

一种求解多目标优化问题的进化算法混合框架

田红军[†], 汪 镭, 吴启迪

(同济大学 电子与信息工程学院, 上海 201804)

摘要: 为了提高多目标优化算法的求解性能, 提出一种启发式的基于种群的全局搜索与局部搜索相结合的多目标进化算法混合框架. 该框架采用模块化、系统化的设计思想, 不同模块可以采用不同策略构成不同的算法. 采用经典的改进非支配排序遗传算法 (NSGA-II) 和基于分解的多目标进化算法 (MOEA/D) 作为进化算法的模块算法来验证所提混合框架的有效性. 数值实验表明, 所提混合框架具有良好性能, 可以兼顾算法求解的多样性和收敛性, 有效提升现有多目标进化算法的求解性能.

关键词: 多目标优化; 进化算法; 混合框架; NSGA-II; MOEA/D

中图分类号: TP18

文献标志码: A

A hybrid framework of evolutionary algorithm for solving multi-objective optimization problems

TIAN Hong-jun[†], WANG Lei, WU Qi-di

(College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

Abstract: In order to improve the solve performance of multi-objective evolutionary algorithms, a heuristic hybrid framework for evolutionary multi-objective optimization combining the global search and local search based on populations is proposed. The framework adopts modular, systematic design thoughts and different strategies can be used in different modules to form different algorithms. The classic non-dominated sorting genetic algorithm II (NSGA-II) and multi-objective evolutionary algorithm based on decomposition (MOEA/D) algorithm are used as the algorithm of evolutionary algorithm module to verify the effectiveness of the proposed hybrid framework. The numerical experiments show that, the proposed hybrid framework has good performance, which can achieve a balance between diversity and convergence of algorithm and effectively improve the solve performance of existing multi-objective evolutionary algorithms.

Keywords: multi-optimization; evolutionary algorithm; hybrid framework; NSGA-II; MOEA/D

0 引言

进化算法是模拟自然界生物群体进化过程的随机优化方法, 具有不依赖于问题模型的特性、寻优过程的自适应性、隐含的并行性以及解决复杂非线性问题的鲁棒性等特点, 已广泛地应用于复杂问题和优化设计. 相对于其他传统多目标求解方法, 多目标进化算法一次求解往往可以得到多个 Pareto 最优解, 多目标进化算法可以处理任何不连续的、多峰的、非凸的目标函数以及具有良好的普适性, 因此, 多目标进化算法自诞生以来已成功地应用于交通运输、电子与电气工程、通信与网络、管理工程、航空航天、金融等领域并取得了不错的成绩^[1-3]. 在过去数十年中, 许多学者提出了不同的进化算法以解决多目标优化问

题. 然而, 根据“世界上没有免费的午餐”理论, 没有一种算法是万能的, 人工蜂群算法、布谷鸟搜索算法、蝙蝠算法、萤火虫算法、粒子群算法、遗传算法、差分进化算法等都有其优点和缺点; 另外, 这些算法往往存在全局搜索能力与收敛速度之间的矛盾, 仅通过调节控制参数往往无法达到“探索”与“开发”能力之间的平衡. 大量研究表明, 融入一些其他信息, 如近似算法、局部搜索技术、特殊组合算子等, 能够极大地改善进化算法的搜索性能和求解精度, 将全局优化算法混合以及与局部搜索策略进行集成, 充分发挥各种算法的优势, 构成混合优化算法, 已经被证明是很有效的策略^[4-9].

到目前为止, 人们相继提出了许多混合多目标

收稿日期: 2016-07-22; 修回日期: 2016-10-03.

基金项目: 国家自然科学基金项目 (61075064, 61034004, 61005090); 教育部新世纪人才计划项目 (NCET-10-0633).

作者简介: 田红军 (1986—), 男, 博士生, 从事智能优化、进化计算的研究; 汪镭 (1971—), 男, 教授, 博士生导师, 从事智能优化、人工智能、进化计算等研究.

