

基于分解和差分进化的多目标粒子群优化算法

李飞^{1†}, 刘建昌¹, 石怀涛³, 傅梓瑛²

(1. 东北大学 信息科学与工程学院, 沈阳 110004; 2. 东北大学 基建管理处, 沈阳 110004;
3. 沈阳建筑大学 机械工程学院, 沈阳 110168)

摘要: 为了提高多目标优化算法解集的分布性和收敛性, 提出一种基于分解和差分进化的多目标粒子群优化算法 (dMOPSO-DE). 该算法通过提出方向角产生一组均匀的方向向量, 确保粒子分布的均匀性; 引入隐式精英保持策略和差分进化修正机制选择全局最优粒子, 避免种群陷入局部最优 Pareto 前沿; 采用粒子重置策略保证群体的多样性. 与非支配排序 (NSGA-II) 算法、多目标粒子群优化 (MOPSO) 算法、分解多目标粒子群优化 (dMOPSO) 算法和分解多目标进化-差分进化 (MOEA/D-DE) 算法进行比较, 实验结果表明, 所提出算法在求解多目标优化问题时具有良好的收敛性和多样性.

关键词: 分解; 差分进化算法; 多目标优化; 粒子群优化算法; 方向角

中图分类号: TP273

文献标志码: A

Multi-objective particle swarm optimization algorithm based on decomposition and differential evolution

LI Fei^{1†}, LIU Jian-chang¹, SHI Huai-tao³, FU Zi-ying²

(1. College of Information Science and Engineering, Northeastern University, Shenyang 110004, China; 2. Infrastructure Management Department, Northeastern University, Shenyang 110004, China; 3. College of Mechanical Engineering, Shenyang Jianzhu University, Shenyang 110168, China)

Abstract: In order to improve the convergence and diversity of the Pareto optimal set in multi-objective optimization algorithms, a multi-objective particle swarm optimization algorithm based on decomposition and differential evolution (dMOPSO-DE) is proposed, in which the direction angle is presented to generate a set of direction vectors for maintaining the uniform distribution of the swarm. To avoid getting trapped into a local Pareto optimal front, decomposition-based strategy and differential evolution operator are used to generate the global best leader. Moreover, particle memory re-initialization is applied to enhance the diversity of the swarm. The preliminary results show that, compared with Non-dominated sorting genetic algorithm-II (NSGA-II), multi-objective particle swarm optimizer (MOPSO), multi-objective particle swarm optimizer based on decomposition (dMOPSO) and multi-objective evolutionary algorithm based on decomposition and differential evolution (MOEA/D-DE), the proposed algorithm has good performance on convergence and diversity.

Keywords: decomposition; differential evolution; multi-objective optimization; particle swarm optimization algorithm; direction angles

0 引言

在科学研究和工程实践中, 许多问题都可以归类为多目标优化问题 (MOPs)^[1]. 多目标优化问题存在多个相互冲突的目标, 其中一个目标的改善必然会降低另外一个或其他多个目标的性能. 因此, 多目标优化问题没有单个最优解, 而是一个 Pareto 最优解集. 进化算法 (EA) 具有较好的全局搜索能力, 且一次

运行可以得到多个解的特性, 进化多目标优化算法已被广泛应用于解决多目标优化问题^[2].

粒子群优化算法 (PSO)^[3] 是根据鸟群飞行觅食行为提出的一种群智能优化算法. PSO 结构简单, 收敛速度快, 已被成功应用于求解单目标优化问题, 也是适合解决多目标优化问题的方法之一.

文献 [4] 提出了多目标粒子群优化 (MOPSO) 算

收稿日期: 2016-02-22; 修回日期: 2016-05-26.

基金项目: 国家自然科学基金项目 (61374137).

作者简介: 李飞 (1988—), 男, 博士生, 从事多目标进化算法的研究; 刘建昌 (1960—), 男, 教授, 博士生导师, 从事复杂工业过程建模控制与优化等研究.

†通讯作者. E-mail: lanceleeneu@126.com

法,算法通过引入 Pareto 支配关系确保其收敛性,采用外部存档的精英保持策略存储 Pareto 最优解,并通过轮盘赌方法从外部档案中随机选择全局最优个体.为了维持种群和算法的多样性,引入了自适应变异操作.文献[5]借鉴 NSGA-II^[6]中快速非支配分类排序和拥挤距离的策略,提出了非劣排序多目标粒子群优化(NSPSO)算法;文献[7]将 Pareto 支配概念与综合学习粒子群(CLPPO)算法^[8]结合,提出了多目标综合学习粒子群优化(MOCLPSO)算法;文献[9]基于混沌序列和自适应变异策略,提出了多目标自适应混沌粒子群优化(MACPPO)算法.上述多目标粒子群优化算法都是基于 Pareto 支配关系的,Pareto 支配关系虽然可以很好地确保算法的收敛性,但是仍需要设计相应的多样性保持策略对显式和隐式档案进行修剪,因此算法设计较为复杂.

基于分解的多目标进化(MOEA/D)算法^[10]自提出以来引起了研究者的广泛关注.MOEA/D 主要将多目标优化问题分解为多个单目标优化问题来协同求解,相对于典型的基于 Pareto 支配关系的多目标优化 NSGA-II 算法^[5]和 SPEA-2^[11]有较低的计算复杂度.文献[12]提出了基于分解的多目标粒子群优化(MOPSO/D)算法,使用 PSO 代替 MOEA/D 中的进化算法,并通过分解方法更新粒子的向导粒子;文献[13]提出了 dMOPSO 算法,通过分解的方法更新外部存档和向导粒子,若个体最优粒子经过一定的代数没有更新,则采用高斯分布重新初始化粒子,协助粒子跳出局部最优.然而,对于复杂的 Pareto 前沿,若仅依靠分解策略,则最终得到的解很难准确均匀地覆盖完整的 Pareto 前沿.

