

一种基于角度信息的约束高维多目标进化算法

刘冰洁¹, 毕晓君²

(1. 哈尔滨工程大学信息与通信工程学院, 黑龙江哈尔滨 150001; 2. 中央民族大学信息工程学院, 北京 100081)

摘要: 目前约束高维多目标进化算法大多注重提高收敛精度, 而收敛速度相对较慢. 为提高算法的收敛速度, 提出一种基于角度信息的约束高维多目标进化算法. 该算法提出基于角度违反度函数的选择操作, 依据动态的收敛性和分布性直接选择较优个体, 提高收敛速度; 此外, 提出了基于差分进化算法的交叉操作, 在不同的进化阶段选用不可行解参与交叉操作, 补偿收敛精度. 在标准测试函数集 C-DTLZ 上进行仿真实验, 并与当前国内外性能优异的 4 种约束高维多目标进化算法进行对比, 证明了本文算法收敛精度保持良好, 而收敛速度得到了提升, 且目标维数越高提升效果越明显.

关键词: 约束高维多目标优化; 角度违反度; 选择操作; 差分算法; 交叉操作

中图分类号: TP18

文献标识码: A

文章编号: 0372-2112(2021)11-2208-09

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20201044

A Constrained Many-Objective Evolutionary Algorithm Based on Angle Information

LIU Bing-jie¹, BI Xiao-jun²

(1. College of Information and Communication Engineering, Harbin Engineering University, Harbin, Heilongjiang 150001, China;

2. Department of Information Engineering, Minzu University of China, Beijing 100081, China)

Abstract: Most of the current constrained many-objective evolutionary algorithms focus on the convergence accuracy, but the convergence speed is relatively slow. In order to improve the convergence speed, a constrained many-objective evolutionary algorithm based on angle information (CMaOEA-AI) is proposed. In the algorithm, a selection operation based on the angle violation function is proposed to improve the convergence speed, which directly selects the superior individuals according to the dynamic convergence and diversity. Thereafter a crossover operation based on the differential evolutionary algorithm is proposed, which can select the infeasible solutions to participate in the crossover operation at different evolutionary stages. Simulation experiments are performed on the standard test function sets C-DTLZ. Compared with four state-of-the-art constrained many-objective evolutionary algorithms, the proposed algorithm shows good convergence accuracy while the convergence speed is greatly improved, and the higher the objective dimension, the better the effect.

Key words: constrained many-objective optimization; angle violation function; selection operation; differential evolution algorithm; crossover operation

1 引言

随着科学技术的高速发展, 约束高维多目标优化问题已成为当前迫切需要解决的一个难题, 它涉及航空航天、网络通信、机械设计、作业调度和决策科学等诸多应用领域^[1-4]. 在约束高维多目标的实际问题中, 由于目标维数的增加和约束条件的限制, 目标空间的拓扑结构变得十分复杂, 可行域可能十分狭小或者互不连通, 这使得算法很难跨越不可行域在高维目标空

间中找到同时兼顾收敛性和分布性的最优可行解. 为此, 目前大多算法只能牺牲收敛速度来确保收敛精度, 但这样又影响了算法在实际问题中的应用, 因此在保证算法收敛精度的情况下如何提高收敛速度是约束高维多目标优化领域中一个重要的研究课题^[5].

早期的约束处理技术多是针对约束单目标和约束多目标优化问题^[6], 如罚函数法^[7]、Deb 准则^[8]、随机排序法^[9]、 ϵ 约束准则^[10]和多目标优化法^[11]. 目前, 约束

高维多目标进化算法(Constrained Many-Objective Evolutionary Algorithms, CMAOEAs)多采用Deb准则^[12-15]和 ε 约束准则^[16-19]. Deb准则强调可行解优于不可行解,致使有限大小的种群极易陷入局部最优或局部可行域:文献[14]同时提出了两个各有侧重的代表性算法且均采用Deb准则,其中C-NSGA-III继承了NSGA-III^[15]在求解高维多目标优化问题时较好的收敛性和分布性;C-MOEA/D^[14]将高维多目标转换成单目标进行优化,提高了收敛速度,但这两种算法在进化过程中强调可行解占优,而忽略了不可行解,影响了算法的收敛精度和收敛速度. ε 约束法通过设定 ε 值将不可行解引入到进化中:文献[17]提出的C-MOEA/DD通过综合考虑不可行解的约束违反度和小生境数,将与孤立区域相关联的不可行解保留下来,这有助于种群逃离局部可行域,但其在匹配选择中仍优先选择可行解,导致对分布性良好的不可行解利用不够充分,在一定程度上影响了算法的收敛精度;文献[18]提出的C-TAEA通过两个相互协作的归档集CA和DA,实现将种群推向Pareto最优前沿面(Pareto Front, PF)的同时探索未开发区域,在收敛精度上取得了较好的效果,但其每次迭代需要更新两个种群,影响了收敛速度.

基于约束支配的非支配排序是目前CMAOEAs采用的一类主流的选择操作,是影响算法收敛速度的关键,且随着目标维数的增加其所需时间将呈指数增长,而分布性和收敛性保持机制的加入,进一步增加了选择操作需要的时间.综合分析,现有算法虽然提高了收敛精度,但降低了收敛速度,影响了在实际问题中的应用.针对上述问题,本文提出了一种基于角度信息的约束高维多目标进化算法(Constrained Many-objective Evolutionary Algorithm Based on Angle Information, CMAOEA-AI).为了提高收敛速度,提出了一种基于角度违反度函数的选择策略,给出了更精准的分布性度量公式和更快速的收敛性度量公式,省略非支配排序过程,直接从种群中选择较优个体;此外,为了补偿选择操作降低的收敛精度,提出了基于差分进化算法(Differential Evolution, DE)的交叉操作,设计了动态的比例因数,在不同的进化阶段有针对性地让部分不可行解参与交叉.

