

基于动态调整的多目标粒子群优化算法^①

李克文, 张永哲

(中国石油大学(华东) 计算机与通信工程学院, 青岛 266580)
通讯作者: 张永哲, E-mail: 864238735@qq.com

摘要: 为了改善多目标粒子群优化算法生成的最终Pareto前端的多样性和收敛性, 提出了一种针对多目标粒子群算法进化状态的检测机制. 通过对外部Pareto解集的更新情况进行检测, 进而评估算法的进化状态, 获取反馈信息来动态调整进化策略, 使得算法在进化过程中兼顾近似Pareto前端的多样性和收敛性. 最后, 在ZDT系列测试函数中, 将本文算法与其他4种对等算法比较, 证明了本文算法生成的最终Pareto前端在多样性和收敛性上均有显著的优势.

关键词: 多目标优化; 粒子群算法; 反馈信息; 进化状态

引用格式: 李克文, 张永哲. 基于动态调整的多目标粒子群优化算法. 计算机系统应用, 2017, 26(7): 161-166. <http://www.c-s-a.org.cn/1003-3254/5820.html>

Multi Objective Particle Swarm Optimization Based on Dynamic Adjustment

LI Ke-Wen, ZHANG Yong-Zhe

(College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China)

Abstract: To improve the diversity and convergence of Pareto front generated by multi objective particle swarm optimization, a detection mechanism for evolutionary state of multi objective particle swarm optimization is presented in this paper. The evolutionary state of the algorithm is assumed by detecting the updating situation of the external Pareto set to get the feedback information to adjust the evolutionary strategy of the algorithm dynamically. It enables the algorithm to take the diversity and convergence of the approximate Pareto front into account in the process of the evolution. Finally, the proposed algorithm shows a good performance compared with other four kinds of equivalence algorithms in the ZDT series test function.

Key words: multi objective optimization; particle swarm optimization; feedback information; evolutionary state

1 引言

粒子群算法(Particle swarm optimization, PSO)^[1]最早由Kennedy和Eberhart在1995年提出. 自Coello等人^[2]在2002年提出采用粒子群算法求解多目标优化问题后, 多目标进化领域中的许多成功经验被借鉴到多目标粒子群优化算法中, 并出现了大量的成果^[3-8]. 文献^[9]提出的agMOPSO算法采用自适应网格的方法评估Pareto最优解的个体密度, 进而剔除质量较差的粒子; 文献^[10]提出的pdMOPSO算法采用ROUND、RANDOM以及PROB三种策略来选取全局最优解, 这

些策略有利于提高Pareto前端的多样性, 但不易收敛到真实的Pareto前端. 文献^[11]提出的 σ MOPSO算法通过比较目标向量之间的 σ 值来选取全局最优解, 这种策略会导致算法容易早熟, 陷入局部极值. 文献此外, 还有小生境^[12]、最大最小适应度^[13]策略等. 这些多目标粒子群优化算法忽略了一个问题: 如何平衡算法在进化过程中的开发和开采? 过度的开发会影响最终的优化精度, 过度的开采会陷入局部极值. 当前所存在的多目标粒子群优化算法缺乏对进化状态的动态检测机制, 难以获取反馈信息来决定算法在何时采用何种进化策

^① 基金项目: 山东省自然科学基金(ZR2013FL034)

收稿时间: 2016-10-20; 收到修改稿时间: 2016-11-14

略. 基于此, 本文提出了一种基于动态调整的多目标粒子群优化算法进化策略, 通过检测外部档案中解的更新情况评估算法的进化状态, 获取反馈信息来动态调整算法的进化策略, 使得算法在进化过程中能够平衡开发和开采, 兼顾到最终解集的多样性和收敛性.

本文第二节介绍多目标优化问题和粒子群优化算法的基本概念, 第三节介绍通过检测外部档案的更新情况来评估算法进化状态的方法, 第四节描述本文提出的动态调整策略和算法的完整流程, 第五节分析对比试验结果, 第六节给出全文结论.

