第 28 卷第 11 期
2011 年 11 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 28 No. 11
Nov. 2011

文章编号: 1000−8152(2011)11−1613−08

# 基于N阶近邻分析的自适应差分进化算法

洪 榛, 张贵军, 俞 立

(浙江工业大学 信息工程学院, 浙江 杭州 310023)

**摘要**: 针对差分进化算法在求解多模优化问题解可靠性较低的问题, 在N阶近邻理论分析及参数整定的基础上, 提出一种基于N阶近邻分析的自适应差分进化算法(N−NNADE). N−NNADE算法在缺少先验知识的情况下, 通过分析群体个体间的N阶最短近邻计算种群的全局分布, 并利用阶跃信息自适应统计获得种群数量; 同时采用$K$−means算法划分种群, 进一步引入不同种群间的交叉变异思想以及父子代同种群则替换最差个体的选择策略实现种群间的协同进化. 通过获取更多的全局最优解和部分高质量的局优解来提高算法的可靠性. 20个优化问题的数值研究结果表明N−NNADE算法具有比DE(differential evolution), DERL(differential evolution algorithm with random localizations), ADE(adaptive differential evolution)算法更适合求解复杂的高维多模优化问题.

**关键词**: 多模优化; 差分进化; N阶近邻; $K$−means聚类

**中图分类号**: TP391 **文献标识码**: A

# Adaptive differential evolution algorithm based on $N$th-order nearest-neighbor analysis

HONG Zhen, ZHANG Gui-jun, YU Li

(College of Information Engineering, Zhejiang University of Technology, Hangzhou Zhejiang 310023, China)

**Abstract:** To improve the reliability of differential evolution(DE) algorithm in dealing with multimodal optimization problem, we propose an adaptive differential evolution(ADE) algorithm based on the $N$th-order nearest-neighbor analysis($N$−NNADE). Global distribution information of species is obtained by analyzing the $N$th-order nearest-neighbor in population, and the number of species is adaptively determined by the step-jumping information in lacking prior knowledge, Furthermore, the $K$−means algorithm is used for partitioning the population. To realize the co-evolution among species, we introduce the crossover mutation among different species and replace the worst members of current species if parents and children are belonging to the same species. The reliability of the algorithm is continuously improved by acquiring more global optimal solutions and high-quality local suboptimal solutions. The results of 20 benchmark optimization problems show that $N$−NNADE algorithm is more suitable than DE, DERL(differential evolution algorithm with random localizations) and ADE for solving complex high-dimensional multimodal optimization problems

**Key words:** multimodal optimization; differential evolution; Nth-order nearest neighbor; $K$−means clustering

## 1 引言(Introduction)

多模(多极值)全局优化问题的解向量空间一般存在一个或多个全局最优解和大量局部最优解[1], 此类问题通常可以描述成:

$$\min f(x_i), x_i \in \Omega \subset \mathbb{R}^n,\ i=1,2,\cdots,D, \quad (1)$$

其中: $x$为可行域$\Omega \subset \mathbb{R}^n$上的$D$维连续(不一定光滑)的向量, 可行域定义为$[x_{i,\min}, x_{i,\max}]$区间, $f(x_i)$是定义在$\Omega \subset \mathbb{R}^n$上的连续实值目标函数. 这类问题广泛存在于光纤通信、电磁设备设计、生物医药等领域[2~5], 其主要面临的困难是: 1) 找到全局最优解, 避免过早陷入局部解(早熟); 2) 在解空间搜索全局最优解或者高质量可供选择的局部最优解. 现有诸如基于梯度的传统优化方法[6]、遗传算法(genetic algorithm, GA)等的进化策略由于其全局选择压力往往最后只能收敛至(某)一个最优解[7], 且极有可能是局优解.

差分进化(differential evolution, DE)是Storn等[8]于1997年提出的一种基于群体集求解非线性、非可微的连续空间函数问题的启发式算法, ALI等[9]已经证明其比GA、控制随机搜索(controlled random search, CRS)等搜索算法拥有更好的全局搜索能力和收敛速度. 但是THOMSEN[7]、KAELO等[10]研究表明, DE算法不能同时搜索定位若干个全局最优解

收稿日期: 2010−03−22; 收修改稿日期: 2011−04−15.

基金项目: 国家自然科学基金资助项目(61075062, 60974017); 浙江省自然科学基金资助项目(Y1100891); 浙江省科技厅创新团队子项目资助项目(2011R09007-09).

或者是高质量的局部最优解, 解的可靠性也随之降低, 且在进化后期也容易出现收敛缓慢的问题. 为此, KAELO等[10]在基本DE的基础上提出了基于锦标赛机制同时结合采用反射与收缩算子的改进算法DERL(differential evolution algorithm with random localizations)和DERB; SUN等[11]基于概率模型提出了EDA算法, 并与原有DE算法结合使用, 因为额外的全局统计信息克服DE随机进化所带来的盲目性; SALMAN等[12]在参数调整方面入手, 提出了改进的自适应DE算法, 引入正态分布对控制参数做自动调整, 取得了一定的效果. 上述针对DE的改进算法, 往往倾向于在变异、交叉两个步骤上通过改进变异、交叉算子以及初始向量的选择策略来提高算法本身的收敛性和成功率, 却忽视了算法的可靠性; 其选择策略中倾向于用较好子代替代父代的思想也在一定程度上限制了算法的搜索能力; 此外, 对于大规模的高维多模问题和实际问题而言, 由于算法的特性均最终收敛于一点, 极易陷入早熟现象.

而针对多模函数问题特性, LOPZE CRUZ等[13]将高效DE用于解决一类多模最优控制问题; THOMSEN[7]在拥挤策略的基础上扩展DE算法来实现追踪与保持多个最优解; SHIBASAK等[14]按照物种形成原理将群体分割成多个种群, 并采用全局与局部相结合的搜索方式获取多个最优解. 上述虽已考虑到多模特性, 但是算法的效率和解的可靠性普遍较低, 无法保证其在大规模多模问题与实际问题中的应用. 因此, 如何提高解的可靠性, 在可行域范围内搜索尽可能多的优质解, 是工程领域中亟待解决的一个重要问题.