[†]通讯作者. E-mail: tianhongjun86@163.com

进化算法^[10-14]. Ishibuchi等^[10]首次将局部搜索融入多目标遗传算法求解多目标优化问题,提出了多目标遗传局部搜索算法(MOGLS). MOGLS算法采用加权和方法生成局部搜索的目标函数,以进化算法产生的后代种群作为初始点优化该加权函数,以此提高解的质量. Goel等^[11]提出了两种方法将局部搜索融入进化算法中,一种是MOGLS算法采用的在线方法,一种是后验方法. Deb等^[12-13]以NSGA-II作为多目标优化算法,以标量化函数——成绩标量函数(ASF)为目标函数进行局部搜索. 文献[14]提出了一种基于局部搜索和混合多样性策略的多目标粒子群算法. 在上述算法中,尽管融入局部搜索的多目标进化算法收敛性能有所提高,然而局部搜索只用于改进进化算法得到的解的质量,没有考虑所得解分布的均匀性,算法并不能很好地兼顾解的分布性和均匀性. 另外,上述混合算法往往只针对某一种算法进行改进,并没有提出一种系统化的混合框架.

反馈是控制领域的灵魂. 基于这一理念,本文在现有算法基础上,基于“检测-响应”的反馈控制思想,通过对种群进化状态进行检测来采取不同的进化策略,提出一种启发式的基于种群的全局搜索与局部搜索相结合的多目标进化算法混合框架. 该框架采用模块化设计思想,在这个框架下,不同模块可以采用不同的策略构成不同的混合算法. 与已有的改进的混合算法相比,所提出的混合框架基于模块化系统设计思想,投影聚类模块可以维持和增加进化群体的多样性和分布性,局部搜索模块和种群多样性增强模块可使算法兼顾解的多样性和收敛性. 这样的结合机制使其搜索效率及求解质量优于单一算法,有效提升现有多目标进化算法的求解性能,可应用于求解许多问题并得到满意的结果. 数值实验验证了所提框架的有效性.

1 多目标优化问题

不失一般性,以最小化问题为例,多目标优化问题的定义如下.

定义1(多目标优化问题) 在空间 Ω 中寻找向量 $x = (x_1, x_2, \dots, x_n)$,使得目标函数 $F(x)$ 中的 m 个目标 $f_i(x)$ ($i = 1, 2, \dots, m$)最小,即

$$\min_{x \in \Omega} y = F(x) = \{f_1(x), f_2(x), \dots, f_m(x)\}. \quad (1)$$

通常,当目标个数 m 大于3时,称为高维多目标优化问题;相应地,目标个数为2或3的问题称为低维多目标优化问题.

定义2(Pareto支配) 假设 $x_a, x_b \in \Omega$ 是式(1)所示多目标优化问题的两个可行解,当且仅当

$$\forall i \in \{i = 1, 2, \dots, m\}, f_i(x_a) \leq f_i(x_b),$$

$$\text{且} \exists j \in \{j = 1, 2, \dots, m\}, f_j(x_a) < f_j(x_b), \quad (2)$$

则称与 x_b 相比, x_a 是Pareto占优的,记作 $x_a \prec x_b$,也称 x_a 支配 x_b .

定义3(Pareto最优解) 可行解 $x^* \in X$ 被称为Pareto最优解(或非支配解),当且仅当 x^* 满足如下条件:

$$\neg \exists x \in \Omega : x \prec x^*. \quad (3)$$

定义4(Pareto最优解集(PS)) 决策空间 X 中的所有Pareto最优解构成的集合称为Pareto最优解集,定义如下:

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

定义5(Pareto前沿(PF)) Pareto最优解集 PS 中的所有Pareto最优解在目标空间中对应的目标向量组成的集合称为Pareto前沿,即

$$PF = \{x^* \in PS | F(x^*) = (f_1(x^*), f_2(x^*), \dots, f_m(x^*))\}. \quad (5)$$

多目标优化问题就是获取一组在目标空间中尽量靠近Pareto前沿且均匀分布的解.

