

参数适应性分布式差分进化算法

张春美^{1,2}, 陈杰¹, 辛斌¹

(1. 北京理工大学 a. 自动化学院, b. 复杂系统智能控制与决策教育部重点实验室, 北京 100081; 2. 太原科技大学 电子信息工程学院, 太原 030024)

摘要: 提出一种适应性分布式差分进化算法. 将初始种群分为多个子种群, 并设计子种群间的迁移机制, 当满足迁移条件时, 根据冯·诺依曼拓扑结构, 子种群内的优秀个体代替其邻域的较差个体, 使得整个种群实现信息共享. 同时, 根据个体适应值变化情况, 对每一个体分配不同的缩放因子 F 和交叉率 CR , 提出 F 和 CR 的适应性策略. 实验结果表明, 所提出算法有利于对解空间进行广泛探索, 避免算法陷入早熟收敛, 能够搜索到性能较好的解.

关键词: 分布式差分进化; 适应性参数; 迁移机制

中图分类号: TP18

文献标志码: A

Distributed differential evolution algorithm with adaptive parameters

ZHANG Chun-mei^{1,2}, CHEN Jie¹, XIN Bin¹

(1a. School of Automation, 1b. Key Laboratory of Complex System Intelligent Control and Decision, Ministry of Education, Beijing Institute of Technology, Beijing 100081, China; 2. School of Electronic Information Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China. Correspondent: ZHANG Chun-mei, E-mail: zcm10606@163.com)

Abstract: An adaptive distributed differential evolution algorithm is proposed based on the change of the individual's fitness value. Firstly, the initial population is divided into several subpopulations. When the migration condition is satisfied, the best individual in each subpopulation will replace the worst individual of its neighbor subpopulations according to the Von Neumann topology. The migration mechanism among subpopulations enables the information to be communicated in the whole population. Meanwhile, the adaptive mechanism of F and CR is presented for assigning different F and CR to each individual according to the individual's fitness. Numerical results show that, the proposed algorithm is beneficial to explore the solution space, which can avoid the premature convergence and search the excellent solutions.

Key words: distributed differential evolution; adaptive parameters; migration mechanism

0 引言

差分进化 (DE) 算法是一种基于种群的全局搜索优化算法, 借助于种群个体之间的差分信息对个体形成扰动进而探索整个种群空间, 并利用贪婪竞争机制选择下一代个体, 寻求问题的最优解^[1]. 该算法控制参数少、原理相对简单、易于理解和实现, 且其表现出来的高可靠性、强鲁棒性和良好的优化性能, 已成为进化计算研究领域的热点课题^[2-3].

鉴于 DE 算法在操作上具有高度的并行性, 分布式运行方式成为 DE 算法研究的重要分支, 目前已引起众多学者的关注. Tasoulis 等^[4]讨论了 DE 算法如何以并行方式进行运行, 并结合单向环拓扑结构, 提出

并行 DE 算法 (PDE), 以提高解的性能和收敛速度, 同时对并行 DE 算法的迁移规则, 包括迁移策略、迁移频率、子种群的数目等进行研究. Kozlov 等^[5]对并行 DE 算法提出新的迁移机制, 从而提高了算法的收敛速度. Singh 等^[6]对并行分布式 DE 算法的参数进行了详细的研究, 并对不同的迁移周期、迁移个体数目和子种群数对算法的影响进行了讨论. Apolloni 等^[7]对文献 [4] 进行推广, 提出了一种更通用的分布式 DE 算法 (IBDDE), 该算法由五元数组阐述算法的主要原理. 为了解决图像的配准问题, Falco 等^[8]提出分布式 DE 算法 (DDE), 与 PDE 和 IBDDE 不同, DDE 采用环形拓扑结构代替单向环. Weber 等^[9]对分布式 DE 算法中

收稿日期: 2013-01-16; 修回日期: 2013-03-11.

基金项目: 国家杰出青年科学基金项目(60925011); 国家自然科学基金委国际(地区)合作项目(61120106010); 山西省青年科技研究基金项目(2012021012-4); 太原科技大学校青年基金项目(20113003).

作者简介: 张春美(1978-), 女, 讲师, 博士生, 从事智能优化理论、方法与应用的研究; 陈杰(1965-), 男, 教授, 博士生导师, 从事复杂生产过程建模、优化与调度等研究.

采用多个缩放因子 F 进行研究, 设计了一种缩放因子 F 适应性变化的分布式 DE 算法 (FACPDE)^[10]. 文献 [11] 提出动态调节 F 和 CR 的分布式 DE 算法 (DDE-PTS), 实验结果显示, 采用多个缩放因子和交叉率能够有效提高分布式 DE 算法的性能.

