

基于多样性检测的双子群多目标粒子群算法

韩 敏[†], 张丽君

(大连理工大学 电子信息与电气工程学院, 辽宁 大连 116023)

摘 要: 为了平衡多目标粒子群算法的多样性和收敛性, 提出一种基于多样性检测的多子群多目标粒子群算法. 首先, 将多样性检测方法引入到多目标粒子群算法中, 并结合多目标粒子群算法的特点进行改进. 然后, 将种群分为两个不同分工的子群, 一个子群保持较好的多样性, 在搜索空间进行全局搜索; 另一个子群保持较好的收敛性, 在 Pareto 前沿附近进行局部搜索. 最后, 根据多样性度量指标调整两个子群的搜索行为, 以达到兼顾多样性和收敛性的目的. 在标准测试问题上的仿真结果表明了所提算法的有效性.

关键词: 多样性; 子群; 自适应; 多目标优化; 粒子群优化

中图分类号: TP18 **文献标志码:** A

Bi-group multi-objective particle swarm optimization algorithm based on diversity metric

HAN Min[†], ZHANG Li-jun

(Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116023, China)

Abstract: In order to keep the balance between the diversity and convergence, a bi-group multi-objective particle swarm optimization algorithm based on diversity metric is propose. Firstly, a diversity metric is introduced to multi-objective particle swarm optimization(MOPSO) algorithm and improved based on its characteristics. Then, the whole swarm is divided to two bi-groups with different searching tasks. One of the groups keeps population's diversity during evolution to search better in the whole search space. The other group keeps its convergence to local search nearby the Pareto front. Further more, the searching behavior of the groups based on the diversity metric is adjusted to balance the diversity and convergence. The simulations on several standard test functions verify the effectiveness of the proposed method.

Keywords: diversity; bi-group; adaptive adjustment; multi-objective optimization; particle swarm optimization

0 引 言

在现实生活中,许多实际工程和科学研究问题都可以归结为多目标优化问题,其特点是各个优化目标之间存在冲突,最优解不是唯一的,而是一组相互之间无法比较的 Pareto 最优解集. 粒子群优化算法^[1]是一种可并行处理多个目标函数的智能优化算法,控制参数少,收敛速度快,适合于求解多目标优化问题,得到了各学者的广泛关注.

通常,评价多目标优化算法所得解集的好坏包含收敛性和多样性两个方面^[2]. 收敛性是指所得解集离真实 Pareto 解集距离尽可能小,但过度强调收敛又容易使得种群陷入局部最优,且粒子群优化算法本身具有快速收敛的特性,很容易使得粒子聚集,陷入局部最优. 因此,需要保持种群具有良好的多样性以避免

该问题,但过度增强种群多样性又会造成收敛速度缓慢的问题,影响最终所得解集的收敛性. 可见,收敛性与多样性之间存在冲突^[3]. 为了协调该冲突,一些学者从种群角度出发,提出多子群协同搜索策略^[4-5]和动态改变种群大小的策略^[6];另一些学者从调整粒子群运动的角度出发,提出自适应调整参数的策略^[5]和更改粒子运动方式等^[7];还有一部分学者采用变异^[8]的方法,在保持种群收敛性的同时防止粒子过度聚集,从而避免陷入局部最优. 以上的多目标粒子群优化算法虽然能在一定程度上平衡收敛性和多样性,但还缺乏一种获取种群搜索状态的反馈机制来自适应调整其收敛性和多样性的平衡性,存在一定的盲目性.

为了合理地平衡多样性与收敛性的冲突,有必要

收稿日期: 2016-11-01; 修回日期: 2017-01-17.

基金项目: 国家自然科学基金项目(61374154).

作者简介: 韩敏(1959—),女,教授,博士生导师,从事复杂工业系统建模与控制、智能技术及优化算法等研究; 张丽君(1993—),女,硕士生,从事智能优化算法的研究.

