

一种增强型多目标烟花爆炸优化算法

谢承旺¹, 许 雷¹, 汪慎文², 肖 驰¹, 夏学文¹

(1. 华东交通大学软件学院, 江西南昌 330013; 2. 河北地质大学信息工程学院, 河北石家庄 050031)

摘 要: 现实中多目标优化问题的多样化和复杂化要求发展新的多目标优化算法. 在混合多目标进化算法设计思想和新型进化模型的启发下, 提出一种增强型多目标烟花爆炸算法 eMOFEOA, 该算法利用均匀化与随机化相结合的方法生成均匀分布的初始种群, 为算法后续搜索提供较好的起始点; 对烟花爆炸半径采用精细化控制策略, 即不同世代的种群具有不同的爆炸半径, 而且同一种群内部因个体支配强度的差异而具有不同的爆炸半径, 以节省计算资源; 利用简化的 k-最近邻方法维持外部档案的多样性. 本文算法与另 5 种对等比较算法一同在 12 个基准多目标测试函数上进行性能比较, 实验结果表明 eMOFEOA 算法在收敛性、多样性和稳定性上具有总体上显著的性能优势.

关键词: 烟花爆炸优化; 多目标优化算法; 半径精细化控制

中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2017)10-2323-09

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2017.10.002

An Enhanced Multi-objective Fireworks Explosion Optimization Algorithm

XIE Cheng-wang¹, XU Lei¹, WANG Shen-wen², XIAO Chi¹, XIA Xue-wen¹

(1. School of Software, East China Jiaotong University, Nanchang, Jiangxi 330013, China;

2. School of Information Engineering, Hebei Dazhi University, Shijiazhuang, Hebei 050031, China)

Abstract: In reality, the diversification and complexity of the multi-objective optimization problems (MOPs) require the development of some novel multi-objective optimization algorithms. Inspired by the hybrid multi-objective evolutionary algorithms (MOEAs) and new evolutionary instances, an enhanced multi-objective fireworks explosion optimization algorithm (eMOFEOA for short) is proposed to solve the hard MOPs efficiently in the paper. Firstly, the proposed approach uses the approach of combining uniformization and randomization to generate an initial population that are scattered uniformly over the feasible search space, so that the algorithm can acquire a good beginning for the subsequent iterations. Secondly, a fine control strategy of explosion radius is adopted in the eMOFEOA, that is to say, different generation of population has different radius, and the different firework in the same generation have different radius based on its strength of Pareto dominance, so as to save the computation resource to the maximum extent. Thirdly, a simplified k-nearest neighbor approach is employed to maintain the diversity of external archive in the eMOFEOA. The proposed eMOFEOA is compared with the other five peer comparison algorithms in the performance of convergence and diversity based on 12 benchmark multi-objective test functions, and the experimental results show that our eMOFEOA has the overall performance advantages in convergence, diversity and stability.

Key words: fireworks explosion optimization; multi-objective evolutionary algorithm; radius fine-controlled

1 引言

科学研究与工程实践中存在大量需要同时优化多目标目标的问题, 这些问题通常被称为多目标优化问题 (Multi-objective Optimization Problem, MOP). MOP 问题一般不存在单个的最优解, 而往往是一组折中解的

集合, 即 Pareto 解集. 由于现实 MOP 问题十分复杂, 使得经典的数学解析方法难以凑效.

进化算法 (Evolutionary Algorithm, EA) 是一种随机优化方法, 它具有适于解决 MOP 问题的一些特征^[1]: (1) EA 算法具有隐含的并行性和群体搜索的特点, 使其在一次运行中能获得多个 Pareto 最优解; (2) EA 算

收稿日期: 2016-07-15; 修回日期: 2016-10-21; 责任编辑: 蓝红杰

基金项目: 国家自然科学基金 (No. 61763010, No. 61165004, No. 61402481, No. 61663009); 江西省自然科学基金 (No. 20114BAB201025, No. 20151BAB207022, No. 20161BAB202064); 河北省青年拔尖人才支持计划 (冀字[2013]17号); 河北省自然科学基金 (No. F2015403046); 科学计算与智能信息处理广西高校重点实验室 (No. GXSCIP201604); 江西省教育厅科技项目 (No. GJJ12307, No. GJJ14373, No. GJJ150539)

法对待解 MOP 问题的数学性质不作特别要求,它可以处理各种类型的目标函数和约束条件.因此 EA 算法被广泛地用于 MOP 问题的求解,并由此产生了各具特色的多目标进化算法 (Multi-objective Evolutionary Algorithm, MOEA).

典型的 MOEAs 包括基于非劣排序的遗传算法 NSGA^[2]及其改进版 NSGA-II^[3],强度 Pareto 进化算法 SPEA^[4]及其改进版 SPEA2^[5]等.随后,一些新的进化机制陆续被引入到多目标优化领域,例如基于粒子群优化 (Particle Swarm Optimization, PSO) 模型的 MOPSO 算法^[6]以及改进的算法 OMOPSO^[7],基于分解技术的 MOEA/D 算法^[8]等.

2010 年, Tan 等^[9]通过模拟烟花爆炸中炸点的扩散机制提出了一种新颖的烟花爆炸算法 (Fireworks Explosion Algorithm, FEA). 由于烟花弹爆炸时,释放出的火星散布在以烟花弹 (炸点) 为中心的一个圆形邻域内,如果将该邻域视为问题的一个局部区域,爆炸产生的火星视为区域内的点,那么一次爆炸就类似于对局部区域的一次探索.这种探索可看成是在解空间中对该炸点附近的一次局部搜索.遗憾的是,目前大部分关于 FEA 算法的研究集中在对基本 FEA 算法的改进以及混合 FEA 算法的设计上,较少涉及到多目标烟花爆炸算法的研究.2013 年, Zheng 等^[10]将烟花爆炸算法与差分进化算法相结合,提出一种混合的多目标烟花爆炸算法 (Multi-objective Fireworks Explosion Algorithm, MOFEA) 以求解精准农业中油料作物的施肥问题.谢等^[11]将精英反向学习机制引入 FEA 算法以解决现实中复杂的 MOP 问题,取得了较好的效果.

