·人工智能及识别技术· 文章编号:1000-3428(2016)04-0168-05 文献标识码:A 中图分类号:TP301.6

基于跳跃基因的多目标差分进化算法

刘 峥,郑力明

(暨南大学电子工程系,广州 510000)

摘 要: 为解决现有多目标差分进化算法容易陷入局部最优的问题,引入跳跃基因算子,提出基于跳跃基因的多目标差分进化算法。改进现有的多目标差分算法,在传统交叉算子之后执行跳跃基因操作,以保持种群多样性。数值实验结果表明,该算法能较好地解决局部最优问题,在 ZDT 和 DTLZ 测试函数集上具有明显优于现有算法的收敛性能。

关键词: 跳跃基因;局部最优;困难问题;多目标差分进化算法

中文引用格式: 刘 峥,郑力明. 基于跳跃基因的多目标差分进化算法[J]. 计算机工程,2016,42(4):168-172.

英文引用格式: Liu Zheng,Zheng Liming. Multi-objective Differential Evolution Algorithm Based on Jumping Genes[J]. Computer Engineering,2016,42(4):168-172.

Multi-objective Differential Evolution Algorithm Based on Jumping Genes

LIU Zheng,ZHENG Liming

(Department of Electronic Engineering,Jinan University,Guangzhou 510000,China)

【Abstract】 In order to handle the local optimum problem of the existing multi-objective differential evolution algorithms,the jumping genes operation is introduced and a novel multi-objective differential evolution algorithm based on jumping genes is presented. Different from the existing algorithms,the proposed algorithm performs jumping genes operation after the classical crossover operation,in order to improve the population diversity. Numerical experimental results indicate that the proposed algorithm is capable of dealing with the local optimum problem and exhibits significantly better convergence performance than the existing algorithms on ZDT and DTLZ function tests.

【Key words】 jumping gene;local optimum;difficult problem;multi-objective differential evolution algorithm

DOI: 10.3969/j.issn.1000-3428.2016.04.030

1 概述

近年来,多目标进化算法(Multi-objective Evolutionary Algorithm,MOEA)及其在实际优化问题中的应用得到了日益广泛的研究。许多高效算法相继提出,如基于快速非支配排序的多目标遗传算法(Non-dominated Sorting based Genetic Algorithm,NSGA-II)[1]、基于分解的多目标进化(Multi-objective Evolutionary Algorithm based on Decomposition,MOEA/D)算法[2]、基于局部搜索的多目标优化(New Local Search-based multi-objective optimization algorithm,NSLS)算法[3]等。差分进化(Differential Evolution,DE)算法凭借简单高效的优势已成为最有竞争力的算法之一[4-5]。使用 DE 解决多目标问题也被大量研究[6-7]。在多目标差分算法中,较著名的算法有多目标优化的差分进化算法(Differential Evolution algorithm for Multi-objective Optimization,DEMO)[8]和多目标差分进化算法(Multi-objective Differential Evolution algorithm,MODEA)[9]等。然而,DEMO 使用的快速父代替换和 MODEA 使用的贪婪变异策略使其在求解复杂问题时存在明显不足,容易陷入局部最优。这是由于 DE 是利用个体间的差异寻优,使用贪婪的策略将使种群的多样性迅速下降。

跳跃基因现象(Jumping Genes Phenomenon)最初在玉米的基因中发现[10],提供了一种不同的基因传递方式:水平传输。跳跃基因现象发现染色体上的跳跃基因可以从一个位置跳跃到另一个新的位置。与差分算法传统的垂直传递交叉算子不同,跳

基金项目: 广东省对外科技合作基金资助项目(2013B051000060);广东省教育部产学研结合基金资助重点项目(2011A090200085);深圳市科技创新委员会基金资助项目(ZYC201105180515A)。