[†]通讯作者. E-mail: minhan@dlut.edu.cn

动态获取当前种群的搜索状态,并根据当前的搜索状态来决定该采取何种搜索行为^[9-10].因此,本文的目的在于利用种群当前所处的搜索状态,自适应地平衡种群的收敛性和多样性,最大化种群搜索效率.但在多目标优化中,Pareto最优解集互不支配的特点使得检测种群搜索状态比单目标优化更加具有挑战性.针对该问题,本文首先引入基于目标空间的多样性检测方法^[11],并根据多目标粒子群优化算法的特点对其进行改进.然后,采取分群的策略,将种群分成两个分工不同的子群进行协同搜索.其中一个子群保持良好的多样性,在整个搜索空间进行全局搜索;另一个子群保持良好的收敛性,在Pareto前沿附近进行局部搜索,以平衡种群的多样性和收敛性,在维持算法向真实Pareto最优前沿推进的同时,避免陷入局部最优.两个子群的大小随着算法运行而动态改变.

为了验证所提算法的有效性,本文采用多个多目标优化算法进行仿真对比实验,采用较常用的ZDT和DTLZ系列多目标测试问题进行测试,实验结果表明所提算法具有一定的优势.

1 基于多样性检测的自适应多目标粒子群算法

在本文算法中,首先引入多样性检测方法并进行改进,然后将种群分成两个子群,一个子群保持较好的多样性,进行全局搜索,另一个子群向真实Pareto前沿收敛,进行局部搜索,并根据迭代次数动态调整子群大小.以下进行详细描述.

1.1 多样性检测方法及其改进

Adra等^[11]曾提出一种针对进化算法的种群多样性检测方法.该方法通过对比种群中个体的目标值和真实的Pareto前沿目标值来判断其多样性状态,并依据多样性状态调整其搜索行为来维持种群的多样性.在该方法中,种群多样性度量指标的计算方法如下:

$$I_s = D / \left[\sum_{m=1}^M (\max_{z_* \in Z_*} \{z_{*m}\} - \min_{z_* \in Z_*} \{z_{*m}\})^2 \right]^{1/2}. \quad (1)$$

其中: M 代表目标个数; Z_* 代表真实解集; z_* 代表真实解; z_{*m} 代表 z_* 在第 m 个目标上的值; D 代表种群的多样性,其计算方法为

$$D = \left[\sum_{m=1}^M (\max_{z_A \in Z_A} \{z_{Am}\} - \min_{z_A \in Z_A} \{z_{Am}\})^2 \right]^{1/2}. \quad (2)$$

Z_A 代表种群, z_A 代表种群中的粒子, z_{Am} 代表 z_A 在第 m 个目标上的值.

I_s 的取值范围是 $[0, \infty)$,当 $I_s = 1$ 时,说明种群的分布与真实解集的分布一致,达到理想状态;当 $I_s <$

1时,说明种群的多样性较差,可能陷入局部最优;当 $I_s > 1$ 时,说明种群的多样性较好,但如果 I_s 值太大则容易导致种群难以收敛.

该度量指标还存在一些不足之处.首先,该指标值的计算需要预先得到的真实Pareto前沿目标值通常难以获得.因为多目标粒子群算法中维持了一个用于存储所得解集的外部档案,所以本文采用该外部档案的解集代替 I_s 计算方法中的真实解集,即式(1)中的 z_{*m} 表示外部档案解集 A 中解集在第 m 个目标上的值.通过将真实解集改为外部档案解集,充分利用了外部档案解集的信息,得出当前种群相对于已找到解集的多样性度量值,且不需要预先知道真实解集的信息.其次,式(1)和(2)中均采用种群中第 m 个目标上的极差作为多样性的度量,容易受到极值的影响.当种群中存在个别比较分散的个体时,度量指标的准确性大大降低.针对该问题,本文将其替换为四分位差,本文 I_s 的计算方式如下:

$$I_s = D / \left[\sum_{m=1}^M (Q_3(z_{*m}) - Q_1(z_{*m}))^2 \right]^{1/2}. \quad (3)$$

其中

$$D = \left[\sum_{m=1}^M (Q_3(z_{Am}) - Q_1(z_{Am}))^2 \right]^{1/2}, \quad (4)$$

Q_1 和 Q_3 分别为下四分位数值和上四分位数值.