本文在已有研究的基础上,提出一种增强型多目标烟花爆炸算法 (Enhanced Multi-objective Fireworks Explosion Optimization Algorithm, eMOFEA), eMOFEA 算法与已有单/多目标 FEA 算法的不同之处在于:(1) 利用均匀化与随机化相结合的方式初始化种群;(2) 对烟花爆炸半径进行精细化控制,不同世代的烟花群体具有不同的爆炸半径,相同世代的烟花个体因其支配强度的差异而具有不同的爆炸半径;(3) 采用一种简化的 k-最近邻方法维持档案群体的多样性.上述三种策略有机结合,以增强算法求解复杂 MOP 问题的性能.

2 eMOFEA 算法

2.1 种群初始化方法

初始种群分布性对 EA 算法的性能有重要影响,为提高 MOEA 算法的解题性能,这里采用我们此前提出的一种均匀化与随机化相结合的方法初始化种群^[12].该方法的原理是将决策向量 x 中任意一维决策变量 x_i ($i = 1, 2, \dots, n$) 均匀划分成与种群规模相等的若干等

长的子区间,初始代个体在任一基因位上的取值需首先随机选定子区间,然后再在选定的子区间内产生随机值而获得.均匀划分后的各子区间有且仅有一次机会产生基因值,这样可在最大程度上保证初始群体基因的多样化.算法 1 给出了种群多样性的初始化算法的流程.

算法 1 种群多样性的初始化方法

输入: 种群的规模 N , 决策向量的维度 n , 各决策变量 x_j ($j \in [1:n]$) 的区间 $[a_j, b_j]$.

输出: 多样性的初始群体.

Step1 FOR $j = 1$ TO n

Step2 $\Delta_j = (b_j - a_j) / N$ // 将决策变量 x_j 的区间均匀划分成 N 等份.

Step3 $A_j = \{ [a_j, a_j + \Delta_j], [a_j + \Delta_j, a_j + 2\Delta_j], \dots, [a_j + (N-2)\Delta_j, a_j + (N-1)\Delta_j], [a_j + (N-1)\Delta_j, b_j] \}$.

Step4 FOR $i = 1$ TO N

Step5 从集合 A_j 中随机选择一个子区间,并在该子区间内随机产生一个基因值赋给 x_j^i .

Step6 更新集合 A_j : 从集合 A_j 中删除在 Step5 中已选中的子区间.

Step7 END FOR

Step8 END FOR

Step9 输出多样性的初始种群 $\{x^1, x^2, \dots, x^N\}$.

这种初始化方法既保持了选取子区间以及从子区间中产生基因值的随机性,又保证了决策变量区间划分的均匀性,算法执行的结果可获得一组在决策空间中均匀分布的初始解点,这就为算法的搜索提供了一个良好的开端.

2.2 烟花爆炸半径精细化控制

基于烟花爆炸优化的方式如下:首先利用算法 1 产生规模为 N 的初始种群,即确定第一次爆炸的炸点位置,用以表征问题的初始解.例如,在 n 维搜索空间的第 i 个炸点可表示为 $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$,并设炸点执行均匀爆炸且预设的最大爆炸半径为 r ,即火星最大散开区域.若炸点爆炸的层数为 w ,那么每一层爆炸半径则为 $j \cdot r/w$ ($j = 1, 2, \dots, w$).已有 FEA 算法一般将爆炸半径 r 设置为随迭代次数的增加而呈非线性递减的方式,以保证算法在初期可以执行全局勘探,而在算法末期执行最优前沿附近的局部精确搜索.

需要指出的是,现有烟花爆炸半径的变化仅局限于对烟花的群体采用逐代递减的策略,而并未考虑到同一世代的烟花弹因其支配强度的差异而赋予不同的爆炸半径.例如,在复杂多目标优化环境下,对于相同世代的烟花弹,非支配个体相对具有较好的质量,它们更靠近 Pareto 前沿,而应在较小的邻域内进行求精;而那

些被支配的个体质量相对较差,它们一般离 Pareto 前沿较远,其应在较大的邻域空间内进行勘探.通过精细化分配有限的搜索资源,可改善算法寻优的效率.基于此,本文对 eMOFEOA 算法不同世代的烟花种群采用非线性递减的半径变化方式,而在同一代群体内则基于个体间 Pareto 支配强度的差异而被赋予不同的爆炸半径.

下面给出同一代烟花群 $pop(t)$ 中各烟花弹爆炸半径的计算方法:

(a) 基于 Pareto 关系计算第 t 代种群 $pop(t)$ 中个体 i 的强度值 $fitness^t(i)$:

$$fitness^t(i) = |\{j | i, j \in pop(t) : i < j\}| \quad (1)$$

这里的 $|\cdot|$ 表示集合的势,即个体 i 占优群体内其他个体的数目.