2 基本概念

2.1 多目标优化问题

对于最小化无约束连续多目标优化问题, 如公式(1)所示:

$$\min \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})]^T \quad (1)$$

for all $\mathbf{x} \in R^D$

其中, $\mathbf{x} = [x_1, x_2, \dots, x_D]^T \in S$, S 为 D 维决策空间. 一组目标函数 f_1, f_2, \dots, f_M 将决策空间映射到目标空间, $S \rightarrow \Omega \in R^M$, M 为目标个数, Ω 是 M 维目标空间.

Pareto占优: 对于任意两个向量 $\mathbf{u}, \mathbf{v} \in \Omega$ 称 \mathbf{u} 占优 \mathbf{v} , 或者 \mathbf{v} 被 \mathbf{u} 占优, 当且仅当:

$$\begin{aligned} \forall i = 1, 2, \dots, M, u_i \leq v_i \text{ 且} \\ \exists j = 1, 2, \dots, M, u_j < v_j \end{aligned} \quad (2)$$

Pareto最优解: 一个解 $\mathbf{x}^* \in \Omega$ 被称为Pareto最优解或Pareto非占优解, 当且仅当:

$$\neg \exists \mathbf{x} \in \Omega : \mathbf{x} > \mathbf{x}^* \quad (3)$$

Pareto最优解集: 所有Pareto最优解的集合 PS 被称为Pareto最优解集.

$$PS = \{\mathbf{x}^* | \neg \exists \mathbf{x} \in \Omega : \mathbf{x} > \mathbf{x}^*\} \quad (4)$$

2.2 粒子群优化算法

在粒子群优化算法中, 个体 i 没有质量, 在进化过程中其位置和速度按照公式(5)更新.

$$\begin{aligned} \mathbf{v}_i(t+1) &= \omega \mathbf{v}_i(t) + c_1 r_1 (\mathbf{pBest}_i - \mathbf{x}_i(t)) \\ &\quad + c_2 r_2 (\mathbf{gBest} - \mathbf{x}_i(t)) \\ \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \end{aligned} \quad (5)$$

在公式(5)中, \mathbf{v}_i 表示速度向量, \mathbf{x}_i 表示位置向量, t 表示迭代次数, $\omega \geq 0$ 表示惯性权重系数, $c_1, c_2 \geq 0$ 表示加速系数, r_1 和 r_2 是 $[0, 1]$ 上均匀分布的随机数, \mathbf{pBest}_i 表示第 i 个粒子的个体最优解, \mathbf{gBest} 表示群体的全局最优解.

3 进化状态评估检测

根据外部档案的更新情形, 对算法的进化状态做如下定义:

停滞状态: 在第 t 次迭代过程中, 算法产生的新解被拒绝进入外部集, 则称算法在第 t 次迭代中处于停滞状态.

多样化状态: 在第 t 次迭代过程中, 算法产生的新解替换了外部集中适应度较差的旧解, 则称算法在第 t 次迭代中处于多样化状态.

收敛状态: 在第 t 次迭代过程中, 如果算法产生的近似Pareto前端与真实的Pareto前端发生了目标空间上的距离逼近, 则称算法在第 t 次迭代过程中处于收敛状态.

种群在一次的迭代过程中可产生 N 个新解, 不同性质的新解可能使得算法处于不同的进化状态. 因此, 需要从宏观上来判断算法所处的进化状态. 下面举例说明如何判断算法的进化状态.

假设在第 t 次迭代中, 算法产生 m 个新解 $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m$ 进入到外部集. 对于某一个新解 $\mathbf{X}_i (1 < i < m)$, 在其进入外部集时, 认为 \mathbf{X}_i 使得Pareto前端拥有更好的多样性. 则 \mathbf{X}_i 满足以下三个条件: (1)外部集满. (2) \mathbf{X}_i 与外部集中的旧解互为非占优解. (3) \mathbf{X}_i 拥有比某一个旧解更好的适应度. 若不满足其中的至少一个条件, 则认为 \mathbf{X}_i 使得Pareto拥有更好的收敛性. 假设使得Pareto前端拥有更好多样性的新解个数为 a , 使得Pareto前端拥有更好收敛性的新解个数为 b .