本文在基本DE算法的基础上, 提出一种适用于高维多模优化问题的基于N阶近邻分析的自适应差分进化算法(N-NNADE). 论文在N阶最短近邻理论分析及参数设定研究的基础上自适应确定划分合适种群的数量, 各种群实行局部搜索以达到协同进化的目的, 最终获取全局最优解和高质量的局部最优解来进一步提高解的可靠性. 此外, 论文给出了N-NNADE算法与DE, DERL, ADE[15]在20个全局多模优化问题上的比较结果.

## 2 群体N阶近邻分析(The Nth-order nearest neighbor analysis based population)

N-NNADE算法的核心是求解多模优化问题中可行域范围内的多个符合条件的极值解. 而满足该目的, 必须要求对现有可行域范围内的群体有较准确的先验认知, 但这在现实中往往是不可行的. N阶最短近邻分析在缺少先验知识的条件下, 对当前整个群体的结构、距离进行度量分析, 并最终根据分析结果自适应确定种群的数量, 这对于后续的种群局部搜索将有重大意义.

对于随机产生的群体, 确定合适的种群数量是相当困难的[16]. 假设给定由N个个体组成的群体

$$S=\{x_i|x_i\in\mathbb{R}^n, i=1,2,\cdots,N\},$$

通过分析群体$S$, 找到满足一定条件的$K$个种群. 考虑单个种群的情况, 当前群体可以描述成单峰概率分布, 其唯一顶点即是种群的中心点(极值点). 同理可知, 在多种群情况下, 种群视为多峰概率分布, 可看作多个种群的复制与线性合并. 因此, $K$个种群的概率密度函数为

$$f(x)=\sum_{i=1}^{K}p_if_i(x), \tag{2}$$

其中: $\sum\limits_{i=1}^{K}p_i=1$, 且满足$p_i>0$. 式(2)中的概率密度由$K$个种群的概率密度叠加而成, 其$K$值本身通常是未知的, 与当前群体的结构有关. 计算N阶最短近邻的代价则与群体的稠密程度有关, 稀疏群体需要更大的计算代价.

令$d_1^i$为群体中个体$x_i$的一阶最短近邻距离, $d_2^i$为其二阶最短近邻距离, 以此类推, 按照个体$x_i$的最短近邻距离从小到大的排列则有

$$d_1^i<d_2^i<\cdots<d_{N-1}^i,\ \forall i=1,2,\cdots,N-1. \tag{3}$$

进一步, 分别计算$j$阶最短近邻距离的平均值$\overline{d}_j$和平方均值$\overline{d}_j^2$:

$$\overline{d}_j=\frac{1}{N}\sum_{i=1}^{N}d_j^i,\ \forall j=1,2,\cdots,N-1, \tag{4}$$

$$\overline{d}_j^2=\frac{1}{N}\sum_{i=1}^{N}(d_j^i)^2,\ \forall j=1,2,\cdots,N-1. \tag{5}$$

群体中个体的分散程度将直接决定种群的数量, 为分析群体的分散程度, 求解方差$\sigma_{N,m}^2$:

$$\sigma_{N,m}^2=\frac{1}{m}\sum_{j=1}^{m}(\overline{d}_j^2-(\overline{d}_j)^2),\ \forall j,m=1,2,\cdots,N-1. \tag{6}$$

假定$y_i$为原群体个体中$x_i$的次序排列, 且满足$y_1\leqslant y_2\leqslant\cdots\leqslant y_{N-1}$的关系, 同时令$\tilde{y}_i$为其对应的独立且具有相同概率的概率密度函数$f(\tilde{y}_i)$, 且必须满足$\tilde{y}_i\geqslant 0$. 根据次序统计的相关原理[17], 计算得到概率密度的$K$阶统计$f_{N-1,K(y)}$:

$$f_{N-1,K(y)}=\frac{\mathrm{d}}{\mathrm{d}y}F_{N-1,K(y)}=\frac{\mathrm{d}}{\mathrm{d}y}\mathrm{P}(K(y)\leqslant y), \tag{7}$$

其中: $\mathrm{P}(K(y)\leqslant y)$为随机试验$K$阶次序群体中小于等于随机变量$y$的概率, $F_{N-1,K(y)}$为$y_i$的累计概率分布函数. 相应的, 上述过程可以看成在$N-1$次试验中大于等于$K$次成功的事件概率, 由于$y_i$的概率

密度函数相同, 且满足关系$y_1 \leqslant y_2 \leqslant \cdots \leqslant y_{N-1}$, 则式(7)可转化成

$$f_{N-1,K(y)} = \frac{\mathrm{d}}{\mathrm{d}y}\sum_{j=K}^{N-1}\frac{(N-1)!}{j!(N-1-j)!}\mathrm{P}(y_1 \leqslant y)^j \times (1-\mathrm{P}(y_1 \leqslant y))^{N-1-j}. \tag{8}$$

显然, 式(8)可进一步看成

$$f_{N-1,K(y)} = \frac{\mathrm{d}}{\mathrm{d}y}\sum_{j=K}^{N-1}\frac{(N-1)!}{j!(N-1-j)!}F(y)^j \times (1-F(y))^{N-1-j}. \tag{9}$$

分解式(9)计算得

$$\begin{aligned} & f_{N-1,K(y)} = \\ & \sum_{j=K}^{N-1}\frac{(N-1)!}{j!(N-1-j)!}[jF(y)^j f(y)(1-F(y))^{N-1-j} - \\ & (N-1-j)F(y)^j f(y)(1-F(y))^{N-2-j}] = \\ & \sum_{j=K}^{N-1}[\frac{(N-1)(N-2)!}{(j-1)!(N-1-j)!}F(y)^{j-1} \times \\ & (1-F(y))^{N-1-j} - \frac{(N-1)(N-2)!}{j!(N-2-j)!} \times \\ & F(y)^j(1-F(y))^{N-2-j}]f(y) = \\ & (N-1)f(y) \times \\ & [\sum_{j=K-1}^{N-2}\frac{(N-2)!}{j!(N-2-j)!}F(y)^j(1-F(y))^{N-2-j} - \\ & \sum_{j=K}^{N-1}\frac{(N-2)!}{j!(N-2-j)!}F(y)^j(1-F(y))^{N-2-j}]. \end{aligned} \tag{10}$$

将式(10)的求和项展开, 相同项约掉, 得

$$\begin{aligned} & f_{N-1,K(y)} = \\ & (N-1)f(y) \times [\frac{(N-2)}{(K-1)!(N-1-K)!} \\ & F(y)^{K-1}(1-F(y))^{N-1-K} - \\ & \frac{(N-2)!}{(N-1)!(-1)!}F(y)^{N-1}(1-F(y))^{-1}]. \end{aligned} \tag{11}$$

