

一种基于均匀分布策略的 NSGAI 算法

乔俊飞^{1,2} 李 霏^{1,2} 杨翠丽^{1,2}

摘 要 针对局部搜索类改进型非劣分类遗传算法 (Nondominated sorting genetic algorithm II, NSGAI) 计算过程中种群分布不均的问题, 提出一种基于均匀分布的 NSGAI (NSGAI based on uniform distribution, NSGAI-UID) 多目标优化算法. 首先, 该算法将种群映射到目标函数对应的超平面, 并在该平面上进行聚类以增加解的多样性. 其次, 为了提高解的分布性, 将映射平面进行均匀分区. 当分段区间不满足分布性条件时, 需要激活分布性加强模块. 与此同时在计算过程中分段区间可能会出现种群数量不足或无解的状况, 为了保证每个区间所选个体数目相同. 最后, 采用将最优个体进行极限优化变异的方法来获得缺失个体. 实验结果显示该算法可以保证种群跳出局部最优且提高收敛速度, 并且在解的分布性和收敛性方面均优于文中其他多目标优化算法.

关键词 改进型非劣分类遗传算法, 映射, 聚类, 分布性加强, 局部变异

引用格式 乔俊飞, 李霏, 杨翠丽. 一种基于均匀分布策略的 NSGAI 算法. 自动化学报, 2019, 45(7): 1325–1334

DOI 10.16383/j.aas.c180085

An NSGAI Algorithm Based on Uniform Distribution Strategy

QIAO Jun-Fei^{1,2} LI Fei^{1,2} YANG Cui-Li^{1,2}

Abstract Because the population distribution is uneven during the local search process of nondominated sorting genetic algorithm II (NSGAI), a multi-objective optimization algorithm for NSGAI based on uniform distribution (NSGAI-UID) is proposed. Firstly, the population which has been clustered is mapped to the hyperplane of the corresponding objective function, then the diversity of population is increased. Secondly, in order to improve the distribution uniformity of the solution, the mapping plane is evenly partitioned. However, when the distribution condition is not satisfied in the corresponding partition, the distribution enhancement module is activated. At the same time the individuals may be insufficient or empty in the piecewise interval during the calculation process, in order to ensure that the number of selected individuals in each interval is the same, the local variation method of the best solution is proposed to get the missing individuals lastly. The experimental results show that the method ensures that the population can jump out the local optimal and the convergence speed can be improved. And the distribution and convergence of this algorithm is superior to the other multi-objective optimization algorithms.

Key words Nondominated sorting genetic algorithm II (NSGAI), map, cluster, distribution enhancement, local variation

Citation Qiao Jun-Fei, Li Fei, Yang Cui-Li. An NSGAI algorithm based on uniform distribution strategy. *Acta Automatica Sinica*, 2019, 45(7): 1325–1334

大多数的工程和科学应用, 如工业制造、城市

运输、污水处理和资本运算等, 几乎每个重要的现实生活中的决策都存在多目标优化问题. 这些目标往往是不可比较, 甚至是相互冲突的. 因此多目标优化问题 (Multiobjective optimization problem, MOP) 一直是近几年主要的研究课题之一. 为了解决该问题, 多目标进化算法 (Multiobjective evolutionary algorithm, MOEAs) 已被广泛研究. 其中 Schaffer^[1] 提出了向量评估遗传算法 (Vector evaluated genetic algorithm, VE-GA), 该算法对单目标遗传算法进行了改进. 但 VEGA 只能找到 Pareto 前端的始末端, 产生不了均匀分布的解. 为此 Fonseca 等^[2] 提出了多目标遗传算法 (Multiobjective genetic algorithm, MOGA), 这种算法效率较高, 而且易于实现. 但该算法的不足在于如果小生境数目信息是基于目标函数的, 那么两个具有相同目

收稿日期 2018-02-05 录用日期 2018-05-07

Manuscript received February 5, 2018; accepted May 7, 2018

国家自然科学基金 (61533002, 61603012), 北京市教委项目 (KM201710005025), 北京市博士后工作经费资助项目 (2017ZZ-028), 中国博士后科学基金, 北京市朝阳区博士后工作经费资助项目 (2017ZZ-01-07)

Supported by National Natural Science Foundation of China (61533002, 61603012), Beijing Municipal Education Commission Foundation (KM201710005025), Beijing Postdoctoral Research Foundation (2017ZZ-028), China Postdoctoral Science Foundation Funded Project, and Beijing Chaoyang District Postdoctoral Research Foundation (2017ZZ-01-07)

本文责任编辑 付俊

Recommended by Associate Editor FU Jun

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

1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124 2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124

标函数向量的不同个体无法在同一代种群中存在. 为了解决该问题, Srinivas 等^[3] 提出了非劣分类遗传算法 (Nondominated sorting genetic algorithm, NSGA), 该算法先按解个体的非劣关系进行排序, 再按照共享机制来保持进化的多样性. 但是该算法计算效率较低, 计算复杂度大, 且共享参数 σ 需要预先确定. 为了减小算法的计算复杂度, Deb 等^[4] 提出了改进型非劣分类遗传算法 (Nondominated sorting genetic algorithm II, NSGAI), 该算法采用非支配排序进行分级, 通过计算拥挤距离选择最优解, 并将其做为精英解保存起来. 但该算法是一种类随机搜索算法, 存在操作次数多收敛速度慢和解分布特性较差的问题^[5].

由于种群在迭代过程中种群分布不均会导致种群多样性变差从而易使解陷入局部最优, 并且在迭代过程中 Pareto 前沿部分区域易出现空白, 收敛速度变慢. 为了解决种群分布不均带来的上述的问题, Goldberg 等^[6] 提出了基于共享机制的小生境技术, 该方法考虑了一个个体与群体中其他个体之间的相似程度, 但计算开销比较大. 朱学军等^[7] 提出用进化群体熵来刻画进化群体的多样性和分布性. 然而这种方法缺乏对群体内部个体之间关系的刻画, 因此不便于调控群体进化过程中的多样性和分布性. 为了解决这个问题, Corne 等^[8] 提出了网格方法, 将网格中聚集密度大的个体删除, 但不能删除极点个体. Knowles 等^[9] 提出了自适应网格技术, 在每一代进化时, 根据当前代的个体分布情况自适应地调整边界. Morse^[10] 提出了聚类分析的方法来保持种群的多样性. Han 等^[11] 提出了基于种群间距离信息和种群分布熵的方法. 但是以上方法只能删除种群中拥挤距离较小的个体, 无法解决解的过程中 Pareto 前沿部分区域出现空白的问题.