(b) 计算第 t 代种群 $pop(t)$ 的最大适应值 $fitness_{max}^t$:

$$fitness_{max}^t = \max_{i \in pop(t)} \{fitness(i)\} \quad (2)$$

(c) 计算第 t 代种群 $pop(t)$ 的爆炸半径 $r(t)$:

$$r(t) = k \cdot (D_{max} - D_{min}) \cdot \left(\frac{T_{max} - t}{T_{max}}\right)^\alpha + r_{end} \quad (3)$$

(d) 计算个体 $i \in pop(t)$ 的爆炸半径 $r_i(t)$:

$$r_i(t) = \begin{cases} r(t) \cdot \frac{fitness_{max}^t - fitness^t(i)}{fitness_{max}^t + \eta}, & \text{if } fitness^t(i) \neq fitness_{max}^t \\ r_{end} & \text{if } fitness^t(i) = fitness_{max}^t \end{cases} \quad (4)$$

步骤(c)中 $r(t)$ 通过缩放参数 k 将每一代烟花群体中各烟花弹的搜索范围控制在一个适当的范围内,其中 D_{max} 和 D_{min} 分别表示当代种群中所有个体各维决策变量中的最大值和最小值, α 表示爆炸半径的衰减指数.步骤(d)中引入最小正数 η 以防止分母为 0, r_{end} 为预设的最小爆炸半径.而对于炸点 i ,其爆炸产生的火星的方法如下:

$$x_i^* = x_i + r_j \cdot b_k \quad (5)$$

x_i 为炸点 i 的当前位置, x_i^* 为炸点 i 爆炸产生的火星位置.考虑到算法在实际执行过程中由于时空资源的限制,同时也为了保证有足够的火星产生,这里规定火星与炸点之间的距离只有 $r/4, r/2, 3r/4$ 和 r 四种情况,且 $r = r(t)$ 表示第 t 代爆炸点的半径,因此式(5)的爆炸半径 $r_1 = r, r_2 = 3r/4, r_3 = r/2$ 以及 $r_4 = r/4$. $b_k (k = 1, 2, \dots, m)$ 为爆炸的方向向量, m 为烟花弹 i 爆炸的总方向数.对于低维度的搜索空间(如 $n \leq 10$),算法选择直观的标准坐标轴爆炸方向,即 n 维搜索空间问题的爆炸方向数为 $2n$.但对于高维度的搜索空间(如 $n > 10$),若仍采用 $2n$ 个爆炸方向,则算法消耗的时空资源过大.因此,在高维搜索空间中,每层爆炸方向采用从 $2n$ 个标准坐标轴方向中随机选择 $n/3$ 个方向,再加上每个方向的

相反方向,构成 $2n/3$ 个爆炸方向.

烟花弹爆炸时炸点和火星可能会越界,需要对它们的位置边界进行限制.设 $x_i (i = 1, 2, \dots, n)$ 为 n 维搜索空间中任一维决策变量,且 $x_i \in [a_i, b_i]$,如果炸点 x 爆炸过程中产生的火星 x_j 跳出边界 $[a_i, b_i]$ 成为非可行解,则将 x_j 在第 i 维决策变量上的值 x_{ji} 按式(6)进行重置,以改善搜索的有效性.

$$x_{ji} = rand(a_i, b_i), \text{ if } x_{ji} < a_i \text{ or } x_{ji} > b_i \quad (6)$$

其中, $rand(\cdot)$ 是区间 $[a_i, b_i]$ 上均匀分布的随机数.

2.3 选择火星

eMOFEOA 算法均匀随机初始化种群之后,从第二代起将当前种群中每一个个体都视为一个炸点,这些炸点爆炸后将会产生大量的火星,这些火星与炸点一起进行评估以选择出较优的火星(炸点)参与下一次爆炸.算法 2 给出了从炸点和火星集合中选择较优个体参与下一代进化的过程.

算法 2 更新种群

输入:种群的规模 N ,当前代种群 $pop(t)$.

输出:下一代种群 $pop(t+1)$.

Step1 利用归一化排序方法^[13]构建当前种群的非支配集 NDS ,即 $NDS = non-dominated_sorting(pop(t))$.

Step2 IF ($|NDS| < N$)

Step3 从集合 $\{i | i \in pop(t) \setminus NDS\}$ 中选出 $(N - |NDS|)$ 个较好的个体并与非支配集 NDS 进行合并,以构造下一代种群 $pop(t+1)$.

Step4 ELSE

Step5 随机从 NDS 中选取 N 个个体组成下一代种群 $pop(t+1)$.

Step6 END

Step7 输出下一代种群 $pop(t+1)$.

需要指出的是,Step3 和 Step5 中较好个体选择的依据是个体适应度值进行的.

2.4 档案群体多样性维护

随着进化过程的推进,MOEA 算法种群中非支配解快速增多,而外部档案的规模受到计算资源的限制一般设置为固定大小,因此需要对档案进行修剪,并维持档案群体的多样性.迄今已发展了多种多样性保持方法,其中代表性的包括:PAES 算法中的自适应网格法^[14]、NSGA-II 中计算拥挤距离的方法^[3]、SPEA2 中的 k -最近邻方法^[5] 和 Deb 等提出的 ϵ -占优^[6] 等,它们中 k -最近邻方法表现尤为突出,该算法将个体的密度值定义为个体与它的第 k 个最邻近个体之间的距离,且取 $k = \sqrt{NP + N'}$,其中 NP 为种群规模, N' 为外部档案的最大容量.

SPEA2 算法在运行过程中,若档案规模已达到上

限,则档案中每加入一个新解就要从中移出一个个体,比如个体 A,个体 A 应满足:对于档案中所有的个体 B, $A <_d B$ 成立. 这里的 $A <_d B$ 定义为当且仅当对于所有的 $l=1, 2, \dots, N'$: $D_{lA} = D_{lB}$; 或者对于某个 $l, D_{lA} < D_{lB}$, 而对于任意的 $l' < l$: $D_{l'A} = D_{l'B}$. 这里的 $D_{lA} (D_{lB})$ 定义为个体 A (B) 与它的第 l 个最邻近个体的距离. 不难发现, 这种个体密度的估计过程较为复杂, 而且需要对所有的个体和所有的 $l=1, 2, \dots, N'$ 比较个体间 D_l 上的差异, 时间开销较大.