受文献 [10-11] 的启发, 本文提出对参数缩放因子 F 和交叉率 CR 适应性调节的分布式 DE 算法. 根据冯·诺依曼拓扑结构将初始种群分为多个子种群, 子种群通过迁移策略与其相邻的子种群进行交流, 从而使得整个种群实现信息共享. 同时根据进化过程中个体适应值的变化情况适应性调节参数 F 和 CR, 提出参数适应性分布式 DE 算法 (APDDE). 实验结果显示, 与几种现有的分布式 DE 算法和新的改进 DE 算法^[12]相比, 所提出算法具有较好的收敛性能, 且能够寻求到质量较好的解.

1 传统 DE 算法

DE 算法是一种基于群体进化的算法, 能够记忆个体向量最优解, 并对种群内信息进行共享. DE 算法进化首先在解的取值范围内生成一个随机的初始种群, 然后通过差分变异、交叉、选择操作, 产生新一代种群. DE 算法基于实数编码, 首先在问题的可行解空间生成随机初始化种群 $\mathbf{x}_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \dots, \mathbf{x}_{i,D})$, $i = \{1, 2, \dots, NP\}$. 其中: D 为问题维数, NP 为种群规模.

1.1 差分变异

在 DE 算法中, 种群内个体的差分向量经过缩放后, 与种群内另外的相异个体相加得到变异向量. 根据变异向量生成方法的不同, 形成了多种变异策略^[13], 其中变异方式 DE/rand/1 的方程为

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_0,g} + F(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}). \quad (1)$$

其中: $r_0 \neq r_1 \neq r_2 \neq i$, F 为缩放因子.

1.2 交叉

交叉操作生成实验向量, 通过随机选择, 使得实验向量至少有一位分量由变异向量贡献. 交叉操作的方程为

$$u_{i,j,g} = \begin{cases} v_{i,j,g}, & \text{rand}(0, 1) \leq \text{CR} \text{ or } j = j_{\text{rand}}; \\ x_{i,j,g}, & \text{otherwise.} \end{cases} \quad (2)$$

其中: $j = 1, 2, \dots, D$; $\text{CR} \in (0, 1)$ 为交叉率.

1.3 选择

DE 采用“贪婪”搜索策略, 根据目标向量和实验向量的适应值选择最优个体, 对于最小化问题, 选择操作的方程

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g}, & f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{i,g}); \\ \mathbf{x}_{i,g}, & \text{otherwise} \end{cases} \quad (3)$$

为下一代的目标向量.

2 适应性分布式 DE 算法

2.1 分布式 DE 的迁移机制

在本文算法中, 将初始种群 P 分为 m 个子种群, 每一子种群的大小为

$$np_i = NP/m, \quad i = 1, 2, \dots, m.$$

子种群按照冯·诺依曼拓扑结构进行排列, 将每一子种群定义为一个节点, 对于各节点而言, 与其相邻的 4 个节点定义为其邻域, 如图 1 所示. 当前节点用黑色表示, 邻域节点用灰色表示. 每一节点仅与其邻域节点进行信息交流, 由于每一节点在同一时刻不仅向其邻域提供信息同时接收其邻域的信息, 信息能够在整个拓扑结构中进行交流.

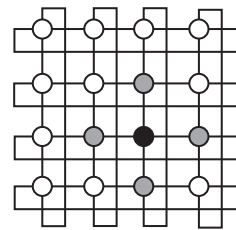


图 1 冯·诺依曼拓扑结构

每一子种群 $P_i (i = 1, 2, \dots, m)$ 的个体利用 DE 算法进行优化并参与竞争, 选出其最好的个体, 每隔 γ 代, 子种群之间采用迁移进行信息交换, 迁移条件为

$$f(\mathbf{x}_{\text{best}}^k) \leq f(\mathbf{x}_{\text{worst}}^{k+n}), \quad n = 1, 2, 3, 4,$$

其中 k 为当前子种群. 即当被迁移个体优于被代替个体时, 将每一子种群内最好的个体代替其邻域子种群的最差个体. 各子种群间通过迁移机制使得信息可以在整个种群内进行交流, 有利于对解空间进行广泛探索并寻求性能较好的解.

2.2 分布式 DE 算法中 F 和 CR 的适应性调节策略

F 和 CR 的取值直接影响到种群最优适应值的大小, 而根据种群适应值反过来调节 F 和 CR, 在不同的情形下采用合适的 F 和 CR, 能够促进算法的优化过程, 使得算法易于求到较优的解.