为了提高 dMOPSO 算法的收敛性和多样性,提出一种基于分解和差分进化的多目标粒子群优化(dMOPSO-DE)算法.本文提出方向角概念,通过方向角产生一组均匀分布的方向向量引导粒子飞行,然后通过分解策略和差分进化算法(DE)^[14]更新种群的全局最优粒子,采用粒子重置策略维护种群多样性.通过对比其他算法的 GD(Generational distance), I_H (Hypervolume indicator),IGD(Inverted generational distance)性能评价指标可知,该算法可以有效解决 MOPSO 易陷入局部最优的问题,可用于处理多前沿多目标优化问题.

1 预备知识

1.1 多目标优化

不失一般性,以最小化为例,多目标优化问题可以定义为如下模型:

$$\min F(x) = \{f_1(x), f_2(x), \dots, f_M(x)\}. \quad (1)$$

其中: M 为目标维数, $F(x)$ 为 M 维目标向量, $f_i(x)$ 为第 i 个目标向量, x 为决策变量.

求解多目标优化问题的最终目的是得到一组尽量收敛于真实 Pareto 前沿且均匀分布的 Pareto 解集.

1.2 分解方法

分解方法给每个个体赋予一个不同的权重向量,通过分解方法将多目标优化问题转化成单目标优化问题.常用的分解方法有 Weighted Sum、Tchebycheff 和 PBI 方法^[10]. Weighted Sum 方法可以很好地处理凸 Pareto 前沿问题,但难以解决非凸问题;对于高维多目标优化问题(通常指目标函数个数大于3个),选取相同的均匀分布的方向向量,PBI 方法得到的解集比 Weighted Sum 方法和 Tchebycheff 方法得到的解集分布性更好;适当选择参数 θ ,PBI 方法在不同的高维多目标优化问题上的性能要好于 Weighted Sum 方法和 Tchebycheff 方法.本文主要解决2个和3个目标的优化问题,采用 Tchebycheff 分解方法更合适.

Tchebycheff 分解方法的计算公式为

$$g^{\text{TE}}(x|w, z^*) = \max_{i=1,2,\dots,M} \{w_i(f_i(x) - z_i^*)\}. \quad (2)$$

其中: z^* 为更新过程的理想点; $z_i^* = \min\{f_i(x)|x \in \Omega\}$, Ω 为当前搜索过程的决策变量范围; w_i 为第 i 个权重向量.

1.3 粒子群优化算法

粒子群优化算法最早是在1995年由 Eberhart 等提出的一种群智能优化算法.在 PSO 中, d 维搜索空间上的一个点被称为一个粒子,也代表优化问题的一个潜在解,每个粒子的速度决定其飞行方向和位置,然后所有粒子跟随当前的最优粒子在解空间中搜索,每个粒子的速度和位置计算公式为

$$v_i^{t+1} = wv_i^t + c_1r_1(x_{pb,i}^t - x_i^t) + c_2r_2(x_{gb}^t - x_i^t), \quad (3)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \quad (4)$$

其中: v_i^t 、 x_i^t 为第 t 代的第 i 个粒子的速度和位置, $x_{pb,i}^t$ 、 x_{gb}^t 为第 t 代的第 i 个粒子个体最优粒子和全局最优粒子, w 为权值, c_1 、 c_2 为两个学习因子, r_1 、 r_2 为两个(0,1)之间的随机数.

1.4 差分进化算法

差分进化算法是一类简单而有效的群智能算法,其子代产生方式不同于其他进化算法.它在种群中随机选取一个个体作为主导个体,然后再以种群中另外两个个体的差向量作为差分向量,将主导个体与差分向量加权产生新的个体,新个体产生的公式为

$$u_{j,i} = \begin{cases} x_{j,r_1} + F(x_{j,r_2} - x_{j,r_3}), & \text{rand}_j < \text{CR}; \\ x_{j,r_1}, & \text{otherwise.} \end{cases} \quad (5)$$

其中: x_{j,r_1} 、 x_{j,r_2} 和 x_{j,r_3} 为3个随机个体, F 为缩放因子, CR 为交叉概率。

2 dMOPSO-DE算法

下面将分别从权重向量和方向向量、向导粒子选择、粒子重置策略以及算法整体流程4方面阐述本文提出的dMOPSO-DE算法。

2.1 权重向量和方向向量

分解方法将多目标优化问题转化成单目标优化问题, 每个权重向量对应一个单目标优化问题, 每个单目标优化问题匹配一个个体。文献[10]提出的均匀分布的权重向量可以引导算法产生均匀分布的解, 但是实际上它们很难达到所期望的效果。

如图1所示, 以Tchebycheff分解方法为例, 对于第 i 个子问题, 个体将会沿着 λ_i 的反方向逐渐向参考点 z^* 靠近, 与Pareto前沿交点 A 即为第 i 个子问题的最优解, 称 λ_i 为第 i 个子问题的方向向量。

