

文章编号: 1001-0920(2009)03-0361-04

多目标混沌差分进化算法

牛大鹏^{a,b}, 王福利^{a,b}, 何大阔^b, 贾明兴^b

(东北大学 a. 教育部流程工业综合自动化重点实验室, b. 信息科学与工程学院, 沈阳 110004)

摘要: 将差分进化算法用于多目标优化问题, 提出了多目标混沌差分进化算法(CDEMO). 该算法利用混沌序列初始化种群, 并用混沌备用种群进行替换操作. 该操作不仅起到了维持非劣最优解集均匀性的作用, 而且增强了算法的搜索功能. 对 CDEMO 的性能进行研究, 数值实验结果表明了 CDEMO 的有效性.

关键词: 差分进化算法; 多目标优化; 混沌备用种群; 非劣最优解集

中图分类号: TP274 **文献标识码:** A

Chaotic differential evolution for multiobjective optimization

NIU Da-peng^{a,b}, WANG Fu-li^{a,b}, HE Da-kuo^b, JIA Ming-xing^b

(a. Key Laboratory of Integrated Automation of Process Industry, Ministry of Education, b. College of Information Science and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: NIU Da-peng, E-mail: niudapeng@ise.neu.edu.cn)

Abstract: By using differential evolution algorithm to solve multiobjective optimization problems, chaotic differential evolution for multiobjective optimization (CDEMO) is proposed. Chaotic sequences are used in the initialization of the evolutionary population and chaotic population candidate is created with chaotic variables to be used in substitution operation. The operation not only helps to maintain uniformity of the Pareto optimal solution set, but enhances the algorithm's searching ability. The optimization performance of CDEMO is evaluated and numerical experimental results show the effectiveness of the proposed method.

Key words: Differential evolution algorithm; Multiobjective optimization; Chaotic population candidate; Pareto optimal solution set

1 引言

科学研究和工程实践中存在着大量的多目标优化问题. 多目标优化问题通常并不存在各目标均为全局最优的解, 而是存在一个非劣解集, 称为 Pareto 最优解集. 求解多目标优化问题的关键是求出尽可能逼近 Pareto 解集, 并且均匀分布的一组解. 进化算法在多目标优化中已有很多应用, 典型的有 NSGA^[1], NSGA-^[2], SPEA^[3] 等.

差分进化算法 (DE)^[4] 是一种随机的并行直接全局搜索算法, 它简单易用, 具有稳健性和全局寻优能力, 已在多个领域得到广泛的应用^[5]. 有的学者将 DE 的功能进行拓展, 用于多目标优化问题. Madavan^[6] 提出了 Pareto 差分进化方法 (PDEA), 该方法能很好地维持种群的多样性, 但收敛速度较慢. Robic^[7] 提出了多目标差分进化方法 (DEMO),

提高了算法的收敛速度, 但在一定程度上降低了种群多样性, 易使算法过早收敛.

本文将混沌序列应用于 DEMO, 引入混沌替换操作来提高种群多样性, 防止种群的过早收敛, 并构建了多目标混沌差分进化算法 (CDEMO), 用于多目标优化问题的求解.

2 多目标优化问题描述

以最小化问题为例, 多目标优化问题可表示如下:

$$\begin{aligned} \min f(X) &= [f_1(X), \dots, f_i(X), \dots, f_n(X)], \\ X &= (x_1, \dots, x_j, \dots, x_m), \\ x_{j \min} &\leq x_j \leq x_{j \max}, \quad j = 1, 2, \dots, m. \end{aligned} \quad (1)$$

其中: $f(X) \in \mathbb{R}^n$ 为目标向量, $f_i(X)$ ($i = 1, 2, \dots, n$) 为第 i 个目标函数; X 为决策向量, x_j ($j = 1, 2, \dots, m$) 为 X 的第 j 个分量, 即第 j 个决策变量, $x_{j \min}$ 和

收稿日期: 2008-03-18; 修回日期: 2008-07-30.