由于项

$$\frac{(N-2)!}{(N-1)!(-1)!}F(y)^{N-1}(1-F(y))^{-1} = 0,$$

则$K$阶统计的概率密度$f_{N-1,K(y)}$为

$$f_{N-1,K(y)} = \frac{(N-1)!}{(K-1)!(N-K-1)!}F(y)^{K-1} \times (1-F(y))^{N-K-1}f(y), \tag{12}$$

其中$F(y)$是对应的累积概率分布函数:

$$F(y) = \int_0^y f(t)\mathrm{d}t. \tag{13}$$

此时, 式(4)(5)可等价转化成下式:

$$\bar{d}_j \equiv \bar{y}_K = \int_0^\infty y f_{N-1,K(y)}\mathrm{d}y, \tag{14}$$

$$\bar{d}_j^2 \equiv \bar{y}_K^2 = \int_0^\infty y^2 f_{N-1,K(y)}\mathrm{d}y. \tag{15}$$

运用次序统计原理[17]及对上述公式的分析, 在给定群体$S$聚类形成单个种群的情况下, 方差$\sigma_{N,m}^2$可表示成

$$\sigma_{N,m}^2 = \alpha(N)(m+1)^2 + \beta(N)(m+1). \tag{16}$$

基于上式, 假设通过理论分析与公式推导求出系数$\alpha(N)$和$\beta(N)$, 结合给定的$m$值, 就可以精确计算得到$\sigma_{N,m}^2$. 将式(14)(15)代入式(6)得

$$\sigma_{N,m}^2 = \frac{1}{m}\sum_{K=1}^{m}[\int_0^\infty y^2 f_{N-1,K(y)}\mathrm{d}y - (\int_0^\infty y f_{N-1,K(y)}\mathrm{d}y)^2], \tag{17}$$

同时, 再将式(12)与式(13)分别代入式(17)得:

$$\begin{aligned} \sigma_{N,m}^2 = & \frac{1}{m}\sum_{K=1}^{m}\{\frac{(N-1)!}{(K-1)!(N-K-1)!} \\ & \int_0^\infty y^2(\int_0^y f(t)\mathrm{d}t)^{K-1} \times \\ & (1-\int_0^y f(t)\mathrm{d}t)^{N-K-1}f(y)\mathrm{d}y - \\ & [\frac{(N-1)!}{(K-1)!(N-K-1)!} \times \\ & \int_0^\infty y(\int_0^y f(t)\mathrm{d}t)^{K-1} \times \\ & (1-\int_0^y f(t)\mathrm{d}t)^{N-K-1} f(y)\mathrm{d}y]^2\}, \end{aligned} \tag{18}$$

从上式可以发现, 若想计算最终的$\sigma_{N,m}^2$, 必须给定相应的概率密度函数$f(y)$. 假设当前群体是在$(0,1)$均匀分布的前提下, 则$f(y)=1$, 式(18)便可简化成

$$\begin{aligned} \sigma_{N,m}^2 = & \frac{1}{m}\sum_{K=1}^{m}\{\frac{(N-1)!}{(K-1)!(N-K-1)!} \times \\ & \int_0^1 y^{K+1}(1-y)^{N-K-1}\mathrm{d}y - \\ & [\frac{(N-1)!}{(K-1)!(N-K-1)!} \times \\ & \int_0^1 y^{K+1}(1-y)^{N-K-1}\mathrm{d}y]^2\}. \end{aligned} \tag{19}$$

进一步运用Beta函数及其性质

$$B(p,q) = \int_0^1 x^{p-1} \times (1-x)^{q-1}\mathrm{d}x = \frac{(p-1)!(q-1)!}{(p+q-1)!},$$

代入式(19)可得:

$$\sigma_{N,m}^2 = \frac{1}{N^2(N+1)}(\frac{N}{m}\sum_{K=1}^{m}K - \frac{1}{m}\sum_{K=1}^{m}K^2). \tag{20}$$

利用求和公式与平方求和公式可将式(20)转化为下式:

$$\sigma_{N,m}^2 = -\frac{1}{3N^2(N+1)}(m+1)^2 + \frac{3N+1}{6N^2(N+1)}(m+1). \tag{21}$$

通过式(16)(21)的对照, 系数$\alpha(N)$, $\beta(N)$确认为

$$\alpha(N)=-\frac{1}{3N^2(N+1)},\ \beta(N)=\frac{3N+1}{6N^2(N+1)}. \quad (22)$$

上述推导均假设在$(0,1)$均匀分布的条件下, 而在高斯分布或者其他分布条件下, 传统的理论推导就相当困难. 因此采用适当的数值分析计算, 可以得到无限接近期望值的近似解. 将$\sigma^2_{N,m}$做$\sigma^2_{N,m}/(m+1)$的处理, 式(16)便可转化成

$$\sigma^2_{N,m}/(m+1)=\alpha(N)(m+1)+\beta(N). \quad (23)$$

显然, 式(23)满足线性关系, 通过判断近邻个体斜率的变化关系, 便可确定阶跃的发生. 假设斜率在某一段内发生较大改变, 可认定中间发生了一次阶跃, 即当前必定存在两个种群, 使得在计算统计群体最短近邻时发生跳跃式的变化. 迭代遍历群体中全部个体, 统计所有的阶跃个体$m^*$, 即可得到最终的种群数量. 如图1(a)所示, 个体随机分布形成单个种群的情况, 计算得到的是一条光滑的曲线(如图1(b)所示), 并没有发生任何阶跃情况; 而对$\sigma^2_{N,m}/(m+1)$求解的所有近邻个体斜率变化相当平稳, 也无任何阶跃发生(如图1(c)所示).