2 相关背景知识

2.1 约束高维多目标优化问题的数学描述

不失一般性,以最小化问题为例,一个具有 n 个决策变量, M 个目标函数以及 $J+K$ 个约束条件的约束优化问题的数学描述如式(1)所示^[1].

$$\begin{aligned} & \underset{\mathbf{x} \in R^n}{\text{minimize}} \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))^T \\ & \text{subject to} \quad g_j(\mathbf{x}) \geq 0, \quad j = 1, \dots, J \\ & \quad \quad \quad h_k(\mathbf{x}) = 0, \quad k = 1, \dots, K \end{aligned} \quad (1)$$

当 $M > 3$ 时,称式(1)为约束高维多目标优化问题.为衡量候选解 \mathbf{x} 对约束条件的满足程度,定义了约束违反度函数,如式(2)所示^[1],其值称为约束违反度.

$$G(\mathbf{x}) = \sum_{j=1}^J \max(0, g_j(\mathbf{x})) + \sum_{k=1}^K \max(0, |h_k(\mathbf{x})|) \quad (2)$$

约束违反度为0,说明候选解 \mathbf{x} 是可行解.约束违反度大于0,说明候选解 \mathbf{x} 是不可行解.

2.2 ε 约束准则

ε 约束准则^[20]通过设置水平参数 ε 将种群中约束违反度小于 ε 的不可行解当作可行解,让其参与进化,是目前效果较好的一种约束处理技术.水平参数 ε 的设置具体如式(3)所示^[20].

$$\varepsilon(t) = \begin{cases} \varepsilon(0) \times \left(1 - t / (0.5 \times G_{\max})\right)^2, & t < 0.5 \times G_{\max} \\ 0, & \text{其他} \end{cases} \quad (3)$$

$$\varepsilon(0) = G(\mathbf{x}_\theta)$$

式中, \mathbf{x}_θ 是种群中个体根据约束违反度升序排列后第 θ ($\theta = 0.05 \times N$, N 为种群规模)个个体, t 为当前迭代次数, G_{\max} 为最大进化迭代次数.随着进化迭代次数的不断增加, ε 逐渐减小直至为零从而保证种群最后收敛到可行的PF.

3 基于角度信息的约束高维多目标进化算法

CMAOEA-AI设计一种直接基于收敛性和分布性动态均衡的个体选择策略,其中收敛性方面采用一种快速的收敛性准则,虽然该准则对个体Pareto支配关系的反映不够精准,从而降低收敛精度,但计算简单,有助于提高收敛速度.在分布性方面,采用一种基于角度信息的分布性度量准则,有助于提高种群的分布性,补偿收敛精度.为进一步弥补选择操作带来的收敛精度的下降,本文在交叉操作中引入DE算法的思想,设计了动态比例因数,在不同的进化阶段选用不可行解参与交叉,保证算法的收敛精度.

3.1 基于收敛性和分布性的选择操作

为了动态考虑不同阶段的分布性和收敛性,这里构造了角度违反度函数(Angle Violation Function, AVF)作为直接选择个体的筛选准则,其表达式如式(4)所示.

$$AVF(\mathbf{x}) = \lambda(t) \times \cos(\alpha(\mathbf{x})) + (1 - \lambda(t)) \times d(\mathbf{x}) \quad (4)$$

$$\lambda(t) = 1 - t/G_{\max}$$

式中, t 为当前迭代次数, G_{\max} 为最大迭代次数, $\alpha(\mathbf{x})$ 为解的分布性度量, 其值越大说明解的分布性越好. $d(\mathbf{x})$ 为解的收敛性度量, 采用一个快速的收敛性准则——个体的 2 范数距离, 按文献[21]进行计算, 其值越小说明解的收敛性越好. 由于 $\alpha(\mathbf{x})$ 是角度值, 为使式中 $\alpha(\mathbf{x})$ 与 $d(\mathbf{x})$ 的取值一致, 对 $\alpha(\mathbf{x})$ 进行余弦变换, 保证式(4)中的两部分都是取值越小越好. $\lambda(t)$ 是迭代次数的函数, 可以动态地控制不同进化阶段收敛性和分布性的权重, 研究表明, 在进化的初期分布性比收敛性重要, 在进化的后期收敛性更重要^[14], $\lambda(t)$ 随迭代次数的增大而逐渐减小, 保证了在进化初期种群的分布性更重要, 在进化后期收敛性更重要, 从而更精准地选择各阶段适合的个体.

在式(4)中, $\alpha(\mathbf{x})$ 为本文设计的基于角度信息的分布性函数. 高维多目标优化问题的近期研究成果证明了个体间的夹角更能准确地评价其相似性和邻域关系^[21-25], 并已收到较好的效果, 但尚未推广到约束高维多目标优化问题中, 因此不可行解的角度信息对种群分布性的影响仍需进一步论证. 如图 1 所示, 不可行解 A 和 B 与已搜索到的可行解夹角的最小值 α_A 和 α_B 较小, 说明其与周围的可行解比较相似; 不可行解 D 和 E 与已搜索到的可行解夹角的最小值 α_D 和 α_E 较大, 说明其与周围的可行解有较大差异, 更有助于探索到目前尚未开发的可行域. 不可行解与可行解夹角的最小值越大, 说明该不可行解处于可行解稀疏的区域, 更易引导种群探索稀疏可行域, 更能准确度量个体的分布性.

基于上述分析, 本文设计一种基于角度信息的分布性度量函数, 用个体 \mathbf{x} 与其他可行个体间夹角的最小值表示, 具体如式(5)所示.

$$\alpha(\mathbf{x}) = \min_{\mathbf{x} \neq \mathbf{y}, \mathbf{y} \in \Omega} \left\{ \arccos \left(\frac{\tilde{\mathbf{F}}(\mathbf{x}) \cdot \tilde{\mathbf{F}}(\mathbf{y})}{\|\tilde{\mathbf{F}}(\mathbf{x})\|_2 \times \|\tilde{\mathbf{F}}(\mathbf{y})\|_2} \right) \right\} \quad (5)$$

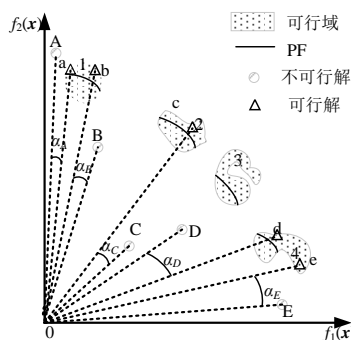


图1 不可行解与可行解间最小夹角的对比示意图

式中, Ω 表示可行域, \mathbf{y} 是可行解且与 \mathbf{x} 不同, \cdot 表示向量的内积, $\tilde{\mathbf{F}}(\mathbf{x})$ 和 $\tilde{\mathbf{F}}(\mathbf{y})$ 分别为 \mathbf{x} 和个体 \mathbf{y} 的最大最小归一化目标向量. $\alpha(\mathbf{x})$ 越大说明个体 \mathbf{x} 的分布性越好. 将约束违反度满足 $G(\mathbf{x}) \leq \varepsilon(t)$ 的个体作为可行解, $G(\mathbf{x}) > \varepsilon(t)$ 的个体作为不可行解, 则本文设计的基于 AVF 的选择操作 (Selection based on AVF, S-AVF) 基于以下 3 条准则.

