

混合分解的多目标粒子群优化算法^①

白 猛, 严宣辉, 吴坤安, 陈振兴

(福建师范大学 数学与计算机科学学院, 福州 350007)

(福建师范大学 福建省网络安全与密码技术重点实验室, 福州 350007)

摘 要: 针对多目标粒子群算法全局最优值的选取缺陷以及多样性保留缺陷, 提出了一种基于分解和拥挤距离的多目标粒子群优化算法(Smoeadpso). 算法采用切比雪夫分解机制, 将邻居向量对应的子问题的中的最优解来作为某个粒子全局最优值的候选解了更有效限制粒子飞行速度以避免粒子飞行超出解空间界限, 引入了新的速度限制因子维持了种群多样性. 本文算法与经典的多目标进化算法在 10 个测试函数上的对比结果表明, Smoeadpso 求得的 Pareto 解集与真实 Pareto 解集的逼近程度有明显提升并且对于 3 目标问题求解的均匀性也比同类粒子群算法优秀.

关键词: 切比雪夫分解; 拥挤距离; 粒子群优化; 多目标优化

Hybrid Particle Swarm Algorithm for Multi-Objective Optimization Based on Decomposition

BAI Meng, YAN Xuan-Hui, WU Kun-An, CHEN Zhen-Xing

(Fujian Normal University, Mathematics and Computer Science College, Fuzhou 350007, China)

(Fujian Normal University, Fujian Province Network Security and Cryptography Laboratory, Fuzhou 350007, China)

Abstract: To deal with the problems of the way for selecting the global best position and reserve the diversity, a multi-objective particle swarm optimization algorithm based on decomposition and crowding distance was proposed. We introduced the Tchebycheff decomposition mechanism and choose the best solution which comes from the neighbour weight vectors to be this particle's global best solution. To confine the flying of the particle, this paper introduced a new speed restriction factor. Comparing with three state-of-the-art multi-objective optimizers on ten test Problems, Smoeadpso outperforms the other algorithms as regards the coverage and approximation to the real Pareto front. Meanwhile, the uniformity of the solution set to the 3 objective problems performs better than other particle algorithms.

Key words: Tchebycheff decomposition; crowding distance; particle swarm optimization; multi-objective optimization

在科学研究与工程实践上存在着大量的多目标优化问题(Multi-objective Optimization Problem, MOP). 其求解方法一直以来都是学术界和工程界关注的热点. 与单目标问题不同, 多目标优化并不只是单纯的求出多目标问题的一个解, 而是为了找到一组称为非支配解(non-dominated set)的解集, 称为 Pareto 最优解集(Pareto-optimal set)^[1]. 解集中的每一个解称之为帕累托最优解, 这些解集的个体在目标空间上的前沿面称为帕累托前沿面. 除此之外, 很多现实生活中的多目标优化问题需要花费很大的代价去计算目标函数值和

约束条件.

粒子群优化算法(Particle Swarm Optimization algorithm)是由 Kennedy 和 Eberhart 提出的一种群智能优化算法^[2]. 由于其具有收敛速度快, 参数调节灵活, 且一次运行可以得到多个解等优势, 因此被认为是求解多目标优化问题最具有潜力的方法之一. 为了将粒子群算法应用到多目标优化上, 很多学者提出了多种方法^[3-9]. Li 等人最早将 PSO 与 NSGA-II 结合^[3]; Coello Coello 等人将自适应网格技术引入 pso, 提出了 MOPSO 算法^[4]; Sierra 等人提出了基于拥挤距离和 epsilon

① 收稿时间:2015-04-09;收到修改稿时间:2015-05-18

占优的粒子群多目标算法 OMOPSO. 上述几种对粒子群算法的改进主要是对每个粒子的全局最优值选取的改进^[5]. 通过实验验证, MOPSO, OMOPSO 对某些测试函数例如 ZDT3 和 ZDT6 上效果比较差, 而且解集与真实 pareto 曲面的逼近程度不是很好而且均匀性有待提高.

本文利用分解方法针对全局最优值的选取问题提出了一种新的策略, 将粒子群和切比雪夫算法^[11]进行混合提出了一种新的算法 Smoeadpso. 而切比雪夫分解方法对处理容易陷入局部最优解的算法有比较好的效果, 而且可以使得解的分布比较均匀, 其在 MOEA/D 中的良好表现就证明了切比雪夫分解方法的优势. 本文有以下几个方面的创新: (1)利用分解方法中的邻居向量来求解每个粒子的全局最优值, 从而改善粒子群算法容易陷入局部最优和有效增加了种群的多样性. (2)通过应用拥挤距离对外部种群的选取与截断有效保留了优秀解, 使得解集不仅在多样性上提高而且质量上也有所提高. 根据实验结果分析, 利用上述 3 个策略求得的解集在解的多样性以及对真实 Pareto 曲面的逼近程度上较对比算法都有一定程度的提升.