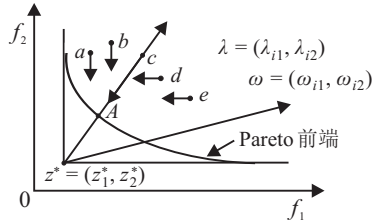


图1 Tchebycheff聚合方法个体收敛

方向向量 λ_i 所在的直线方程为 $\frac{f_1(x) - z_1^*}{\lambda_{i1}} = \frac{f_2(x) - z_2^*}{\lambda_{i2}}$, 同时由Tchebycheff分解方法可知, 该直线上的点应满足 $w_{i1}(f_1(x) - z_1^*) = w_{i2}(f_2(x) - z_2^*)$, 因此可以得到 $\lambda_{i1} : \lambda_{i2} = 1/w_{i1} : 1/w_{i2}$, 对于三维目标空间乃至高维目标空间, 此结果同样成立。当权重向量 w 均匀分布时, $1/w$ 并不是均匀分布的, 因此得到的解也不是均匀分布的, 只有当方向向量 λ 均匀分布时才能得到均匀分布的解。

基于以上分析, 本文首先生成均匀分布的方向向量, 再通过此方向向量产生一组权重向量实现解的均匀分布。但是要产生一组均匀分布的方向向量并不容易, 故本文提出方向角的概念, 在此基础上产生一组均匀分布的方向向量。

方向向量与方向角关系如图2所示。

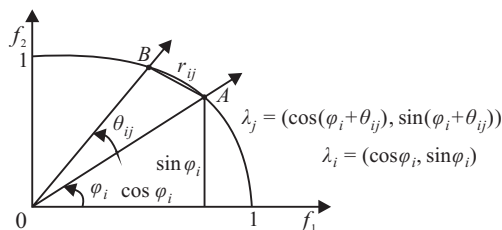


图2 方向向量与方向角关系

图2中, 单位方向向量 λ_i 与方向向量 λ_j 的夹

角为 θ_{ij} , 它们与半径为1的圆弧的交点为 A 和 B , 方向向量与坐标轴的夹角为 φ_i , 则 $\lambda_i = (\cos \varphi_i, \sin \varphi_i)$, $\lambda_j = (\cos(\varphi_i + \theta_{ij}), \sin(\varphi_i + \theta_{ij}))$, 称这两个方向向量之间的夹角 θ_{ij} 为方向角。

当1/4的超球面上有 n 个点时, 其中某个点与其他所有点的位置关系可以表示为

$$z_i = \sum_{j=1, j \neq i}^n \frac{1}{r_{ij}^2}. \quad (6)$$

其中: r_{ij} 为 i, j 点之间的弦长, 可以通过余弦公式得到 $r_{ij}, r_{ij} = \sqrt{2 - 2 \cos \theta_{ij}}$. z_i 值越小, 表示点 i 的分布性越好. 因此可以通过如下最小化问题产生 n 个均匀分布的方向向量

$$\min z_i = \sum_{j=1, j \neq i}^n \frac{1}{2 - 2 \cos \theta_{ij}}. \quad (7)$$

首先, 在1/4的超球面上随机产生 n 个点; 然后, 计算每个点与其他点的位置关系, 删除位置最差的点; 最后, 在超球面上重新随机产生一个点, 重新计算每个点与其他点的位置关系, 删除位置最差的点, 循环10000次就可以得到均匀分布的 n 个点, 即可得到 n 个均匀分布的方向向量。

通过方向向量可以计算出权重向量, 为了防止方向向量为0, 在方向向量上加上一个很小的正数0.00001, 权重向量的计算公式如下:

$$w_i = \frac{1}{\lambda_i + 1e-5}. \quad (8)$$

2.2 向导粒子选择策略

多目标粒子群优化算法中, 向导粒子选择对优化结果的影响十分明显, 向导粒子选择不当可能会导致选择压力过大或者过小, 选择压力过大可能会造成种群早熟收敛失去多样性, 选择压力过小种群收敛速度过慢, 选择合适的向导粒子可以有效地平衡种群的收敛性和多样性。

2.2.1 个体最优粒子选择

粒子 $x_{pb,i}$ 为第 i 个粒子的个体最优粒子, 即第 i 个子问题的当前最优位置。对于粒子 $x_{pb,i}$, 本文采用以下更新方式: 若下一代粒子的位置好于上一代, 则更新粒子 $x_{pb,i}$; 否则保持不变, 并记录粒子 $x_{pb,i}$ 保持不变的次数。算法1给出了个体最优粒子的选择过程。

算法1 个体最优粒子选择过程。

- 1) 选择聚合方法 $dec = \{TE \text{ or } PBI\}$
- 2) If $g^{dec}(x_i^{t+1}|w) \leq g^{dec}(x_{pb,i}|w)$, Then
- 3) $x_{pb,i} = x_i^{t+1}$
- 4) $a_i^t = 0$
- 5) Else
- 6) $a_i^{t+1} = a_i^t + 1$
- 7) End if

2.2.2 全局最优粒子选择

PSO算法中所有粒子都会跟随全局最优粒子移动,全局最优粒子无论是从种群中随机选取一个个体还是采取其他策略选取(如拥挤距离,网格等),得到的全局最优粒子仍在局部Pareto前沿上.为了避免这种情况,dMOPSO-DE算法在分解策略的基础上引入DE变异算子产生全局最优粒子.这种机制产生的全局最优粒子跳出局部Pareto前沿的可能性增大,因此能够引导种群粒子跳出局部Pareto前沿.执行过程为:首先,将新的种群与原种群合并;然后,对于每个权重向量在合并后的种群中选择分解函数值最小的一个个体,将这些个体合并成新的种群;最后,采用DE操作产生全局最优粒子,选择该粒子作为全局最优粒子.算法2给出了全局最优粒子的选择过程.

算法2 全局最优粒子选择过程.