鉴于此, 这里使用一种简化的 k -最近邻方法保持档案群体的多样性. 简化方法中取 $k = \log(NP + N')$, 其个体密度的估计方法可定义如下: $A <_d B$ 当且仅当对于 $l=1, 2, \dots, k$: $D_{lA} = D_{lB}$; 或者对于某个 $l \leq k$: $D_{lA} < D_{lB}$, 而对任意的 $l' < l$: $D_{l'A} = D_{l'B}$. 由于简化后的 k 小于原方法中的 k , 因此简化的 k -最近邻方法在时间复杂度上会有所改善.

2.5 eMOFEOA 算法流程

在前面 2.1 至 2.4 小节的基础上, 算法 3 给出 eMOFEOA 算法的流程.

算法 3 eMOFEOA 算法流程

输入: 种群的规模 N , 外部档案规模 N' , 预设的最小爆炸半径 r_{end} 爆炸半径衰减指数 α , 最大迭代次数 T_{max} .

输出: 算法终止后外部档案个体.

Step1 利用算法 1 产生规模为 N 的初始种群, 并将种群的非支配解复制到外部档案中, 且设迭代器 $t = 0$.

Step2 WHILE ($t < T_{max}$)

Step3 FOR ($i = 1; i \leq N; i++$)

Step4 根据公式(1)至公式(6)产生个体 i 的火星.

Step5 END

Step6 执行算法 2 产生下一代种群.

Step7 利用简化的 k -最近邻方法维持外部档案.

Step8 $t = t + 1$

Step9 END WHILE

Step10 输出外部档案中的所有解.

假设待优化问题的目标数目为 m , 决策空间的维度为 n , 种群规模为 N , 外部档案最大容量为 N' , 算法最大迭代次数为 T_{max} , 则 eMOFEOA 算法的时间复杂度可分析如下:

算法第 1 步中生成多样性的初始种群的时间为 $O(nN)$, 将种群中非支配解复制到外部档案所需要的时间是 $O(mN^2)$, 因此算法第 1 步所消耗的时间为 $O(nN) + O(mN^2) = O(mN^2)$. 算法从第 2 步开始进入迭代循环过程, 在循环体内部存在烟花爆炸产生火星的过程(步骤 3 至步骤 5), 其中一个烟花产生火星所需要的时间复杂度是 $O(8n^2/3)$, 那么 N 个烟花产生火星所

需要的时间复杂度则为 $O(8Nn^2/3)$. 步骤 6 利用归一化排序方法构造非支配集更新群体所需要的时间为 $O(m(N + N') \log(N + N'))$. 步骤 7 利用简化的 k -最近邻方法维持外部档案的多样性所需的时间为 $O(m(NP + N')^2 \log(NP + N'))$. 步骤 8 所需的时间是 $O(1)$. 因此, 循环体内部的时间复杂度应为 $O(8Nn^2/3) + O(m(N + N') \log(N + N')) + O(m(NP + N')^2 \log(NP + N')) + O(1) = O(m(NP + N')^2 \log(NP + N'))$. 由于算法一共执行了 T_{max} 次迭代, 因此算法迭代部分的时间复杂度为 $O(T_{max} m(NP + N')^2 \log(NP + N'))$. 算法最后输出档案中全部的解个体所需要的时间是 $O(N')$. 综上分析, eMOFEOA 算法的时间复杂度应为 $O(T_{max} m(NP + N')^2 \log(NP + N'))$, 亦即, eMOFEOA 算法的复杂度取决于档案群体修剪方法的复杂度.

3 实验与结果分析

3.1 对等比较算法

为评估 eMOFEOA 算法的性能, 选取 5 个代表性多目标优化算法做为对等比较算法, 它们分别是: (1) Deb 等提出的改进型非劣分类遗传算法 NSGA-II^[3]; (2) Zitzler 等提出的改进的强度 Pareto 进化算法 SPEA2^[5]; (3) Coello C. 等提出的多目标粒子群优化算法 MOPSO^[6], 为区别通常的多目标粒子群优化算法的英文简写, 这里将 Coello C. 等的算法简记为 CMOPSO; (4) Sierra 等提出的混合型多目标粒子群算法 OMOPSO^[7]; (5) Zhang 等提出的基于分解的多目标进化算法 MOEA/D^[8].

3.2 多目标测试函数集

本文使用的多目标测试函数集由 ZDT 和 DTLZ 两个系列共计 12 个函数组成, 其中 ZDT 系列^[15]是 5 个 2-目标的测试问题, 由 ZDT1-ZDT4 以及 ZDT6 组成. ZDT5 函数是布尔函数, 需要进行二进制编码, 为保证实验条件的一致性, 本文省去了该测试函数. DTLZ 系列^[16]是由 DTLZ1-DTLZ7 组成的一个可以改变搜索变量的数量与优化目标个数的可扩展的多目标测试问题集, 本文选择 DTLZ 系列做为 3-目标优化测试问题实例. 实验中各测试实例的真实 Pareto 前沿上采样的点数分别是: ZDT1-ZDT3 采样点为 200; ZDT4、DTLZ2-DTLZ6 采样点为 1000; ZDT6、DTLZ1 和 DTLZ7 采样点为 500.