准则 1: 当 \mathbf{x} 和 \mathbf{y} 均为可行解时, 选择 AVF 值小的个体.

准则 2: 当 \mathbf{x} 为可行解而 \mathbf{y} 为不可行解时, 如果 $\alpha(\mathbf{y}) \leq \theta$, 那么选择个体 \mathbf{x} , 否则随机选择一个个体.

准则 3: 当 \mathbf{x} 和 \mathbf{y} 均为不可行解时, 分以下 3 种情形:

- (1) 当 $\alpha(\mathbf{x}), \alpha(\mathbf{y}) \leq \theta$ 时, 选择约束违反度小的个体;
- (2) 当 $\alpha(\mathbf{x}), \alpha(\mathbf{y}) > \theta$ 时, 选择 AVF 值小的个体;
- (3) 当 $\alpha(\mathbf{x}) > \theta, \alpha(\mathbf{y}) \leq \theta$ 时, 选择个体 \mathbf{x} .

其中, 阈值 θ 用来评价不可行解的分布性优劣, 大于 θ 表示分布性较好. 因为本文设计的个体选择准则目的是从合并种群的 $2N$ 个个体中选出 N 个个体, 为使选出的不可行解的分布性在合并种群中相对较好, 动态设置阈值 θ , 设置方式为将合并种群中 $2N$ 个个体的 α 值降序排列, 然后将排在第 N 位的 α 值作为阈值 θ .

本文提出的 S-AVF 具有以下三个特点: (1) 不再对高维目标向量进行非支配排序, 也不再需要收敛性、分布性保持操作, 大大降低了计算量, 提高了收敛速度; (2) 通过保留分布性好的不可行解, 可以加大对未知区域的探索, 相较于约束非支配排序所规定的可行解占优, 更能够确保种群搜索到均匀分布的 PF; (3) 约束非支配排序在两个个体都不可行时, 仅根据约束违反度来选择, 而 S-AVF 在考虑约束违反度的基础上, 还考虑了不可行解的分布性和 AVF 值, 有利于提高种群的分布性.

3.2 基于 DE 算法的交叉操作

为了进一步弥补选择操作降低的收敛精度, 对交叉操作进行改进. 目前 CMaOEA 中交叉算子均采用模拟二进制方式^[26], 其两个父代个体是随机选择的, 不能保证不可行解一定参与其中, 影响了种群的分布性. DE 算法^[27] 在种群分布性和全局搜索能力方面目前效果最好^[28], 由至少三个父代个体产生一个子代个体, 能够保证至少有一个不可行解作为父代个体. 本文对交叉操作进行了两个部分的改进, 提出了基于 DE 算法的交叉公式, 并在不同进化阶段设计了不同的比例因数, 动态控制不可行解的参与数量; 对于进行交叉操作的不可行解, 设计了动态筛选准则, 进而有效

地提升了算法的收敛精度. 本文提出的交叉操作如式(6)所示.

$$v_i(t) = x_i(t) + F_1 \times (x_{inf}(t) - x_{r1}(t)) + F_2 \times (x_{r3}(t) - x_{r2}(t)) \quad (6)$$

式中, $x_{r1}(t), x_{r2}(t), x_{r3}(t)$ 为与 $x_i(t)$ 不同的可行解, 且 $r1 \sim r3$ 为 $1 \sim N$ 上互不相等的正整数. x_{inf} 是不可行解, F_1, F_2 为比例因数, 用来决定可行解、不可行解在交叉操作中所占的比重.

通过前文分析已知, 进化初期应偏重分布性好的不可行解; 在进化中期不可行解的分布性和收敛性同等重要; 进化后期应偏重收敛性好的不可行解. 为此, 本文提出一种基于 $d(x)$ 和 $\alpha(x)$ 的不可行解筛选准则, 随机选择两个不可行解, 在进化初期, 选择 $\alpha(x)$ 值大的不可行解; 在进化中期, 选择折中值 $p(x) = \cos(\alpha(x)) + d(x)$ 大的不可行解; 在进化后期选择 $d(x)$ 值大的不可行解.

3.3 算法总体流程

通过对选择操作和交叉操作的改进, 本文实现了 CMaOEA-AI, 其总体流程如算法 1 所示.

算法 1 基于角度信息的约束高维多目标进化算法

输入: 最大进化代数 G_{max} , 种群大小 N , 比例因数 F_1, F_2 , 变异概率 CR

输出: 最优解集 P_{t+1}

1. 随机生成初始种群 P_0
2. WHILE $t < G_{max}$
3. B_t = 不可行解筛选(P_t)
4. Q_t = 交叉、变异操作(B_t, P_{t+1})
5. $U_t = P_t \cup Q_t$
6. P_{t+1} = S-AVF 操作(U_t)
7. $t = t + 1$
8. END WHILE

4 实验仿真结果与分析

为了验证 CMaOEA-AI 的有效性和先进性, 选取目前求解效果最好的 4 种 CMaOEAs: C-NSGA-III^[14], C-MOEA/D^[14], C-MOEA/DD^[17]和 C-TAEA^[18]进行对比, 所有实验的仿真程序都是在 Intel Core 4G 内存, 2.6GHz 主频, win10 64 位操作系统的计算机上运行, 采用 MATLAB 2016b 编程仿真.