- 1) 将新的种群与旧的种群合并,形成规模为 $2N$ 的种群
- 2) For $i = 1 : N$
- 3) For $j = 1 : 2N$
- 4) 在规模为 $2N$ 的种群中找出使权重 w_i 聚合函数值最小的个体
- 5) End for
- 6) End for
- 7) 产生一个规模为 N 的新种群
- 8) If $\text{rand} < \text{CR}$, Then
- 9) 从新种群中随机选取3个不同个体 x_{r_1} 、 x_{r_2} 和 x_{r_3}
- 10) $x_{gb,i} = x_{r_1} + F(x_{r_2} - x_{r_3})$
- 11) 保持个体 $x_{gb,i}$ 在 $[x_{lb}, x_{ub}]$ 范围内
- 12) Else
- 13) 在新种群中随机选取一个个体作为全局最优个体
- 14) End if

2.3 粒子重置策略

为了增加种群的多样性,dMOPSO-DE采用粒子的重置策略^[13].当粒子的聚合函数值在一定的代数(T_a)内一直好于新的粒子,即产生的新解的性能多次没有更新,粒子可能陷入局部最优解,此时,可以通过下式重置该粒子的位置.

$$x_i^{t+1}(j) \sim N\left(\frac{x_{gb}(j) - x_{pb,i}(j)}{2}, |x_{gb}(j) - x_{pb,i}(j)|\right). \quad (9)$$

其中: $x_i^{t+1}(j)$ 为第 i 个粒子在 $t+1$ 代的第 j 维位置, $x_{gb}(j)$ 为全局最优粒子的第 j 维位置, $x_{pb,i}(j)$ 为第 i 个个体最优粒子的第 j 维位置.该策略不仅让粒子向优秀个体学习,而且可在优秀粒子陷入局部最优时提

高种群的勘探能力并避免在差的方向上浪费资源.

2.4 dMOPSO-DE算法整体流程

dMOPSO-DE算法首先初始化种群并通过均匀化方向角产生均匀化方向向量;然后采用粒子重置策略或最基本的速度和位置更新公式更新种群信息,通过所设计的算法1和算法2选取个体最优向导和全局最优向导;最后通过隐式精英保持策略更新种群信息,算法3给出了dMOPSO-DE算法的整体流程.

算法3 dMOPSO-DE算法整体流程

- 1) 初始化种群 P
- 2) 根据式(6)~(8)产生 N 个权重向量 w
- 3) For $j = 1 : N$
- 4) If $a_i^t < T_a$, Then
- 5) 根据式(3)更新速度 v_i^{t+1}
- 6) 根据式(4)更新速度 x_i^{t+1}
- 7) Else
- 8) 根据式(9)重置粒子的位置 x_i^{t+1}
- 9) End if
- 10) 计算粒子的目标函数值并更新参考点
- 11) 根据算法1更新粒子的个体最优粒子
- 12) 根据算法2更新粒子的全局最优粒子
- 13) End for
- 14) 达到最大迭代次数终止循环
- 15) 输出种群

3 算法测试及结果分析

为了验证本文提出的dMOPSO-DE算法的性能,将该算法与NSGA-II^[6]、MOPSO^[4]、dMOPSO^[12]和MOEA/D-DE^[14]进行对比,选取ZDT系列测试函数ZDT1、ZDT2、ZDT3、ZDT4、ZDT6,DTLZ系列中的DTLZ2、DTLZ6和DTLZ7,如表1所示.通过多目标优化算法的收敛性和多样性评价指标验证算法在求解不同多目标优化问题时的性能.

表1 多目标优化测试函数

测试函数	特征	决策变量	目标
ZDT1	凸	30	2
ZDT2	凹	30	2
ZDT3	不连续	30	2
ZDT4	多前沿	10	2
ZDT6	非均匀	10	2
DTLZ2	凸	10	3
DTLZ6	非均匀	20	3
DTLZ7	不连续	22	3

3.1 性能评价指标

为了定量评价不同的多目标优化算法的性能,选用3个指标IGD、 I_H 、GD综合评价算法的收敛性和多样性.

1) IGD指标.IGD可用来综合评价算法的收敛性和多样性.IGD的函数定义如下:

$$IGD = \left(\sum_{i=1}^{\bar{n}} \bar{d}_i \right) / \bar{n}. \quad (10)$$

其中: \bar{d}_i 为目标空间上每一维向量与 PF_{true} 中最近向量之间的欧氏距离, \bar{n} 为目标空间中解的个数. IGD 越小表明找到解集的收敛性和分布度整体效果越好.

2) I_H 指标. I_H 可用来综合评价算法的收敛性和分布度, 其定义如下:

$$I_H(A) = I_h(R) - I_h(A). \quad (11)$$

其中: A 为所求解集; R 为理论解集; I_h 为纯超体积指标, 它可以计算在目标空间中被近似前沿支配的区域的超体积. I_H 值越大说明找到的解集收敛性和分布度整体效果越好.

3) GD 指标. GD 可用来表示 PF_{true} 与算法搜索得到的 Pareto 最优前沿之间的距离, 其函数定义如下:

$$GD = \left(\sum_{i=1}^n d_i^2 \right)^{1/2} / n. \quad (12)$$

其中: n 为 PF_{know} 中向量的个数, d_i 为目标空间上每一维向量与 PF_{true} 中最近向量之间的欧氏距离. GD 值越小说明找到的解集收敛性效果越好.

3.2 实验参数设定

为了保证算法对比的公平性, 对算法做 30 次独立实验. 对于两个目标的测试函数, 种群的规模设为 100, 对于 3 个目标的测试函数, 种群的规模设为 300, 迭代次数均为 300 代. 在 dMOPSO-DE 中, $c_1, c_2 = 2.0$, w 从 0.9 降到 0.4, $T_a = 2$. CR、 F 分别取 0.2 和 0.8, 从而使种群在搜索空间中进行开采, 找到更好的全局最优个体^[15]. 其他算法的参数参考相应的文献. 当计算 I_H 指标时, 对于相同的测试函数, 每种算法选择相同的参考点.