基金项目: 国家自然科学基金项目 (60774068); 教育部暨辽宁省流程工业综合自动化重点实验室开放课题项目.

作者简介: 牛大鹏 (1980—), 男, 山东淄博人, 助教, 博士生, 从事复杂工业过程建模与优化的研究; 王福利 (1957—), 男, 辽宁辽阳人, 教授, 博士生导师, 从事复杂工业过程建模、控制与优化等研究.

$x_{j \max}$ 分别为其最小值和最大值.

下面给出多目标优化中常用的几个基本定义:

定义1(优劣性) 若 u 部分小于 v , 即 $\forall i \in \{1, 2, \dots, n\}, (u_i < v_i) \quad (\exists i, u_i < v_i)$, 则称目标向量 $u = (u_1, \dots, u_j, \dots, u_n)$ 优于 $v = (v_1, \dots, v_i, \dots, v_n)$, 或称 v 劣于 u .

定义2(Pareto 支配) 若 X_1 和 X_2 为可行域内的决策向量, 当且仅当 $f(X_1)$ 优于 $f(X_2)$ 时, 则称解 X_1 Pareto 支配 X_2 , 或称解 X_2 被 X_1 Pareto 支配.

定义3(非劣最优解) 若某一可行的决策向量 X^* 不被可行域内任意解 Pareto 所支配, 则称 X^* 为多目标优化问题的非劣最优解.

所有非劣最优解组成的集合称为多目标优化问题的非劣最优解集, 记为 P_S ; 非劣最优解集对应的目标向量构成了多目标问题的非劣最优目标域, 记为 P_F . P_S 和 P_F 由多目标优化问题中具体的目标函数决定, 是一个确定的不变集合.

3 多目标混沌差分进化算法

3.1 DEMO 的关键操作

3.1.1 变异操作

变异成分为父代的差分向量, 每个向量由父代(第 G 代) 群体中两个不同的个体 (X_{i1}^G, X_{i2}^G) 生成. 差分向量定义为

$$D_{1,2} = X_{i1}^G - X_{i2}^G. \quad (2)$$

在差分向量的基础上进行变异操作, 其方程为

$$X_i^{G+1} = X_i^G + F(D_{1,2}). \quad (3)$$

其中: X_i^{G+1} 为变异个体, X_i^G 为父代个体, X_{i1}^G 和 X_{i2}^G 为与 X_i^G 不同的 2 个互不相同的个体; F 为缩放因子, 表示差分向量对下一代个体的影响程度.

F 取值大小对算法性能具有重要影响, 取值过大, 算法收敛速度变慢; 取值过小, 种群多样性降低, 算法易陷入局部最优. F 的取值范围通常选为 $[0, 1.2]$.

3.1.2 交叉操作

将群体中第 i 个个体 X_i^G 与 X_i^{G+1} 进行交叉操作, 产生试验个体 X_{iT} . 交叉操作的方程为

$$x_{iT} = \begin{cases} X_{ji}^{G+1}, & \text{rand}() < C_r; \\ X_{ji}^G, & \text{otherwise.} \end{cases} \quad (4)$$

其中: $\text{rand}()$ 为 $[0, 1]$ 之间的均匀随机数; $j = 1, 2, \dots, m$ 表示第 j 个变量(基因), m 为变量的维数; C_r 为交叉概率因子. C_r 越大, X_i^{G+1} 对 X_{iT} 的贡献越大, 算法进化速度越快, 也越易陷入局部最优; C_r 越小, X_i^G 对 X_{iT} 的贡献越大, 越利于保持种群的多样性, 提高算法的全局搜索能力.

可见, 保持种群多样性和提高算法收敛速度是一对矛盾, 需通过选择合适的参数来协调. C_r 的取

值范围通常选为 $[0, 1]$.

