

基于锦标赛选择变异策略的改进差分进化算法及函数优化

傅嗣鹏 乔俊飞 韩红桂

(北京工业大学电子信息与控制工程学院 北京 100124)

摘要 针对差分进化算法传统变异策略在全局收敛鲁棒性和搜索效率上不能达到一个很好的折衷,并且算法的操作算子固定,导致搜索效率低、易早熟收敛等问题,文中在差分进化算法变异策略性能分析的基础上,提出了一种基于锦标赛选择的变异策略。该策略采用“锦标赛选择”对随机选取的变异向量排序选出基向量,差分向量选择有利于搜索的方向并对其“强化”,以提高收敛速率和维持种群多样性;同时操作算子采用随机正态缩放因子 F 和时变交叉概率因子 CR ,以平衡局部搜索和全局搜索;最后,利用 4 个典型 Benchmarks 测试函数对改进算法进行测试。实验结果表明,该改进型差分进化算法能有效避免早熟收敛,较好地提高算法的全局收敛能力和搜索效率。

关键词 差分进化算法,锦标赛选择,变异策略,可变操作算子

中图分类号 TP18 文献标识码 A

Improved Differential Evolution Algorithm Based on Mutation Strategy of Tournament Selection for Function Optimization

FU Si-peng QIAO Jun-fei HAN Hong-gui

(College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China)

Abstract The traditional mutation strategy of differential evolution algorithm can not reach a good tradeoff between robustness in global convergence and the search efficiency. The operators are constants, and the differential evolution algorithm leads to many problems, such as the low search efficiency and the premature convergence. Based on analysis of performance of the mutation strategies, a new mutation strategy with tournament selection rule taking the best individual vector from the random individual vectors as the base vector was proposed in this paper. Meanwhile, selecting the direction for the difference vector beneficial to search and making strengthen on the difference vectors is to improve the convergence rate and maintain the diversity of population. The random normal scaling factor F and the time-varying crossover probability factor CR are used synchronously to advance the local search and global search. Finally, the improved differential evolution algorithm was tested on four benchmark functions. The simulation results show that the improved algorithm can effectively avoid the premature convergence, as well as improve the global convergence ability and the search efficiency remarkably.

Keywords Differential evolution algorithm, Tournament selection, Mutation strategy, Variable operator

1 引言

差分进化(DE)算法^[1]是 Storn 和 Price 于 1995 年为求解 Chebyshev 多项式而提出的一种智能优化算法。DE 算法具有以下优点:(1)有能力处理非线性、不可微、多峰函数;(2)能实现并行计算增强型(intensive)复杂函数;(3)算法要易于使用,即算法所需控制参数少,且控制参量要易于选择,鲁棒性好;(4)良好的收敛特性,即对于任意给定的独立试验解总能收敛到全局最优值。因此,其在许多优化问题中都表现出了优于其他智能优化算法的性能,尤其是在约束优化计算、聚类优化计算、非线性优化控制、神经网络训练、滤波器设计等方面

得到了广泛应用^[2]。

作为一种基于种群的寻优方法,与其他进化算法(如遗传算法)一样,DE 算法也存在早熟收敛、易陷入局部最优的问题。为此,许多学者做了大量工作。比较典型的有:文献[3]提出一种基于种群适应度方差自适应二次变异的差分算法(ASMDE),其较好地解决了早熟收敛问题;文献[4]提出一种带局部搜索动态多群体的自适应进化算法(DMSDELS),它具有较强的跳出局部最优的能力。以上两种观点都是从不同角度利用种群分布情况分析改进算法,一定程度上提高了 DE 算法的搜索效率,但是这些方法都需要进行种群多样性评估,在种群信息上的研究直接导致了算法对适应度函数计算量的

本文受国家 863 计划资助项目(2007AA04Z160, 2009AA04Z155),国家自然科学基金资助项目(60873034, 60674066),北京市自然科学基金资助项目(4092010)资助。

傅嗣鹏(1987—),男,硕士生,主要研究方向为计算智能与智能优化控制等, E-mail: fusp1987@163.com; 乔俊飞(1968—),男,教授,博士生导师,主要研究方向为复杂过程建模与控制、计算智能与智能优化控制等; 韩红桂(1983—),男,副教授,硕士生导师,主要研究方向为智能计算及污水处理过程的建模、控制与优化等。

