

自适应分解式多目标粒子群优化算法

韩红桂^{1,2}, 阿音嘎^{1,2}, 张璐^{1,2}, 乔俊飞^{1,2}

(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优化算法解的分布性, 文中提出了一种自适应分解式多目标粒子群优化算法 (Adaptive Multiobjective Particle Swarm Optimization based on Decomposed Archive, AMOPSO-DA). 首先, 设计了一种基于优化空间分布信息的外部档案更新策略, 有效提升了 AMOPSO-DA 的空间搜索能力; 其次, 提出了一种基于粒子进化方向信息的飞行参数调整方法, 有效平衡了 AMOPSO-DA 的探索 and 开发能力. 最后, 将提出的 AMOPSO-DA 应用于多目标优化问题, 实验结果表明, 文中提出的 AMOPSO-DA 能够获得分布性较好的优化解.

关键词: 多目标粒子群优化算法; 分解; 外部档案; 分布性; 自适应

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2020)07-1245-10

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2020.07.001

Adaptive Multiobjective Particle Swarm Optimization Based on Decomposition Archive

HAN Hong-gui^{1,2}, A Yin-ga^{1,2}, ZHANG Lu^{1,2}, QIAO Jun-fei^{1,2}

(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 distribution performance of multiobjective particle swarm optimization algorithm, an adaptive multiobjective particle swarm optimization algorithm, based on the decomposed archive, named AMOPSO-DA, is developed in this paper. First, an external archive update strategy, based on the spatial distribution information of optimal solutions, is designed to improve the searching ability of AMOPSO-DA. Second, an adaptive flying parameter adjustment strategy, based on the evolutionary direction information of each particle, is proposed to balance the exploration ability and the exploitation ability. Finally, this proposed AMOPSO-DA is applied to some multiobjective optimization problems. The experiment results demonstrate that AMOPSO-DA can obtain well-distributed optimal solutions.

Key words: multiobjective particle swarm optimization algorithm; decomposition approach; external archive; distribution; adaptive

1 引言

多目标优化问题 (Multiobjective Optimization Problems, MOPs) 广泛存在于科学研究和工业生产过程中, 由于 MOPs 中待优化的目标之间往往互相冲突, 其优化过程存在较大的挑战^[1-4]. 为了获得 MOPs 的优化解, 多目标粒子群优化算法 (Multiobjective Particle Swarm Optimization, MOPSO) 由于具有调整参数少、收敛速度快、易于实现等特点, 获得了国内外学者的广泛关注^[5-8].

MOPSO 的性能主要包括收敛性和分布性, 其中, 分

布性是反映 MOPSO 获取优化解多样化程度的性能指标, 关系到优化解集在优化空间中的分布质量^[9-11]. 为了提升 MOPSO 的分布性, Moubayed 等通过删减外部档案中拥挤距离较小的解, 设计了一种基于拥挤距离的 MOPSO, 保证了引导解集的空间分布多样性, 提高了 MOPSO 的分布性^[12]. Coello 等通过优化引导解的分布性, 提出了一种基于网格策略的 MOPSO, 提升了优化解集的分布性^[13]. Han 等通过评估外部档案中优化解集的分布情况, 设计了一种自适应 MOPSO, 实验结果表明利用自适应 MOPSO 获得的优化解集该具有较好的分

收稿日期: 2019-09-24; 修回日期: 2020-03-09; 责任编辑: 孙瑶

基金项目: 国家自然科学基金 (No. 61890930-5, No. 61622301); 国家重点研发计划 (No. 2018YFC1900800-5); 北京高校卓越青年科学家项目 (No. BJJWZYJH01201910005020); 北京市朝阳区协同创新 (No. K6041001201801)

布性^[14]. 然而,上述改进型 MOPSO 需要计算优化解之间的 Pareto 支配关系,以及优化解之间的空间距离,增加了优化过程的计算量,限制了其应用^[15].

为了降低优化过程的计算复杂度, Lin 等设计了一种分解式 MOPSO, 采用分解方法将 MOPs 问题分解为若干单目标优化子问题, 降低了优化过程的计算复杂度^[16]. Hu 等提出了一种基于分解方法和 Pareto 支配策略的 MOPSO, 利用分解方法为种群选择全局最优点, 降低了引导点选择过程中的计算负载^[17]. Dai 等设计了一种多种群搜索策略, 通过使用分解方法将种群按不同区域进行划分, 避免了大规模种群中粒子的比较过程, 有效降低了优化过程的计算复杂度^[18]. 以上分解方法有效降低了 MOPSO 优化过程的计算复杂度, 并获得了分布较好的优化解集. 但是这些算法在优化过程中偏重于保持种群的搜索多样性, 缺乏良好的收敛引导手段, 算法的收敛性不能得到有效保证^[19-21].

为了提高 MOPSO 的分布性, 同时避免上述方法存在的问题, 文中设计了一种自适应分解式多目标粒子群优化算法 (Adaptive Multiobjective Particle Swarm Optimization based on Decomposed Archive, AMOPSO-DA). 首先, 设计了一种基于分解方法的外部档案更新策略, 利用一组均匀分布的方向向量将目标空间分割为具有若干进化方向的子空间, 通过计算优化解与子空间的隶属信息选择外部档案中合适的解; 其次, 设计了一种基于粒子进化方向信息的飞行参数调整方法, 通过平衡算法的探索和开发能力, 提高了 AMOPSO-DA 的收敛性. 最后, 将 AMOPSO-DA 用于标准测试函数和双环管网优化模型, 获得了具有较好分布性的优化解集.

2 多目标粒子群优化算法 (MOPSO)

2.1 多目标优化问题 (MOP)

对于一个具有 M 个优化目标的 MOP, 其目标函数可以描述为

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

式中, \mathbf{x} 是 n 维决策变量, $f_i(\mathbf{x})$ 是第 i 个目标函数, M 是目标函数的个数.

对于一组决策变量 \mathbf{x} 和 \mathbf{y} , 假设待优化问题为最小化目标值, 如果 $f(\mathbf{x})$ 对于任意目标值小于等于 $f(\mathbf{y})$, 且 $f(\mathbf{x})$ 至少存在一个目标小于 $f(\mathbf{y})$, 那么称 \mathbf{x} 支配 \mathbf{y} :

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

式中, $i=1, 2, \dots, M, j=1, 2, \dots, M$; 如果 \mathbf{x} 和 \mathbf{y} 存在支配关系, 那么 \mathbf{x} 和 \mathbf{y} 可以比较优劣, 否则 \mathbf{x} 和 \mathbf{y} 无法比较. 当一个解不被任何解所支配时, 称该解为非支配解, 算法最终获得的全部非支配解, 即为该算法所获得的最优解集.