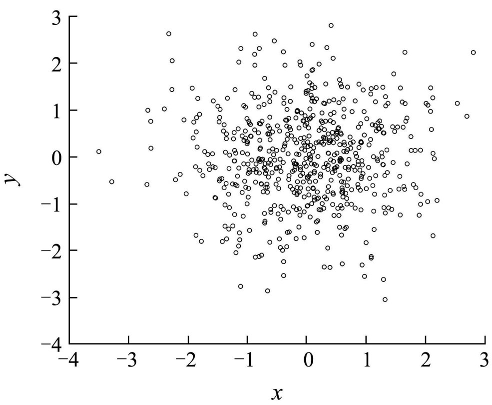


图 1(a) 群体随机形成单个种群情况

Fig. 1(a) Situation of single species

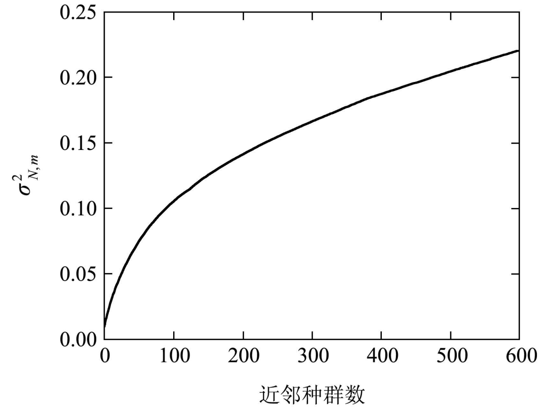


图 1(b) 单个种群$\sigma^2_{N,m}$计算统计

Fig. 1(b) $\sigma^2_{N,m}$ statistics under single species

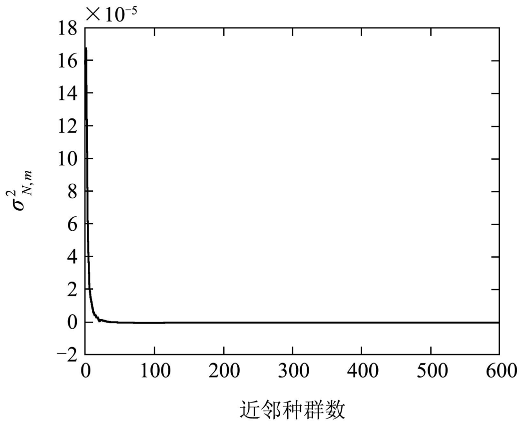


图 1(c) 单个种群近邻点斜率变化

Fig. 1(c) Slope change under single species

考虑多种群的情况, 如图2(a)所示, 群体随机分布形成两个种群的情况下, 可以发现图2(b)中的曲线明显有两个阶跃点; 而对$\sigma^2_{N,m}/(m+1)$求解得到的全部近邻个体斜率变化曲线(如图2(c)所示)也同样发现二次阶跃的情况.

将单个种群与多种群的情况作对比, 发现在统计分析群体个体之间的分散度时, 一旦有超过一个种群的情况存在, 即会有相应的阶跃情况出现. 引起阶跃发生的直接因素是由于在分析群体中个体分散度的过程中, 同一种群间的近邻个体斜率变化较小, 但当从一个种群跨越到另一种群计算个体斜率时将发生巨大波动. 通过该方式捕捉全部阶跃信息, 即可知晓当前群体的全局分布结构, 进而统计得到合适的种群数量.

此外, 通过记录所有的阶跃个体$m^*$, 则可统计出该种群中所包含的个体数量. 图2(b)和图2(c)进一步证实了阶跃个体同种群中包含个体数量之间的关系, 即种群$C_1$中包含了200个体, 而种群$C_2$中包含了400个体.

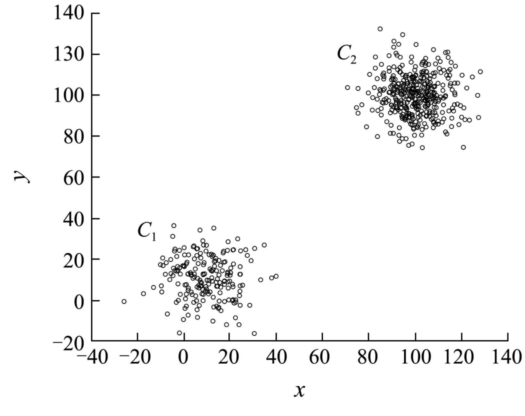


图 2(a) 群体随机形成多个种群情况

Fig. 2(a) Situation of multiple species

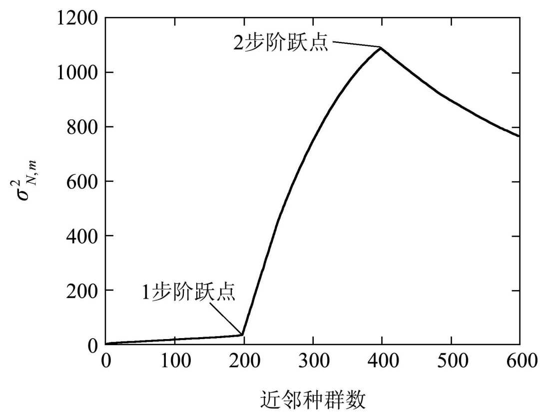


图 2(b) 多个种群$\sigma^2_{N,m}$计算统计

Fig. 2(b) $\sigma^2_{N,m}$statistics under multiple species

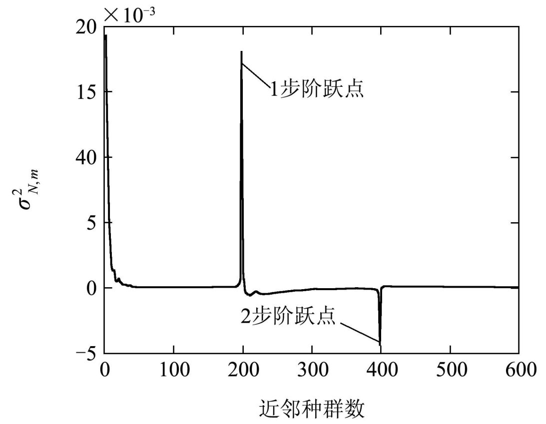


图 2(c) 多个种群近邻点斜率变化

Fig. 2(c) Slope change under multiple species

上述$N$阶近邻理论分析过程证明通过对群体的$N$阶最短近邻分析, 无需先验知识即能发现整个群体的分布情况, 从而较准确统计得出可划分的种群数量. 同时, 分析过程也阐明在理论推导无法得出结果的情况下, 可进一步采用式(24)的数值计算来获取与期望值极其接近的近似结果.

$$|\frac{\sigma^2_{N,m+1}}{m+2}+\frac{\sigma^2_{N,m-1}}{m}-2\frac{\sigma^2_{N,m}}{m+1}|<\varepsilon|\frac{\sigma^2_{N,m}}{m+1}|, \tag{24}$$

其中: 阈值$\varepsilon$是满足$(0,1)$区间足够小的常数, 且式(24)必须满足$m \geqslant 2$的情况, 否则将不成立.