相对增加。另外, Dorronsoro 等人^[5]提出了一种紧凑型差分进化算法(CDE),该改进算法在收敛速率和避免早熟收敛两方面取得了比较好的结果,然而对有些测试函数也不能成功地跳出局部最优,如 Shifted Rastrigin 函数。Ali 等人^[6]在 DE 算法中加入了惩罚函数思想,较好地解决了具有约束条件的优化问题。然而,上述文献大都没有分析变异策略对算法性能的影响,算法的操作算子也多采用固定或者线性改变的方式,不能有效地平衡全局搜索和局部搜索。

针对 DE 上述存在的问题,文中提出一种改进 DE 算法,记作 TSIDE。TSIDE 算法采用基于锦标赛选择的变异策略,以提高收敛速度和维持种群多样性。同时,操作算子采用随机正态缩放因子 F 和时变交叉概率因子 CR ,以平衡局部搜索和全局搜索。通过几个典型的 Benchmarks 测试函数对改进算法进行了测试,结果表明,文中提出的改进算法能有效避免早熟收敛,全局收敛能力和搜索效率也得到显著提高。

2 DE 算法^[7]

DE 算法作为一种进化计算技术,是一种基于实数编码的全局优化方法,其实质是具有择优思想的贪婪遗传算法^[8]。具体算法流程如下。

2.1 初始化种群

以下标记方式表示一个初始种群:

$$\{x_{i,0} | x_{ij}^l \leq x_{ij} \leq x_{ij}^u\}$$

$$i=1,2,\dots, NP; j=1,2,\dots, D$$

按式(1)初始化:

$$x_{i,0} = x_{ij}^l + rand(0,1) \cdot (x_{ij}^u - x_{ij}^l) \quad (1)$$

式中, $x_{i,0}$ 表示初始种群中的第 i 条“染色体”(或个体), $x_{i,0}$ 表示第 0 代的第 i 条“染色体”的第 j 个“基因”。 NP 表示种群大小, D 表示问题的维数, $rand(0,1)$ 表示在 $(0,1)$ 区间均匀分布的随机数。

2.2 变异操作

在 DE 中常用的差分策略为随机选取种群中两个不同个体,将其向量差缩放后与待变异个体(基向量)合并,生成变异个体。即

$$v_{i,G} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (2)$$

$$i \neq r_1 \neq r_2 \neq r_3$$

式中, r_1, r_2, r_3 为互不相同的属于 $[1, NP]$ 区间上的随机整数; F 为 $[0,2]$ 区间上的固定常数,称为缩放因子,控制偏差向量的放大作用; $x_{i,G}$ 表示第 G 代种群中的第 i 个个体, $x_{r_1,G}$ 为待扰动基向量。

2.3 交叉(重组)操作

为了维持扰动向量的多样性,引入了交叉操作。交叉模式包括二项式交叉模式和指数交叉模式,这里只引入二项式交叉模式。生成试验向量:

$$u_{i,G} = \begin{cases} v_{i,G}, & \text{if } (randb(j) \leq CR \text{ or } j = rnbr(i)) \\ x_{i,G}, & \text{if } (randb(j) > CR \text{ and } j \neq rnbr(i)) \end{cases} \quad (3)$$

式中, CR 为 $[0,1]$ 区间上的固定常数,称为交叉概率因子,作用是控制当前种群每个个体中有多少参量被替换; $randb(j)$ 为 $[0,1]$ 之间的均匀分布随机数; $rnbr(i)$ 为 $\{1, 2, \dots, D\}$ 中的一个随机数,确保 $u_{i,G}$ 中至少有一位可以由 $v_{i,G}$ 贡献。

2.4 选择操作

为了维持种群数为固定常值,采用了“贪婪”选择方式来

决定是试验向量 $u_{i,G}$ 或目标向量 $x_{i,G}$ 进入下一代群体,描述为:

$$x_{i,G+1} = \begin{cases} u_{i,G}, & \text{if } f(u_{i,G}) \leq f(x_{i,G}) \\ x_{i,G}, & \text{if } f(u_{i,G}) > f(x_{i,G}) \end{cases} \quad (4)$$