2.2.1 F 的适应性调节策略

在分布式 DE 算法中, 子种群内的每一个体通过差分变异、交叉和选择生成新的个体, F 的作用是对子种群内每一个体所对应的差分变异向量进行缩放, 确定当前个体的搜索范围, 生成变异向量. 当个体适应值较优时, 希望继续利用该个体的信息, 因此, 此时需要采用较小的 F , 使算法在此较优个体的附近继续进化, 易于求得较好的解; 当个体的适应值较差时, 个体的信息应较少被利用, 此时需要扩大解的搜索范围, 因此, 采用较大的 F , 使算法探索较大的解空间, 增加

寻求到最优解的几率.

鉴于以上分析, 本文根据个体适应值的变化情况确定每一个体对应的 F_i , 对 F 的调节策略如下所示:

$$F_i = \left(\frac{\Delta f(\mathbf{x}_i)}{\lambda + (f(\mathbf{x}_{\text{worst}}) - f(\mathbf{x}_{\text{best}}))} \right) (F_{\max} - F_{\min}) + F_{\min}. \quad (4)$$

其中: $\Delta f(\mathbf{x}_i) = |f(\mathbf{x}_i) - f(\mathbf{x}_{\text{best}})|$; \mathbf{x}_{best} 为当前种群内具有最优适应值的目标向量; $\mathbf{x}_{\text{worst}}$ 为当前种群内具有最差适应值的目标向量; $\lambda = 10^{-13}$ 为避免分母是“0”而加的较小的数, 此时 $f(\mathbf{x}_{\text{worst}}) - f(\mathbf{x}_{\text{best}}) = \Delta f(\mathbf{x}_i) = 0$ 成立. 由式(4)可见, 当 $\Delta f(\mathbf{x}_i) \rightarrow 0$ 时, $F_i \rightarrow F_{\min}$; 当 $\Delta f(\mathbf{x}_i) \rightarrow \infty$ 时, $f(\mathbf{x}_{\text{worst}}) - f(\mathbf{x}_{\text{best}}) \rightarrow \infty$ 成立, 此时 $F_i \rightarrow F_{\max}$. F_{\max} 和 F_{\min} 分别为 F 取值的上限和下限.

式(4)所示的调节策略中, 当个体的适应值与当前种群内最优适应值较接近时, 采用较小的缩放因子 F_i , 对其进行较小的扰动, 从而使其在最优值附近的小区域内进行探索, 增强算法的开发能力. 反之, 当个体的适应值与当前种群内最优适应值相距较远时, 对其进行较大扰动, 探索解空间的较大区域, 寻求更优的适应值, 增强算法的探索能力.

2.2.2 CR 的适应性调节策略

CR 决定子种群内个体的实验向量取自变异向量还是目标向量, 当变异向量的适应值 ($f(\mathbf{v}_i)$) 较优时, 实验向量要以较大的概率取自变异向量, CR 应取较大的值; 当变异向量的适应值较差时, 实验向量要以较小的概率取自变异向量, 此时 CR 应取较大的值. 鉴于以上分析, 本文提出根据适应值的变化情况调节 CR 的适应性调节机制如下所示:

$$CR_i = \begin{cases} CR_{\max}, & f(\mathbf{v}_i) \leq f(\mathbf{x}_{\text{best}}); \\ CR_{\min}, & f(\mathbf{v}_i) \geq f(\mathbf{x}_{\text{worst}}); \\ CR_{\min} + \frac{CR_{\max} - CR_{\min}}{\frac{\Delta f(\mathbf{x}_i)}{e^{f(\mathbf{x}_{\text{worst}}) - f(\mathbf{x}_{\text{best}})}}}, & \text{otherwise.} \end{cases} \quad (5)$$

其中: $\Delta f(\mathbf{v}_i) = f(\mathbf{v}_i) - f(\mathbf{x}_{\text{best}})$, CR_{\max} 和 CR_{\min} 分别为 CR 取值的上限和下限. 式(5)中, 当 $f(\mathbf{v}_i)$ 较优, 即 $f(\mathbf{v}_i) \leq f(\mathbf{x}_{\text{best}})$ 时, 实验向量要以较大的概率取自变异向量, 此时, $CR_i = CR_{\max}$; 当 $f(\mathbf{v}_i) \geq f(\mathbf{x}_{\text{worst}})$, 变异向量的适应值较差时, 实验向量要以较小的概率取自变异向量, 此时 CR 设置为 $CR_i = CR_{\min}$. 当 $\Delta f(\mathbf{v}_i) \rightarrow 0$ 时, $CR_i \rightarrow CR_{\max}$; 当 $\Delta f(\mathbf{v}_i) \rightarrow \infty$ 时, $CR_i \rightarrow CR_{\min}$.