本文余下部分结构如下: 第一章介绍多目标优化问题相关定义; 第二章对粒子群算法和切比雪夫分解算法进行了介绍; 第三章详细阐述本文所提出的算法 Smoeadpso; 第四章在基准测试函数上对算法 Smoeadpso 的性能进行验证, 并和其他著名的多目标优化算法进行对比; 对本文的总结与展望将被放在第五章.

1 多目标优化问题的相关定义

不失一般性, 一个具有 n 个决策变量, M 个目标变量的多目标优化问题可表述为

$$\begin{cases} \min y = F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ s.t. g_i(x) \leq 0, & i = 1, 2, \dots, q \\ h_j(x) = 0, & j = 1, 2, \dots, p \end{cases} \quad (1)$$

其中, $x = (x_1, \dots, x_n) \in X \subset R^n$ 为 n 维的决策矢量, X 为 n 维的决策空间, $y = (y_1, \dots, y_m) \in Y \subset R^m$ 为 m 维的目标矢量, Y 为 m 维的目标空间. 目标函数 $F(x)$ 定义了 m 个由决策空间向目标空间的映射函数; $g_i(x) \leq 0 (i = 1, 2, \dots, q)$ 定义了 q 个不等式的约束;

$h_j(x) = 0 (j = 1, 2, \dots, p)$ 定义了 p 个等式约束. 在多目标优化问题中, 集合 $\Omega = \{x | g_i(x) \leq 0, h_j(x) = 0\}$ 为问题的可行域. 在此基础上, 给出以下几个重要定义.

定义 1 (Pareto 支配). x_A 支配 x_B : $\forall i \in \{1, 2, \dots, m\}$, 记做 $x_A \succ x_B$ (2)

定义 2 (Pareto 最优解). 个体 $x^* \in \Omega$ 被称为 Pareto 最优解(或非支配解/非劣解), 当且仅当满足以下条件: $\neg \exists x \in \Omega: x \succ x^*$ (3)

$$f_i(x_A) \leq f_i(x_B) \wedge \exists j \in \{1, 2, \dots, m\}, f_j(x_A) < f_j(x_B)$$

定义 3 (Pareto 最优解集). Pareto 最优解集是所有 Pareto 最优解组成的集合, 定义如下:

$$P^* = \{x | \neg \exists x \in \Omega: x \succ x^*\} \quad (4)$$

定义 4 (Pareto 前沿面). Pareto 最优解集 P^* 中所有 Pareto 最优解对应的目标矢量组成的曲面为 Pareto 前沿面 PF^* :

$$PF^* = \{F(x^*) = (f_1(x^*), f_2(x^*), \dots, f_m(x^*))^T | x^* \in P^*\} \quad (5)$$

2 粒子群优化算法

在多目标优化问题中, 假设目标数为 D , 群体规模为 N 的粒子 i 在 D 维空间中进行搜索, 该粒子位置和速度分别由下式来表示.

$$\text{粒子 } i \text{ 的当前位置: } X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$$

$$\text{粒子 } i \text{ 的当前速度: } V_i = (V_{i1}, V_{i2}, \dots, V_{iD})$$

随着迭代的进行, 每个粒子的速度和位置都会随时进行更新. 粒子通过下面两个公式来更新自己的速度和位置:

$$V_{ij}^{t+1} = w * V_{ij}^t + c_1 r_1 (pbest_{ij}^t - x_{ij}^t) + c_2 r_2 (gbest_{ij}^t - x_{ij}^t) \quad (6)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (7)$$

式中 w 为惯性权重, 表示粒子运动的惯性. r_1, r_2 为均匀分布在 $[0, 1]$ 内的随机数; c_1 和 c_2 为学习因子, 一般取 $[1.5, 2.5]$ 之间的数. v_{ij}^t 和 x_{ij}^t 分别表示第 i 个粒子在第 t 次迭代中第 j 维的速度和位置. $pbest_{ij}^t$ 是第 i 个粒子第 t 次迭代中找到的个体极值的第 j 维分量; $gbest_{ij}^t$ 为第 i 个粒子在第 t 次迭代中找到的全局极值的第 j 维分量.

