文章编号: 1001-0920 (2011) 03-0457-07

分阶段二次变异的多目标混沌差分进化算法

王筱珍[a]，李 鹏[b]，俞国燕[b]

(广东海洋大学 a. 信息学院，b. 工程学院，广东 湛江 524088)

摘 要: 提出一种结合分阶段二次变异和混沌理论的改进差分进化 (DE) 算法, 以解决多目标约束优化问题. 其核心思想是, 在 DE 进化前期采用基于非支配解的随机二次变异来提高算法的全局寻优能力, 进化后期采用基于非支配解的混沌二次变异来提高 DE 的局部寻优能力. 通过对典型测试问题的仿真实验验证了所提出的算法能在全局搜索性能与局部搜索性能之间维持较好平衡, 而且保持了 DE 算法的简洁性能, 其收敛性、分布度和均衡性均优于标准 DE.

关键词: 差分进化；混沌；分阶段二次变异；非支配解

中图分类号: TP274 **文献标识码:** A

Multi-objective chaotic differential evolution algorithm with grading second mutation

WANG Xiao-zhen[a], *LI Peng*[b], *YU Guo-yan*[b]

(a. Information College, b. Engineering College, Guangdong Ocean University, Zhanjiang 524088, China. Correspondent: YU Guo-yan, E-mail: yugy@gdou.edu.cn)

Abstract: To solve the multi-objective constraint optimization problem, this paper proposes an advanced differential evolution(DE). In the proposed algorithm, grading second mutation and chaotic theory are combined into standard DE. At early evolution process of DE, random second mutation based on non-dominance Pareto solution is adopted in order to improve global exploring ability. And in the later evolution process, the chaotic second mutation based on non-dominance Pareto solution is added into DE evolution operation in order to enhance local searching ability of algorithm. By testing benchmarks functions, simulation results show that, this algorithm has better convergence and distribution property, and is superior to standard DE in keeping balance between diversity and convergence.

Key words: differential evolution; chaotic; grading second mutation; non-dominance solution

1 引 言

差分进化算法 (DE) 自 Storn 和 Price 于 1996 年提出以来, 便以其受控参数少、收敛速度快、鲁棒性强等优点成为最受关注的进化算法之一[1-2]. 然而, DE 在收敛速度与搜索鲁棒性之间存在冲突, 即伴随快速收敛性的同时, DE 有着更高的概率使搜索陷入局部最优和早熟收敛. 因为算法后期个体之间的差异信息变小时, 若参数设置不当, 进化会变得缓慢甚至出现进化停滞或早熟现象.

在用 DE 解决多目标优化问题时, 大都采用基于 Pareto 排序的贪婪选择策略, 这使得每代群体中的较优个体以较大概率将结构保存下来, 因而在求解具有大量局部优点的问题时, 种群极易陷入局部最优. 对此, 人们提出了多种形式的改进 DE 来提高算法性能, 将混沌序列与 DE 结合便是其中一种. 由于混沌的非重复性, 在进行全局搜索时它比依赖于概率的随机遍历搜索具有更高的速度. 最近, 作为一种可以避免陷于局部最优的特殊机制, 混沌序列的各态历经性已被用于进化算法的多种设计. 已有许多实例证明, 将一些混沌序列应用于进化算法可以有效提高算法的全局搜索能力, 加快收敛速度[1-5].

本文以求解多目标约束优化问题为例, 对混沌理论在 DE 算法中的应用进行了深入研究, 提出一种结合二者优点的混沌差分进化算法. 该算法为提高 DE 在解决多目标优化问题时的整体性能, 在进化前期采用基于非支配解的随机二次变异来提高 DE 的全局寻优能力, 在进化后期则采用基于非支配解的混沌二次变异来提高 DE 的局部寻优能力. 实验结果表

收稿日期: 2009-12-08; **修回日期:** 2010-01-21.

基金项目: 国家自然科学基金项目(50675069); 广东省海洋渔业局项目(A200899G02).

作者简介: 王筱珍(1963−), 女, 副教授, 从事自动化、智能算法及应用等研究; 俞国燕(1970−), 女, 教授, 博士, 从事进化算法与智能设计等研究.

明了所提出的改进DE算法的有效性.

2 混沌理论及其在DE中的应用简介

2.1 混沌理论简介

混沌是存在于非线性系统中的一种较为普遍的现象, 具有随机性、遍历性、初始条件敏感性等特点, 能在一定范围内按其自身“规律”不重复地遍历所有状态. 混沌的内在随机性使得搜索可以避免陷入局部极小点, 遍历性则保证其能找到全局最优点. 但是, 完全基于混沌的优化搜索具有一定的盲目性, 当出现搜索起始点选择不合适、遍历区间很大、控制参数及其控制策略选取不合适等情况时, 混沌优化的搜索精度会降低, 出现很难达到最优值或混沌优化失效的情况[6]. 因此, 很多研究工作试图将混沌搜索与其他算法相结合进行优化搜索, 如前述将混沌引入进化算法便取得了较好的效果. 混沌序列在进化算法中的应用主要是用于替代随机数来增强启发式优化算法的性能, 但不同的混沌映射具有不同的遍历特性和分布范围特性, 它们对于寻优过程具有较大影响. 下面给出2种最常用的混沌映射公式.

