则, 保证种群能快速跟踪种群最优信息. 这也是算法提高局部搜索能力的关键步骤.

标准 DE/rand/1/bin 版本的 DE 流程如下:

步骤 1: 从搜索空间中随机选出 np 个个体作为初始种群, 各个体均为实数向量.

步骤 2: 对所有的个体向量计算目标函数值.

步骤 3: 根据式(6)和(7)的变异和交叉操作, 为各个体创建一个实验向量.

步骤 4: 对所有的实验向量计算目标函数值.

步骤 5: 根据式(8)进行选择操作.

步骤 6: 若满足终止条件, 则输出已找到的最优解; 反之, 返回步骤 3.

4 混合双种群差分进化算法

求解约束优化问题的核心在于保持算法寻优能力与保持可行性之间的平衡. 本文提出一种基于双种群的混合 DE 算法, 对约束和目标函数进行解耦. 第 1 个种群采用 DE 单独对目标函数或约束进行优化; 第 2 个种群只包含可行解, 随时对第 1 个种群中的不可行解进行修复. 为了提高算法的局部寻优能力, 将单纯型法作为一种局部搜索算法融入 DE 算法.

4.1 等式约束处理

式(2)是一个简单的线性等式约束, 可用该式并采用代入法对原问题进行降维. 于是

$$\min \sum_{i=1}^{n-1} F_i(P_i) + F_n(P_D + P_L - \sum_{i=1}^{n-1} P_i); \quad (9)$$

$$\text{s. t. } P_D + P_L - \sum_{i=1}^{n-1} P_i \geq P_n^{\min}, \quad (10)$$

$$P_D + P_L - \sum_{i=1}^{n-1} P_i \leq P_n^{\max}, \quad (11)$$

$$P_i^{\min} \leq P_i \leq P_i^{\max}, i = 1, 2, \dots, n-1. \quad (12)$$

经过上述转换后, 一个等式约束便转化为两个不等式约束.

4.2 修复不可行解

修复的主要任务是将一个不可行解修复成为一个可行解. 在此给出一种较为简单的操作, 利用可行解 x_{feas} 将不可行解 x_{inf} 修复到边界上. 将不可行解修复到边界上的理由如下:

1) 对于绝大多数约束优化问题, 其最优可行解

几乎总在边界上. 对于最优可行解不在边界点上的问题, HDDEP 算法不需进行修复操作.

2) 在所有的可行解集中, 离不可行解 \mathbf{x}_{inf} 最近(欧拉距离)的点一定是边界点. 在连续性条件的假设下, 这样的边界点比其他可行解保留了不可行解 \mathbf{x}_{inf} 更多的信息.

在实际操作中, 找到离 \mathbf{x}_{inf} 最近的边界点十分困难. 为此, 简单地在连接不可行解 \mathbf{x}_{inf} 和可行解 \mathbf{x}_{fea} 的线段上找到一个边界点.

引理 1 对于任何一个连续函数 $g(\mathbf{x})$, 以及一对点 $\mathbf{x}_1 \in \{\mathbf{x} \mid g(\mathbf{x}) > 0\}$ 和 $\mathbf{x}_2 \in \{\mathbf{x} \mid g(\mathbf{x}) \leq 0\}$, 总存在 $[0, 1]$ 上的实数 α_0 , 使得以下二式成立:

$$g(\alpha_0 \mathbf{x}_1 + (1 - \alpha_0) \mathbf{x}_2) = 0, \quad (13)$$

$$g(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2) \leq 0, \forall \alpha \in [0, \alpha_0]. \quad (14)$$

证明 令 $h(\alpha) = g(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2)$, 则 $h(\alpha)$ 是一个定义在 $[0, 1]$ 上的连续函数, 且有 $h(0) \leq 0$, $h(1) > 0$. 因此, 至少存在一个 $[0, 1]$ 上的 α_1 , 使得 $h(\alpha_1) = 0$.

令集合 $A = \{\alpha \mid h(\alpha) = 0, \alpha \in [0, 1]\}$, 集合 A 显然是一个紧集. 则 A 必然有下确界, 记作 α_0 . 由紧集性质知, $\alpha_0 \in A$, 故 $h(\alpha_0) = 0$. 从而式(13)得证.

