

混合整数非线性规划的自适应变异差分进化算法

车林仙^{1,2}, 何兵^{2,3a}, 刘永波^{3b}

(1. 重庆工程职业技术学院机械工程学院, 重庆 402260; 2. 人工智能四川省重点实验室, 四川 自贡 643000;
3. 泸州职业技术学院 a. 机械工程系; b. 信息工程系, 四川 泸州 646005)

摘要:提出了一种求解混合整数非线性规划(Mixed integer nonlinear programming, MINLP)的混合差分进化(Differential evolution, DE)算法。为提高DE算法的优化性能,设计了混沌初始化种群、可平衡全局探索与精细开采能力的混合变异版本、基于种群进化停滞代数记录的自适应二次变异算子等新型策略。将前述策略融入DE算法,形成面向MINLP的自适应变异差分进化(Adaptive mutation differential evolution, AMDE)算法。6个MINLP数值实例的对比实验表明了新算法的有效性和可靠性。最后,应用AMDE算法求解齿轮传动体积最小化工程优化设计实例,显示了该算法的工程应用价值。

关键词:混合整数非线性规划;差分进化算法;人工蜂群算法;自适应变异;齿轮传动优化设计

中图分类号:TP18;O221.4

文献标志码:A

引言

工程设计、系统工程和经济管理等领域存在许多混合整数非线性规划(Mixed integer nonlinear programming, MINLP)问题,它同时包含连续变量和整数变量,其目标函数和约束条件通常呈现强烈非线性。MINLP的求解方法主要有两类。一类为确定性方法^[1],如分支定界算法、外逼近算法、割平面算法和广义Bender分解方法等混合变量规划方法。但这类方法的求解过程通常较复杂,并且对数学理论基础有较高要求。另一类为智能优化算法,如遗传算法(Genetic algorithm, GA)、进化策略(Evolution strategy, ES)、粒子群优化(Particle swarm optimization, PSO)算法和差分进化(Differential evolution, DE)算法等。

很多学者针对MINLP,提出了多种智能优化方法。张甲江等^[2]、谭跃等^[3]分别提出适于MINLP的量子粒子群优化算法和基于交叉与多混沌策略的粒子群优化

算法。张莉等^[4]、邓长寿等^[5]、谭跃等^[6]和摆亮等^[7]先后设计多种面向MINLP的混合差分进化算法,如嵌入正交叉算子的差分进化算法^[4]、交替变异版本的差分进化算法^[5]、混沌局部搜索差分进化(DE with chaotic local search strategy, CLSDE)算法^[6]和差分进化分布估计混合算法^[7]。文献[2-6]仅进行了经典函数优化问题测试,并未给出工程应用实例。因此,面向MINLP的智能算法,尚未在工程优化领域显示出应用价值。

为提高DE算法求解MINLP的精度和可靠性,本文提出一种自适应变异差分进化(Adaptive mutation differential evolution, AMDE)算法。该算法采用混沌序列初始化种群,融合人工蜂群(Artificial bee colony, ABC)算法^[8-10]搜索方式和动态调整变异概率的二次变异,并采用可行性规则^[11]处理约束条件。最后,应用AMDE算法求解6个MINLP数值实例和1个二级斜齿圆柱齿轮传动优化设计案例,得到了满意结果。

收稿日期:2016-10-24

基金项目:重庆市基础科学与前沿技术研究专项资助项目(cstc2015jcyjA70006);重庆市教育委员会科学技术研究项目(KJ1403201);人工智能四川省重点实验室开放基金项目(2013RYJ02)

作者简介:车林仙(1971-),男,四川泸州人,教授,博士,主要从事机械优化设计、机构学及智能信息处理方面的研究,(E-mail)lx.che@163.com

1 MINLP 数学模型描述

不失一般性,将最小化 MINLP 问题表示为

$$\begin{cases} \min f(x, y) \\ \text{s. t. } g_i(x, y) \leq 0, i = 1, 2, \dots, I_1 \\ h_i(x, y) = 0, i = I_1 + 1, I_1 + 2, \dots, I_1 + I_2 \\ x_d \in [x_{L,d}, x_{U,d}], d = 1, 2, \dots, D_1 \\ y_d \in [y_{L,d}, y_{U,d}] \cap \mathbf{Z}, d = 1, 2, \dots, D_2 \end{cases} \quad (1)$$

式中, \mathbf{x} 为连续决策变量, $\mathbf{x} = (x_1, x_2, \dots, x_{D_1})^T$; \mathbf{y} 为整数决策变量, $\mathbf{y} = (y_1, y_2, \dots, y_{D_2})^T$; $x_{U,d}, x_{L,d}$ 为各维连续决策变量的上、下界; $y_{U,d}, y_{L,d}$ 为各维整数决策变量的上、下界; D_1, D_2 为连续、整数决策变量的维数; $f(x, y)$ 是目标函数; $g_i(x, y)$ 为不等式约束; $h_i(x, y)$ 为等式约束; I_1, I_2 为不等式、等式约束数。

为便于叙述,将决策矢量 (x, y) 统一记为混合整数决策矢量:

$$\mathbf{z} = (z_1, z_2, \dots, z_{D_1+D_2})^T = (x_1, x_2, \dots, x_{D_1}, y_1, y_2, \dots, y_{D_2})^T$$