基于以上问题, 本文提出了一种基于均匀分布的 NSGAI (NSGAI-UID) 多目标优化算法. 该算法受文献 [12] 的启发, 将种群映射到目标值对应的超平面上, 并在该平面上聚类. 但是文献 [12] 中提出的算法仍然存在种群分布不均的问题, 从而影响了种群的多样性, 为了解决该问题, 本文将映射平面均匀分区. 当对应区间的分布性不满足时, 分布性加强模块激活. 由于种群在迭代的过程中对应区间会出现种群个体不足或缺失的状况, 此时需要在该区间内放入一些个体. 为了解决该问题, 本文将所选聚类子群体中拥挤距离最大的点进行局部搜索, 采用极限优化变异^[13] 的方法产生新的个体. 实验结果表明, 该方法综合评价指标 (Inverted generational distance, IGD) 值和分布性评价指标 (Spacing, SP) 值均高于其他算法. 因此表明该方法具有较好的种群多样性和分布性, 且收敛速度较快.

1 基本概念

1.1 多目标及相关定义

定义 1. 多目标优化问题 (MOP)

一个具有 n 个决策变量, m 个目标函数的多目标优化问题可以描述为:

$$\min \mathbf{F}(\mathbf{X}) = (f_1(\mathbf{X}), \dots, f_j(\mathbf{X}), \dots, f_m(\mathbf{X})) \quad (1a)$$

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbf{X} \subset \mathbf{R}^n \quad (1b)$$

$$l_i \leq x_i \leq u_i, \quad i = 1, 2, \dots, n \quad (1c)$$

式中, $\mathbf{x}$ 称为决策变量, $\mathbf{X}$ 是 n 维的决策空间; $\mathbf{F}(\mathbf{X}) \in \mathbf{R}^m$ 为 m 维目标向量; $f_j(\mathbf{X})$ ($j = 1, 2, \dots, m$) 为第 j 个目标函数, l_i 和 u_i 分别为第 i 个决策变量的上界和下界.

定义 2. Pareto-占优对于给定的两点 $x, x^* \in X_f$, x^* 是 Pareto-占优 (非支配) 的, 当且仅当式 (2) 成立, 记为 $x^* > x$.

$$(\forall i \in \{1, 2, \dots, m\} : f_i(x^*) \leq f_i(x)) \quad (2a)$$

$$\wedge (\exists k \in \{1, 2, \dots, m\} : f_k(x^*) \leq f_k(x)) \quad (2b)$$

定义 3. Pareto-最优解

若对于任意解 x , 不存在 $x' \in \Omega$ 使得 $\mathbf{F}(x') = (f_1(x'), f_2(x'), \dots, f_m(x'))$ 占优于 $\mathbf{F}(x) = (f_1(x), f_2(x), \dots, f_m(x))$, 则称 x 为 Pareto 最优解或者非劣解.

定义 4. Pareto-最优解集

所有 Pareto-最优解组成的集合 P_s 称为 Pareto-最优解集.

定义 5. Pareto-前沿

Pareto-最优解集合 P_s 中的解对应的目标函数值组成的集合 P_F 称为 Pareto-前沿, 即:

$$P_F = \{\mathbf{F}(x) = (f_1(x), f_2(x), \dots, f_m(x)) | x \in P_s\} \quad (3)$$

1.2 模糊 C -均值聚类算法

设 n 个数据样本为 $P = \{p_1, p_2, \dots, p_n\}$, c ($2 \leq c \leq n$) 是要将数据样本分成不同类型的数目, $\{A_1, A_2, \dots, A_c\}$ 表示相应的 C 个类别, U 是其相似分类矩阵, 各类别的聚类中心为 $\{v_1, v_2, \dots, v_n\}$, $\mu_k(p_i)$ 是样本 p_i 对于类 A_k 的隶属度 (简写为 μ_k). 则目标函数 J_b 可以表达为:

$$J_b(P, v) = \sum_{i=1}^n \sum_{k=1}^c (\mu_{ik})^b (d_{ik})^2 \quad (4)$$

$$d_{ik} = d(p_i - v_k) = \sqrt{\sum_{j=1}^m (p_{ij} - v_{kj})^2} \quad (5)$$

其中, d_{ik} 是欧几里得距离, 用来度量第 i 个样本 p_i 与第 k 类中心点之间的距离; m 是样本的特征数; b 是加权参数, 取值范围是 $1 \leq b \leq \infty$. 模糊 C -均值聚类方法就是寻找一种最佳的分类, 以使该分类能产生最小的函数值 J_b . 它要求一个样本对于各个聚类的隶属度值和为 1, 即满足:

$$\sum_{j=1}^c \mu_j(p_i) = 1, \quad i = 1, 2, \dots, n \quad (6a)$$

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{b-1}}} \quad (6b)$$

$$v_{ij} = \frac{\sum_{k=1}^n (\mu_{ik})^n p_{kj}}{\sum_{k=1}^n (\mu_{ik})^b} \quad (6c)$$

设 $I_k = \{i | 2 \leq c < n; d_{ik} = 0\}$, 对于所有的 i 类, $i \in I_k$, $\mu_{ik} = 0$. 当算法收敛时, 理论上就得到了各类的聚类中心以及各个样本对于各模式类的隶属度, 从而完成了模糊聚类划分.

2 基于均匀分布的 NSGAII 算法

为了解决种群分布不均的问题, 本文提出了将目标空间均匀分区, 选取相同数量个体的方法. 由于目标空间一般为曲线或者曲面, 想要均匀划分较难操作. 为了解决这个问题, 本文受文献 [12] 的启发, 在文献 [12] 中, 种群中的个体映射在一个超平面上, 并在目标空间中聚类. 为了验证种群的多样性将集群质量度量标准. 但是该方法在迭代过程中存在种群分布不均的问题, 从而影响种群的多样性, 进而造成收敛速度慢, Pareto 前沿有些区域出现空白的现象. 针对以上问题本文提出了将种群映射到目标函数值对应的超平面 H , 在 H 上进行均匀分区以增加种群的多样性的方法. 该方法将种群在映射平面上进行聚类, 并记录聚类中心, 计算每个分区中的聚类中心个数, 从而判断是否满足均匀分布条件. 当条件不满足时, 分布性加强模块激活. 由于在计算过程中区间内有时会出现个体数量不足或者为空的情况, 为了在该区间获取缺失的个体, 本文采用极限优化变异产生新的个体. 当该区间个体不为空时, 将聚类子群体按照拥挤距离从大到小排序, 选取第一个个体进行变异. 当该区域为空时, 将离该区间最近的个体进行变异. 该方法不但可以使种群跳出局部最优, 并且能够较快地找到 Pareto 前端.