2.1.1 Logistic 混沌映射模型

Logistic 混沌映射模型为

$$x_{k+1} = \lambda x_k(1 - x_k), \tag{1}$$

式中 λ 为控制参数. λ 的取值不同, 则混沌序列的搜索范围不同, 应根据需要选择合适的控制参数. 当取 $\lambda = 4$, 且 $x_1 \neq \{0.25, 0.5, 0.75, 1\}$ 时, 式(1)是一个混沌系统[1,7], x_k 在[0,1]遍历. 这是混沌与差分进化结合时应用最多的一种映射混沌模型, 是一种具有非均匀概率分布的混沌映射.

将式(1)产生的混沌序列用于优化算法时, 因 x_k 在[0,1]遍历, 所以为了防止进化过程中出现单侧搜索现象, 可将增加的混沌扰动量修改为 $2[0.5 - t_i(k)]$. 这样, 混沌序列在 $[-1, 1]$ 遍历, 使得搜索可在局部最优点附近产生正负两个方向的扰动, 有利于扩大搜索范围, 摆脱局部极值点[8].

2.1.2 Tent 混沌映射模型

Tent 映射, 又称帐篷映射, 其数学表达式为 $x_{n+1} = a - 1 - a|x_n|$. 当 $a \leqslant 1$ 时, 系统处于稳定状态; 当 $a > 1$ 时, 系统处于混沌状态; 当 $a = 2$ 时为中心 Tent 映射. 其数学表达式为

$$x_{k+1} = \begin{cases} 2x_k, \ 0 \leqslant x_k \leqslant 1/2; \\ 2(1 - x_k), \ 1/2 < x_k \leqslant 1. \end{cases} \tag{2}$$

Tent 映射结构简单, 比 Logistic 映射具有更好的遍历性, 其迭代速度快于 Logistic 映射, 但迭代序列中存在小周期、不稳周期点等不足, 需对其进行改进后才能更好地应用于优化问题[9]. 另外, 其他如 Lorenz 系统、Lozi 混沌映射、Zaslavsky 映射和 Ikeda 等混沌映射公式也经常被用于替代产生随机数[10], 并取得了较好效果.

2.2 混沌理论在DE算法中的应用

混沌差分进化算法已经得到越来越多的关注, 并取得了一定成果. 目前将二者相结合的改进算法主要可分为以下3类:

1) 混沌序列产生DE的初始群体或子群体. 主要采用不同的混沌映射模型在可行域内产生DE的初始种群或子群体, 这种方式既不改变初始化时所具有的随机性本质, 又提高了初始群体的差异性和多样性, 增强了DE的全局寻优能力[5,7].

2) 混沌序列设置DE的控制参数. 主要是指利用混沌序列确定DE的交叉率 C_R 和变异因子 F 两个控制参数. DE算法的搜索性能在很大程度上依赖于其控制参数的选取, F 和 C_R 则是影响DE收敛性的关键参数. 在实际应用中很难确定合适的 C_R 和 F 值, 而常用的实验方式确定法则需花费大量的运行时间. 在标准DE中, 控制参数确定之后, 它在整个搜索过程中是一个静态值, 这很容易使算法在进化后期陷入局部最优. 若采用混沌序列在进化过程中自适应调整参数 C_R 与 F, 则可以保证所产生的控制参数在经验的取值范围内达到完全遍历, 从而使算法比采用固定 C_R 或 F 的标准DE更容易避免陷入局部最优, 提高全局收敛性. 用混沌序列确定 C_R 与 F 的值具有如下优点: ① 用户不需要推测 F 和 C_R 到底取什么值最好; ② 这种自适应调整参数 F 和 C_R 的规则非常简单. 其中: Yuan 等人[4]采用混沌序列来产生DE的 C_R 和 F 参数; Leandro dos 等人[1,3]则研究了用不同的混沌映射模型来确定 F 的算法效果.