1.2 子群搜索策略

为了解决种群多样性与收敛性的冲突,本文将种群分成两个子群进行协同搜索.两个子群具有不同的分工,其中一个子群维持其良好的多样性,进行全局搜索,称为多样性子群;另一个子群保持其良好的收敛性,进行局部搜索,称为收敛性子群.此外,在算法的不同运行时期,调整多样性子群和收敛性子群的规模.

在多样性子群中,首先,各参数在较大的范围内随机变化,增加其搜索的随机性,更好地维持其多样性.设置惯性权重 ω 在 $[0.1, 0.729]$ 之间,学习因子 c_1 、 c_2 在 $[0.5, 2.5]$ 内随机变化.然后,多样性子群采用环形拓扑结构,在种群中选择全局最优,维持多样性.即通过比较粒子相邻的两个粒子,选择其中的非支配解作为其全局最优,如果两个邻居都是非支配解,则随机选择其中一个作为全局最优.最后,当多样性子群的 I_s 小于一定的阈值 σ (本文设置 $\sigma = 2$)时,重新初始化多样性子群的位置和速度,使其重新获得探索整个搜索空间的能力.

在收敛性子群中,首先,各参数随着多样性指标 I_s 的值进行变化以控制其收敛能力.当 I_s 逐渐增大

时,说明种群多样性增大,种群中粒子比较分散,因此减小惯性权重 ω 和学习因子 c_1 ,增大学习因子 c_2 ,使得种群朝最优 Pareto 前沿移动,增加搜索精度;当 I_s 逐渐减小时,各参数变化正好相反,适当扩大搜索范围. 假设 ω 随 I_s 变化如下:

$$\omega = a \times e^{-I_s} + b. \quad (5)$$

当 I_s 从0增大到无穷时, ω 从0.729递减到0.1,以保证收敛. 因此,当 $I_s = 0$ 时, $\omega = 0.729$;当 $I_s = \infty$ 时, $\omega = 0.1$. 代入式(5),得到 $a = 0.629, b = 0.1$,即

$$\omega = 0.629 \times e^{-I_s} + 0.1. \quad (6)$$

同理,设置当 I_s 从0增大到无穷时, c_1 从2.5递减到0.5, c_2 从0.5递增到2.5,保证收敛性,即

$$c_1 = 2 \times e^{-I_s} + 0.5, \quad (7)$$

$$c_2 = -2 \times e^{-I_s} + 2.5. \quad (8)$$

其次,收敛性子群从外部档案解集中选择全局最优. Deb等^[2]曾提出拥挤距离的概念,能够快速判断外部档案非支配解的密集程度. 本文采用拥挤距离来选择外部档案分布较稀疏的非支配解作为粒子的全局最优,以保持解集的多样性. 即首先在档案中任意选择两个解,其中拥挤距离较大者作为粒子的全局向导.

在算法运行过程中,多样性子群和收敛性子群所找到的非支配解均加入到外部档案解集中,并采用拥挤距离策略来维护外部档案解集的大小. 即当外部档案解集的非支配解数量超出预定大小时,将拥挤距离最小的非支配解从外部档案中移除,以保持外部档案的多样性.

以上多样性子群和收敛性子群的规模随着迭代次数的增加而变化. 当算法在运行初期时,需要增强探索能力,因此设置多样性子群数量较大;而在算法运行后期时,需要增强解的开发能力,因此设置收敛性子群的数量较大. 两个子群的数量变化如下所示:

$$N_{\text{diversity}} = \left\lfloor N \cdot \left[\rho \cdot \frac{k}{G_{\text{max}}} \right] / \rho \right\rfloor, \quad (9)$$

$$N_{\text{converge}} = N - N_{\text{diversity}}. \quad (10)$$

其中: $N_{\text{diversity}}$ 和 N_{converge} 分别为多样性子群和收敛性子群的大小; $\lfloor \cdot \rfloor$ 为向下取整; N 为种群大小; k 为当前迭代次数; G_{max} 为最大迭代次数; ρ 参数在本文中设置为10,使得算法只在迭代次数每增加10%时改变子群的规模,避免子群规模频繁变化.