作者简介: 刘 峥(1990-),男,硕士研究生,主研方向为人工智能、多目标优化;郑力明,教授、博士。

收稿日期: 2015-09-21 **修回日期:** 2015-10-15 **E-mail:** zlmqit@126.com

跃基因的水平传输方式能产生更多潜在的基因序列,进而提高种群的多样性。文献[11]对此作了原理性的分析。文献[12]总结了跳跃基因现象并提出了 2 个跳跃基因算子,剪切粘贴和复制粘贴。文献[13]提出了跳跃基因改进的单目标差分算法(Jumping Gene Adaptation of Differential Evolution, DE-JGA)并将其应用于核反应堆网络设计的单目标优化实例。其实验结果表明 DE-JGa 在该实例上能够获得比基本 DE 更小的误差值。

为保持种群多样性,提高现有多目标差分进化算法跳出局部最优的能力,本文在传统交叉算子的基础上引入跳跃基因算子,提出基于跳跃基因的多目标差分进化算法。

2　算法基本框架和分析

2.1　差分进化算法

差分进化算法是基于种群的启发式搜索算法,具体流程描述如下:

算法 1　基本差分进化算法(DE)

步骤 1　种群初始化。在 D 维参数空间 $[\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]^D$ 中随机初始化种群 $\{\boldsymbol{x}_{i,0} = (x_{i,1,0}, x_{i,2,0}, \cdots, x_{i,D,0}), i = 1,2,\cdots,NP\}$,其中 NP 是种群的大小。设置代数计数器 $G = 0$。

步骤 2　变异。使用变异策略对目标向量 $\boldsymbol{x}_{i,G}$ 执行差分变异操作生成尝试向量 $\boldsymbol{v}_{i,G}$。常用的"DE/rand/1"变异策略定义如式(1)所示。

$$\boldsymbol{v}_{i,G} = \boldsymbol{x}_{r1^i,G} + F \times (\boldsymbol{x}_{r2^i,G} - \boldsymbol{x}_{r3^i,G}) \tag{1}$$

其中,$\boldsymbol{x}_{r1^i,G}$,$\boldsymbol{x}_{r2^i,G}$和 $\boldsymbol{x}_{r3^i,G}$是从当前种群中随机选取的 3 个个体;r_1^i,r_2^i,r_3^i 是介于[1,NP]的随机整数并且 $r_1^i \neq r_2^i \neq r_3^i \neq i$;$F \in (0,1]$是变异参数。

步骤 3　交叉。对变异之后的种群 $\boldsymbol{v}_{i,G}$执行差分交叉操作产生尝试向量种群 $\boldsymbol{u}_{i,G}$。常用的二进制交叉算子定义如下:

$$\boldsymbol{u}_{i,j,G} = \begin{cases} \boldsymbol{v}_{i,j,G} & \text{if } rand(0,1) < CR \text{ or } j = jrand \\ \boldsymbol{x}_{i,j,G} & \text{otherwise} \end{cases} \tag{2}$$

其中,$jrand$ 是[1,D]之间的随机整数;$CR \in [0,1]$是交叉参数。

步骤 4　选择。对尝试向量种群 $\boldsymbol{u}_{i,G}$进行评价,将其与目标向量种群 $\boldsymbol{x}_{i,G}$的个体进行比较,选择更优的个体进入下一代。选择算子如式(3)所示。

$$\boldsymbol{x}_{i,G+1} = \begin{cases} \boldsymbol{u}_{i,G} & \text{if } f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G}) \\ \boldsymbol{x}_{i,G} & \text{otherwise} \end{cases} \tag{3}$$

步骤 5　代数计数器 $G = G + 1$;如果满足终止条件,则停止算法,否则返回步骤 2。

2.2　多目标差分进化算法的基本框架

不同于基本差分进化(DE)算法,基本多目标差分进化算法(MODE)的选择算子使用的是文献[1]提出的快速非支配和拥挤度排序过程,其他算子与基本 DE 相同。MODE 的流程简述如下:

算法 2　基本多目标差分进化算法(MODE)

