

1-3-2019

Multi-algorithm and Multi-population Co-optimization Differential Evolution Algorithm

Jinghua Zhang

Hebei Engineering Research Center of Simulation & Optimized Control for Power Generation, North China Electric Power University (Baoding), Baoding 71003, China;

Han Pu

Hebei Engineering Research Center of Simulation & Optimized Control for Power Generation, North China Electric Power University (Baoding), Baoding 71003, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Multi-algorithm and Multi-population Co-optimization Differential Evolution Algorithm

Abstract

Abstract: Algorithm fusion or co-evolutionary with multi populations are the solutions for complex engineering application. *A multi-algorithm and multi-population collaborative optimization algorithm is proposed by differential evolution (DE) algorithm, which pays emphasis on algorithm selection and combination. The algorithm designs a parameter-adaptive DE algorithm and selects three different DE algorithm variants which is complementary for each other and provides a multi-population co-optimization scheme according to four algorithms characters.* Stimulation results show that the proposed algorithm could make four different algorithms remedy for each other, gets a better result, and raises the precision, reliability and suitability, which reduces algorithm selection difficulty in engineering application.

Keywords

differential evolution algorithm (DE), co-optimization, multi-algorithm and multi-population, algorithm selection

Recommended Citation

Zhang Jinghua, Han Pu. Multi-algorithm and Multi-population Co-optimization Differential Evolution Algorithm[J]. Journal of System Simulation, 2018, 30(5): 1690-1699.

多算法多种群协同优化差分进化算法

张静华, 韩璞

(华北电力大学(保定)河北省发电过程仿真与优化控制重点实验室, 河北 保定 071003)

摘要: 研究算法融合、多种群协同进化是应用群智能算法求解复杂工程应用问题的一个方法。设计了一个基于差分进化算法的多算法多种群协同优化算法, 注重多算法的选择与组合。设计了一种自适应参数差分进化算法, 选择了3种各具特点的差分进化算法变体与其互补, 基于4种算法的特点设计了相应的多种群协同进化策略。对算法进行了仿真设计, 仿真结果表明该算法通过使4种不同特点的算法互补能得到较优结果, 并获得精度、可靠性与适用性的提升, 弥补工程应用中算法选择的困难。

关键词: 差分进化算法; 协同优化; 多算法多种群; 算法选择

中图分类号: TP18; TP391.9 文献标识码: A 文章编号: 1004-731X (2018) 05-1690-10

DOI: 10.16182/j.issn1004731x.joss.201805009

Multi-algorithm and Multi-population Co-optimization Differential Evolution Algorithm

Zhang Jinghua, Han Pu

(Hebei Engineering Research Center of Simulation & Optimized Control for Power Generation,
North China Electric Power University (Baoding), Baoding 71003, China)

Abstract: Algorithm fusion or co-evolutionary with multi populations are the solutions for complex engineering application. A multi-algorithm and multi-population collaborative optimization algorithm is proposed by differential evolution (DE) algorithm, which pays emphasis on algorithm selection and combination. The algorithm designs a parameter-adaptive DE algorithm and selects three different DE algorithm variants which is complementary for each other and provides a multi-population co-optimization scheme according to four algorithms characters. Stimulation results show that the proposed algorithm could make four different algorithms remedy for each other, gets a better result, and raises the precision, reliability and suitability, which reduces algorithm selection difficulty in engineering application.

Keywords: differential evolution algorithm (DE); co-optimization; multi-algorithm and multi-population; algorithm selection

引言

工程应用涉及到各种各样的优化问题, 且优化



收稿日期: 2016-08-05 修回日期: 2017-06-30;
作者简介: 张静华(1975-), 女, 河北定州, 博士生, 讲师, 研究方向为人工智能、优化算法; 韩璞(1959-2017), 男, 河北平泉, 本科, 教授, 博导, 研究方向智能控制, 火电站节能减排优化运行等。

问题越来越复杂, 计算智能方法常被用来研究解决这些复杂优化问题。当前计算智能算法的研究方兴未艾, 新的算法在不断出现。各种算法层出不穷, 能否产生一种超级优秀的算法, 比任何其他算法都好? 文献[1]提出了“没有免费午餐定理”, 认为没有一种算法能适用于所有问题。实践也说明尽管群体智能算法在不断创新改进, 但仍然不能用一个算法解决所有问题。探索和开发, 效率、精度以及可靠

<http://www.china-simulation.com>

• 1690 •

性都可能是算法矛盾竞争的两面。算法的参数选择或者动态变化, 都不能保证适用所有问题, 或者优于其他所有算法。在算法应用方面, 由于特定工程问题复杂, 特性无法清楚获知, 选择对问题合适的算法、合适的参数仍没有确定的方法。因此研究算法的融合或多种群协同优化方法来达到优势互补, 是当前应用计算智能算法求解优化问题的一个解决方案。

差分进化算法(DE)是 R.Storn 和 K.Price 于 1995 年提出的一种基于群体智能和进化计算的优化算法。算法参数少、原理简单、鲁棒性强, 在众多群智能算法中是很有竞争力的一种算法。DE 算法只有两个参数, 但某些问题会对参数很敏感, DE 算法的变异操作也有很多变体, 各有不同的搜索特点, 对不同的问题寻优效果也不同。本文研究基于 DE 算法的多算法多种群协同优化方法, 通过对 DE 算法的研究现状分析, 设计并选取合适的 DE 算法变体构成算法组, 并通过多种群多算法并行搜索并协同优化, 使不同算法的组合可以互相弥补缺点, 增加算法的鲁棒性, 使能适用更多问题。