2 多目标进化算法混合框架

在混合多目标进化算法中,影响多目标混合算法的因素有很多,如局部搜索所采用的标量函数、应用局部搜索的概率、局部搜索的个体选择以及引入局部搜索后如何平衡“探索”与“开发”能力等. 针对这些问题,本文提出一种模块化的多目标进化算法混合框架. 该框架包含不同的功能模块,包括多目标进化算法模块、投影聚类模块、局部搜索模块和多样性增强模块,这些模块可以采用不同策略组成不同算法. 混合框架可以协调处理上述影响混合算法的各种因素,算法框架如图1所示.

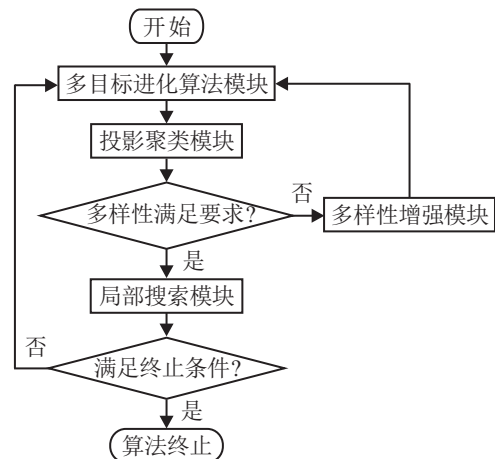


图1 多目标进化算法混合框架

在所提出的框架中,进化算法模块可以采用现有多目标进化算法及其改进算法;投影聚类模块维持并增加进化群体的多样性和分布性,以聚类质量指标作为种群多样性的好坏指标进行局部搜索或种群多样性增强操作;局部搜索模块在种群具有较好多样性时可加快算法收敛;多样性增强模块在保证收敛性的同时可增加种群多样性以保证解的均匀性和分布性。

3 框架分析验证

本节将采用经典的NSGA-II算法和MOEA/D算法来验证所提多目标进化算法混合框架的有效性,并对基于NSGA-II算法的混合算法设计流程进行详细描述,给出各模块所采用的策略。

3.1 成绩标量函数(ASF)

成绩标量函数(ASF)是Wierzbicki^[15]于1980年提出的.用它可以多个目标通过聚合函数转化为单目标优化问题求解.ASF函数通过加权向量 w 和参考点 $\bar{z}$ 来构造,其中参考点 $\bar{z}$ 既可以在可行目标空间内,也可以不在可行目标空间内.对于式(1)给出的多目标优化问题,利用ASF函数 $F(x)$ 将式(1)转化为单目标优化问题的方法如下:

$$\begin{cases} \min F(x) = \max_{i=1,2,\dots,m} w_i(f_i(x) - \bar{z}) + \\ \quad \rho \sum_i w_i(f_i(x) - \bar{z}), \\ \text{s.t. } x \in D. \end{cases} \quad (6)$$

其中: $w = (w_1, w_2, \dots, w_m)$ 是给定的将多个目标进行标量化的加权向量, $w_i = 1/(z_i^{\max} - z_i^{\min})$, $z_i^{\max}$ 和 $z_i^{\min}$ 分别为种群中每一目标函数的最大最小值;系数 ρ 取一非常小的正值,如 $\rho = 10^{-6}$.对于ASF函数(6),已经证明对于任意的参考点 $\bar{z}$,它的解是多目标优化问题(MOP)的(弱)Pareto最优解^[16].然而,式(6)包含有不可微的目标函数,ASF函数的求解不能利用函数梯度信息的传统数学规划方法.当多目标优化问题(1)的目标函数可微时,本文引入实数变量 α 和 k 进行极小化转换,即

$$\begin{cases} \min \alpha + \rho \sum_i w_i(f_i(x) - \bar{z}); \\ \text{s.t. } w_i(f_i(x) - \bar{z}) \leq \alpha, \\ \quad i = 1, 2, \dots, m, \\ \quad x \in D, \alpha \in R. \end{cases} \quad (7)$$