3.3 实验结果分析

5 种算法的性能对比结果如表 2~表 4 及图 3~图 18 所示.

表 2 dMOPSO-DE、dMOPSO、MOPSO、NSGA-II、MOEA/D-DE 算法 IGD 性能指标的平均值和标准差

测试问题	算法				
	dMOPSO-DE	dMOPSO	MOPSO	NSGA-II	MOEA/D-DE
ZDT1	4.80e-03(1.58e-04)	3.98e-03(3.50e-05)	1.28e-01(5.69e-02)	2.94e-02(4.85e-04)	4.05e-03(3.76e-05)
ZDT2	4.16e-03(1.52e-04)	6.05e-01(2.89e-01)	3.70e-01(3.10e-01)	3.62e-02(4.82e-03)	3.07e-01(1.39e-02)
ZDT3	1.08e-02(1.60e-04)	1.51e-02(3.30e-04)	4.47e-02(1.37e-02)	6.38e-02(1.89e-04)	3.25e-02(4.87e-05)
ZDT4	3.44e-03(1.21e-05)	4.61e-03(1.46e-05)	6.12(3.01)	2.11e-01(2.04e-03)	8.10e-01(3.63e-01)
ZDT6	1.32e-03(4.47e-06)	1.91e-03(7.50e-06)	5.20e-03(1.10e-03)	5.89e-02(1.34e-04)	1.85e-03(3.57e-06)
DTLZ2	3.30e-02(1.28e-03)	3.30e-02(2.64e-04)	9.42e-02(1.09e-02)	3.84e-02(8.09e-04)	3.74e-02(8.98e-05)
DTLZ6	1.05e-02(1.21e-04)	1.09e-02(2.16e-04)	6.40(1.15)	3.64e-03(2.02e-04)	4.41e-03(9.13e-06)
DTLZ7	5.88e-02(3.65e-04)	7.88e-02(7.23e-04)	7.06e-01(1.82e-01)	7.72e-01(3.20e-02)	1.44e-1(1.23e-2)

表 3 dMOPSO-DE、dMOPSO、MOPSO、NSGA-II、MOEA/D-DE 算法 I_H 性能指标的平均值和标准差

测试问题	算法				
	dMOPSO-DE	dMOPSO	MOPSO	NSGA-II	MOEA/D-DE
ZDT1	1.92(1.56e-04)	1.92(7.03e-05)	1.84(3.31e-02)	1.85(9.02e-03)	1.91(8.20e-05)
ZDT2	1.58(8.17e-05)	1.56(8.36e-05)	1.47(2.56e-02)	1.48(1.14e-02)	1.57(1.60e-04)
ZDT3	2.76(1.20e-02)	2.67(3.58e-02)	1.78(2.06e-03)	2.46(8.39e-02)	2.67(1.51e-04)
ZDT4	99.95(1.45e-03)	99.57(1.6e-03)	66.74(11.58)	98.19(9.81e-02)	89.87(4.00))
ZDT6	97.80(1.15e-04)	96.80(7.37e-04)	96.79(1.56e-03)	96.42(3.99e-01)	96.85(3.79e-05)
DTLZ2	2.78(2.21e-03)	2.79(1.99e-03)	1.79(4.31e-02)	2.79(1.56e-03)	2.76(4.76e-04)
DTLZ6	9.91e02(6.16e-03)	9.89e02(4.68e-01)	7.70e02(1.79e02)	9.89e02(7.31e-05)	9.89e02(1.75e-04)
DTLZ7	13.08(9.81e-04)	13.06(5.44e-03)	10.67(1.19e-02)	13.01(3.28e-03)	13.21(2.44e-03)

表 4 dMOPSO-DE、dMOPSO、MOPSO、NSGA-II、MOEA/D-DE 算法 GD 性能指标的平均值和标准差

测试问题	算法				
	dMOPSO-DE	dMOPSO	MOPSO	NSGA-II	MOEA/D-DE
ZDT1	1.34e-03(5.84e-05)	1.31e-03(4.70e-05)	2.67e-01(1.32e-01)	3.28e-02(3.28e-03)	1.72e-03(4.13e-05)
ZDT2	8.59e-04(8.17e-05)	4.20e-04(3.35e-04)	1.40e-03(1.80e-03)	5.10e-02(5.10e-04)	9.20e-04(4.85e-05)
ZDT3	5.64e-03(1.93e-04)	6.14e-03(2.60e-04)	2.62e-02(2.32e-02)	7.41e-02(3.41e-03)	4.16e-03(1.20e-04)
ZDT4	1.99e-04(1.14e-04)	1.93e-03(5.84e-03)	7.18(4.16)	2.24e-01(3.52e-03)	7.95e-01(3.58e-01)
ZDT6	6.42e-04(2.04e-05)	7.40e-04(4.30e-05)	8.99e-02(7.39e-02)	8.55e-03(1.72e-04)	7.86e-04(6.19e-06)
DTLZ2	7.66e-03(1.14e-04)	6.80e-03(1.91e-04)	5.14e-02(7.00e-03)	1.13e-02(2.50e-03)	7.61e-03(1.68e-04)
DTLZ6	4.41e-03(1.84e-05)	9.41e-03(1.85e-05)	4.48(0.99)	1.75e-03(1.29e-04)	4.02e-03(5.53e-06)
DTLZ7	2.47e-02(2.04e-03)	2.50e-02(2.05e-03)	3.00e-03(3.02e-02)	7.05e-02(1.14e-04)	1.99e-02(1.98e-04)