假设存在 $\alpha_0 \in [0, \alpha_0]$, 使得 $h(\alpha_2) > 0$. 由函数 $h(\alpha)$ 的连续性知, 至少存在一个 $\alpha_3 \in [0, \alpha_2]$, 使得 $h(\alpha_3) = 0$, 这与 α_0 是集合 A 的下确界矛盾, 因此假设不成立. 即 $\forall \alpha \in [0, \alpha_0], h(\alpha) \leq 0$. 从而式(14)得证. \square

定理 1 对于任意一对可行解 \mathbf{x}_{fea} 和不可行解 \mathbf{x}_{inf} , 总存在 $[0, 1]$ 上的实数 α_0 , 使得 $\alpha \in [0, \alpha_0]$, $g(\alpha \mathbf{x}_{\text{inf}} + (1 - \alpha) \mathbf{x}_{\text{fea}}) \leq 0$ 是边界点. 即已知

$$g_i(\mathbf{x}_{\text{inf}}) > 0, i = 1, 2, \dots, n_1;$$

$$g_i(\mathbf{x}_{\text{inf}}) \leq 0, i = n_1 + 1, \dots, n;$$

$$g_i(\mathbf{x}_{\text{fea}}) \leq 0, i = 1, 2, \dots, n.$$

求证存在 $[0, 1]$ 上的实数 α_0 , 使得 $g_i(\alpha \mathbf{x}_{\text{inf}} + (1 - \alpha) \mathbf{x}_{\text{fea}}) \leq 0, i = 1, 2, \dots, n_1$, 且至少有一个等号成立.

证明 对于不满足条件的约束 $g_i(\mathbf{x}_{\text{inf}}), i = 1, 2, \dots, n_1$, 由引理 1 知, 存在 $\alpha_i \in [0, 1]$, 使得 $g_i(\alpha_i \mathbf{x}_{\text{inf}} + (1 - \alpha_i) \mathbf{x}_{\text{fea}}) = 0$, 且 $\forall \alpha \in [0, \alpha_i], g_i(\alpha \mathbf{x}_{\text{inf}} + (1 - \alpha) \mathbf{x}_{\text{fea}}) \leq 0$.

对于满足条件的约束 $g_i(\mathbf{x}_{\text{inf}}), i = n_1 + 1, n_1 + 2, \dots, n$, 若存在任意的 $\alpha_j, j = n_1 + 1, n_1 + 2, \dots, n$, 均有 $g_j(\alpha_j' \mathbf{x}_{\text{inf}} + (1 - \alpha_j') \mathbf{x}_{\text{fea}}) \leq 0, i = 1, 2, \dots, n_1$, 则令 $\alpha_j = 1$; 否则, 由引理 1 知, 存在 $\alpha_j \in [0, \alpha_j']$, 使得 $g_j(\alpha_j \mathbf{x}_{\text{inf}} + (1 - \alpha_j) \mathbf{x}_{\text{fea}}) = 0$, 且 $\forall \alpha \in [0, \alpha_j], g_j(\alpha \mathbf{x}_{\text{inf}} + (1 - \alpha) \mathbf{x}_{\text{fea}}) \leq 0, i = 1, 2, \dots, n_1$.

综上所述, 对于所有的约束 $g_i(\mathbf{x}_{\text{inf}}), i = 1, 2,$

\dots, n_1 , 存在相应的 $\alpha_i \in [0, 1]$, 使得 $g_i(\alpha_i \mathbf{x}_{\text{inf}} + (1 - \alpha_i) \mathbf{x}_{\text{fea}}) = 0$, 且 $\forall \alpha \in [0, \alpha_i], g_i(\alpha \mathbf{x}_{\text{inf}} + (1 - \alpha) \mathbf{x}_{\text{fea}}) \leq 0$. 于是, 令 $\alpha_0 = \min_{i=1, 2, \dots, n} \{\alpha_i\}$, 则 α_0 便是满足条件的实数. \square

上述定理表明, 在连续性条件的假设下, 连接不可行解和可行解的线段上一定存在边界点. 这样, 采用简单的二分法便可找到边界点.