3.3 实验参数

参与对等比较的算法 NSGA-II、SPEA2、CMOPSO、OMOPSO 和 MOEA/D 算法的实验参数采用对应参考文献的设置. 为使实验结果准确、公平起见, eMOFEOA 算法中部分与对比算法相同的参数设置与对比算法相同的参数. 例如: 对于 2-目标优化问题的种群规模取值 100, 外部档案的最大容量为 100, 测试函数最大评估次

数为 50000 次;对于 3-目标优化问题的种群规模取值为 200,外部档案集的规模为 200,测试函数的最大评估次数为 200000 次,SBX 算子的分布指数 $\mu = 20$. 另外,对于 eMOFEOA 算法新引入的参数参照文献[17]的研究结果,取半径衰减指数 $\alpha = 3$,预设的最小爆炸半径 $r_{end} = 10^{-6}$.

为减少多目标进化算法中随机因素对算法性能分析的影响,本文对每一种对等比较算法在每一测试范例上均独立执行 30 次. 所有实验在 Think Pad X200 笔记本电脑上运行,电脑配置 5G 内存和 2.4GHz 双核 CPU,安装 Windows 7 X64 操作系统,算法运用 Java 编程实现,并利用 Matlab2010a 环境出图.

3.4 性能指标

为了评估算法获得的近似 Pareto 前沿的收敛性和多样性,这里采用反转世代距离(Inverted generation distance, IGD)^[8]来评估算法的性能. IGD 指标度量真实 Pareto 前沿到算法获得的近似 Pareto 前沿之间的距离,由于本文选取的多目标测试问题的真实 Pareto 前沿是已知的,通过对真实 Pareto 前沿进行均匀采样,计算这些采样点到近似 Pareto 解点之间的距离既反映了算法的收敛性又反映出算法获得解集的多样性. 一般地,IGD 指标值越低,表明算法获得的近似 Pareto 前沿的收敛性和多样性越好,越接近真实 Pareto 前沿. 设 P 为真实 Pareto 前沿的最优解集, A 为算法获得的近似 Pareto 最优解集,则 IGD 值可根据式(7)进行计算

$$IGD = \frac{1}{|P|} \sum_{i=1}^{|P|} Dist_i \quad (7)$$

其中, $Dist_i = \min_{j=1}^{|A|} \sqrt{\sum_{m=1}^M \left(\frac{f_m(p_i) - f_m(a_j)}{f_m^{\max} - f_m^{\min}} \right)^2}$ 为最小归一化欧式距离; f_m^{\max} 和 f_m^{\min} 分别表示 P 中在第 m 个目标上

的最大值和最小值, $m = 1, 2, \dots, M, M$ 为目标个数; $p_i \in P, i = 1, 2, \dots, |P|; a_j \in A, j = 1, 2, \dots, |A|$.

3.5 实验结果与分析

实验 1 通过比较 6 个对等多目标进化算法在 12 个多目标测试问题上的 IGD 指标来考察 eMOFEOA 算法的性能. 表 1 列出了 6 种对比算法在 12 个测试实例上的 IGD 性能结果,其中 IGD 的均值(mean)和标准差(std.)是同一算法在同一测试问题独立运行 30 次的统计结果;t-test 值是 eMOFEOA 算法与其他对比算法在同一测试问题上进行 t 检验时的结果;“+”、“=”和“-”分别表示本文的算法获得的 IGD 值在显著性水平为 5% 的双尾 t 检验中优于、等于和劣于对应劣的对比算法在对应行的测试问题上的显著性区分结果;“Score”表示本文算法显著优于对应列的对比算法在 12 个测试问题中的净胜得分,即得“+”与得“-”的测试问题个体之差(下文同). 例如, eMOFEOA 算法与 NSGA-II 算法进行性能比较时,其在 10 个测试问题上显著优于 NSGA-II 算法,而在 DTLZ5 问题上显著劣于 NSGA-II 算法,其在 ZDT2 问题上与 NSGA-II 算法无显著差别,因此 eMOFEOA 算法在 12 个测试问题上的净胜得分(列举在 NSGA-II)所在列的最后一行)为 $10 - 1 = 9$. 同时用粗体字突显所有对比算法在每一个测试问题中的最小 IGD 值.

表 1 的 CMOPSO 算法获得了 4 个最优的 IGD 均值, eMOFEOA 算法和 OMOPSO 算法各获得 3 个最优的 IGD 值, MOEA/D 获得 2 个,而 SPEA2 和 NSGA-II 无一能获得最优 IGD 值. 从表 1 的 t -检验结果来看,本文算法对比其他 5 种算法的净胜得分均大于 0,表明 eMOFEOA 算法在所有测试题上获得的 IGD 性能总体上要显著地优于其他 5 种对比算法.

表 1 6 种对比算法在 12 个测试问题上的 IGD 性能对比结果

函数		eMOFEOA	NSGA-II	SPEA2	CMOPSO	OMOPSO	MOEA/D
ZDT1	mean	1.45E-3	5.31E-3	5.94E-3	3.27E-3	5.68E-3	1.43E-3
	std.	1.21E-7	7.63E-8	1.65E-7	3.00E-6	2.77E-6	1.27E-7
	t-test		+	+	+	+	=
ZDT2	mean	5.19E-3	5.26E-3	1.61E-2	2.10E-2	5.29E-3	2.16E-2
	std.	4.71E-7	7.92E-8	8.52E-4	3.11E-4	7.65E-7	6.31E-3
	t-test		=	+	+	+	+
ZDT3	mean	4.16E-3	4.43E-3	2.59E-3	1.18E-3	8.69E-3	6.18E-3
	std.	1.19E-8	4.40E-5	2.47E-5	3.13E-5	1.11E-5	1.48E-5
	t-test		+	-	+	+	+

