

# 一种基于多样性信息和收敛度的 多目标粒子群优化算法

韩红桂<sup>1,2</sup>, 卢 薇<sup>1,2</sup>, 乔俊飞<sup>1,2</sup>

(1. 北京工业大学信息学部, 北京 100124; 2. 计算智能与智能系统北京市重点实验室, 北京 100124)

**摘 要:** 为了提高多目标粒子群算法优化解的多样性和收敛性, 提出了一种基于多样性信息和收敛度的多目标粒子群优化算法(Multiobjective Particle Swarm Optimization based on the Diversity Information and Convergence Degree, dicdMOPSO). 首先, 利用非支配解多样性信息评估知识库中最优解的分布状态, 设计出一种全局最优解选择机制, 平衡了种群的进化过程, 提高了非支配解的多样性和收敛性; 其次, 基于种群多样性信息设计出一种飞行参数调整机制, 增强了粒子的全局探索能力和局部开发能力, 获得了多样性和收敛性较好的种群. 最后, 将 dicdMOPSO 应用于标准测试函数测试, 实验结果表明, dicdMOPSO 与其他多目标算法相比不仅获得了多样性较高的可行解, 而且能够较快的收敛到 Pareto 前沿.

**关键词:** 智能优化算法; 多目标粒子群优化; 种群多样性信息; 非支配解多样性信息; 收敛度

**中图分类号:** TP18 **文献标识码:** A **文章编号:** 0372-2112 (2018)02-0315-10

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2018.02.009

## A Multiobjective Particle Swarm Optimization Algorithm Based on the Diversity Information and Convergence Degree

HAN Hong-gui<sup>1,2</sup>, LU Wei<sup>1,2</sup>, QIAO Jun-fei<sup>1,2</sup>

(1. Department of Information, Beijing University of Technology, Beijing 100124, China;

2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China)

**Abstract:** To improve the diversity and convergence of optimal solutions in multiobjective particle swarm optimization (MOPSO) algorithm, a multiobjective particle swarm optimization algorithm, based on the diversity information and convergence degree, named dicdMOPSO, is developed in this paper. Firstly, a global optimal solution selection mechanism, based on the distribution of optimal solutions in the knowledge base with the diversity information of non-dominated solutions, is introduced to balance the evolutionary process of population to improve the diversity and convergence of non-dominated solutions. Then, to enhance global exploration and local exploitation abilities of particles, a flight parameter adjustment mechanism is proposed to obtain the particles with better diversity and convergence by using the population diversity information. Finally, the experiment results demonstrate that, compared with other multiobjective algorithms, this proposed dicdMOPSO algorithm can not only obtain the optimal solutions with better diversity, but also be faster to catch the Pareto front.

**Key words:** intelligent optimization algorithm; multiobjective particle swarm optimization; population diversity information; diversity information of non-dominated solutions; convergence degree

## 1 引言

大多数工程和科学问题都是多目标优化问题(Multiobjective Optimization Problems, MOPs), 各个被优化的目

标之间通常相互冲突, 但往往又需要同时考虑这些目标<sup>[1-4]</sup>. 为了解决 MOPs, 多目标进化算法(Multiobjective Evolutionary Algorithm, MOEA) 得到了广大学者的关注<sup>[5-7]</sup>. 近年来, 多目标粒子群优化(Multiobjective Particle

收稿日期: 2016-08-07; 修回日期: 2016-11-07; 责任编辑: 孙瑶

基金项目: 国家自然科学基金(No. 61622301, No. 61533002); 中国博士后科学基金(No. 2014M550017); 教育部博士点基金(No. 20131103110016); 北京市教育委员会项目(No. km201410005001, No. KZ201410005002)

Swarm Optimization, MOPSO) 算法由于具有并行性和全局搜索能力等特点, 已被越来越广泛地用来解决 MOPs<sup>[8]</sup>.

在 MOPSO 算法中, 如何提高算法所得非支配解的收敛性和多样性是算法需要解决的主要难题<sup>[9]</sup>. 合理的知识库更新策略能够保证 MOPSO 算法非支配解的多样性<sup>[10]</sup>. Coello 等提出利用自适应超网格策略更新知识库, 增强了知识库中非支配解的多样性<sup>[11]</sup>. 优先权顺序法和集群操作法<sup>[12,13]</sup> 虽然保证了非支配解的多样性, 但是没有考虑非支配解集的收敛性, 并且耗时间较长, 收敛速度较慢.

为了提高 MOPSO 算法优化过程中非支配解集的收敛性, 选取合适的全局最优解是平衡整个种群和非支配解的收敛性的一种有效的方式<sup>[14]</sup>. Knowles 等提出通过自适应超网格选择最优解, 该方法增强了非支配解的多样性, 但是会使得全局最优解朝着非代表性的支配前沿靠近<sup>[15]</sup>. Omran 等提出利用聚类技术将整个种群划分为多个子群, 从各个子群中选择全局最优解<sup>[16]</sup>. Zhang 等提出将多目标分解为多个单目标的分解法, 通过 Pareto 排序的方法容易使算法陷入局部最优<sup>[17]</sup>. 以上三种方法<sup>[15-17]</sup> 没有综合考虑到非支配解集的多样性和考虑非支配解的收敛性, 并且其全局最优解选择机制不能平衡粒子的全局和局部搜索能力.

为了能够平衡粒子的全局探索能力和局部开发能力, 粒子飞行参数的调整对其起了关键作用. Hu 等提出了一种基于平行单元坐标系的自适应多目标粒子群算法, 根据当前种群的分布状态利用 Pareto 熵实现自适应调整粒子的飞行参数, 实验结果表明该方法提高了知识库中非支配解的多样性和收敛性<sup>[18]</sup>. 通过 Pareto 熵或文化框架机制设计飞行参数调整机制, 平衡了粒子的全局探索能力和局部开发能力, 由于在计算过程中引入大量的变量, 增大了计算过程的复杂度.

为提高多目标粒子群算法优化解的多样性和收敛性, 文中设计出一种基于非支配解多样性信息的多目标粒子群优化算法 (Multiobjective Particle Swarm Optimization Based on the Non-Dominated Solution Diversity Information, nsdiMOPSO); 同时, 在 nsdiMOPSO 算法的基础上, 加入自适应飞行参数调整机制, 设计出一种基于多样性信息和收敛度的多目标粒子群优化算法 (Multiobjective Particle Swarm Optimization based on the Diversity Information and Convergence Degree, dicdMOPSO). 文中设计出的 dicdMOPSO 同时考虑了知识库更新、全局最优选择和飞行参数调整三个部分.