4.3 可行性规则

单纯型法(SM)^[7]是不用梯度信息直接处理无约束优化问题的一种算法. SM算法在 d 维空间中构造一个包含 $d+1$ 个顶点的单纯型, 通过反射、扩张、内缩和缩边 4 种操作, 求出一个较好解来替换单纯型中的最差解. SM 算法的具体步骤参见文献[7].

在许多混合算法^[8,9]中, SM 算法常用于局部搜索. 对于无约束优化问题, 所有解的集合以目标函数的大小为比较标准组成一个全序集. 对于约束优化问题, 需要重新定义两个解之间序的关系, 才能使用 SM 算法. HDDEP 采用可行性规则来比较两个解的优劣. 若 \mathbf{x}_1 和 \mathbf{x}_2 是约束优化问题的两个解向量, 则称 $\mathbf{x}_1 \leq \mathbf{x}_2$, 当且仅当 \mathbf{x}_1 和 \mathbf{x}_2 满足下列 3 种情况之一:

1) \mathbf{x}_1 是可行解, \mathbf{x}_2 是不可行解;

2) \mathbf{x}_1 和 \mathbf{x}_2 均为可行解, $f(\mathbf{x}_1) \leq f(\mathbf{x}_2)$;

3) \mathbf{x}_1 和 \mathbf{x}_2 均不可行, 且 $\text{voil}(\mathbf{x}_1) \leq \text{voil}(\mathbf{x}_2)$. 其中 $\text{voil}(\mathbf{x})$ 是不可行解 \mathbf{x} 违反约束的大小, 即

$$\text{voil}(\mathbf{x}) = \sum_{\forall j, \text{s. t. } g_j(\mathbf{x}) > 0} g_j(\mathbf{x}). \quad (15)$$

该可行性规则在保证解的可行性的前提下, 对目标函数进行排序. HDDEP 算法根据可行性规则, 采用 SM 算法对可行域进行局部搜索.

4.4 混合算法设计

HDDEP 算法的进化过程由两个种群间的互动来完成: 种群 1 采用 DE 算法单独对目标函数或约束进行优化, 种群 2 在整个进化过程中只用于保存可行解. 两个种群的初始化过程如下: 种群 1 在解空间上随机初始化, 并将初始化过程中产生的可行解存入种群 2. 若初始化过程中没有产生可行解, 则种群 1 利用 DE 进化若干代来寻找可行解, 并将可行解存入种群 2. 若仍未产生可行解, 则退出; 反之, 种群 1 进入下一步进化过程.

HDDEP 算法的框图如图 1 所示. 初始化种群完成后, 在种群 2 不为空的情况下, 种群 1 采用 DE 算法进化一代, 并选出目标函数值最好的个体. 若最优个体不可行, 则从种群 2 中随机选取一个可行解, 利用修复操作把最优个体修复到边界上, 再用 SM 法对可行的最优解进行局部搜索. 图 1 将 HDDEP 算

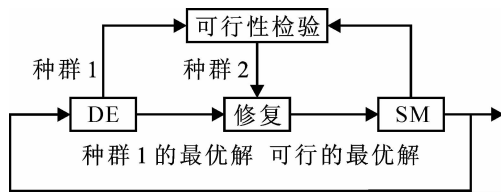


图 1 HDDEP 算法框图

法分成 4 个模块,每个模块负责完成自己的任务.除本文的设计外,每个模块也可单独进行改进或重新设计,例如 SM 模块和修复算法均可用其他算法替代.

在 HDDEP 算法中,只有种群 1 通过 DE 进化,种群 2 只是可行解的储存库.因此,维护种群 2 所需的时间花费很小,所需的存储空间就是种群 2 的大小.相比于其他智能约束优化算法,HDDEP 付出的额外开销较小.