3.1.3 选择操作

对于多目标优化问题, 本文利用以下准则来选出子代个体:

1) 令试验个体 X_T 与父代个体 X_i^G 进行竞争, 若一方 Pareto 支配另一方, 则支配方进入新种群;

2) 若两个个体互不支配, 则将 X_T 和 X_i^G 同时加入新种群.

这种选择策略可能使新生成的种群规模变大(介于 N_P 和 $2N_P$ 之间, N_P 为进化种群规模), 应从中选择 N_P 个个体组成子代种群.

在选择之前, 需要对种群进行排序. 本文所采用的种群排序方法, 是根据个体的优劣水平和拥挤距离两个指标实现的. 首先找出种群中的所有非劣最优解, 其优劣等级都取 1; 然后找出剩余个体中的所有非劣最优解, 其等级都取 2. 重复进行, 直到种群中所有个体都具有相应的优劣等级为止.

为了维持种群的多样性, 保证生成均匀分布的 Pareto 最优解, 引入拥挤距离 (i) , 其计算方法为

$$(i) = \sum_{j=1}^n (i, j), \quad (5)$$

$$(i, j) = \frac{f_{j, i \text{ next}} - f_{j, i \text{ form}}}{f_{j \max} - f_{j \min}}. \quad (6)$$

其中: (i) 为种群中第 i 个个体的拥挤距离, (i, j) 为第 i 个个体在第 j 个目标分量上的拥挤距离; 将与第 i 个个体优劣等级相同的所有个体对应的第 j 个目标分量值从小到大排序, $f_{j, i \text{ next}}$ 为 $f_{j, i}$ (第 i 个个体在第 j 个目标分量上的值) 的后一相邻值, $f_{j, i \text{ form}}$ 为 $f_{j, i}$ 的前一相邻值, $f_{j \max}$ 和 $f_{j \min}$ 分别为最大值和最小值.

计算种群中的每一个体与同级别相邻两个个体之间的拥挤距离, 等级越高, 拥挤距离越小的个体在种群中的排序越靠后, 从排完序的种群中选择前 N_P 个个体组成子代种群.

3.2 CDEMO 的提出

混沌信号具有随机性和遍历性, 该性质可用于优化问题, 作为搜索过程中避免陷入局部最优的一种优化机制, 提高搜索效率. 本文利用混沌序列初始化种群, 并将文献[8]中的混沌替换操作引入多目标差分进化算法, 以克服算法在解决复杂多目标优化问题时易过早收敛的现象, 构建了多目标混沌差分进化算法(CDEMO).

3.2.1 混沌初始化

将常用的一维 Logistic 映射混沌模型^[9] 引入 CDEMO 种群的初始化, 即

$$r_{k+1} = \mu r_k (1 - r_k). \quad (7)$$

其中： $r_k \in (0, 1)$ ； μ 为控制参数，当 $\mu \in (3.56, 4.0]$ 时，式(7)处于混沌状态。 μ 的取值不同，混沌序列的搜索范围也不同，应根据需要选择合适的控制参数。

令式(7)中 r_k 的初值分别取 m 个介于 0 和 1 之间的随机数 $r_{ji}^0 (j = 1, 2, \dots, m)$ ，从而产生向量 $r_{i+1}^0 = (r_{1,i+1}^0, \dots, r_{j,i+1}^0, \dots, r_{m,i+1}^0)$ 的各个分量 $r_{j,i+1}^0$ ，即 $r_{j,i+1}^0 = \mu_1 r_{ji}^0 (1 - r_{ji}^0)$ ，
 $j = 1, 2, \dots, m, i = 1, 2, \dots, N_P - 1$ 。 (8)

取 $\mu_1 = 3.6$ ，此时混沌序列的覆盖范围可满足种群初始化的要求。

将产生的混沌变量映射到决策变量的取值范围 $(x_{j \min}, x_{j \max})$ ，得到初始种群第 i 个个体 $X_i^0 = (x_{1i}^0, \dots, x_{ji}^0, \dots, x_{mi}^0)$ 的第 j 个分量 x_{ji}^0 ，即

$$x_{ji}^0 = x_{j \min} + (x_{j \max} - x_{j \min}) r_{ji}^0, \quad j = 1, 2, \dots, m, i = 1, 2, \dots, N_P. \quad (9)$$