称算法在宏观上朝着多样化状态发展, 当且仅当:

$$a > b, |a - b| > v \quad (6)$$

称算法在宏观上朝着收敛状态发展, 当且仅当:

$$a < b, |a - b| > v \quad (7)$$

称算法在宏观上平衡发展, 当且仅当:

$$|a - b| < v \quad (8)$$

其中, v 为阈值, v 取值过大, 会使得算法在进化过程中陷入局部最优, v 取值过小, 会使得算法在评估进化状态时出现状态抖动(在多样化状态和收敛状态之间频繁转换). 在总结多次试验中的阈值取值后, v 的取值范围宜为此次迭代过程中产生的新解个数的5%-10%.

4 基于动态调整的多目标粒子群优化算法

4.1 全局最优解选择策略

为了使算法在进化过程中兼顾收敛性和多样性,

需要从外部档案中选取代表收敛性和多样性的Pareto最优解作为全局最优解,引导算法种群平衡开采和开发过程。通常,收敛性和多样性是相互冲突的,因此,需要选取不同的度量标准来评估外部档案Pareto解集的收敛性和多样性。

外部档案中的Pareto最优解的个体密度可以作为多样性指标,个体密度最小的Pareto解作为全局最优解引导种群在目标空间的稀疏区域进行探索,可增加Pareto前端的多样性。在自适应网格中,Pareto最优解的个体密度 $den(X_i)$ 定义为同 X_i 格坐标相同的Pareto最优解的个数。在自适应网格中,第 k 个Pareto最优解的格坐标分量计算方式如公式(9)所示:

$$f_{k,m} = \left\lceil K \frac{f_{k,m} - f_m^{\min}}{f_m^{\max} - f_m^{\min}} \right\rceil \quad (9)$$

其中, $p_{k,m}$ 表示第 k 个Pareto最优解在第 m 个目标分量上的值, $m=1, 2, \dots, M$, M 表示目标个数, $k=1, 2, \dots, K$, K 表示当前迭代过程中外部Pareto解集中的解个数, $f_{k,m}$ 表示该最优解在第 m 个目标分量上的取值, f_m^{\min} 和 f_m^{\max} 表示目前第 m 个目标分量上的最小值和最大值。 $\lceil X \rceil$ 为向上取整。这里的目标维度分割数 K 不需要事先取值,不依赖用户对MOPSO问题的相关先验知识,将随着外部Pareto解集中的解个数变化而变化。

为了选取合适的收敛性指标,在这里借鉴参考文献[14]提出的格占优和格占优强度并稍作修改,提出 G -占优和 G -占优强度两个定义。

G -占优:假设 $G_{x,m}$ 和 $G_{y,m}$ ($m=1, 2, \dots, M$, M 为目标个数)是外部档案中任意两个解 X, Y 在自适应网格中的格坐标,称 XG -占优 Y ,当且仅当:

$$\begin{aligned} \forall m_1 = 1, 2, \dots, M, G_{x,m_1} \leq G_{y,m_1} \text{ 且} \\ \exists m_2 = 1, 2, \dots, M, G_{x,m_2} < G_{y,m_2} \end{aligned} \quad (10)$$

记作 $X >_G Y$ 。

G -占优强度:对于一个Pareto最优解 X ,其 G -占优强度是它在自适应网格中 G -占优外部档案 A 中其它Pareto最优解的数目,即:

$$S_G(X) = |\{Y | Y \in A \wedge X >_G Y\}| \quad (11)$$

G -占优强度反应了一个Pareto最优解逼近真实Pareto前端的程度,因此可以作为外部档案的收敛性指标。根据以上分析,可得出全局最优解选择策略:

输入: 1) 外部档案 A 和目标个数 M ;

2) 算法所处的进化状态 $Status \in \{\text{停滞状态, 多样}$

化状态, 收敛状态\}。

输出: 全局最优解 $gBest$ 。

步1: 遍历 A 中的所有Pareto最优解,分别计算其个体密度和 G -占优强度;

步2: 个体密度集合 D 中升序存放Pareto最优解, G -占优强度集合 G 中降序存放Pareto最优解,全局最优解候选集合 C 置空;

步3: 如果进化状态为停滞状态或平衡进化, $C = \text{TOP}(D, |C|/2) \cup \text{TOP}(G, |C|/2)$;

步4: 如果进化状态为多样化状态, $C = \text{TOP}(D, |C|/2--) \cup \text{TOP}(G, |C|/2++)$;