5 算法行为测试

本节以国际上常用的 $g06$ 函数^[10] 对 HDDEP 的进化过程进行测试.该函数可行域非常小,随机产生 100 万个初始解中仅有 0.0065% 可行.

$$g06: \min f(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3;$$

$$\text{s. t. } \begin{cases} g_1(\mathbf{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0, \\ g_2(\mathbf{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0, \\ 13 \leq x_1 \leq 100, 0 \leq x_2 \leq 100. \end{cases}$$

在 HDDEP 算法中,种群 1 规模为 400,种群 2 规模为 1000,最大进化步数为 500.算法高效地获得了问题的最优解 (14.095, 0.84296),最优值为 -6961.814.

图 2 ~ 图 5 显示了每隔 200 代种群 1 中解的分布,图 6 和图 7 则显示了进化过程中种群 2 中解的分布.其中:图 2 中左下角的曲线和图 3 ~ 图 7 中的月牙形部分是问题的可行域,虚线为目标函数等高线.另外,图 8 显示了种群 2 中解的数量随着进化变化的情况.

由图 2 ~ 图 5 可见,初始阶段种群 1 几乎没有解在可行域上.从 100 代开始,种群 1 中的解基本集中在可行域附近,大量的解分布在可行域的边界上.随着进化代数的增加,种群 1 沿可行域边界逐渐向最优可行解逼近,最终整个种群收敛到最优可行解.

由图 6 ~ 图 8 可见,初始阶段种群没有可行解产生,种群 1 以减小约束为目的进行进化,最终找到可行解;在随后的进化过程中,种群 2 迅速扩大到最大规模,且基本上是较好的可行解.随着种群 1 的进化,种群 2 中的可行解也逐渐向最优可行解靠近,

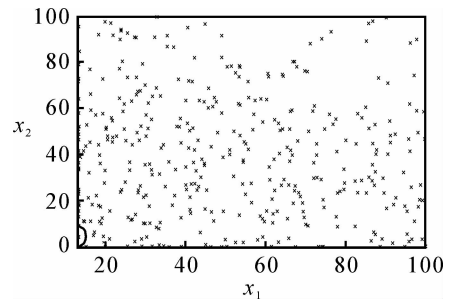


图 2 初始种群 1 中解的分布

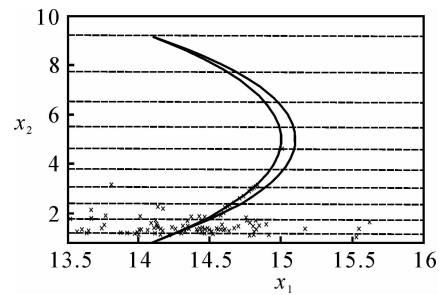


图 3 第 100 代时种群 1 中解的分布

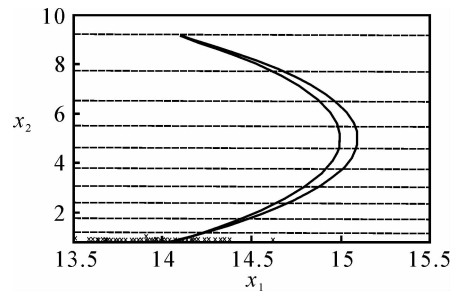


图 4 第 300 代时种群 1 中解的分布

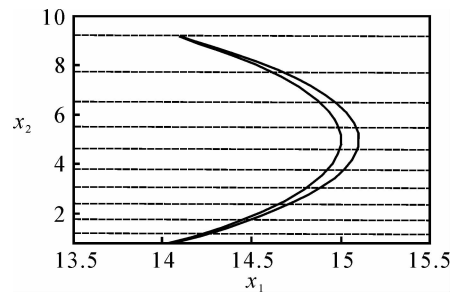


图 5 第 500 代时种群 1 中解的分布

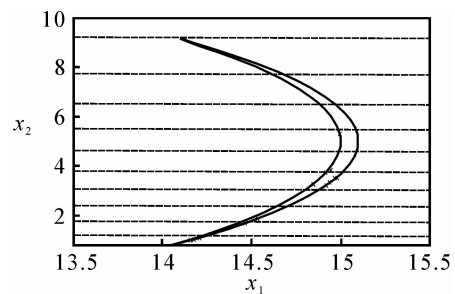


图 6 初始化结束时种群 2 中解的分布

最后收敛到最优可行解.