并将 \mathbf{z} 的各维分量上下界记为 $z_{U,d}$ 和 $z_{L,d}$ ($d = 1, 2, \dots, D_1 + D_2$)。

应用 DE 算法求解 MINLP 式(1)时,涉及候选解(个体)比较与选择,本文应用 Deb 提出的可行性规则^[11]比较两个候选解。该规则为:(1)任何可行解都优于不可行解;(2)对于两个可行解,目标函数值较小者为优;(3)对于两个不可行解,约束违反度较小者为优。为此,将任意候选解 \mathbf{z} 的约束违反度函数 $f_{\text{vio}}(\mathbf{z})$ 定义为:

$$\begin{aligned} \boldsymbol{\omega}(\mathbf{z}) &= \sum_{i=1}^{I_1} \max\{0, g_i(\mathbf{z})\} + \sum_{i=I_1+1}^{I_1+I_2} \max\{0, |h_i(\mathbf{z})| - \varepsilon\} \\ f_{\text{vio}}(\mathbf{z}) &= [\boldsymbol{\omega}(\mathbf{z})]^\zeta \end{aligned}$$

式中, ε 为小正数,表示允许违反等式约束的容忍度; ζ 为强度指数,取 $\zeta(\boldsymbol{\omega}) = \begin{cases} 1, & 0 \leq \boldsymbol{\omega} \leq 1 \\ 2, & \boldsymbol{\omega} > 1 \end{cases}$ 。

2 混合变量 AMDE 算法

DE 算法的 4 个控制参数为:种群规模 N_p , 缩放因子 F , 交叉因子 C_R 和最大进化代数 t_{max} 。

应用 DE 算法求解式(1)时,将进化至第 t 代的第 n 个个体编码记为:

$$\begin{aligned} \mathbf{z}_n^{(t)} &= (z_{n,1}^{(t)}, z_{n,2}^{(t)}, \dots, z_{n,D_1+D_2}^{(t)})^T \\ t &= 0, 1, \dots, t_{\text{max}}; n = 1, 2, \dots, N_p \end{aligned}$$

另,将当前种群中的最优记为 $\mathbf{z}_b^{(t)} = (z_{b,1}^{(t)}, z_{b,2}^{(t)}, \dots, z_{b,D_1+D_2}^{(t)})^T$ 。

2.1 混沌初始化

Kent 映射生成的混沌序列分布均匀,且具有良好遍历性,本文应用该混沌序列生成初始种群。一维 Kent 映射可表示为:

$$q_{k+1,d} = \begin{cases} q_{k,d}/\mu_0, & 0 < q_{k,d} \leq \mu_0 \\ (1 - q_{k,d})/(1 - \mu_0), & \mu_0 < q_{k,d} < 1 \end{cases} \quad (2)$$

$$d = 1, 2, \dots, D_1 + D_2; k = 0, 1, \dots, N_1$$

式中: $\mu_0 \in (0, 1)$, 为常数,且 $\mu_0 \neq 0.5$, 本文取 $\mu_0 = 0.4$; $q_{0,d}$ 为 $(0, 1)$ 内服从均匀分布的随机量,且满足 $q_{0,d} \neq \mu_0$; N_1 为混沌序列个数。

由式(2)生成 N_1 个 $(D_1 + D_2)$ 维混沌序列点之后,再按下式载波为 N_1 个初始个体

$$z_{k,d}^{(0)} = z_{L,d} + (z_{U,d} - z_{L,d})q_{k,d} \quad k = 1, 2, \dots, N_1$$

对 $z_k^{(0)}$ 的第 $(D_1 + 1) \sim (D_1 + D_2)$ 维分量,应用 $\text{round}(\cdot)$ 函数执行四舍五入取整操作,即:

$$z_{k,d}^{(0)} = \text{round}(z_{k,d}^{(0)}) \quad d = D_1 + 1, D_1 + 2, \dots, D_1 + D_2$$

在 N_1 个初始个体中,应用 Deb 规则选择较优的 N_p 个体作为初始种群。

2.2 自适应变异概率

基本 DE 算法的种群多样性将随进化代数的增大而降低,求解某些多极值 MINLP 问题时,可能出现进化停滞和早熟收敛现象。为增强算法的全局优化能力,克服该缺陷,有必要对新生成的试验个体执行二次变异操作。当连续停滞代数较小时,宜设置较小的变异概率,避免过多无用操作而降低搜索效率;当连续停滞代数较大时,宜增大变异概率,利于保持种群多样性,增大获得全局最优解的概率。

定义 1^[12] 对于足够小的精度阈值 $\delta^{(t)}$ (随进化代数改变),若满足

$$|f(z_b^{(t-1)}) - f(z_b^{(t)})| \leq \delta^{(t)} \quad (3)$$

或

$$|f_{\text{vio}}(z_b^{(t-1)}) - f_{\text{vio}}(z_b^{(t)})| \leq \delta^{(t)} \quad (4)$$

则认为种群在第 t 代进化停滞。根据实践经验,取 $\delta^{(0)} = 10^{-5}$, $\delta^{(t)} = \delta^{(t-1)}/1.035$ 时,效果较理想。

若连续两代的最优个体均为可行解,则按式(3)判断;若连续两代的最优个体均为不可行解,则按式(4)判断;若连续两代的最优个体分别为不可行解和可行解,则认为种群在第 t 代进化未停滞。

算法设置计数器 t_{st} 记录种群连续停滞代数,随着 t_{st} 的变化,自适应调整变异概率 p_m , 其计算式为:

$$p_m^{(t)} = 0.01 + 0.09 \cdot \min\left\{1, \frac{t_{\text{st}}}{30}\right\} \quad (5)$$