1.3 算法步骤

将所提算法记为BMOPSO,其流程如下所示:

Step 1: 设定种群为 $P^{(0)}$,种群大小为 N ,最大迭代次数为 G_{max} ;初始化所有粒子的个体最优 p_i 为当前位置 x_i ;设定初始外部档案解集 $A^{(0)} = \emptyset$,最大容量为 \bar{A} ;令迭代次数 $k = 0$.

Step 2: 对 $P^{(0)}$ 中的个体进行非支配比较,将非支配解加入到 $A^{(0)}$ 中,形成 $A^{(1)}$.

Step 3: 令 $k := k + 1$,开始循环迭代.

Step 4: 根据式(3)和(4)计算多样性指标 I_s ,分别计算多样性子群和收敛性子群各自的参数 ω 、 c_1 、 c_2 ,并选择全局最优,然后更新位置,形成 $P^{(k)}$.

Step 5: 将多样性子群和收敛性子群产生的个体与 $A^{(k)}$ 中的解进行Pareto支配比较,如果不被 $A^{(k)}$ 中的解支配,则加入到 $A^{(k)}$ 中. 判断 $A^{(k)}$ 中解的数量是否大于 \bar{A} ,如果是,则删除拥挤距离最小的多余解.

Step 6: 若不满足算法终止条件,则返回Step3继续迭代;否则结束程序,输出 $A^{(k)}$ 中的非支配解.

2 仿真实验

为了验证所提算法的有效性,与多个多目标优化算法在常用的标准测试问题集中进行求解,然后比较所得解集的收敛性和多样性. 对比算法分别是MOPSO^[8]、OMOPSO^[12]、TV-MOPSO^[13]、MOPSO-CD^[14]、NSGA-II^[2]和MOEA/D^[15]. 各算法的种群大小和档案最大容量均设置为100,MOEA/D的权重向量个数也设置为100,每个测试问题的最大评价次数为25000次. 其他对比算法的参数设置均采用了原文献的推荐设置. ZDT系列测试问题^[3]集包含ZDT1~ZDT6共6个二目标测试问题. 因为ZDT5是二进制测试问题,需要二进制编码,所以本次仿真省去了ZDT5的测试. DTLZ测试问题集^[3]包含DTLZ1~DTLZ7共7个测试问题,每个问题的变量维数和目标个数均可以自由设定,本节仿真中的DTLZ问题均为三目标的测试问题.

使用反转世代距离(IGD)指标^[3]评价算法所得解集的收敛性和多样性. IGD指标能够同时评价所得解集的收敛性和多样性,其值越小,所得解集的收敛性和多样性越好,算法的优化效果越好.

表1列出了5个对比算法在12个测试问题上的IGD平均值和方差,每个结果均为同一算法在同一测试问题上独立运行30次的统计结果,采用粗体字表示所有对比算法在对应行测试问题中的最小评价指标值.“+”、“-”和“=”分别表示在显著性水平为5%的Mann-Whitney统计检验下,BMOPSO所得的指标值分别优于、劣于和等于对应列算法的指标值.