4.1 测试函数与评价指标

测试函数采用目前公认的标准约束高维多目标测试函数集 C-DTLZ^[14], 其包括 3 种类型: 第一类问题(C1-DTLZ1 和 C1-DTLZ3)主要测试算法收敛到 PF 的性能; 第二类问题(C2-DTLZ2 和 C2-DTLZ2*)主要测试算法跨越孤立可行域, 处理不连续前沿面的能力; 第三类问题(C3-DTLZ1 和 C3-DTLZ4)主要测试算法收敛到

可行域边界的能力. 在本文的测试中, 目标维数设为 $M = \{3, 5, 8, 10, 15\}$. 为综合衡量算法收敛精度, 评价指标采用反向世代距离 (Inverted Generation Distance, IGD)^[29].

4.2 实验参数设置

4 种对比算法的交叉算子均采用模拟二进制交叉, $\eta_c = 30$, 交叉概率为 $p_c = 1.0$; 变异算子采用多项式变异, $\eta_m = 20$, 变异概率为 $p_m = 1/n$, 其中 n 为解空间的维度^[18]. C-MOEA/DD 中邻域大小 $T = 20$, 邻域个体选择概率 $\delta = 0.9$, 惩罚参数 $\theta = 5$ ^[14]; C-TAEA 中 CA 和 DA 的两个归档集的大小设置为种群大小^[18]. 为了保证算法的公平性, 5 种算法的种群大小相同, 且按文献[14]设置, 具体如表 1 所示, 其中 H_1 是双层分布法中的外层分段数, H_2 是内层分段数.

表 1 设置种群大小

M	参考方向规模	种群大小(N)
3	91($H=12$)	92
5	210($H=6$)	212
8	156($H_1=3, H_2=2$)	156
10	275($H_1=3, H_2=2$)	276
15	135($H_1=2, H_2=1$)	136

4.3 实验结果与分析

为验证 CMaOEA-AI 的整体性能, 在 C-DTLZ 函数集上分别进行收敛精度和收敛速度的对比实验. 收敛精度的实验结果如表 2~表 4 所示, 分别为 5 种算法独立运行 30 次所求得的 IGD 平均值与标准差, 其中每个测试问题中最小平均值加粗标出. 另外, 为了判断 CMaOEA-AI 与 4 种对比算法取得 IGD 平均值是否有显著差异, 采用 Wilcoxon 秩和检验^[30]进行两两比较, 显著性水平取 0.05^[18], 用“+”、“-”、“~”分别表示对比算法显著地优于、显著地劣于和无差别于 CMaOEA-AI, 并将 4 种对比算法“+”、“-”、“~”的次数统计在表格的最后一行.

C1-DTLZ1 的可行域仅为靠近 PF 的狭小区域, 如何在庞大的目标空间中尽可能均匀地找到可行解极为重要. CMaOEA-AI 在 3、5、8 目标上的收敛精度排名第 5, 而在 10 目标上排第 4, 在 15 目标上排第 3, 说明 CMaOEA-AI 维持了收敛精度, 且目标维数越高收敛精度的补偿越明显. C1-DTLZ3 的可行域互不连通阻截了种群向 PF 的收敛, 不可行解对进化的作用较大. CMaOEA-AI 的收敛精度在所有目标上都排名第 3, 较可行解占优的 C-NSGA-III 和 C-MOEA/D 提高了两个数量级. 从表 2 可以看出, CMaOEA-AI 的改进都有效地引导种群跨过不可行域, 保持了分布性, 并最终收敛到 PF, 且目标维数越多, 对收敛精度的补偿越明显.

表 2 5种算法在第一类约束问题上实验结果的IGD平均值与标准差

测试问题	M	G_{max}	C-NSGA-III	C-MOEA/D	C-MOEA/DD	C-TAEA	CMaOEA-AI
C1-DTLZ1	3	5000	2.0376e-2 (2.08e-4) +	2.1100e-2 (3.17e-3) ≈	2.0371e-2 (1.50e-4) +	2.0726e-2 (1.23e-3) ≈	2.6617e-2 (1.13e-3)
	5	6000	5.1603e-2 (4.10e-4) +	5.2940e-2 (7.79e-3) ≈	5.1352e-2 (2.90e-4) +	5.9782e-2 (4.98e-4) -	6.2075e-2 (1.24e-3)
	8	8000	1.2470e-1 (1.15e-2) -	1.0935e-1 (6.75e-3) +	1.2066e-1 (2.98e-4) -	1.0007e-1 (2.32e-3) +	1.3925e-2 (2.62e-3)
	10	10000	1.3617e-1 (7.93e-3) -	1.8022e-1 (7.81e-3) -	1.3179e-1 (8.07e-4) ≈	1.3010e-1 (1.85e-4) +	1.5584e-1 (7.77e-4)
	15	15000	1.9966e-1 (9.80e-4) +	2.6049e-1 (7.74e-3) ≈	2.4220e-1 (1.03e-1) -	1.9058e-1 (1.68e-4) +	2.3030e-1 (4.21e-3)
C1-DTLZ3	3	10000	8.0133e+0 (6.31e-3) -	8.007e+0 (1.72e-3) -	8.6610e-2 (3.05e-2) -	5.7687e-2 (8.17e-3) +	9.7757e-2 (3.55e-2)
	5	15000	1.1571e+1 (1.02e-2) -	1.1554e+0 (3.69e-3) -	2.1601e-1 (1.60e-2) -	1.7177e-1 (3.19e-1) -	6.6539e-1 (2.29e-3)
	8	25000	1.1668e+1 (7.83e-2) -	1.1610e+1 (2.64e-3) -	3.1596e-1 (6.90e-2) ≈	2.1762e-1 (1.45e-2) +	8.1711e-1 (1.31e-2)
	10	35000	1.4267e+1 (5.12e-2) -	1.4140e+1 (4.30e-2) -	2.4128e-1 (5.72e-3) -	5.8109e-1 (1.09e-3) +	8.4374e-1 (4.23e-3)
	15	50000	1.4630e+1 (2.26e-1) -	1.4463e+1 (8.22e-2) -	7.4207e-1 (1.24e-3) +	8.6244e-1 (4.52e-2) -	9.4453e-1 (3.09e-2)
+/-			3/0/7	1/3/6	3/2/5	6/1/3	