2.1 映射

设目标向量为:

$$\mathbf{A}^i = (0, \dots, 0, a_i^{\max}, 0, \dots, 0)^T \in \mathbf{R}^k \quad (7)$$

$a_i^{\max}$ 为种群 P 中目标向量 a_i 的最大函数值.

$$a_i^{\max} \in \mathbf{R} \setminus 0, \quad i = 1, 2, \dots, k \quad (8)$$

定义超平面^[12] H 为 k 维欧氏空间的 $k-1$ 维线性子空间, $b^s = -1$, $\langle \cdot \rangle$ 为欧氏内积

$$H = \{a \in \mathbf{R}^k : \langle \mathbf{W}^s, f \rangle + b^s = 0\} \quad (9)$$

目标向量 f 映射到超平面^[12] H 为 PI , 公式如下所示:

$$\mathbf{W}^s = \left(\frac{1}{f_1^{\max}}, \dots, \frac{1}{f_i^{\max}}, \dots, \frac{1}{f_k^{\max}} \right) \quad (10)$$

$$PI = \frac{1 - \langle \mathbf{W}^s, f \rangle}{\|\mathbf{W}^s\|^2} \mathbf{W}^s + f \quad (11)$$

映射到超平面的子群体为 RP_i^t , $i = 1, 2, \dots, K$, 其中 K 为子群体的个数, 在公式中, 如果 $f_i^{\max} = 0$, 则设置 $f_i^{\max} = 10^{-6}$.

$$Q = \sum_{i=1}^K \frac{1}{|RP_i^t|} \sum_{C_j \in RP_i^t} (D(C_j, \sigma_i)) \quad (12)$$

Q : 集群质量指数, $|RP_i^t|$ 为聚类 i 的个体数量, σ_i 为聚类中心, C_j 为聚类的一个点.

2.2 分布性

由于 NSGAII 在运算过程中种群的分布性较差, 为了解决这个问题, 本文加入了分布性判断模块. 该模块用来判断每个均匀分段区间内聚类子群体的分布情况, 同时选取聚类子群体中种类数量最大值作为阈值来判断分布性. 当区间内种群类别小于该阈值时, 则判断该区间不满足分布性, 分布性加强模块激活. 否则, 该区域满足分布性, 且选取每个聚类子群体中拥挤距离最大的个体放入精英档案.

该算法首先将目标区间均匀分区, 其中每个区间的种群分布如下:

$$D_o = \{D_{o1}, \dots, D_{oi}, \dots, D_{on}\} \quad (13)$$

n 为分段区间的数量, D_{oi} 为第 i 个区间所包含的子群体所有类别的集合, $i \in [1, n]$, 如下所示:

$$D_{on} = [NRP_{n1}, \dots, NRP_{ij}, \dots, NRP_{nr}] \quad (14)$$

$$NRP_{ij} = (AB_{nr1}, \dots, AB_{nrk}, \dots, AB_{nrm}) \quad (15)$$

NRP_{ij} 为第 i 个区间内的种群中心编号, 其中 $i \in [1, n]$, $j \in [1, r]$, r 为种群中心数量, AB_{nrk} 为第 i

个区间第 r 个聚类中心所对应的个体, $k \in [1, m]$, m 为该子群体个体数量. 区间内种群中心数量最大值为 $\max nr$. 判断种群分布性的方法如下所示:

$$\begin{aligned} ir &< \max nr, & \text{情况 1} \\ ir &= \max D_{on}, & \text{情况 2} \end{aligned} \quad (16)$$

其中, ir 为第 i 个区间内的种群中心数量, $\max D_{on}$ 为分段区间内种群中心数量最大值.

情况 1. 若该条件成立, 则分布性加强模块激活.

情况 2. 将该类子群体按照拥挤距离从大到小排序, 选择每个聚类子群体第一个个体.

2.3 分布性加强

当种群不满足上述分布性条件时, 分布性加强模块被激活. 在种群选取的过程中, 每个区间选择相同数量的个体. 同时为了增加种群的多样性, 从每个聚类子群体中按照方法 1) 与 2) 选取拥挤距离较大的个体放入精英档案. 然而在运算的过程中, 区间内有时会出现种群数量不足或空缺的状况. 在以往文献中多是采用将拥挤距离较小的个体删除的方法, 当种群空缺或不足时, 并未有相关文献进行介绍. 为了解决这个问题, 本文提出了对所选精英个体进行极限优化变异的方法, 来填补欠缺的个体. 由于在计算过程中区间内种群中心的数量不同, 采取的策略也不相同. 具体方法如下所示:

1) 第 i 个区间种群中心个数量 ir 为 0:

$$\begin{cases} icm \geq \max nr, & \text{情况 1} \\ icm < \max nr, & \text{情况 2} \\ icm = 0, & \text{情况 3} \end{cases} \quad (17)$$

其中 icm 为在第 i 个区间内第 c 个聚类中心对应的子群体中的个体数量, $c \in [1, ir] \cup 0$.

情况 1. 将该类子群体按照拥挤距离从大到小排序, 取前 $\max nr$ 个个体.

情况 2. 选取所有个体, 剩余 $\max nr - icm$ 个个体由拥挤距离最大的个体变异得到. (详见第 2.4 节)

情况 3. 由于该区间无个体, 本文选取离该区间最近的两个个体变异得到 $\max nr$ 个个体 (详见第 2.4 节). 若该区间连续 20 次为空则该区间设定为空.

2) 第 i 个区间种群中心个数量 ir 不为 0:

当该区间种群中心个数不为零时, 需要从聚类中心对应的子群体中选择相应的个体. 选择方式如下所示:

$$S_num = \text{floor}(\max nr / ir) \quad (18)$$

$$|Re_{nr}| = \max nr - S_num \times ir \quad (19)$$

$$Re_{nr} = (Re_{nr1}, \dots, Re_{nrir}, \dots, Re_{nrir}) \quad (20)$$

S_num 为每个聚类子群体应当平均选择的个体数目, Re_{nr} 为需要选择的剩余个体的集合, $|Re_{nr}|$ 为需要选择的剩余个体的集合的数量, Re_{nrir} 为剩余的个体对应编号, $i \in [1, t]$, t 为余数的个数. 根据余数的是否为 0, 选择方式也不相同, 具体如下:

a) 余数 Re_r 为 0:

$$\begin{cases} icm \geq S_num, & \text{情况 1} \\ icm < S_num, & \text{情况 2} \end{cases} \quad (21)$$

情况 1. 将该类子群体按照拥挤距离排序, 取前 S_num 个个体.

情况 2. 选取所有个体, 剩 $nrm - S_num$ 个个体由变异得到. (详见第 2.4 节)

b) 余数 Re_r 不为 0:

当剩余个体数量不为 0 时, 选择的方式如下所示:

$$\begin{cases} icm \geq S_num + Re_r, & \text{情况 1} \\ icm < S_num + Re_r, & \text{情况 2} \end{cases} \quad (22)$$

情况 1. 将该类子群体按照拥挤距离排序, 取前 S_num 个个体, 剩余 Re_{nr} 个个体随机从第 $rand(1, ir)$ 个子群体选取. 若 NRP_{ij} 对应的子群体被选中 k 次, 则从第 $S_num + k + 1$ 个个体开始选取.