1) 由表 2 可以看出, dMOPSO 算法在解决 ZDT1 时得到的 IGD 均值和方差为 3.98e-03(3.50e-05), 其次

为 MOEA/D-DE、dMOPSO-DE、NSGA-II 和 MOPSO, 由此可得 dMOPSO 算法在解决 ZDT1 问题上具有优

势,在DTLZ2上的实验结果也验证了dMOPSO在解决此类问题上的优势,而MOEA/D-DE算法在DTLZ6问题上的效果最好,在其他问题上,dMOPSO-DE算法的均值和方差比其他3种算法要好。

2) 由表3可以看出,dMOPSO-DE算法在ZDT1和DTLZ2上表现出的 I_H 性能与dMOPSO算法近似

相等,相差不到0.01,但是dMOPSO算法的方差要比dMOPSO-DE算法低。故所设计的算法在解决ZDT1、DTLZ2问题时表现出的鲁棒性稍差;MOEA/D-DE算法在DTLZ7问题上的效果远远好于其他算法,在其他问题上,dMOPSO-DE算法具有较好的收敛性和多样性。

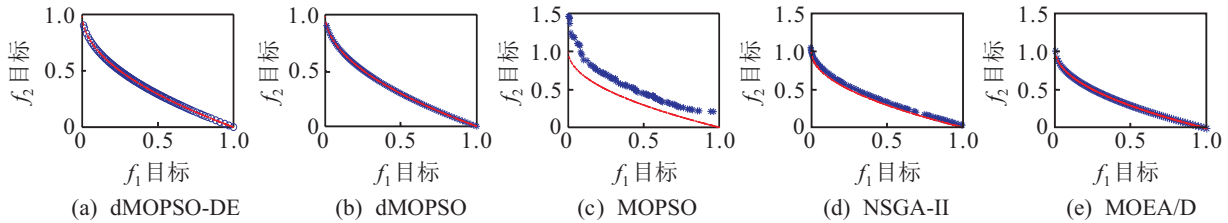


图3 5种算法在ZDT1函数上的对比

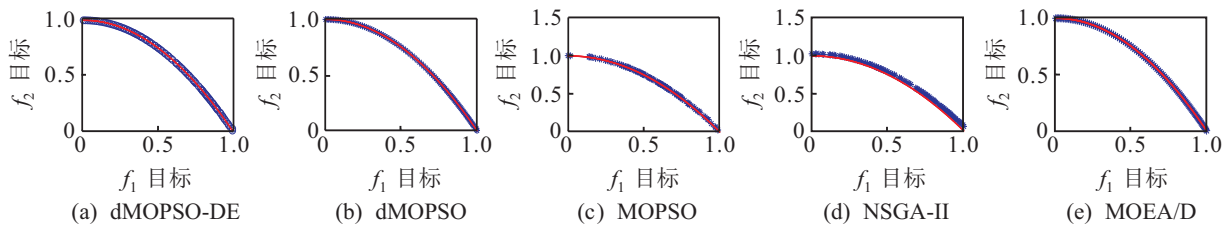


图4 5种算法在ZDT2函数上的对比

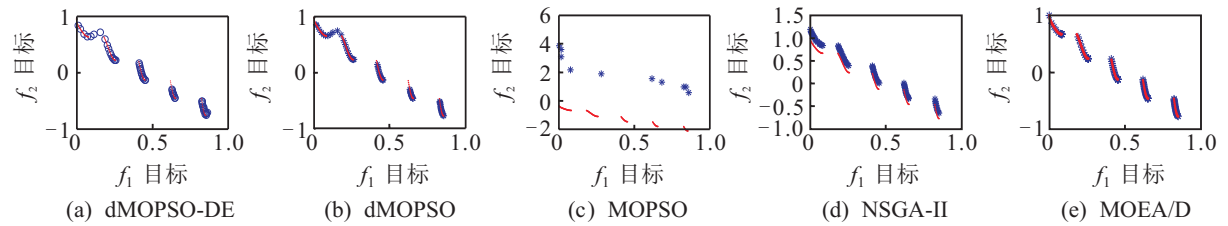


图5 5种算法在ZDT3函数上的对比

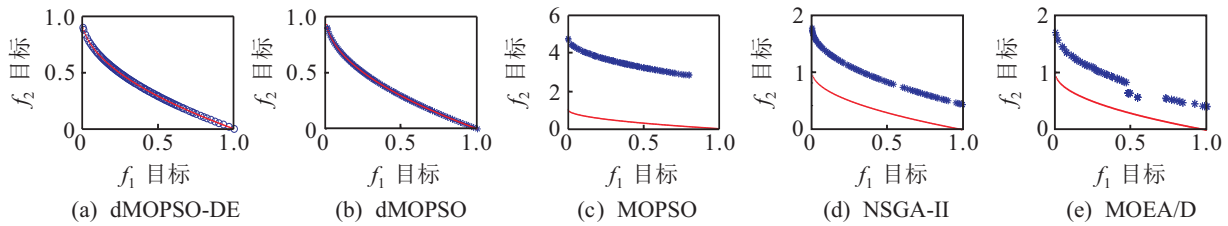


图6 5种算法在ZDT4函数上的对比

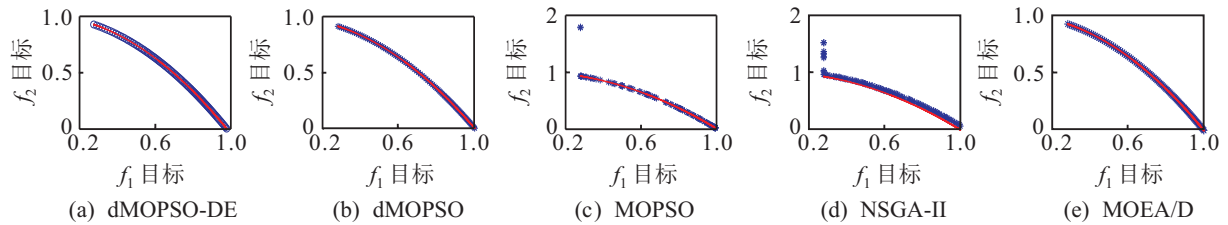


图7 5种算法在ZDT6函数上的对比

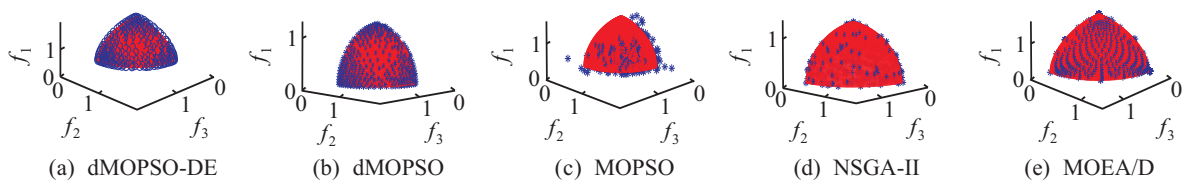


图8 5种算法在DTLZ2函数上的对比

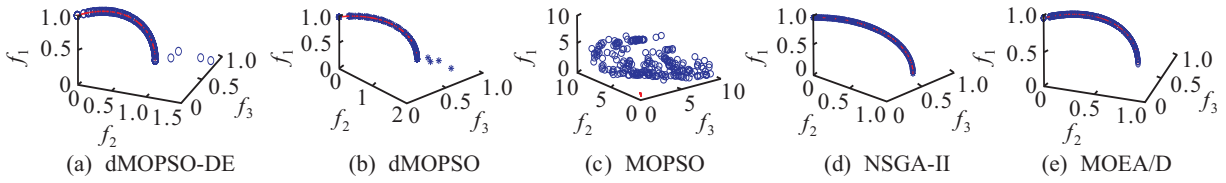


图9 5种算法在DTLZ6函数上的对比

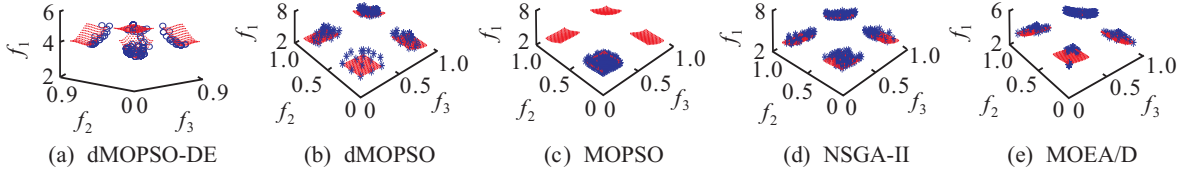


图10 5种算法在DTLZ7函数上的对比

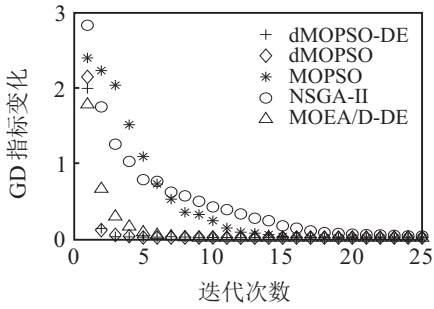


图11 ZDT3的GD趋势

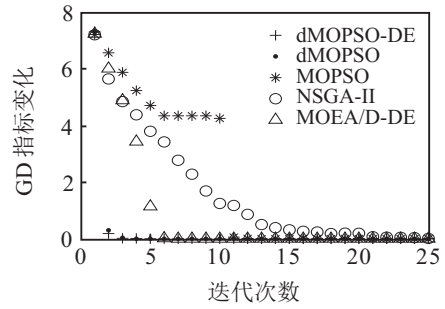


图15 ZDT6的GD趋势

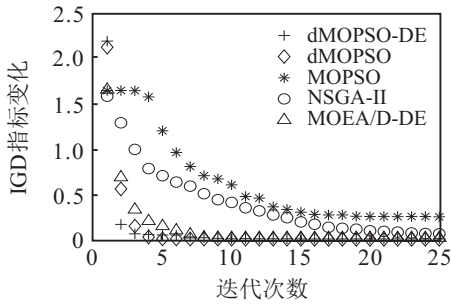


图12 ZDT3的IGD趋势

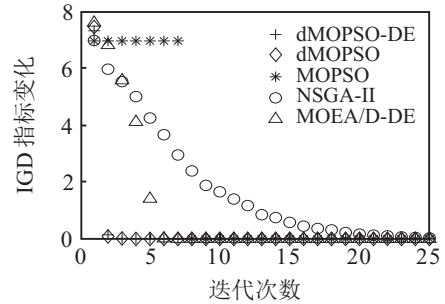


图16 ZDT6的IGD趋势

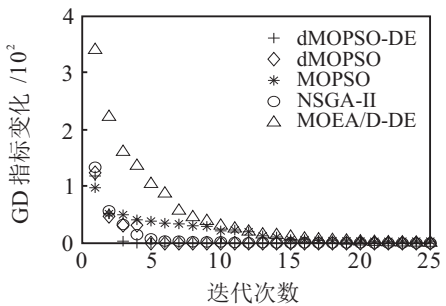


图13 ZDT4的GD趋势

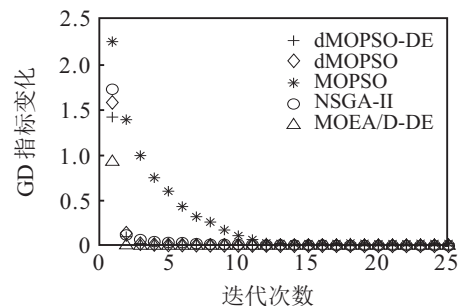


图17 DTLZ2的GD趋势

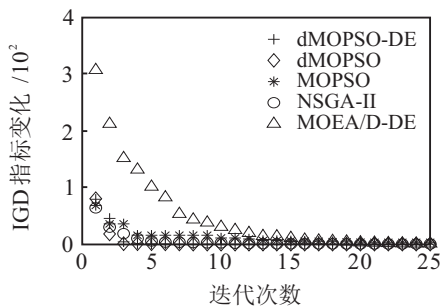


图14 ZDT4的IGD趋势

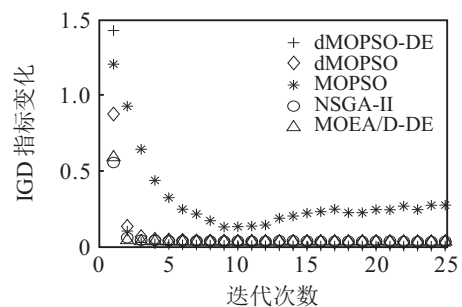


图18 DTLZ2的IGD趋势