在式(5)中, 当变异向量取得较优适应值时, 希望将此变异向量的信息保留下来, 因此取较大的 CR; 反

之, CR 取值要尽量小. 上述调节 CR 的策略使得每一个体的交叉率 CR_i 根据变异向量的适应值进行变化. 当 CR 较大时, 实验向量的信息会更多地取自于变异向量, 而当 CR 较小时, 实验向量的分量则更多地取自目标向量, 达到预期的变异效果.

3 数值实验和分析

3.1 标准测试函数

所提出算法用于对标准测试函数进行优化, 采用的标准测试函数为

$$f_1(x) = \sum_{i=1}^D |x_i| + \prod_{i=1}^D |x_i|, |x_i| < 10;$$

$$f_2(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2, |x_i| < 1;$$

$$f_3(x) = \sum_{i=1}^D ix_i^4 + U(0, 1), |x_i| < 1.20;$$

$$f_4(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e,$$

$$|x_i| < 1;$$

$$f_5(x) = \sum_{i=1}^D |x_i \sin x_i + 0.1x_i|, |x_i| < 10;$$

$$f_6(x) = \frac{1}{4000} \sum_{i=1}^D (x_i^2) - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1,$$

$$|x_i| < 600;$$

$$f_7(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10), |x_i| < 5.12;$$

$$f_8(x) = \sum_{i=1}^D 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2,$$

$$|x_i| < 2.048;$$

$$f_9(x) = 418.9829D - \sum_{i=1}^D \sin(|x_i|^{1/2}), |x_i| < 500.$$

其中: f_1 和 f_2 为单模态函数; f_3 为有噪声二次函数; $U(0, 1)$ 为在区间 $[0, 1]$ 内均匀分布的随机数; 当维数 $D > 3$ 时, f_8 为多模态函数; $f_4 \sim f_7$ 和 f_9 为多模态函数, 存在多个局部最优值, 且最优值个数随着问题维数的增大呈指数倍增长. 以上函数的最小值均为“0”.

3.2 APDDE 算法分析

为了分析 APDDE 算法中 F 和 CR 的调节机制, 选取单模态函数 f_2 和多模态函数 f_6 进行分析, DE 变体选为 DE/rand/1/bin, 种群规模 $NP = 400$, 终止条件为函数计算次数达到 $NFE = 10^6$. 为了考察算法的综

合性能,考虑维数 $D = 10$ 、 $D = 50$ 和 $D = 100$ 三种情况. 迁移周期 $\gamma = 100$, 子种群个数 $m = 5$ ^[14]. F_{\max} 和 F_{\min} 分别设置为 0.5, 0.1^[15], CR_{\max} 和 CR_{\min} 分别设置为 0.9, 0.1^[16]. 当优化函数为 f_2 和 f_6 时, 随机选择种群内的一个个体 (选择结果为第 300 个个体, $i = 300$) 的 F 、 CR 和整个种群适应值 (f) 的变化情况如图 2 所示, 图 2 中, “*” 线为 CR , “×” 线为 F , 直线为 f .

由图 2 可见, 随着优化过程的进行, 对 F 的适应性调节使得生成的变异向量较优, 因此, 进化后期算法的 CR 逐渐增加, 实验向量的分量以较大的概率取自变异向量, 使整个种群得到优化. 随着 CR 的适应性调节, 个体适应值逐渐减小, 算法在进化后期着重于对优秀解的开发功能, 因此, F 逐渐减小, 在当前个体附近的较小范围内进行探索. 对随机选择的种群内个体 F 和 CR 进行调节, 当维数 $D = 10$, 函数 f_2 和 f_6 分别在计算次数达到 1×10^5 和 2×10^5 时, F 和 CR 经过调节稳定在 0.1, 0.9; 当 $D = 50$, 函数 f_2 和 f_6 分别在计算次数达到 7×10^5 和 6×10^5 时, F 和 CR 经过调节分别稳定在 0.1, 0.9; 当 $D = 100$ 时, 对于 f_2 , CR 在计算次数为 7×10^5 时稳定在 0.9, F 在计算次数为 6×10^5 时稳定在 0.1, 对于函数 f_6 , F 和 CR 分别在计算次数为 7×10^5 和 8×10^5 时稳定在 0.1, 0.9.

3.3 结果和性能比较

本文将所提出算法与分布式 DE 算法 (IBDDE、DDE、FACPDE、DDE-PTS) 和改进的 DE 算法 (GDE)

相比较, 鉴于比较的合理性和公平性, DE 变体为 DE/rand/1/bin, 种群规模 $NP = 400$, 终止条件为函数计算次数达到 $NFE = 10^6$. IBDDE 和 DDE 中, F 设置为 0.5^[7-8], FACPDE 的 F 根据文献 [10] 的建议进行设置, 以上 3 种算法中 CR 均设置为 0.9^[7], DDE-PTS 和 GDE 的 F 和 CR 分别按文献 [11] 和 [12] 设置. 算法主要参数如表 1 所示.