3.2.2 混沌替换操作

在进化过程中，当种群多样性较低时，用生成的混沌备用种群中的个体按一定概率替换当前种群中的个体，以引导算法跳出局部最优。进行替换时，等级越高、拥挤距离越小的个体被替换的概率越高。这样既有利于保留种群中的优秀个体，又能提高算法的多样性。对种群中的个体进行排序后，第 i 个个体的替换概率 P 由个体本身在种群中的顺序决定，即

$$P = i / N_P. \quad (10)$$

其中： i 为个体在种群中的排序， N_P 为种群规模。在 r_{ji}^0 的基础上，产生混沌备用种群第 i 个个体 $\bar{X}_i^{G+1} = (\bar{x}_{1i}^{G+1}, \dots, \bar{x}_{ji}^{G+1}, \dots, \bar{x}_{mi}^{G+1})$ 的第 j 个分量 \bar{x}_{ji}^{G+1} ，即

$$\bar{x}_{ji}^{G+1} = \mu_2 r_{ji}^G (1 - r_{ji}^G), \quad (11)$$

$$\bar{x}_{ji}^{G+1} = x_{j \min} + (x_{j \max} - x_{j \min}) r_{ji}^{G+1}. \quad (12)$$

其中： $G = 0, 1, \dots, G_{\max} - 1$ 为当前进化代数， G_{\max} 为最大进化代数。为了生成同式(8)产生的混沌序列不同的序列，从而增加种群的多样性，取控制参数 $\mu_2 = 3.8$ 。

需要说明的是，只有当种群多样性较低且未达到最优时，才进行混沌替换操作。为此，设计度量种群多样度的参数

$$G = \frac{1}{mN_P} \sum_{j=1}^m \frac{\sum_{i=1}^{N_P} |x_{ji}^G - x_{j^*}^G|}{x_{j \max} - x_{j \min}}, \quad (13)$$

其中 $x_{j^*} = \frac{1}{N_P} \sum_{i=1}^{N_P} x_{ji}^G$ 。由 G 的定义可知， G 在 $[0, 1]$ 之间， G 越小，种群越聚集在一起； G 越大，种群则处于随机搜索阶段。

通过判断种群中优劣等级为 1 的个体数目 N_{Fir} 是否与种群数目 N_P 相等，判断进化种群是否为非劣最优解集。当 $N_{\text{Fir}} = N_P$ 且 $G < G_{\min}$ 时，即调用替

换操作，令混沌备用种群中的个体 \bar{X}_i^G 按概率 P 替换当前种群中的相应个体 X_i^G 。

3.2.3 CDEMO 的实现步骤

Step1: 选定种群规模 N_P ，缩放因子 F ，最大进化代数 G_{\max} 和交叉概率因子 C_R 。

Step2: 令进化代数 $G = 0$ ，混沌初始化种群。

Step3: 进行变异操作，生成变异个体 X_i^{G+1} 。

Step4: 进行交叉操作，生成试验个体 X_T 。

Step5: 进行选择操作，生成下一代个体，并记录新生成种群中优劣等级为 1 的个体数目 N_{Fir} 。

Step6: 生成混沌备用种群。当 $N_{\text{Fir}} = N_P$ 时，判断进化种群多样性 G 是否小于 G_{\min} ，若是则对种群进行混沌替换操作。

Step7: 若 G 小于最大进化代数 G_{\max} ，则令 $G + 1$ ，并返回 Step3；否则，将种群中优劣等级为 1 的所有个体构成多目标优化问题的非劣最优解集，输出并结束运行。