3) 由表4可以得到与上述1)和2)相似的结果,单纯从收敛性指标上也能体现dMOPSO-DE算法的优势.比较特殊的是,在DTLZ7的GD计算时,MOPSO为 $3.00e-03(3.02e-2)$,比dMOPSO-DE和dMOPSO都好,但由图10可知,dMOPSO-DE和dMOPSO虽然在GD上比MOPSO要差一些,但是在算法多样性方面,前两种算法比MOPSO好.

4) 由图11~图18可知,dMOPSO算法在解决DTLZ2时,算法收敛速度都高于dMOPSO-DE,而在ZDT3、ZDT4、ZDT6上的效果都比dMOPSO-DE的效果差,但两种算法都优于MOPSO、MOEA/D-DE和NSGA-II.

综上所述,针对ZDT和DTLZ函数,dMOPSO-DE算法无论从IGD、 I_H 还是GD性能上都比MOPSO、NSGA-II和MOEA/D-DE的收敛性和多样性好.通过与dMOPSO算法对比可知,dMOPSO在简单的多目标优化问题(ZDT1、DTLZ2)上具有较好的收敛性和多样性,dMOPSO-DE算子在前多、非均匀、不连续多目标优化问题上具有较好的性能(ZDT2、ZDT3、ZDT4、ZDT6、DTLZ6、DTLZ7),图3~图10及表2~表4表明了dMOPSO-DE算法在处理多目标优化问题上的有效性.

4 结论

为了更好地权衡多目标粒子群优化算法解集的收敛性和多样性,提出一种基于分解和差分进化的多目标粒子群优化算法.通过引入方向角概念产生一组均匀分布的方向向量,进而提高种群的分布性;采用差分进化算法选取全局最优粒子,防止种群陷入局部最优,并采用粒子重置策略提高种群的多样性.实验结果表明,所提出的算法能够得到收敛性和多样性更好的Pareto解集.但是,将算法拓展到高维多目标优化问题时,算法仍存在许多待解决的问题,而且在实际优化问题中,如何求解Nadir点对目标空间进行归一化处理在本文中并没有考虑,下一步主要针对具体的实际问题设定相应的算法.

参考文献(References)

- [1] Collo C C, Lamont G B, David A. Van Veldhuizen. Evolutionary algorithms for solving multi-objective problems[M]. New York: Springer, 2007: 5-7.
- [2] Gong M G, Jiao L C, Yang D D, et al. Research on evolutionary multi-objective optimization algorithms[J]. J of Software, 2009, 20(2): 271-289.
- [3] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of the 6th Int Symposium on Micro Machine and Human Science. Nagoya: IEEE, 1995: 39-43.
- [4] Coello C 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.
- [5] Li X. A non-dominated sorting particle swarm optimizer for multiobjective optimization[C]. Proc of the 2003 International Conf on Genetic and Evolutionary Computation: Part I. Chicago: Springer-Verlag, 2003: 37-48.