步骤 1　种群初始化,设置代数计数器 $G = 0$。

步骤 2　变异。

步骤 3　交叉。

步骤 4　非支配排序。对尝试向量种群 $\boldsymbol{u}_{i,G}$和目标向量种群 $\boldsymbol{x}_{i,G}$组成的混合种群进行快速非支配和拥挤度排序[1],选择最优的 NP 个向量进入下一代。

步骤 5　代数计数器 $G = G + 1$;如果满足终止条件,则停止算法,否则返回步骤 2。

2.3　跳跃基因算子

跳跃基因现象表示生物实验观察发现染色体上的跳跃基因以 2 种方式移动[10]:剪切粘贴和复制粘贴。剪切粘贴是指剪下染色体某段基因粘贴到新的位置,复制粘贴是指拷贝某段基因覆盖另一位置的基因。

在跳跃基因算子(Jumping Genes operator, JG)[12]中,2 种基因跳跃操作,剪切粘贴和复制粘贴以同等概率 0.5 被执行的流程如下:

算法 3　跳跃基因算子(JG)

步骤 1　生成[0,1]之间的随机数 p。

步骤 2　如果 $p < 0.5$,执行剪切粘贴操作;否则,执行复制粘贴操作。

2.4　基于跳跃基因的多目标差分进化算法

传统的二进制交叉算子(式(2))属于垂直型的算子,即决策变量的维度只能垂直传递,相互不能越界。而在跳跃基因算子中,变量维度之间既可以垂直移动,也可以水平移动。理论分析表明跳跃基因算子的这种变量传递方式可以提高种群的多样性[11]。

现有的多目标差分进化算法,如 DEMO[8],MODEA[9]的交叉操作都沿用了基本 DE 算法的二进制算子(式(2)),加之使用了贪婪的变异策略,使得个体间的维度差异快速消失,种群趋于一致。而在优化具有复杂 Pareto 前端的优化问题(如 ZDT4[14],ZDT6[14]和 DTLZ3[15])时,种群多样性的消失容易使这些算法容易陷入局部最优。为解决该问题,本文提出基于跳跃基因的多目标差分算法(Jumping Gene based Multi-objective Differential Evolution,JGMODE)。JGMODE 算法在传统二进制交叉算子之后引入跳跃基因操作,以保持种群多样性,提高算法的勘探能力。JGMODE 的具体流程如下:

算法 4　基于跳跃基因的多目标差分进化算法

(JGMODE)

步骤 1 种群初始化,设置代数计数器 $G=0$。

步骤 2 变异。

步骤 3 交叉。

步骤 4 跳跃基因。对生成的尝试向量种群 $\boldsymbol{u}_{i,G}$ 执行跳跃基因操作,产生新的种群 $\boldsymbol{t}_{i,G}$。

步骤 5 非支配排序。对新种群 $\boldsymbol{t}_{i,G}$ 和目标向量种群 $\boldsymbol{x}_{i,G}$ 组成的混合种群进行快速非支配和拥挤度排序[1],选择最优的 NP 个向量进入下一代。

步骤 6 代数计数器 $G=G+1$;如果满足终止条件,则停止算法,否则返回步骤 2。

3 实验结果与分析

3.1 测试函数及性能评价指标

本文选取广泛使用的 ZDT[14] 和 DTLZ[15] 函数集作为测试函数,分别是两目标函数 ZDT1,ZDT2,ZDT3,ZDT4 和 ZDT6,三目标函数 DTLZ1,DTLZ2 和 DTLZ3。

性能评价指标使用常用的 IGD 指标[16] 定义如下:

$$IGD(PF,P)=\frac{\sum_{x\in PF}dis(x,P)}{|PF|}$$

其中,$dis(x,P)$ 表示真实 Pareto 前端 PF 中的个体 x 与算法获得的非支配解集 P 中个体的最小欧拉距离;IGD 指标可以同时评价算法的收敛性和多样性[3],显示算法的综合性能。