表 3 5种算法在第二类约束问题上实验结果的IGD平均值与标准差

测试问题	M	G_{max}	C-NSGA-III	C-MOEA/D	C-MOEA/DD	C-TAEA	CMaOEA-AI
C2-DTLZ2	3	2500	4.8195e-2 (1.86e-4) +	4.9069e-2 (3.74e-4) +	4.9725e-2 (5.80e-4) -	4.4169e-2 (1.46e-3) -	4.8765e-2 (5.05e-4)
	5	3500	1.4868e-1 (3.28e-4) ≈	1.4863e-1 (5.93e-4) ≈	1.4800e-1 (3.86e-4) -	1.4569e-2 (1.63e-3) +	1.4835e-1 (6.59e-4)
	8	5000	4.0162e-1 (2.57e-1) -	4.1220e-1 (7.64e-1) -	2.9225e-1 (1.33e-3) -	2.4970e-1 (1.70e-3) +	3.5213e-1 (1.29e-3)
	10	7500	2.7665e-1 (6.10e-2) -	2.5412e-1 (5.57e-3) ≈	2.7047e-1 (2.43e-4) +	1.9095e-1 (9.78e-4) +	2.4783e-1 (1.86e-2)
	15	10000	3.8095e-1 (8.77e-2) -	3.9317e-1 (6.43e-2) -	3.8126e-1 (1.13e-1) -	2.5915e-1 (2.49e-3) +	3.3123e-1 (4.97e-2)
C2-DTLZ2*	3	2500	3.4636e-2 (5.26e-4) +	3.9069e-2 (1.05e-3) +	4.3562e-2 (2.19e-4) +	4.1667e-2 (1.04e-3) +	5.0590e-2 (2.21e-3)
	5	7500	6.5637e-2 (7.01e-4) +	7.1074e-2 (1/84e-3) ≈	8.7447e-2 (1.35e-3) -	6.3779e-2 (7.37e-4) +	7.8708e-2 (4.52e-3)
	8	15000	1.4006e-1 (1.06e-3) +	1.6735e-1 (7.64e-3) +	1.3209e-1 (1.13e-3) ≈	2.1033e-1 (9.76e-3) -	2.8256e-1 (3.32e-3)
	10	25000	1.1358e-1 (2.24e-3) -	1.4919e-1 (1.03e-3) +	1.4979e-1 (4.35e-4) -	3.0437e-1 (2.34e-2) -	3.1828e-1 (4.22e-2)
	15	35000	2.4336e-1 (3.69e-2) -	2.5052e-1 (2.38e-2) -	2.8508e-1 (1.70e-2) -	3.3610e-1 (1.61e-2) -	3.5038e-1 (2.01e-2)
+/-			4/1/5	4/3/3	2/1/7	6/0/4	

C2-DTLZ2的孤立可行域使PF不再连续, C-NSGA-III、C-MOEA/D和C-MOEA/DD所依据的均匀参考方向会严重影响种群的分布性, CMaOEA-AI在3、5、8目标上的收敛精度排名第3, 在10和15目标上排第2, 说明其对约束支配排序的省略在一定程度上影响了种群的收敛能力. C2-DTLZ2*的PF是凸的且不可行域仅为较

细带状区域, C-NSGA-III和C-MOEA/DD的环境选择标准较适合此类问题, CMaOEA-AI的收敛精度在5目标上排名第4, 在其他目标上排第5, 但是随着目标维数的增加, CMaOEA-AI与对比算法收敛精度的差距越来越小, 这主要归功于本文交叉操作具有良好的种群分布性和全局搜索能力, 能有效地保障收敛精度.

表 4 5种算法在第三类约束问题上实验结果的IGD平均值与标准差

测试问题	M	G_{max}	C-NSGA-III	C-MOEA/D	C-MOEA/DD	C-TAEA	CMaOEA-AI
C3-DTLZ1	3	7500	5.0031e-2 (1.27e-2) +	5.9263e-2 (2.86e-2) -	5.6206e-2 (1.00e-2) -	5.5203e-2 (9.83e-3) -	5.0761e-2 (1.21e-3)
	5	12500	1.0723e-1 (3.11e-4) +	1.1073e-1 (5.84e-3) +	1.0693e-1 (2.57e-4) +	1.2539e-1 (4.86e-4) ≈	1.0695e-1 (1.80e-3)
	8	20000	2.5019e-1 (1.12e-3) -	2.4099e-1 (4.97e-3) -	2.4226e-1 (3.48e-4) -	2.4734e-1 (1.14e-3) ≈	2.4053e-1 (3.11e-3)
	10	30000	2.6326e-1 (5.73e-4) -	2.6151e-1 (2.72e-3) ≈	2.6568e-1 (2.60e-5) -	2.7160e-1 (2.83e-2) -	2.5216e-1 (2.77e-3)
	15	40000	4.0143e-1 (1.82e-3) +	4.3989e-1 (8.52e-3) -	4.0194e-1 (1.17e-3) +	5.0268e-1 (8.22e-2) ≈	4.0187e-1 (1.23e-2)
C3-DTLZ4	3	7500	2.4315e-1 (3.35e-1) +	2.4841e-1 (4.21e-1) -	2.7033e-1 (3.23e-1) -	1.6168e-1 (2.53e-3) +	2.1126e-1 (3.30e-1)
	5	12500	3.2141e-1 (1.62e-1) -	3.0484e-1 (4.89e-1) ≈	2.4368e-1 (1.42e-4) +	2.4255e-1 (1.27e-3) +	3.0017e-1 (2.40e-3)
	8	20000	5.9044e-1 (1.33e-1) ≈	5.5268e-1 (7.46e-3) +	4.9654e-1 (2.27e-5) +	5.8925e-1 (3.10e-3) -	5.4185e-1 (6.57e-2)
	10	30000	5.6901e-1 (6.20e-4) +	5.5651e-1 (1.18e-3) ≈	5.6741e-1 (2.91e-5) +	5.9013e-1 (2.02e-3) -	5.4002e-1 (8.39e-2)
	15	40000	1.1437e+0 (1.15e-1) -	7.7589e-1 (4.40e-2) -	7.9192e-1 (1.19e-2) +	7.6794e-1 (2.59e-5) +	7.6903e-1 (3.49e-2)
+/-			5/1/4	2/3/5	6/0/4	3/3/4	

C3-DTLZ1 和 C3-DTLZ4 的 PF 都是由可行边界构成的, 进化过程基本在可行域内进行, 不可行解的分布性信息对收敛到 PF 作用不大, 且 PF 是连续的, 对利用均匀参考方向的 C-NSGA-III、C-MOEA/D 和 C-MOEA/DD 较为有利. 从表 4 可以看出, CMaOEA-AI 在 C3-DTLZ1 的 8、10 目标上收敛精度最好, 在 3、5、15 目标上排名第 2, 在 C3-DTLZ4 的 10 目标上收敛精度最好, 其余目标上次好. 这说明对于第三类约束优化问题, CMaOEA-AI 的选择操作中收敛性度量对于收敛精度的影响不大, 而且更精准的分布性度量和改进的交叉操作使得种群分布更加均匀, 更进一步补偿了收敛精度.