表1 7种对比算法在标准测试问题上的IGD指标对比结果

测试问题	BMOPSO	MOPSO ^[8]	OMOPSO ^[12]	TV-MOPSO ^[13]	MOPSO-CD ^[14]	NSGA-II ^[2]	MOEA/D ^[15]
ZDT1	1.351e-4(8.7e-7)	1.007e-3(1.3e-4) +	1.376e-4(2.1e-6) +	1.495e-4(6.3e-6) +	1.821e-4(7.2e-6) +	1.893e-4(6.8e-6) +	6.451e-4(1.9e-4) +
ZDT2	1.407e-4(1.4e-6)	9.115e-4(1.1e-4) +	1.427e-4(2.7e-6) +	1.574e-4(9.6e-6) +	1.784e-4(5.0e-6) +	1.910e-4(7.8e-6) +	4.838e-4(1.9e-4) +
ZDT3	1.055e-4(4.0e-6)	1.213e-3(2.3e-4) +	1.113e-4(4.7e-6) +	1.131e-4(7.2e-6) +	1.372e-4(1.5e-5) +	2.928e-4(6.0e-4) +	2.096e-3(6.9e-4) +
ZDT4	3.485e-2(1.9e-2)	4.095e-1(1.6e-1) +	2.211e-1(1.1e-1) +	1.811e-1(9.8e-2) +	3.946e-1(1.7e-1) +	2.954e-4(3.2e-4) -	1.371e-2(8.6e-3) -
ZDT6	1.356e-4(1.9e-6)	8.788e-4(8.5e-4) +	1.365e-4(2.3e-6) +	1.428e-4(4.5e-6) +	2.307e-4(8.2e-5) +	3.239e-4(2.7e-5) +	1.493e-4(6.9e-5) +
DTLZ1	3.831e-2(1.8e-2)	2.754e-2(1.1e-2) -	4.387e-2(1.8e-2) =	3.682e-2(1.5e-2) =	5.139e-2(2.1e-2) +	1.312e-3(1.7e-3) -	9.500e-4(1.1e-3) -
DTLZ2	7.274e-4(2.5e-5)	6.501e-3(4.3e-4) +	7.476e-4(3.2e-5) +	7.661e-4(3.1e-5) +	7.330e-4(2.9e-5) =	7.795e-4(4.7e-5) +	7.476e-4(4.1e-6) +
DTLZ3	8.717e-2(3.2e-2)	7.383e-2(2.6e-2) =	8.086e-2(3.3e-2) =	9.033e-2(3.2e-2) =	1.122e-1(4.2e-2) +	2.722e-3(3.8e-3) -	1.992e-3(4.5e-3) -
DTLZ4	1.137e-3(2.0e-4)	1.308e-3(1.8e-4) +	1.208e-3(1.7e-4) =	1.177e-3(1.3e-4) =	9.432e-4(8.2e-5) -	1.220e-3(1.2e-4) +	1.575e-3(2.4e-4) +
DTLZ5	6.062e-5(3.1e-6)	1.093e-4(8.0e-6) +	6.440e-5(2.6e-6) +	6.525e-5(4.1e-6) +	9.060e-5(8.1e-6) +	8.140e-5(4.2e-6) +	1.870e-4(9.6e-7) +
DTLZ6	5.652e-5(3.9e-6)	1.843e-2(5.9e-3) +	5.779e-5(2.9e-6) =	5.817e-5(3.3e-6) =	6.491e-5(5.8e-6) +	6.400e-3(6.6e-4) +	1.901e-4(2.8e-7) +
DTLZ7	6.909e-4(6.2e-5)	5.391e-3(1.3e-3) +	7.113e-4(8.5e-5) =	6.794e-4(5.5e-5) =	7.368e-4(6.1e-5) +	6.088e-4(3.8e-5) -	1.862e-3(9.9e-4) +