## 2 MOPSO 算法

### 2.1 多目标问题 (MOPs)

多目标优化问题是指在给定的区域上优化多个冲

突的目标的问题. 最小化一个  $m$  维子目标的 MOPs 可以描述为:

$$\text{Min} (F(\mathbf{x})) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})] \quad (1)$$

式中,  $\mathbf{x}$  是  $n$  维决策变量的向量,  $f_i(\mathbf{x})$  是第  $i$  个目标函数,  $m$  是子目标函数的个数.

在多目标优化中, 给定决策向量  $\mathbf{y}$  和  $\mathbf{z}$ , 如果  $f(\mathbf{y})$  对于任意目标小于等于且至少存在一个目标小于  $f(\mathbf{z})$ , 那么  $\mathbf{y}$  支配  $\mathbf{z}$ , 或者  $\mathbf{z}$  被  $\mathbf{y}$  支配,

$$\forall i: f_i(\mathbf{y}) \leq f_i(\mathbf{z}) \text{ and } \exists j: f_j(\mathbf{y}) < f_j(\mathbf{z}) \quad (2)$$

式中,  $i = 1, 2, \dots, m, j = 1, 2, \dots, m$ . 如果  $\mathbf{y}$  和  $\mathbf{z}$  为存在支配关系, 那么  $\mathbf{y}$  和  $\mathbf{z}$  可以比较优劣. 如果  $\mathbf{y}$  和  $\mathbf{z}$  互不支配, 那么  $\mathbf{y}$  和  $\mathbf{z}$  无法比较.

### 2.2 MOPSO 算法

粒子在一个  $n$  维的搜索空间中, 粒子在每一次的迭代过程中, 粒子的速度和位置更新公式如下所示:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

$$v_i(t+1) = \omega v_i(t) + c_1 R_1 (p_i(t) - x_i(t)) + c_2 R_2 (g_d(t) - x_i(t)) \quad (4)$$

式中,  $\omega$  是惯性权重,  $c_1$  和  $c_2$  是学习因子,  $R_1$  和  $R_2$  是  $[0, 1]$  之间的随机值,  $p_i(t)$  是粒子在第  $t$  次迭代的历史最优位置,  $g_d(t)$  是第  $t$  次迭代的全局最优位置,  $x_i(t)$  是第  $i$  个粒子在第  $t$  次迭代的位置,  $v_i(t)$  是第  $i$  个粒子在第  $t$  次迭代的速度.

## 3 dicdMOPSO 算法

由知识库更新机制和全局最优解选择机制得到的 nsdiMOPSO 算法既能够保留知识库中具有代表性的非支配解, 又能准确的引导种群的进化方向; 加入自适应飞行参数调整机制设计出的 dicdMOPSO 算法进一步实现了粒子的全局探索能力和局部开发能力的平衡.

### 3.1 知识库更新机制

随着 MOPSO 算法迭代过程的进行, 当知识库中非支配解的数量超过容量时, 提出一种知识库更新机制. 第  $t$  次迭代后知识库  $C_t$  中解的构成包括第  $t$  次迭代产生的非支配解集  $A_t$  和前  $t-1$  次迭代保留下来的非支配解集  $M_{t-1}$ . 算法在第  $t$  次迭代产生的解被知识库  $C_{t-1}$  中的解所支配时, 该解不能进入知识库  $C_t$ ; 当知识库  $C_{t-1}$  中的解被第  $t$  次迭代产生的解所支配时, 则从知识库中删除被支配的解, 放入集合  $B_{t-1}$  中.

**定义 1 (支配强度)** 一个解  $\mathbf{x} \in A_t$  的支配强度  $S_c(\mathbf{x})$  是  $\mathbf{x}$  在知识库  $C_t$  中能够支配的解  $\mathbf{y} \in B_{t-1}$  的数目, 即

$$S_c(\mathbf{x}) = |\{\mathbf{y} | \mathbf{y} \in B_{t-1} \wedge \mathbf{x} > \mathbf{y} \wedge \mathbf{x} \in A_t\}| \quad (5)$$

其中,  $A_t$  是知识库  $C_t$  中第  $t$  次迭代产生的非支配解集,  $B_{t-1}$  是知识库  $C_t$  中第  $t$  次迭代产生的解所支配的解的集合, 支配强度  $S_c(\mathbf{x})$  是解  $\mathbf{x}$  能够支配  $B_{t-1}$  中解的数

目. 当  $S_c(\mathbf{x})$  为零时, 该解为无支配强度.

**定义 2 (收敛度)** 一个解  $\mathbf{x} \in A_i$  与所有被其支配的解  $\mathbf{y}_i \in B_{i-1}$  的平均欧式距离, 即

$$D_{ave}(\mathbf{x}) = \frac{\sum_{i=1}^{s_c(\mathbf{x})} \|\mathbf{x} - \mathbf{y}_i\|}{s_c(\mathbf{x})} \quad (6)$$

式中,  $D_{ave}(\mathbf{x})$  是解  $\mathbf{x}$  的收敛度,  $\mathbf{y}_i$  是被解  $\mathbf{x}$  支配的第  $i$  个解. 非支配解集  $M_{i-1}$  中解的收敛度为零, 无支配强度的解其收敛度也为零.

**定义 3 (密度)** 在知识库中, 其中的一个非支配解的密度是该非支配解和其相邻的两个非支配解的平均欧式距离, 即:

$$Dist_i = \frac{dist_{1i} + dist_{2i}}{2} \quad (7)$$

式中,  $Dist_i$  是第  $i$  个非支配解的密度值,  $dist_{1i}$  是第  $i$  个非支配解和相邻的一个非支配解的欧氏距离,  $dist_{2i}$  是第  $i$  个非支配解和另一个相邻的非支配解的欧氏距离. 非支配解的密度越大, 表示该非支配解附近越稀疏.

非支配解集  $A_i$  中的解包含两种: 一种是能够支配  $B_{i-1}$  集合中的解 (有支配强度); 另一种是不能支配  $B_{i-1}$  集合中的解 (无支配强度).

知识库  $C_i$  更新的具体流程:

**Step1** 从非支配解集的多样性考虑, 计算知识库中非支配解的密度, 删减解集  $M_{i-1}$  中密度值较小的解和非支配解集  $A_i$  中无支配强度的解.

**Step2** 如果执行 Step1 之后仍然超出知识库容量时, 考虑删减  $B_{i-1}$  集合中有支配强度的解. 计算此时知识库中非支配解的收敛度  $D_{ave}$ , 删减  $D_{ave}$  较小的解, 若  $D_{ave}$  相等, 则执行 Step3.

**Step3** 计算解集  $A_i$  中解的支配强度, 比较支配强度大小, 优先删减支配强度较小的解.

### 3.2 全局最优解选择机制

全局最优解能够引导整个粒子群的进化方向. 因此考虑根据多样性信息选择具有代表性的全局最优解.