续表

函数		eMOFEOA	NSGA-II	SPEA2	CMOPSO	OMOPSO	MOEA/D
ZDT4	mean	5.46E-3	1.41E-2	5.99E-2	5.62E-3	7.89E-1	2.36E-3
	std.	6.17E-6	8.10E-5	2.68E-3	2.51E-5	1.75E-1	4.28E-4
	t-test		+	+	+	+	-
ZDT6	mean	1.22E-2	1.95E-2	3.75E-2	2.82E-2	3.83E-3	8.17E-3
	std.	4.20E-7	4.26E-6	2.70E-5	3.61E-5	1.06E-7	2.69E-6
	t-test		+	+	+	-	-
DTLZ1	mean	3.33E-2	4.22E-2	3.39E-2	9.26E-2	3.16E+1	7.61E-2
	std.	1.48E-5	6.05E-5	1.82E-5	2.27E-5	4.31E+1	1.20E-6
	t-test		+	+	+	+	+
DTLZ2	mean	1.36E-2	4.77E-2	3.82E-2	6.15E-2	5.14E-2	4.62E-2
	std.	8.92E-7	5.91E-6	9.96E-7	2.45E-7	5.96E-6	5.22E-6
	t-test		+	+	+	+	+
DTLZ3	mean	6.28E-2	4.33E-1	4.67E-1	3.54E-3	1.30E-1	6.58E-2
	std.	1.19E-2	3.13E-1	3.28E-1	2.97E-3	1.18E-1	1.16E-2
	t-test		+	+	-	+	=
DTLZ4	mean	2.31E-2	3.14E-2	3.23E-2	1.15E-2	1.08E-0	2.03E-2
	std.	3.82E-5	3.64E-5	3.80E-5	3.78E-5	1.25E-1	3.11E-5
	t-test		+	+	-	+	-
DTLZ5	mean	1.84E-3	1.11E-3	1.01E-3	7.12E-4	6.06E-4	4.26E-3
	std.	9.58E-6	3.56E-6	2.69E-6	2.62E-6	1.79E-6	5.98E-5
	t-test		-	-	-	-	+
DTLZ6	mean	2.36E-4	1.21E-2	1.10E-3	3.30E-4	5.39E-5	5.71E-4
	std.	1.14E-5	4.12E-4	4.84E-5	1.45E-5	1.48E-7	4.36E-5
	t-test		+	+	=	-	+
DTLZ7	mean	9.16E-4	7.52E-3	6.05E-3	8.17E-4	8.32E-4	1.93E-2
	std.	1.24E-5	2.76E-5	2.87E-5	1.02E-5	1.16E-57.63E-3	
	t-test		+	+	-	-	+
Better(+)			10	10	7	8	7
Same(=)			1	0	1	0	2
Worse(-)			1	2	4	4	3
Score			9	8	3	4	4

由于 IGD 指标能够同时反映算法的收敛性和多样性,而实验 1 的结果表明 eMOFEOA 算法总体上具有最好的 IGD 性能,这表明 eMOFEOA 算法的均匀与随机化初始化种群、精细化控制烟花爆炸半径以及简化的 k -最近邻策略有机地结合,增强了算法平衡全局勘探和局部开采的能力。

进一步地,通过分析对等比较算法在 12 个测试函数中的 IGD 均值排名和方差排名来评估算法的准确性和稳定性^[18]。由于每一种对比算法在所有测试问题上

都有一个 IGD 均值的排名,假设 $R_A = \{r_1, r_2, \dots, r_{|R|}\}$ 表示算法 A 在 $|R|$ 个测试实例中获得排名的集合,则 A 的准确性可通过计算 R_A 的排名平均值 $\mu_A = \frac{1}{|R|} \sum_{r \in R} r$ 而获得;相应地, A 的稳定性通过计算排名的方差 $\sigma_A = \frac{1}{|R|} \sum_{r \in R} (r - \mu_A)^2$ 得到。因此,若某个算法 A_1 的排名均值和方差都比另一个算法 A_2 的要小,则可认为算法 A_1 的性能要好于算法 A_2 的性能。图 1 给出了 6 种算法在所有 12 个测试实例上获得的 IGD 均值排名和方差排

名的结果.

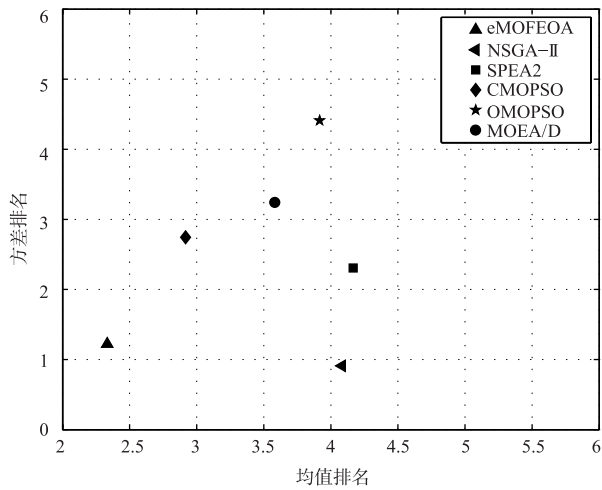


图1 6种算法在12个函数上的IGD均值和方差的排名

从图1中不难看出,eMOFEOA算法在IGD均值和方差两方面占优了SPEA2、CMOPSO、OMOPSO和MOEA/D四种对等算法.而与NSGA-II算法相互非劣,即NSGA-II在方差的排名上略优于本文算法,而本文算法在IGD均值排名上要明显地优于NSGA-II算法.图1表明了eMOFEOA算法在所有对等比较算法中具有相对最优的准确性和稳定性.