3) 基于混沌序列的DE进化操作. 标准DE算法的特点是初期探索能力较强, 但随着演化代数的增加, 群体之间的差异度和多样性可能迅速减小. 此时若缺少外部扰动, 算法的后期收敛速度会变慢, 容易陷入局部最优解, 出现早熟收敛. 对此, 可采用基于混沌序列的DE进化操作, 其改进方式有: ① 当目标函数值连续几代进化都没有改进时, 用初始化时产生的混沌序列代替部分群体来促进进化, 跳出局部最优[5,7]; ② 当进化种群个体的适度值连续几代未改善时, 则利用混沌序列对当前种群中最优个体进行遍历性局部搜索, 并将局部搜索到的优化个体替换原最优个体, 即引入混沌序列重新初始化陷入局部极小点的个体, 迭代产生局部最优解的邻域点, 帮助个体逃离束缚并快速搜寻到最优解[4,11-12]. 其中后者首先利用差分进化算法进行全局搜索, 再用混沌搜索进行局部搜索,

充分利用了差分进化算法和混沌搜索各自的优点, 使算法在增加种群多样性的同时改进了种群的最优个体. 使用时需注意的是: 在最佳个体基础上的混沌局部寻优, 最好采用能够产生对称区域的混沌映射公式.

对于上述 3 种不同的混沌、DE 结合方法, 前 2 种方式较为简单, 尤其是利用混沌序列替代原先随机产生个体的应用非常方便, 只需选择合适的混沌映射模型产生混沌序列替代原随机数即可. 但是, 对于第 3 种基于混沌序列的 DE 进化操作, 其执行相对复杂, 原因在于: 与混合 DE (HDE) 只在群体多样性测量结果不能满足所期望的值时才执行迁移一样, 上述 2 种改进方式使用的关键之一是判断什么条件下进行混沌进化操作. 一般可通过计算种群的多样性因子或最优个体适度值连续几代不变化或其变化小于某一阈值来判断算法是否陷入局部极值点. 若是, 则将混沌扰动量引入进化种群, 并随着搜索过程的进行逐渐调整扰动幅度来实现优化. 第 2 种方式 (基于最佳个体基础上的混沌局部寻优方式) 是当前基于混沌序列的 DE 进化操作最常采用的, 由于混沌序列提供了在最佳个体附近进行均匀分布、一定数量的局部搜索, 使其可快速有效地跳出局部最优. 因此, 这种方法有效地改进了 DE 求解单目标优化问题的算法性能. 但对于多目标优化问题, 由于存在多个 Pareto 解, 导致如何执行基于最佳个体的局部寻优存在一定难度.

3　多目标差分进化算法

标准 DE 算法的演化过程与遗传算法类似, 包括种群初始化、变异、交叉和选择操作. 算法的伪代码表示如下:

```
Initialization
Evaluation
Repeat
    Mutation
    Crossover
    Evaluation
    Selection
Until(终止原则满足)
```

本文讨论的是解决多目标约束优化问题的 DE 算法. 采用双群体约束处理技术, 即对交叉、变异后生成的新个体进行判断, 将其分为可行种群和不可行种群, 用群体 PopF 保存搜索过程中遇到的可行解, 用 PopIF 保存搜索过程中遇到的占优不可行解. 将每代可行群体 PopF 中的最佳个体保存到 g_{best}, 并将其用于改进 DE 算法的变异操作, 以利用搜索过程中的最佳可行解信息来加快 DE 的收敛速度. 详见文献 [13]. 其他关键操作如下.

3.1　变异操作

变异操作由父代 3 个不同的向量组成, 即

$$C = X_{r1} + F(X_{r2} - X_{r3}). \tag{3}$$

式中: $X_{r1}, X_{r2}, X_{r3} \in \text{PopF}$, 且分别属于 3 个不同小种群的不同个体, 其划分按可行解的 Pareto 排序将前 1/3 部分、后 1/3 部分和中间部分各组成一个小种群而得到; F 为缩放因子, 其取值大小对算法性能具有重要影响, 范围通常选为 [0.3,0.8].

3.2　交叉操作

变异操作产生的个体 $C=(c_1, c_2, \cdots, c_m)$ 与父代个体 $X[i]=(x_1, x_2, \cdots, x_m)$ 进行交叉操作, 以生成一试验个体 $P_r=(x_1', x_2', \cdots, x_m')$. 其中

$$x_j' = \begin{cases} c_j, \ \text{rand}(\ ) < C_R; \\ x_j, \ \text{rand}(\ ) \geqslant C_R. \end{cases} \tag{4}$$

式中: rand( ) 为 [0,1] 中的随机数, $j \in [1, m]$, m 为变量个数. 为保证交叉所产生的新解的利用性, 在交叉操作完成后, 检查交叉之后生成的所有个体之中是否有重复的个体, 如果有, 则随机产生新个体来替换重复个体, 以保证产生新解的利用性. 判断交叉操作生成的新个体 P_r 的可行性, 并统计新生成个体的可行解总数 M_1 与不可行解个体数 M_2, 可行个体与不可行个体分别存放于数组 I_P, I_{NP} 中.