由图 2 可知, 当 $D = 10$ 时, 各种算法对于函数 $f_1 \sim f_9$ 均在较小的计算次数寻求到较好的解, 因此表 2 仅列出了用于优化性能比较的各种算法对 50 维和 100 维 $f_1 \sim f_9$ 的优化结果, 每一测试函数均运行 30 次, 所取得的最小值、平均值 \pm 标准方差如表 2 所示. 对于函数 $f_1 \sim f_9$, 当维数为 50 时, APDDE 均取得最好的最小值、平均值; 除函数 f_4 外 (DDE 取得最好的标准方差, APDDE 次之), APDDE 对其余测试函数均取得最好的标准方差; 尤其对于函数 f_1, f_2, f_3, f_5 和 f_9 , APDDE 取得的优化结果明显优于其他算法. 当维数为 100 时, 对于函数 $f_1 \sim f_9$, APDDE 均取得最好的最小值; 除函数 f_2 和 f_6 , APDDE 取得的平均值与标准方差与其他函数相当外, 其余函数 APDDE 均取得最小的平均值和标准方差.

为了测试算法 APDDE 的收敛性能, 对 10 维、50 维和 100 维 $f_1 \sim f_9$ 函数进行优化, 每一次优化过程记录 11 个点, 取每一个点 30 次运行结果的平均值得到收敛曲线. 以其中较难寻优的多模态函数 f_4, f_5, f_7 和 f_9 为例, 算法的收敛性能比较如图 3 所示.

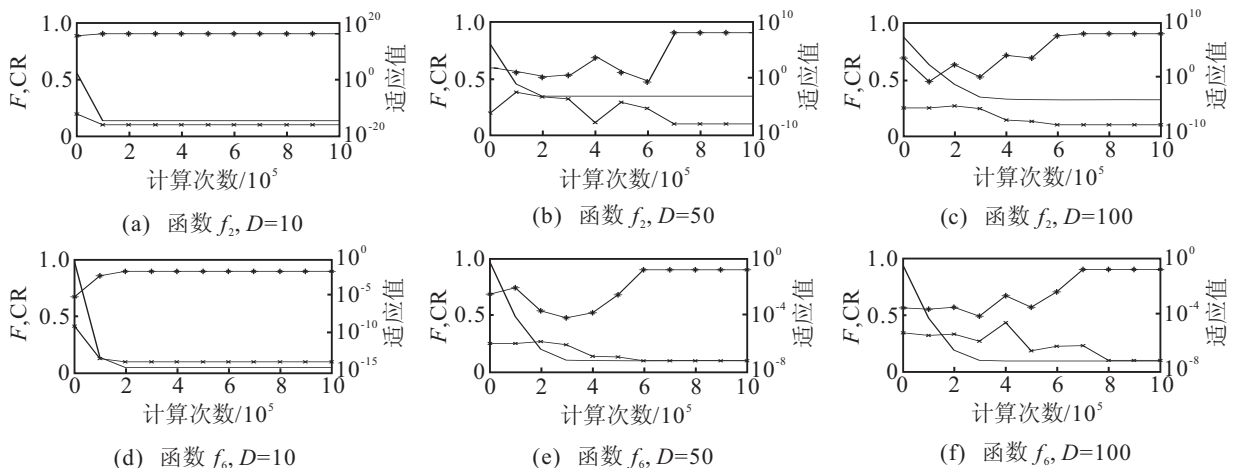


图 2 F 、 CR 和整个种群适应值 (f) 的变化情况

表 1 IBDDE、DDE、FACPDE 和 APDDE 的主要参数设置

算法	子种群数目	迁移周期/代	拓扑结构	迁移个体数	被复制个体选择方式	被代替个体选择方式
IBDDE	6	100	单向环	1	随机	随机
DDE	16	5	环形网状	4	最好	最差
FACPDE	5	1	单向环	1	最好	随机
DDE-PTS	4	10	单向环	1	最好	最差
APDDE	5	100	冯·诺依曼	4	最好	最差