2.2 多目标粒子群优化算法 (MOPSO)

MOPSO 是利用一组随机分布的种群, 通过选择合

适的解来引导种群对优化空间进行搜索, 获得 MOP 最优解集的一种多目标优化算法. 每次迭代过程中, 粒子的速度和位置更新公式为

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (3)$$

$$\mathbf{v}_i(t+1) = \omega_i \mathbf{v}_i(t) + c_{i,1} R_{i,1} (\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_{i,2} R_{i,2} (\mathbf{g}(t) - \mathbf{x}_i(t)) \quad (4)$$

式中, ω_i 是惯性权重, $c_{i,1}$ 和 $c_{i,2}$ 是学习因子, $R_{i,1}$ 和 $R_{i,2}$ 是随机值, $\mathbf{p}_i(t)$ 是粒子在第 t 次迭代的个体最优位置, $\mathbf{g}(t)$ 是第 t 次迭代的全局最优位置, $\mathbf{x}_i(t)$ 是第 i 个粒子在第 t 次迭代的位置, $\mathbf{v}_i(t)$ 是第 i 个粒子在第 t 次迭代的速度.

3 AMOPSO-DA 算法

AMOPSO-DA 算法主要围绕外部档案更新策略和速度参数更新策略进行改进. 外部档案更新策略是将目标空间分解方法和 Pareto 策略结合, 实现外部档案优化的有效选择, 改善算法的空间搜索能力; 速度参数更新策略主要利用粒子的进化方向信息实现参数的自适应调整, 平衡 AMOPSO-DA 算法的全局探索能力和局部开发能力.

3.1 优化解空间分布信息

给定一组均匀分布的方向向量 $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k\}$, 将目标域 O 均匀分解为 K 个子空间 ($\Omega_1, \Omega_2, \dots, \Omega_k$). 优化解 $s_i(t)$ 的空间分布信息由其位置向量与方向向量夹角 $\langle \boldsymbol{\lambda}(s_i(t)), \mathbf{v} \rangle$ 决定, 其中 $\boldsymbol{\lambda}(s_i(t)) = [\lambda_1(s_i(t)), \lambda_2(s_i(t)), \dots, \lambda_M(s_i(t))]$ 为优化解 $s_i(t)$ 的位置向量:

$$\lambda_m(s_i(t)) = \arccos\left(\frac{|f_m(s_i(t)) - r_m(t)|}{\sqrt{\sum_{j=1}^M (f_j(s_i(t)) - r_j(t))^2}}\right) \quad (5)$$

式中, $m=1, 2, \dots, M, M$ 为目标函数的个数, $R = [r_1, r_2, \dots, r_M]$ 为目标空间上的参考点, $r_m(t) = \min(f_m(S(t)))$. 位置向量满足以下条件:

$$\sum_{m=1}^M \cos^2(\lambda_m(s_i(t))) = 1 \quad (6)$$

如果对于任意方向向量 \mathbf{v}_j , 满足 $\langle \boldsymbol{\lambda}(s_i(t)), \mathbf{v}_k \rangle > \langle \boldsymbol{\lambda}(s_i(t)), \mathbf{v}_j \rangle$, 则第 i 个优化解 $s_i(t)$ 在目标空间上隶属于方向向量 \mathbf{v}_k 所决定的子空间 Ω_k . 图 1 给出了双目标优化问题 $f_1(\mathbf{x})$ 和 $f_2(\mathbf{x})$ 优化解 $s_i(t)$ 的分配示意图.

3.2 外部档案更新策略

为了保证算法在优化空间上的搜索能力, 文中设计了一种基于外部档案中优化解空间分布信息的外部档案更新策略, 该更新策略包括子空间解集分配过程和子空间解集选择过程两部分.

子空间解集分配过程 计算新产生优化解 $s_i(t)$ 与

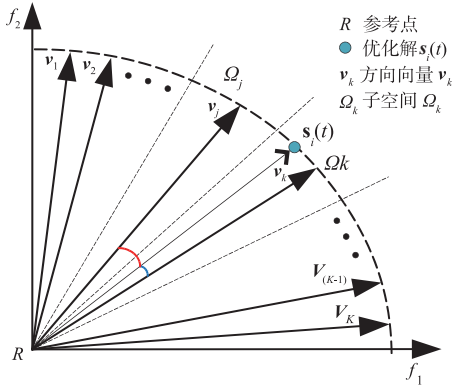


图1 双目标优化问题目标空间上优化解分配示意图

所有给定方向向量的夹角,选择夹角最小的方向向量所定义子空间为归属空间,对所有的优化解进行归属空间分配.根据优化解的空间归属结果,对任意子空间 α ,其所包含的优化解集 $c_\alpha(t)$ 为

$$c_\alpha(t) = \{s_i(t) \mid \langle \lambda(s_i(t)), v_\alpha \rangle \leq \langle \lambda(s_i(t)), v_j \rangle, s_i(t) \in S(t), i=1, 2, \dots, |S(t)|\} \quad (7)$$

式中, v_α 为定义子空间 α 的方向向量, v_j 为其他的任意方向向量, $j=1, 2, \dots, K$ 并且 $j \neq \alpha$. $S(t)$ 为更新之前外部档案中的优化解集.

子空间解集选择过程 为了避免由于外部档案优化解聚集程度不同,导致算法空间搜索能力较差的问题,算法将对子空间优化解集进行筛选,每个子空间的优化解数量上限值为 $\eta(t)$,其中 $\eta(t)$ 定义为

$$\eta(t) = \left\lfloor \frac{R}{H(t)} \right\rfloor \quad (8)$$

式中, R 为给定的外部档案的预设值, $H(t)$ 为当前迭代过程中具有优化解的子空间数量.当算法对目标空间探索不够充分时, $H(t)$ 较小,此时子空间需要保留较多优化解以保证算法在目标空间上进行充分探索;随着优化过程进行, $H(t)$ 逐渐变大,此时子空间需要保留收敛性较好的优化解来保证算法的收敛性.子空间内优化解集的选择规则设定如下:

(1) 当 $c_\alpha(t) > \eta(t)$,利用 Pareto 支配策略依次判断子空间中优化解的支配能力,删除子空间中被支配的优化解,保留非支配解,此时子空间优化解集为 $c'_\alpha(t)$;在删除被支配解后:①如果 $c'_\alpha(t) > \eta(t)$,将比较子空间优化解集 $c'_\alpha(t)$ 中非支配解的位置向量与给定方向向量的夹角大小,保留具有最小夹角的非支配解,直到 $c'_\alpha(t) = \eta(t)$;②如果 $c'_\alpha(t) \leq \eta(t)$,则保留子空间优化解集中剩余的优化解.