3.3　选择操作

选择操作在原群体与新产生群体共同组成的群体中进行, 但由于处理的是带约束的多目标优化问题, 并采用了基于双群体的约束处理技术, 选择操作无法采用标准 DE 基于贪婪策略的简单择优方式. 因此, 借鉴 NSGA II 中的 Pareto-based 排序和拥挤距离分选技术, 对于可行群体与不可行群体采用不同策略进行选择操作, 以产生新一代的可行种群与不可行种群.

3.3.1　新可行解群的确定

若 $N_F + M_1 \leqslant N_1$ (即原群体中的可行个体数 N_F 与新产生的可行个体数 M_1 之和小于可行种群规模 N_1), 则保留所有可行解作为下一代群体的新可行解群 (以数组 PopF 保存, 并更新 $N_F = N_F + M_1$). 若可行解数 $N_F + M_1 > N_1$, 则选择前 N_1 个目标函数值较优的个体作为新的可行解群 (以数组 PopF 保存). 其选择方法如下: 首先两两比较可行种群中的各个个体, 如果存在两个个体 $X_i, X_j \in \text{PopF}$, 满足 $F(X_i)$ Pareto 优于 $F(X_j)$, 则将个体 X_j 删除; 如果不存在, 即可行解集合 PopF 中任意两个个体所对应的目标向量都不可比较, 则计算 PopF 中任意两个个体间的距离, 随机删除具有最小拥挤距离的个体 (采用欧氏距离, 即各个子目标之间数值的平方和).

3.3.2 新不可行解群的确定

若 $I_{NF}+M_2\leqslant N_2$, 则将原群体中的不可行个体与新产生的不可行个体共同组成新的不可行解种群, 并更新 $I_{NF}=I_{NF}+M_2$; 若 $I_{NF}+M_2>N_2$, 则保留较佳的前 N_2 个不可行个体. 方法如下: 计算原不可行种群 PopF 与新不可行种群组成的总不可行种群中任意两个个体的约束向量, 如果存在两个个体 X_i, X_j, 满足约束向量 $(C(X_i), N(X_i))$ Pareto 优于约束向量 $(C(X_j), N(X_j))$, 则删除 X_j; 如果不存在, 则删除满足 $C(X_i)=\max\limits_{X\in\mathrm{PopIF}} C(X_i)$ 的个体, 直至保留 N_2 个不可行解.

4 基于分阶段非支配解二次变异的混沌多目标 DE

已有诸多文献研究表明[11-13], 通过对当代种群最佳个体的二次变异或迁移操作可在局部最优点附近改善种群多样性, 提高算法对全局最优点的搜索能力. 前述基于最佳个体的混沌变异操作就是通过在最佳个体基础上的迁移来增加新个体, 从而保证了候选个体的多样性.

对于多目标优化问题, 由于每代群体都可能存在多个不同数目的非支配解, 很难选择合适的最佳个体进行二次变异. 尤其在算法初期, 即使选择了合适的非支配解个体引入混沌序列进行二次变异搜索, 在迭代中产生当前最优解的许多邻域点, 其种群多样性也只是在局部区域内有所增加, 并不能有效扩大对整个解空间的搜索范围, 其效果甚至比不上随机产生新个体来替代种群中的部分非优化个体. 例如, 前述用混沌序列产生的备选群体在 DE 进化多样性不佳时替代当代部分较差个体. 另一方面, DE 的主要缺陷是其机制在优化进程的每个阶段只进行了有限数目的全局探索, 若探索数目不足以产生新希望解, 则算法会严重妥协, 在求解多目标优化问题时算法极易陷入局优点. 在实际测试时发现, 当将种群规模按决策变量个数的 3～10 倍这一常用原则选取时, 对于低维多目标约束优化问题, 在运行前期找到的可行 Pareto 非支配解的数量, 相对而言偏少, 这对算法的全局优化性能造成了极大影响. 因此, 为了加强 DE 的性能, 应该有替代的多样性注入机制来支持原始的 DE 机制, 以提高优化进程的成功延续. 对此, 本文提出对所存档非支配解个体进行分阶段的二次变异操作来进一步提高 DE 性能. 算法的核心思想是: 在 DE 进化前期采用基于非支配解的随机二次变异来提高算法的全局寻优能力, 进化后期则采用基于非支配解的混沌二次变异来提高 DE 的局部寻优能力. 具体如下:

用一外部存档集专门保存每代群体中的 Pareto 非支配解, 其规模与 g_{best} 相同, 对所保存的最优非支配解进行二次变异. 进化前期和后期分别采用以下两个不同公式:

$$X'=X_{\mathrm{best}}\pm 0.1L\mathrm{e}^{-\mu t}. \tag{5}$$

其中: L 为变量取值范围, μ 为一常量, t 为进化代数[13].

$$X'=X_{\mathrm{best}}+y_k, \tag{6}$$

其中 y_k 是用前述混沌映射模式产生的混沌序列. 为保证产生的每一个新解只在其原解附近进行对称变化, 可按前述方法对 Logistic 混沌映射模型进行修改, 使其产生的序列值在 $(-1,+1)$ 之间.