步5: 如果进化状态为收敛状态, $C = \text{TOP}(D, |C|/2++) \cup \text{TOP}(G, |C|/2--)$;

步6: $gBest = \text{Random}(C)$ 。

注释: (1) $\text{TOP}(P, n)$ 表示从有序集合 P 中返回最前面 n 个成员;

(2) $n++$ (或 $n--$)表示每次迭代中 n 加1(减1);

(3) $\text{Random}(Q)$ 表示从集合 Q 中随机选择一个成员。

4.2 个体最优解选择策略

在个体最优解选择策略方面,为了使算法在选择个体最优解时最大化地保留个体最优解选择的多样性,本文提出的算法将为个体最优解维护一个外部档案来存储个体最优解,同时为了降低算法的复杂度,将其大小设为全局最优解外部档案的1/6,为了保证个体最优解的变化能够跟踪全局最优解的快速变化,在个体选择最优解时将选择距离全局最优解最近(已经通过公式(9)确定)的粒子作为个体最优解。

4.3 局部极值扰动策略

为了改善粒子群算法在多目标优化中易陷入局部最优的不足,本文将采用文献[15]提出的精英学习策略(Elitism Learning Strategy, ELS)来辅助算法跳出局部极值。当算法陷入停滞状态的时候在全局最优解候选集合 C 中随机选择某个成员作为参与学习的精英粒子,学习率 lp 的计算方法如公式(12)所示:

$$lp(t) = lp(t-1) + Stepl \quad (12)$$

在公式(12)中, $Stepl$ 表示精英学习率的步长,其大小为最大学习率和最小学习率的区间长度除以迭代次数,在本文算法中,最大学习率和最小学习率分别为0.5和0,算法开始时设定精英学习率为最大值,扰动幅度参考文献[15],如公式(13)所示:

$$x_d(t) = (x_d^{max} - x_d^{min}) \times Gaussian(0, r^2) + x_d(t-1) \quad (13)$$

在公式(13)中, X_d 表示精英学习粒子被随机选中的第 d 个决策变量, x_d^{max} 和 x_d^{min} 分别为第 d 个决策变量的最大值和最小值, r 为 $[0, 1]$ 上的随机数, $Gaussian(0, r^2)$ 是均值为0、方差为 r 的高斯函数。

4.4 算法整体流程

- 输入: 1) 多目标待优化问题;
 2) 具有 D 个决策变量的搜索空间;
 3) 初始化相关参数: 外部档案的最容量 K , 最大迭代次数 T_{max} , 种群数量 N 。

输出: 外部档案中的近似Pareto最优解集。

步1: 初始化种群

- (1) 迭代次数设置为0;
 (2) 在搜索空间中均匀随机生成 N 个粒子;
 (3) 根据多目标问题计算每个粒子的第 m 个目标函数值;

(4) 判断哪些粒子可以进入外部档案。

步2: 迭代次数加1

步3: 评估进化环境

- (1) 根据公式(6)(7)(8)判定算法的进化状态;
 (2) 根据公式(9)和公式(11)计算外部档案中每个解的个体密度和 G -占优强度。

步4: 调整算法的运动参数: 算法的学习因子设定为2.05, 惯性权重随着迭代次数从0.9线性下降到0.4。

步5: 更新种群

(1) 根据全局最优解选择策略和个体最优解选择策略选择选择全局最优解 $gBest_t$ 和个体最优解 $pBest_t$ 。

(2) 根据运动方程以及运动参数更新粒子的位置和速度;

(3) 如果算法陷入停滞状态, 将根据公式(12)所确定的学习率 lp 按照公式(13)对粒子进行局部极值扰动。

(4) 评估粒子的目标函数值;

(5) 根据全局最优解选择策略和个体最优解选择策略更新种群的个体最优解候选集和全局最优解候选集;