(2) 当 $c_\alpha(t) \leq \eta(t)$,说明算法对该子空间的搜索不够充分,或算法满足了优化解的保留条件,子空间中的全部优化解将被保留.

最终,将全部子空间中的优化解集合为一个优化解集 $c'(t)$,作为新的外部档案来指导算法下一代的进化过程.基于外部档案中优化解空间分布信息的外部档案更新策略,其有效性证明如下:

假设候选解 i 在 t 时刻所对应的搜索区域为 $\tau_i(t)$,在 MOPSO 中,区域 $\tau_i(t)$ 的搜索概率定义为 $P_{\tau_i}(t)$;在不采用外部档案分解策略的 MOPSO 中,区域 $\tau_i(t)$ 的搜索概率满足关系

$$P_{\tau_i}(t) \propto R_i(t) \quad (9)$$

其中 $R_i(t)$ 为候选解 i 在 t 时刻对应的搜索区域非支配解密度.而在 AMOPSO-DA 中,区域 $\tau_i(t)$ 的搜索概率满足关系

$$0 < P_{\tau_i}(t) \leq \frac{1}{H(t)} \quad (10)$$

在不采用外部档案分解策略的 MOPSO 中, $P_{\tau_i}(t)$ 随着候选解的密度变化,算法不能充分搜索稀疏区域,因此分布性不能得到有效保证;而在 AMOPSO-DA 中,每个候选解对应搜索区域的搜索概率不高于特定值,保证了搜索过程的平衡性,因此 AMOPSO-DA 能够有效提升 MOPSO 的搜索能力.基于分解法的外部档案更新策略如图 2 所示.

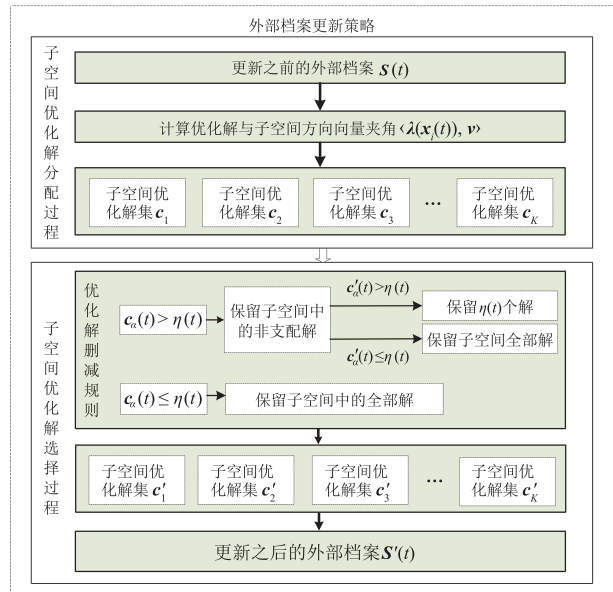


图2 基于优化解空间分布信息的外部档案更新策略

3.3 自适应飞行参数调整机制

在算法搜索过程中,如果粒子 $x_i(t)$ 被 $p_i(t)$ 支配,为了加强粒子的探索能力,则需要提高 ω_i 和 $c_{i,2}$ 的值,降低 $c_{i,1}$ 的值;如果粒子 $x_i(t)$ 支配 $p_i(t)$,为了加强粒子的开发能力,则需要降低 ω_i 和 $c_{i,2}$ 的值,提高 $c_{i,1}$ 的值.基于以上分析,为了适应粒子的搜索特点,文中设计了一种基于粒子进化方向信息的飞行参数调整机制.粒子 $x_i(t)$ 的自适应飞行参数 $\delta_i = [\omega_i, c_{i,1}, c_{i,2}]$ 调整机制为

$$\delta_i(t+1) = \begin{cases} \delta_i(t)\lambda_{i,1}(t), & \mathbf{p}_i(t) > \mathbf{x}_i(t) \\ \delta_i(t)\lambda_{i,2}(t), & \mathbf{p}_i(t) < \mathbf{x}_i(t) \\ \delta_i(t), & \text{otherwise} \end{cases} \quad (11)$$

式中, $\lambda_{i,1}(t)$ 和 $\lambda_{i,2}(t)$ 分别为粒子 $\mathbf{x}_i(t)$ 的探索参数矩阵和开发参数矩阵:

$$\lambda_{i,1}(t) = \begin{bmatrix} P_i(t) + 1 & 0 & 0 \\ 0 & P_i(t) & 0 \\ 0 & 0 & P_i(t) + 1 \end{bmatrix} \quad (12)$$

$$\lambda_{i,2}(t) = \begin{bmatrix} P_i(t) & 0 & 0 \\ 0 & P_i(t) + 1 & 0 \\ 0 & 0 & P_i(t) \end{bmatrix} \quad (13)$$

式中, $P_i(t)$ 为微调参数:

$$P_i(t) = (1 + e^{(\bar{D}(t) - D_i(t))})^{-1} \quad (14)$$

式中, $D_i(t)$ 为粒子 $\mathbf{x}_i(t)$ 与全局引导点 $\mathbf{g}(t)$ 的欧式距离, $\bar{D}(t)$ 为全部粒子与引导点 $\mathbf{g}(t)$ 距离的平均值. 当粒子 $\mathbf{x}_i(t)$ 与 $\mathbf{g}(t)$ 距离较近时, $P_i(t)$ 值较低, 粒子能够在 $\mathbf{g}(t)$ 邻域进行深度开发; 当粒子 $\mathbf{x}_i(t)$ 与 $\mathbf{g}(t)$ 距离较远时, $P_i(t)$ 值较高, 粒子能够充分探索 $\mathbf{g}(t)$ 邻域的优化空间. 探索参数矩阵和开发参数矩阵将随着粒子调和参数的变化进行更新, 根据粒子的飞行特点对粒子速度进行有效调整.

3.4 AMOPSO-DA 算法

基于外部档案更新策略和自适应飞行参数调整策略, 自适应分解式多目标粒子群优化算法 (AMOPSO-DA) 计算流程如下:

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

Step2 计算每个粒子的适应度值. 由式(5)计算优化的方向向量, 由式(7)计算优化解与给定方向向量的角度值并确定优化解归属区域.

Step3 根据外部档案更新策略, 为子空间解集选择合适的优化解, 来更新外部档案.

Step4 从外部档案中任意选择优化解作为种群的 gbest, 并更新每个粒子的 pbest 点.

Step5 根据每个粒子引导点的进化方向影响, 由式(12)~(14)计算进化和系数. 并由式(11)自适应调整粒子的飞行参数.