实验2:为进一步考察eMOFEOA算法的收敛速度,设计如下实验:利用2个较难的2-目标测试问题ZDT4、ZDT6和2个较难的3-目标测试函数DTLZ1、DTLZ4组成测试集合,考察eMOFEOA算法与其他5种对等比较算法在运行100代过程中其GD值的变化情况.由于GD指标^[19]度量了算法的近似解集与真实Pareto前沿的距离,即收敛性能,因此通过观察个算法近似解集的GD值随进化代数t的变化,可以判断出算法收敛速度的快慢.实验2中各算法分别运行100代,并每隔10代

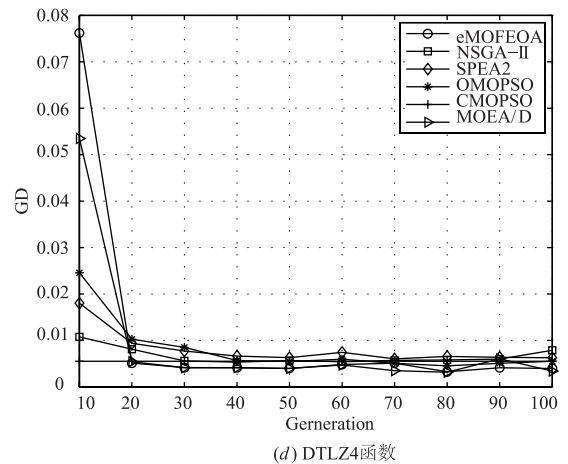
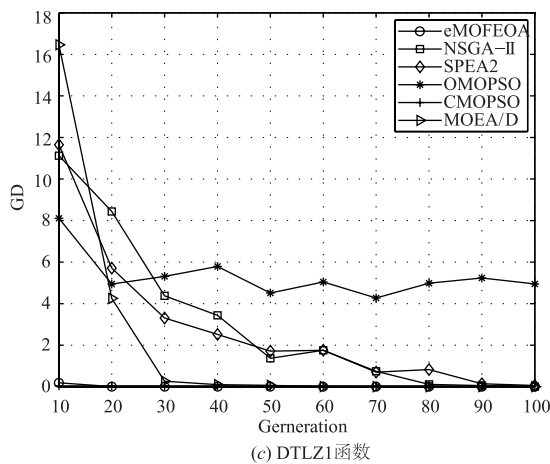
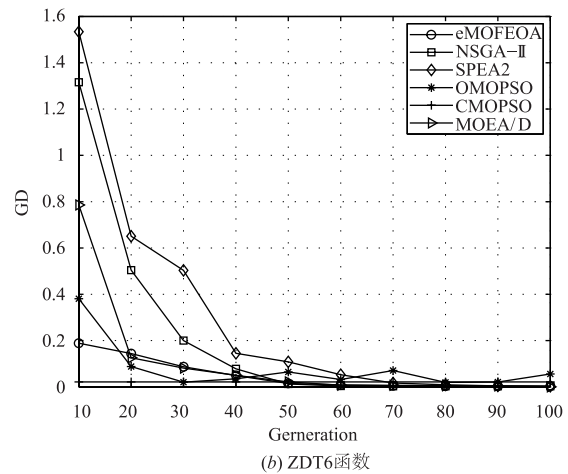
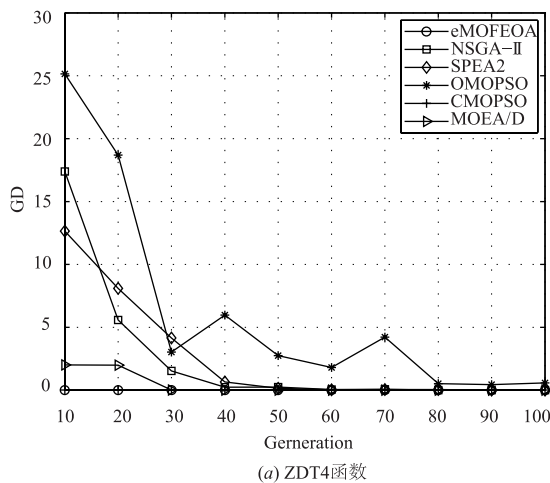


图2 6种算法在4个测试函数上的收敛速度曲线

计算其获得的GD值.从图2可知,对于ZDT4测试函数,本文的eMOFEOA算法和CMOPSO算法获得了最好

的收敛速度,其次是MOEA/D算法,随后是NSGA-II算法,表现最差的则属OMOPSO算法,其收敛过程出现了

起伏. 对于 ZDT6 问题, eMOFEOA 算法能够稳步收敛, 其收敛速度明显快于 NSGA-II、SPEA2 和 MOEA/D 算法, 而与 OMOPSO 算法相比, eMOFEOA 算法在前面 30 代收敛的速度不及 OMOPSO 算法快, 但在后面的进化过程中 eMOFEOA 算法能够稳定地收敛, 而 OMOPSO 算法则出现一定的波折. 在 DTLZ1 函数上, eMOFEOA 算法和 CMOPSO 算法同样获得了最好的收敛速度, 其次是 MOEA/D 算法, 随后是 SPEA2 算法, OMOPSO 算法的 GD 值在第 40 代和第 70 代出现不降反升的现象, 但最终在第 60 代趋于收敛状态. 在 DTLZ4 问题上, 各算法能以较快的速度收敛, 它们的差别主要体现在算法运行的前 40 代, CMOPSO 算法、eMOFEOA 算法、MOEA/D 算法和 OMOPSO 算法能较快收敛, 而 SPEA2 和 NSGA-II 算法收敛较慢, 但各算法在第 60 代均能趋于收敛状态. 由此可见, 本文的 eMOFEOA 算法较之其他 5 个对等比较算法能够获得较好的收敛速度, 表明了 eMOFEOA 算法具有较好的稳定的收敛性能.