步6: 迭代终止条件(如果迭代次数大于预设定次数, 结束; 否则转步2)。

5 实验及结果分析

为了验证本文提出的算法(daMOPSO), 本文在实

验中选取了4种常用于比较多目标优化的对等算法, 包括 agMOPSO^[9], pdMOPSO^[10], σ MOPSO^[11]及 MOEA/D^[16], 在对比实验中, 将选取各对比算法的种群大小均设定为 $N=100$, 外部档案规模均设定为 $K=100$, 算法的迭代次数均设定为 $T=300$ 。各对比算法的参数都按照各自的参考文献设置。daMOPSO的学习因子设置为2.05, 惯性权重随迭代次数从0.9线性下降到0.4, 阈值 v 取各迭代过程中产生的新解个数的6%。各对比算法在所有的测试函数上都独立运行30次以降低随机影响。

为了评估本文提出的算法生成的近似Pareto前端的多样性和收敛性, 本文将采用反转世代距离(Inverted Generational Distance, IGD)^[16]作为评估指标。其计算方法如公式(14)所示:

$$IGD = \frac{1}{|P|} \sum_{i=1}^{|P|} Dist_i \quad (14)$$

其中, P 表示真实的 Pareto 解集, $Dist_i = \min_{j=1}^{|A|}$

$\sqrt{\sum_{m=1}^M \left(\frac{f_m(p_i) - f_m(a_j)}{f_m^{max} - f_m^{min}} \right)^2}$, A 表示算法获得的近似Pareto解集。 f_m^{max} 和 f_m^{min} 分别为真实解集 P 中第 m 个目标上的最大值和最小值。IGD的值越小, 表明获得的近似Pareto前端的多样性和收敛性越好。

本文选择ZDT1、ZDT2、ZDT3、ZDT4和ZDT6这组测试函数来测试各个算法的性能, ZDT5作为布尔函数, 需要二进制编码, 本文略去该测试函数。获得的近似Pareto前端如图1至图5所示。 σ MOPSO在ZDT4测试中获得的Pareto前端和agMOPSO、 σ MOPSO、pdMOPSO在ZDT6测试中获得的Pareto前端质量太差, 在图中未表示。