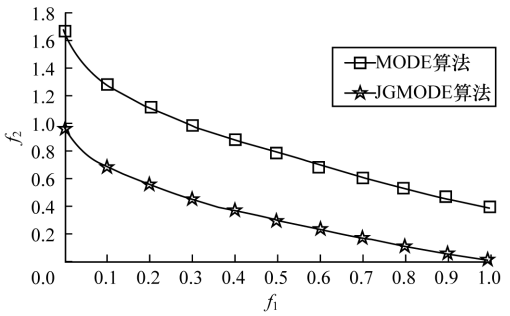
3.2 跳跃基因的有效性验证

ZDT 和 DTLZ 测试函数集中,ZDT4,ZDT6 和 DTLZ3 是较困难的函数:两目标函数 ZDT4 函数有 21^9 个局部最优 Pareto 前端[14],很容易误导算法的搜索过程使其陷入局部最优;ZDT6 函数的真实 Pareto 前端具有薄的解密度,并且是非均匀分布[9];三目标函数 DTLZ3 具有大量的局部最优解,也是较难优化的函数。本节通过比较 MODE 和 JGMODE 在这 3 个测试函数上的表现验证跳跃基因的有效性。

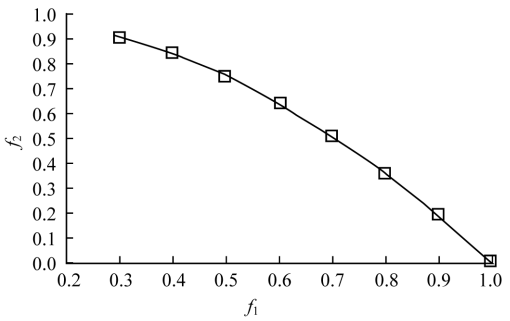
MODE 和 JGMODE 分别在 ZDT4,ZDT6 和 DTLZ3 上独立地运行 20 次,每次函数评估次数为 25 000(ZDT4,ZDT6)和 30 000(DTLZ3),得到的 IGD 均值和方差如表 1 所示。由表 1 可知,JGMODE 在这 3 个困难函数上获得了显著优于 MODE 的结果。图 1 显示了 MODE 和 JGMODE 在 3 个函数上的最大 IGD 值对应的 Pareto 前端,可以看出:对于 ZDT4 和 DTLZ3,JGMODE 每次运行都能收敛到真实的 Pareto 前端,而 MODE 不能很好收敛,容易陷入局部最优。对于 ZDT6,由图 1(b),JGMODE 获得了多样性更好的 Pareto 解集。

表 1 2 种算法性能比较

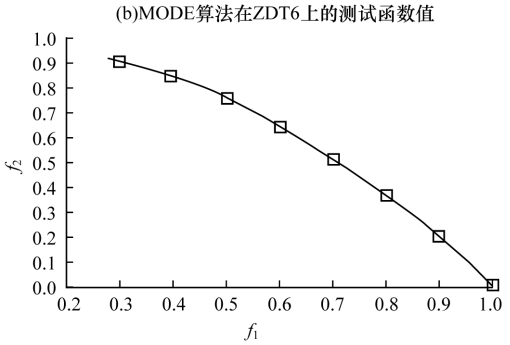
算法	ZDT4		ZDT6		DTLZ3	
	均值	方差	均值	方差	均值	方差
MODE	6.181E-2	4.358E-2	7.806E-3	4.245E-4	7.383E-1	6.087E-1
JGMODE	4.291E-3	1.433E-4	3.523E-3	1.591E-4	6.251E-2	8.326E-3