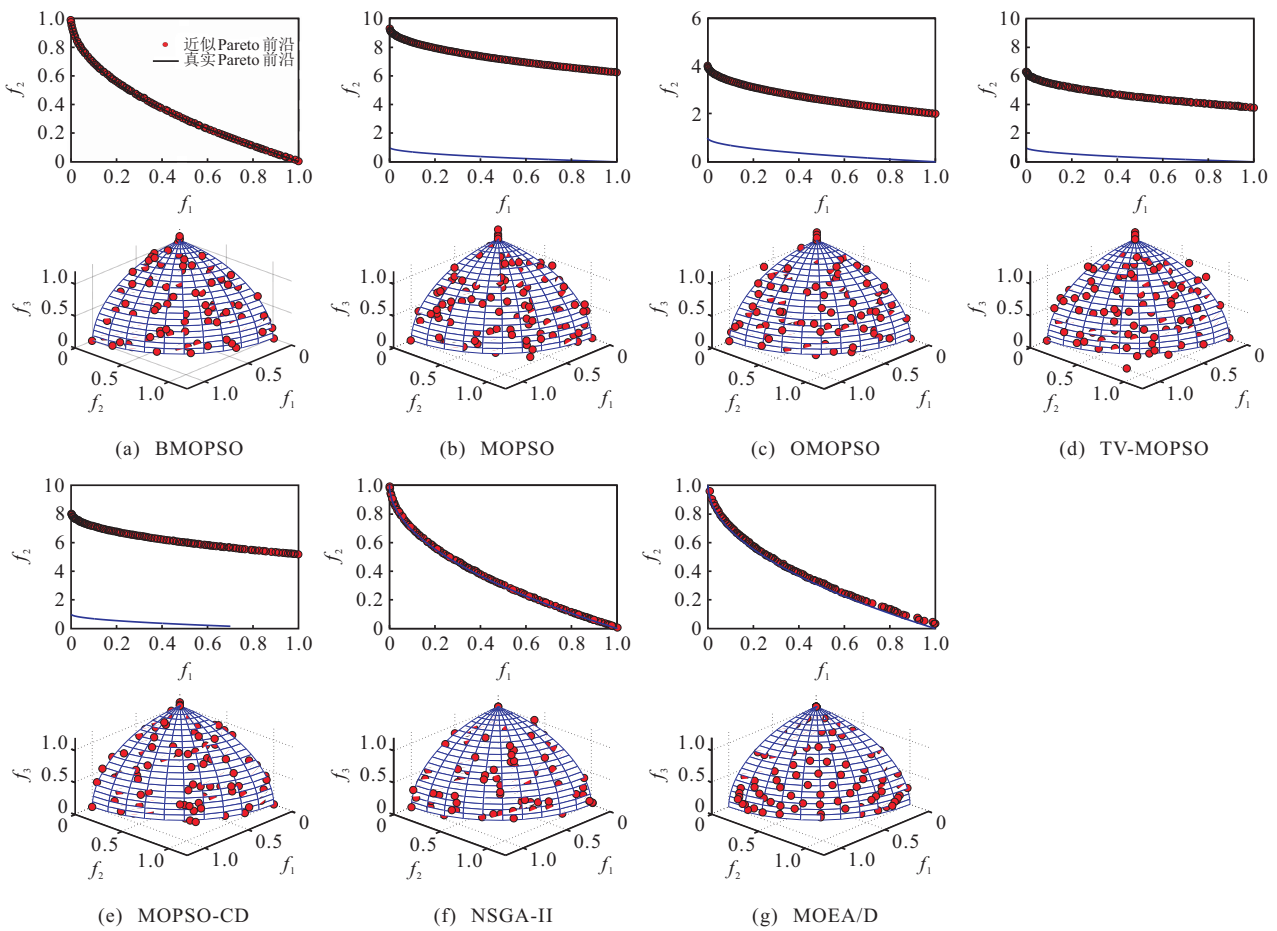


图1 7种对比算法在ZDT4和DTLZ2上的Pareto前端比较

从表1可以看出,相比其他对比算法,所提算法BMOPSO在12个测试函数上取得了7个最好的IGD值(其中5个测试问题的IGD值显著性好于其他算法),2个并列最优,1个次优值.说明所提算法在IGD指标上具有一定的优势.这是因为所提算法设置的分群策略,多样性子群专注于在整个搜索空间进行全局搜索,减少了陷入局部最优的情况;而收敛性子群专注于在Pareto前沿附近搜索,当其种群多样性较大时,根据收敛性度量指标动态调节粒子运动参数,

使得种群精确搜索的能力增强,且随着算法的运行后期,增大了收敛性子群的数量,更进一步增强了算法的后期收敛能力.而在解集多样性方面,一方面因为多样性子群进行全局搜索,有利于覆盖真实的Pareto前沿区域,发现更广区域的解;另一方面,收敛性子群的局部搜索能力有助于算法得到分布性更加均匀的解,使得算法所得解集的多样性更好.

此外,为了直观展示各个算法的收敛性和均匀性,图1给出了7个对比算法在二目标多模态问题

ZDT4上和三目标DTLZ2问题的近似Pareto前沿(所有图例均与图1(a)相同).可以看出,其他多目标粒子群算法(如MOPSO、OMOPSO、TV-MOPSO、MOPSO-CD)虽能得到较好的解集多样性,但收敛性不足,而本文所提算法在收敛到真实Pareto前沿附近的同时又具有较好的多样性.同样在DTLZ4问题上,其他多目标粒子群优化算法也有收敛性精度不足的缺点,NSGA-II和MOEA/D虽然收敛性较好,但综合收敛性和多样性来看,所提算法更好,得到的IGD值更小.