**定义 4 (多样性信息)** 文中的多样性信息包含种群多样性信息和非支配解多样性信息两种.

**种群多样性信息:** 从粒子群本身考虑, 群体寻优过程中粒子聚集是导致精度较低的主要原因, 而种群多样性信息能够反映粒子的分布情况, 其表达式为:

$$SP_p(t+1) = \sqrt{\frac{1}{n_1 - 1} \sum_{i=1}^{n_1} (\overline{d_{pi}}(t+1) - d_{pi}(t+1))^2} \quad (8)$$

式中,  $d_{pi}(t+1)$  是在第  $t+1$  次迭代第  $i$  个粒子与其他粒子之间最小的曼哈顿距离,  $\overline{d_{pi}}(t+1)$  是所有的  $d_{pi}(t+1)$  的平均值,  $n_1$  是粒子的个数,  $SP_p(t+1)$  是第  $t+1$  次迭代粒子群的种群多样性信息.

**非支配解多样性信息:** 从整个知识库中的非支配解考虑, 非支配解的多样性信息表征着算法分布性的好与差. 非支配解多样性信息的表达式如下:

$$SP_n(t+1) = \sqrt{\frac{1}{n_2 - 1} \sum_{i=1}^{n_2} (\overline{d_{ni}}(t+1) - d_{ni}(t+1))^2} \quad (9)$$

式中,  $d_{ni}(t+1)$  是在第  $t+1$  次迭代第  $i$  个非支配解与其他非支配解之间最小的曼哈顿距离,  $\overline{d_{ni}}(t+1)$  是所有的  $d_{ni}(t+1)$  的平均值,  $n_2$  是非支配解的个数,  $SP_n(t+1)$  是第  $t+1$  次迭代的非支配解多样性信息.

根据式(9)计算的非支配解多样性信息可以作为知识库中非支配解集的多样性度量指标, 通过比较非支配解多样性信息  $SP_n(t+1)$  值与  $\alpha$  值的大小关系 ( $\alpha$  为非支配解多样性信息  $SP_n(t+1)$  好与差的分界值), 判断当前整个知识库中非支配解集的分布状态. 图 1 所示为全局最优解选择机制图.

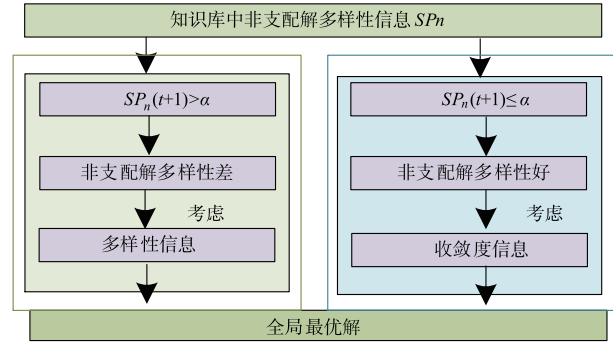


图1 全局最优解选择机制图

全局最优解选择机制执行过程如下:

(1) 若  $SP_n(t+1) \leq \alpha$ , 整个非支配解的多样性比较好, 那么此时考虑加快算法的收敛速度, 因此选择收敛度最大的非支配解作为全局最优解.

(2) 若  $SP_n(t+1) > \alpha$ , 整个非支配解的多样性比较差, 则考虑增加粒子分布的均匀性, 因此选择密度最大的非支配解作为全局最优解.

### 3.3 自适应飞行参数调整机制

在 nsdiMOPSO 算法中, 运动方程的惯性权重  $\omega$ 、学习因子  $c_1$  和  $c_2$  的取值都影响着整个粒子群的全局探索能力和局部开发能力的平衡. 因此在 nsdiMOPSO 算法的基础上, 利用式(8)中种群多样性信息反映当前的粒子群的分布状态, 设计自适应的参数调整策略, 从而获得 dicdMOPSO 算法.

其中, 当  $\omega$  增大、 $c_1$  增大和  $c_2$  减小时, 粒子的全局搜索能力比较强; 当  $\omega$  减小、 $c_1$  减小和  $c_2$  增大时, 粒子的局部搜索能力比较强<sup>[18]</sup>.

由于粒子在飞行过程中具有较为复杂的非线性特征, 特定的非线性函数难以表示其飞行状态, 因此基于

种群多样性信息及指数函数调整粒子飞行过程中惯性权重和学习因子的变化,从而更好的拟合粒子的飞行过程.基于种群多样性信息  $SP_p(t+1)$  设计非线性函数  $X_1(t+1)$ 、 $X_2(t+1)$ ,使粒子的惯性权重和学习因子按照种群多样性信息增大:

$$X_1(t+1) = e^{\frac{1}{SP_p(t+1)+1} - 1} + 1 \quad (10)$$

粒子的惯性权重和学习因子按照种群多样性信息减小,函数表达式为:

$$X_2(t+1) = e^{\frac{1}{SP_p(t+1)+1} - 1} \quad (11)$$

利用式(10)和(11),得出基于种群多样性信息的自适应惯性权重  $\omega$ 、学习因子  $c_1$  和  $c_2$  调整机制如下:

$$\omega_i(t+1) = \begin{cases} \omega_i(t) (e^{\frac{1}{SP_p(t+1)+1} - 1} + 1), & SP_p(t+1) > SP_p(t) \\ \omega_i(t) (e^{\frac{1}{SP_p(t+1)+1} - 1}), & SP_p(t+1) \leq SP_p(t) \end{cases} \quad (12)$$

$$c_{1i}(t+1) = \begin{cases} c_{1i}(t) (e^{\frac{1}{SP_p(t+1)+1} - 1} + 1), & SP_p(t+1) > SP_p(t) \\ c_{1i}(t) (e^{\frac{1}{SP_p(t+1)+1} - 1}), & SP_p(t+1) \leq SP_p(t) \end{cases} \quad (13)$$

$$c_{2i}(t+1) = \begin{cases} c_{2i}(t) (e^{\frac{1}{SP_p(t+1)+1} - 1}), & SP_p(t+1) > SP_p(t) \\ c_{2i}(t) (e^{\frac{1}{SP_p(t+1)+1} - 1} + 1), & SP_p(t+1) \leq SP_p(t) \end{cases} \quad (14)$$

$\omega_i(t+1)$  是第  $t+1$  次迭代第  $i$  个粒子的惯性权重.  $c_{1i}(t+1)$  是第  $t+1$  次迭代第  $i$  个粒子的自身学习因子.  $c_{2i}(t+1)$  是第  $t+1$  次迭代第  $i$  个粒子的社会学习因子.