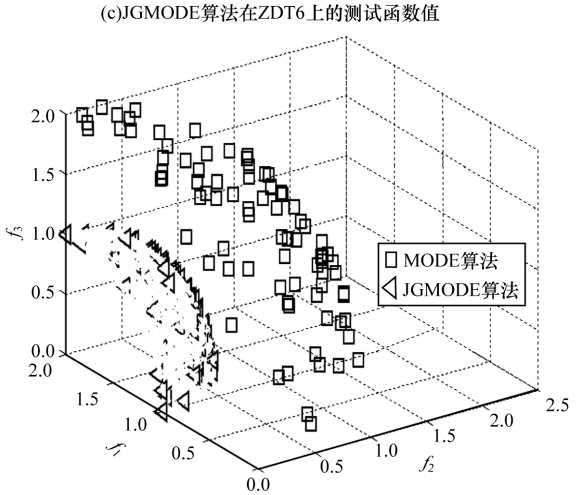
(a)ZET4测试函数值



(b)MODE算法在ZDT6上的测试函数值



(c)JGMODE算法在ZDT6上的测试函数值



(d)DTLZ3的测试函数值

图 1 最大 IGD 值对应的 Pareto 前端

图 2 显示了 20 次运行的 IGD 中位值对应的收敛曲线。对于 ZDT4 问题,相比 JGMODE,MODE 虽然在 120 代 ~ 170 代之间 IGD 值下降较快,但之后变化缓慢;对于 ZDT6 和 DTLZ3,JGMODE 在收敛速度和最终结果上都显著优于 MODE。

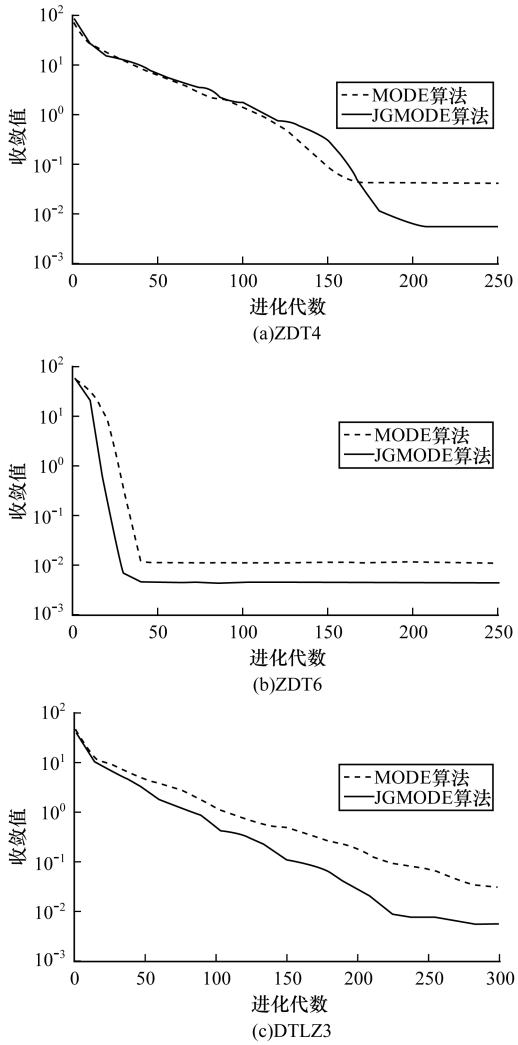


图 2　2 种算法 IGD 中位值对应的收敛曲线

3.3　与其他算法的比较

本节将 JGMODE 和其他高效的进化算法进行比较,包括:跳跃基因算子改进的遗传算法(Jumping Genes based Genetic Algorithm,JGGA),著名的多目标差分进化算法 DEMO 和 MODEA。算法的参数设置如下:

JGGA:种群大小 $NP = 100$,交叉概率 0.8,交叉指数 $\eta_c = 20$,变异指数 $\eta_m = 20$,变异概率 $p_m = 1/D$,基因跳跃概率 0.4,跳跃基因数 1,跳跃基因长度 3。