在各算法的计算复杂度比较方面,首先分析本文算法的计算复杂度.假设多目标优化问题含有 m 个目标,算法的种群规模为 N ,档案最大容量为 \bar{A} ,则对种群进行函数评价复杂度为 $O(mN)$,计算拥挤距离的复杂度为 $O(m\bar{A}\log\bar{A})$,种群和档案进行Pareto支配比较的计算复杂度为 $O(mN(N+\bar{A}))$.在本次仿真实验中,档案最大容量等于种群数量,因此BMOPSO算法的计算复杂度为 $O(mN^2)$;MOEA/D的计算复杂度为 $O(mNT)$, T 为权重向量的个数,仿真中 $T=N$,所以其计算复杂度为 $O(mN^2)$;其他对比算法的计算复杂度均为 $O(mN^2)$,因此各个对比算法的计算复杂度一致.

从以上分析可以看出:所提算法在基于多样性度量指标下设置分群策略,在搜索资源相同的情况下,合理控制了探索和开发过程;在保持种群多样性、探索整个搜索空间、避免陷入局部最优的前提下,对Pareto解集区域进行充分的开发,获得较好的IGD值,有效平衡了收敛性和多样性.

3 结论

本文提出了一种能够有效平衡收敛性和多样性的多目标粒子群算法,引入多样性检测方法并进行了改进;将种群分割成两个不同分工的子群,分别进行全局搜索和局部搜索,同时保证了多样性和收敛性.与其他典型多目标粒子群算法在ZDT和DTLZ系列测试问题上进行仿真对比的结果表明,本文算法能够较好地平衡收敛性和多样性,所得非支配解集的收敛性和分布性均较好.

参考文献(References)

[1] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.

[2] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.

[3] Hu W, Yen G G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system[J].

IEEE Trans on Evolutionary Computation, 2015, 19(1): 1-18.

- [4] Daneshyari M, Yen G G. Constrained multiple-swarm particle swarm optimization within a cultural framework[J]. IEEE Trans on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2012, 42(2): 475-490.
- [5] Li C, Yang S, Yang M. An adaptive multi-swarm optimizer for dynamic optimization problems[J]. Evolutionary Computation, 2014, 22(4): 559-594.
- [6] Leong W F, Yen G G. PSO-based multiobjective optimization with dynamic population size and adaptive local archives[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B, 2008, 38(5): 1270-1293.
- [7] Li Y, Zhan Z H, Lin S, et al. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems[J]. Information Sciences, 2015, 293: 370-382.
- [8] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.
- [9] 申晓宁, 郭毓, 陈庆伟, 等. 一种子群体个数动态变化的多目标优化协同进化算法[J]. 控制与决策, 2007, 22(9): 1011-1016.
(Shen X N, Guo Y, Chen Q W, et al. A multi-objective optimization co-evolutionary algorithm with dynamically varying number of subpopulations[J]. Control and Decision, 2007, 22(9): 1011-1016.)
- [10] 夏立荣, 李润学, 刘启玉, 等. 基于动态层次分析的自适应多目标粒子群优化算法及其应用[J]. 控制与决策, 2015, 30(2): 215-221.
(Xia L R, Li R X, Liu Q Y, et al. An adaptive multi-objective particle swarm optimization algorithm based on dynamic AHP and its application[J]. Control and Decision, 2015, 30(2): 215-221.)
- [11] Adra S F, Fleming P J. Diversity management in evolutionary many-objective optimization[J]. IEEE Trans on Evolutionary Computation, 2011, 15(2): 183-195.
- [12] Sierra M R, Coello C A C. Improving PSO-based multi-objective optimization using crowding, mutation and dominance[C]. Proc of Evolutionary Multi-Criterion Optimization. Guanajuato, 2005: 505-519.
- [13] Tripathi P K, Bandyopadhyay S, Pal S K. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients[J]. Information Sciences, 2007, 177(22): 5033-5049.
- [14] Raquel C R, Naval P C. An effective use of crowding distance in multiobjective particle swarm optimization[C]. Proc of Genetic and Evolutionary Computation Conf. New York, 2005: 257-264.
- [15] Qingfu Z, Hui L. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2009, 13(2): 284-302.