1 差分进化算法

差分进化算法在种群的每代进化中涉及变异、交叉、选择 3 个步骤。

1.1 经典的 DE 算法及参数分析

变异步骤: 差分变异公式为

$$v_i = x_{r1} + F(x_{r2} - x_{r3})$$

式中: x_{r1} 、 x_{r2} 、 x_{r3} 为每代种群中随机选择的 3 个个体向量, 且 $r1 \neq r2 \neq r3 \neq i$ 。F 为变异步长, 得到 v_i 为针对目标个体 x_i 的变异向量。

交叉步骤: 有两种方法, 二项式交叉与指数交叉。二项式交叉是常用的方法, 其将当前的目标个体 x_i 与变异个体向量 v_i 进行交叉操作。交叉公式为

$$u_i = u_{i,j} = \begin{cases} v_{i,j} & \text{if } \text{rand}_j(0,1) \leq CR \text{ or } j = j_{\text{rand}} \\ x_{i,j} & \text{otherwise} \end{cases}$$

CR 为交叉概率, 得到 u_i 为候选向量。

选择步骤: 在目标向量 x_i 与候选向量 u_i 之间进行贪婪选择, 适应度好的保留。

经典差分进化算法有 3 个参数 NP、F、CR。NP 为种群的个体数, F、CR 是两个非常重要的参数, 取值范围一般为[0,1]。对于参数的选择一些研究给出了一些建议。R.Storn 和 K.Price 推荐 NP 值为 5D~10D (D 为个体向量维数), F 的初始选择为 0.5, 有效取值范围为[0.4,1]。文献[2]检查了不同的参数设置来优化 Sphere、Rosenbrock、Rastrigin 函数, 其实验结果表明优化过程的搜索能力和收敛速度对参数的选择非常敏感, 其推荐 NP 为 3D~8D, F 为 0.6, CR 为[0.3,0.9]。文献[3]认为 F 的范围为[0.4,0.95], 0.9 是一个好的初始选择, 并且建议当函数是可分离的问题时 CR 范围为[0,0.2], 当函数参数是互相依赖的, 则 CR 区间选取[0.9,1]。这些建议各不相同, 也说明选择合适的参数本身是一件困难的事情, 目标函数单模或多模问题, 参数可分离或不可分离, 复杂的空间特性都会对 CR、F 的参数选择产生影响。但是解决实际工程问题时, 问题的函数特征通常是不了解的, 因而预先选择合适的参数值就很困难, 一些动态变化的参数策略被提出。一些文献对 F 和(或)CR 使用各种形式的随机值, 如某范围内的均匀分布随机值^[4], 正态分布随机值^[5], 柯西分布随机值^[6]。除使用随机值外, 一些研究使用自适应策略改变 F 与 CR。文献[4,7]使 F 与(或)CR 随进化代数变化。文献[8-11]通过种群个体是否使选择成功来调整参数值。文献[12]通过种群个体的适应度来调整参数值。

针对 DE 算法, 一般的经验认知为^[13]: F 越小扰动量越小, 易于局部搜索, 但可能会陷入局部极值; F 较大则扰动量大, 利于全局探索, 维持多样性, 但可能会错过最优值; CR 越高, 在候选个体中就引入越多变种, 增加了多样性和探索性; 降低 CR 每次只引入少量变化, 会增加搜索鲁棒性。基于此认知, 随进化代数变化的参数策略, 如对 F 采用单调递减方式^[4,7], 其可以在算法初期保持多样性, 后期进入精细搜索。而 F、CR 取随机数的

方案^[4]，则在算法中同时兼顾探索性和开发性。也有研究认为有的问题对 CR 非常敏感，需要特定范围的 CR ^[3]，因此基于进化代数的线性自适应策略虽然很好的改进了算法对某些问题的精度和效率，但对有些问题仍无能为力。而自适应参数的研究随进化过程的状态来改变 F 、 CR ^[8-12]，对具体问题更有针对性，也使算法性能获得了较大提升，此类算法关键点是通过什么方式来判断进化状态以及依据所获得的进化状态对 F 、 CR 的更新机制，不同的研究各有不同的设计。

1.2 变异策略的变体

差分进化算法的变异策略有多种变体，其命名规则为 DE/x/y，x 为变异的基向量的选择方式，y 为差异向量的个数。常见的几种变异公式如表 1 所示。其中 x_i 是目标个体， x_{r1} 、 x_{r2} 、 x_{r3} 、 x_{r4} 、 x_{r5} 是当代种群中的随机个体，且各不相同，并与 x_i 不同。 x_{rb} 是当前代的最优个体。

表 1 差分进化算法变异公式
Tab. 1 Mutation formular of DE algorithm

序号	变异策略	公式
1	DE/rand/1(经典)	$v_i = x_{r1} + F(x_{r2} - x_{r3})$
2	DE/rand/2	$v_i = x_{r1} + F(x_{r2} + x_{r3} - x_{r4} - x_{r5})$
3	DE/best/1	$v_i = x_{rb} + F(x_{r2} - x_{r3})$
4	DE/best/2	$v_i = x_{rb} + F(x_{r2} + x_{r3} - x_{r4} - x_{r5})$
5	DE/rand-to-best/1	$v_i = x_{r1} + P*(x_{rb} - x_{r1}) + F(x_{r2} - x_{r3})$
6	DE/target-to-best/1	$v_i = x_i + P*(x_{rb} - x_i) + F(x_{r2} - x_{r3})$
7	DE/target-to-rand/1	$v_i = x_i + P*(x_{r1} - x_i) + F(x_{r2} - x_{r3})$