## 3 $N$阶近邻自适应差分进化算法($N$-NNADE algorithm)

与传统基本DE算法相似, 基于$N$阶最短近邻分析的自适应差分进化算法($N$-NNADE)同样包含了变异、交叉以及选择步骤. $N$-NNADE的基本思想: 在随机初始化群体后, 按照预先设计的窗口大小$\omega$, 借助于$N$阶近邻分析理论, 通过对群体中个体间求解方差线性化后的曲率变化进行分析以及对阶跃点的统计, 得到当前群体的全局分布状态信息, 根据统计阶跃点信息自适应确定可分种群的数量; 运用高效的聚类算法($K$-means算法)对群体进行分类, 按照近亲不得结合的原则选择本种群与其他任意种群中的个体进行杂交变异, 判定经过一定交叉概率产生的新个体归属于哪个种群, 若与当前父个体是同一种群, 则替换掉种群中目标函数值最差的个体, 否则不做任何变化. $N$-NNADE详细步骤描述如下:

**Step 1**　在可行域$\Omega$范围内初始化群体$S=\{x_1, x_2,\cdots,x_N\}$, 其中$x_i$为$D$维单独个体, $N$为整个群体中包含的个体数量, 且满足$N >> D$. 求解初始群体个体的目标函数值$f(x_i)$, 设置计算器初值counter $=0$, 最大进化代数maxGeneration $=1000$, $N$阶最短近邻分析过程窗口$\omega$, 设置如下式:

$$\omega=(\lfloor D/5 \rfloor+1)\times 5, \tag{25}$$

其中符号$\lfloor\ \rfloor$表示向下取整.

**Step 2**　如果$\mathrm{Mod}\{\mathrm{counter},\omega\}==0$, 则转到Step 3, 否则转到Step 5.

**Step 3**　对群体$S$进行步骤3a)～3e)的$N$阶最短近邻分析, 根据对群体个体分散度的统计分析, 进一步计算确定可分种群的数量$K$.

3a) 设置误差参数$\varepsilon=0.01$, 可分种群数量初始置$K=0$.

3b) 计算个体$x_i$的1阶、2阶$\cdots N$阶距离, 并满足式(3), 进行从小到大的排序; 根据式(4)(5)分别计算$j$阶最短近邻距离的平均值$\overline{d}_j$和平方均值$\overline{d}^2_j$.

3c) 根据式(6)求解方差$\sigma^2_{N,m}$, 得到群体的个体分散程度, 并对$\sigma^2_{N,m}$做$\sigma^2_{N,m}/(m+1)$处理.

3d) 迭代计算获取全部近邻个体的斜率变化, 一旦满足式(24), 置$K=K+1$.

3e) 如果步骤3d)未发生满足式(24)成立的情况, 则置$K=1$.

**Step 4**　采用$K$-means聚类算法对当前群体$S$进行划分, 形成$K$个种群Clusters $=\{C_1,C_2,\cdots, C_K\}$.

**Step 5**　遍历种群$C_K(C_K\in$ Clusters), 根据式(26)对所有种群个体进行变异操作:

$$\tilde{x}_i=x_{p\,(in)}+F\cdot(x_{p\,(o1)}-x_{p\,(o2)}), \tag{26}$$

其中: $x_{p\,(in)},x_{p\,(o1)},x_{p\,(o2)}\in C_K$并满足$x_{p\,(in)}$属于当前种群且$p(in)\neq i$, 而$x_{p\,(o1)}$, $x_{p\,(o2)}$为其他任意种群的任意两个个体, 且满足$x_{p\,(o1)}\neq x_{p\,(o2)}$, 必须保证群体个体数量$N\geqslant 4$; $F$为变异因子, 且满足$F\in[0,2]$.

**Step 6**　利用式(27)进行交叉操作, 通过一定的边界检测机制验证当前新产生的个体(试验向量)是否在可行域$\Omega$范围内, 若满足$y_i\notin\Omega$, 则强制令其回

归到可行域范围内.

$$y_{i,j}=\begin{cases}\tilde{x}_{i,j},\ \mathrm{rand}(j)\leqslant CR\text{ 或 }j=\mathrm{Index}_i,\\ x_{i,j},\ \mathrm{rand}(j)>CR\text{ 或 }j\neq\mathrm{Index}_i,\end{cases}\tag{27}$$

其中: $i=1,2,\cdots N,\ j=1,2,\cdots,D,\mathrm{rand}(j)$表示随机生成第$j$维交叉概率且满足$\mathrm{rand}(j)\in[0,1]$, $\mathrm{Index}_i$表示当前选取的开始维数索引号并且满足$\mathrm{Index}_i\in\{1,2,\cdots,D\}$, $CR$为交叉因子, 并且满足$CR\in[0,1]$.

**Step 7** 选择策略: 判定当前新个体与父个体是否同属一个种群, 如果属于同一个种群, 那么查找当前种群中目标函数值最差个体并用新个体替换, 反之不做任何替换.

**Step 8** 计数器累加$\mathrm{counter}=\mathrm{counter}+1$, 若计数器达到最大进化代数, 算法停止, 否则转到Step 2继续迭代.

**注 1** $N$-NNADE算法设置的最大进化代数为1000代, 但由于高维问题计算困难, 在部分10维及10维以上的问题将采用更大的预设进化代数.

**注 2** $N$-NNADE与DE, DERL, ADE在计算目标函数评价次数时, 群体中所有的个体都必须满足是收敛到某一吸收态, 而不仅仅是某个个体找到问题的全局最优解. $N$-NNADE算法计算目标函数评价次数以算法搜寻到全局最优解为准.

$N$-NNADE算法强调在一定窗口大小下实行$N$阶最短近邻计算分析, 通过探测当前群体中个体之间的全局分布情况, 进一步确定可划分种群的数量$K$. 在此基础上, $K$个种群协同进化, 并逐步收敛至各自的中心点(最优解), 从而形成最终包含多个优质解的解集. $N$-NNADE算法通过求解得到的多个优质解来提高解的可靠性, 相比基本DE, DERL, ADE, 更适合在大规模高维多模优化问题和解不能确定的实际工程中应用.