表2 $f_1 \sim f_9$ 的最小值、平均值 \pm 标准方差

算法	D	指标	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
IBDDE	50	min	1.27e+08	4.45e+04	2.73e+01	2.45e+00	6.03e+01	2.76e+02	4.90e+02	1.97e+03	1.40e+04
		mean	1.47e+08	8.54e+04	4.85e+01	2.78e+00	7.18e+01	3.69e+02	5.75e+02	3.31e+03	1.53e+04
		std	1.31e+02	2.35e+04	1.17e+01	1.30e-01	5.10e+00	5.32e+01	3.03e+01	6.43e+02	5.20e+02
	100	min	2.90e+02	4.02e+06	8.60e+02	2.87e+00	1.55e+02	8.51e+02	1.15e+03	8.78e+03	3.35e+05
		mean	1.01e+55	1.13e+07	1.59e+03	3.15e+00	1.83e+02	1.28e+03	1.34e+03	1.45e+05	3.44e+05
		std	3.35e+55	1.78e+07	1.55e+03	2.80e-01	3.81e+01	6.91e+02	1.50e+02	1.09e+05	2.87e+03
DDE	50	min	1.18e+02	2.27e+04	1.16e+01	2.47e+00	4.67e+01	1.97e+02	4.67e+01	1.97e+02	1.49e+04
		mean	1.53e+02	6.92e+04	3.84e+01	2.47e+00	6.29e+01	3.26e+02	6.29e+01	3.26e+02	1.59e+04
		std	1.39e+02	2.62e+02	1.31e+01	4.51e-16	5.49e+00	7.29e+01	5.49e+00	7.29e+01	3.73e+02
	100	min	2.22e+02	1.96e+06	4.94e+02	2.63e+01	1.19e+02	5.54e+02	1.05e+03	3.84e+03	3.25e+05
		mean	2.86e+45	7.23e+06	1.07e+03	2.07e+01	1.34e+02	7.85e+02	1.12e+03	9.14e+03	3.33e+05
		std	2.49e+45	1.75e+07	1.91e+03	3.84e-01	4.71e+01	6.64e+02	2.24e+02	1.55e+05	1.86e+03
FACPDE	50	min	6.81e+01	1.71e+03	7.26e+01	1.63e+01	4.49e+01	9.41e+01	4.23e+02	7.71e+02	1.28e+04
		mean	1.07e+02	2.67e+04	1.63e+02	2.09e+01	5.51e+01	12.21e+02	4.60e+02	1.46e+03	1.44e+04
		std	1.14e+01	1.36e+04	7.88e+00	2.12e-01	6.96e+00	5.29e+01	2.45e+01	3.60e+02	4.25e+02
	100	min	2.21e+20	3.31e+06	6.17e+02	2.88e+00	1.39e+02	8.65e+02	1.22e+03	8.37e+03	3.33e+04
		mean	5.90e+52	9.62e+06	1.37e+03	3.06e+00	1.76e+02	1.25e+03	1.32e+03	1.39e+04	3.51e+04
		std	1.96e+53	1.51e+07	1.59e+03	2.46e-01	1.89e+01	7.25e+02	1.63e+02	1.13e+04	1.60e+03
DDE-PTS	50	min	7.12e+00	1.30e+00	1.27e+00	1.20e+00	3.45e+01	9.47e+00	5.99e+02	3.69e+01	7.82e+03
		mean	1.28e+31	1.78e+05	1.19e+02	1.76e+00	5.65e+01	1.99e+02	7.16e+02	2.97e+03	1.04e+04
		std	4.25e+31	5.91e+05	2.04e+03	7.95e-01	5.89e+01	3.97e+02	1.47e+02	9.65e+03	3.25e+03
	100	min	6.08e+01	1.01e+01	4.89e+00	1.11e+00	1.21e+01	3.06e+01	4.17e+02	1.91e+02	1.80e+04
		mean	6.16e+58	5.61e+06	6.48e+02	1.82e+00	7.69e+01	4.70e+02	6.89e+02	6.00e+03	2.30e+04
		std	2.04e+59	1.85e+07	3.75e+02	9.11e-01	8.63e+01	9.26e+02	3.74e+02	1.87e+04	7.64e+03
GDE	50	min	4.04e+00	5.02e-06	6.20e-02	2.71e-02	1.01e+01	3.98e-01	1.34e+02	4.85e+01	6.10e+03
		mean	1.59e+30	2.46e+05	1.31e+02	5.55e-01	3.01e+01	1.37e+02	2.43e+02	3.42e+03	9.55e+03
		std	5.27e+30	8.17e+05	4.33e+02	1.11e+01	4.41e+01	4.30e+02	2.48e+02	1.10e+04	4.28e+03
	100	min	3.20e+01	3.28e+00	3.86e-01	1.69e-01	4.39e+01	2.76e+00	4.13e+02	1.42e+02	2.09e+04
		mean	2.21e+50	3.82e+06	4.44e+02	8.70e-01	7.75e+01	3.20e+02	6.81e+02	4.55e+03	2.45e+04
		std	7.35e+50	1.26e+07	1.46e+03	1.08e+00	7.86e+01	9.22e+02	4.27e+02	1.37e+04	5.93e+03
APDDE	50	min	8.58e-05	8.20e-08	4.68e-03	2.56e-03	2.84e-05	8.14e-05	2.32e+01	3.74e+01	1.64e-03
		mean	4.58e-03	2.94e-05	8.31e-03	1.11e-02	2.07e-03	1.12e-01	3.28e+01	4.52e+01	3.58e-03
		std	5.08e-03	3.94e-05	1.76e-03	9.00e-03	2.58e-03	1.47e-01	4.58e+00	2.29e+00	3.15e-03
	100	min	5.09e+00	8.29e-02	2.99e-01	2.00e-09	4.86e-01	2.15e-02	1.19e-02	1.27e-04	1.21e-04
		mean	2.45e+19	3.75e+06	4.07e-01	3.69e-02	3.11e-01	5.07e+02	4.73e-03	4.32e+00	1.93e+01
		std	8.44e+19	1.18e+07	1.12e+01	1.00e-02	1.03e-01	4.71e+02	1.51e-03	1.43e+00	8.24e+01