这 7 种变异策略的寻优特点各不相同。策略 1、2 的多样性较好，但是收敛较慢，而策略 2 与策略 1 相比多样性更好一些，但收敛性要差一些。变异策略 3、4、5 收敛非常快，但有时会陷入局部极值。策略 3 收敛最快，也更易陷入局部极值。策略 6、7 的独立寻优效果较差。策略 6 当 $CR=1$ 时，与粒子群算法思想有部分相似之处。策略 7 当 P 为随机值、 $F=0$ 、 $CR=0$ 时，是蜂群算法采用的变异策略。策略 6、7 当 $CR=1$ 时被认为是旋转不变策略^[14-15]。

由于变异策略各自不同的特点，一些研究探讨

变异策略的组合或自适应选择。文献[7-8]将变异策略组合为新的变异策略。文献[16]随机对个体分组并应用不同的变异策略。文献[5, 10]设置变异策略算法池，根据变异策略是否成功来自适应选择当前进化阶段最合适的变异策略。文献[17]根据种群个体适应度改善信息修改每个变异策略的选择概率，来自适应选择变异策略。

不同问题、甚或一个问题优化过程的不同阶段适合的参数、变异策略可能都不同。自适应算法的研究避免了选择的困难，也一定程度提高了算法的性能。但是自适应算法仍会遇到困难，问题的复杂性可能会使自适应过程产生偏差，而一旦产生偏差，偏离正确位置的参数或种群个体能否回到正确位置是对算法很大的考验。一些研究从另一途径入手，探讨多种群协作方法以弥补单一种群单一算法(自适应算法)的不足^[15]。相关研究探讨种群协作机制，如文化算法框架、memetic 框架、孤岛模型与种群迁移机制^[18]等。而本文则将多种算法以多种群方式结合，以算法的设计选择为重要环节，将不同特点的 DE 算法变体用多种群机制将其融合，协同优化。

2 基于差分进化算法的多算法多种群协同优化设计

不同的变异策略与参数组合会产生不同的搜索特性，在多算法多种群协同进化策略中，算法的设计与选取是重要一环。本文选择了经典的 DE/rand/1 变异策略，以及在测试中表现较优的 DE/rand-to-best/1 变异策略。交叉策略为二项式交叉。选择策略为贪婪选择。对于参数(F, CR)给出了 3 种方案：固定参数、随机参数、自适应参数。基于这些策略模块和参数方案可以构成具有不同特点的算法。种群群体个数参数 NP 本文不做特殊处理，根据具体问题复杂度、问题维数来选择 NP 。依据相关文献推荐的 NP 值，对于低维问题采用 10D~20D，高维问题采用 3D~8D。

2.1 差分进化算法的设计与选择

由于求解具体问题时参数选择困难,且有些问题是参数敏感的,因此一个自适应参数的差分进化算法是有必要的。本文首先设计了一个自适应参数的 DE 算法,并选择了 3 种不同的 DE 算法变体对其进行互补。

1) 自适应参数 DE/rand/1 差分进化算法

采用经典的 DE/rand/1 变异策略。对于参数 F 采用随机参数方法,取 $[0,1]$ 之间的随机数。而参数 CR 对于某些问题的敏感性很强,对 CR 进行自适应选择。

将对 CR 的自适应选择看做对 CR 的优化。 CR 的取值范围为 $[0,1]$,因为适合的 CR 值并不需要精确的实数值,故将 CR 取值范围设计为离散值集合 $\{0\ 0.1\ 0.2\ \dots\ 1\}$,11 个数字,相邻数字相差 0.1。其中 0、1 表示 CR 为 0 和 CR 为 1 的特殊交叉率, CR 为 0 表示每次只交叉一维, CR 为 1 表示不交叉,直接使用变异向量。

采用种群寻优机制,借助 DE 算法的种群个体同时作为 CR 优化的个体,每个个体有各自的 CR 值。对参数值优劣评价的适应函数是自适应参数策略的一个很重要问题。文献[2]曾指出从代到代接收的新个体多是收敛较好的表示,即使最优值减小缓慢。基于此思想,本文将选择操作的成功率作为参数优劣的评价函数 g 。

$$g(cr) = \text{countS}(cr) / \text{countA}(cr) \quad (1)$$

$$cr \in \{0\ 0.1\ 0.2\ \dots\ 1\}$$

$\text{countA}(cr)$ 为每代个体中应用 cr 个体数, $\text{countS}(cr)$ 为应用 cr 成功的个体数。

借鉴蚁群算法信息素的思想,为 CR 的离散值集合建立信息素表。用信息素指导 CR 值的选择,信息素高的选择概率大。并在每代优化结束时更新信息素。步骤如下:

(a) 算法初始,进入探测阶段,对取值范围内的每个离散值做一轮测试,通过公式(1)得每个值的初始成功率。用初始成功率作为信息素表的初始信息素。

(b) 每个个体单独选择 CR ,基于信息素,采用轮盘赌的方法,在离散取值范围内选择一个 cr 。将选则的 cr 作为正态分布函数的均值 μ ,求满足正态分布 $N(\mu, \sigma^2)$ 的随机值作为 CR 的取值, σ 本文取值为 0.1。

(c) 每一代进化完成后更新信息素。按最近原则将个体的 CR 值归到对应的 cr 。依据公式(1)求取值范围中每个 cr 的成功率。信息素更新涉及信息素挥发和信息素增强,方法公式为

$$\text{pher}(cr) = \text{pher}(cr) \times (1 - p) + g(cr)$$

其中: $\text{pher}(cr)$ 是 cr 的信息素; p 是信息素挥发率,本文采用 0.1。旧信息素以 p 速率挥发,每代增加新的信息素 $g(cr)$ 。