3 多目标粒子群优化算法Smoeadpso

3.1 切比雪夫分解算法

分解在多目标进化策略中已经有了成功的应用。Zhang 和 Li 将这种传统的多目标求解策略与进化算法相结合构造了一种新颖的基于分解的多目标进化算法 MOEA/D^[10]。它将多目标优化问题分解为若干个标量子问题并且同时优化这些子问题。每个子问题通过从它的若干个邻居中获得信息来优化自己。如果将问题分成 N 个标量子问题, 那么就要产生 N 个权重向量, 每一个子问题是由一个单独的权重向量来定义的。利用切比雪夫方法, 每一个子问题可以由下式说明:

$$\text{minimize } g(x|\lambda^j, z^*) = \max_{1 \leq i \leq m} \{\lambda_i^j | f_i(x) - z_i^*\} \quad (8)$$

其中 $\lambda^j = (\lambda_1^j, \dots, \lambda_m^j)^T$ 是权重向量, 该权重向量满足 $\lambda_i \geq 0, 1 \leq i \leq m$ 并且所有权重向量的和为 1。本文利用均匀分布权重向量生成方法来生成与标量子问题个数相同的权重向量。由于 g 函数是关于 λ 的连续函数, 所以如果 λ_i 与 λ_j 是邻居关系那么 $g(x|\lambda^i, z^*)$ 的优化解应该与 $g(x|\lambda^j, z^*)$ 的优化解相接近, 则任何一个与 λ^i 的权重向量相接近的向量所构成的子问题的解都对子 $g(x|\lambda^i, z^*)$ 的解的优化有帮助, 确定 λ^i 的邻居权重向量对子问题 $g(x|\lambda^i, z^*)$ 求得优化解是必要的。在切比雪夫分解算法中, 权重向量 λ^i 的邻居被定义为一组最近权重向量集合。最近权重向量集合里面元素的个数定义为 T , T 是用户自己定义的数值。 $z^* = (z_1^*, \dots, z_m^*)^T$ 是参考点集合, $z_i = \min\{f_i(x) | x \in \Omega\}$ 。对每个帕累托最优解 x , 存在一个权重向量 λ , 每一个式(8)的最优解对应一个式(1)的帕累托最优解。切比雪夫分解的方法如图 1 所示

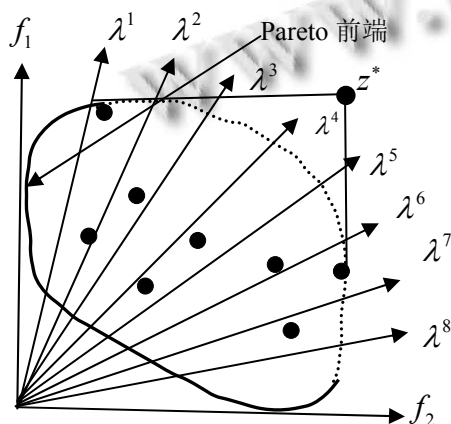


图 1 切比雪夫分解方法

假设所求多目标问题为 2 目标问题, 首先产生一组数目为 N 的初始种群, 在该目标空间区域产生与初始种群数目相同的均匀分布的权重向量, 图中椭圆区域为解空间区域, 椭圆实线部分为 Pareto 前端, $z^* = (y_1, y_2)$ 是当前解空间中各个目标函数上的最大值。由于公式(1)中求多目标函数的最小值, 所以需要使得解空间中的粒子朝远离 z^* 的区域飞行, 即朝 Pareto 前端飞行, 如果多目标问题求最大值则粒子要朝靠近 z^* 的区域飞行, 来靠近 Pareto 前端。最终收敛到 Pareto 前端。

3.2 Smoeadpso 算法的三大策略

3.2.1 全局最优值更新策略

由于分解方法会在每个权重向量所产生的标量子问题上产生一个解, 所以这种方法本身就使得解的多样性增加了而且解的收敛性也比较快, 在 MOEA/D 算法的实验结果中可以发现分解方法所产生的解相比于其他算法会更接近真实 Pareto 曲面, 所以利用分解方法可以在收敛性好的前提下更进一步进行了精细搜索。本文的 Smoeadpso 算法中每个粒子都对应一个权重向量以及一个标量子问题。根据 λ^i 与 λ^j 的欧几里得距离来判断他们两个是不是邻居向量, 如果是邻居向量, 那么根据 MOEA/D 算法中的思想, 由于标量子问题是 λ 的连续函数, 如果 λ^i 与 λ^j 是邻居关系那么 $g(x|\lambda^i, z^*)$ 的优化解应该与 $g(x|\lambda^j, z^*)$ 的优化解相接近。所以, 任何一个与 λ^i 的权重向量相接近的向量所构成的标量子问题的解都对该问题 $g(x|\lambda^i, z^*)$ 的解的优化有帮助。利用这个思想可以把本文每个粒子的所有邻居权重对应的最优的解做为它的全局最优解。