统计表 2~表 4 显著性检测的结果, 在 30 例测试问题中, CMaOEA-AI 显著优于 C-NSGA-III 16 次、C-MOEA/D 14 次、C-MOEA/DD 16 次、C-TAEA 11 次, 平均是 30 个算例的半数, 说明在 C-DTLZ 测试问题上, CMaOEA-AI 与对比算法的收敛精度非常相近, 其收敛精度得到了保持.

为进一步直观展现 CMaOEA-AI 的收敛精度, 图 2 和图 3 给出了 5 种算法在求解最困难的三目标 C2-DTLZ2 和 C3-DTLZ4 上 30 次独立运行中获得最优 IGD 的一次结果.

从图 2 中可以看出, 在 C2-DTLZ2 上, 5 种算法均未找到全部 PF, 而 CMaOEA-AI 所求 Pareto 最优解集分

布的宽广程度和 C-NSGA-III、C-MOEA/DD 相差不多. 在图 3 中, C-NSGA-III 和 C-TAEA 在 C3-DTLZ4 的边界区域上搜索到了所有 PF, CMaOEA-AI 与 C-MOEA/D 和 C-MOEA/DD 均出现了丢失部分边界解的现象, 但相比于 C-MOEA/DD, CMaOEA-AI 在分布性和收敛性上更优, 说明 CMaOEA-AI 在收敛精度上和对比算法有一定程度的下降, 但基本维持在了相同水平上.

为验证 CMaOEA-AI 在收敛速度方面的提升效果, 表 5 分别统计了 CMaOEA-AI 与 4 种对比算法在 C-DTLZ 上独立运行 30 次的平均运行时间, 其中每个测试问题的最短运行时间加粗标出.

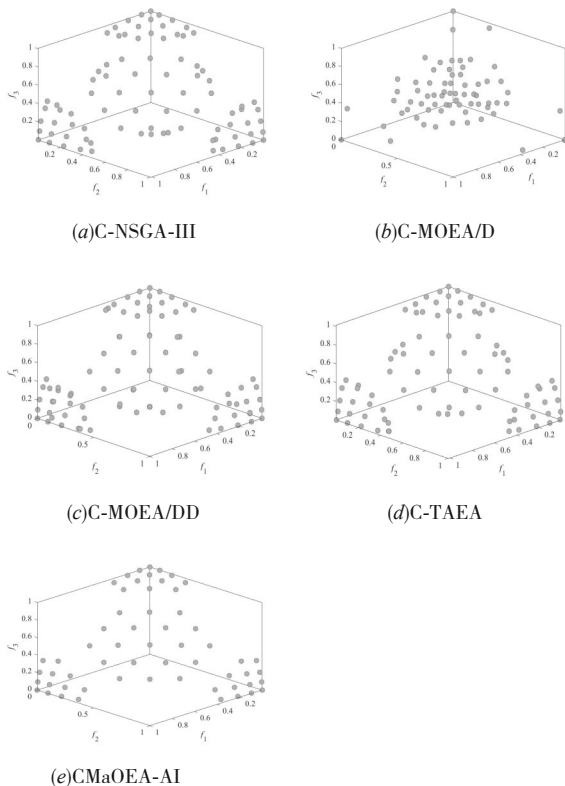


图 2 5 种算法在三目标 C2-DTLZ2 上的最优解

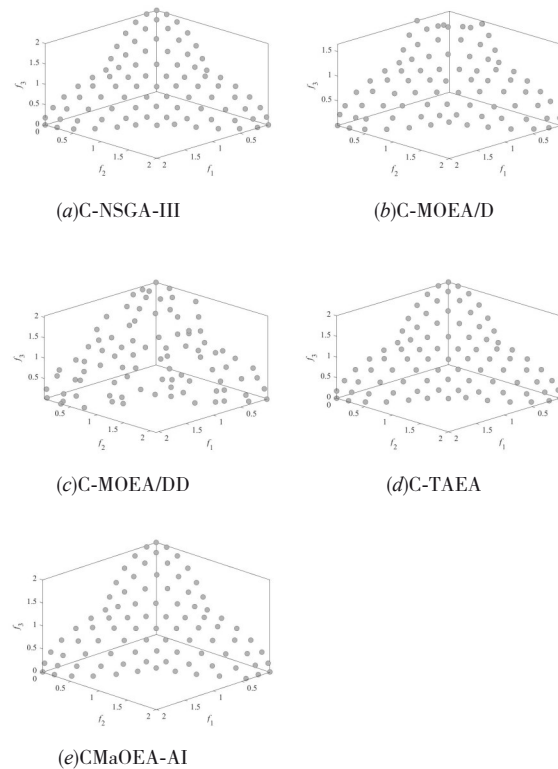


图 3 5 种算法在三目标 C3-DTLZ4 上的最优解

从表 5 可以看出, CMaOEA-AI 在 30 例测试问题的平均运行时间最短. 3 目标时 CMaOEA-AI 的平均运行时间比 4 种对比算法在 6 个测试问题上分别取得的最短运行时间平均缩短了 2%, 但 5 目标以后特别是 8、10、15 目标的问题, CMaOEA-AI 的优势非常明显, 且目标维数越高缩短的运行时间越多. 在 3 目标上 CMaOEA-AI 优势不明显的原因是本文改进的重点是考虑目标维数的增加, 约束非支配排序操作带来的计算量呈指数扩大, CMaOEA-AI 通过省略非支配排序来减少搜索时间, 因此在目标维数较少的情况下改进不明显. 在 5 目标问题上, CMaOEA-AI 比 4 种对比算法在 6