由式(5)可知,当 t_{st} 较小时, p_m 也较小;当 t_{st} 增大

时, p_m 也随之增大,但最大不超过 0.1。因 p_m 过大,将加大随机搜索成分,反而降低搜索效率。

2.3 计算流程

计算流程各步骤及生成新解时,参照 2.1 节。

(1) 初始化

应用混沌初始化方法初始化种群,并应用 Deb 规则确定 $z_b^{(0)}$ 。置 $t=0, t_{st}=0$ 。

(2) 变异操作

DE 算法的关键在于先应用差分算子生成变异个体,再执行交叉操作生成试验个体。应用 DE/rand/1 策略生成变异个体时,算法具有较强全局探索能力,但局部精细开采能力偏弱。为提高收敛速度和精度,本文借鉴文献[10]中改进 ABC 算法的引领蜂搜索方式,引入一种基于最优个体的变异策略,记为 DE/rand/best-rand。该策略仍以随机个体为基向量,但差分向量指向最优个体,有利于精细开采且不易陷入局部最优区域。为了在全局探索和精细开采之间取得平衡,本文算法每次生成变异个体时,均在上述两种策略中随机选择一种。现将两种策略的变异算子描述如下:

(i) DE/rand/1 策略

$$u = z_{r_1}^{(t)} + F(z_{r_2}^{(t)} - z_{r_3}^{(t)}) \quad (6)$$

式中, $r_1^n, r_2^n, r_3^n \in [1, N_p] \cap \mathbf{Z} \setminus \{n\}$, 为随机互异自然数,且 $r_1^n \neq r_2^n \neq r_3^n$; F 为缩放因子,通常取 $F \in [0.4, 0.9]$ 。

(ii) DE/rand/best-rand 策略

$$u = z_{r_1}^{(t)} + R_{(D_1+D_2)} \otimes (z_b^{(t)} - z_{r_2}^{(t)})$$

式中, $R_{(D_1+D_2)}$ 为 $(-1, 1)$ 内服从均匀分布的 $(D_1 + D_2)$ 维随机数; \otimes 表示点对点乘法运算符。

若 u 之某维分量越界,则在其取值区间内随机生成该维分量。

(3) 交叉操作

变异个体 u 与父代个体 $z_n^{(t)}$ 执行交叉操作,生成试验个体 v ,其各维分量为:

$$v_d = \begin{cases} u_d, \text{rand}() < C_R \vee d = d_r \\ z_{n,d}^{(t)}, \text{其它} \end{cases}$$

式中, $\text{rand}()$ 表示 $(0, 1)$ 内服从均匀分布的随机数; d_r 为随机自然数,且 $d_r \in [1, D] \cap \mathbf{Z}$; C_R 为交叉因子,且 $C_R \in [0, 1]$ 。

(4) 二次变异操作

应用式(5)确定 $p_m^{(t)}$, 对 v 的每维分量按概率 $p_m^{(t)}$ 执行变异,生成候选个体 w ,其各维分量为:

$$w_d = \begin{cases} M(v_d), \text{rand}() < p_m^{(t)} \\ v_d, \text{其它} \end{cases}$$

式中, $M(\cdot)$ 为二次变异算子。

为增强算法的适应性,文中对连续变量和整数变量,分别执行非均匀变异和均匀变异操作。即,当 v_d ($d=1, 2, \dots, D_1$) 为连续变量时,二次变异算子为:

$$M(v_d) = \begin{cases} v_d + (z_{U,d} - v_d)[1 - r^{(1-r/T)^2}], \text{rand}() < 0.5 \\ v_d - (v_d - z_{L,d})[1 - r^{(1-r/T)^2}], \text{其他} \end{cases}$$

式中, r 表示在 $(0, 1)$ 内服从均匀分布的随机数。

当 v_d ($d = D_1 + 1, D_1 + 2, \dots, D_1 + D_2$) 为整数变量时,又分为两种情形:若为 0-1 整数规划问题,二次变异算子为:

$$M(v_d) = 1 - v_d$$

若为一般整数规划问题,二次变异算子为:

$$M(v_d) = z_{L,d} + \text{round}[\text{rand}() \cdot (z_{U,d} - z_{L,d})]$$

(5) 选择操作

应用 Deb 规则比较 $z_n^{(t)}$ 和 w ,通过贪婪选择生成子代个体,选择算子为:

$$z_n^{(t+1)} = \begin{cases} w, \text{若 } w \text{ 优于 } z_n^{(t)} \\ z_n^{(t)}, \text{其它} \end{cases}$$

对 $n = 1, 2, \dots, N_p$,重复执行 N_p 次步骤(2)~(5)。

(6) 更新最优个体

应用 Deb 规则确定 $z_b^{(t+1)}$ 。若种群进化停滞,则置 $t_{st} = t_{st} + 1$, 否则,置 $t_{st} = 0$ 。

(7) 终止判断

若满足终止条件,则输出 $z_b^{(t+1)}$, 并结束;否则,置 $t = t + 1$, 转步骤(2)。

2.4 计算复杂度

据第 2.3 节的计算流程分析 AMDE 算法的计算复杂度。步骤(1)的计算复杂度为 $O(N_1 D)$, 步骤(2)~(4)的计算复杂度均为 $O(N_p D)$; 步骤(5)的计算复杂度为 $O(N_p)$; 步骤(6)的计算复杂度:搜索 N_p 个个体中的最优个体为 $O(N_p)$, 更新连续停滞代数计数器为 $O(1)$ 。求和并忽略低阶项,可求得 AMDE 算法的计算复杂度为 $O(N_p D t_{\max} + N_1 D)$ 。算法独立运行 1 次的函数评价次数为 $N_1 + N_p t_{\max}$ 。