# 4 数值仿真研究(Numerical simulation)

## 4.1 基准问题(Benchmark problem)

为保证$N$-NNADE算法的有效性, 应用20个典型的全局优化问题测试验证其性能, 维数范围为2～100维, 由于篇幅的限制, 具体问题的表现形式详见文献[18]. 每个问题都有相应不同的目标评价函数和复杂度, 表1给出了20个问题(P)的维数($n$)以及存在的极值点数量($ep$), 其中"?"表示未知.

对于表1中20个优化问题, 其中半数以上为2维以上的高维优化复杂问题, 求解过程十分困难; 此外, 本文还对部分测试函数进行同一问题不同维度下的测试来验证算法的性能. 现在对DE, DERL, ADE, $N$-NNADE算法参数具体设置: 群体规模$N=10n$, 变异因子$F=0.5$, 交叉因子$CR=0.5$, 最大进化代数maxGeneration $=1000$, 对于个别高维问题在规定最大进化代数中无法计算出结果, maxGeneration允许调高至20000代, 尤其是超高维SWF-100问题允许为1000000以上. 本文分别采用目标函数评价数(function evaluation, FE)、成功率(success ratio, SR)、平均方差(mean of variance, MV)以及计算所得的极值点数量(number of extreme point, NEP)作为3种算法比较的评价指标. FE作为同类算法应用最广的性能评价指标, 能够较好地反映每种算法的时间复杂度. SR为成功运行获得全局最优解的次数与总运行次数之比, 它能充分衡量一个算法的可靠性. 此外, MV与NEP是进一步衡量所得解精确性的指标, 即极值点数量越多, 方差越大, 最终求解的全局最优值也最为准确, 这在实际工程领域尤其适用.

表 1 20个全局优化问题

Table 1 Twenty benchmark problems

| P | $n$ | $ep$ | P | $n$ | $ep$ | P | $n$ | $ep$ |
|---|---|---|---|---|---|---|---|---|
| AP | 2 | 2 | BL | 2 | 4 | BP | 2 | 3 |
| CB6 | 2 | 3 | EP | 2 | ? | SBT | 2 | 760 |
| ACK | 2,10 | ?,? | EM | 5 | ? | EXP | 10 | ? |
| GW | 10 | ? | H6 | 6 | 5 | LM1 | 3,30 | 125,? |
| LM2 | 10 | ? | ML | 10 | ? | OSP | 10 | ? |
| RG | 30 | ? | SAL | 10 | ? | SWF | 100 | ? |
| SIN | 20 | ? | ST | 9 | ? | | | |

## 4.2 仿真结果(Simulation results)

为消除算法在运行过程中的随机因素, DE, DERL, ADE, $N$-NNADE算法各迭代运行100次, 即总运行次数为$4\times22\times100=8800$. 表2给出了低维优化问题在DE, DERL, ADE, $N$-NNADE算法上运行100次的平均结果以及各项指标平均值(AVE), 而表3则为高维优化问题运行100次的平均结果以及平均值. 表2和表3中: $fe$为调用评价函数的次数; $sr$为求解成功率大小; $mv$是平均方差; $nep$是最优解的个数.

表2说明对于低维(2维)7个问题, $N$-NNADE算法的平均函数评价数为1594, 较之传统DE有所减少, 但比DERL, ADE的计算代价要高; 而在成功率上, $N$-NNADE算法要明显高于DE, 比DERL, ADE算法稍高, 达到100%, 其可靠性得到了较好改善. 此外, DE, DERL, ADE算法的平均方差几乎趋向于0, 而$N$-NNADE算法的平均方差达到了55.131, 这从侧面反映了$N$-NNADE算法解的多样性. 从所得极值点数量也可以发现, $N$-NNADE算法明显要好于DE, DERL, ADE算法, 能得到所有的全局最优解和部分的局部优质解.

表 2 2维全局优化问题的计算结果

Table 2 Results of two dimension problems

| P-$n$ | DE | | | | DERL | | | | ADE | | | | $N$-NNADE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $fe$ | $sr$ | $mv$ | $nep$ | $fe$ | $sr$ | $mv$ | $nep$ | $fe$ | $sr$ | $mv$ | $nep$ | $fe$ | $sr$ | $mv$ | $nep$ |
| ACK-2 | 1405 | 1.00 | 0.47E−4 | 1.0 | 1577 | 1.00 | 0.1253E−2 | 1.0 | 1063 | 1.00 | 0.152E−2 | 1.0 | 1299 | 1.00 | 6.504 | 1.23 |
| AP-2 | 789 | 1.00 | 0.1829E−2 | 1.0 | 748 | 1.00 | 0.1984E−2 | 1.0 | 707 | 1.00 | 0.1919E−2 | 1.0 | 791 | 1.00 | 1.009 | 2.0 |
| BL-2 | 1183 | 1.00 | 0.0000 | 1.0 | 935 | 1.00 | 0.547E−3 | 1.0 | 851 | 1.00 | 0.217E−3 | 1.0 | 1197 | 1.00 | 87.192 | 4.0 |
| BP-2 | 2561 | 1.00 | 0.2212E−2 | 1.0 | 1096 | 1.00 | 0.351E−2 | 1.0 | 892 | 1.00 | 0.416E−2 | 1.0 | 2378 | 1.00 | 101.958 | 3.0 |
| CB6-2 | 1592 | 1.00 | 0.5241E−2 | 1.0 | 937 | 1.00 | 0.4158E−2 | 1.0 | 795 | 1.00 | 0.373E−2 | 1.0 | 1448 | 1.00 | 33.026 | 3.0 |
| EP-2 | 1227 | 0.97 | 0.1001E−2 | 1.0 | 901 | 1.00 | 0.981E−3 | 1.0 | 803 | 1.00 | 0.887E−3 | 1.0 | 1313 | 1.00 | 65.102 | 2.37 |
| SBT-2 | 3023 | 0.12 | 0.4574E−2 | 1.0 | 2582 | 0.98 | 0.768E−2 | 1.0 | 2499 | 0.99 | 0.616E−2 | 1.0 | 2731 | 1.00 | 91.125 | 11.59 |
| AVE | 1683 | 0.87 | 0.2129E−2 | 1.0 | 1254 | 0.99 | 0.2873E−2 | 1.0 | 1087 | 0.99 | 0.2656E−2 | 1.0 | 1594 | 1.00 | 55.131 | 3.88 |

∗附注: $nep$为计算获得的极值点数量, 包含全局最优值和局部最优解, 即满足梯度为0点(下表同).