情况 2. 当 $icm < S_num$ 时, 选取全部个体, 剩余 $S_num - icm$ 个个体由变异得到 (详见第 2.4 节). 当 $S_num < icm < S_num + Re_{nr}$ 时, NRP_{ij} 对应的子群体选择前 S_num 个个体, 若该群体被选中 k 次, 且 $icm \geq S_num + k$, 则从第 $S_num + k + 1$ 个开始选取. 若 $icm < S_num + k$, 则重新随机选择.

2.4 局部变异策略

由于在迭代过程中有的时候区间会出现种群数量不足或为空的情况. 为了保证每个区间个体选取数量相同. 本文采取最优个体极限优化变异策略. 首先找到需要选择子群体中拥挤距离最大的个体或者离该区间最近的两个个体, 然后对其进行变异, 本文采用两种变异策略来产生局部解^[13]:

1) 第一种变异策略:

对于选定的对象, 每次只有一个决策变

量发生变异, 这样有效的提高了种群的局部搜索能力, 从而提高了解的计算精度. 设当前选定个体 $AB_{nrm} = (AB_{nrm1}, \dots, AB_{nrmir}, \dots, AB_{nrmj})$, j 为决策变量的个数. 种群总数为 N , 则产生局部解的个数为 μ , μ 的值由上文 (详见第 2.3 节) 的需要动态决定. 具体如下:

$$AB_{nrm} = (AB_{nrm1}, \dots, AB'_{nrmir}, \dots, AB_{nrmj})$$

$$0 < i \leq j \quad (23)$$

$$AB'_{nrmi} = AB_{nrmi} + \delta \cdot \theta_{\max}(AB_{nrmi}), \quad 0 < i \leq j \quad (24)$$

$$\delta = \begin{cases} (2\alpha)^{\frac{1}{1+\beta}} - 1 \\ 1 - [2(1 - \alpha)]^{\frac{1}{1+\beta}} \end{cases} \quad (25)$$

$$\theta_{\max}(AB_{nrmi}) = \max[AB_{nrmi} - l_i, u_i - AB_{nrmi}], \quad 0 < i \leq j \quad (26)$$

其中, AB_{nrmi} 为待变异个体的决策变量; α 为 $(0, 1)$ 区间的随机变量; $\beta \in \mathbf{R}$, 且为形状参数, 本文通过多次实验得到 β 设为 11 为最佳. $\theta_{\max}(AB_{nrmi})$ 为决策变量 AB_{nrmi} 可变动区间的最大值. 该变异方法因为每次只将一个决策变量变异, 所以具有较强的局部微调能力. 但是该方法只能够在小范围内搜索. 为了避免种群陷入局部最优, 同时提高搜索速度, 本文加入了第二种变异策略. 该策略将产生 $20\% \times \mu$ 个局部解^[14] (如非整数则取整), 具体方法如下所示:

2) 第二种变异策略:

$$AB^*_{nrm}(k) = (AB'_{nrm1}, \dots, AB'_{nrmj}, \dots, AB'_{nrmj}) \quad (27)$$

$$0 < i \leq j \quad (28a)$$

$$k = 1, 2, \dots, [0.2\mu] \quad (28b)$$

$$AB'_{nrm} = \lambda AB_{nrm} \quad (28c)$$

$$0 < \lambda < 1.2 \quad (28d)$$

其中, $\lambda \in (0, 1.2)$ 为一随机数. 以上两种变异策略共产生 $\lambda + [0.2\lambda]$ 个局部解.

2.5 算法流程

根据具体问题初始化参数: 设定目标函数有 m 个, 初始种群数量为 N , 精英解数量为 $N/2$, 函数最大调用次数设为 $I_{\max}$, 决策变量维数为 j , 决策变量的取值上下界为 $\mathbf{u} = (u_1, u_2, \dots, u_i, \dots, u_j)$, 下界为 $\mathbf{l} = (l_1, l_2, \dots, l_i, \dots, l_j)$, 形状参数设为 g , 交叉参数 θ_c , 交叉概率 P_c , 变异参数 θ_m , 变异概率 P_m , 具体的算法流程如下所示:

步骤 1. 初始化种群 $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m, \dots, \mathbf{p}_N\}$, 其中 $\mathbf{p}_m = (x_1, x_2, \dots, x_j)$, $m = 1, 2, \dots, N$, 其中 $x_j \in (l_m, u_m)$. 计算种群中每个个体对应的目标函数值.

步骤 2. 对种群 P 中的解进行非支配排序, 排序后当前种群的所有非支配解记为 P_c .

步骤 3. 将种群 P_c 映射到目标函数值对应的超平面 H , 在 H 上进行均匀分区. 种群 P_c 映射后的矩阵为 P'_c .

步骤 4. 计算种群 P'_c 的拥挤距离, 并按照非支配解值和拥挤距离进行排序. 排序后种群记为 RP .

步骤 5. 将 H 平面上的种群 RP 进行聚类, 聚类子群体为 $RP'_i, i = 1, 2, \dots, K$, 其中 K 为子群体的个数. 记录聚类中心及聚类子群体编号. 判断种群 RP'_i 在均匀区间的分布性, 详见第 2.2 节. 如果不满足分布性, 则激活分布性加强模块.

步骤 6. 当区间内的子群体不满足均匀分布条件时, 需要按照第 2.3 节所示增加种群的分布性, 若该区间个体数量不足时, 缺少的数量通过对拥挤距离最大的个体变异得到, 详见第 2.4 节. 当前选择出来的用来增加分布性的种群记为 P_e , 种群数量为 $n \times \max nr$.

步骤 7. 将种群 P_c 中包含在 P_e 中的个体去掉, 然后按照拥挤距离选取 $N/2 - n \times \max nr$ 个个体, 记为 P_s .

步骤 8. 合并 P_e 与 P_s , 得到种群 P_m . 将 P_m 进行交叉变异, 得到新的种群 P_h . 合并 P_m 和 P_h 为下一代种群 P_T .

步骤 9. 重复步骤 2 至步骤 8, 当达到最大迭代次数或者预设的目标时, 进行下一步.

步骤 10. 当前 PT 中的非支配解即为得到的最优解.

3 仿真实验

本文采用 Matlab R2013b 版本, 处理器为 3.60 GHz, 8.00 GB 内存, Microsoft 实验环境. 为了验证算法的收敛性和多样性采用了以下性能评价指标对算法进行了验证.