在进化前期, 按式 (5) 对外部存档集中的全部 Pareto 非支配解实施二次变异, 以产生相同数目的新个体; 在进化后期, 则按式 (6) 选择具有代表性、个体间隔距离大的非支配解实施混沌二次变异, 对所选中的非支配解, 在其附近用混沌方法进行一定次数的遍历性局部搜索, 搜索步长按混沌规律自动无重复地变化. 判断执行二次变异产生的新个体的可行性, 并统计新生成个体的可行解总数 M_1 和不可行解个体数目 M_2, 再次将可行个体与不可行个体分别存放于数组 I_P, I_{NP} 中.

基于非支配解集的二次变异操作有以下优点: 首先, 符合生物进化中含优良基因的非支配解个体具有更多繁衍后代的机会, 极大提高了算法在有限时间的收敛速度. 由于 DE 的进化速度依赖于变异公式中的 g_{best} 所包含的优良基因信息, 在理论上这种单一的最优个体保留策略更多的是保证算法的收敛性 (即保证算法能收敛于全局最优解), 但对算法收敛速度的影响尚无明确证明. 其次, 此种方式增加了可供选择操作的群体规模, 这等同于变相增加了群体规模, 由此有效地提高了解集的多样性. 因此, 基于非支配解集的变异操作可同时提高算法的收敛速度和多样性,尤其是进化后期的混沌二次变异极大地提高了 DE 算法后期的局部寻优能力.

5 数值仿真实验与分析

5.1 算法流程

Step 1: 设置最大迭代次数, 算法参数 C_R, F 及可行种群与不可行种群规模 N_1 和 N_2, 初始化最佳个体集, 置当前迭代计数器 $t=1$;

Step 2: 判断是否满足终止条件, 满足则转 Step 7, 否则转 Step 3;

Step 3: 进行标准变异、交叉操作, 生成新的群体;

Step 4: 按前述方法进行分阶段二次变异操作, 生成新的群体;

Step 5: 进行选择操作, 生成新的可行群体与不可行群体;

Step 6: 根据新生成的群体 PopF 更新最佳个体集 g_{best}, 置 $t=t+1$, 转 Step 2;

Step 7: 输出最佳个体集.

5.2 测试函数

为验证本文算法的有效性, 选用表 1 所列 5 个带约束的 Benchmark 多目标约束优化问题来测试算法性能.

5.3 算法性能评价标准

在评价多目标优化算法时, 通常需考虑所找到的非劣解集是否收敛于 Pareto 最优解集及这些非劣解是否均匀分布于整个均衡面上. 本文采用 G_D, S, D_{1_R} 3 个指标分别评价算法的收敛性、分布性以及收敛性和多样性的均衡性能.

1) 指标 G_D (generational distance)[14]. 理想情况下, 多目标优化进化算法的求解过程是一个不断逼近最优边界、最终达到最优边界的过程. 但在实际应用中, 并不能保证找到真正的 Pareto 最优解, 而是尽可能找到一个很好的近似解. G_D 是用来表示所求得的 Pareto 前沿与真实 Pareto 前沿之间的间隔距离, 用来衡量算法的收敛性, 其值越小越好. 定义为

$$G_D = \Big(\frac{1}{n}\sum_{i=1}^{n} d_i^p\Big)^{1/p}. \tag{7}$$

其中: n 为所求得 Pareto 前沿解的个数, $p=2$, d_i 为所求得 Pareto 前沿中的个体 i 与真实 Pareto 前沿中最近个体之间的欧氏距离 (在目标空间上).

2) S (the metric of spacing)[14]. 是由 Schott 于 1995 年提出的一种衡量算法分布度的指标, 其值越小, 表明所求得点的分布均匀性越好. 若 $S=0$, 则表示 Pareto 前沿中所有解点呈均等分布. 指标定义为

$$S = \Big[\frac{1}{n}\sum_{i=1}^{n}(d_i-\bar{d})^2\Big]^{1/2}\Big/\bar{d}, \tag{8}$$

$$d_i = \min_j\Big(\sum_{m=1}^{M}|f_k^i(x)-f_k^j(x)|\Big). \tag{9}$$

其中: n 为所求得 Pareto 前沿解的个数; d_i 为所求得 Pareto 前沿中个体 i 与真实 Pareto 前沿中最近个体之间的欧氏距离 (在目标空间上); M 为目标维数, 本文取 $M=2, m=1,2,\cdots,M$; $\bar{d}$ 为所有 d_i 的平均值, i, $j=1,2,\cdots,n$.