若整个种群陷入局部最优,粒子群会逐渐聚集,分布不均匀,此时整个种群的多样性会变差,即  $SP_p(t+1) > SP_p(t)$ ,为避免粒子继续陷入局部最优,此时希望粒子能够跳出局部,此时由式(12)(13)和(14)可知, $\omega$  增大、 $c_1$  增大和  $c_2$  减小,会使得粒子的全局搜索能力增强,从而会加速粒子跳出局部.当整个种群的多样性逐渐变好,即  $SP_p(t+1) \leq SP_p(t)$  时,为了使得粒子群能够在比较分散的区域内找到更优的个体最优解和全局最优解,此时需要降低粒子的飞行速度,使得粒子在各自的区域内加强其局部寻优能力,此时由式(12)(13)和(14)可知, $\omega$  减小、 $c_1$  减小和  $c_2$  增大,会使得粒子的局部搜索能力加强,寻得更优解. $\omega$ 、 $c_1$  和  $c_2$  的取值会随着整个粒子群的分布状态实现自适应调整,实现粒子群多样性性能的提升,从而获得具有较好的收敛性和多样性的最优解.

### 3.4 全局最优解选择机制和自适应飞行参数调整机制的关系分析

全局最优解选择机制和自适应飞行参数调整机制对于 MOPSO 算法的运行是相辅相成的关系.其执行过程共分为以下四种情况:

(1) 当  $SP_n(t+1) > \alpha$  且  $SP_p(t+1) > SP_p(t)$ ,此种情况下,全局最优解为密度比较大的解,种群的全局搜索能力比较强,这将会有利于粒子找到更加多样化的非支配解,有利于非支配解集多样性的增强.

(2) 当  $SP_n(t+1) > \alpha$  且  $SP_p(t+1) \leq SP_p(t)$ ,此种情况下,种群的局部搜索能力比较强,整个种群将会向密度最大的解靠近,会使整个种群在比较稀疏的解附近搜索最优解,有利于增强非支配解的多样性,同时局部搜索也能够提高非支配解的收敛性.

(3) 当  $SP_n(t+1) \leq \alpha$  且  $SP_p(t+1) \leq SP_p(t)$ ,此种情况下,非支配解的多样性比较好,有利于粒子在收敛度较大的解附近找到更多最优解,同时种群的局部搜索能力比较强,能够提高非支配解的收敛性.

(4) 当  $SP_n(t+1) \leq \alpha$  且  $SP_p(t+1) > SP_p(t)$ ,此种情况下,整个种群将会向收敛度最大的解靠近,但此时整个粒子群分布不均,可能会陷入局部最优,种群的全局搜索能力的增强能够避免粒子陷入局部,同时在收敛度最大的解周围的区域中搜索最优解.

## 4 dicdMOPSO 算法具体流程

基于多样性信息和收敛度的多目标粒子群优化算法(dicdMOPSO)的具体流程如下:

**Step1** 随机初始化粒子群中各粒子的速度  $v_i(0)$ 、位置  $x_i(0)$ 、惯性权重  $\omega_i(0)$ 、学习因子  $c_{1i}(0)$  和  $c_{2i}(0)$ .并将各粒子的初始位置设为当前历史最优位置  $p_i$ .

**Step2** 计算每个粒子的适应度值,得到知识库  $C_{t+1}$ .根据式(5)(6)和(7)分别计算各个非支配解的支配强度、收敛度和密度.

**Step3** 计算此时整个非支配解多样性信息  $SP_n(t+1)$ ,执行 3.2 节中全局最优解选择过程.

**Step4** 比较 Step2 中知识库  $C_{t+1}$  中非支配解的密度值,执行 3.1 节中知识库更新过程.

**Step5** 比较种群多样性信息  $SP_p(t+1)$  和  $SP_p(t)$ ,根据式(12)(13)和(14)调整飞行参数惯性权重  $\omega_i(t+1)$ 、学习因子  $c_{1i}(t+1)$  和  $c_{2i}(t+1)$ .

**Step6** 按照式(3)和(4)更新粒子的位置和速度.

**Step7** 判断是否达到终止条件,如达到,则结束,否则返回到 Step2.

文中提出的 dicdMOPSO 算法既能够获得收敛性和多样性较好非支配解集,而且能够实现全局探索能力和局部开发能力的平衡.

## 5 实验及结果分析

### 5.1 dicdMOPSO 算法参数分析

为了充分分析 dicdMOPSO 算法的性能,首先对其非支配解多样性信息、全局最优解的密度和收敛度随迭代次数变化关系进行分析.

从图 2~图 4 中可以看出,当  $SP_n \leq \alpha$  ( $\alpha$  取 0.05) 时,整个知识库中非支配解的多样性比较好,因此选择收敛度最大的非支配解作为全局最优解.当  $SP_n > \alpha$  时,如图 2 所示,dicdMOPSO 算法在第 40 次迭代时,整个非支配解集的多样性比较差,则考虑增加非支配解分布的均匀性,因此选择密度最大的非支配解作为全局最优解,图 3 的实验结果与分析相符,dicdMOPSO 算法在第 41 次迭代找到新的有较大收敛度的解,因此在图 4 第 42 次迭代的全局最优解的收敛度会突然增大.

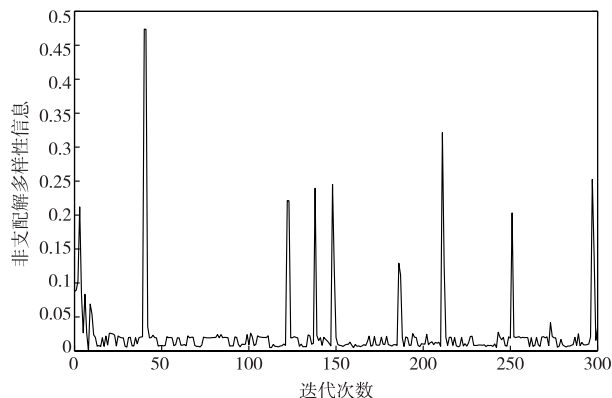


图2 非支配解多样性信息  $SP_n$  随迭代次数变化的曲线图

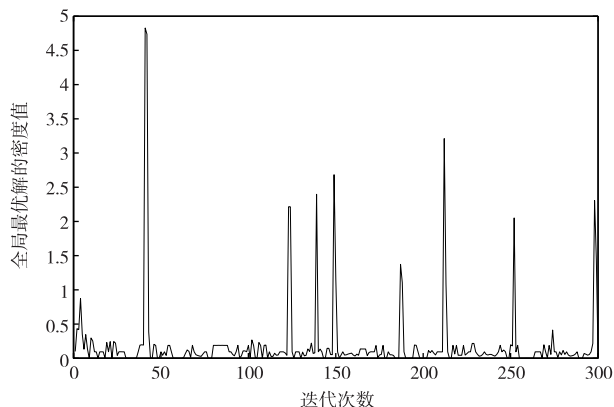


图3 全局最优解的密度值  $Dist$  随迭代次数变化的曲线图

### 5.2 MOP 测试函数实验

#### 5.2.1 多目标测试问题

为了验证 dicdMOPSO 算法的收敛效果,文中通过标准测试函数 ZDT1、ZDT2、ZDT3、ZDT4、ZDT6、DTLZ2 和 DTLZ7 测试,将 dicdMOPSO 算法与 nsdiMOPSO 算法、pccsAMOPSO 算法<sup>[18]</sup>、拥挤距离 MOPSO (cdMOPSO) 算