式中, $f(x)$ 为所要优化的目标函数。

上述方案并不是唯一被证明可用的差分版本。为了区分不同的 DE 变种,一般可以表示为: DE/X/Y/Z。其中, DE 代表差分进化算法; X 表示待扰动的基向量,通常表示为“rand”(当前种群随机选取的一个向量个体)或“best”(当前种群中适应度最高的向量个体)的形式; Y 表示差分向量个数; Z 指代所采取的交叉方案——二项式交叉方案和指数交叉方案,分别用 exp(exponential) 和 bin(binomial) 表示。那么,前面描述的基本 DE 算法可描述为 DE/rand/1/bin 的形式。其他 DE 变种可参照文献^[9]。

通常,不同的变异策略具有不同的搜索性能。实际上,任何其他向量线性组合方式也可用作一种变异策略^[2]。同时文献^[10,11]都指出没有单一的变异策略适合所有的优化问题。其中, DE/rand/1/bin 和 DE/best/2/bin 是目前使用最广、应用最为成功的差分策略。前一种策略有利于保持群体多样性,后一种策略更强调算法的收敛速度^[12]。

3 DE 算法的改进

3.1 DE 变异策略改进

为了更深入地了解各种变异策略下 DE 算法的性能,文中将最常用的 DE/rand/1/bin 和 DE/best/2/bin 两种变异策略应用于 30 维的 Rastrigin 函数。其中 Rastrigin 函数为多峰函数,在 $S = \{x_i \in (-5.12, 5.12), i = 1, 2, \dots, n\}$ 内约有 10 n 个局部极小点。将搜索成功率、迭代次数、收敛时间列于表 1 (表中数据均来自文献^[8]), DE 参数设置及函数具体描述见文中第 4 节。

表 1 DE 两种变异策略对 Rastrigin 函数的测试结果对比

DE	变异策略	搜索成功率	迭代步数	收敛时间
1	DE/rand/1/bin	100%	6170	71.9
2	DE/best/2/bin	80%	5970	68.3

从表中可以看到两种变异策略具有不同的搜索性能。DE/rand/1/bin 具有更好的搜索成功率,说明此策略全局收敛鲁棒性好;而 DE/best/2/bin 求解迭代步数及收敛时间更少,可以看出这种策略搜索效率更高;以上两种策略分别采用随机选取的向量个体和当前种群最优个体作为基向量。一般来说,前者能更好地引导算法遍历整个种群,提高算法的“探索能力”,使算法不易陷入局部最优,但同时也使得局部搜索能力不足,表现为收敛所需迭代步数和收敛时间偏大;而后者利用最优个体引领算法前进,使搜索更有目的性和方向性地寻优,增强了算法的“开采能力”,使算法更快地收敛,但同时也大大增加了算法陷入局部最优的可能性,表现为搜索的成功率不高。由此可以看出,如果对两种策略的长处加以利用,取长补短,算法的性能便会得到较大改善,即在保证算法良好收敛性的同时,加快算法收敛速度。

基于以上分析,文中提出一种基于“锦标赛选择”的复合变异策略,即从当前种群中随机选取 3 个个体向量 $x_{r_1,G}, x_{r_2,G}, x_{r_3,G} (r_1 \neq r_2 \neq r_3 \neq i)$, 按照锦标赛选择规则比较适应度值,按照适应度值大小分别记作 $x_{br,G}, x_{\sigma,G}, x_{wr,G}$, 则有

$f(x_{br,G}) \leq f(x_{br,G}) \leq f(x_{br,G})$ ($f(x)$ 为待优化的函数); 选取 $x_{br,G}$ 作为变异基向量, 差分向量采用 $x_{br,G} - x_{i,G}, x_{br,G} - x_{wr,G}$ 作为变异扰动项引入。具体形式如下:

$$v_{i,G} = x_{br,G} + F \cdot ((x_{br,G} - x_{i,G}) + (x_{br,G} - x_{wr,G})) \quad (5)$$

根据式(5), 采用 $x_{br,G}$ 作为基向量, 使得每个待变异个体可以探索 $x_{br,G}$ 周围区域, 在保持了算法探索能力的同时加速了算法的收敛速率。而差分 $x_{br,G} - x_{i,G}, x_{br,G} - x_{wr,G}$ 矢量运算方式, 使得搜索选择适应度更优的方向, 更快地找到全局最优解, 同时四向量差分方式可以使变异后的个体在算法迭代后期尽可能保持大的差异, 保持种群多样性。改进后的变异策略不再只是一种完全的随机搜索, 而是在差分向量更优方向上的目的性的搜索, 即加强了算法的局部搜索能力。由于变异的个体向量都是随机的, 因此这种局部特性是在随机特性基础上的局部效应, 可以保证算法的全局搜索特性。