3 MINLP 算例测试

3.1 测试实例

以下 6 个测试实例均引自文献[6]。

$$P1: \min f(x, y) = 2x + y$$

$$\text{s. t. } g_1(x, y) = 1.25 - x^2 - y \leq 0$$

$$g_2(x, y) = x + y - 1.6 \leq 0$$

$$0 \leq x \leq 1.6, y \in \{0, 1\}$$

已知最优解 $(x, y; f^*) = (0.5, 1; 2)$ 。

$$P2: \min f(x, y) = 2x - \ln(x/2) - y$$

$$\text{s. t. } g(x, y) = -x - \ln(x/2) + y \leq 0$$

$$0.5 \leq x \leq 1.4, y \in \{0, 1\}$$

已知最优解 $(x, y; f^*) = (1.375, 1; 2.124)$ 。

$$P3: \min f(x, y) = 5(x_1 - 0.5)^2 - 0.7y + 0.8$$

$$\text{s. t. } g_1(x, y) = -\exp(x_1 - 0.2) - x_2 \leq 0$$

$$g_2(x, y) = x_2 + 1.1y + 1 \leq 0$$

$$g_3(x, y) = x_1 - 1.2y - 0.2 \leq 0$$

$$0.2 \leq x_1 \leq 1, -2.22554 \leq x_2 \leq -1, y \in \{0, 1\}$$

已知最优解 $(x, y; f^*) = (0.94194, -2.1, 1;$

1.07654)。

$$P4: \min f(x, y) =$$

$$\begin{cases} 5.5 + 6x + \frac{50}{0.8[1 - \exp(-0.4x)]}, y = 0 \\ 7.5 + 7x + \frac{50}{0.9[1 - \exp(-0.5x)]}, y = 1 \end{cases}$$

$$\text{s. t. } g(x, y) =$$

$$\begin{cases} \frac{10}{0.8[1 - \exp(-0.4x)]} - 20 \leq 0, y = 0 \\ \frac{10}{0.9[1 - \exp(-0.5x)]} - 20 \leq 0, y = 1 \end{cases}$$

$$0 \leq x \leq 10, y \in \{0, 1\}$$

已知最优解 $(x, y; f^*) = (3.51424, 1; 99.23964)$ 。

$$P5: \min f(x, y) = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (y_1 - 1)^2 + (y_2 - 1)^2 + (y_3 - 1)^2 - \ln(y_4 + 1)$$

$$\text{s. t. } g_1(x, y) = x_1 + x_2 + x_3 + y_1 + y_2 + y_3 - 5 \leq 0$$

$$g_2(x, y) = x_1^2 + x_2^2 + x_3^2 + y_3^2 - 5 \leq 0$$

$$g_3(x, y) = x_1 + y_1 - 1.2 \leq 0$$

$$g_4(x, y) = x_2 + y_2 - 1.8 \leq 0$$

$$g_5(x, y) = x_3 + y_3 - 2.5 \leq 0$$

$$g_6(x, y) = x_1 + y_4 - 1.2 \leq 0$$

$$g_7(x, y) = x_2^2 + y_2^2 - 1.64 \leq 0$$

$$g_8(x, y) = x_3^2 + y_3^2 - 4.25 \leq 0$$

$$g_9(x, y) = x_3^2 + y_2^2 - 4.64 \leq 0$$

$$0 \leq x_1 \leq 1.2, 0 \leq x_2 \leq 1.8, 0 \leq x_3 \leq 2.5$$

$$y_1, y_2, y_3, y_4 \in \{0, 1\}$$

已知最优解 $(x, y; f^*) = (0.2, 1.280624, 1.954483,$

1, 0, 0, 1; 3.557463)。

$$P6: \max f(x, y) = -5.357854x_1^2 - 0.835689x_3y_1 - 37.29329y_1 + 40792.141$$

$$\text{s. t. } g_1(x, y) = a_1 + a_2x_3y_2 + a_3x_2y_1 - a_4x_1x_3 - 92 \leq 0$$

$$g_2(x, y) = a_5 + a_6x_3y_2 + a_7y_1y_2 + a_8x_1^2 - 110 \leq 0$$

$$g_3(x, y) = a_9 + a_{10}x_1x_3 + a_{11}x_1y_1 + a_{12}x_1x_2 - 25 \leq 0$$

$$27 \leq x_1, x_2, x_3 \leq 45, y_1 \in [78, 102] \cap Z,$$

$$y_2 \in [33, 45] \cap Z$$

已知最优解 $(x, y; f^*) = (27, *, 27, 78, *, 32217.42778)$ (*表示最优解不确定)。对应系数取值见表1。

表1 P6中的系数取值

系数	取值	系数	取值	系数	取值
a_1	85.334407	a_5	80.51249	a_9	9.300961
a_2	0.0056858	a_6	0.0071317	a_{10}	0.0047026
a_3	0.0006262	a_7	0.0029955	a_{11}	0.0012547
a_4	0.0022053	a_8	0.0021813	a_{12}	0.0019085

3.2 实验设置

AMDE算法控制参数: N_p 和 t_{\max} 取值见表2, $N_1 = 100N_p, F = 0.5 + 0.05\text{rand}(\cdot), C_R = 0.9$ 。

表2 AMDE算法的控制参数

参数	实例					
	P1	P2	P3	P4	P5	P6
N_p	30	30	30	50	30	50
t_{\max}	100	100	200	50	300	200

为考察AMDE算法求解测试实例的优化性能,文中还以PSO算法,人工蜂群(Artificial bee colony, ABC)算法^[11-12]和基本DE算法(第2.3节的算法流程中,步骤(2)仅用式(6)生成变异个体,且不执行步骤(4))等作为对比算法,进行比较分析。