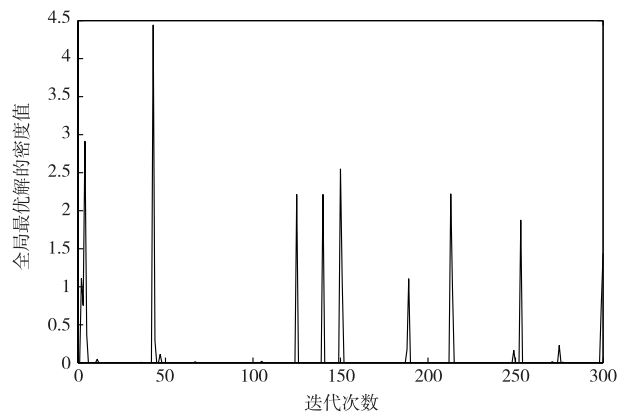


图4 全局最优解的收敛度  $D_{avg}$  随迭代次数变化的曲线图

法、聚类 MOPSO (clusterMOPSO) 算法、NSGA-II<sup>[7]</sup> 算法的测试数据相比较.文中对比算法的参数均参照相应的参考文献.其中,测试函数的参数设置如表 1 所示.

表 1 测试函数参数设置

测试函数	维数		Pareto 前沿样本数
	变量	目标	
ZDT1	30	2	1000
ZDT2	30	2	1000
ZDT3	30	2	536
ZDT4	30	2	1000
ZDT6	10	2	420
DTLZ2	10	3	5000
DTLZ7	20	3	4700

#### 5.2.2 实验参数

在比较实验中,各算法的种群大小和知识库的最大容量均设置 100,所有函数的最大迭代次数设置为 300,每种算法在所有测试函数上均独立运行 30 次.

#### 5.3 MOP 测试函数实验结果及比较分析

将 dicdMOPSO 算法利用二维测试函数测试,从图 5 至图 9 中可以看出,dicdMOPSO 算法和 nsdiMOPSO 算法逼近前沿的效果比 cdMOPSO 算法好,但 dicdMOPSO 算法比 nsdiMOPSO 算法得到的非支配解具有更好的分布性.

将 dicdMOPSO 算法利用三维测试函数 DTLZ2 和 DTLZ7 进行测试,从图 10 和图 11 中可以看出 dicdMOPSO 算法能够较好的分布在前沿上,而且其多样性也明显好于 cdMOPSO 算法.

图 12~图 13 是六种算法分别在二维和三维测试函数下得到的 IGD 盒图实验结果图.从实验中可以看出,在二维测试效果下,dicdMOPSO 算法在 ZDT1、ZDT2、ZDT4 和 ZDT6 的测试效果都明显优于其他算法,只有 ZDT3 的测试效果中其 IGD 值稍微差一点.在三维