- [6] 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.
- [7] Huang V L. Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems[J]. Int J of Intelligent Systems, 2006, 21(2): 209-226.
- [8] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Trans on Evolutionary Computation, 2006, 10(3): 281-295.
- [9] Yang J M, Ma M M, Che H J, et al. Multi-objective adaptive chaotic particle swarm optimization algorithm[J]. Control and Decision, 2015, 30(12): 1-7.
- [10] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Trans on Evolutionary Computation, 2007, 11(6): 712-731.
- [11] Laumanns M. SPEA2: Improving the strength Pareto evolutionary algorithm[R]. Zurich: Computer Engineering and Networks Laboratory(TIK), Swiss Federal Institute of Technology(ETH), 2001.
- [12] Peng W, Zhang Q. A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems[C]. IEEE Int Conf on Granular Computing. Hangzhou: IEEE, 2008: 534-537.
- [13] Zapotecas M S, Coello Coello C. A multi-objective particle swarm optimizer based on decomposition[C]. Proc of the 13th Annual Conf on Genetic and Evolutionary Computation. Dublin: ACM Order Department, 2011: 69-76.
- [14] Zhang Q, Liu W, Li H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances[C]. IEEE Congress on Evolutionary Computation. Trondheim: IEEE, 2009: 203-208.
- [15] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters[J]. IEEE Trans on Evolutionary Computation, 2011, 15(1): 55-66.