针对具体问题,一个参数在初期可能合适,在后期可能不再合适。同时参数优化过程也可能因为某些原因对信息素产生不正确的影响。因此算法需要能脱离当前不合适极值的能力。轮盘赌方法、正态分布随机值的方法,使信息素低的 CR 值也有机会选择。同时每隔若干代,参数采用 $[0,1]$ 均匀随机方式选择,给信息素低的值更多机会。若参数不再合适,会逐步调整。

此算法用来处理 CR 敏感的问题。

2) 随机参数 DE/rand/1 算法

此算法采用经典 DE 变异策略 DE/rand/1,参数采用随机参数方案。对 F 与 CR ,每个个体分别进行均匀分布的随机数采样。此算法,通过参数的多样性,兼顾寻优过程的开发与探索。

3) 固定参数 DE/rand-to-best/1 算法

此算法采用 DE/rand-to-best/1 变异策略,强调搜索效率,但容易陷入局部极值。选择固定参数方案,并采用(0.5,0.5)来平衡其快速收敛速度。

4) 组合变异策略算法

此算法同时采用 DE/rand/1 与 DE/rand-to-best/1 两种变异策略。在算法的迭代过程中交替使用,参数选择固定值(0.5, 0.5)。此算法,通过不同的变异策略组合来平衡探索与开发。

2.2 多算法多种群协同优化策略

本算法采用4个种群协同进化,分别采用本文设计选择的4种算法:

算法A: DE/rand/1, 自适应参数

算法B: DE/rand/1, 随机参数

算法C: DE/rand-to-best/1, 固定参数

算法D: DE/rand/1与DE/rand-to-best/1组合变异策略, 固定参数

算法B与算法D通过不同的方法来达到探索与开发的平衡。算法A适用于CR敏感的问题,算法C收敛速度很快,但可能会陷入局部极值。算法D与算法C相比增加了多样性,比C收敛慢但不易陷入局部极值。4种方法各有特点,多算法多种群并行搜索,通过种群间的交流协同进化,达到互相弥补缺点,共同进化的目的。

多种群协同优化需要解决如下两个问题:

- 避免算法D陷入局部最优值。
- 加快算法A、B、C的寻优进程。

基于各算法的特点,采取了如下3种种群间信息共享的方案:

(a) 共享最优值: 算法A, 算法B, 算法C的最优值加入到算法D, 替代D的最差值, 避免算法D过早进入局部极值, 当D陷入局部极值, 以致最优值差于其他3种方法, 此步用以将种群带出局部极值域。

(b) 种群间个体移动: 每隔50代, 拥有最优值的种群将其前5%的优势个体复制到其他种群替代其最差值, 帮助其他种群加快搜索进程。

(c) 跨种群变异: 每隔50代, (与步骤(b)隔开), 种群D个体分别与种群A、B、C个体跨种群变异。变异策略为DE/rand/2, 参数为(0.5,0.5)。此步骤将种群D向A、B、C种群偏移, 用来避免算法D陷入局部极值, 或将已经陷入局部极值的种群D拉扯出来。

3种方案共同作用, 使4个种群4种算法共同进化。

3 算法仿真测试

3.1 仿真方案设计

1) 基准测试函数

选择5个基准测试函数:

$$F1(x) = \sum_{i=1}^D x_i^2$$

$$F2(x) = \sum_{i=1}^d \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i^2)^2 \right)$$

$$F3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

$$F4(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$F5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20$$

F1是Sphere函数, 简单的单峰函数, 用来测试算法的寻优精度。F2是Rosenbrock函数, 由于函数为优化算法提供的信息比较有限, 使算法很难辨别搜索方向, 查找最优解十分困难。F3、F4、F5分别是Rastrigin、Griewank、Ackley函数, 是典型的非线性多峰函数, 用于检验算法的全局搜索能力。各函数的测试域与极值如表2所示。

表2 测试函数的域与极值

Tab. 2 Domain and extremum of the benchmark functions

函数	搜索空间	极值
F1	$[-100, 100]^D$	0
F2	$[-2.048, 2.048]^D$	0
F3	$[-5.12, 5.12]^D$	0
F4	$[-600, 600]^D$	0
F5	$[-30, 30]^D$	0

2) 测试算法集

本文测试选择6种算法, 重点比较多算法多种群协同优化算法比各算法单种群独立寻优带来的优势。算法1为经典DE算法作为测试基准, 参数为常用参数(0.5, 0.5)。然后对本文设计选择的4种算法进行单独测试: 算法2为DE/rand/1随机参数算法; 算法3为组合变异策略算法; 算法4为DE/rand/1自适应参数算法; 算法5为DE/rand-to-best/1固定参数算法。算法6为本文设计的多算法

多种群协同优化算法。

3) 测试方案

本文选用 matlab 软件对算法进行仿真测试, 对被测函数 50 维进行寻优。算法 1~5 种群个体数 200。算法 6 的 4 个子种群个体数均为 200。选择四种方案来测试算法的效果:

a) 算法进化代数确定(2 000 代), 每个算法对每个函数分别独立运行 20 次, 获得最优值、最差值、均值与标准差。

b) 通过最优值进化轨迹图, 表明算法的收敛过程。

c) 通过种群个体实时分布图, 在运行时实时观察个体随进化代数的变化。由于高维问题无法表示, 采用其中某 2 维数据进行观察。

d) 特殊测试: 设计一些特殊测试, 验证多算法多种群方法的优势。

3.2 测试结果分析

本测试及分析的目的在于验证通过多种群的方式将 4 种不同特点的算法互补, 能否得到较优结果, 并获得精度、可靠性与适用性的提升。

测试方案 a) 结果如表 3 所示。

表 3 各算法进化 2 000 代测试结果
Tab. 3 Experimental results of the selected 6 algorithms evolving 2 000 generations