近年来,ASF函数主要用于求解偏好多目标优化问题,许多学者研究了权向量取值对于决策者偏好的

影响^[17-18].基于ASF函数的局部搜索可搜索PF特定区域的解,不但能在进化初始阶段加快算法的收敛速度,而且能够改善归档集中非支配解在PF上分布的均匀性^[19].本文采用ASF为目标函数的局部搜索可以改善非支配解在PF上分布的均匀性,同时加快进化算法收敛。

3.2 多目标进化算法模块

多目标进化算法模块主要包含进化、选择、变异等操作,该模块可以采用一种完整的多目标优化算法,也可以采用多种进化算法的混合.本文采用经典的NSGA-II和MOEA/D算法来验证所提混合框架的有效性。

3.3 投影聚类模块

在该模块中,各种聚类算法都可以应用在模块2中.种群个体投射到一个超平面上并在目标空间内进行聚类操作,聚类的目的是维持并增加进化群体的多样性和分布性,它将进化群体中的个体划分为若干个由相似个体组成的子种群,通过计算个体或类之间的相似度来实现.将聚类的质量指标作为种群多样性的近似度量,保持种群多样性以搜索更广的目标空间并获得均匀分布的Pareto前沿是非常必要的.该模块算法设计如下。

图2为种群个体投射超平面示意图.其中:实心圆点1~5代表种群 P_t 中的个体;正方形 a, b, c 为它们在超平面 H 上的正交投影,属于种群 P'_t 中的个体.可以看出,尽管个体2、3、4是种群 P_t 中的不同个体,但是经正交投影后,在种群 P'_t 中用同一个个体 b 表示。

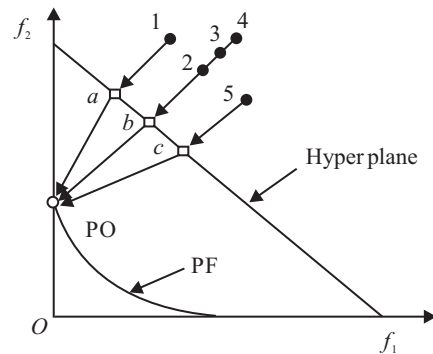


图2 种群个体投射超平面示意图

在图2中,种群 P'_t 中的所有个体形成单个聚类.对种群 P'_t 进行局部搜索会产生同一个Pareto最优解.需要指出的是,如果每一次局部搜索都产生同一个Pareto最优解,则无疑会降低种群的多样性,关于这一点将在种群多样性增强模块中探讨.如果聚类质量指标 Q 值较小,则说明种群 P'_t 中个体非常接近;若 Q 值较大,则种群多样性较好.在图2中,当 P'_t 中个

体都属于同一个聚类时, Q 取值较低. 这时保持种群 P'_t 的良好分布性, 探索 Pareto 前沿的不同区域以获得不同的 Pareto 最优解非常重要.

算法1 投影聚类模块.

模块输入: 种群规模 N , 聚类数 K , 第 t 代种群 P_t ;

模块输出: 第 t 代聚类质量指标 Q -current, 第 t 代聚类 $RP_i^t, i = 1, 2, \dots, K$.

Step 1: 考虑包含目标向量的解集 Ψ , 有

$$F_i = (0, \dots, 0, f_i^{\max}, 0, \dots, 0)^T \in R^k,$$

其中 $f_i^{\max}$ 是种群 P_t 中目标函数 f_i 的最大值. 定义超平面 H 为

$$H = \{f \in R^k : \langle w^S, f \rangle + b^S = 0\}.$$

其中: H 是 k 维欧氏空间里的一个 $k-1$ 维线性子空间. $\langle \cdot, \cdot \rangle$ 为欧氏点积, $w^S = \left(\frac{1}{f_1^{\max}}, \dots, \frac{1}{f_m^{\max}}\right)$, $b^S = -1$, $\Psi \subset H$. 目标向量 f 在超平面 H 上的正交投影 PI 定义如下:

$$PI = \frac{1 - \langle w^S, f \rangle}{\|w^S\|^2} w^S + f. \quad (8)$$

利用式(8), 种群 P_t 中所有个体在超平面 H 上的正交投影即可得到种群 P'_t , 在式(8)中, 若 $f_i^{\max} = 0$, 则令 $f_i^{\max} = 10^{-06}$.