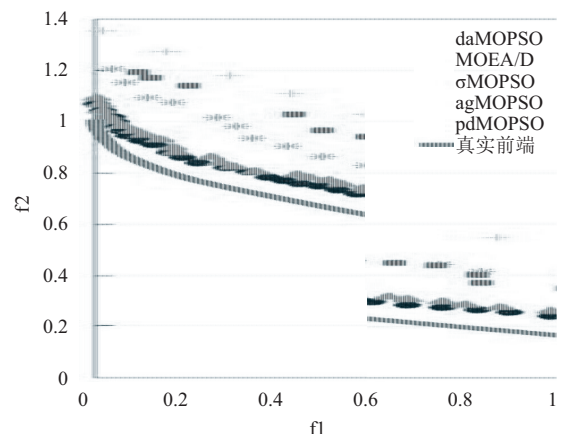


图1 ZDT1的测试结果

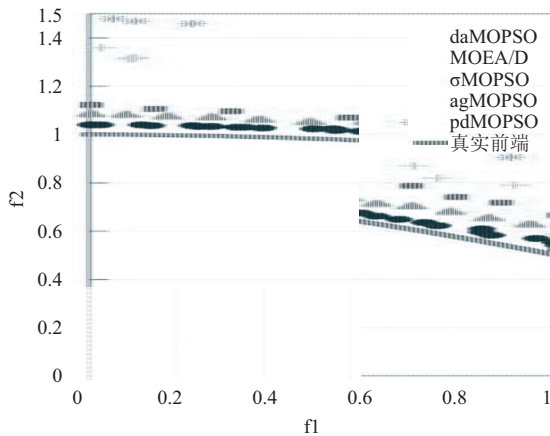


图2 ZDT2的测试结果

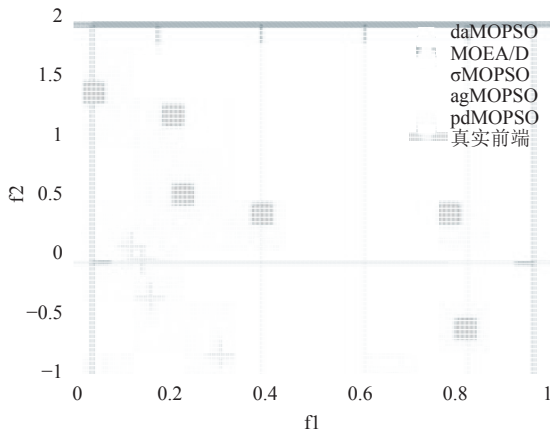


图3 ZDT3的测试结果

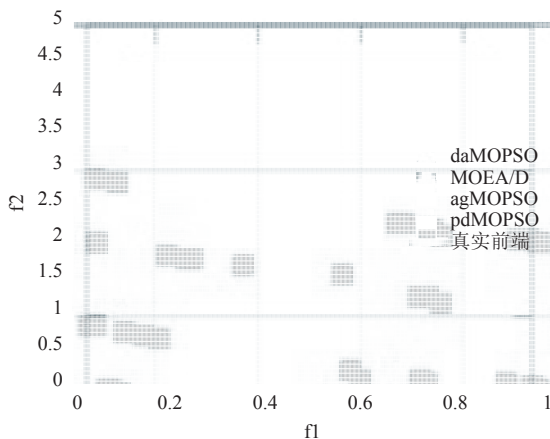


图4 ZDT4的测试结果

表1给出了各个算法在独立运行30次后获得的IGD的平均值(mean)和方差值(std).

通过表1可以看出, daMOPSO在测试函数中取得的整体效果非常理想, 仅在ZDT4的测试中结果不如MOEA/D, ZDT4函数具有 21^9 个不同的局部Pareto最优

前沿, 其中只有一个对应全局最优前沿, 这说明daMOPSO在跳出局部最优时与MOEA/D相比略显不足.

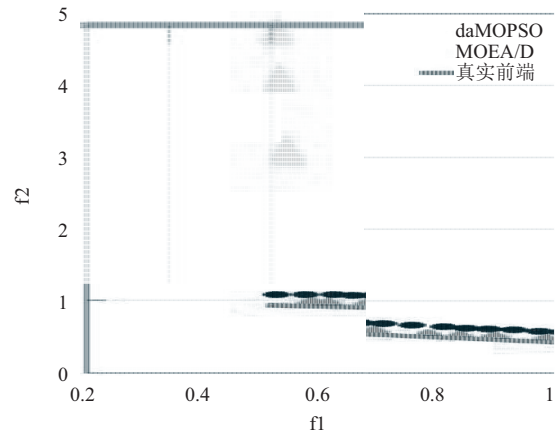


图5 ZDT6的测试结果

表1 各个算法在5种测试函数中获得的IGD对比

测试函数		daMOPSO	agMOPSO	pdMOPSO	sigmaMOPSO	MOEA/D
ZDT1	mean	4.02E-03	1.18E-01	5.81E-01	6.41E-03	4.09E-03
	std	5.56E-05	8.05E-02	2.07E-01	1.23E-03	3.12E-04
ZDT2	mean	3.81E-03	1.11E-02	6.58E-02	3.59E-01	4.02E-03
	std	4.20E-05	7.61E-03	4.25E-02	3.11E-01	1.89E-04
ZDT3	mean	3.05E-03	2.89E-01	3.24E-01	6.18E-03	8.41E-03
	std	8.01E-05	9.32E-02	1.23E-01	2.85E-03	7.12E-03
ZDT4	mean	5.13E-03	5.77E+00	4.14E+00	1.34E+00	4.57E-03
	std	8.34E-04	2.88E+00	1.47E+00	2.45E+00	8.12E-04
ZDT6	mean	2.99E-02	4.53E+00	2.46E+00	3.85E+01	3.62E-02
	std	1.45E-03	7.53E+00	8.19E+00	4.16E+00	3.18E-02

与其它几种对等算法相比, 由于daMOPSO在维护全局外部档案集的同时维护了一个个体外部档案集, 因此空间复杂度要大于其它几种对等算法, daMOPSO在每一次迭代中都会计算每一个最优解的个体密度和G-占优强度, 因此时间复杂度也略高于其它几种算法.

下一步笔者将会对daMOPSO的跳出局部最优能力以及降低时空复杂度进行探索提升. 总体来看, daMOPSO在处理这类优化问题时的能力要优于其它几种对比算法.