表 3 高维全局优化问题的计算结果

Table 3 Results of high dimension problems

| P-$n$ | DE | | | | DERL | | | | ADE | | | | $N$-NNADE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $fe$ | $sr$ | $mv$ | $nep$ | $fe$ | $sr$ | $mv$ | $nep$ | $fe$ | $sr$ | $mv$ | $nep$ | $fe$ | $sr$ | $mv$ | $nep$ |
| ACK-10 | 30652 | 1.00 | 0.146E−3 | 1.0 | 31080 | 1.00 | 0.212E−3 | 1.0 | 24198 | 1.00 | 0.173E−3 | 1.0 | 23729 | 1.00 | 12.234 | 6.12 |
| EM-5 | 21633 | 0.95 | 0.511E−4 | 1.0 | 17820 | 0.90 | 0.429E−4 | 1.0 | 14560 | 1.00 | 0.491E−4 | 1.0 | 15958 | 1.00 | 59.910 | 4.18 |
| EXP-10 | 14433 | 1.00 | 0.2E−4 | 1.0 | 14216 | 1.00 | 0.3174E−4 | 1.0 | 10659 | 1.00 | 0.2511E−4 | 1.0 | 10156 | 1.00 | 21.575 | 8.23 |
| GW-10 | 80283 | 1.00 | 0.8003E−3 | 1.0 | 93260 | 1.00 | 0.6151E−3 | 1.0 | 69773 | 0.94 | 0.5495E−3 | 1.0 | 71503 | 1.00 | 1560.833 | 19.78 |
| H6-6 | 11269 | 0.78 | 0.138E−3 | 1.0 | 9068 | 0.91 | 0.852E−4 | 1.0 | 5724 | 0.93 | 0.918E−4 | 1.0 | 6510 | 1.00 | 82.193 | 4.88 |
| LM1-3 | 1876 | 1.00 | 0.1905E−4 | 1.0 | 1451 | 1.00 | 0.2738E−4 | 1.0 | 1319 | 1.00 | 0.3017E−4 | 1.0 | 1399 | 1.00 | 29.129 | 3.98 |
| LM1-30 | 188700 | 0.72 | 0.5076E−2 | 1.0 | 161485 | 0.74 | 0.9042E−2 | 1.0 | 149460 | 0.79 | 0.812E−2 | 1.0 | 131947 | 0.89 | 5031.600 | 33.87 |
| LM2-10 | 17416 | 0.67 | 0.5457E−2 | 1.0 | 16703 | 1.00 | 0.4689E−2 | 1.0 | 14796 | 1.00 | 0.3523E−2 | 1.0 | 14914 | 1.00 | 2215.875 | 13.19 |
| ML-10 | 100 | 0.00 | 0.2971 | 1.0 | 100 | 0.00 | 0.119 | 1.0 | 100 | 0.00 | 0.913 | 1.0 | 59028 | 0.38 | 8902.507 | 11.51 |
| OSP-10 | 61430 | 0.83 | 0.1793E−3 | 1.0 | 55618 | 0.89 | 0.1086E−3 | 1.0 | 51058 | 0.94 | 0.1181E−3 | 1.0 | 49005 | 0.98 | 5194.659 | 9.76 |
| RG-30 | 1689310 | 0.00 | 0.1489 | 1.0 | 1620451 | 0.19 | 0.6723E−2 | 1.0 | 1568920 | 0.32 | 0.7167E−2 | 1.0 | 1471935 | 0.65 | 11498.522 | 47.13 |
| SAL-10 | 97643 | 0.69 | 0.9786E−3 | 1.0 | 93559 | 0.57 | 0.8755E−3 | 1.0 | 75319 | 0.88 | 0.658E−3 | 1.0 | 71359 | 0.97 | 3877.798 | 5.61 |
| SWF-100∗ | — | — | — | — | — | — | — | — | — | — | — | — | 1.179E9 | 0.21 | 28331.441 | 88.53 |
| SIN-20 | 169095 | 0.01 | 0.634E−2 | 1.0 | 159902 | 0.09 | 0.509E−2 | 1.0 | 150728 | 0.22 | 0.471E−2 | 1.0 | 133570 | 0.60 | 18971.593 | 37.62 |
| ST-9 | 799830 | 0.59 | 0.1664E−2 | 1.0 | 716380 | 0.71 | 0.2391E−2 | 1.0 | 681190 | 0.86 | 0.2516E−2 | 1.0 | 655311 | 0.93 | 5333.152 | 17.93 |
| AVE | 227405 | 0.66 | 0.3335E−1 | 1.0 | 213650 | 0.71 | 0.1064E−1 | 1.0 | 201272 | 0.78 | 0.672E−1 | 1.0 | 194023 | 0.89 | 4485.113 | 15.985 |

∗附注: SWF-100问题的计算结果未被统计到AVE中; ML-10的计算最终影响AVE的统计.

从表3中不难发现, 对于高维(维数3∼100)15个问题, $N$-NNADE算法的平均评价函数为194023, 分别是DE算法平均函数评价数的85.3%, 是DERL的90.8%, 是ADE的96.4%; 成功率也较DE, DERL, ADE有了显著的提高. 此外, $N$-NNADE在高维问题上取得与低维问题相类似的效果, 获得更多的全局最优解和部分高质量的局部最优解, 且进一步验证了无论是低维还是高维复杂问题, DE, DERL, ADE算法的平均方差都趋向于0, 其最终解收敛于一个最优值, 而$N$-NNADE算法的平均方差必不趋向0. 这也证明在吸收态下, 相对于DE, DERL, ADE算法, $N$-NNADE算法通过获取大量的最优解来提高与保证最终解的可靠性. 这里需要特别指出的是, ML-10问题的目标函数曲面为脉冲山峰, 在有限的群体规模内差异几乎为0, 导致4类算法很难寻优; 对于SIN-20, RG-30和SWF-100 3个超高维问题, SIN-20, RG-30在既定最大进化代数maxGeneration=20000里, DE, DERL, ADE大部分运行次数内都限于局部最小值而丧失了进一步进化的能力; 而SWF-100则由于本身问题的复杂程度, 以及本文采用的有限群体规模$N = 10n$, 同样是群体差异极小, 从而几乎无法计算求得全局最优解.