这些图形所显示的现象恰好符合算法的设计动机和机制,说明 HDDEP 能有效处理约束优化问题,包括可行域比例很小的强约束优化问题.

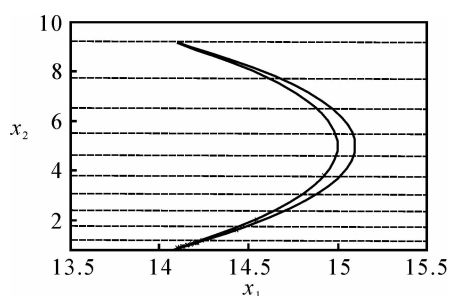


图7 第100代时种群2中解的分布

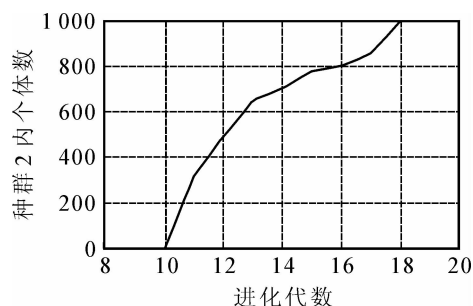


图8 种群2中解数量随进化的变化

6 实验与算法比较

将HDEDP算法用于求解国际上典型的3维、13维和40维ED问题.其中3维问题的最优解如表1所示,并与改进的进化规划算法(IFEP)^[4]进行比较.

表1 3维ED问题最优解

变量	P_1	P_2	P_3	f
数值	300.27	149.73	400.00	8234.07

HDEDP的参数如下:种群1规模为150,种群2规模为1000,最大进化代数为150, $F = 0.8$, $p_c = 0.1$.每代进化中SM算法连续执行15步,算法对各算例均运行50次.实验统计结果如表2~表4所示.

表2 3维ED问题统计结果比较

方法	最优解	平均解	最差解	标准偏差
IFEP	8234.07	8234.16	8234.54	NA
HDEDP	8234.07	8234.07	8234.07	0

表3 13维电力分配问题统计结果比较

方法	最优解	平均解	最差解	标准偏差
IFEP	17994	18127	18267	NA
HDEDP	17979	18131	18239	62.56

表4 40维电力分配问题统计结果比较

方法	最优解	平均解	最差解	标准偏差
IFEP	122624	123382	125741	NA
HDEDP	122177	122702	123102	228.02

由表2可见,对于3维ED问题,HDEDP算法每次运行都能找到最优解,各项性能均优于IFEP算法,算法鲁棒性良好.由表3可见,对于13维ED问题,HDEDP算法找到的最优解优于IFEP算法,但

平均解略劣于IFEP算法.这是因为当维数较小时,问题的不可行域相对较小,HDEDP算法的优势还未充分发挥.由表4可见,对于40维ED问题,HDEDP算法的各项性能均优于IFEP算法.因此,本文算法能适应问题维数增大的情况.另外,就计算量而言,HDEDP总评价次数大约在20000次左右,算法效率较高.

根据上述统计结果可得出结论:在求解ED问题时,HDEDP算法与当前最具代表性的算法相比,具有很强的搜索能力,能找到性能更优或满意解,并且效率较高,适于处理高维非线性约束优化问题.随着问题维数的增加,搜索空间扩大,尽管算法的鲁棒性稍有变差,但可通过适当扩大种群规模和增大进化代数,以增强对解空间的探索,提高算法的搜索质量.

7 结论

本文提出一种基于双种群的差分进化算法,可有效解决电力分配问题.实验结果显示了该算法的有效性,其性能优于该领域中的有效算法.HDEDP也是处理约束优化问题的一种有效框架,在设计上具有一定的柔性,通过对各模块的多样化设计,可形成更有效的约束优化技术.进一步的工作是将算法推广应用于其他工程约束优化问题.

参考文献(References)

- [1] David C, Walters G, Sheble B. Genetic algorithm solution of economic dispatch with value point loading [J]. IEEE Trans on Power Systems, 1993, 8(3): 1325-1332.
- [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] 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.)
- [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.