为公平起见,3种对比算法均采用与AMDE算法相同的初始化、约束函数、取整和越界处理方法;对比算法的 N_p 和 t_{\max} 均与AMDE算法一致。DE算法的 F, C_R 与AMDE算法一致。

PSO算法的惯性权重 w 线性递减,取 $w = 1.2 \sim 0.2$,加速度系数为 $c_1 = c_2 = 2$,最大速度取 $v_{\max, d} = 0.3(z_{U, d} - z_{L, d})$ ($d = 1, 2, \dots, D_1 + D_2$)。

ABC算法的引领蜂和跟随蜂各占种群规模的50%;跟随蜂采用2元锦标赛方法选择被跟随的引领蜂;引领蜂和跟随蜂执行邻域搜索生成新蜜源时,均以概率 C_R 随机改变多维分量。

3.3 实验结果

4种算法分别独立运行50次的统计结果见表3。其中 f_b, f_{av}, f_w 和 σ_f 分别表示最优目标函数值的最好、平均、最差值及标准差。同时,将CLSDE算法^[9]的性能指标也列入表5,以便比较。

表 3 5 种算法求解实例的性能指标

算例	算法	最好值 f_b	平均值 f_{av}	最差值 f_w	标准差 σ_f
P1	PSO	2.000000	2.000000	2.000006	1.152×10^{-6}
	ABC	2.000000	2.000000	2.000000	0
	DE	2.000000	2.000000	2.000000	0
	CLSDE ^[6]	2.0000	2.0000	2.0000	2.341×10^{-16}
	AMDE	2.000000	2.000000	2.000000	0
P2	PSO	2.124468	2.124687	2.125395	2.415×10^{-4}
	ABC	2.124468	2.124468	2.124468	4.441×10^{-16}
	DE	2.124468	2.124682	2.126523	4.783×10^{-4}
	CLSDE ^[6]	2.1245	2.1245	2.1245	4.681×10^{-16}
	AMDE	2.124468	2.124468	2.124468	4.486×10^{-16}
P3	PSO	1.076545	1.079864	1.140703	9.948×10^{-3}
	ABC	1.076545	1.076720	1.083438	9.849×10^{-4}
	DE	1.076543	1.083796	1.240230	2.859×10^{-2}
	CLSDE ^[6]	1.0769	1.0777	1.0787	5.8905×10^{-4}
	AMDE	1.076543	1.076543	1.076543	7.958×10^{-13}
P4	PSO	99.239635	99.239635	99.239635	1.456×10^{-8}
	ABC	99.239635	99.239635	99.239635	4.960×10^{-14}
	DE	99.239635	99.239635	99.239635	4.931×10^{-14}
	CLSDE ^[6]	99.2396	99.2396	99.2396	1.498×10^{-14}
	AMDE	99.239635	99.239635	99.239635	5.187×10^{-14}
P5	PSO	3.561538	3.725333	4.276274	1.583×10^{-1}
	ABC	3.557467	3.558061	3.560028	6.744×10^{-4}
	DE	3.557461	3.557461	3.557461	1.573×10^{-13}
	CLSDE ^[6]	3.5755	3.6287	3.6926	3.740×10^{-2}
	AMDE	3.557461	3.557461	3.557461	6.615×10^{-9}
P6	PSO	32210.16420	32128.85373	32073.03268	3.086×10^1
	ABC	32217.42778	32217.42778	32217.42778	1.877×10^{-11}
	DE	32217.42778	32217.42778	32217.42778	1.877×10^{-11}
	CLSDE ^[6]	3.2217×10^4	3.2217×10^4	3.2217×10^4	6.930×10^{-2}
	AMDE	32217.42778	32217.42778	32217.42778	1.929×10^{-11}

为了便于应用对数坐标清晰显示各种算法的收敛精度,首先给出关于算法种群最小误差的概念。

定义 2 设待求 MINLP 问题的理论最优目标函数值为 f^* , 定义第 t 代种群的最小误差为:

$$e_b^{(t)} = |f(z_b^{(t)}) - f^*|$$

定义 3 设算法求解某问题时,独立运行 N_{run} 次,第 j 次第 t 代种群的最小误差为 $e_{b,j}^{(t)}$, 定义第 t 代平均最小误差为:

$$e_{b,av}^{(t)} = \frac{1}{N_{\text{run}}} \sum_{j=1}^{N_{\text{run}}} e_{b,j}^{(t)}$$

4 种算法的平均最小误差 $e_{b,av}$ (图中取对数) 进化曲线对比如图 1 所示。

由表 3 和图 1 可知,求解 P1、P4、P5 和 P6 时,PSO 算法的 $e_{b,av}$ 最大,求解 P2 和 P3 时,PSO 算法的 $e_{b,av}$ 与 DE 算法相当,但大于 ABC 和 AMDE 算法;求解 6 个实例时,PSO 算法的 σ_f 均最大,稳健性最差。在 5 种算法中,PSO 算法的优化性能最差。AMDE 与 ABC 算法比较,求解 P1、P2、P4 和 P6 时,2 种算法相当;而求解 P3 和 P5 时,AMDE 明显优于 ABC 算法。AMDE 与 DE 算法比较,求解 P1、P4 和 P6 时,2 种算法相当;求解 P2 和

P3 时,AMDE 明显优于 DE 算法;求解 P5 时,AMDE 差于 DE 算法。求解 6 个实例时,AMDE 算法的 $e_{b,av}$ 和 σ_f 均很小,表明其收敛精度高、稳健性好。对于其余 4 种对比算法,无一算法在求解 6 个实例时同时具有良好优化性能。由表 2 可知,AMDE 算法求解实例时的最大函数评价次数为 15 000,而 CLSDE 算法为 52 000^[6],因此前者的计算效率更高。综上可知,AMDE 算法的性能指标优于 4 种对比算法。