由于$N$-NNADE为获得更多的最优解所采取的$N$阶近邻分析以及种群协同进化的策略, 带来了一定的计算代价, 因此在全局寻优能力上有所下降, 即使采取一定的窗口设置, 在部分低维问题上的收敛性仍然不如DE, DERL, ADE算法. 相对于

低维问题而言, $N$-NNADE算法在高维问题的计算代价有了明显改善, 这是由于DE, DERL, ADE算法往往因为高维带来的复杂运算极易陷入局部最优值而无法自拔, 且有限的寻优能力在大范围的可行域范围内搜索全局最优解将花费更多时间; 而$N$-NNADE算法在可行域内将群体划分为多个种群, 各种群之间共同进化, 在高维问题能够更快、更可靠地搜寻到全局最优解以及优质的局部解. 由此可见, $N$-NNADE算法相对于DE, DERL, ADE算法更适合于高维复杂问题的求解. 此外, 由于实际工程应用中无法知晓全局最优解, 只能通过尽可能地获取最优解进而通过比较各个可行解来提高解的可靠性, 这也为最终的应用实施提供了保障.

## 5 结论(Conclusion)

传统DE算法在多模优化问题里极易陷入早熟现象, 这在大规模高维问题上显得尤为突出, 如何提高解的可靠性是亟待解决的重要研究课题. 本文提出的基于$N$阶最短近邻的自适应差分进化算法通过对群体中个体的$N$阶最短近邻计算来获取种群的全局分布信息, 进而统计阶跃信息自适应确定可分种群的数量, 各种群之间协同进化, 最终得到更多的全局最优解及部分高质量的局部最优解, 从而提高解的可靠性. 数值研究结果也表明了$N$-NNADE算法具有比DE, DERL, ADE算法更适合求解复杂的高维多模优化问题.

## 参考文献(References):


[1] EL IMRANI A, EL ABIDINE HZ, LIMOURI M, et al. Multimodal optimization of thermal histories[J]. *Comptes Rendus De L Academie Des Sciences Series II, Fascicule A: Sciences DE La Terre ET DES Planetes*, 1999, 329(8): 573 – 577.

[2] LIU X M, LEE B. Optimal design of fiber Raman amplifier based on hybrid genetic algorithm[J]. *IEEE Photonics Technology Letters*, 2004, 16(2): 428 – 430.

[3] DILETTOSO E, SALERNO N. A self-adaptive niching genetic algorithm for multimodal optimization of electromagnetic devices[J]. *IEEE Transactions on Magnetics*, 2006, 42(4): 1203 – 1206.

[4] DE TORO F, ROS E, MOTA S, et al. Evolutionary algorithms for multiobjective and multimodal optimization of diagnostic schemes[J]. *IEEE Transactions on Biomedical Engineering*, 2006, 53(2): 178 – 189.

[5] HERRERO J M, BLASCO X, MARTINEZ M, et al. Non-linear robust identification using evolutionary algorithms: application to a biomedical process[J]. *Engineering Applications of Artificial Intelligence*, 2008, 21(8): 1397 – 1408.

[6] MORDECAL A. *Nonlinear Programming Analysis and Methods*[M]. New Jersey: Prentice Hall Press, 1976.

[7] THOMSEN R. Multimodal optimization using crowding-based differential evolution[C] //*Proceedings of the 2004 Congress on Evolutionary Computation*. Portland, USA: IEEE, 2004: 1382 – 1389.

[8] STORN R, PRICE K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341 – 359.

[9] ALI M M, TORN A. Population set-based global optimization algorithms: some modification and numerical studies[J]. *Computer and Operations Research*, 2004, 31(10): 1703 – 1725.

[10] KAELO P, ALI M M. A numerical study of some modified differential evolution algorithms[J]. *European Journal of Operational Research*, 2006, 169(3): 1176 – 1184.

[11] SUN J Y, ZHANG Q F, TSANG E P K. DE/EDA: a new evolutionary algorithm for global optimization[J]. *Information science*, 2005, 169(3/4): 249 – 262.

[12] SALMAN A, ENGELBRECHT A P, OMRAN M G H. Computing, artificial intelligence and information management – empirical analysis of self-adaptive differential evolution[J]. *European Journal of Operational Research*, 2007, 183(2): 785 – 804.

[13] LOPZE CRUZ I L, VAN WILLIGENBURG L G, VAN STRATEN G. Efficient differential evolution algorithms for multimodal optimal control problems[J]. *Applied Soft Computing*, 2003, 3(2): 97 – 122.

[14] SHIBASAKA M, HARA A, ICHIMURA T, et al. Species-based differential evolution with switching search strategies for multimodal function optimization[C] //*Proceedings of 2007 IEEE Congress on Evolutionary Computation*. Singapore: IEEE, 2007: 1183 – 1190.

[15] 张贵军, 王信波, 俞立, 等. 求解高维多模优化问题的自适应差分进化算法[J]. 控制理论与应用, 2008, 25(5): 862 – 866.
(ZHANG Guijun, WANG Xinbo, YU Li, et al. Adaptive differential evolution for high-dimension multimodal optimization problems[J]. *Control Theory & Applications*, 2008, 25(5): 862 – 866.)

[16] BLEKAS K, LAGARIS I E. Newtonian clustering: an approach based on molecular dynamics and global optimization[J]. *Pattern Recognition*, 2007, 40(6): 1734 – 1744.

[17] DAVID H A, NAGARAJA H N. *Order Statistics*[M]. 3rd Edition. New Jersey, USA: Wiley Press, 2003.

[18] ALI M M, KHOMPATRAPORN C, ZABINSKY Z B. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems[J]. *Journal of Global Optimization*, 2005, 31(4): 635 – 672.


## 作者简介:


**洪 榛** (1983—), 男, 博士研究生, 主要研究方向为群体智能、优化调度、无线传感器网络等, E-mail: hongzhen614@126.com;

**张贵军** (1974—), 男, 副教授, 主要研究方向为优化调度、智能交通系统等, E-mail: zgj@zjut.edu.cn;

**俞 立** (1961—), 男, 教授, 博士生导师, 主要研究方向为网络控制系统、鲁棒控制、优化调度等, E-mail: lyu@zjut.edu.cn.