4 数值实验

4.1 算法性能评价标准

多目标优化算法的性能可通过 2 个标准进行评价：1) 求得的非劣最优目标域应尽可能逼近真实非劣最优目标域 P_F ；2) 求得的非劣最优目标域应尽可能沿着 P_F 均匀分布。这也是大多数多目标优化算法的设计原则。本文通过逼近性指标和均匀性指标^[7]，分别度量 CDEMO 算法的逼近性和均匀性，并与有关文献中的结果进行比较。

算法的逼近性可通过所获得的非劣最优解集对应的目标向量和真实非劣最优目标域 P_F 之间的距离 d_i 来度量，即

$$d_i = \min_{j \in P_F} |x_i - x_j| \quad (14)$$

其中： d_i 是求得的非劣最优解集中第 i 个 Pareto 解对应的目标向量到 P_F 中的最小欧氏距离， Q 表示所得非劣解的个数。显然， d_i 值越小，算法逼近最优目标域的程度越高。

算法的均匀性可通过 d 来度量，即

$$d = \frac{d_f + d_l + \sum_{i=1}^{Q-1} |d_i - \bar{d}|}{d_f + d_l + (Q - 1) \bar{d}} \quad (15)$$

其中： d_i 是求得的非劣最优解集对应的目标向量集中两个相邻向量间的欧氏距离， \bar{d} 是所有 d_i 的均值， d_f 是真实非劣最优目标域 P_F 中极值目标向量， d_l 是求得的非劣最优目标向量集中边界向量间的距离。 d 值越小，说明所求得的非劣解的多样性越好。

4.2 实验结果与分析

采用文献[10]中的测试函数 ZDT1, ZDT2 和



ZDT3 测试本文算法的性能. 这 3 个函数的决策变量 x 都取 30 维, 即 $m = 30$. 它们的理论非劣最优解均为 $0 \leq x_1 \leq 1, x_i = 0, i = 2, 3, \dots, m$.

在用 CDEMO 求解多目标优化问题时, 种群规模太小, 则算法不易找到最优解; 种群规模太大, 则算法收敛速度缓慢. 经实验研究, 种群规模可取决策变量个数的 3 ~ 10 倍, 此时算法可达到较好的性能. 本实验中取种群规模 $N_p = 100$. 考虑到缩放因子 F 和交叉概率因子 C_R 对算法性能的影响, 兼顾种群多样性和算法收敛速度, 在多次实验的基础上, 取 $F = 0.6, C_R = 0.5$. 为了同有关文献中的实验相统一, 取最大进化代数 $G_{max} = 250$.

表 1 ~ 表 3 给出了 CDEMO 对 3 个测试函数分别运行 10 次, 逼近性标准和均匀性标准的均值和范围, 同时给出了实数编码 NSGA-, SPEA, PDEA 和 DEMO/Parent 的相应实验结果^[7]. 从表中可以看出, CDEMO 对 3 个测试函数的逼近性比 NSGA-, SPEA 和 PDEA 都好, 特别是对 ZDT3 更具优势; 与 DEMO/Parent 相比, 除了在 ZDT3 上算法的逼近性接近外, 在其余 2 个函数上都好于 DEMO/Parent. 对 3 个测试函数在解的均匀性上, CDEMO 比 NSGA-, SPEA, PDEA 和 DEMO/Parent 都有较大的优势.

表 1 测试函数 ZDT1 评价准则的统计结果

算 法	逼近性指标	均匀性指标
NSGA-	0.033482 ± 0.004750	0.390307 ± 0.001876
SPEA	0.001799 ± 0.000001	0.784525 ± 0.004440
PDEA	N/A	0.298567 ± 0.000742
DEMO/Parent	0.001083 ± 0.000113	0.325237 ± 0.030249
CDEMO	0.000705 ± 0.000079	0.198237 ± 0.027840