1) 综合评价指标 (Inverted generational distance, IGD):

IGD 用来评价算法的性能, 它的值越小则解的收敛性和多样性就越好. 具体计算公式如下所示:

$$IGD(P^*_{\text{new}}, P_{\text{new}}) = \frac{\sum_{x \in P^*_{\text{new}}} d(x, P_{\text{new}})}{|P^*_{\text{new}}|} \quad (29)$$

其中, P_{new} 为 Pareto 前端, P^*_{new} 为算法计算出来的 Pareto 解集, $d(x, P_{\text{new}})$ 为 Pareto 解与 Pareto 前沿的欧氏距离最小值. $|P^*_{\text{new}}|$ 为 Pareto 解集中解的个数.

2) 分布性评价指标 (Spacing, SP):

SP 用来测量已知帕累托前沿相邻解间距离的方差. 定义如下所示:

$$SP = \sqrt{\frac{1}{q-1} \sum_{i=1}^q (\bar{u} - u_i)^2} \quad (30)$$

$$u_i = \min_{j,l} \left\{ \sum_{k=1}^m |f_k(x_j) - f_k(x_l)| \right\} \quad (31)$$

q 为非支配解的个数, $q = 2, 3, \cdots, n$, $\bar{u}$ 为 u_i 的平均值.

3.1 实验 1

由于双目标 ZDT 系列函数和三目标 DTLZ 系列函数被广泛地用于验证多目标优化算法^[13], 本实验分别采用 (ZDT1 ~ ZDT4, ZDT6) 函数和 (DTLZ1、DTLZ2) 函数来验证算法的性能. 该系列函数的特征、维度和种群规模如表 1 所示.

表 1 测试函数参数
Table 1 Paramter setting of the test function

函数	pareto 前沿特征	维度		种群规模
		变量	目标	
ZDT1	凸	30	2	1 000
ZDT2	凹	30	2	1 000
ZDT3	凸且非连续	30	2	536
ZDT4	凸	30	2	1 000
ZDT6	凹	10	2	420
DTLZ2	凹	10	3	5 000
DTLZ7	多峰且非连续	20	3	4 700

为了检验 NSGAII-UID 算法的性能, 该算法分别和一种基于密度的局部搜索 NSGAII 算法 (NSGAII-DLS^[14])、标准 NSGAII^[15]、定向搜索算法 NSGAII-els^[16]、随机局部搜索算法 HMOEA/D^[17]、自适应多目标粒子群算法 (AMOPSO^[11])、多目标差分进化算法 (MODE^[16]) 6 种算法进行了对比.

本实验采用模拟二进制交叉和多项式变异方法, 交叉参数 η_c 和变异参数 η_m 均设为 20, 交叉和变异概率分别为 0.9 和 $1/n$, n 为决策变量的数量; 形状参数 q 设为 11. 在进行 ZDT 实验时, 不同的多目标优化算法采用相同的种群数量: 最大计算量设为 $I_{\max} = 5\,000$, 种群规模设为 100, 由于通过交叉变异会产生 50 个子代, 所以共计迭代了 100 次. 为了验证算法的有效性, 实验结果如图 1 ~ 图 5 所示.

在进行 DTLZ 系列实验时, 参数选择为: 最大计算量设为 $I_{\max} = 15\,000$, 种群规模设为 100, 由于通过交叉变异会产生 50 个子代, 所以共计迭代了 300 次. 为了验证算法的有效性, 实验结果如图 6 ~ 图 7 所示.

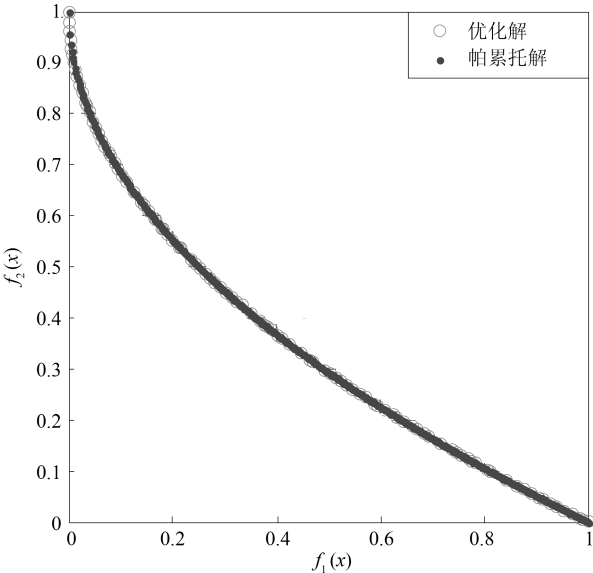


图 1 ZDT1 优化效果
Fig. 1 The optimization effect of ZDT1

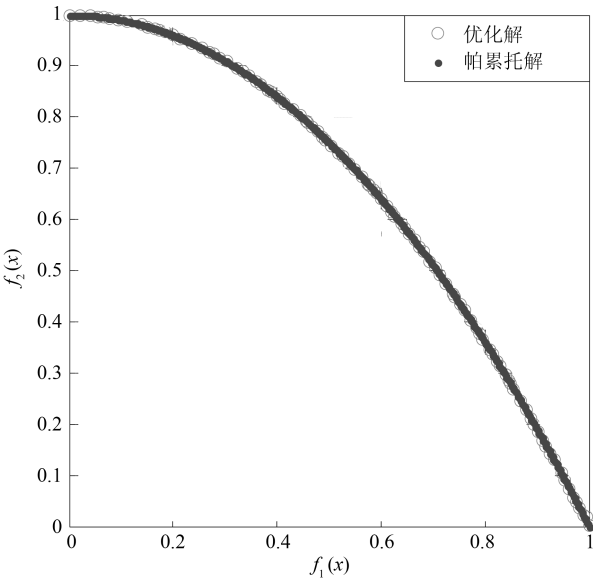


图 2 ZDT2 优化效果
Fig. 2 The optimization effect of ZDT2

由图 1 ~ 图 7 可视, 对于凹、凸和非连续的多目标函数, NSGAII-UID 可以较好地逼近 pareto 前端且分布较均匀, NSGAII-UID 与其他算法对比结果如表 2 所示. 该表分别列出了当函数调用次数分别为 5 000 次和 15 000 次时, 实验结果连续 30 次的最大值、最小值和平均值. 从该表可以看出 NSGAII-UID 在 5 个测试函数 (ZDT1、ZDT2、ZDT4、DTLZ2、DTLZ7) 中 IGD 的最大值、最小值和平均值均优于其他算法. 在 ZDT3 和 ZDT6 中由于 Pareto 前沿在指定区间内存在空白区域, NSGAII-UID 的 IGD 值略小于

表 2 NSGAI-UID 与其他局部搜索算法的 ZDT 和 DTLZ 系列实验 IGD 结果
Table 2 ZDT and DTLZ series performance IGD comparison of different algorithms