4 齿轮传动优化设计应用

4.1 齿轮传动优化设计模型

已知某企业使用的二级斜齿圆柱齿轮传动原始数据为:高速轴输入功率 $P_1 = 8 \text{ kW}$;高速级主动轴转速 $n_1 = 1450 \text{ r/min}$;传动比 $i_{12} = 9.7$,允许传动比误差 $\pm 5\%$;动力机为电动机,工作机载荷均匀平稳。大、小齿轮材料为 45 钢,大齿轮正火处理,硬度 185 ~ 205 HBS;小齿轮调质处理,硬度为 230 ~ 255 HBS。许用应力 $[\sigma_H]_I = [\sigma_H]_{II} = 520 \text{ MPa}$, $[\sigma_F]_I = 155 \text{ MPa}$, $[\sigma_F]_{II} = 145 \text{ MPa}$ 。载荷系数 $K = 1.25$ 。要求以齿轮总体积最小为目标,设计该齿轮传动机构。

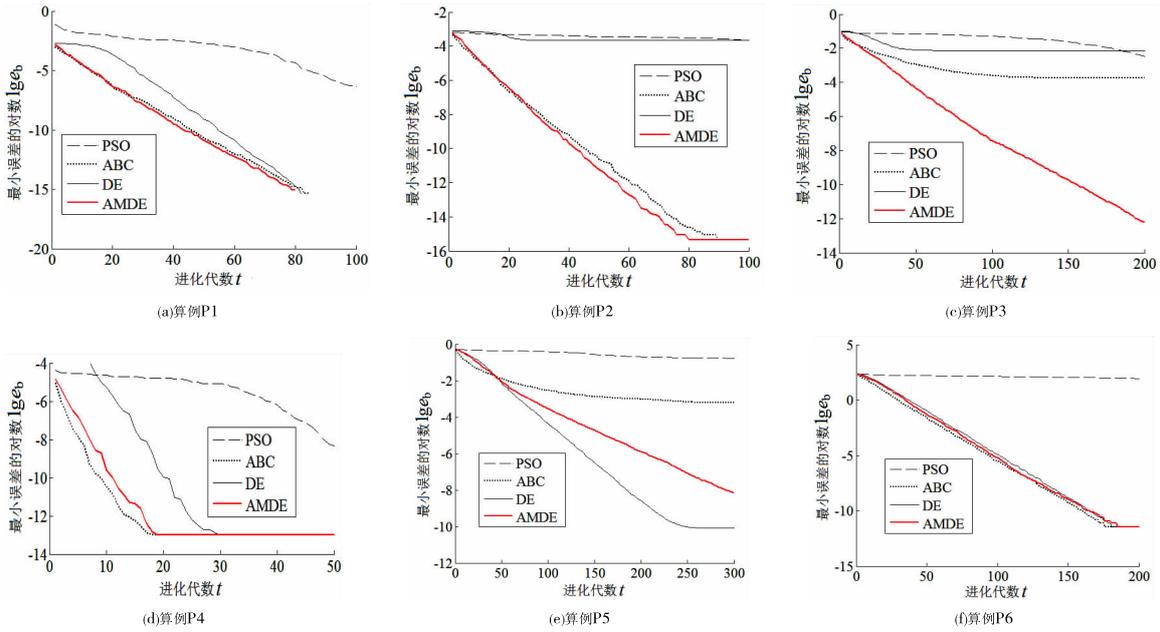


图1 平均最小误差进化曲线

该问题的混合设计变量为:第 I 级传动,法面模数 m_{nI} 、小齿轮齿数 z_1 、螺旋角 β_1 、齿宽系数 ψ_{dI} 和传动比 i_1 ; 第 II 级传动,法面模数 m_{nII} 、小齿轮齿数 z_3 、螺旋角 β_{II} 和齿宽系数 ψ_{dII} 。其中, m_{nI} 和 m_{nII} 为非等间隔离散变量, z_1 和 z_3 为整数变量; β_1 、 β_{II} 、 ψ_{dI} 、 ψ_{dII} 和 i_1 为连续变量。将决策变量记为 $x = (x_1, x_2, \dots, x_5)^T = (\beta_1, \beta_{II}, \psi_{dI}, \psi_{dII}, i_1)^T$, $y = (y_1, y_2, y_3, y_4)^T = (m_{nI}, m_{nII}, z_1, z_3)^T$ 。可根据国家标准,建立模数的离散变量取值列表,再映射为整数变量^[12-14]。故,该问题可转化为等效 MINLP,再应用第 2 节的方法求解。

若以齿轮分度圆柱体积近似表示齿轮体积,则总体积最小化设计问题的目标函数为:

$$\min f(x, y) = \frac{\pi}{4} \left[\frac{y_1^3 (y_3^2 + z_2^2) y_3 x_3}{\cos^3 x_1} + \frac{y_2^3 (y_4^2 + z_4^2) y_4 x_4}{\cos^3 x_2} \right]$$

式中, z_2 表示第 I 级传动大齿轮齿数,取 $z_2 = \text{round}(x_5 y_3)$; z_4 表示第 II 级传动大齿轮齿数,取 $z_4 = \text{round}(i_{tr} y_3 y_4 / z_2)$ 。