表 2 测试函数 ZDT2 评价准则的统计结果

算 法	逼近性指标	均匀性指标
NSGA-	0.072391 ± 0.031689	0.430776 ± 0.004721
SPEA	0.001339 ± 0.000000	0.755148 ± 0.004521
PDEA	N/A	0.317958 ± 0.001389
DEMO/Parent	0.000755 ± 0.000045	0.329151 ± 0.032408
CDEMO	0.000523 ± 0.000057	0.215287 ± 0.028531

表 3 测试函数 ZDT3 评价准则的统计结果

算 法	逼近性指标	均匀性指标
NSGA-	0.114500 ± 0.007940	0.738540 ± 0.019706
SPEA	0.047517 ± 0.000047	0.672938 ± 0.003587
PDEA	N/A	0.623812 ± 0.000225
DEMO/Parent	0.001178 ± 0.000059	0.309436 ± 0.018603
CDEMO	0.001095 ± 0.000072	0.254082 ± 0.023280

图 1 ~ 图 3 分别为本文算法求得的 3 个测试函数的非劣最优目标域与相应的真实非劣最优目标域的对比, 所求得的非劣最优目标域为从 10 次运行中随机选择的 1 次运行结果. 从图中可以直观地看出, 利用 CDEMO 得到的非劣最优目标域, 能很好地逼近真实的非劣最优目标域, 并且分布也很均匀.

上述分析表明, 在进化过程中引入混沌替换操作, 不仅有助于保持种群的多样性, 起到维持解集均匀性的作用, 而且在一定程度上增强了算法的搜索功能.

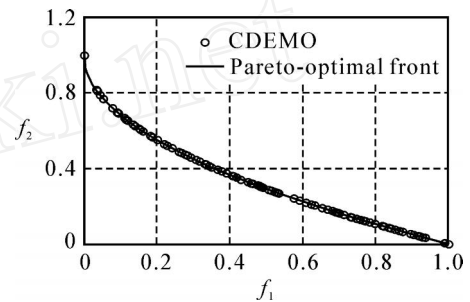


图 1 ZDT1 的仿真曲线

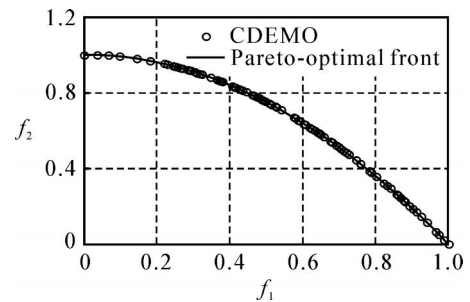


图 2 ZDT2 的仿真曲线

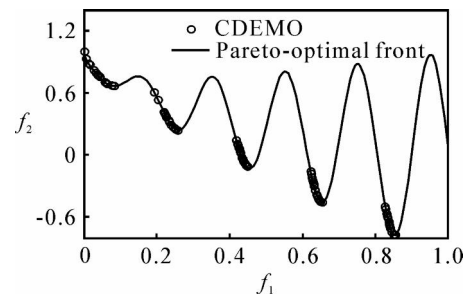


图 3 ZDT3 的仿真曲线

5 结 论

本文基于混沌原理, 在基本差分进化算法的基础上, 给出一种多目标差分进化算法 (CDEMO). 在混沌初始化种群之后, 利用差分进化操作产生新的种群, 并引入混沌替换操作, 以保持种群的多样性. 测试函数的数值仿真结果表明, 本文算法无论在解集的逼近性还是均匀性方面, 都优于有关文献中的多目标进化算法, 这表明所提出的 CDEMO 算法在求解多目标优化问题时具有一定的优势.

(下转第 370 页)

统的粗糙控制方法,实现了柔性制造系统的粗糙控制.实例分析表明所提出的方法是有效的.本文方法亦可推广到混杂间歇生产过程及其他混杂系统领域.