Step6 按照式(3)和式(4)更新粒子的位置和速度.

Step7 判断算法是否达到终止条件, 如达到, 则退出循环, 输出最终优化解集; 否则返回到 Step2.

在 AMOPSO-DA 的执行过程中, 利用基于优化解空间分布信息的外部档案更新策略保证算法的解集分布性; 同时, 结合粒子的进化方向信息来自适应更新飞行

参数, AMOPSO-DA 能够获得具有较好分布性和收敛性表现的优化解集.

3.5 AMOPSO-DA 的计算复杂度分析

AMOPSO-DA 的计算复杂度分析如下:

设 M 为问题目标个数, N 为种群规模, N_a 为方向向量个数. 对单次迭代过程进行分析, 参考点的计算复杂度为 $O(MN)$; 确定每个优化解空间分布信息的计算复杂度为 $O(MN)$; 子空间解集分配过程的计算复杂度为 $O(N \times N_a)$; 外部档案优化解选择部分的计算复杂度为 $O(M(N \log N))$; 微调参数 $P_i(t)$ 的计算复杂度为 $O(MN)$, 自适应参数调整计算复杂度为 $O(3MN)$. 最终, AMOPSO-DA 的计算复杂度为

$$C(\text{AMOPSO-DA}) = O(M(N \log N) + N \times N_a + 6MN) \quad (15)$$

即 AMOPSO-DA 的计算复杂度为 $O(M(N \log N))$. AMOPSO-DA 与其他 MOPSO (cdMOPSO^[12] 和 MOPSO^[13]) 的计算复杂度如表 1 所示, 根据表 1 的结果可以看出 AMOPSO-DA 具有较小的计算复杂度.

表 1 不同算法的计算复杂度

算法	计算复杂度
AMOPSO-DA	$O(M(N \log N))$
MOPSO ^[13]	$O(MN^2)$
cdMOPSO ^[12]	$O(MN^2)$

4 实验及结果分析

为了充分验证 AMOPSO-DA 算法的有效性, 文中分别将 AMOPSO-DA 算法应用于标准函数测试^[22] 和双环管网模型优化问题^[23], 验证算法的有效性.

4.1 标准函数测试

(1) 评价指标

为了评价算法的求解性能, 采用反世代距离 (IGD)^[24] 和解集间隔指标 (SP)^[25] 来对算法的收敛性和分布性进行比较和评价.

IGD 指标为

$$\text{IGD}(\mathbf{F}^*, \mathbf{F}) = \frac{\sum_{\mathbf{x} \in \mathbf{F}^*} \text{mindis}(\mathbf{x}, \mathbf{F})}{|\mathbf{F}^*|} \quad (16)$$

式中, \mathbf{F}^* 为真实 Pareto 前沿, \mathbf{F} 为算法获得的最优解集, $\text{mindis}(\mathbf{x}, \mathbf{F})$ 为 Pareto 解 \mathbf{x} 和 \mathbf{F} 的最小欧氏距离. IGD 指标能够同时反映优化解的收敛性能和分布性能.

SP 指标为

$$\text{SP} = \sqrt{\frac{1}{q-1} \sum_{u=1}^q (\bar{d} - d_u)^2} \quad (17)$$

式中, d_u 是第 u 个解和其他解的最小欧式距离, q 为非支配解的数量, \bar{d} 为所有 d_u 的均值. SP 指标能够反映优化解的分布性能.

(2) 实验参数

为了验证 AMOPSO-DA 的分布性和收敛性,将 AMOPSO-DA 与基于目标分解的 MOPSO 算法 (dMOPSO)^[16]、基于区域分解的 MOPSO 算法 (MPSO/D)^[18]、采用拥挤距离的 MOPSO 算法 (cdMOPSO)^[12] 及多目标遗传算法 NSGA-II 算法^[26] 在 ZDT1-4、DTLZ2 和 7 测试函数中进行了比较. 测试函数如表 2 所示. 在比较实验

中,各算法的种群大小 N 和外部档案 $S(t)$ 的最大容量均设置为 100,初始惯性权重 ω_i 为 0.5,学习因子 $c_{i,1}$ 和 $c_{i,2}$ 为 1. 所有测试函数的最大迭代次数设置为 200. 对于 AMOPSO-DA,方向向量个数为 100. 对于分解类算法 dMOPSO 和 MPSO/D,参考向量个数设定为 100. 文中对比算法的其他参数与参考文献一致,每种算法均独立运行 30 次取平均值.

表 2 基准测试函数描述

名称	目标函数	维数	说明
ZDT1	$\text{ZDT1} = \begin{cases} \min f_1(\mathbf{x}) = x_1 \\ \min f_2(\mathbf{x}) = g(1 - \sqrt{f_1/g}) \\ g(\mathbf{x}) = 1 + 9 \sum_{i=2}^m x_i / (m - 1) \\ \text{s. t. } 0 \leq x_i \leq 1, i = 1, 2, \dots, 30 \end{cases}$	$m = 30$	凸,连续
ZDT2	$\text{ZDT2} = \begin{cases} \min f_1(\mathbf{x}) = x_1 \\ \min f_2(\mathbf{x}) = g(1 - (f_1/g)^2) \\ g(\mathbf{x}) = 1 + 9 \sum_{i=2}^m x_i / (m - 1) \\ \text{s. t. } 0 \leq x_i \leq 1, i = 1, 2, \dots, 30 \end{cases}$	$m = 30$	凹,连续
ZDT3	$\text{ZDT3} = \begin{cases} \min f_1(\mathbf{x}) = x_1 \\ \min f_2(\mathbf{x}) = g(1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1)) \\ g(\mathbf{x}) = 1 + 9 \sum_{i=2}^m x_i / (m - 1) \\ \text{s. t. } 0 \leq x_i \leq 1, i = 1, 2, \dots, 30 \end{cases}$	$m = 30$	凹,不连续
ZDT4	$\text{ZDT4} = \begin{cases} \min f_1(\mathbf{x}) = x_1 \\ \min f_2(\mathbf{x}) = g(1 - \sqrt{f_1/g}) \\ g(\mathbf{x}) = 1 + 10(m - 1) + \sum_{i=2}^m (x_i^2 - 10\cos(4\pi x_i)) \\ \text{s. t. } 0 \leq x_1 \leq 1, -5 \leq x_i \leq 5, i = 1, 2, \dots, 9 \end{cases}$	$m = 10$	凸,多模态
DTLZ2	$\text{DTLZ2} = \begin{cases} \min f_1(\mathbf{x}) = \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)(1 + g(\mathbf{x})) \\ \min f_2(\mathbf{x}) = \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2)(1 + g(\mathbf{x})) \\ \min f_3(\mathbf{x}) = \sin(\frac{\pi}{2}x_1)(1 + g(\mathbf{x})) \\ g(\mathbf{x}) = \sum_{i=1}^3 (x_i - 0.5)^2 \\ \text{s. t. } 0 \leq x_i \leq 1, i = 1, 2, \dots, 12 \end{cases}$	$m = 10$	凸,连续
DTLZ7	$\text{DTLZ7} = \begin{cases} \min f_j(\mathbf{x}) = \frac{1}{ n/3 } \sum_{i=\lfloor \frac{j-1}{3} \rfloor}^{\lfloor \frac{j}{3} \rfloor} x_i, j = 1, 2, 3 \\ \text{s. t. } g_j(\mathbf{x}) = f_3(\mathbf{x}) + 4f_j(\mathbf{x}) - 1 \geq 0, j = 1, 2 \\ g_3(\mathbf{x}) = 2f_3(\mathbf{x}) + \min_{i,j \neq 1, i \neq j} [f_j(\mathbf{x}) + f_j(\mathbf{x})] - 1 \geq 0 \\ 0 \leq x_i \leq 1, i = 1, 2, \dots, n, n > 3 \end{cases}$	$m = 20$	不连续,多模态