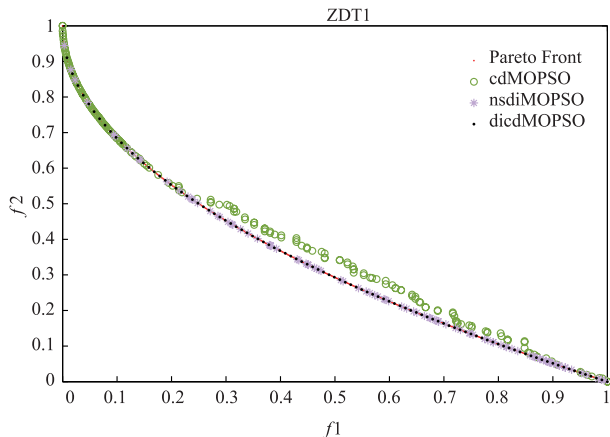


图5 dicdMOPSO、nsdiMOPSO、cdMOPSO对ZDT1测试结果图

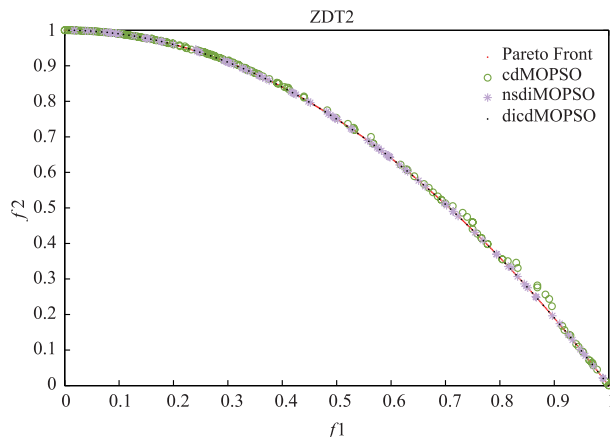


图6 dicdMOPSO、nsdiMOPSO、cdMOPSO对ZDT2测试结果图

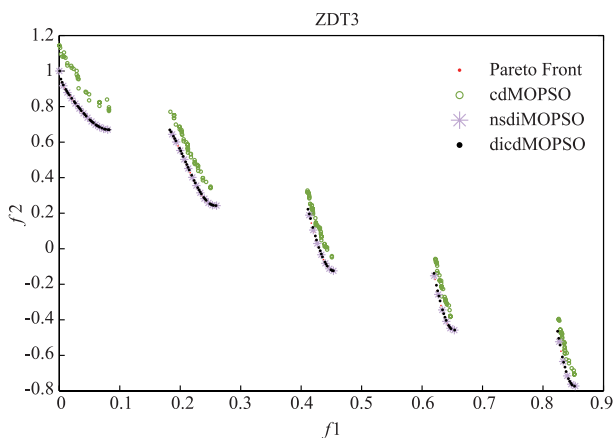


图7 dicdMOPSO、nsdiMOPSO、cdMOPSO对ZDT3测试结果图

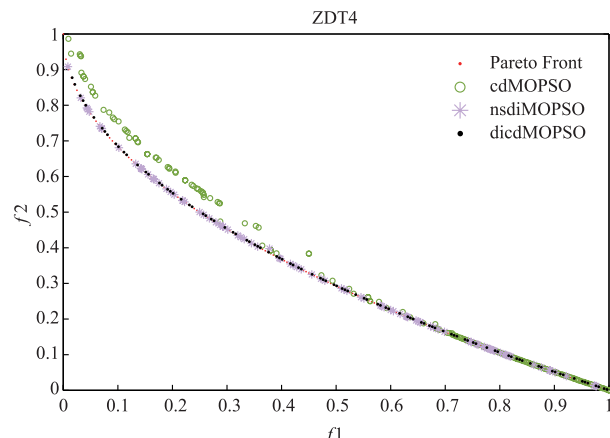


图8 dicdMOPSO、nsdiMOPSO、cdMOPSO对ZDT4测试结果图

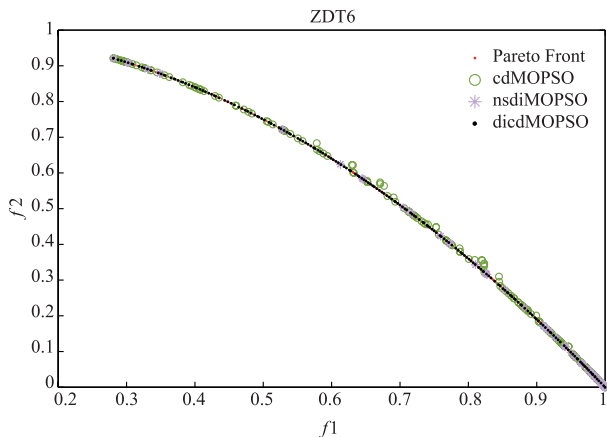


图9 dicdMOPSO、nsdiMOPSO、cdMOPSO对ZDT6测试结果图

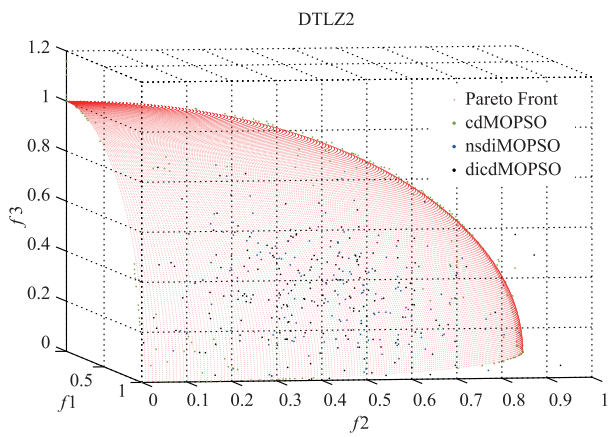


图10 dicdMOPSO、nsdiMOPSO、cdMOPSO对DTLZ2测试结果

的测试效果下, dicdMOPSO 算法在 DTLZ7 的测试效果明显优于其他算法, 在 DTLZ2 的测试效果中其 IGD 值稍微差一点. 并且在相同的函数调用次数下, 与 pccsAMOPSO<sup>[18]</sup> 算法相比, dicdMOPSO 算法的 IGD 值要优于 pccsAMOPSO 算法.

表 2 是 dicdMOPSO 算法与其它算法的 IGD 指标测试结果. 其中加粗的数值代表最优结果值, 从表 2 中看

出, dicdMOPSO 算法相比 nsdiMOPSO 算法具有更好的 IGD 值, 说明将算法加入了飞行参数调节机制后, dicdMOPSO 算法具有了更高的收敛精度. 同时 dicdMOPSO 能够具有最优的 IGD 平均值 (除了 ZDT3 函数时, cdMOPSO 有更好的 IGD 值), 同时具有最小的 IGD 标准差 (除了 ZDT3 函数, pccsAMOPSO 具有最小的标准差).

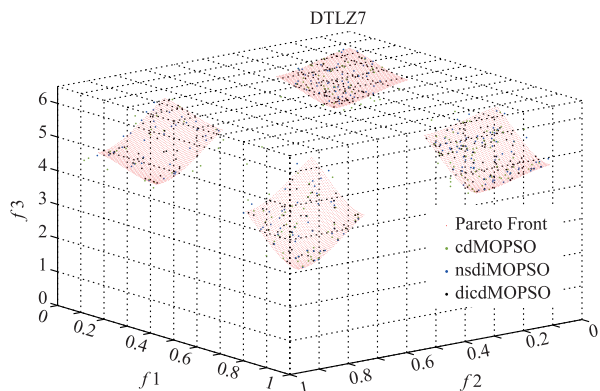


图11 dicdMOPSO、nsdiMOPSO、cdMOPSO对DTLZ7测试结果

表 3 是 dicdMOPSO 算法与其它算法的平均测试时间结果.从表 3 中总体可以看出,在相同的条件下,所比较的算法的测试速度差距不大,但 dicdMOPSO 算法的收敛速度最快(除了 ZDT6 函数和 DTLZ2 函数).

从表 4 中可以看出,与 nsdiMOPSO 算法相比,dicdMOPSO 算法具有更小的 SP 性能,说明在加入了飞行参数调节机制后,dicdMOPSO 算法的多样性更好,且逼近前沿的能力得到加强,同时与其他算法相比,dicdMOPSO 具有更好的 SP 性能(除了 ZDT3 函数时,cdMOPSO 有更好的 SP 值).综合分析数据可以看出,与其他算法相比 dicdMOPSO 能够更好的逼近前沿并且具有更好的收敛速度.

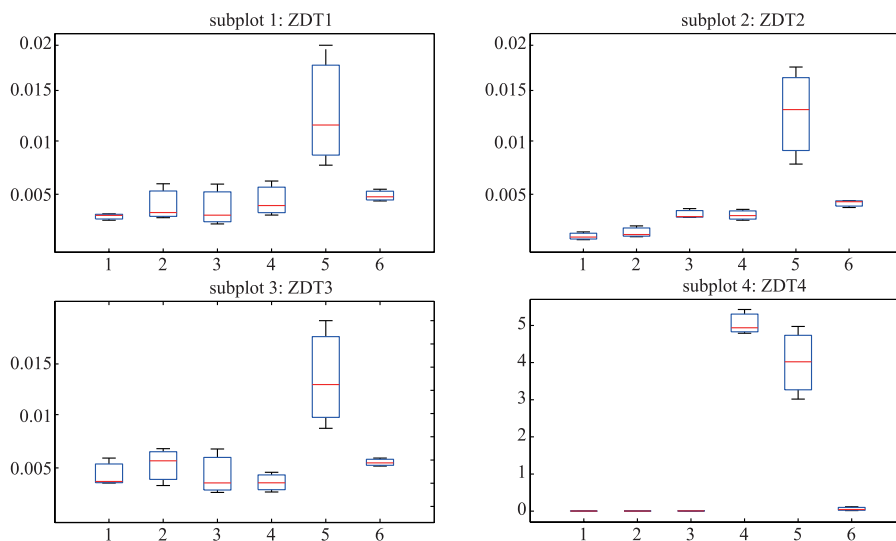


图12 六种算法在分别测试函数ZDT1,ZDT2, ZDT3和ZDT4下得到的IGD盒图比较结果. 1,2,3,4,5,6分别代表 dicdMOPSO, nsdiMOPSO, pccsAMOPSO, cdMOPSO, clusterMOPSO 和 NSGA-II算法