函数	算法 1	算法 2	算法 3	算法 4	算法 5	算法 6	
F1	最优值	5.62e-09	7.44e-21	2.12e-58	1.77e-19	5.27e-104	2.16e-100
	最差值	1.57e-08	1.84e-19	1.15e-57	5.50e-17	3.66e-102	2.04e-98
	均值	7.84e-09	8.87e-20	5.31e-58	1.34e-17	7.76e-103	5.41e-99
	标准差	1.51e-09	1.21e-19	2.51e-58	1.51e-17	9.36e-103	6.50e-99
F2	最优值	4.23e+1	3.86e+1	3.32e+1	3.91e+1	4.49e+1	2.35e+1
	最差值	4.29e+1	4.16e+1	4.29e+1	4.51e+1	4.86e+1	2.60e+1
	均值	4.27e+1	3.98e+1	3.50e+1	4.21e+1	4.57e+1	2.48e+1
	标准差	1.67e-1	1.06e+0	3.32e+0	2.07e+0	7.41e-1	6.62e-1
F3	最优值	2.59e+2	1.900e-10	1.99e+2	2.10e-12	1.99e+1	0
	最差值	3.07e+2	9.95e-1	2.42e+2	8.07e-12	2.14e+2	0
	均值	2.94e+2	4.98e-2	2.26e+2	5.15e-12	1.41e+2	0
	标准差	1.19e+1	2.22e-1	1.01e+1	2.13e-12	7.66e+1	0
F4	最优值	7.20e-08	0	0	0	0	0
	最差值	1.58e-08	0	0	1.22e-15	7.40e-3	0
	均值	1.12e-08	0	0	6.10e-17	3.70e-4	0
	标准差	2.26e-09	0	0	2.73e-16	1.65e-3	0
F5	最优值	1.48e-05	1.32e-11	7.99e-15	1.43e-10	7.99e-15	4.44e-15
	最差值	2.54e-05	1.71e-10	7.99e-15	2.62e-09	1.51e-14	4.44e-15
	均值	1.99e-05	5.13e-11	7.99e-15	8.44e-10	1.10e-14	4.44e-15
	标准差	2.78e-06	4.40e-11	0	8.36e-10	3.51e-15	0

由表 3 数值进行分析, 算法 1 作为测试基准, 其寻优精度与其他算法相比是最差的。

再分析比较算法 2~5。算法 4 用于解决 CR 参数敏感的问题。F3 是 CR 敏感的问题, 算法 4 对其寻优精度最高, 可见算法 4 达到了目的。算法 5 为 DE/rand-to-best/1 算法, 有快速收敛的特性, 如针对单峰函数 F1 表现出了最好的寻优精度。但是对于

复杂多峰函数易于陷入局部极值, 如其在 F4 函数寻优过程中, 最优值与最差值相差很大, 20 次测试有一次失败, 可以推测原因为陷入局部极值。算法 2 将参数设计为变化的随机参数, CR 与 F 的多样性使其对于问题的适应性和寻优精度与算法 1 比都有提高。如针对函数 F1 与 F5 提高了精度, 在 F3 问题上在算法 1、3、5 都对其无效时, 算法 2 对其有

一定的效果。算法3性能在算法1与算法5之间，精度优于算法1差于算法5，但与算法5相比不易陷入局部极值，如对于函数F4，20次测试中，算法5有一次寻优失败，算法3全部成功。

算法6是算法2-5的组合，其中算法4在算法6中用于解决CR敏感的问题，算法5用于加快收敛提高精度，算法2与算法3的存在都是为了在探索与开发之间平衡，但处理角度不同，在算法6中也负责算法的多样性问题。从数值分析看，算法6能使子算法互相促进，最终的寻优精度在F2-F5问题上都有改进，比其使用的4种子算法单独求解精度都高，且可靠性也较好。算法6在F1问题上精度稍差于算法5，因为种群协同策略一定程度上减慢了DE/rand-to-best/1的收敛速度，但是也一定程度上避免了其陷入局部极值。可见算法6在复杂问题上相对更有效。

在迭代次数2000次内，算法1~6对于F2函数寻优似乎都无效，算法6精度比算法1~5有极小改进。本文尝试将进化代数增加到5000代，并重复测试20次。测试结果如表4所示，算法1-5仍没有明显改进，但算法6精度有了很大提高，可见随着进化代数的增加，算法6能将算法的优势放大，扩展了算法的适用性。