参照文献[12],列出以下约束函数。其中, $g_1 \sim g_4$ 为强度条件, g_5 为避免第 II 级大齿轮与第 I 级传动轴发生干涉的限制条件, $g_6 \sim g_7$ 为最小齿数约束, $g_8 \sim g_9$ 为轴向重合度约束, g_{10} 为传动比精度约束, $g_{11} \sim g_{14}$ 为齿宽约束。

$$g_1(x, y) = 2KT_1 \left(\frac{Z_{E1} Z_{H1} Z_{\alpha1} Z_{\beta1}}{[\sigma]_{H1}} \right)^2 - \left(\frac{y_1 y_3}{\cos x_1} \right)^3 \frac{x_3 x_5}{x_5 + 1} \leq 0$$

$$g_2(x, y) = 2KT_3 \left(\frac{Z_{EII} Z_{HII} Z_{\alphaII} Z_{\betaII}}{[\sigma]_{HII}} \right)^2 - \left(\frac{y_2 y_4}{\cos x_2} \right)^3 \frac{x_4 i_{tr}}{x_5 + i_{tr}} \leq 0$$

$$g_3(x, y) = 2KT_1 Y_{\alpha1} Y_{\beta1} \left(\frac{Y_{FS1}}{[\sigma]_{F1}} \right) - \frac{x_3 y_1^3 y_3^2}{\cos^2 x_1} \leq 0$$

$$g_4(x, y) = 2KT_3 Y_{\alphaII} Y_{\betaII} \left(\frac{Y_{FSII}}{[\sigma]_{FII}} \right) - \frac{x_4 y_2^3 y_4^2}{\cos^2 x_2} \leq 0$$

式中, T_1, T_3 为第 I, II 级传动计算转矩, $N \cdot mm$, 且 $T_1 = 9.55 \times 10^6 P_1 / n_1, T_3 = 9.55 \times 10^6 x_5 P_1 / n_1$; Z_{E1}, Z_{EII} 为第 I, II 级传动弹性系数; Z_{H1}, Z_{HII} 为第 I, II 级传动节点区域系数; $Z_{\alpha1}, Z_{\alphaII}$ 为第 I, II 级传动重合度系数; $Z_{\beta1}, Z_{\betaII}$ 为第 I, II 级传动螺旋角系数; $Y_{\alpha1}, Y_{\alphaII}$ 为第 I, II 级传动重合度系数; $Y_{\beta1}, Y_{\betaII}$ 为第 I, II 级传动螺旋角系数; Y_{FS1}, Y_{FSII} 为第 I, II 级传动复合齿形系数。

以上系数的确定方法见文献[15],其中,将 Y_{FS1}, Y_{FSII} 表达为当量齿数的拟合函数。

$$g_5(x, y) = y_1 z_2 \cos x_2 + 2(y_1 + 50) \cos x_1 \cos x_2 - y_2 (y_4 + z_4) \cos x_1 \leq 0$$

$$g_6(x, y) = 17 \cos^3 x_1 - y_3 \leq 0$$

$$g_7(x, y) = 17 \cos^3 x_2 - y_4 \leq 0$$

$$g_8(x, y) = \pi - x_3 y_3 \tan x_1 \leq 0$$

$$g_9(x, y) = \pi - x_4 y_4 \tan x_2 \leq 0$$

$$g_{10}(x, y) = \left| \frac{z_2}{z_1} \cdot \frac{z_4}{z_3} - i_{tr} \right| - 0.05 i_{tr} \leq 0$$

$$g_{11}(x, y) = 40 - x_3 y_1 y_3 \sec x_1 \leq 0$$

$$g_{12}(x, y) = x_3 y_1 y_3 \sec x_1 - 120 \leq 0$$

$$g_{13}(x, y) = 40 - x_4 y_2 y_4 \sec x_2 \leq 0$$

$$g_{14}(x, y) = x_4 y_2 y_4 \sec x_2 - 120 \leq 0$$

设计变量的取值范围设置为: $0.1396 \text{ rad} \leq x_1, x_2 \leq 0.2618 \text{ rad}, 0.6 \leq x_3, x_4 \leq 1.2, 2 \leq x_5 \leq 4, y_1 \in \{2, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4, 4.5, 5\} \text{ mm}, y_2 \in \{3.5, 3.75, 4, 4.5, 5, 5.5, 6\} \text{ mm}, y_3, y_4 \in [15, 30] \cap \mathbb{Z}$ 。

4.2 优化设计结果

应用 AMDE 算法求解该设计实例,除取 $N_p = 100$ 及 $t_{\max} = 1500$ 外,其余参数设置同第 3.2 节。算法独立运行 10 次,最好、最差和平均最优目标函数值分别为 $2.7623 \text{ dm}^3, 2.7758 \text{ dm}^3$ 和 2.7690 dm^3 , 标准差为 0.003949 dm^3 。该标准差很小,表明算法具有良好稳健性。平均最优值进化曲线如图 2,由图可知,算法大约进化 1200 代即收敛于近优解。

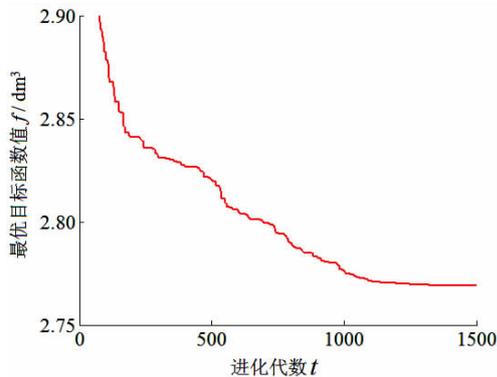


图 2 AMDE 算法求解案例的平均最优值进化曲线

AMDE 算法的最优解见表 4,将原设计方案也列入表中,以便比较。由表 4 可知,应用 AMDE 算法求得的结果明显优于原设计方案,齿轮体积和较原设计下降约 25.5%。