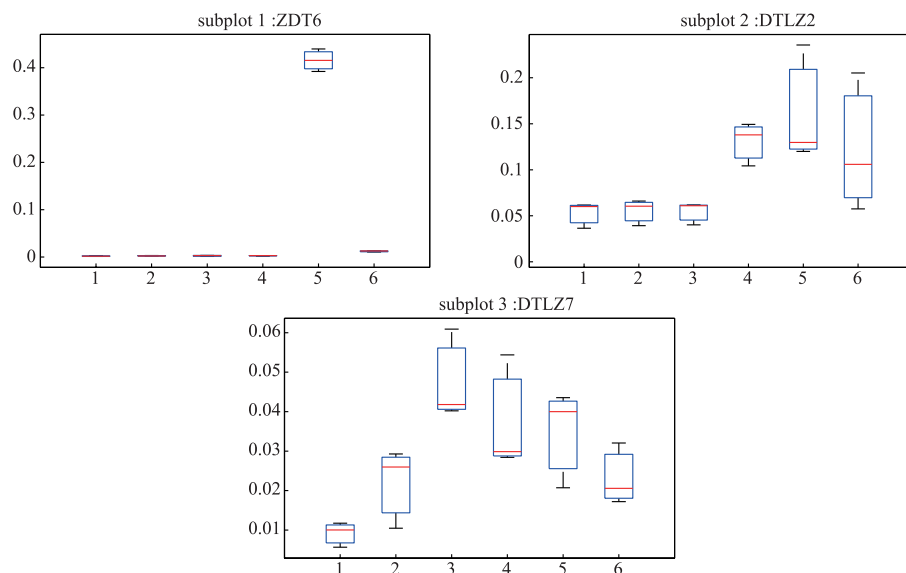


图13 六种算法在分别测试函数ZDT6, DTLZ2和DTLZ7下得到的IGD盒图比较结果. 1,2,3,4,5,6分别代表 dicdMOPSO, nsdiMOPSO, pccsAMOPSO, cdMOPSO, clusterMOPSO 和 NSGA-II算法

表 2 diedMOPSO 算法与其它算法的 IGD 指标

测试函数	IGD 指标	diedMOPSO	nsdiMOPSO	pccsAMOPSO	cdMOPSO	clusterMOPSO	NSGA-II
ZDT1	Best	3.562E-03	3.800E-03	3.206E-03	4.039E-03	8.730E-03	5.367E-03
	Worst	6.153E-03	6.978E-03	6.951E-03	4.937E-02	2.372E-02	6.466E-03
	Mean	4.009E-03	4.280E-03	4.013E-03	4.241E-03	1.250E-02	5.755E-03
	std	5.700E-05	7.800E-05	6.281E-05	2.580E-04	1.784E-03	3.390E-04
ZDT2	Best	1.739E-03	2.048E-03	2.148E-03	3.790E-03	9.580E-03	5.134E-03
	Worst	2.503E-03	3.152E-03	5.054E-03	4.960E-03	2.061E-02	5.806E-03
	Mean	2.053E-03	2.105E-03	4.091E-03	4.280E-03	1.784E-02	5.355E-03
	std	5.200E-05	1.430E-04	4.815E-05	1.140E-04	5.090E-03	2.020E-04
ZDT3	Best	2.040E-03	3.562E-03	3.105E-03	3.109E-03	8.720E-02	5.447E-03
	Worst	6.160E-03	1.032E-02	1.018E-02	2.899E-02	1.983E-01	6.105E-03
	Mean	4.092E-03	5.658E-03	3.323E-03	3.063E-03	1.047E-01	5.834E-03
	std	3.513E-03	6.091E-03	9.951E-05	7.130E-05	7.052E-02	2.020E-04
ZDT4	Best	1.851E-03	3.069E-03	2.053E-03	4.976E+00	3.021E+00	4.623E-03
	Worst	3.855E-03	4.521E-03	9.944E-03	6.361E+00	5.674E+00	1.117E-01
	Mean	3.243E-03	3.609E-03	7.971E-03	5.912E+00	3.989E+00	1.655E-02
	std	2.960E-04	4.050E-04	1.470E-03	4.518E+00	2.616E+00	3.174E-02
ZDT6	Best	9.280E-05	1.135E-04	2.149E-04	8.970E-04	3.966E-01	1.119E-02
	Worst	2.210E-04	3.120E-04	3.674E-03	3.627E-03	4.000E-01	1.498E-02
	Mean	1.670E-04	2.029E-04	3.406E-03	2.988E-03	4.394E-01	1.286E-02
	std	8.600E-05	1.080E-04	2.287E-04	1.543E-04	2.368E-02	1.004E-03
DTLZ2	Best	1.280E-02	1.59E-02	2.025E-02	3.221E-02	1.895E-02	7.831E-02
	Worst	9.871E-02	1.120E-01	9.674E-02	2.067E-01	2.365E-01	2.74E-01
	Mean	6.025E-02	6.569E-02	6.143E-02	1.015E-01	1.265E-01	1.059E-01
	std	6.941E-04	1.080E-03	1.898E-03	1.343E-03	1.683E-02	8.383E-03
DTLZ7	Best	5.911E-03	1.031E-02	3.258E-03	7.015E-03	9.471E-03	6.14E-03
	Worst	1.325E-02	3.120E-02	6.674E-02	5.439E-02	6.295E-02	3.208E-02
	Mean	1.013E-02	2.329E-02	4.277E-02	2.988E-02	4.022E-02	1.799E-02
	std	8.600E-04	1.001E-03	9.506E-04	1.651E-03	2.068E-03	1.294E-03

表 3 diedMOPSO 算法与其它算法的所需时间的平均值对比

算法	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ2	DTLZ7
diedMOPSO	105s	120s	130 s	136s	141s	230s	274s
nsdiMOPSO	94s	116s	120s	131s	122s	210s	260s
cdMOPSO	110s	124s	135s	130s	143s	188s	287s
clusterMOPSO	106s	120s	130s	138s	140s	215s	294s
NSGA-II	138s	146s	159s	140s	149s	250s	300s