Step 2: 种群 P'_t 生成 K 个聚类, 每个聚类的子群记为 $RP_i^t, i = 1, 2, \dots, K$.

Step 3: 评价种群 P'_t 的多样性, 即

$$Q = \sum_{i=1}^K \frac{1}{|RP_i^t|} \sum_{P_j \in RP_i^t} (D(P_j, \sigma_i)). \quad (9)$$

其中: K 为聚类数, σ_i 为第 i 个聚类的中心点, P_j 为第 i 个聚类中中心点之外的某一个点, $D(P_j, \sigma_i)$ 为 P_j 到 σ_i 的欧氏距离, $|RP_i^t|$ 为第 i 个聚类的个体数.

Step 4: 用种群 P_t 中的相应个体更新每个聚类中的子群.

算法中, Q -current 为第 t 代聚类质量指标, 利用 Q -current 为后面 t_c 次迭代设置下限阈值 Q -bound = $0.25 \times Q$ -current. 从 $(t+1)$ 代起, 每一代 Q -current 都与 Q -bound 相比较, 直到第 $(t+t_c)$ 次迭代完成. 在这 t_c 次迭代中, 如果 Q -current 小于 Q -bound, 则表示种群具有较差的种群多样性, 此时, 种群多样性增强模块会被激活以增加种群多样性; 反之, 种群具有较好的种群多样性, 种群多样性增强模块不会被激活. 研究表明, 在该模块中, Q -bound 代替第 t 代的 Q -current 作为后续 t_c 次迭代的下限阈值, 既可以防止陷入局部最优, 又可以防止频繁激活种群多样性增强模块.

3.4 局部搜索模块

局部搜索模块如算法2所示, 该模块以概率 p_{local} 选择改进个体, 加快算法收敛和提高算法求解精度. 只有当投影聚类模块的聚类质量指标表示种群具有良好的种群多样性时该模块才会被激活, 否则算法转入种群多样性增强模块, 增加种群多样性.

算法2 局部搜索模块.

模块输入: 第 t 代种群 P_t , 第 t 代聚类 $RP_i^t, i = 1, 2, \dots, K$, 局部搜索概率 p_{local} ;

模块输出: 改进的 t 代种群 p'_t .

Step 1: 随机选择一个聚类 i , 以概率 p_{local} 从 RP_i^t 中选择个体 A , 如果没有个体被选中, 则终止算法.

Step 2: 利用式(7), 令 $\bar{z} = f$, f 为被选中个体 A 的目标向量, 然后利用式(7)采用合适的局部搜索算法进行局部搜索.

Step 3: 利用局部搜索获得的解取代 P_t 中与 A 相对应的个体. 更新后的种群记为 p'_t .

Step 4: 根据所选择的全局进化算法, 重置算法参数, 以 NSGA-II 作为全局搜索算法为例, 重置种群 p'_t 个体排序和拥挤距离.

3.5 种群多样性增强模块

只有当投影和聚类模块检测到种群具有较差的种群多样性时, 该模块才会被激活以增加种群多样性. 算法3适用于 NSGA-II, 也可用于其他多目标进化算法.

算法3 多样性增强模块.

模块输入: 第 t 次迭代混合种群 $R_t = P_t \cup Q'_t$, 种群规模 N ;

模块输出: 第 $t+1$ 次迭代父代种群 P_{t+1} .

Step 1: 将规模为 $2N$ 的种群 R_t 作为投影和聚类模块输入.