4 结论

现实中复杂多目标优化问题不断涌现, 迫切需要发展新型多目标优化算法应对挑战. 本文提出一种增强型多目标烟花爆炸算法 eMOFEOA. 该算法均匀化与随机化相结合的方法产生初始化种群, 为算法随后的迭代过程提供较好的开端; 其次, eMOFEOA 算法对烟花爆炸半径实行了精细化控制策略, 不仅不同世代的种群具有不同的爆炸半径, 而且相同世代种群中的个体因其支配强度的差异也将具有不同的半径; 此外, 算法采用简化的 k-最近邻方法维持档案群体的多样性. 上述三种策略有机结合, 并在算法的不同阶段实施, 以增强算法解决复杂 MOP 问题的能力. 性能对比实验表明本文算法具有总体显著的性能优势.

未来将利用更多、更困难的测试问题以及若干工程实践中的 MOP 问题全面地评估 eMOFEOA 算法的性能, 并进一步将一些新近提出的进化范例引入多目标优化算法的框架中, 不断提高算法解决现实 MOP 问题的能力.

参考文献

- [1] 周爱民, 张青富, 张桂戌. 一种基于混合高斯模型的多目标进化算法. 软件学报, 2014, 25(5): 913-928.
Zhou AM, Zhang QF, Zhang GX. Multiobjective evolutionary algorithm based on mixture Gaussian models[J]. Journal of Software, 2014, 25(5): 913-928. (in Chinese)
- [2] Srinivas N, Deb K. Multiobjective optimization using non-dominated sorting in genetic algorithms[J]. Evolutionary Computation. 1994, 2(3): 221-248.
- [3] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [4] Zitzler E, Thiele L. Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257-27.
- [5] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[A]. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems[C]. Berlin: Springer-Verlag, 2002. 95-100.
- [6] Coello Coello C A, Pulido G T, Lechuga M S. Handling multiple objectives with particles swarm optimization[J]. IEEE Transactions on Evolutionary Computation. 2004, 8(3): 256-279.
- [7] Sierra M R, Coello Coello C A. Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance[A]. Proceedings of 3rd International Conference on Evolutionary Multi-criterion Optimization[C]. Berlin: Springer, 2005, 505-519.
- [8] Zhang QF, Li H. MOEA/D: A Multi-objective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [9] TAN Y, ZHU Y. Fireworks algorithms for optimization[A]. Proceedings of International Conference on Swarm Intelligence[C]. Piscataway: IEEE Press, 2010. 355-364.
- [10] Yu-Jun Zheng, Qin Song, Sheng-Yong Chen. Multi-objective fireworks optimization for variable-rate fertilization in oil crop production[J]. Applied Soft Computing, 2013, 13(11): 4253-4263.
- [11] 谢承旺, 许雷, 赵怀瑞等. 应用精英反向学习的多目标烟花爆炸算法[J]. 电子学报, 2016, 44(5): 1180-1189.
Xie Cheng-wang, XU Lei, ZHAO Hui-rui, et al. Multi-objective fireworks optimization algorithm using elite opposition-based learning[J]. Acta Electronic Sinica, 2016, 44(5): 1180-1189. (in Chinese)
- [12] 谢承旺, 邹秀芬, 夏学文等. 一种多策略融合的多目标粒子群优化算法[J]. 电子学报, 2015, 44(5): 1538-1544.
Xie Cheng-wang, Zou Xiu-fen, Xia Xue-wen, et al. A multi-objective particle swarm optimization algorithm integrating multiply strategies[J]. Acta Electronic Sinica, 2015, 43(8): 1538-1544. (in Chinese)
- [13] 鲍培明, 朱庆保. 用于多目标进化的归一化排序非支配

- 集构造方法[J]. 电子学报,2009,37(9): 2010-2015.
BAO Pei-ming, ZHU Qing-bao. A technique of building non-dominated set based on normalized sort in evolutionary multi-objective optimization[J]. Acta Electronic Sinica,2009,37(9): 2010-2015. (in Chinese)
- [14] Knowles J D, Corne D W. Approximating the non-dominated front using the Pareto archived evolution strategy [J]. Evolutionary Computation,2000,8(2): 149-172.
- [15] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: Empirical results[J]. Evolutionary Computation,2000,8,173-195.
- [16] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems [A]. Proceeding of the IEEE Congress on Evolutionary Computation[C]. Piscataway: IEEE Service Center,2002. 825-830.
- [17] 曹炬,贾红,李婷婷. 烟花爆炸优化算法[J]. 计算机工程与科学,2011,33(1):138-142.
CAO Ju, JIA Hong, LI Ting-ting. A fireworks explosion optimization algorithm[J]. Computer Engineering & Science,2011,33(1): 138-142. (in Chinese)
- [18] 胡旺, Yen GG, 张鑫. 基于 Pareto 熵的多目标粒子群优化算法[J]. 软件学报,2014,25(5):1025-1050.
Hu W, Yen GG, Zhang X. Multiobjective particle swarm optimization based on Pareto entropy[J]. Journal of Software,2014,25(5):1025-1050. (in Chinese)
- [19] Van Veldhuizen D A, Lamont G B. On measuring multi-objective evolutionary algorithm performance [A]. Congress on Evolutionary Computation [C]. Piscataway: IEEE Press,2000. 204-211.

作者简介



谢承旺 男,1974年10月出生,湖北武汉人,副教授、硕士生导师、中国计算机学会高级会员. 2005年、2010年分别在武汉理工大学、武汉大学获得计算机软件与理论专业工学硕士和工学博士学位,2015年从武汉大学数学博士后流动站出站. 现主要从事智能计算与智能信息处理方面的研究工作.
E-mail: chengwangxie@163.com



许雷 男,1989年1月出生,湖北咸宁人,硕士研究生. 主要从事智能计算方面的研究.
E-mail: 13077911256@163.com