利用粒子的邻居来确定粒子的全局最优值, 初始每个粒子的全局最优值都是粒子的初始位置, 随着迭代进行, 每个粒子 i 的最新位置都会影响它的邻居的全局最优位置, 假设权重向量 j 属于 i 的邻居集合 $B(i)$, 则根据 $g(p^i|\lambda^j)$ 与 $g(gbest^j|\lambda^j)$ 的大小来确定 $gbest^j$ 的最新值。因为本文(1)式需要求的是最小值, 所以如果 $g(p^i|\lambda^j) \leq g(gbest^j|\lambda^j)$ 即它的邻居 j 相对于 i 更靠近 z^* , 则 $gbest^j = p^i$ 。

3.2.2 速度限制策略

粒子的速度控制是防止算法进入局部最优解的一个影响因素, 目前为了限制粒子的速度, 防止粒子每步的飞行距离过大, 一般粒子群算法是利用粒子飞行的速度上限和下限值来对速度进行限制, 利用这种方

法的很多粒子群算法对 ZDT4 的效果都不是很好,即算法对多模态问题的处理能力薄弱,这个根本原因还是粒子因为速度控制不好,使得粒子飞行偏离了趋近pareto 曲面的方向,为了提高处理多模态问题的能力,本文提出了一种新的速度限制策略,该策略是根据 Clerc 提出的收缩因子(constriction factor) c [11]而进行了改进,而不是利用之前在某一维上对粒子速度进行上限($upper_limit_j$)和下限($lower_limit_j$)的控制来控制粒子飞行.粒子的速度更新公式修改为:

$$v_{ij}^{t+1} = x[0.1 * v_{ij}^t + c_1 r_1 (pbest_{ij}^t - x_{ij}^t) + c_2 r_2 (gbest_{ij}^t - x_{ij}^t)] \quad (9)$$

$$\text{其中 } x = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (10)$$

$$\varphi = \begin{cases} c_1 + c_2 & c_1 + c_2 > 4 \\ 0 & c_1 + c_2 \leq 4 \end{cases} \quad (11)$$

但是上述方法在文献[5]等一些论文中对某些测试函数效果并不是十分好,那么为了进一步的限制速度,本文通过如下的速度限制方程对速度进行进一步的限制

$$v_{ij}(t) = \begin{cases} \lim it_j & v_{ij}(t) > \lim it_j \\ -\lim it_j & v_{ij}(t) \leq -\lim it_j \\ v_{ij}(t) & \text{其他} \end{cases} \quad (12)$$

$$\lim it_j = \frac{(upper_limit_j - lower_limit_j)}{2} \quad (13)$$

速度计算过程,就是通过公式(10)的限制因子乘以公式(6),通过多次试验验证,(6)式的 w 取 1 效果最好.结果最后再通过公式(12)限制求出的速度,得出最后速度.

3.2.3 精英解集保留策略

在把单目标优化问题扩展到多目标优化问题时一个需要解决的问题是如何保留精英解集,本文利用一个 keys 集合来保存.最简单直接的办法就是将每一个非支配解作为一个 key 值放入到 keys 集合.但是这种方法会使得 keys 集合增长过快,所以考虑利用拥挤距离来进行 keys 集合的第二次截断策略,第一次是利用支配策略来选择.首先计算 keys 集合的粒子的拥挤值,该拥挤值利用 NSGA-II[12]算法中的拥挤距离来进行计算.

然后利用二进制锦标赛方法为每一个粒子选择 key 值来做为全局最优解. keys 集合的个数不固定,但是不能超过最大值,最大值与内部种群的大小相同.

3.3 Smoeadpso 算法描述

在 pso 算法中,每个粒子的新的速度都是利用当前速度,粒子本身最优值 pbest 以及全局最优值 gbest.粒子的本身最优值由其先前最优位置决定.但是,对每个粒子的全局最优值的选取是个关键问题.由于每个粒子都有一个权重向量和一个子问题与自己相联,因此可利用 MOEA/D 算法中介绍的邻居,我们可以将该粒子 i 的邻居粒子中最优的位置作为 i 的全局最优值.