(3) 标准函数测试实验结果及比较分析

将 AMOPSO-DA 与其他算法在两目标测试函数中进行对比分析. 由图 3 ~ 图 6 可以看出, 在 ZDT1-3 测试过程中, AMOPSO-DA 的收敛性比 dMOPSO 和 cdMOPSO 好. 在 ZDT4 测试过程中, AMOPSO-DA 的分布性较 dMOPSO 差, 但 AMOPSO-DA 具有较好的收敛性. 同时, 由图 7 图 8 可以看出, 在 DTLZ2 测试过程中, AMOPSO-DA 具有良好的收敛性和分布性; 在 DTLZ7 测试过程中, AMOPSO-DA 的多样性优于 cdMOPSO, 收敛性优于 dMOPSO.

表 3 为 AMOPSO-DA 与其他算法的 IGD 指标比较结果. 表 3 的结果显示, 在 ZDT1-4, DTLZ2 和 DTLZ7 测试过程中, AMOPSO-DA 的收敛性优于 dMOPSO 和 MPSO/D. 实验结果说明, 采用了局部 Pareto 支配策略的 AMOPSO-DA, 其外部档案中的优化解具有较好的收敛引导能力, 算法的收敛性表现较 dMOPSO 和 MPSO/D 更加良好. 在与 cdMOPSO 和 NSGA-II 进行比较时, AMOPSO-DA 也具有较好的收敛表现, 特别是在 ZDT4 这种具有多个局部前沿的