表 4 diedMOPSO 算法与其它算法的 SP 指标

测试函数	SP 指标	diedMOPSO	nsdiMOPSO	pccsAMOPSO	cdMOPSO	clusterMOPSO	NSGA-II
ZDT1	Best	1.580E-02	2.216E-02	9.533E-03	7.732E-02	6.575E-02	4.340E-02
	Worst	2.908E-02	3.257E-02	3.742E-02	1.023E-01	8.014E-02	7.538E-02
	Mean	2.453E-02	2.813E-02	2.574E-02	8.561E-02	7.249E-02	5.830E-02
	std	6.830E-04	9.180E-04	3.164E-04	1.425E-02	6.170E-04	9.385E-03
ZDT2	Best	1.799E-02	2.361E-02	1.023E-02	1.097E-02	8.755E-03	6.015E-03
	Worst	3.036E-02	3.719E-02	3.021E-02	2.377E-02	1.798E-02	8.287E-03
	Mean	2.107E-02	2.517E-02	2.322E-02	1.907E-02	1.768E-02	7.241E-03
	std	2.975E-03	5.144E-03	1.987E-03	6.420E-04	1.953E-03	7.410E-04
ZDT3	Best	6.489E-02	6.947E-02	5.974E-02	1.036E-01	4.931E-03	8.157E-02
	Worst	9.827E-02	1.002E-01	8.771E-02	8.745E-01	1.567E-01	1.066E-01
	Mean	7.052E-02	8.453E-02	7.159E-02	5.968E-01	7.784E-02	9.222E-02
	std	6.971E-03	3.523E-02	4.054E-03	2.247E-01	4.362E-02	8.415E-03
ZDT4	Best	2.127E-02	2.913E-02	2.014E-02	1.396E-01	6.578E-02	3.139E-02
	Worst	2.944E-02	4.020E-02	3.145E-02	3.010E-01	1.099E-01	4.425E-02
	Mean	2.767E-02	3.193E-02	2.881E-02	2.046E-01	7.692E-02	3.838E-02
	std	4.760E-04	8.140E-04	5.132E-04	9.556E-02	1.931E-02	3.837E-03
ZDT6	Best	2.918E-02	3.855E-02	1.457E-02	1.240E-02	7.466E-03	6.851E-03
	Worst	4.425E-02	5.002E-02	5.145E-02	4.021E-02	1.732E-02	1.013E-02
	Mean	3.751E-02	5.886E-02	4.024E-02	3.457E-02	9.354E-03	8.266E-03
	std	1.463E-03	6.487E-03	2.523E-03	3.884E-03	4.661E-03	9.180E-04
DTLZ2	Best	7.431E-02	9.557E-02	9.484E-02	9.32E-02	4.39E-02	2.145E-02
	Worst	3.151E-01	7.332E-01	5.002E-01	5.897E-01	8.902E-01	7.314E-01
	Mean	1.997E-01	4.015E-01	3.411E-01	3.562E-01	5.762E-01	4.162E-01
	std	9.732E-03	3.222E-02	1.242E-02	1.772E-02	4.472E-02	3.655E-02
DTLZ7	Best	4.491E-02	4.874E-02	4.059E-02	1.347E-01	7.861E-02	6.32E-02
	Worst	7.219E-01	9.152E-01	8.102E-01	9.307E-01	6.913E-01	7.466E-01
	Mean	3.751E-01	6.554E-01	3.988E-01	5.972E-01	3.902E-01	4.191E-01
	std	7.463E-03	7.932E-02	8.417E-03	2.133E-01	1.293E-02	7.961E-03

## 6 结论

为提高多目标非支配解的多样性和收敛性,文中通过设计知识库更新机制、全局最优选择机制和飞行参数调整机制提出 diedMOPSO 算法.该算法能够平衡了种群的进化过程,提高了非支配解的多样性和收敛性,并且实验结果证明该算法不仅获得了多样性较高的可行解,而且能够较快的收敛到 Pareto 前沿.

尽管 diedMOPSO 算法已经能够获得较高的收敛精度和较好的多样性,但是对于知识库更新机制的研究还需要更加深入,将从收敛性和多样性两方面同时考虑优化解的特性开展知识库更新机制的设计.

## 参考文献

- [1] LI H, LANDA-SILVA D. An adaptive evolutionary multi-objective approach based on simulated annealing[J]. *Evolutionary Computation*, 2011, 19(4): 561-595.
- [2] MARTIN D, ROSETE A, ALCALA-FDEZ J, et al. A new multiobjective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(1): 54-69.
- [3] JARA E C. Multi-objective optimization by using evolutionary algorithms: The-optimality criteria[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(2): 167

- 179.
- [4] ZHANG X, TIAN Y, CHENG R, et al. An efficient approach to nondominated sorting for evolutionary multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(2): 201-213.
- [5] 邵鹏, 吴志健, 周炫余, 等. 基于折射原理反向学习模型的改进粒子群算法[J]. 电子学报, 2015, 43(11): 2137-2144.  
SHAO Peng, WU Zhi-jian, ZHOU Xuan-yu, et al. Improved particle swarm optimization algorithm based on opposite learning of refraction[J]. Acta Electronica Sinica, 2015, 43(11): 2137-2144. (in Chinese)
- [6] 李迎秋, 迟玉红, 温涛. 一种基于动态边界的粒子群优化算法[J]. 电子学报, 2013, 41(5): 865-870.  
LI Ying-qiu, CHI Yu-hong, WEN Tao. A dynamic boundary based particle swarm optimization[J]. Acta Electronica Sinica, 2013, 41(5): 865-870. (in Chinese)
- [7] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [8] 申元霞, 曾传华, 王喜凤, 等. 并行协作骨干粒子群优化算法[J]. 电子学报, 2016, 44(7): 1643-1648.  
SHEN Yuan-xia, ZENG Chuan-hua, WANG Xi-feng, et al. A parallel-cooperative bare-bone particle swarm optimization algorithm[J]. Acta Electronica Sinica, 2016, 44(7): 1643-1648. (in Chinese)
- [9] HELWIG S, BRANKE J, MOSTAGHIM S. Experimental analysis of bound handling techniques in particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(2): 259-271.
- [10] PEHLIVANOGLU Y V. A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(3): 436-452.
- [11] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256-279.
- [12] WANG Y, YANG Y. Particle swarm optimization with preference order ranking for multi-objective optimization [J]. Information Sciences, 2009, 179(12): 1944-1959.
- [13] TSAI S J, SUN T Y, LIU C C, et al. An improved multi-objective particle swarm optimizer for multi-objective problems[J]. Expert Systems with Applications, 2010, 37(8): 5872-5886.
- [14] MUKHOPADHYAY A, MAULIK U, BANDYOPADHYAY S, et al. A survey of multiobjective evolutionary algorithms for data mining; Part I[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(1): 4-19.
- [15] 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.
- [16] OMRAN M, ENGELBRECHT A P, SALMAN A. Particle swarm optimization method for image clustering[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2005, 19(3): 297-321.
- [17] ZHANG Q, LI H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [18] HU W, YEN G G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(1): 1-18.

#### 作者简介



韩红桂 男, 1983年8月出生, 江苏泰州人. 北京工业大学教授、博士生导师. 主要研究领域为城市污水处理过程智能优化控制、计算智能与智能系统等.  
E-mail: recharadhan@bjut.edu.cn



卢薇 女, 1993年8月出生, 山东日照人. 2015年本科毕业于菏泽学院, 并于2015年进入北京工业大学信息学部, 现为硕士研究生. 主要研究领域为污水处理过程智能优化和神经网络.  
E-mail: luweiwei@emails.bjut.edu.cn