在算法的每次迭代中, Smoeadpso 都维持如下的结构:

- 1)含有 N 个粒子 x^1, \dots, x^N 的种群,每个粒子都有一个位置向量,一个本身最优值向量和一个速度向量,例如第 i 个粒子的 $p^i, pbest^i, v^i$. N 也是子问题的数目.
- 2)所有粒子的一个全局最优值向量,例如第 i 个粒子或者第 i 个子问题的 $gbest^i$.
- 3)一个参考点 $z = (z_1, \dots, z_m)^T$.
- 4)一个外部存档集 keys,用来存储在搜索过程中发现的非支配解.

算法流程

Smoeadpso 算法具体描述如下:

输入: 种群大小 P , keys 集合大小 N , 最大迭代次数 M .

输出: keys 集合中的所有个体.

步骤 1: 初始化

- a 初始化种群 p ,随机产生 x^1, \dots, x^N N 个粒子.
- b 初始化 keys 集合. 随机产生 N 个粒子,并且将 p 中每个粒子与 keys 集合中的 N 个粒子进行比较,如果 p 种群中的粒子 n 支配 keys 集合中粒子 t ,则将 t 从 keys 集合中移除,如果 keys 集合中没有粒子支配 n ,那么就把 n 粒子放入 keys 集合.

c 计算任意两个权重向量之间的欧几里得距离.

For $i=1, \dots, N$, 令 $B(i) = \{i_1, \dots, i_T\}$ 表示第 i 个粒子的所有邻居向量标号. $\lambda^{i_1}, \dots, \lambda^{i_T}$ 是 λ^i 的 T 个最近的权重向量. T 是邻居大小.

d 初始化 keys 集合中粒子的拥挤距离.

e 初始化种群 p 中粒子的速度. 每个粒子的初始速度设置为 0.

f 初始化 p 中每个粒子的个体极值和全局最优值.

For $i=1, \dots, N$. 令 $pbest^i = gbest^i = p^i$.

g 初始化 Z 向量, Z 的各个分量 i 初始化为初始种

群中在子目标 f_i 中的最小值, 当本文(1)中求最大值时候, 则分量 i 取 f_i 的最大值.

步骤 2: 迭代

For $i=1, \dots, N$ do

a 根据公式 6-12 更新种群 p 中每个粒子的速度.

b 根据公式 7 更新种群 p 中每个粒子新的位置.

c 更新 Z 参考点

For $j=1, \dots, m$, 如果 $z_j = f_j(p^i)$, 则 $z_j = f_j(p^i)$.

d 更新内部种群 p 中粒子的个体最优值. 如果 $g(p^i | \lambda^i) \leq g(pbest^i | \lambda^i)$, 那么 $pbest^i = p^i$.

e 更新内部种群 p 中粒子的全局最优值.

For $j \in B(i)$, 如果 $g(p^i | \lambda^j) \leq g(gbest^j | \lambda^j)$, 则 $gbest^j = p^i$.

f 计算 $keys$ 集合中粒子的拥挤距离值.

g 利用精英解集保留策略来将内部种群 p 中的非支配粒子选择到 $keys$ 集合中. 当粒子数目超过 $keys$ 集合最大值时候利用 本文的截断策略来删除 $keys$ 集合中的多余粒子.

步骤 3: 输出结果

当满足结束条件时候, 输出 $keys$ 集合中的解, 算法停止, 否则返回步骤 2.

4 数值实验及结果分析

为了验证本文算法的性能, 将本文的算法与到目前为止比较经典的多目标优化算法 NSGA-II, SPEA2, MOEA/D, OMOPSO, SMPSO 算法在标准测试集上进行了比较. 为了实验数据测试的公平性, 本文的 NSGA-II, SPEA2, MOEA/D, OMOPSO, SMPSO 这些对比算法都是利用了 Jmetal 上的源代码 (<http://jmetal.sourceforge.net/>)进行对比试验.

4.1 测试函数

为了验证本文算法的有效性, 测试函数利用 2 目标的 ZDT^[13]系列测试函数和 3 目标的 DTLZ^[14]系列测试函数. 这些测试集的真实解集是已知的. 所以可以对本文算法的实验数据来进行测试.

4.2 性能评价指标

为了评估多目标优化非劣解集的分布均匀性以及真实解的逼近程度, 本文利用 IGD 指标来对实验结果进行评价.