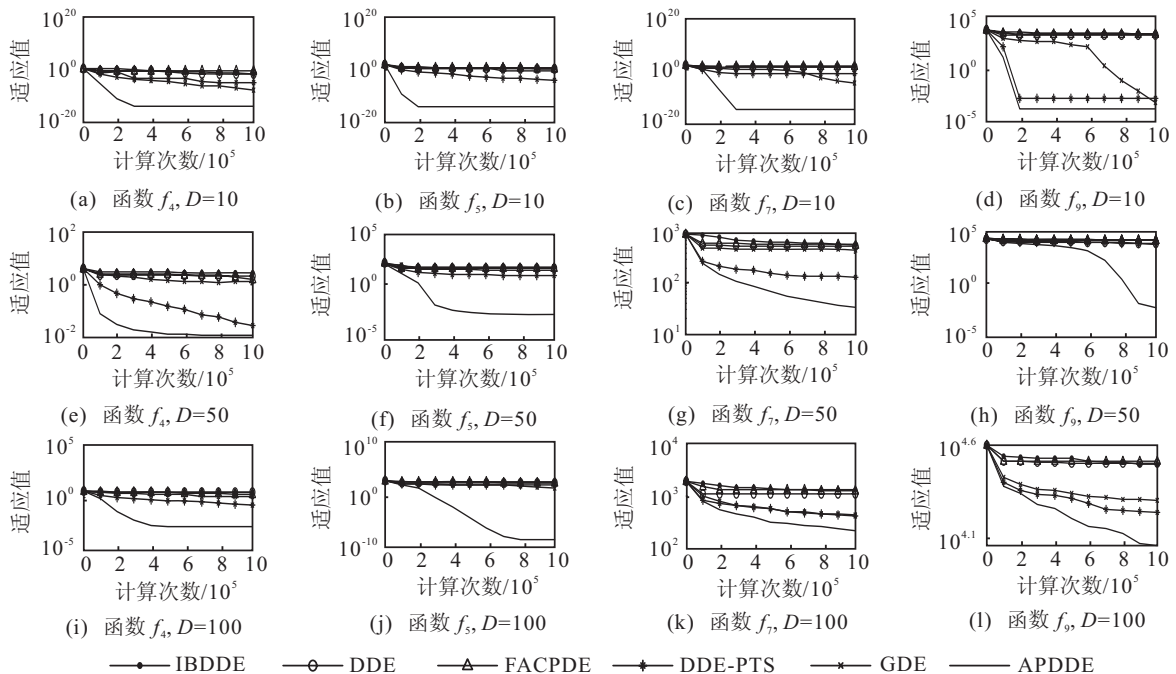


图3 算法的收敛性能比较

由图3可知,当 $D = 10$ 时,对于函数 f_4, f_5, f_7 和 f_9 , APDDE有较好的收敛性能,且在计算次数较小时寻求到较好的解(对于 f_7 , APDDE求得的最小值与DDE-PTS和GDE较接近).当 $D = 50$ 时,对于函数 f_5 和 f_9 , APDDE的性能超越其他算法很大距离,并搜索到高质量的解;对于函数 f_4 和 f_7 , APDDE的收敛性能与GDE相当,但APDDE取得了较好的最小值;对于函数 f_7 和 f_9 , APDDE算法在最大计算次数即使已经达到 10^6 ,仍具有继续寻求最优解的能力.当 $D = 100$ 时,对于函数 f_4 和 f_5 , APDDE算法具有最好的收敛性能,而其他算法均出现早熟收敛;对于函数 f_7 和 f_9 , APDDE的收敛性能与DDE-PTS、GDE相当,却取得好于DDE-PTS、GDE的解.由上述数值结果和收敛性能可知,在优化单模态函数和多模态函数时, APDDE与其他算法相比,不仅具有最好的收敛性能,而且能搜索到高质量的解,是一种高性能的数值优化算法.