3) D_{1_R}[15]. D_{1_R} 是同时衡量收敛性和分布性的指标, 是通过将所得 Pareto 优解与一参考 Pareto 解集进行对比得到的结果. D_{1_R} 值越小, 表明算法在收敛性和分布性上的平衡维持能力越好.

$$D_{1_R}(A_j) = \frac{1}{|Z^*|}\sum_{y\in z^*}\min\{d_{xy}|x\in A_j\}, \tag{10}$$

$$d_{xy} = \sqrt{(f_1(x)-f_1(y))^2+(f_2(x)-f_2(y))^2}. \tag{11}$$

其中: A_j 为所求得的 Pareto 解集, Z^* 和 $|Z^*|$ 分别为 Pareto 参考解集及其所含解的个数.

5.4 实验结果与分析

为验证本文提出的分阶段二次混沌变异改进 DE 算法的效果, 将其与进化全程采用基于非支配解的混沌二次变异改进 DE 算法和基本 DE 进行比较, 并采用相同的决策规模和相同的迭代次数.

对于算法最后所得 Pareto 优解的质量或有效性, 在收敛性能上, 主要通过计算所求解集到参照集的最小间隔距离来衡量 (即前述 G_D 指标), 该距离越小, 表明所找到解集与最优解集的趋近度越高. 在解的分布均匀性上, 则通过前述 S 指标值反映, 其值越小, 表明所求得优解的分布均匀性越好, 并采用 D_{1_R} 指标综合评估所求得解在收敛性和均匀性上的总体性能. 其中参照集理论上应是真实的 Pareto 最优解集, 但由于大部分 MOP 问题的真实 Pareto 最优解是未知的,

表 1　测试函数

	函　数	约　束
F_1	$\min f_1(x)=x_1$ $\min f_2(x)=x_2$	$x_1^2+x_2^2-1-0.1\cos(\arctan x_1/x_2)\geqslant 0$ $(x_1-0.5)^2+(x_2-0.5)^2\leqslant 0.5$ $0\leqslant x_1\leqslant\pi,\ 0\leqslant x_2\leqslant\pi$
F_2	$\min f_1(x)=-x_1^2+x_2$ $\min f_2(x)=0.5x_1+x_2+1$	$x_1+x_2-13/2\leqslant 0,\ x_1+x_2-15/2\leqslant 0$ $5x_1+x_2-30\leqslant 0,\ 0\leqslant x_1\leqslant 7,\ 0\leqslant x_2\leqslant 7$
F_3	$\min f_1(x)=x_1$ $\min f_2(x)=(x_2+1)/x_1$	$-9x_1-x_2+6\leqslant 0,\ -9x_1+x_2+1\leqslant 0$ $0.1\leqslant x_1\leqslant 1,\ 0\leqslant x_2\leqslant 5$
F_4	$\min f_1(x)=4x_1^2+4x_2^2$ $\min f_2(x)=(x_1-5)^2+(x_2-5)^2$	$-(x_1-8)^2-(x_2+3)^2+7.7\leqslant 0$ $(x_1-5)^2+x_2^2-25\leqslant 0$ $0\leqslant x_1\leqslant 5,\ 0\leqslant x_2\leqslant 3$
F_5	$\min f_1(x)=x_1$ $\min f_2(x)=g(x)\times(1-f_1(x)/g(x))$ $g(x)=41+\sum_{i=2}^{5}(x_i^2-10\cos 2\pi x_i)$	$a\lvert\sin(b\pi(\sin\theta(f_2(x)-e)+\cos\theta\times f_1(x))^c)\rvert^d$ $-\cos\theta(f_2(x)-e)+\sin\theta\times f_1(x)\leqslant 0$ $0\leqslant x_1\leqslant 1,\ -5\leqslant x_i\leqslant 5,\ i=2,3,4,5$ $a=40,\ b=5,\ c=1,\ d=6,\ e=0,\ \theta=-0.05\pi$

一般将其定义为历代非支配解集或几种不同算法并集的非支配解[16-18]. 在实验中, 由于真实Pareto最优解未知, 对于每个函数, 首先用不同算法各自独立运行15次, 然后合并15次计算的最优解集的Pareto滤集作为此函数的真实Pareto最优解集, 以减少算法的随机性的影响. 为进一步保证参照集的质量, 再将3种算法所得的Pareto最优解集之并集的Pareto滤集作为每个函数最终的真实Pareto最优解集.

其他参数设置如下: F_1, F_2, F_3, F_4, F_5 的最大迭代次数分别为 $60, 40, 40, 20, 200$; 缩放因子均取为 $F \in \mathrm{rand}(0.3, 0.8)$, 交叉概率均为 $C_R \in \mathrm{rand}(0.2, 0.6)$.