个测试问题上收敛精度分别最高时的运行时间平均短了 10%，比 4 种对比算法分别取得的最短运行时间平均缩短了 3%。在 8 目标问题上，CMaOEA-AI 比 4 种对比算法收敛精度分别最高时的运行时间平均短了 35%，比最短的运行时间平均缩短了 28%。在 8 目标问题上，CMaOEA-AI 比 4 种对比算法收敛精度分别最高时的运行时间平均短了 46%，比最短的运行时间平均缩短了 21%。特别是对于 15 目标约束问题，CMaOEA-AI 比 4 种对比算法收敛精度分别最高时的运行时间平

均短了 61%，比最短的运行时间平均缩短了 17%。综上所述，随着目标维数的增加，CMaOEA-AI 在收敛速度方面的优势越来越明显。

5 结束语

本文提出一种基于角度信息的约束高维多目标进化算法，在确保收敛精度的情况下提高了收敛速度。创新工作主要包括两个方面，首先提出全新的选择操作，给出了基于角度信息的分布性度量定义，通过构造角度违反度函数直接从合并种群中选择个体，省略基于约束支配的非支配排序，提升了收敛速度；其次，提出了基于 DE 算法的交叉操作，在不同进化阶段有针对性地选用部分不可行解参与交叉操作，并给出了不同阶段的比例因数，有效地补偿了收敛精度。通过上述两个方面的改进，算法的收敛精度略有下降但收敛速度得到大幅提升。在标准测试函数集 C-DTLZ 上进行了对比实验，实验结果说明 CMaOEA-AI 相较于目前较好的 4 种算法，收敛精度保持了较好水准，但收敛速度有了较大的提升，并且目标维数增加越多，收敛速率提升越快，更适合在实际约束高维多目标问题中的应用。

参考文献

- [1] ZHOU Y, MIN Z, WANG J, et al. Tri-goal evolution framework for constrained many-objective optimization[J]. IEEE Transactions on Systems, Man & Cybernetics Systems, 2018, 1 – 14.
- [2] LI M Q, ZHOU Y R, XIANG Y, et al. An angle based constrained many-objective evolutionary algorithm[J]. Applied Intelligence, 2017, 47: 705 – 720.
- [3] 顾清华, 莫明慧, 卢才武, 陈露. 求解约束高维多目标问题的分解约束支配 NSGA-II 优化算法[J]. 控制与决策, 2020, 35(10): 2466 – 2474.
GU Q H, MO M H, LU C W, CHEN L. Decomposition-based constrained dominance principle NSGA-II for constrained many-objective optimization problems[J]. Control and Decision, 2020, 35(10): 2466 – 2474. (in Chinese)
- [4] MIYAKAWA M, SATO H, SATO Y. Utilization of infeasible solutions in MOEA/D for solving constrained many-objective optimization problems[A]. Proceedings of the Genetic and Evolutionary Computation Conference Companion [C]. Berlin, Germany: ACM, 2017. 35 – 36.
- [5] MIYAKAWA M, SATO H, SATO Y. Directed mating in decomposition-based MOEA for constrained many-objective optimization[A]. Proceedings of the Genetic and Evolutionary Computation Conference Companion [C]. Kyoto, Ja-

表 5 5 种算法在不同目标维数的 C-DTLZ 测试集上的运行时间平均值(s)

M	G_{\max}	测试问题	C-NS-GA-III	C-MOEA/D	C-MOEA/DD	C-TAEA	CMaOEA-AI
3	5000	C1-DTLZ1	19.00	20.88	18.50	19.20	18.43
	10000	C1-DTLZ3	21.43	20.52	20.34	21.78	20.10
	2500	C2-DTLZ2	19.71	22.87	21.02	20.41	19.22
	2500	C2-DTLZ2*	19.53	21.52	22.50	26.83	19.44
	7500	C3-DTLZ1	21.42	19.21	20.93	23.45	18.79
	7500	C3-DTLZ4	22.01	20.70	20.30	21.54	19.91
5	6000	C1-DTLZ1	67.11	45.17	52.91	43.91	40.04
	15000	C1-DTLZ3	70.32	49.46	55.33	46.31	43.71
	3500	C2-DTLZ2	65.86	47.01	53.03	43.33	41.22
	7500	C2-DTLZ2*	69.39	49.38	53.89	41.56	41.48
	12500	C3-DTLZ1	70.19	50.74	56.60	44.06	43.08
	12500	C3-DTLZ4	71.06	51.19	56.88	43.72	43.58
8	8000	C1-DTLZ1	120.29	80.32	103.63	84.24	54.58
	25000	C1-DTLZ3	132.40	85.18	111.07	89.58	58.15
	5000	C2-DTLZ2	119.33	81.20	104.58	88.27	60.22
	15000	C2-DTLZ2*	114.39	86.01	103.26	85.01	62.12
	20000	C3-DTLZ1	130.43	87.55	113.79	91.40	65.99
	20000	C3-DTLZ4	127.21	86.12	109.51	93.15	65.10
10	10000	C1-DTLZ1	230.13	110.38	198.53	165.09	87.02
	35000	C1-DTLZ3	240.37	115.01	212.21	188.12	92.03
	7500	C2-DTLZ2	225.38	107.23	203.79	171.20	90.15
	25000	C2-DTLZ2*	229.42	119.64	202.50	189.44	91.09
	30000	C3-DTLZ1	239.92	120.26	210.27	183.13	94.01
	30000	C3-DTLZ4	241.46	124.78	208.34	191.74	94.14
15	15000	C1-DTLZ1	563.66	179.86	400.71	273.14	136.81
	50000	C1-DTLZ3	575.32	189.11	412.64	290.93	150.71
	10000	C2-DTLZ2	557.86	183.90	402.61	282.55	151.23
	35000	C2-DTLZ2*	567.34	194.83	404.96	285.86	159.25
	40000	C3-DTLZ1	584.56	191.21	417.55	299.73	166.32
	40000	C3-DTLZ4	578.62	191.20	402.78	283.51	168.99