本文利用反向世代距离(Inverted Generational distance, IGD)指标. IGD 指标是用来测试近似解集与真实 Pareto 解的逼近程度指标.

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (14)$$

式中, n 是真实帕累托解的数目, d_i 是每一个真实帕累托解和与它欧氏距离最近的近似解中粒子的欧氏距离. IGD 值越小, 表明越接近真实 Pareto 前端. 算法收敛性越好.

4.3 实验参数设置

算法 NSGA-II, SPEA2 在 2 目标问题上的初始种群 p 大小设置为 100, 外部存档集为 100, 在 3 目标问题上初始种群大小设置为 300, 外部存档集为 300; OMOPSO, SMPSO 在 2 目标问题上的初始粒子群数为 100 在 3 目标问题上为 300. MOEA/D 在 2 目标问题上子问题个数也为 100, 在 3 目标问题上 300, 邻居数 T 为 20; 本文 Smoeadpso 在 2 目标子问题个数为 100, 在 3 目标问题上为 300, 邻居数 T 为 20, 2 目标上 $Keys$ 集合最大值为 100, 3 目标上 $Keys$ 集合最大值为 300; 对算法进行多次迭代求出结果. 本文求出的解都是独立运行 30 次求得的平均值.

表 1 算法参数值设置

参数	NSGII	SPEA2	OMOPSO	SMPSO	MOEA/D	Smoeadpso
交叉概率 Pc	0.9	0.9	0.9	0.9	0.9	0.9
交叉分布指数	20	20	20	20	20	20
变异概率 Pm	1/n	1/n	1/n	1/n	1/n	1/n
变异分布指数	20	20	20	20	20	20

4.4 实验结果与分析

表 2 各算法在 IGD 指标上的均值统计结果

测试函数	NSGA-II	SPEA2	MOEA/D	SMPSO	OMOPSO	Smoeadpso
ZDT1	1.8637e-04	1.5286e-04	7.5937e-05	1.3498e-04	1.3879e-04	<u>3.6239e-05</u>

ZDT2	1.9145e-04	1.5619e-04	5.8415e-05	1.4013e-04	1.4314e-04	2.8441e-05
ZDT3	2.5541e-04	2.3524e-04	1.3047e-04	2.0347e-04	2.1820e-04	1.3879e-04
ZDT4	2.4714e-04	1.4257e-03	3.0594e-04	1.3739e-04	1.8936e-01	3.6232e-05
ZDT6	3.5379e-04	6.5388e-04	2.0613e-05	1.1615e-04	1.2202e-04	1.3746e-04
DTLZ1	1.3104e-03	1.1158e-03	2.4890e-04	7.5803e-04	2.7850e-01	2.4876e-04
DTLZ2	7.6880e-04	5.8997e-04	3.1579e-04	7.8619e-04	7.6760e-04	3.1486e-04
DTLZ3	8.9075e-02	8.7669e-02	1.0171e-03	2.5172e-03	1.9906e+0	5.1325e-04
DTLZ4	1.2026e-03	2.4322e-03	4.2356e-04	1.2151e-03	1.1989e-03	4.4996e-04
DTLZ7	2.2669e-03	1.8752e-03	3.2272e-03	2.6699e-03	2.7357e-03	2.0033e-03

表 3 各算法在 IGD 指标上的方差统计结果

测试函数	NSGA-II	SPEA2	MOEA/D	SMPSO	OMOPSO	Smoeadpso
ZDT1	6.5008e-11	1.1179e-11	4.0985e-10	8.3583e-13	1.2744e-11	3.9879e-18
ZDT2	7.2118e-11	3.1070e-11	2.0649e-10	2.4141e-12	4.3927e-12	3.7183e-19
ZDT3	2.0474e-10	2.0474e-10	4.6942e-12	3.6185e-10	1.7412e-10	7.8599e-11
ZDT4	5.2740e-09	5.2740e-09	2.5320e-07	9.6096e-13	8.4124e-03	1.9773e-17
ZDT6	1.9773e-17	5.2826e-09	1.6002e-15	4.2359e-11	4.4990e-11	2.8065e-12
DTLZ1	2.9351e-06	2.5357e-06	1.3673e-12	4.1167e-07	4.1167e-07	6.5977e-13
DTLZ2	6.5977e-13	6.5977e-13	2.3261e-12	2.3261e-12	2.3261e-12	2.3261e-12
DTLZ3	2.3261e-12	2.3261e-12	2.3261e-12	2.3261e-12	1.8376e-01	1.2191e-06
DTLZ4	1.2191e-06	8.4757e-06	3.8246e-10	3.6320e-08	2.0416e-08	4.4380e-08
DTLZ7	2.5408e-08	3.3950e-09	3.3950e-09	9.1346e-08	1.2353e-07	1.2601e-10