Step 2: 对 RP_j^t 的每一个子种群进行非支配排序, 确定不同的 Pareto 前沿 $F_i^j, i = 1, 2, \dots, j = 1, 2, \dots, K$.

Step 3: 初始化 $P_{t+1} = \emptyset, i = 1, j = 1$. 如果 $|P_{t+1}| + |F_i^j| \leq N$, 则 $P_{t+1} = P_{t+1} \cup F_i^j, j = j+1$; 如果 $j > K$, 则令 $j = 1, i = i+1$.

Step 4: 根据所选择的全局进化算法, 重置算法参数, 以 NSGA-II 作为全局搜索算法为例, 重置种群 P_{t+1} 个体排序和拥挤距离.

在混合多目标进化算法中, 局部搜索中会产生远离当前种群的个体, 该个体适应度远大于其他个体的适应度, 在种群中占主导地位, 称之为超级个体 (superindividual), 如图3所示. 超级个体会支配当前

整个种群,图3中灰色区域所示,超级个体由于其主导地位,其进入交配池繁殖后代的几率远大于种群中其他个体,由超级个体增加的选择压力会导致其他支配解的迅速删除.由于种群多样性增强模块只作用于非支配解,导致种群多样性的丢失,选择合适的种群多样性增强策略有利于增加种群多样性.另外,目标空间的多样性可以提高混合框架的求解效率,局部搜索模块可以根据不同的参考点获取Pareto前沿不同区域的解.在该模块中,既可以采用与多目标进化算法全局搜索模块相同的多样性增强策略,也可以采取其他不同的多样性增强策略.

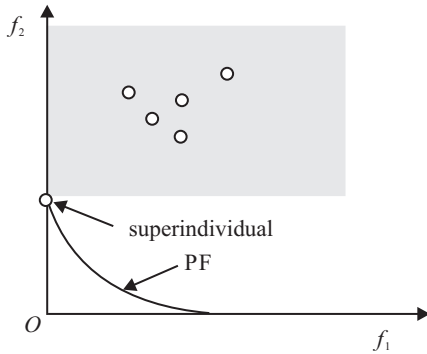


图3 超级个体(superindividual)

针对超级个体带来的问题和NSGA-II算法,本文给出一种种群多样性增强策略,如图4所示.

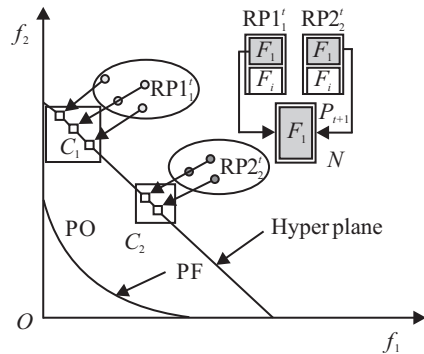


图4 NSGA-II多样性增强策略

图4中,RP1_t和RP2_t为种群R_t投影聚类前的两个子种群,与之一对应的聚类为C₁和C₂.对RP1_t和RP2_t进行非支配排序得到不同Pareto前沿F₁和F_i,然后利用两个子群中的F₁个体获得规模为N的种群P_{t+1}.

本节中提出了一种多目标进化算法混合框架,下面结合经典的NSGA-II和MOEA/D算法论证并分析各模块作用及所提框架的有效性.

4 数值仿真实验

4.1 性能评价标准

在本节数值实验测试中,采用IGD指标来度量算法的性能.IGD指标在一定程度上能同时反映出所

求得的近似Pareto前沿的多样性和收敛性.令P*为均匀分布的理想Pareto前沿,P是通过多目标进化算法得到的近似Pareto前沿.IGD指标度量P*到P的距离,定义为

$$\begin{cases} \text{IGD}(P^*, P) = \sum_{y^* \in P^*} d(y^*, P) / |P^*|, \\ d(y^*, P) = \min_{y \in P} \sqrt{\sum_{i=1}^m (y_i^* - y_i)^2}. \end{cases} \quad (10)$$

其中: $y^* = (y_1^*, y_2^*, \dots, y_m^*)^T$, $y = (y_1, y_2, \dots, y_m)^T$, m 为目标函数的个数, $d(y^*, p)$ 为 y^* 距离近似Pareto前沿P的最小欧氏距离.