3.2 操作算子改进

DE 算法的控制参数有以下 3 个: NP, F, CR , 而 DE 搜索性能取决于算法全局搜索和局部开发能力的平衡, 这很大程度上依赖于算法的 3 个控制参数选取^[8]。这里只针对操作算子 F, CR 进行研究。

3.2.1 随机正态缩放因子思想

对于缩放因子 F , 文献[13]采取正态分布, 仿真结果表明具有正态分布 F 值的 DE 算法一般要比标准的 DE 有效, Zaharie 也证明在这种情形下 DE 会以概率收敛到问题的全局最优解^[14], 但是这样分布的 F 并不会从本质上增加 DE 的求解性能。文献[15]中将 F 修改为 $F = a + b \cdot rand(0, 1)$, $rand(0, 1)$ 是 $[0, 1]$ 之间的随机数, 修改的 DE 在很多实际问题求解中比标准 DE 更加有效和鲁棒。基于以上文章分析, 文中对 F 做了如下调整:

$$\begin{cases} F(G) = \mu + \sigma \times N(0, 1) \\ \mu = F_{\min} + (F_{\max} - F_{\min}) \times rand(0, 1) \end{cases} \quad (6)$$

式中, $N(0, 1)$ 为标准正态分布, σ 为人为给定的缩放因子随机方差数, 按经验取为 0.2; $rand(0, 1)$ 是 $[0, 1]$ 之间的随机数。 F_{\min}, F_{\max} 为缩放因子 F 的最小最大边界, 按照文献[3]推荐的 F 取值范围, 文中设定两者值分别为 0.4, 1; $F(G)$ 为第 G 代的缩放因子。

式(6)综合了正态分布和随机分布缩放因子 F 的特点。可以预见, 此方式也能在一定程度上改善 DE 的性能。究其原因: 一方面因为 DE 算法本身和解决的问题多为非线性, 可以克服基本 DE 算法固定 F 值或线性改变 F 值带来的不足; 另一方面可以使得差分向量振幅随机变化, 有利于维持种群多样性, 同“强化”的差分向量一起降低因采用适应度高的个体作为变异基向量而导致算法陷入局部最优的可能。

3.2.2 时变交叉概率因子思想

通过对式(3)分析, 知道 CR 越大, $v_{i,G}$ 对 $u_{i,G}$ 的贡献越多, 当 $CR=1$ 时, $u_{i,G} = v_{i,G}$, 有利于局部搜索和加速收敛。 CR 越小, $x_{i,G}$ 对 $u_{i,G}$ 的贡献越多, 当 $CR=0$ 时, $u_{i,G} = x_{i,G}$, 有利于保持种群多样性和全局搜索。按文献[3]所述, 良好的搜索策略应该是在搜索的初始阶段保持种群多样性, 进行全局搜索, 而在搜索的后期应加强局部搜索能力以提高算法的精度。基于这种思想, 文中提出一种时变交叉概率因子 CR 的方法, 即 CR 的值随着迭代次数的增加由小非线性变大, 非线性函数采用对称型 sigmoid 函数。这样能够克服操作算子 CR 固定

或者线性改变 CR 所带来的不足; 另外, 使得算法初始阶段 $x_{i,G}$ 对 $u_{i,G}$ 贡献多, 维持种群多样性, 提高算法的全局搜索能力, 后期 $v_{i,G}$ 对 $u_{i,G}$ 贡献多, 可以提高局部搜索能力, 加快收敛速度。具体形式如下:

$$CR(G) = CR_{\min} + (CR_{\max} - CR_{\min}) \times \frac{1 - e^{-\beta G / G_{\max}}}{1 + e^{-\beta G / G_{\max}}} \quad (7)$$

式中, $CR(G)$ 是第 G 代种群交叉概率因子值; G 为当前迭代次数; G_{\max} 是设置的最大迭代次数; CR_{\min}, CR_{\max} 分别为 CR 的最小、最大值, 分别设为 0.3, 0.9, 这样 $CR \in [0.3, 0.9]$ 。文献[2]指出, CR 位于上述范围时, DE 算法性能会大幅提升; sigmoid 函数中的 β 值按经验取为 5。