IGD 指标是为了测试求得的近似 Pareto 解与真实 Pareto 前端的逼近程度. 它可以直接测试求得解的真实程度, 这是优化算法的关键, 也是优化算法的根本点. 通过表 2 的均值可以看出在 2 目标问题上 MOEA/D 和 Smoeadpso 这两种混合算法要明显好于传统的进化多目标优化算法, 这能明显体现出分解方法的巨大优势. 单独对比 Smoeadpso 和 MOEA/D 两类算法, Smoeadpso 在 7 个函数上优于 MOEA/D 算法, 这说

明粒子群与分解方法的混合会优于单独的分解算法. 也体现出了粒子群算法的优势. 继续观察方差, Smoeadpso 在七个测试函数上优于其他算法, 算法 SMPSO 与 MOEA/D 大体相当, 每个算法都在不同的测试函数上有优劣, 这体现出了粒子群算法求出的解得稳定性, 绝大多数粒子都会在与真实 Pareto 曲面相似的一个曲面上, 但是 Smoeadpso 的方差相比于 SMPSO 更是明显优于 SMPSO.

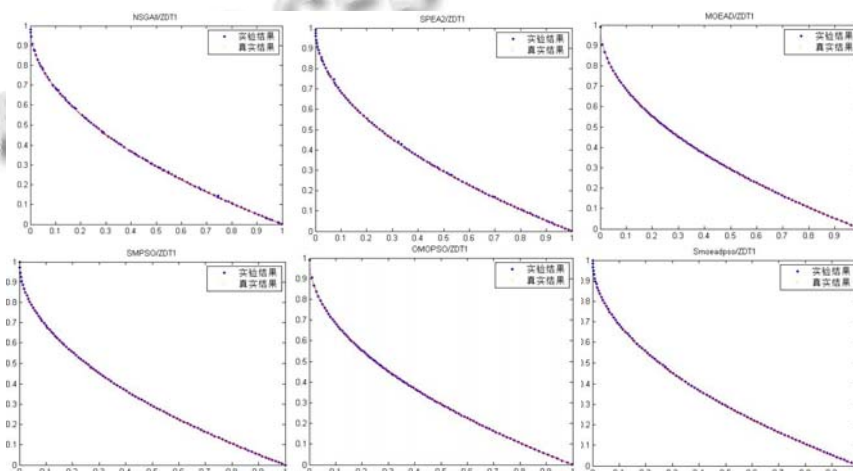


图 2 六个算法在 ZDT1 上一次随机运算结果

分析图 2, 在 2 目标问题 ZDT1 上, 除了 SMPSO 产生的解均匀性比较差之外, 其他算法表现都比较好, 在收敛性上 MOEA/D 和 Smoeadpso 表现水平相当, 但

是 Smoeadpso 相比于其他粒子群算法 OMOPSO 以及 SMPSO 它的收敛性和均匀性都是要好一些的所以 Smoeadpso 在 2 目标问题上继承了 MOEA/D 的优点.