- pan: ACM, 2018. 721 – 728.
- [6] 李智勇, 黄滔, 陈少森, 李仁发. 约束优化进化算法综述[J]. 软件学报, 2017, 28(6): 1529 – 1546.
- LI Z Y, HUANG T, CHEN S M, LI R F. Overview of constrained optimization evolutionary algorithms[J]. Journal of Software, 2017, 28(6): 1529 – 1546. (in Chinese)
- [7] MATIAS J, CORREIA A, MESTRE P, et al. Adaptive penalty and barrier function based on fuzzy logic[J]. Expert Systems with Applications, 2015, 42(19): 6777 – 6783.
- [8] DEB K, PRATAP A, AGARWAL S, MEYARIVAN T. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182 – 197.
- [9] ZHANG M, LUO W J, WANG X F. Differential evolution with dynamic stochastic selection for constrained optimization[J]. Information Sciences, 2008, 178(15): 3043 – 3074.
- [10] 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)
- [11] ZENG S Y, JIAO R W, LI C H, et al. A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization[J]. IEEE Transactions on Cybernetics, 2017, 47(9): 2678 – 2688.
- [12] ASAFUDDOULA M, RAY T, SARKER R. A decomposition-based evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(3): 445 – 460.
- [13] JAN M A, ZHANG Q F. MOEA/D for constrained multi-objective optimization: Some preliminary experimental result[A]. 2010 UK Workshop on Computational Intelligence [C]. UKCI: IEEE, 2010. 1 – 6.
- [14] JAIN H, DEB K. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(4): 602 – 622.
- [15] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(4): 577 – 601.
- [16] CAI X, MEI Z, FAN Z. A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors[J]. IEEE Transactions on Cybernetics, 2018, 48(8): 2335 – 2348.
- [17] LI K, DEB K, ZHANG Q F, SAM K. An evolutionary many-objective optimization algorithm based on dominance and decomposition[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(5): 694 – 716.
- [18] LI K, CHEN R Z, FU G T, YAO X. Two-archive evolutionary algorithm for constrained multi-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 23(2): 303 – 315.
- [19] FAN Z, WANG Z J, LI W J, et al. Push and pull search embedded in an M2M framework for solving constrained multi-objective optimization problems[J]. Swarm and Evolutionary Computation, 2020, 54: 100651.
- [20] MARTINEZ S Z, COELL C C. A multi-objective evolutionary algorithm based on decomposition for constrained multi-objective optimization[A]. Proceedings of 2014 IEEE Congress on Evolutionary Computation[C]. Beijing, China: IEEE, 2014. 429 – 436.
- [21] 王雨萍, 章鸣雷, 邱飞岳, 江波. 基于角度惩罚距离精英选择策略的偏好高维目标优化算法[J]. 计算机学报, 2018, 41(1): 236 – 253.
- WANG L P, ZHANG M L, QIU F Y, JIANG B. Many-objective optimization algorithm with preference based on the angle penalty distance elite selection strategy[J]. Chinese Journal of Computers, 2018, 41(1): 236 – 253. (in Chinese)
- [22] 刘元, 郑金华, 邹娟, 喻果. 基于邻域竞赛的多目标优化算法[J]. 自动化学报, 2018, 44(7): 1304 – 1320.
- LIU Y, ZHENG J H, ZOU J, YU G. An evolutionary algorithm through neighborhood competition for multi-objective optimization[J]. Acta Automatica Sinica, 2018, 44(7): 1304 – 1320. (in Chinese)
- [23] 陈振兴, 严宣辉, 吴坤安, 白猛. 融合张角拥挤控制策略的高维多目标优化[J]. 自动化学报, 2015, 41(6): 1145 – 1158.
- CHEN Z X, YAN X H, WU K A, BAI M. Many-objective optimization integrating open angle based congestion control strategy[J]. Acta Automatica Sinica, 2015, 41(6): 1145 – 1158. (in Chinese)
- [24] LIU Z Z, WANG Y, HUANG P Q. AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation[J]. Information Sciences, 2018, doi: 10.1016/j.ins.2018.06.063.
- [25] 毕晓君, 王朝. 一种基于角度惩罚距离的高维多目标进化算法[J]. 电子与信息学报, 2018, 40(02): 314 – 322.

- BI X J, WANG C. A many-objective evolutionary algorithm based on angle penalized distance[J]. Journal of Electronics & Information Technology, 2018, 40(02): 314 – 322. (in Chinese)
- [26] DEB K, AGRAWAL R B. Simulated binary crossover for continuous search space[J]. Complex Systems, 1994, 9(3): 115 – 148.
- [27] 王亚辉, 吴金妹, 贾晨辉. 基于动态种群多策略差分进化模型的多目标进化算法[J]. 电子学报, 2016, 44(6): 1472 – 1480.
WANG Y H, WU J M, JIA C H. Multi-objective evolutionary algorithm based on dynamic population multi-strategy differential models[J]. Acta Electronica Sinica, 2016, 44 (6): 1472 – 1480. (in Chinese)
- [28] 王柳静, 张贵军, 周晓根. 基于状态估计反馈的策略自适应差分进化算法[J]. 自动化学报, 2020, 46(04): 752 – 766.
WANG L J, ZHANG G J, ZHOU X G. Strategy self-adaptive differential evolution algorithm based on state estimation feedback[J]. Acta Automatica Sinica, 2020, 46(04): 752 – 766.(in Chinese)
- [29] ZITZLER E, THIELE L, LAUMANN M, et al. Performance assessment of multi-objective optimizers: An analy-

sis and review[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(2): 117 – 132.

- [30] DEB K, THIELE L, LAUMANN M, ZITZLER Z. Scalable test problems for evolutionary multi-objective optimization [A]. Proceedings of Evolutionary Multi-objective Optimization: Theoretical Advances and Applications[C]. London, UK: IEEE, 2005. 105 – 145.

作者简介



刘冰洁 女, 1989年3月生, 河北秦皇岛人. 哈尔滨工程大学博士研究生. 主要从事进化计算、智能信息处理技术方面的研究.
E-mail:993910899@qq.com



毕晓君(通讯作者) 女, 1964年11月生, 黑龙江哈尔滨人. 中央民族大学信息工程学院教授、博士生导师. 主要从事智能信息处理技术、数字图像处理、智能优化算法及机器学习的理论与应用方面的研究.
E-mail:bixiaojun@hrbeu.edu.cn