3.3 TSIDE 算法收敛性分析

对于智能优化算法, Solis 等人提出了以概率 1 收敛于全局最优解的充分条件。为了便于分析, 下面给出其主要结论^[16]。

假设 1 设 $f(x)$ 是待优化函数, 则要求算法在进化过程父代和子代形成一个单调非递增序列, 即 $f_{i+1}^* \leq f_i^*, f_i^* = f(x_G)$ 表示第 G 代种群最优解, $x_G \in U$ 表示第 G 代种群序列, U 为变量可行域。

假设 2 对 U 的任意 Borel 子集 A 有 $V(A) > 0$, 则有 $\prod_{G=0}^{\infty} (1 - \mu_G(A)) = 0$ 成立, $V(A)$ 为子集 A 的 NP 维闭包, $\mu_G(A)$ 为相关采样策略得到 A 中部分元素的概率。

定理 1 设目标函数 f 为可测函数, 同时 U 为可测子集, $\{X_G, G \geq 0\}$ 为算法生成的种群序列, 并且同时满足假设 1 和假设 2, 则 $\lim_{G \rightarrow \infty} \{f(x^*) = f^* \cap x^* \in x_G\} = 1$ 成立, 算法全局收敛于最优解 f^* , 其中 x^* 为全局最优解的集合。

证明: 根据定理 1, 只要证明 TSIDE 满足上述两个假设, 即可证明算法是可收敛的。首先, TSIDE 算法依然采用标准的 DE 的“贪婪”选择策略, 如式(4)所示, 显然假设 1 成立。对于假设 2, TSIDE 算法中种群向量个体 $x_{i,G}$ 基于变异策略随机产生, 并且算法机制保证 $x_{i,G}$ 经过进化得到的 $x_{i,G+1}$ 依然在定义域内, 并且各个变异向量之间互不影响, 则由文献[17]结论, NP 个样本空间的并集必包含 $U, U \subseteq \bigcup_{i=1}^{NP} M_{i,G}$, 其中 $M_{i,G}$ 为个体 $x_{i,G}$ 的样本空间支撑集。那么, 对于给定的 U 的任意 Borel 子集 A , 使得 $V(A) > 0$, 即 A 与其导集 A' 并集为非空真子集, 有 $V(A) \in U$, 种群内则可以搜索到一个最小区域 Ω 闭包集合 A , 所以 $\prod_{G=0}^{\infty} (1 - \mu_G(A)) = 0$, 即假设 2 成立。

3.4 TSIDE 算法实现步骤

综合以上对 DE 的分析和改进, 得到的改进算法即 TSIDE 算法的流程如下:

Step1 初始化种群 NP , 维数 D , 缩放因子 F_{\min} 和 F_{\max} 的值、交叉概率因子 CR_{\min} 和 CR_{\max} 的值、最大迭代次数 G_{\max} 及搜索空间边界 $[x_1^l, x_1^r]$, 令 $G=1$, 按式(1)初始化当前种群的每一个个体。

Step2 计算当前种群每个个体适应度 Fitness, 求出最优适应度 Bestfitness。

Step3 判断最优适应度 Bestfitness 是否达到精度或者是否达到最大迭代步数, 若是则退出; 否则, 执行下一步。

Step4 从当前种群中随机选择 3 个不同的个体, 适应度值最大个体的作为基向量 $x_{br,G}$, 同时按式(6)求出缩放因子

F, 带入式(5)进行变异操作, 生成变异个体 $v_{i,G}$ 。

Step5 按式(3)进行交叉操作, 其中 CR 由式(7)确定, 生成试验个体 $u_{i,G+1}$ 。

Step6 按式(4)进行选择操作, 生成 $G+1$ 代种群。

Step7 $G = G+1$, 返回 Step2。

4 改进差分进化算法(TSIDE)性能测试

为了验证文中提出的 TSIDE 算法的性能, 下面通过 4 个 Benchmarks 函数进行测试。这 4 个测试函数分别为 Sphere 函数(f_1)、Rastrigin 函数(f_2)、Griewank 函数(f_3)和 Schafer 函数(f_4), 其中前者为单峰二次函数, 后 3 者为多峰函数, 在最优值附近有大量局部极小值, Rastrigin 函数和 Griewank 函数是典型的非线性多模态函数, 峰型成高低起伏不定跳跃性地出现, 很难优化查找到全局最优值。而 Schafer 函数在全局极小附近剧烈震荡, 一般算法难以得到全局最优解。各测试函数表达式如下:

$$f_1 = \sum_{i=1}^n x_i^2, [-100, 100]$$

$$f_2 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), [-60, 60]$$

$$f_3 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, [-5.12, 5.12]$$

$$f_4 = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}, [-100, 100]$$

为了比较的公平, 各函数参数设置参见文献[8], 仿真维数 $D=30$, 种群数 $NP=200$, 最大迭代步数 8000, 仿真精度 $VTR=10^{-6}$ 。表 2 中对比数据及相应算法参数设置仍然选自文献[8], 各个算法出处亦可参见文献[8]中参考文献, 这里不再赘述。另外, 文中 Schafer 函数因与文献[8] Schafer 函数数值相差 1, 故 f_4 栏数据为文献[8]中数据各自加 1 后所得。

表 2 TSIDE、ACPFDE、ASMDE、DE1、DE2 各运行 20 次求平均值

函数	算法	最优解	平均解	迭代步数	方差	成功次数
f_1	TSIDE	4.06×10^{-6}	8.27×10^{-6}	421	5.36×10^{-13}	20
	ACPFDE	6.22×10^{-6}	9.09×10^{-6}	365	8.85×10^{-13}	20
	ASMDE	6.47×10^{-6}	9.01×10^{-6}	1087	8.21×10^{-18}	20
	DE/rand/1	7.12×10^{-6}	9.18×10^{-6}	2178	5.14×10^{-13}	20
	DE/best/2	7.42×10^{-6}	9.02×10^{-6}	2363	4.80×10^{-13}	20
f_2	TSIDE	8.01×10^{-7}	7.95×10^{-7}	1087	4.46×10^{-13}	20
	ACPFDE	6.29×10^{-6}	9.20×10^{-6}	1927	7.46×10^{-13}	20
	ASMDE	3.92×10^{-3}	7.31×10^{-3}	8000	9.51×10^{-6}	0
	DE/rand/1	6.18×10^{-6}	8.92×10^{-6}	6170	1.19×10^{-12}	20
f_3	DE/best/2	7.58×10^{-6}	1.99×10^{-1}	5970	7.95×10^{-1}	19
	TSIDE	4.80×10^{-7}	7.86×10^{-7}	303	1.61×10^{-14}	20
	ACPFDE	7.45×10^{-6}	9.10×10^{-6}	297	4.45×10^{-13}	20
	ASMDE	6.85×10^{-6}	9.04×10^{-6}	1159	7.19×10^{-18}	20
	DE/rand/1	5.93×10^{-6}	3.78×10^{-4}	2170	2.73×10^{-6}	20
f_4	DE/best/2	7.86×10^{-6}	6.28×10^{-3}	5110	7.05×10^{-5}	10
	TSIDE	0	0	127	2.07×10^{-14}	20
	ACPFDE	0	0	809	7.90×10^{-12}	20
	ASMDE	0	1.00×10^{-3}	1522	8.93×10^{-6}	19
	DE/rand/1	0	1.01×10^{-2}	7615	4.95×10^{-5}	2
	DE/best/2	0	7.38×10^{-2}	5710	2.08×10^{-5}	5

通过分析表 2 中仿真数据, 对比采用两种传统变异策略的 DE1 和 DE2 算法, TSIDE 兼具了两者的优点, 并且无论在迭代步数、收敛精度上都要明显优于两者, 因此文中的改进方案在保证算法成功率的同时也大大提高了算法的搜索效率, 很好地解决了算法传统变异策略在全局收敛性与搜索效率上

不能很好折衷的矛盾。另外对比其他的两种改进方案, ASMDE 算法在 f_3 函数出现了早熟, 原因是在与 ACPFDE 算法比较上, TSIDE 算法只在函数 f_1 、 f_3 迭代步数上稍大于方, 但依然是有可比较性的; 而在其他算法性能上都要优于对方, 同时由方差可知, 文中提出的改进算法兼具了 DE 算法鲁棒性较好的特点。