DEMO:种群大小 $NP = 100$,$F = 0.5$,$CR = 0.3$。

MODEA:种群大小 $NP = 100$,$F = 0.5$,$CR = 0.3$。

JGMODE:种群大小 $NP = 100$,$F = 0.5$,$CR = 0.3$,跳跃基因长度$\lceil 0.1D \rceil$。

函数评估次数:ZDT 问题 25 000,DTLZ 问题 30 000。

所有算法在 8 个测试函数上独立运行 20 次,得到的性能指标 IGD 均值如表 2 所示。由表 2 可知,JGMODE 在 8 个函数中的 5 个(ZDT3,ZDT4,ZDT6,DTLZ1 和 DTLZ3)展现了显著优于其他算法的性能。ZDT1 和 ZDT2 函数是 ZDT 函数集中较为简单的函数,在这 2 个问题上,MODEA 和 DEMO 分别获得了最优的结果;ZDT3 函数真实 Pareto 前端的不连续性增加了问题的求解难度,JGMODE 获得了最好的结果;对于 ZDT4,ZDT6,DTLZ1 和 DTLZ3 等具有复杂真实 Pareto 前端难以优化的问题,JGMODE 显示了显著优于其他多目标差分算法 DEMO 和 MODEA 的性能。图 3 所示为所有算法在 ZDT4 和 DTLZ3 函数上 20 次运行的 IGD 中位值对应的 Pareto 前端。由图 3 可以看出,DEMO 和 MODEA 算法获得的 Paerto 前端是距离真实 Pareto 前端较远的局部最优解集,而 JGMODE 算法能跳出局部最优,很好地收敛到真实前端。

表 2　4 种算法性能比较

函数	JGGA 算法		DEMO 算法		MODEA 算法		JGMODE 算法	
	均值	方差	均值	方差	均值	方差	均值	方差
ZDT1	4.798E-3	1.142E-4	4.564E-3	1.694E-4	4.457E-3	1.665E-4	4.493E-3	1.731E-4
ZDT2	5.215E-3	1.806E-4	4.403E-3	1.943E-4	4.528E-3	1.757E-4	4.479E-3	4.896E-4
ZDT3	1.510E-2	1.281E-2	5.277E-3	1.742E-4	5.125E-3	1.913E-4	5.011E-3	1.483E-4
ZDT4	4.929E-3	2.702E-4	3.682E-2	1.207E-1	4.639E-2	1.967E-1	4.291E-3	1.433E-4
ZDT6	4.929E-3	2.092E-4	7.598E-3	3.367E-4	7.859E-3	4.205E-4	3.523E-3	1.591E-4
DTLZ1	2.519E-2	8.928E-3	2.529E-2	8.693E-4	2.476E-2	7.314E-4	2.297E-2	6.273E-4
DTLZ2	7.643E-2	6.105E-3	6.223E-2	2.140E-3	6.172E-2	2.406E-3	6.402E-2	3.230E-3
DTLZ3	9.149E-2	6.764E-3	3.176E+0	3.956E+0	6.102E+0	4.764E+0	6.251E-2	8.326E-3