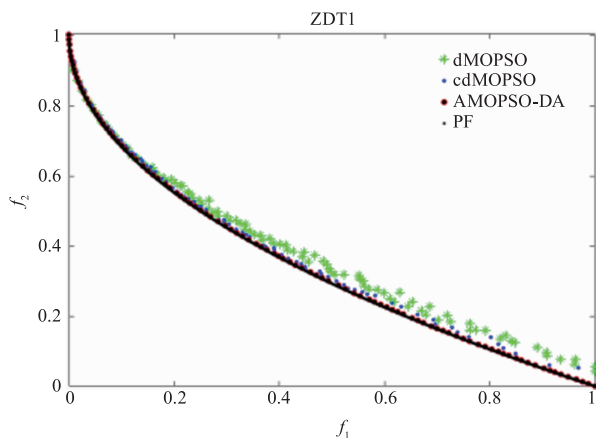


图3 dMOPSO、cdMOPSO、AMOPSO-DA对ZDT1测试

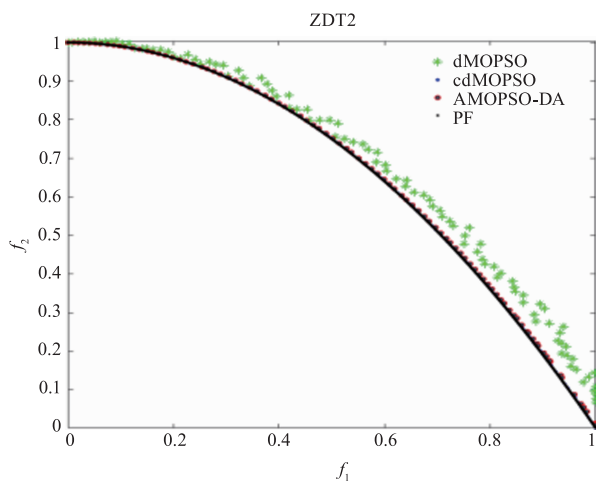


图4 dMOPSO、cdMOPSO、AMOPSO-DA对ZDT2测试

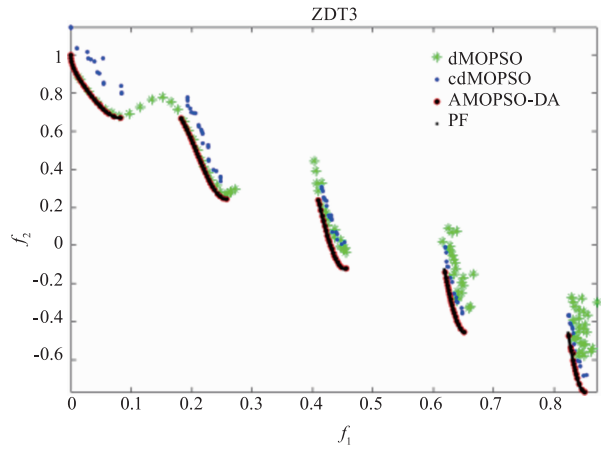


图5 dMOPSO、cdMOPSO、AMOPSO-DA对ZDT3测试

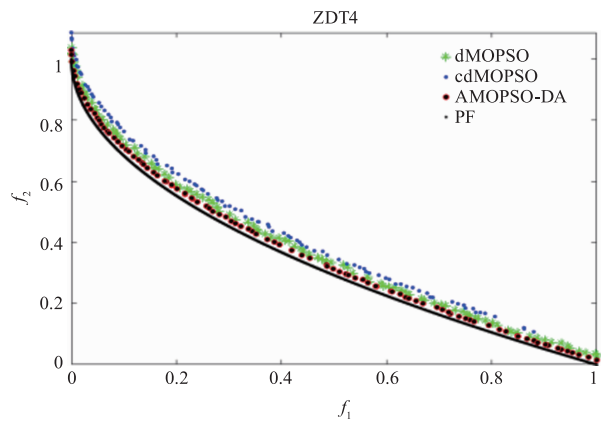


图6 dMOPSO、cdMOPSO、AMOPSO-DA对ZDT4测试

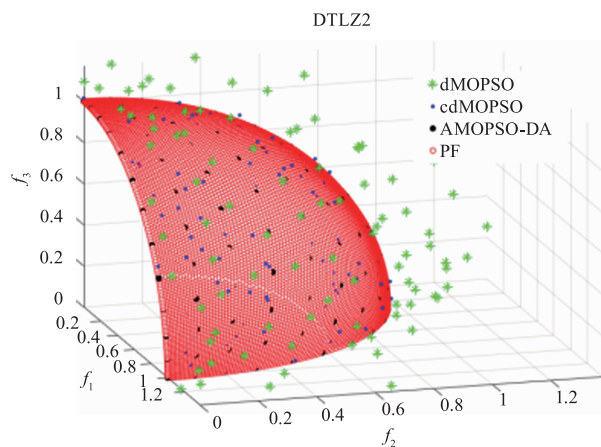


图7 dMOPSO、cdMOPSO、AMOPSO-DA对DTLZ2测试

测试函数中, AMOPSO-DA 的平均 IGD 值达到了 3.209E-03, 优于 cdMOPSO. 该试验结果说明, AMOPSO-DA 在保留搜索空间上分布性较好的优化解的同时, 采用自适应的飞行参数调整策略, 使得算法具有较好的空间探索能力和开发能力, 提高了算法的收敛精度.

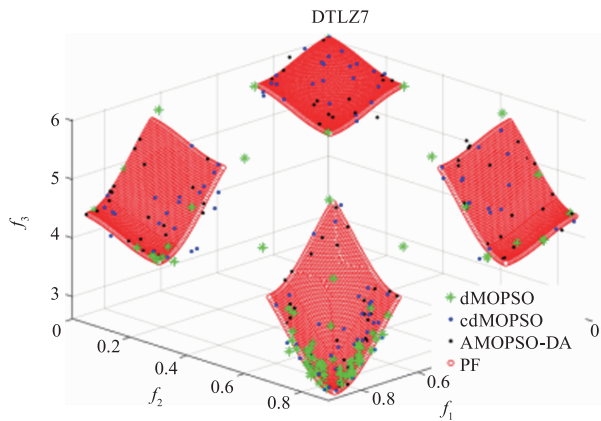


图8 dMOPSO、cdMOPSO、AMOPSO-DA对DTLZ7测试

表 4 为 AMOPSO-DA 与其他算法的 SP 指标比较结果. 表 4 的结果显示, 与 MPSO/D 相比, AMOPSO-DA 在 ZDT4 和 DTLZ2 测试过程中优化解集分布性表

现一般,但是在其他函数测试过程中都获得了较好的分布性表现. 在与 cdMOPSO 和 NSGA-II 的比较中, AMOPSO-DA 的优化解集有着较好的分布性表现,特别是在 ZDT3 测试过程中, AMOPSO-DA 的 SP 值为 7.052E-01, 优于 cdMOPSO 的 9.968E-01 和 NSGA-II 的 9.222E-01. 因此, AMOPSO-DA 具有较好的前沿逼近能力和解集分布性.

4.2 AMOPSO-DA 双环管网测试

为测试 AMOPSO-DA 在解决实际优化问题中的有效性,将 AMOPSO-DA 应用于双环管网模型优化问题. 在双环管网模型优化的测试中,采用 MPSO/D 及 MOPSO 作为 AMOPSO-DA 的对比算法,其中, AMOPSO-DA、MPSO/D 及 MOPSO 算法的初始种群规模设定为 200, 最大迭代次数设定为 300; AMOPSO-DA 的方向向量个数为 200, MPSO/D 的参考向量个数为 200; 所有算法的初始惯性权重 ω_i 设置为 0.5, 学习因子 $c_{i,1}$ 和 $c_{i,2}$ 设置为

表 3 AMOPSO-DA 算法与其它算法的 IGD 指标

测试函数	IGD 指标	AMOPSO-DA	dMOPSO	MPSO/D	cdMOPSO	NSGA-II
ZDT1	Best	3.764E-03	4.004E-03	5.206E-03	4.039E-03	5.367E-03
	Worst	4.153E-03	6.978E-03	2.951E-02	6.937E-02	9.466E-03
	Mean	3.991E-03	4.783E-03	2.013E-02	5.241E-02	6.645E-03
	Std.	8.774E-05	7.576E-05	6.281E-03	6.665E-04	3.390E-04
ZDT2	Best	2.739E-03	4.048E-03	4.148E-03	5.790E-03	5.134E-03
	Worst	4.335E-03	8.152E-03	3.054E-02	1.960E-02	5.806E-03
	Mean	3.179E-03	5.105E-03	7.091E-03	7.280E-03	5.355E-03
	Std.	5.200E-05	3.430E-04	2.815E-04	1.140E-03	2.020E-04
ZDT3	Best	4.241E-03	5.562E-03	5.105E-03	7.109E-03	5.447E-03
	Worst	6.160E-03	4.032E-02	1.018E-02	2.899E-02	6.105E-02
	Mean	4.172E-03	7.658E-03	6.323E-03	8.063E-03	7.834E-03
	Std.	3.513E-03	6.091E-03	9.951E-05	7.130E-05	2.020E-04
ZDT4	Best	3.051E-03	3.069E-03	8.053E-03	4.976E-02	4.623E-03
	Worst	3.857E-03	4.521E-03	3.944E-02	6.361E-01	1.117E-01
	Mean	3.209E-03	3.443E-03	9.971E-03	5.912E-01	1.655E-02
	Std.	2.960E-04	4.050E-04	1.470E-03	4.518E-04	3.174E-02
DTLZ2	Best	1.280E-02	1.59E-02	4.025E-02	3.221E-02	7.831E-02
	Worst	9.871E-01	1.120E-01	3.674E-01	1.067E-01	2.74E-01
	Mean	4.025E-02	8.569E-02	7.143E-02	6.015E-02	1.059E-01
	Std.	6.941E-04	1.080E-03	1.898E-03	1.343E-03	8.383E-03
DTLZ7	Best	5.911E-03	6.031E-02	3.258E-02	7.015E-02	6.14E-02
	Worst	6.325E-02	8.120E-01	1.674E+00	5.439E-01	3.208E-01
	Mean	3.013E-02	5.329E-01	7.277E-01	8.988E-02	1.799E-01
	Std.	8.600E-04	1.001E-03	9.506E-03	1.651E-03	1.294E-03