从测试数据中, 随机抽取一组实验结果作图对比。为了便于观察, 取收敛值的对数为纵坐标, 横坐标为迭代步数。各函数优化图形如图 1—图 4 所示。

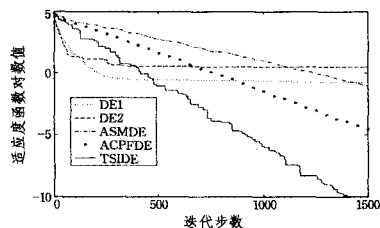


图 1 f_1 目标函数收敛曲线

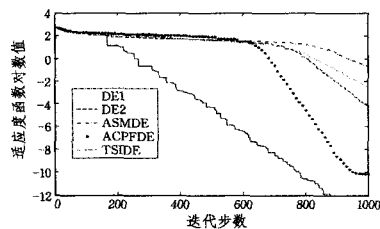


图 2 f_2 目标函数收敛曲线

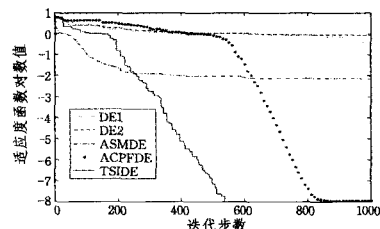


图 3 f_3 目标函数收敛曲线

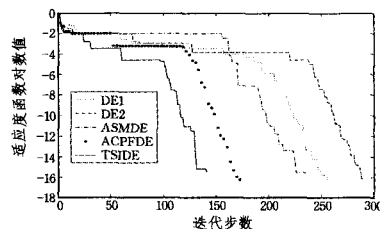


图 4 f_4 目标函数收敛曲线

结束语 文中在差分进化算法传统变异策略及操作算子对算法性能的影响分析的基础上, 提出了一种基于锦标赛选择变异策略的改进差分进化算法—TSIDE 算法, 该变异策略选择有利于搜索的方向并对差分向量“强化”, 以提高收敛速度和维持种群多样性。采用随机正态缩放因子 F 和时变交叉概率因子 CR , 平衡局部搜索和全局搜索。通过 4 个 Benchmarks 测试函数进行测试, 并与其他算法进行比较。实验结果表明 TSIDE 算法有很强的全局搜索能力, 能有效避免早熟收敛, 并且大大提高了搜索效率, 稳定性良好, 较好地解决了 DE 算法在全局收敛鲁棒性和搜索效率上不能达到一个很好的折衷的矛盾。

(下转第 36 页)

表 2 4 种算法的性能比较(100 次实验平均)

		SRPF	PSOPF	MPSOPF	MHMPSOPF
N=50	x_k	0.56482	0.32642	0.28945	0.17871
	\dot{x}_k	0.12579	0.047492	0.04716	0.044083
	y_k	3.5704	1.9323	1.6901	1.1159
	\dot{y}_k	0.097863	0.05239	0.047519	0.041283
N=100	x_k	0.15296	0.14995	0.10695	0.09983
	\dot{x}_k	0.006479	0.0031815	0.003169	0.002757
	y_k	0.25006	0.22665	0.1721	0.16086
	\dot{y}_k	0.005793	0.003877	0.003351	0.003231

结束语 本文在 MH 变异粒子群优化规则的基础上,利用 MH 变异算子来改进传统的全变异粒子群优化规则,进而提出了一种基于 MH 变异粒子群优化的粒子滤波重采样方法。该方法按照状态的后验概率密度产生新粒子,保证了重采样粒子的有效性,提高了滤波器的跟踪估计精度。仿真结果表明,在粒子数量较少的情况下,MHMPSOPF 算法的滤波性能明显优于 PF、PSOPF、MPSOPF 算法,只是少量增加了运算时间,为减少粒子滤波所需的粒子数量和减少粒子滤波的跟踪估计精度提供了一条有效途径。

参 考 文 献

- [1] Park S, Hwang J, Rou K, et al. A new particle filter inspired by biological evolution: genetic filter[J]. International Journal of Applied Science Engineering and Technology, 2007, 4(1): 459-463
- [2] 叶龙, 王京玲, 张勤. 遗传重采样粒子滤波器[J]. 自动化学报, 2007, 33(8): 885-887
- [3] 胡振涛, 潘泉, 梁彦, 等. 基于进化采样的粒子滤波算法[J]. 控制