参考文献(References)

- [1] Chen H, Yang P, Yao D D. Control and scheduling in two-station queuing network: Optimal policies heuristics [J]. *Queueing Systems*, 1994, 18(4): 301-332.
- [2] Zhang R, Phillis Y A. Multiple control policies of two-station production network with two types of parts using fuzzy logic [C]. *IEEE Int Conf on Robotics and Automation*. Leuven, 1998: 2759-2764.
- [3] David R, Alla H. Hybrid Petri nets [C]. *European Control Conf*. Grenoble, 1991: 1472-1477.
- [4] Balduzzi F, Gua A, Menga G. First-order hybrid Petri nets: A model for optimization and control [J]. *IEEE Trans on Robotics and Automation*, 2000, 16(14): 382-399.
- [5] Demongodin I, Koussoulas N T. Differential Petri nets: Representing continuous systems in discrete event world [J]. *IEEE Trans on Automatic Control*, 1998, 43(4): 573-579.
- [6] Demongodin I, Prunet F. Batches Petri nets [C]. *Proc of Computer in Design, Manufacturing and Production*. Paris, 1993: 29-37.
- [7] 廖伟志, 古天龙. 基于一种新型混杂 Petri 网的混杂系统建模与控制 [J]. *控制与决策*, 2007, 22(4): 366-372.
(Liao W Z, Gu T L. Modeling and control for hybrid system based on general hybrid Petri nets [J]. *Control and Decision*, 2007, 22(4): 366-372.)
- [8] 马志锋, 邢汉承, 郑晓妹. 粗糙控制中的规则获取策略研究 [J]. *工业仪表与自动化装置*, 2000, 177(6): 58-61.
(Ma Z F, Xing H C, Zheng X M. Research on the strategy of rule acquisition in rough control [J]. *Industrial Instrumentation & Automation*, 2000, 177(6): 58-61.)
- [9] 常晓艳, 刘振娟. 基于粗糙集属性约简的过程控制规则提取 [J]. *仪器仪表学报*, 2004, 25(增): 881-883.
(Chang X Y, Liu Z J. Rule abstract in process control based on attribute reduction of rough set [J]. *Chinese J of Scientific Instrument*, 2004, 25(S): 881-883.)
- [10] 张赢, 李琛. 基于粗糙集理论的神经网络研究及应用 [J]. *控制与决策*, 2007, 22(4): 462-464.
(Zhang Y, Li C. Research on neural networks based on rough set theory [J]. *Control and Decision*, 2007, 22(4): 462-464.)

(上接第 364 页)

参考文献(References)

- [1] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms [J]. *Evolutionary Computation*, 1994, 2(3): 221-248.
- [2] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA- [J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(2): 182-197.
- [3] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach [J]. *IEEE Trans on Evolutionary Computation*, 1999, 3(4): 257-271.
- [4] Rainer S, Price K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces [J]. *J of Global Optimization*, 1997, 11(4): 341-359.
- [5] 刘波, 王凌, 金以慧. 差分进化算法研究进展 [J]. *控制与决策*, 2007, 22(7): 721-729.
(Liu B, Wang L, Jin Y H. Advances in differential evolution [J]. *Control and Decision*, 2007, 22(7): 721-729.)
- [6] Madavan N K. Multiobjective optimization using a Pareto differential evolution approach [C]. *Congress on Evolutionary Computation*. Piscataway: IEEE Service Center, 2002, 2: 1145-1150.
- [7] Robić T, Filipič B. DEMO: Differential evolution for multiobjective optimization [C]. *Lecture Notes in Computer Science*. Berlin: Springer, 2005: 520-533.
- [8] 胡桂武, 彭宏. 利用混沌差分进化算法预测 RNA 二级结构 [J]. *计算机科学*, 2007, 34(9): 163-166.
(Hu G W, Peng H. An algorithm-based chaos differential evolution for predicting RNA secondary structure [J]. *Computer Science*, 2007, 34(9): 163-166.)
- [9] Coelho L S, Mariani V C. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect [J]. *IEEE Trans on Power Systems*, 2006, 21(2): 989-996.
- [10] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results [J]. *Evolutionary Computation*, 2000, 8(2): 173-195.