	算法	NSGAI-UID	NSGAI-DLS ^[13]	NSGAI ^[18]	AMOPSO ^[11]	MODE ^[16]
ZDT1	最大值	6.526E-03	9.728E-03	6.466E-03	6.153E-03	4.94E-02
	最小值	2.528E-03	4.259E-03	5.367E-03	3.562E-03	4.04E-03
	平均值	3.229E-03	6.427E-03	5.755E-03	4.009E-03	4.54E-03
	标准差	6.700E-05	3.090E-04	3.390E-04	5.700E-05	2.58E-04
ZDT2	最大值	2.732E-03	7.036E-03	5.806E-03	2.493E-03	3.12E-02
	最小值	1.126E-03	4.226E-03	5.134E-03	1.539E-03	3.79E-03
	平均值	1.854E-03	5.026E-03	5.355E-03	1.983E-03	4.31E-04
	标准差	4.970E-05	1.950E-04	2.020E-04	5.200E-05	2.64E-03
ZDT3	最大值	9.236E-03	9.123E-03	6.105E-03	9.160E-03	1.48E-02
	最小值	2.526E-03	5.228E-03	5.447E-03	2.752E-03	3.01E-03
	平均值	4.016E-03	6.326E-03	5.834E-03	3.982E-03	6.24E-03
	标准差	3.623E-03	2.013E-04	2.020E-04	3.623E-03	7.13E-05
ZDT4	最大值	4.237E-03	4.659E-03	1.117E-01	5.845E-03	8.144E-02
	最小值	9.926E-04	4.123E-03	4.623E-03	1.851E-03	3.145E-03
	平均值	3.256E-03	4.326E-03	1.655E-02	4.147E-03	1.021E-02
	标准差	3.160E-04	1.065E-02	3.174E-02	2.980E-04	2.145E-02
ZDT6	最大值	4.735E-03	5.321E-03	1.498E-02	2.110E-04	1.024E-02
	最小值	8.669E-04	2.768E-03	1.119E-02	9.310E-05	6.119E-03
	平均值	1.126E-03	3.412E-03	1.286E-02	1.754E-04	8.155E-03
	标准差	1.075E-04	9.875E-04	1.004E-03	8.600E-05	4.121E-03
DTLZ2	最大值	2.065E-02	1.2334E-01	2.74E-01	9.991E-02	1.221E-01
	最小值	8.431E-02	2.435E-02	7.831E-02	2.180E-02	6.152E-02
	平均值	6.271E-02	1.026E-01	1.059E-01	6.595E-02	9.251E-02
	标准差	6.241E-04	6.345E-03	8.383E-03	7.241E-04	9.251E-03
DTLZ7	最大值	1.015E-02	1.068E-02	3.208E-02	1.552E-02	4.97E-02
	最小值	3.109E-03	3.259E-03	6.14E-03	6.055E-03	7.02E-03
	平均值	1.206E-02	1.365E-02	1.799E-02	1.215E-02	2.87E-02
	标准差	9.324E-04	1.013E-03	1.294E-03	8.600E-04	2.01E-03

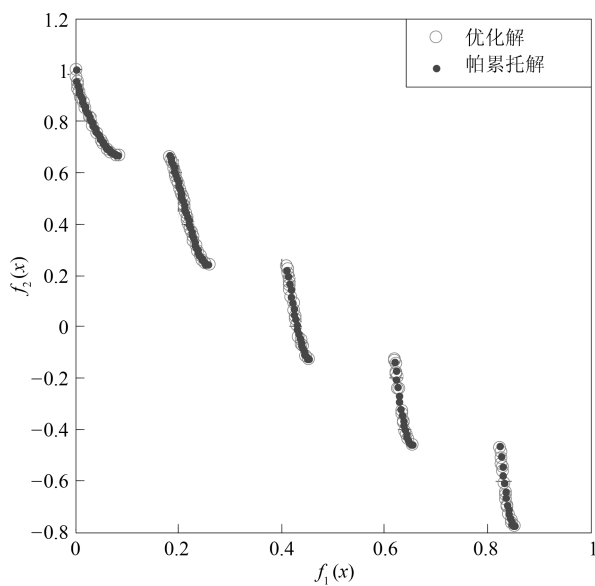


图 3 ZDT3 优化效果图
Fig. 3 The optimization effect of ZDT3

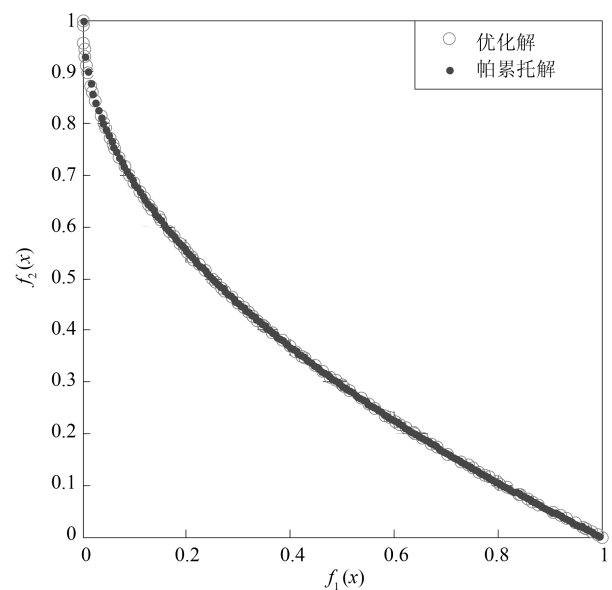


图 4 ZDT4 优化效果
Fig. 4 The optimization effect of ZDT4

表 3 NSGAI-UID 与其他局部搜索算法的 ZDT 和 DTLZ 系列实验 SP 结果
Table 3 ZDT series performance SP comparison of different algorithms