理论与应用, 2009, 26(3): 269-273

- [4] 沈艳, 郭兵, 古天祥. 粒子群优化算法及其与遗传算法的比较[J]. 电子科技大学学报, 2005, 34(5): 696-699
- [5] Guofeng T, Zheng F, Xinhe X. A Particle Swarm Optimized Particle Filter for Nonlinear System State Estimation[C]// 2006 IEEE Congress on Evolutionary Computation. Vancouver, 2006: 438-442
- [6] 方正, 佟国锋, 徐心和. 粒子群优化粒子滤波方法[J]. 控制与决策, 2007, 22(3): 273-277
- [7] Kennedy J, Eberhart R C. Particle swarm optimization[C]// Proc IEEE international conference on Neural Networks. Piscataway: IEEE Service Center, 1995: 1941-1948
- [8] De Oca M A, Stutzle T, Birattari M, et al. Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 1120-1132
- [9] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报, 2004, 32(3): 416-420
- [10] Xu D, Li Y, Tang X D, et al. Adaptive Particle Swarm Optimization with Mutation[C]// Proceedings of the 30th Chinese Control Conference, 2012: 2044-2049
- [11] Zhang J L, Pan T S, Pan J S. A Parallel Hybrid Evolutionary Particle Filter for Nonlinear State Estimation[C]// IEEE 2011 First International Conference on Robot, Vision and Signal Processing, 2011: 308-312
- [12] Gordon N, Salmond D J, Smith A F M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation[J]. IEE Proceedings of Radar and Signal Processing, 1993, 140(2): 107-113

(上接第 18 页)

参 考 文 献

- [1] 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
- [2] Das S, Suganthan P N. Differential Evolution: A Survey of the State-of-the-Art[J]. IEEE Transactions on evolutionary computation, 2011, 15(1): 4-28
- [3] 吴亮红, 王耀南, 袁小芳, 等. 自适应二次变异差分进化算法[J]. 控制与决策, 2006, 21(8): 898-902
- [4] 张雪霞, 陈维荣, 戴朝华. 带局部搜索的动态多群体自适应差分进化算法及函数优化[J]. 电子学报, 2010, 38(8): 1825-1830
- [5] Dorronsoro B, Ferrante N, David N. Compact Differential Evolution[J]. IEEE Transactions On Evolutionary computation, 2011, 15(1): 67-98
- [6] Ali M M, Kajeer-Bagdadi Z. A local exploration-based differential evolution algorithm for constrained global optimization[J]. Applied Mathematics and Computation, 2009, 208(1): 31-48
- [7] 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): 721-727
- [8] 杨卫东, 姚峰, 张明. 基于自适应交叉概率因子的差分进化算法及其应用[J]. 信息与控制, 2010, 39(2): 187-193
- [9] Ernesto M, Price K, Lampine J. Differential Evolution—A Practical Approach to Global Optimization[M]. New York: Springer, 2005

- [10] Beyer H G, Deb K. On self-adapting features in real-parameter evolutionary algorithms[J]. IEEE Transactions on Evolutionary Computation, 2001, 5(3): 250-270
- [11] Brest J, Greiner S, Bo B, et al. Self adapting control parameters in differential evolution: A comparative study on numerical benchmark problems[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(6): 646-657
- [12] 杨启文, 蔡亮, 薛云灿. 差分进化算法综述[J]. 模式识别与人工智能, 2008, 21(4): 506-512
- [13] Qin A K, Suganthan P N. Self-adaptive differential evolution algorithm for numerical optimization [A] // Proceedings of the IEEE Congress on Evolutionary Computation [C]. Edinburgh, USA: Institute of Electrical and Electronics Engineers Computer Society, 2005: 1785-1791
- [14] Zaharie D. Critical values for the control parameters of differential evolution algorithms[A] // Eighth International MENDEL Conference on Soft Computing[C]. Brno, Czech Republic: Brno University of Technology, 2002: 62-67
- [15] Kim H K, Chong J K, Park K Y, et al. Differential evolution strategy for Constrained global optimization and application to practical engineering problems[J]. IEEE Transactions on Magnetics, 2007, 43(4): 1565-1568
- [16] 刘荣辉, 郑建国. 分区交叉差分进化算法及其约束优化[J]. 计算机科学, 2008, 39(2): 283-304
- [17] Dob K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2-4): 311-338