表2为3种不同算法在解集的收敛性、分布性、均衡性和解的个数上这4个指标的优化性能对比. 其中: 方法I为本文提出的分阶段二次变异改进DE法; 方法II为全程混沌变异的改进DE法; 方法III为标准DE.

表 2　3种不同算法的实验对比结果

平均值		F_1	F_2	F_3	F_4	F_5
解的个数	方法I	20.67	93.60	35.67	56.6	73.7
	方法II	3.00	89.33	5.00	46.40	16.93
	方法III	2.60	35.67	5.17	17.07	15.04
G_D	方法I	0.0045	0.0021	0.0048	0.0735	0.0001
	方法II	0.0086	0.0026	0.0103	0.0969	1.0902
	方法III	0.0107	0.0043	0.0102	0.2188	0.6394
S	方法I	0.0071	0.1479	0.0448	0.6536	0.0165
	方法II	0.0201	0.1528	0.1147	0.6980	0.4513
	方法III	0.0188	0.1550	0.0998	1.1125	0.1116
D_{1_R}	方法I	0.0071	0.1087	0.0251	0.4956	0.0969
	方法II	0.0183	0.1098	0.0574	0.4960	1.0952
	方法III	0.0162	0.1102	0.0501	0.8576	0.7155

从表2可知: 对于4个指标, 本文所采用的分阶段二次混沌变异方式所得的结果均为最佳, 但全程采用混沌局部寻优的改进DE算法与标准DE相比, 从解的个数上看, 基于非支配解的全程混沌局部寻优的改进DE比标准DE好很多, 验证了前述基于非支配解的二次变异能有效提高多目标优化问题进化过程中的Pareto非支配解数量的结论. 而对于另外3个指标, 基于非支配解的全程混沌局部寻优改进DE和标准DE的性能相当, 即相对于标准DE, 全程采用基于非支配解的混沌局部寻优改进DE并没有显著优势, 原因在于: 在DE算法进化全程选择具有代表性的非支配解, 并在这些优点的基础上引入混沌序列进行二次变异搜索所产生当前最优解的许多邻域点, 其种群多样性只是在局部区域内有所增加, 并不能有效扩大对整个解空间的搜索范围. 这也是本文采用在进化前期和后期分别实施基于非支配解的随机二次变异和混沌二次变异的分阶段二次变异方法的原因.

需要指出的是: 由于所提出的算法通过在标准DE上增加额外操作来改进其所求Pareto优解的质量, 在一定程度上影响了算法效率, 即所提出算法的计算量和运行时间相对于标准DE有所增加. 原因在于额外操作使算法的程序流程和程序代码比标准DE更加复杂, 用于个体评估与排序的计算量也增加了很多. 因此, 运行同样的迭代次数, 其计算所用的CPU时间相对长一些. 在实际测试中发现, 对相同测试函数运行相同的迭代次数, 本文算法的CPU运行时间基本上双倍于标准DE的CPU运行时间. 这也反映了任何一种算法都很难在所有的性能方面占尽优势.

6　结　论

本文将基于非支配解的二次变异技术结合到标准DE中, 以求解多目标约束优化问题, 并在DE进化前期使用基于非支配解的随机二次变异来加大DE的全局区域搜索能力, 在进化后期使用基于非支配解的混沌二次变异, 充分利用混沌的遍历性来提高DE的局部寻优能力. 仿真实验结果表明, 该方法对于低维多目标优化问题具有较好的寻优能力, 克服了标准DE算法易在求解多目标优化问题时非支配解数目过少、易陷入局部极值的缺点, 确保了算法在全局搜索性能与局部搜索性能之间的较好平衡, 而且保持了DE算法的简洁性能, 在收敛速度和精度上均优于标准DE.

参考文献(References)

[1] Leandro dos S C. Reliability-redundancy optimization by means of a chaotic differential evolution approach[J]. Chaos, Solitons and Fractals, 2009, 41(2): 594-602.

[2] Mahamed G H O, Ayed S. Constrained optimization using CODEQ[J]. Chaos, Solitons and Fractals, 2009, 42(2): 662-668.

[3] Leandro dos S C, Viviana C M. Chaotic artificial immune approach applied to economic dispatch of electric energy using thermal units[J]. Chaos, Solitons and Fractals, 2009, 40(5): 2376-2383.

[4] Yuan X H, Cao B, Yang B, et al. Hydrothermal scheduling using chaotic hybrid differential evolution[J]. Energy Conversion and Management, 2009, 49(12): 3627-3633.

[5] Wang Y J, Zhang J S. Global optimization by an improved differential evolutionary algorithm[J]. Applied Mathematics and Computation, 2007, 188(1): 669-680.