表 4 优化设计与原方案结果比较

设计变量或目标函数	AMDE 算法	原设计
第 I 级法面模数 m_{n1}/mm	2	3
第 I 级小齿轮齿数 z_1	22	17
第 I 级齿宽系数 ψ_{d1}	0.8873	0.8631
第 I 级螺旋角 $\beta_1/^\circ$	12.565	12.000
第 I 级传动比 i_1	3.3016	3.5294
第 II 级法面模数 m_{nII}/mm	3.5	4
第 II 级小齿轮齿数 z_3	21	19
第 II 级齿宽系数 ψ_{dII}	0.6110	0.7722
第 II 级螺旋角 $\beta_{II}/^\circ$	15.000	12.000
齿轮体积 f/dm^3	2.7623	3.7083

5 结论

(1) 为提高 DE 算法求解 MINLP 的优化性能,设计

混沌初始化种群、可平衡全局探索与精细开采能力的混合变异版本、基于种群进化停滞代数记录的自适应二次变异算子等新型策略。将诸策略融入 DE 算法,形成求解 MINLP 的混合智能算法 AMDE。

(2) 应用 6 个 MINLP 数值实例测试 AMDE 算法,结果表明:AMDE 算法可靠、有效,稳健性优于 4 种对比算法。

(3) 以二级斜齿圆柱齿轮传动体积最小化设计问题,验证 AMDE 算法求解混合变量工程优化问题的可行性。根据映射关系,将混合离散变量约束优化问题转化为等效 MINLP 问题,再应用 AMDE 算法求解该问题。结果显示:AMDE 算法可行、有效,具有良好稳健性。与原设计方案相比,AMDE 算法的最优解下降了约 25.5%。

参考文献:

- [1] 刘明明,崔春风,童小娇,等.混合整数非线性规划的算法软件及最新进展[J].中国科学:数学,2016,46(1):1-20.
- [2] 张甲江,高岳林,高晨阳.非线性混合整数规划问题的改进量子粒子群算法[J].太原理工大学学报,2015,46(2):196-200.
- [3] 谭跃,谭冠政,邓曙光.基于遗传交叉和多混沌策略改进的粒子群优化算法[J].计算机应用研究,2016,33(12):3643-3647.
- [4] 张莉,李宏,冯大政.求解混合整数规划的嵌入正交杂交的差分进化算法[J].系统工程与电子技术,2011,33(9):2126-2132.
- [5] 邓长寿,任红卫,彭虎.混合整数非线性规划问题的改进差分进化算法[J].计算机应用研究,2012,29(2):445-448.
- [6] 谭跃,谭冠政,杨冰,等.解决混合整数非线性规划问题的混沌局部搜索差分进化算法[J].小型微型计算机系统,2012,33(6):1306-1309.
- [7] BAI L, WANG J Y, JIANG Y H, et al. Improved hybrid differential evolution-estimation of distribution algorithm with feasibility rules for NLP/MINLP engineering optimization problems[J]. Chinese Journal of Chemical Engineering, 2012, 20(6):1074-1080.
- [8] AKAY B, KARABOGA D. A modified artificial bee colony algorithm for real-parameter optimization[J]. Information Sciences, 2012, 192:120-142.
- [9] GAO W F, LIU S Y, HUANG L L. A global best arti-

- cial bee colony algorithm for global optimization[J]. Journal of Computational and Applied Mathematics, 2012,236(11):2741-2753.
- [10] 罗钧,林子晴,刘学明,等.改进蜂群算法及其在圆度误差评定中的应用[J].机械工程学报,2016,52(16):27-32.
- [11] MALLIPEDDI R,SUGANTHAN P N.Ensemble of constraint handling techniques[J].IEEE Transactions on Evolutionary Computation,2010,14(4):561-579.
- [12] 车林仙.面向机构分析与设计的差分进化算法研究[D].徐州:中国矿业大学,2012.
- [13] 车林仙,程志红.工程约束优化的自适应罚函数混合离散差分进化算法[J].机械工程学报,2011,47(3):141-151.
- [14] 车林仙.多样性保持离散差分进化算法及齿轮传动优化应用[J].机械工程学报,2016,52(21):44-55.

Adaptive Mutation Differential Evolution Algorithm for Mixed Integer Nonlinear Programming

CHE Linxian^{1,2}, HE Bing^{2,3a}, LIU Yongbo^{3b}

(1. School of Mechanical Engineering, Chongqing Vocational Institute of Engineering, Chongqing 402260, China;

2. Artificial Intelligence Key Laboratory of Sichuan Province, Zigong 643000, China; 3a. Department of Mechanical Engineering; 3b. Department of Information Engineering, Luzhou Vocational and Technical College, Luzhou 646005, China)

Abstract: A hybrid differential evolution (DE) algorithm is presented to solve the mixed integer nonlinear programming (MINLP). To enhance the optimization performance of DE algorithm, several new strategies are designed, such as chaotic initializing population, hybrid mutation scheme which can balance global exploration and fine mining abilities, and adaptive second mutation operator based on the recording generations of evolutionary stagnation for a population. The aforementioned strategies are embedded in DE algorithm and an adaptive mutation DE (AMDE) algorithm is formed for solving MINLP. Six numerical examples of MINLP are tested comparatively to show that this new approach is valid and reliable. Finally the proposed AMDE algorithm is used to solve a real case of engineering optimization for minimizing volumes of gear transmission, so the practical applicability of the new algorithm is indicated.

Key words: mixed integer nonlinear programming; differential evolution algorithm; artificial bee colony algorithm; adaptive mutation operator; optimal design of gear transmission