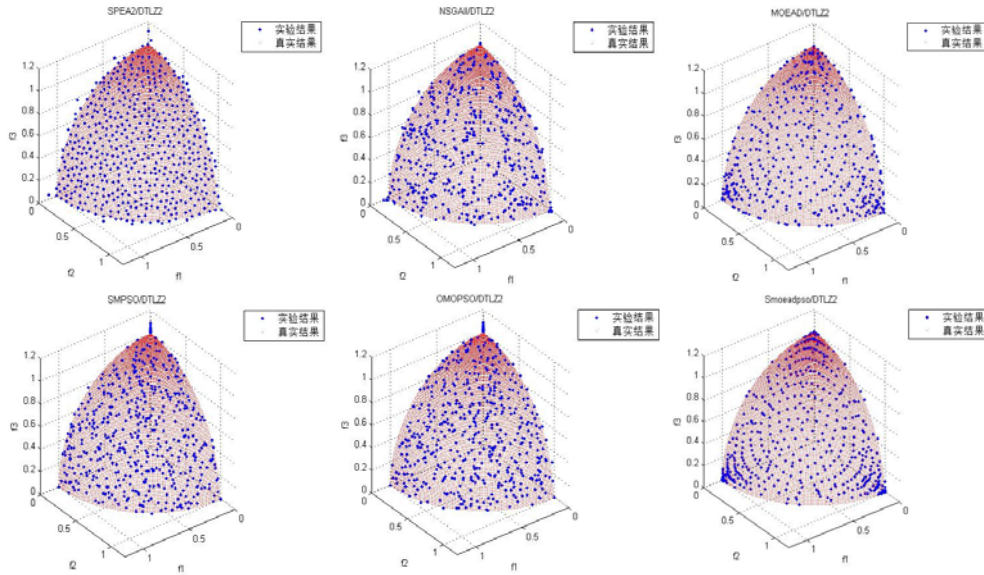


图 3 六个算法在 DTLZ2 上一次的随机运算结果

分析图 3, NSGA-II, SMPSO, OMOPSO 算法在三目标问题上的表现水平相当, SMPSO, OMOPSO 相较于 NSGA-II 在解得分布上求得的解会比 NSGA-II 更更接近于真实前端, 在分布性上 OMOPSO 会比 SMPSO 和 NSGA-II 要好一些. SPEA2 求得的解得均匀性很好, 但是在靠近真实 Pareto 前端上相比较于 MOEA/D 和 Smoeadpso 还是差些的. MOEA/D 和 Smoeadpso 的均匀性相比于其他粒子群算法还是很好的. 而且根据图示还是可以发现 Smoeadpso 的 IGD 指标会比 MOEA/D 好一些. 综合分析, Smoeadpso 在三目标问题上产生的解比当前优秀的粒子群算法产生的结果是要好的.

综上所述将分解的思想与粒子群算法混合能更好地解决多目标优化问题, 体现出了本文算法的有效性和实用性.

4 结语

本文提出了一种混合切比雪夫分解和拥挤距离的多目标粒子群优化算法 Smoeadpso. 算法利用分解的思想将多目标问题分解为 N 个子问题. 更改了粒子群算法的优化问题, 利用分解方法能够快速找到 Pareto 前沿面. 相对于粒子群的全局最优速度更新方法, 本

文利用分解方法中的 T 个邻居向量对应的子目标值来更新粒子的全局最优解, 进而进行速度更新. 本文改进的速度更新公式能够更加有效的限制粒子飞行速度, 结合分解方法对真实 Pareto 解集的优秀逼近程度, 产生了比分解方法更加优秀的 IGD 值. 通过粒子群的部分变异策略, 既增加了种群多样性, 同时保留了优良解. 通过和经典的进化多目标优化算法 NSGA-II, SPEA2, MOEA/D 以及粒子群算法中比较优秀的 SMPSO, OMOPSO 在 5 个 ZDT 测试函数, 5 个 DTLZ 测试函数共 10 个标准测试函数的实验结果对比和分析, 表明了算法 Smoeadpso 在对 Pareto 前沿面的逼近程度和有明显提高, 但是明显优于同类粒子群算法.

参考文献

- 1 Deb K. Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, 2001.
- 2 Kennedy J, Eberhart R C. Particle swarm optimization. Proc. IEEE International Conference on Neural Networks. 1995.
- 3 Li X. A non-dominated sorting particle swarm optimizer for multiobjective optimization. Genetic and Evolutionary Computation-GECCO 2003. Springer Berlin Heidelberg, 2003: 37-48.

- 4 Coello CAC, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. *IEEE Trans. on Evolutionary Computation*, 2004, 8(3): 256–279.
- 5 Sierra MR, Coello CAC. Improving PSO-based multiobjective optimization using crowding, mutation and ϵ -dominance. *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2005:505–519.
- 6 公茂果,焦李成,杨咚咚,马文萍.进化多目标优化算法研究. *软件学报*,2009,20(2):271–289.
- 7 郑金华.多目标进化算法及其应用.北京:科学出版社,2007.
- 8 刘鑫,李敏强.多目标优化进化算法及应用研究.天津:天津大学,2009.
- 9 Miettinen K M. *Nonlinear multiobjective optimization*. Norwell: Kluwer Academic Publishers,1999.
- 10 Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. on Evolutionary Computation*, 2007,11(6):712–731.
- 11 Nebro AJ, Durillo JJ, Coello CAC. Analysis of leader selection strategies in a multi-objective particle swarm optimizer. 2013 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2013: 3153–3160.
- 12 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2002,6(2): 182–197.
- 13 Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. on Evolutionary Computation*, 2002, 6(1): 58–73.
- 14 Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 2000, 8(2): 173–195.