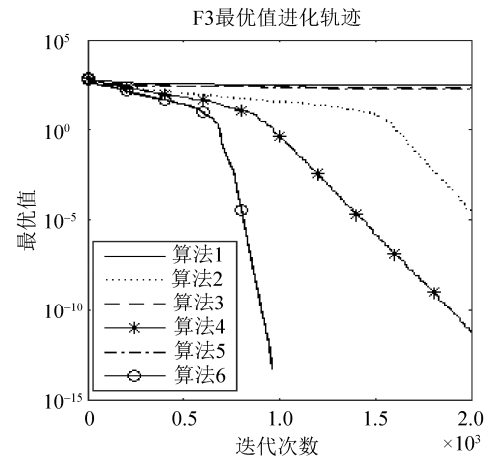
表4 F2寻优5000代测试结果

Tab. 4 Optimization results for F2 by 5000 generation evolution

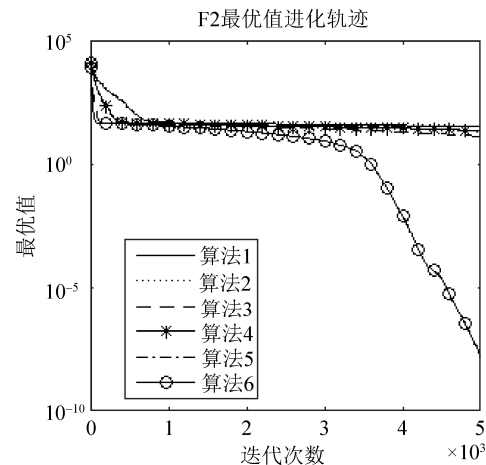
算法	均值(标准差)	最优(最差)
1	3.31e+1(2.14e-1)	3.29e+1(3.36e+1)
2	2.31e+1(1.40e+0)	2.07e+1(2.55e+1)
3	1.44e+1(2.13e+0)	1.31e+1(2.03e+1)
4	2.22e+1(9.90e-1)	2.12e+1(2.48e+1)
5	3.84e+1(4.43e+1)	3.43e+1(4.82e+1)
6	1.87e-08(1.40e-08)	2.59e-09(4.19e-08)

测试方案b)通过最优值进化轨迹图，表明算法的收敛过程，选取了典型的两幅图，如图1所示。图1(a)为F3的最优值进化轨迹，对于函数F3算法4是起决定作用的算法，其他算法都无效，但是算法6通过算法组合不仅提高了寻优精度，

并且比算法4显著的加快了收敛速度。图1(b)为F2运行5000代的最优值进化轨迹。在其他算法都无效的情况下，算法6通过算法组合找到了进化方向并加快了收敛速度。



(a) F3进化2000代



(b) F2进化5000代

图1 平均最优值进化轨迹

Fig. 1 Convergence curves of average optimal value

测试方案c)通过运行时种群个体的实时分布图可以分析算法6中4种子算法的互补性体现。通过对各个函数运行时观察第1、2维平面个体的实时分布，发现算法5极易陷入局部极值，在对F2、F3、F4寻优时都曾偏离最优值区域，说明算法5探索能力较弱，但其开发能力强，搜索精度高。而算法1~4探索能力强于算法5，但深度开采能力不够。图2为各算法对F2函数运行5000次终止时的个体在第1、2维平面的分布，图2(a)为算法1~5

的状态,可见算法 5 已收敛到一点,从坐标看寻优精度极高,但是从表 4 结果看是几个算法中最差的,可见陷入了局部极值。而其他算法虽未偏离最优值区域,但进化极慢或停滞。图 2(b)是算法 6 的 4 个子种群的状态,可见算法 6 避免了其算法 5 子种群陷入局部极值或是将算法 5 子种群从局部极值域中牵引出来,因而能带动其他算法子种群快速向极值移动。两幅图对比可以看出,通过种群协同策略,算法 5 易陷入局部极值的问题,一定程度上可以被其他算法弥补。而算法 5 收敛较快的特性可以帮助其他三个算法种群加速进化。

测试方案 d)通过特殊设计的测试来验证多种

种群多算法方案的意义。将函数 F5 作用域变大到一定范围并且不对称会使寻优难度增加。将 F5 作用域设为 $[-100,60]$ 进行寻优测试,重复测试 20 次,结果如表 5 所示。可见算法 6 拥有明显的优势,表明算法 6 适用性比原单个算法扩大。

多种群方案相当于增加了个体数,若其他算法简单的增加个体数是否也能提高寻优效果,本文进行了测试验证。将算法 1~5 的种群个体数设计为 500,运行 20 次结果如表 6 所示。从结果看,算法寻优精度并没有明显提升,甚至有的算法结果会变差。即简单的增加种群个体数,并不一定能获得寻优性能的提升。