6 总结

基于动态调整的多目标粒子群优化算法(daMOPSO)通过检测外部档案的更新情况确认算法的进化状态, 通过个体密度和G-占优强度来评价个体适应度, 并根据不同的进化状态动态调整算法的进化策略, 保证算法在进化过程中能够兼顾Pareto前端的多样性和收敛性. 选择了ZDT系列测试函数对daMOPSO算法进行了测试, 结果表明该算法在处理不同类型的Pareto前端时

都表现出了比较优秀的性能。

参考文献

- 1 Kennedy J, Eberhart R. Particle swarm optimization. Proc. of IEEE International Conference on Neural Networks. Perth, WA, Australia. 1995. 1942–1948.
- 2 Coello Coello CA, Lechuga MS. MOPSO: A proposal for multiple objective particle swarm optimization. Proc. of the 2002 Congress on Evolutionary Computation. Honolulu, HI, USA. 2002. 1051–1056.
- 3 Maltese J, Ombuki-Berman B, Engelbrecht A. Co-operative vector-evaluated particle swarm optimization for multi-objective optimization. Proc. 2015 IEEE Symposium Series on Computational Intelligence. Cape Town, South Africa. 2015. 1294–1301.
- 4 Padhye N, Branke J, Mostaghim S. Empirical comparison of MOPSO methods—guide selection and diversity preservation. Proc. of IEEE Congress on Evolutionary Computation. Trondheim. 2009. 2516–2523.
- 5 Dai C, Wang YP, Ye M. A new multi-objective particle swarm optimization algorithm based on decomposition. Information Sciences, 2015, 325: 541–557. [doi: [10.1016/j.ins.2015.07.018](https://doi.org/10.1016/j.ins.2015.07.018)]
- 6 Coelho LDS, Guerra FA, Leite JV. Multiobjective exponential particle swarm optimization approach applied to hysteresis parameters estimation. IEEE Trans. on Magnetics, 2012, 48(2): 283–286. [doi: [10.1109/TMAG.2011.2172581](https://doi.org/10.1109/TMAG.2011.2172581)]
- 7 Moubayed NA, Petrovski A, McCall J. Clustering-based leaders' selection in multi-objective particle swarm optimisation. Yin H, Wang W, Rayward-Smith V. Intelligent Data Engineering and Automated Learning-IDEAL 2011. Berlin, Heidelberg, Germany. 2011. 100–107.
- 8 Zhou AM, Qu BY, Li H, *et al.* Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation, 2011, 1(1): 32–49. [doi: [10.1016/j.swevo.2011.03.001](https://doi.org/10.1016/j.swevo.2011.03.001)]
- 9 Coello CCA, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. IEEE Trans. on Evolutionary Computation, 2004, 8(3): 256–279. [doi: [10.1109/TEVC.2004.826067](https://doi.org/10.1109/TEVC.2004.826067)]
- 10 Alvarez-Benitez JE, Everson RM, Fieldsend JE. A MOPSO algorithm based exclusively on Pareto dominance concepts. Coello Coello CA, Hernández AA, Zitzler E. Evolutionary Multi-Criterion Optimization. Berlin, Heidelberg, Germany. 2005. 459–473.
- 11 Mostaghim S, Teich J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). Proc. of the 2003 IEEE Swarm Intelligence Symposium. Indianapolis, IN, USA. 2003. 26–33.
- 12 Zhao QL. The Research of the Niche particle swarm optimization based on self-adaptive radius technology. Proc. of Asia-Pacific Conference on Information Processing. Shenzhen, China. 2009. 97–100.
- 13 徐佳, 李绍军, 王惠, 等. 基于最大最小适应度函数的多目标粒子群算法. 计算机与数字工程, 2006, 34(8): 31–34.
- 14 胡旺, Yen GG, 张鑫. 基于Pareto熵的多目标粒子群优化算法. 软件学报, 2014, 25(5): 1025–1050.
- 15 Zhan ZH, Zhang J, Li Y, *et al.* Adaptive particle swarm optimization. IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics), 2009, 39(6): 1362–1381. [doi: [10.1109/TSMCB.2009.2015956](https://doi.org/10.1109/TSMCB.2009.2015956)]
- 16 Zhang QF, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. on Evolutionary Computation, 2007, 11(6): 712–731. [doi: [10.1109/TEVC.2007.892759](https://doi.org/10.1109/TEVC.2007.892759)]