[6] 程志刚, 张立庆, 李小林, 等. 基于Tent映射的混沌混合粒子群优化算法[J]. 系统工程与电子技术, 2007, 29(1): 103-106.
(Cheng Z G, Zhang L Q, Li X L, et al. Chaotic hybrid particle swarm optimization algorithm based on Tent

map[J]. Systems Engineering and Electronics, 2007, 29(1): 103-106.)

[7] 牛大鹏, 王福利, 何大阔, 等. 多目标混沌差分进化算法[J]. 控制与决策, 2009, 24(3): 361-364.
(Niu D P, Wang F L, He D K, et al. Chaotic differential evolution for multiobjective optimization[J]. Control and Decision, 2009, 24(3): 361-364.)

[8] 张劲松, 李歧强, 王朝霞. 基于混沌搜索的混和粒子群优化算法[J]. 山东大学学报: 工学版, 2007, 37(1): 47-50.
(Zhang J S, Li Q Q, Wang Z X. Hybrid particle swarm optimization algorithm based on the chaos search[J]. J of Shandong University: Engineering Science, 2007, 37(1): 47-50.)

[9] 张浩, 张铁男, 沈继红, 等. Tent混沌粒子群算法及其在结构优化决策中的应用[J]. 控制与决策, 2008, 23(8): 857-862.
(Zhang H, Zhang T N, Shen J H, et al. Research on decision-makings of structure optimization based on improved Tent PSO[J]. Control and Decision, 2008, 23(8): 857-862.)

[10] Yang L, Chen T. Application of chaos in genetic algorithm[J]. Communications in Theoretical Physics, 2002, 38(2): 168-172.

[11] 王凤蕊, 王文宏. 解决位置管理问题的混沌混合差分进化算法[J]. 系统仿真学报, 2009, 21(6): 1609-1614.
(Wang F R, Wang W H. Hybrid chaotic differential evolution algorithm for location management problem of mobile computing[J]. J of System Simulation, 2009, 21(6): 1609-1614.)

[12] 郭振宇, 程博, 叶敏, 等. 一种并行混沌差异演化算法[J]. 西安交通大学学报, 2007, 41(3): 299-302.
(Guo Z Y, Cheng B, Ye M, et al. Parallel chaos differential evolution algorithm [J]. J of Xi'an Jiaotong University, 2007, 41(3): 299-302.)

[13] 俞国燕, 李鹏, 何真, 等. 一种用于多目标约束优化的改进进化算法[J]. 计算机集成制造系统, 2009, 15(6): 1172-1179.
(Yu G Y, Li P, He Z, et al. Advanced evolutionary algorithm used in multi-objective constrained optimization problem[J]. Computer Integrated Manufacturing System, 2009, 15(6): 1172-1179.)

[14] Tan K C, Goh C K, Mamun A A, et al. An evolutionary artificial immune system for multi-objective optimization[J]. European J of Operational Research, 2008, 187(2): 371-392.

[15] Knowles J D, Corne D W. On metrics for comparing non-dominated sets[C]. Proc of the Int Congress on Evolutionary Computation Conf(CEC02). New York: IEEE Press, 2002: 711-716.

[16] 郑金华. 多目标进化算法及其应用[M]. 北京: 科学出版社, 2007: 63-75.
(Zheng J H. Multiobjective Evolutionary Algorithms and their Applications[M]. Beijing: Science Press, 2007: 63-75.)

[17] 孟红云, 张小华, 刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. 计算机学报, 2008, 31(2): 228-235.
(Meng H Y, Zhang X H, Liu S Y. A differential evolution based on double populations for constrained multi-objective optimization problem[J]. Chinese J of Computers, 2008, 31(2): 228-235.)

[18] 谷峰, 陈华平, 卢冰原. 基于均匀设计的多目标遗传算法在柔性工作车间调度中的应用[J]. 系统工程理论方法应用, 2006, 15(6): 548-551.
(Gu F, Chen H P, Lu B Y. Multi-objective genetic algorithm based on uniform design for flexible job shop scheduling[J]. Systems Engineering-Theory Methodology Applications, 2006, 15(6): 548-551.)

(上接第456页)

[7] 丁锋, 萧德云. 多变量系统状态空间模型的递阶辨识[J]. 控制与决策, 2005, 20(8): 848-853.
(Ding F, Xiao D Y. Hierarchical identification of state space models for multivariable systems[J]. Control and Decision, 2005, 20(8): 848-853.)

[8] Ding F, Chen T. Hierarchical least squares identification methods for multivariable systems[J]. IEEE Trans on Automatic Control, 2005, 50(3): 397-402.

[9] Ding F, Chen T. Hierarchical identification of lifted state-space models for general dual-rate systems[J]. IEEE Trans on Circuits and Systems — I: Regular Papers, 2005, 52(6): 1179-1187.

[10] Liu X G, Lu J. Least squares based iterative identification for a class of multirate systems[J]. Automatica, 2010, 46(3): 549-554.