4 结 论

本文提出了适应性分布式DE算法,将初始种群分为多个子种群,根据冯·诺依曼拓扑结构,用子种群的优秀个体代替其邻域子种群的较差个体,使信息周期性地进行交流,从而实现整个种群信息共享.对子种群内各个体分配不同的 F 和 CR ,根据个体适应值的变化情况提出 F 和 CR 的适应性策略.对进化过程中个体的 F 和 CR 的变化情况进行分析,并将APDDE算法与其他几种分布式DE算法和改进的DE算法进行比较,结果显示,所提出算法在收敛速度和解的性能上均取得了较满意的结果.

参考文献(References)

- [1] Storn R, 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.
- [2] Zhang Chun-mei, Chen Jie, Xin Bin, et al. Differential evolution with adaptive population size combining lifetime and extinction mechanisms[C]. *The 8th Asian Control Conf. Kaohsiung: IEEE*, 2011: 1221-1226.
- [3] Xin Bin, Chen Jie, Zhang Jia, et al. Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy[J]. *IEEE Trans on Systems Man and Cybernetics, Part C: Applications and Reviews*, 2012, 42(5): 744-767.
- [4] Tasoulis D K, Pavlidis N G, Plagianakos V P, et al. Parallel differential evolution[C]. *IEEE Congress on Evolutionary Computation. Portland: IEEE*, 2004: 2023-2029.
- [5] Kozlov K N, Sanderson A C. New migration scheme for parallel differential evolution[C]. *Int Conf on Bioinformatics of Genome Regulation and Structure. Novosibirsk: Springer*, 2006: 141-144.
- [6] Singh L, Kumar S. Parallel evolutionary asymmetric subtheorem product fuzzy-neural inference system: an island model approach[C]. *Int Conf on Computing: Theory and Applications. Kolkata: IEEE*, 2007: 282-286.
- [7] Apolloni J, Leguizamón G, García-Nieto J, et al. Island based distributed differential evolution: An experimental study on hybrid testbeds[C]. *IEEE Int Conf on Hybrid Intelligent Systems. Barcelona: IEEE*, 2008: 696-701.
- [8] Falco I D, Maisto D, Scafuri U, et al. Distributed differential evolution for the registration of remotely sensed images[C]. *IEEE Euromicro Int Conf on Parallel, Distributed and Networkbased Processing. Naples: IEEE*, 2007: 358-362.
- [9] Weber M, Neri F, Tirronen V. A study on scale factor in distributed differential evolution[J]. *Information Science*, 2011, 181(12): 2488-2511.
- [10] Weber M, Tirronen V, Neri F. Scale factor inheritance mechanism in distributed differential evolution[J]. *Soft Computing*, 2010, 14(11): 1187-1207.
- [11] Saadi H A, Chouinard J Y, Guessoum A. Distributed differential evolution algorithm for papr reductin of ofdm signal[C]. *The 11th Int Conf on Information Science, Signal Processing and Application. Montreal: IEEE*, 2012: 567-572.
- [12] Han Ming-feng, Lin Chin-teng, Chang Jyh-yeong, et al. Group-based differential evolution for numerical optimization problems[J]. *Int J of Innovative Computing, Information and Control*, 2013, 9(3): 1357-1372.
- [13] Brest J, Maučec M S. Population size reduction for the differential evolution algorithm[J]. *Applied Intelligence*, 2008, 29(3): 228-247.
- [14] Zhang Chun-mei, Chen Jie, Xin Bin. Distributed memetic differential evolution with the synergy of lamarckian and baldwinian learning[J]. *Applied Soft Computing*, 2013, 13(5): 2947-2959.
- [15] McGinley B, Maher J, O’Riordan C. Maintaining healthy population diversity using adaptive crossover, mutation and selection[J]. *IEEE Trans on Evolutionary Computation*, 2011, 15(5): 692-714.
- [16] Ghosh A, Das S, Chowdhury A, et al. An improved differential evolution algorithm with fitness-based adaptation of the control parameters[J]. *Information Sciences*, 2011, 181(18): 3749-3765.

(责任编辑: 郑晓蕾)