	算法	NSGAI-UID	NSGAI ^[18]	AMOPSO ^[11]	cdMOPSO ^[15]	MODE ^[16]
ZDT1	最大值	1.067E-02	7.538E-02	2.907E-02	1.023E-01	1.01E-01
	最小值	8.687E-03	4.340E-02	1.512E-02	7.732E-02	4.06E-02
	平均值	9.764E-03	5.830E-02	2.551E-02	8.561E-02	5.21E-02
	标准差	5.712E-04	9.385E-03	6.710E-04	1.425E-02	1.43E-02
ZDT2	最大值	9.671E-03	8.287E-03	3.336E-02	2.377E-02	2.25E-02
	最小值	7.248E-03	6.015E-03	1.999E-02	1.012E-02	1.01E-02
	平均值	6.795E-03	7.241E-03	2.257E-02	1.097E-02	1.06E-02
	标准差	7.918E-04	7.410E-04	3.245E-03	6.420E-04	4.02E-04
ZDT3	最大值	1.095E-01	1.066E-01	9.957E-02	8.745E-01	7.63E-01
	最小值	7.264E-02	8.157E-02	6.489E-02	1.036E-01	1.40E-01
	平均值	8.738E-02	9.222E-02	7.025E-02	3.568E-01	3.64E-01
	标准差	8.276E-03	8.415E-03	7.171E-03	2.247E-01	2.14E-01
ZDT4	最大值	9.173E-03	4.425E-02	2.954E-02	3.010E-01	6.251E-02
	最小值	7.968E-03	3.139E-02	2.130E-02	1.369E-01	3.103E-02
	平均值	8.624E-03	3.838E-02	2.654E-02	2.046E-01	3.210E-02
	标准差	6.5316E-04	3.837E-03	4.998E-04	9.556E-02	3.914E-3
ZDT6	最大值	8.734E-03	1.013E-02	4.742E-02	4.021E-02	1.214E-02
	最小值	6.974E-03	6.851E-03	3.218E-02	1.240E-02	3.899E-03
	平均值	7.598E-03	8.266E-03	3.651E-02	3.457E-02	9.145E-03
	标准差	9.648E-04	9.180E-04	1.363E-03	3.884E-03	3.251E-04
DTLZ2	最大值	4.598E-02	7.314E-01	3.553E-01	5.897E-01	7.516E-01
	最小值	8.431E-02	2.145E-02	7.569E-02	9.32E-02	3.145E-02
	平均值	6.271E-02	4.162E-01	2.399E-01	3.562E-01	4.021E-01
	标准差	8.381E-04	3.655E-02	9.732E-03	1.772E-02	4.215E-02
DTLZ7	最大值	6.983E-01	7.466E-01	7.564E-01	9.307E-01	9.31E-01
	最小值	4.027E-02	6.32E-02	4.491E-02	1.347E-01	1.35E-01
	平均值	9.921E-02	4.191E-01	3.758E-01	5.972E-01	5.97E-01
	标准差	8.169E-03	7.961E-03	7.593E-03	2.133E-01	2.45E-01

AMOPSO^[11] 算法, 在后续的工作中将会就此问题进行更深一步的研究. 因此从以上结果可以看出和对比算法相比, 该算法具有较好的精度和收敛性.

由图 1~图 7 可视, 对于凹、凸和非连续的多目标函数, NSGAI-UID 可以较好地逼近 Pareto 前端且分布较均匀, NSGAI-UID 与其他算法对比结果如表 2 所示. 该表分别列出了当函数调用次数分别为 5 000 次和 15 000 次时, 实验结果连续 30 次的最大值、最小值和平均值. 从该表可以看出 NSGAI-UID 在 5 个测试函数 (ZDT1、ZDT2、ZDT4、DTLZ2、DTLZ7) 中 IGD 的最大值、最小值和平均值均优于其他算法. 在 ZDT3 和 ZDT6 中由于 Pareto 前沿在指定区间内存在空白区域, NSGAI-UID 的 IGD 值略小于 AMOPSO^[11] 算法, 在后续的工作中将会就此问题进行更深一步的研究. 因此从以上结果可以看出和对比算法相比, 该算法具有较好的精度和收敛性.

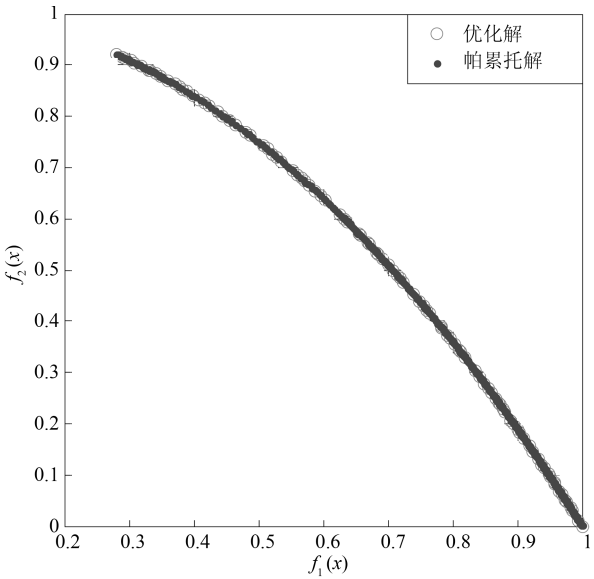


图 5 ZDT6 优化效果
Fig. 5 The optimization effect of ZDT6

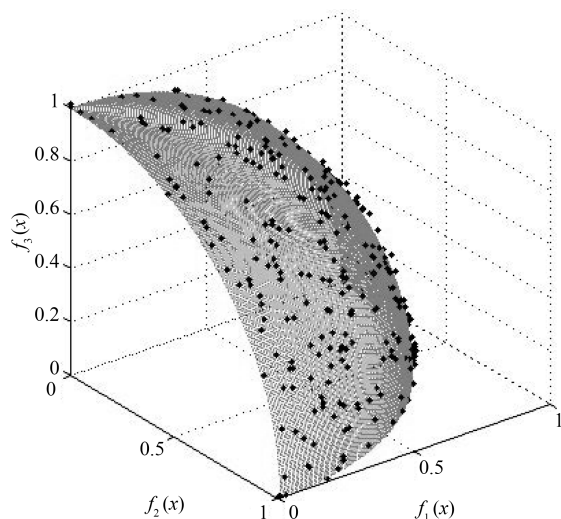


图 6 DTLZ2 优化效果图

Fig. 6 The optimization effect of DTLZ2

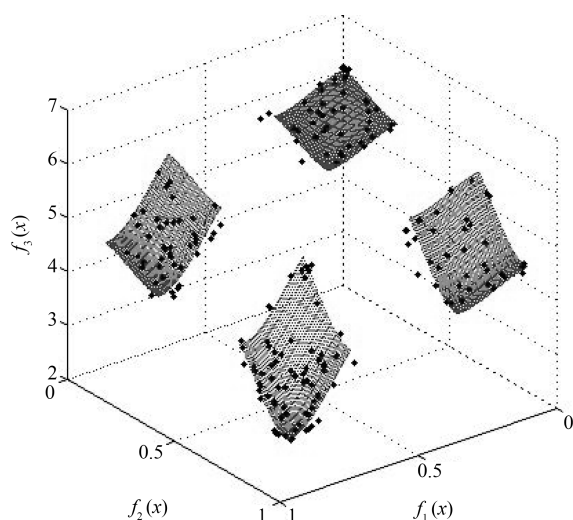


图 7 DTLZ7 优化效果

Fig. 7 The optimization effect of DTLZ7

3.2 实验 2

参照文献 [11] 的实验, 对算法 NSGAII-UID 的分布性进行检验, 不同多目标优化算法采用相同的参数, 具体参数设置同实验 1 所示. 该实验共与 4 个算法 (AMOPSO^[11]、cdMOPSO^[19]、NSGAII^[8]、MODE^[17]) 进行了比较, 取连续 30 次实验平均值作为结果如表 3 所示.