表 4 AMOPSO-DA 算法与其它算法的 SP 指标

测试函数	SP 指标	AMOPSO-DA	dMOPSO	MPSO/D	cdMOPSO	NSGA-II
ZDT1	Best	1.004E-02	1.216E-02	9.533E-03	3.732E-02	4.340E-02
	Worst	1.908E-02	2.557E-02	3.742E-02	4.023E-02	7.538E-02
	Mean	1.107E-02	2.413E-02	2.274E-02	3.861E-02	5.830E-02
	Std.	6.830E-04	9.180E-04	9.164E-04	1.425E-02	9.385E-04
ZDT2	Best	1.799E-02	2.361E-02	1.023E-02	1.097E-02	6.015E-02
	Worst	3.036E-02	3.719E-02	3.021E-02	7.377E-02	8.287E-02
	Mean	2.107E-02	2.517E-02	2.231E-02	3.907E-02	7.241E-02
	Std.	2.975E-03	5.144E-03	1.987E-03	6.420E-04	7.410E-04
ZDT3	Best	5.489E-01	6.947E-01	5.974E-01	8.036E-01	8.157E-01
	Worst	8.827E-01	9.002E-01	8.771E-01	1.145E+00	1.066E+00
	Mean	7.052E-01	7.453E-01	7.259E-01	9.968E-01	9.222E-01
	Std.	6.971E-03	3.523E-02	4.054E-03	2.247E-03	8.415E-03
ZDT4	Best	2.127E-02	2.413E-02	2.014E-02	1.396E-01	3.139E-02
	Worst	2.944E-02	4.020E-02	3.145E-02	3.010E-01	4.425E-02
	Mean	2.867E-02	2.563E-02	2.181E-02	2.646E-01	3.838E-02
	Std.	4.760E-04	8.140E-04	5.132E-03	9.556E-02	3.837E-03
DTLZ2	Best	7.431E-02	9.557E-02	7.484E-02	9.32E-02	2.145E-02
	Worst	3.151E-01	4.332E-01	3.002E-01	5.897E-01	7.314E-01
	Mean	2.205E-01	2.715E-01	1.997E-01	3.562E-01	4.162E-01
	Std.	9.732E-03	3.222E-03	9.242E-03	1.772E-03	3.655E-04
DTLZ7	Best	4.491E-02	4.874E-02	4.059E-02	1.347E-01	6.32E-02
	Worst	7.219E-01	9.152E-01	8.102E-01	9.307E-01	7.466E-01
	Mean	2.751E-01	4.554E-01	2.988E-01	4.972E-01	4.191E-01
	Std.	7.463E-03	7.932E-03	8.417E-03	2.133E-03	7.961E-03

1. 每个算法均独立运行 10 次并取平均值. 双环管网的模型如图 9 所示.

双环管网的优化目标表达式为

$$\begin{cases} \min f_1 = \sum_{j=1}^R (a + bDd_j^\alpha)L_j, & j = 1, 2, \dots, R \\ \min f_2 = \sum_{i=1}^I (I_{si} - \bar{I}_s)^2, & i = 1, 2, \dots, I \end{cases}$$

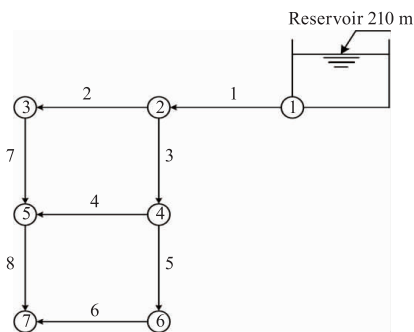


图9 双环管网模型图

$$\text{s. t. } Aq + Q = 0, Lh = 0, h = sq^n,$$

$$Dd_k \in \{Dd_1, Dd_2, \dots, Dd_z\} \quad (18)$$

式中, f_1 是经济目标函数; R 是管道总数, Dd_j 是第 j 根管道直径, $j = 1, 2, \dots, R$, L_j 是第 j 个管道长度, a, b 分别是两个统计常数, α 是指标; f_2 是可靠性目标函数, I 为管网系统节点数目, I_{si} 为第 i 个节点的富余水头值, \bar{I}_s 为节点富余水头的均值.

图 10 给出了双环管网管道造价与节点富余水头方差的优化结果. 从结果中可以看出, 采用 AMOPSO-DA 对双环管网进行优化, 所得的优化解集具有较好的分布性, 能够满足决策者根据不同的管网可靠性需求选择合理的管道造价方案; 同时, 表 5 给出了在相同节点富余水头方差情况下, 不同算法所得到的管网造价方案. 从表 5 可以看出, 相比于 MPSO/D 和 MOPSO, AMOPSO-DA 可以获得更为经济的管道费用方案.

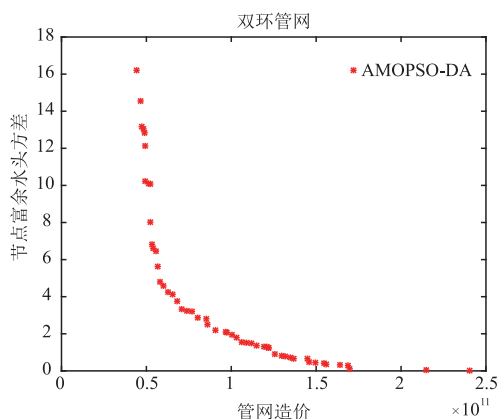


图10 双环管网的测试结果

表 5 在相同节点富余水头方差下双环管网的费用

类型	节点富余水头方差	造价(元)		
		AMOPSO-DA	MPSO/D	MOPSO
双环管网	7	5.127E+10	6.174E+10	6.796E+10
	6	5.646E+10	6.851E+10	7.232E+10
	5	6.124E+10	7.199E+10	8.471E+10
	4	6.341E+10	7.732E+10	8.896E+10
	3	7.511E+10	8.467E+10	1.726E+11
	2	8.147E+10	1.205E+11	3.857E+11

5 结论

为了改善 MOPSO 优化解的分布性,文中提出了一种自适应分解式多目标粒子群优化算法 (AMOPSO-DA). 对比其他算法, AMOPSO-DA 的主要优势包括以下几个方面:(1) 基于优化解的空间分布信息所设计的外部档案更新策略能够有效平衡外部档案的收敛性和多样性,从而改善算法的空间搜索能力;(2) 基于引导点方向信息的飞行参数调整策略能够平衡算法的探索和开发能力,从而提高算法的收敛性. 实验结果表明,在标准函数测试和双环管网优化问题上, AMOPSO-DA 均可以获得具有良好分布性的优化解,并且具有较低的计算复杂度和较好的收敛表现.