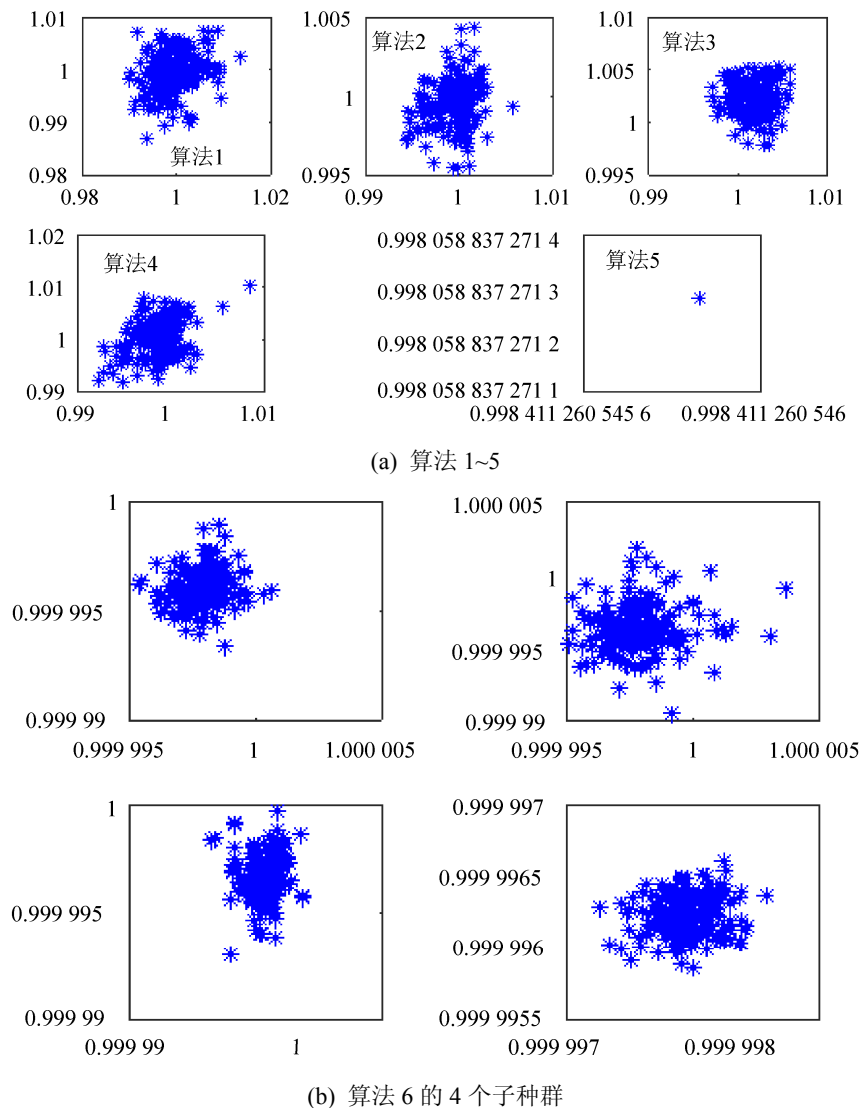


图 2 F2 函数寻优第 1、2 维平面个体位置分布图

Fig. 2 Population distribution for F2 in the plane of the 1,2 dimension

表5 变作用域 F5 寻优结果(均值/标准差)

Tab. 5 Optimization results for F5 with changed domain (mean value/ standard deviation)

算法1	算法2	算法3	算法4	算法5	算法6
2.00e+1/ 3.17e-4	2.00e+1/ 3.06e-4	2.00e+1/ 1.45e-3	2.00e+1/ 1.92e-3	2.04e+1/ 4.44e-2	1.05e-13/ 1.38e-13

表6 种群个体500寻优结果(均值/标准差)

Tab. 6 Optimization results of 5 algorithms with population size 500 (mean value/ standard deviation)

函数	算法1	算法2	算法3	算法4	算法5
F1	2.05e-8/ 3.53e-9	9.49e-19/ 1.95e-19	3.68e-59/ 8.38e-60	4.16e-17/ 1.32e-17	3.83e-108/ 1.85e-108
	4.28e+1/ 5.29e-2	3.68e+1/ 2.64e-1	3.29e+1/ 1.61e-1	3.61e+1/ 7.79e-1	3.38e+1/ 3.28e+0
F2	2.86e+2/ 7.89e+0	1.88e-6/ 1.22e-6	2.23e+2/ 8.15e+0	1.90e-11/ 4.20e-12	1.94e+2/ 1.37e+1
	3.06e-8/ 3.24e-9	0/ 0	0/ 0	0/ 0	0/ 0
F3	3.27e-5/ 2.59e-6	1.59e-10/ 1.31e-11	7.28e-15/ 1.50e-15	1.06e-9/ 9.63e-11	7.99e-15/ 0
	F4	F5			

多种群寻优方法,各子种群算法的选择与组合会影响寻优效果。仍采用4个种群进行测试,测试一为多种群同一算法(算法4),种群间采用个体循环迁移方案,每50代一次选每个种群5%的最优个体迁移至下个种群。测试二将测试一的4个种群其中一个的算法替换为算法5进行测试。测试三为本文算法6。测试函数选择算法4最适合的函数F3,其最优值进化轨迹如图3所示。测试结果说明算法选择的不同会影响寻优过程,多种群单一算法优于单种群单一算法(500个体),多种群算法4、5组合优于多种群单一算法说明算法4、5可以互补促进。而本文的多种群4种算法优于2种算法组合,说明算法6选择的4种算法在算法6中各有其作用。

通过基本测试与特殊测试,可见本文算法的4种子算法各有特点,而本文算法通过种群协同优化达到优势互补的目的。

多种群算法的缺点在于增加了计算量,因此在计算花费这一点上明显不占优势。若以函数评价次数为标准,其他算法可以增加进化代数与算法6的计算量等价。本文将算法1-5进化代数增加至8000次,运行20次结果如表7所示。

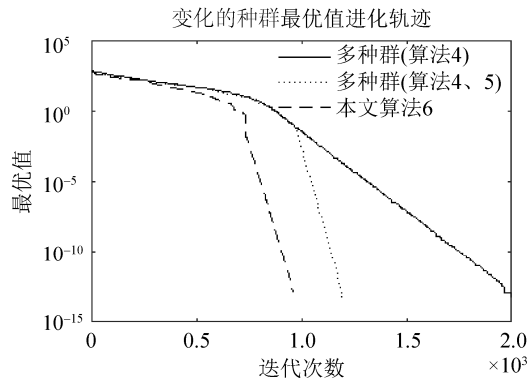


图3 多种群不同算法测试(最优值进化轨迹)
Fig. 3 Test for different multi-population design with different sub-algorithms (convergence graph)

表7 算法寻优8000代寻优结果(均值/标准差)

Tab. 7 Optimization results of 5 algorithms evolving 8000 generations (mean value/ standard deviation)

函数	算法1	算法2	算法3	算法4	算法5
F1	5.33e-48/ 2.16e-48	5.93e-91/ 2.83e-91	9.05e-245/ 0	1.20e-83/ 1.10e-83	0/ 0
	2.41e+1/ 1.73e-1	8.58e+0/ 2.14e+0	1.36e-3/ 1.03e-3	7.21e+0/ 9.02e-1	3.73e+1/ 2.97e+0
F2	2.45e+2/ 1.38e+1	9.94e-2/ 3.15e-1	1.87e+2/ 1.00e+1	0/ 0	1.81e+1/ 5.91e+0
	0/ 0	0/ 0	0/ 0	0/ 0	9.86e-4/ 3.12e-3
F3	7.99e-15/ 0	6.57e-15/ 1.83e-15	6.92e-15/ 1.72e-15	6.92e-15/ 1.72e-15	9.41e-15/ 3.00e-15
	F4	F5			