由表 3 可见 NSGAII-UID 算法在 6 个测试函数 (ZDT1、ZDT2、ZDT4、ZDT6、DTLZ2、DTLZ7) 的 SP 最大值、最小值和平均值均小于其他对比算法. 在 ZDT3 中 SP 值略小于 AMOPSO^[11] 算法, 由以上结果显示该算法和其他算法相比算法相比具有更好的分布性.

3.3 实验 3

为了验证算法的收敛速度, 本文采用记录算法达到指定性能指标时所调用的函数次数的方法. 参照文献 [9] 的实验, 设定当 IGD 值达到 0.01 时停止优化, 记录函数调用次数. 实验结果如表 4 所示.

表 4 NSGAII-UID 与其他局部搜索算法的函数计算次数结果

Table 4 Function calculation comparison of different algorithms

算法	目标函数调用次数	函数调用次数	最大值	最小值	平均值
NSGAII-UID	2 800	2 800	3 200	1 850	1 950
NSGAII-DLS	3 010	2 240	3 500	2 030	2 100
NSGAII ^[2]	20 000+	20 000+	20 000+	20 000+	5 150
HMOEA/D ^[8]	13 200	9 600	10 500	7 050	6 900
NSGAII-els	13 350	8 550	17 250	13 200	3 000

由表 4 可见: 在 ZDT1-ZDT4 和 ZDT6 的实验中, 当 IGD 达到设定值时, 连续实验 10 次的平均值, NSGA2-UID 所用函数调用次数均小于其他方法, 因此 NSGA2-UID 达到指定标准消耗的计算量更少, 即该算法收敛到 Pareto 前沿速度更快.

4 结论

针对 NSGAII 在种群进化过程中会出现解分布不均的问题, 本文提出了一种基于均匀分布的 NSGAII (NSGAII-UID) 多目标优化算法. 为了使解能够在目标空间均匀分布, 而真正的 Pareto 前沿大都是曲线或者是曲面, 本文采用映射的方法, 将解映射到目标空间对应的超平面, 并在该平面均匀分区. 当对应分区的解不满足均匀性时, 均匀性加强模块被启用. 同时采用聚类分析的方法来维持和增加进化种群的多样性和分布性. 为了验证算法的性能, 本文采用 5 个 ZDT 函数和示, 两个 DTLZ 函数来进行实验, 实验结果显示该算法和其他多目标优化算法相比具有更好的收敛性和分布性, 同时收敛速度较快.

References

- 1 Schaffer J D. Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the 1st International Conference on Genetic Algorithms. Hillsdale, NJ, USA: L. Erlbaum Associates, Inc., 1985. 93–100
- 2 Fonseca C M, Fleming P J. Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In: Proceedings of the 5th International Conference. San Mateo, CA: Morgan Kauffman Publishers, 1993. 416–423
- 3 Srinivas N, Deb K. Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evolutionary Computation*, 1994, 2(3): 221–248

- 4 Deb K, Agrawal S, Pratap A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature. Berlin, Heidelberg: Springer, 2000. 849–858
- 5 Ahmadi A. Memory-based adaptive partitioning (MAP) of search space for the enhancement of convergence in Pareto-based multi-objective evolutionary algorithms. *Applied Soft Computing*, 2016, **41**: 400–417
- 6 Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the 2nd International Conference on Genetic Algorithms. Hillsdale, NJ, USA: L. Erlbaum Associates, Inc., 1987. 41–49
- 7 Zhu Xue-Jun, Chen Tong, Xue Liang, Li Jun. Pareto multi-objective genetic algorithm with multiple individual participation. *Acta Electronica Sinica*, 2001, **29**(1): 106–109
(朱学军, 陈彤, 薛量, 李峻. 多个体参与交叉的 Pareto 多目标遗传算法. 电子学报, 2001, **29**(1): 106–109)
- 8 Corne D W, Knowles J D, Oates M J. The Pareto envelope-based selection algorithm for multiobjective optimization. In: Proceedings of the International Conference on Parallel Problem Solving from Nature. Berlin, Germany: Springer-Verlag, 2000. 839–848
- 9 Knowles J, Corne D. Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation*, 2003, **7**(2): 100–116
- 10 Morse J N. Reducing the size of the nondominated set: pruning by clustering. *Computers & Operations Research*, 1980, **7**(1–2): 55–66
- 11 Han H G, Lu W, Qiao J F. An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. *IEEE Transactions on Cybernetics*, 2017, **47**(9): 2754–2767
- 12 Li San-Yi, Li Wen-Jing, Qiao Jun-Fei. A local search strategy based on density for NSGA2 algorithm. *Control and Decision*, 2018, **33**(1): 60–66
(栗三一, 李文静, 乔俊飞. 一种基于密度的局部搜索 NSGA2 算法. 控制与决策, 2018, **33**(1): 60–66)
- 13 Sindhya K, Miettinen K, Deb K. A hybrid framework for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(4): 495–511
- 14 Yang S X. Genetic algorithms with memory- and elitism-based immigrants in dynamic environments. *Evolutionary Computation*, 2008, **16**(3): 385–416
- 15 Helwig S, Branke J, Mostaghim S. Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(2): 259–271
- 16 Basu M. Economic environmental dispatch using multi-objective differential evolution. *Applied Soft Computing*, 2011, **11**(2): 2845–2853
- 17 Kim H, Liou M S. Adaptive directional local search strategy for hybrid evolutionary multiobjective optimization. *Applied Soft Computing*, 2014, **19**: 290–311
- 18 Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(2): 182–197
- 19 Sindhya K, Miettinen K, Deb K. A hybrid framework for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(4): 495–511



乔俊飞 北京工业大学教授. 主要研究方向为智能控制, 神经网络分析与设计. 本文通信作者.

E-mail: junfeq@bjut.edu.cn

(QIAO Jun-Fei Professor at Faculty of Information Technology, Beijing University of Technology. His research interest covers intelligent control, analysis and design of neural networks. Corresponding author of this paper.)



李 霏 北京工业大学博士研究生. 主要研究方向为智能控制, 多目标优化, 神经网络结构设计和优化.

E-mail: lfglw521@163.com

(LI Fei Ph.D. candidate at Faculty of Information Technology, Beijing University of Technology. Her research interest covers intelligent control, multi-objective optimization algorithm, analysis and design of neural networks.)



杨翠丽 北京工业大学讲师. 主要研究方向为神经网络和智能优化算法.

E-mail: clyang5@bjut.edu.cn

(YANG Cui-Li Lecture at Faculty of Information Technology, Beijing University of Technology. Her research interest covers neural network and intelligent optimization algorithm.)