尽管 AMOPSO-DA 在解决复杂多目标问题时能够获得分布均匀的优化解集,但是在解决具有不连续前沿的优化问题时,目标空间上多余的方向向量会造成不必要的计算消耗,影响算法的计算效率. 如何探测前沿的分布信息以分配合理的方向向量数量是未来的一个研究重点.

参考文献

[1] 黎明,黄珊,陈昊,等. 基于类圆映射的高维多目标可视化方法[J]. 电子学报,2019,47(6):1185-1193.

- LI M, HUANG S, CHEN H, et al. The quasi-circular mapping visualization for many-objective[J]. Acta Electronica Sinica, 2019, 47(6): 1185-1193. (in Chinese)
- [2] 严峻坤,陈林,刘宏伟,等. 基于机会约束的 MIMO 雷达多波束稳健功率分配算法[J]. 电子学报,2019,47(6): 1230-1235.
- YAN J-K, CHEN L, LIU H-W, et al. Chance constrained based robust multibeam power allocation algorithm for MIMO radar[J]. Acta Electronica Sinica, 2019, 47(6): 1230-1235. (in Chinese)
- [3] 张磊,毕晓君,王艳娇. 基于重新匹配策略的 ϵ 约束多目标分解优化算法[J]. 电子学报,2018,46(5):1032-1040.
- ZHANG L, BI X-J, WANG Y-J. The ϵ constrained multi-objective decomposition optimization algorithm based on re-matching strategy [J]. Acta Electronica Sinica, 2018, 46(5): 1032-1040. (in Chinese)
- [4] ZHAO S Z, IRUTHAYARAJAN M W, BASKAR S, et al. Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization [J]. Information Sciences, 2011, 181(16): 3323-3335.
- [5] HU W, YEN G G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system [J]. IEEE Transactions on Evolutionary Computation, 2013, 19(1): 1-18.
- [6] YUE C T, QU B Y, LIANG J. A multiobjective particle swarm optimization using ring topology for solving multimodal multiobjective problems [J]. IEEE Transactions on Evolutionary Computation, 2018, 22(5): 805-817.
- [7] HU M, WU T, WEIR J D. An adaptive particle swarm optimization with multiple adaptive methods [J]. IEEE Transactions on Evolutionary Computation, 2012, 17(5): 705-720.
- [8] 刘兆广,纪秀花,刘云霞. 一种快速收敛的非参数粒子群优化算法及其收敛性分析[J]. 电子学报,2018,46(7): 1669-1674.
- LIU Z-G, JI X-H, LIU Y-X. A non-parameter particle swarm optimization algorithm with fast convergence speed and its stability analysis [J]. Acta Electronica Sinica, 2018, 46(7): 1669-1674. (in Chinese)
- [9] DANESHYARI M, YEN G G. Cultural-based multiobjective particle swarm optimization [J]. IEEE Transactions on Systems, Man and Cybernetics, 2011, 41(2): 553-568.
- [10] CHEN W N, ZHANG J, LIN Y, et al. Particle swarm optimization with an aging leader and challengers [J]. IEEE Transactions on Evolutionary Computation, 2012, 17(2): 241-258.
- [11] SIERRA M R, COELLO C A C. Multi-objective particle swarm optimizers: A survey of the state-of-the-art [J]. International Journal of Computational Intelligence Research, 2012, 42(2): 287-308.

- [12] MOUBAYED N A I, PETROVSKI A, MCCALL J. D2-MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution [J]. *Evolutionary Computation*, 2014, 22(1):47–77.
- [13] 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.
- [14] HAN H G, LU W, QIAO J F. An adaptive multiobjective particle swarm optimization based on multiple adaptive methods [J]. *IEEE Transactions on Cybernetic*, 2017, 47(9):2754–2767.
- [15] ZHANG Y, GONG D, et al. A decomposition-based archiving approach for multi-objective evolutionary optimization [J]. *Information Sciences*, 2018, 430(5):397–413.
- [16] LIN Q, LI J, DU Z, et al. A novel multi-objective particle swarm optimization with multiple search strategies [J]. *European Journal of Operational Research*, 2015, 247(3):732–744.
- [17] HU Z Y, YANG J M, CUI H H, et al. Multi-objective particle swarm optimization algorithm based on leader combination of decomposition and dominance [J]. *Journal of Intelligent & Fuzzy Systems*, 2017, 33(3):1577–1588.
- [18] DAI C, WANG Y P, YE M. A new multi-objective particle swarm optimization algorithm based on decomposition [J]. *Information Science*, 2015, 325(20):541–557.
- [19] CAI X, LI Y, FAN Z, et al. An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2014, 19(4):508–523.
- [20] TRIVEDI A, SRINIVASAN D, SANYAL K, et al. A survey of multiobjective evolutionary algorithms based on decomposition [J]. *IEEE Transactions on Evolutionary Computation*, 2016, 21(3):440–462.
- [21] YANG S, JIANG S, JIANG Y. Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes [J]. *Soft Computing*, 2017, 21(16):4677–4691.
- [22] RINI D P, SHAMSUDDIN S M, YUHANIX S S. Particle swarm optimization: technique, system and challenges [J]. *International Journal of Computer Applications*, 2011, 14(1):19–26.
- [23] FU G, KAPELAN Z, REED P. Reducing the complexity of multiobjective water distribution system optimization through global sensitivity analysis [J]. *Journal of Water Resources Planning and Management*, 2011, 138(3):196–207.
- [24] JIANG S, YANG S. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts [J]. *IEEE Transactions on Cybernetics*, 2015, 46(2):421–437.
- [25] LI M, YANG S, LIU X. Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2015, 20(5):645–665.
- [26] DEB K, PRATAP A, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2):182–197.

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、神经网络结构设计及优化等.

E-mail: rechardhan@bjut.edu.cn



阿音嘎 男, 1994 年 9 月出生, 内蒙古锡林浩特人. 2016 年本科毕业于江南大学. 现为北京工业大学信息学部硕士研究生. 主要研究领域为多目标进化算法和城市污水处理过程智能优化.

E-mail: ayinga1994@sina.com



张璐 女, 1991 年 10 月出生, 山东枣庄人. 2014 年本科毕业于菏泽学院. 现为北京工业大学信息学部博士研究生. 主要研究领域为城市污水处理过程智能优化控制.

E-mail: zhulu1991@163.com



乔俊飞 男, 1968 年 11 月出生, 内蒙古人. 北京工业大学信息学部教授. 主要研究领域为城市污水处理过程智能检测与智能优化控制等.

E-mail: jisibox@sina.com