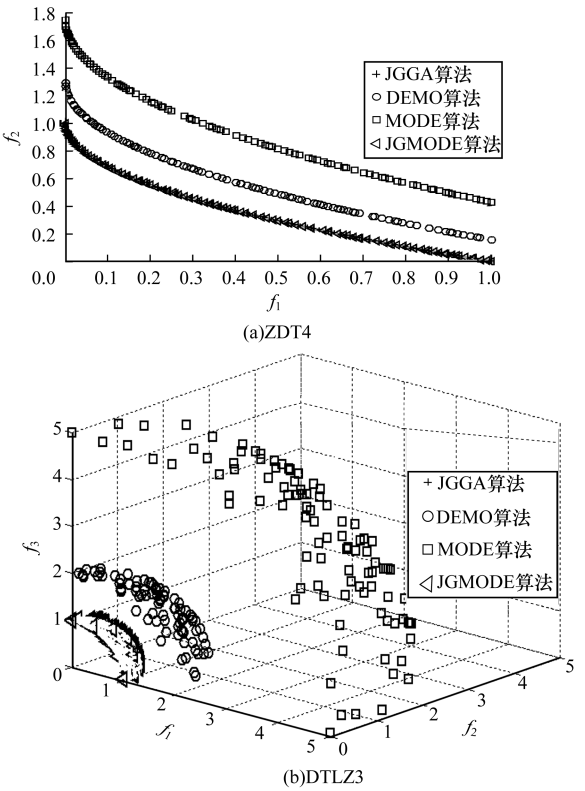


图 3 4 种算法 IGD 中位值对应的 Pareto 前端

4 结束语

现有的多目标差分进化算法使用的贪婪策略容易导致种群多样性迅速丢失,难以求解局部最优问题。本文引入跳跃基因操作来增强多样性,提出了基于跳跃基因的多目标差分进化算法。该算法使用跳跃基因算子弥补了差分算法传统二进制交叉算子的不足,增强了算法的探索能力。实验显示了跳跃基因算子的有效性,与其他算法的比较结果表明,提出的算法能更有效地优化具有复杂 Pareto 前端的测试问题,得到的最终解集更靠近真实最优解集。下一步的研究目标是将提出的算法应用于解决实际的复杂多目标优化问题。

参考文献

[1] Deb K, Agrawal S, Pratap A, et al. A Fast and Elitist Multi Objective Genetic Algorithm NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[2] Liu Hailin, Gu Fangqing, Zhang Qingfu. Decomposition of a Multiobjective Optimization Problem into a Number of Simple Multiobjective Subproblems[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 450-455.

[3] Chen Bili, Zeng Wenhua, Lin Yangbin, et al. A New Local Search-based Multiobjective Optimization Algorithm[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(1): 50-73.

[4] Das S, Suganthan P N. Differential Evolution——A Survey of the State-of-the-art[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.

[5] 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):506-513.

[6] 覃 晖,周建中,王光谦,等. 基于多目标差分进化算法的水库多目标防洪调度研究[J]. 水利学报,2009,40(5):513-519.

[7] 沈佳杰,江 红,王 肃. 基于多变异个体的多目标差分进化改进算法[J]. 计算机工程,2014,40(8):203-215.

[8] Robic T, Filipic B. DEMO: Differential Evolution for Multi-objective Optimization[C]//Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization. Washington D. C., USA: IEEE Press, 2005: 520-533.

[9] Ali M, Siarry P, Pant M. An Efficient Differential Evolution Based Algorithm for Solving Multi-objective Optimization Problems[J]. European Journal of Operational Research, 2012, 217(2): 404-416.

[10] McClintock B. The Origin and Behavior of Mutable Loci in Maize[J]. Proceedings of the National Academy of Sciences of the United States of America, 1950, 36(1): 344-355.

[11] Tang Kit-sang, Yin R J, Kwong S, et al. A Theoretical Development and Analysis of Jumping Gene Genetic Algorithm[J]. IEEE Transactions on Industrial Informatics, 2011, 7(3): 408-418.

[12] Chan T M, Man K F, Kwong S, et al. A Jumping Gene Paradigm for Evolutionary Multiobjective Optimization[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(2): 143-159.

[13] Gujarathi A M, Purohit S, Srikanth B. Optimization of Reactor Network Design Problem Using Jumping Gene Adaptation of Differential Evolution[J]. Journal of Physics: Conference Series, 2015, 622(1): 12-44.

[14] Zitzler E, Deb K, Thiele L. Comparison of Multi-objective Evolutionary Algorithms: Empirical Results[J]. Evolutionary Computation, 2000, 8(2): 173-195.

[15] Deb K, Thiele L, Laumanns M, et al. Scalable Multi-objective Optimization Test Problems[C]//Proceedings of Congress on Evolutionary Computation. Washington D. C., USA: IEEE Press, 2002: 825-830.

[16] Zitzler E, Thiele L, Laumanns M, et al. Performance Assessment of Multiobjective Optimizers: An Analysis and Review[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(2): 117-132.

编辑 顾逸斐