从结果看,增加进化代数可以改变寻优精度,但不能改变算法的适用性,算法1~5中没有算法对所有5个函数都适用。

对问题的适用性、鲁棒性正是多种群多算法优化策略的最大优势,在一定程度上避免了工程应用中算法选择的困难。因此研究多种群多算法协同优化有实际意义。

4 结论

多种群并行协同进化能做到种群信息共享,达到种群间互相弥补互相促进的目的。本文在多个种群基础上,探讨算法的选择问题。算法的选择在多种群进化中有一定的意义,算法应该具有不同特点,达到互补的功能。这样种群间数据的共享可以更有效地达到算法优势共享,弥补缺陷的目的。从仿真

结果看, 本文的算法增加了鲁棒性, 增加了算法的适用性、同时也增加了寻优精度和可靠性。在工程应用中, 多种群多算法协同进化, 虽然增加了算法耗费的时间, 但可以避免算法选择的困难, 避免对不同算法的一次次尝试, 在应用研究中很有意义。

参考文献:

- [1] David H Wolpert, William G. Macready. No Free Lunch Theorems for Optimization[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 1997, 1(1): 67-82.
- [2] Gämperle R, Müller S D, Koumoutsakos P. A Parameter Study for Differential Evolution[C]// Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, WSEAS Press, 2002:293-298.
- [3] Ronkkonen J, Kukkonen S, Price K V. Real Parameter Optimization with Differential Evolution[C]// Proc. IEEE CEC. USA: IEEE, 2005: 506-513.
- [4] Das S, Konar A, Chakraborty U K. Two Improved Differential Evolution Schemes for Faster Global Search[C]// GECCO'05. USA: ACM, 2005: 991-998.
- [5] Qin A K, Huang V L, Suganthan P N. Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2009, 13(2): 398-417.
- [6] 刘荣辉, 王斌斌, 郑建国. 分步交叉差分进化算法及仿真应用[J]. 系统仿真学报, 2013, 25(7): 147-153.
Liu Rong-hui, Wang Bin-bin, Zheng Jian-guo. Two Section Crossover Differential Evolution Algorithm and Its Application[J]. Journal of System Simulation, 2013, 25(7): 147-153.
- [7] 徐小平, 白博, 钱富才. 基于改进差分进化算法的 Wiener 模型辨识[J]. 系统仿真学报, 2016, 28(1): 147-153.
Xu Xiaoping, Bai Bo, Qian Fucui. Identification of Wiener Model Based on Improved Differential Evolution(SADE) Algorithm[J]. Journal of System Simulation, 2016, 28(1): 147-153.
- [8] Brest J, Greiner S, Boškovic B, et al. Self-adapting Control Parameters in Differential Evolution: a Comparative Study on Numerical Benchmark Problems[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2006, 10(6): 646-657.
- [9] Jingqiao Z, Sanderson A C. JADE: Adaptive Differential Evolution with Optional External Archive[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2009, 13(5): 945-958.
- [10] Quan-Ke Pan, P N Suganthan, Ling Wangc, et al. A Differential Evolution Algorithm with Self-adapting Strategy and Control Parameters[J]. Computers & Operations Research (S0305-0548), 2011, 38(1): 394-408.
- [11] Ruhul A Sarker, Saber M Elsayed, Tapabrata Ray. Differential Evolution With Dynamic Parameters Selection for Optimization Problems[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2014, 18(5): 689-707.
- [12] Arnob Ghosh, Swagatam Das, Aritra Chowdhury, et al. An Improved Differential Evolution Algorithm with Fitness-based Adaptation of the Control Parameters[J]. Information Sciences (S0020-0255), 2011, 181(18): 3749-3765.
- [13] 汪慎文, 丁立新, 张文生, 等. 差分进化算法研究进展[J]. 武汉大学学报(理学版), 2014, 60(4): 283-292.
Wang Shenwen, Ding Lixin, Zhang Wensheng, et al. Survey of Differential Evolution[J]. Journal of Wuhan University(Natural Science Edition), 2014, 60(4): 283-292.
- [14] Swagatam Das, Ponnuthurai Nagaratnam Suganthan. Differential Evolution: A Survey of the State-of-the-Art[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2011, 15(1): 4-32.
- [15] Guohua Wua, Rammohan Mallipeddi, P N Suganthan, et al. Differential Evolution with Multi-population based Ensemble of Mutation Strategies[J]. Information Science (S0020-0255), 2016, 328(1): 329-345.
- [16] 夏慧明, 王志刚, 周永权. 多种群自适应差分进化算法[J]. 小型微型计算机系统, 2014, 35(4): 850-853.
Xia Hui-ming, Wang Zhi-gang, Zhou Yong-quan. Adaptive Differential Evolution Based on Multiple Population[J]. Journal of Chinese Computer Systems, 2014, 35(4): 850-853.
- [17] Wenyin Gong, Álvaro Fialho, Zhihua Cai, et al. Adaptive Strategy Selection in Differential Evolution for Numerical Optimization: An Empirical Study[J]. Information Sciences (S0020-0255), 2011, 181(24): 5364-5386.
- [18] 王凌, 沈婧楠, 王圣尧, 等. 协同进化算法研究进展[J]. 控制与决策, 2015, 30(2): 193-202.
Wang Ling, Shen Jing-nan, Wang Sheng-yao, et al. Advances in co-evolutionary algorithms[J]. Control and Decision, 2015, 30(2): 193-202.