4.2 基于NSGA-II的算法框架验证

4.2.1 算法流程

NSGA-II算法是目前最常用的多目标进化算法之一,它以Pareto占优关系比较个体优劣,为了保持种群的多样性,采用拥挤距离比较同一等级中的个体.这是迄今为止最适合用来解决2或3个目标的优化问题的算法之一,与NSGA-II算法相对应,将其称为混合NSGA-II算法(HNSGA-II).基于所提混合框架的混合NSGA-II算法流程如算法4所示.

算法4 HNSGA-II.

Step 1: 随机生成种群规模为N的初始种群P_t,迭代次数t = 0.

Step 2: 对初始种群进行非支配排序得到种群P'_t.

Step 3: 对父代种群P'_t采用锦标赛选择方法进行遗传操作,得到规模为N的子代种群Q_t.

Step 4: 合并父代和子代种群R_t = P'_t ∪ Q_t.

Step 5: 对种群R_t进行非支配排序得到不同的Pareto前沿F_i, i = 1, 2, ...

Step 6: 令P_{t+1} = ∅, i = 1. 如果|P_{t+1}| + |F_i| ≤ N, 则P_{t+1} = P_{t+1} ∪ F_i, i = i + 1; 否则转Step 7, |P_{t+1}|和|F_i|分别为P_{t+1}、F_i的数量.

Step 7: 对F_i中的个体计算拥挤距离并按降序排列,选择F_i中的前N - |P_{t+1}|个体并入P_{t+1}.

Step 8: 将P_{t+1}作为投影聚类模块的输入进行投影聚类得到种群P'_{t+1}.

Step 9: 若种群P'_{t+1}具有良好的种群多样性,则转Step 10; 否则转Step 11.

Step 10: 将种群P'_{t+1}作为局部搜索模块的输入进行局部搜索,然后转Step 12.

Step 11: 采用算法3,将种群P'_{t+1}作为种群多样性增强模块的输入,增加种群多样性;然后转Step 2.

Step 12: 检查算法是否满足终止条件,若满足则转Step 13; 否则令t = t + 1, 转Step 2.

Step 13: 退出算法.

算法4中,多目标进化算法模块采用NSGA-II算法,Step 1~Step 7与标准NSGA-II算法一样. Step 8~Step 13融入了局部搜索模块及多样性增强模块,多样性增强模块采用算法3所示算法.局部搜索模块采用序列二次规划(SQP)方法求解ASFs^[19]. SQP方法最早由Wilson^[20]于1963年提出,特别适用于非线性约束规划问题,其算法思想是将原问题的Lagrange方程进行二次近似,形成QP子问题,即

$$\begin{cases} \min_{d \in R^n} \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d, \\ \text{s.t. } \nabla g(x_k)^T d + g(x_k) \leq 0. \end{cases} \quad (11)$$

其中:下标 k 为当前迭代次数; H 是Hession矩阵,可由拟牛顿法近似求出. QP子问题的解作为下一步迭代的线性搜索方向.

4.2.2 实验分析

在本实验设计中,选用多目标测试函数中的两个目标函数的ZDT4、3个目标函数的DTLZ1-6以及4个目标函数的DTLZ2作为测试函数. DTLZ2、DTLZ3和DTLZ4测试问题都具有凹的Pareto最优前沿,DTLZ3在DTLZ2的基础上增加了更多的局部最优前沿,容易使算法陷入局部最优;DTLZ4对DTLZ2进行修改,使其能够用来测试算法的分布性;ZDT4测试函数含有多个局部Pareto最优,可以很好地测试算法性能. 仿真采用Matlab2013a,实验平台:win7企业版,Pentium(R) Dual-Core CPU E5300 @2.60 GHz,32位操作系统,4.00 G RAM. 两种算法的种群规模设置为200;交叉算子设为0.9;变异因子设为 $1/n$, n 为变量个数;SBX和变异分布指数分别设为15和20;每种算法独立运行30次;HN算法投影聚类模块采用K-mean聚类算法. 算法运行停止条件为最大函数值评估次数. 由于实验中测试函数的计算复杂性,对于不同的问题本文采用不同的函数值评估次数. 根据相关研究,非支配解收敛至最优Pareto附近需要的函数值评估次数分别为:

- 1) DTLZ2, DTLZ4, DTLZ5: 5 000;
- 2) DTLZ6: 15 000;
- 3) ZDT4, DTLZ1, DTLZ24: 20 000;
- 4) DTLZ3: 5 000.

局部搜索停止条件为KNITRO终止准则(KKT optimality error)^[21].

- 1) 参数 P_{local} .

P_{local} 取值不同会影响算法的种群多样性和收敛

性:较大 P_{local} 取值意味着更多个体参与局部搜索,这样会增大生成超级个体的概率和消耗更多的函数评价次数;较小的 P_{local} 取值会导致局部搜索加速收敛的能力不足. 因此本文采用启发式方法研究 P_{local} 应如何取值. 采用上述的参数设置,根据下式设置 P_{local} 不同的取值:

$$P_{\text{local}} = 1/(\eta \cdot N), \quad \eta = 1, 2, \dots, 5, \quad (12)$$

其中 N 为种群规模. 通过改变 P_{local} 的取值测试HNSGA-II算法对不同测试问题对 P_{local} 的敏感性,对所选8个测试函数运行结果如表1所示.

表1 P_{local} 取值

test problem	FE	$P_{\text{local}} = 1/(\eta \cdot N)$				
		medium IGD value(取值越小越好)				
		$\eta = 1$	$\eta = 2$	$\eta = 3$	$\eta = 4$	$\eta = 5$
ZDT4	20 000	0.005 4	0.005 8	0.005 2	<i>0.005 1</i>	0.005 0
DTLZ1	20 000	0.440 6	<i>0.399 1</i>	0.365 8	0.453 2	0.524 0
DTLZ2	5 000	0.056 3	0.056 9	0.055 1	<i>0.055 2</i>	0.055 8
DTLZ3	50 000	0.200 1	1.171 8	<i>0.678 9</i>	1.051 4	1.045 7
DTLZ4	5 000	0.056 8	0.055 9	0.052 9	<i>0.053 0</i>	0.053 5
DTLZ5	5 000	<i>0.004 6</i>	0.005 1	0.004 5	0.004 9	0.005 0
DTLZ6	15 000	0.159 3	0.785 1	<i>0.236 6</i>	0.238 4	0.384 1
DTLZ24	20 000	0.119 6	<i>0.118 7</i>	0.118 5	0.120 1	0.119 2

通过测试获取了 P_{local} 的不同取值下每一个测试问题IGD的中间值. 对于每一个测试问题,最好的IGD值和第2好的IGD取值分别以加粗和斜体表示. 由表1可看出,对于不同 P_{local} 取值,只有DTLZ3和DTLZ6测试问题的IGD值变化较大,这表明HN算法对于 P_{local} 取值不是很敏感. DTLZ1和DTLZ3测试问题拥有多个局部最优Pareto前沿,较小的 P_{local} 取值有利于HN算法向最优Pareto前沿收敛. 较大的 P_{local} 取值易产生上文描述的超级个体而使种群丧失多样性,较小的 P_{local} 取值又使得算法加速收敛能力不足. 为了平衡 P_{local} 对种群多样性及算法收敛性的影响,本文选择 $P_{\text{local}} = 1/(3 \cdot N)$.

- 2) 混合多目标进化算法框架有效性验证.

针对每一个测试问题,NSGA-II算法和HNSGA-II算法采用相同的进化算子,个体均采用实数编码,对于不同测试问题分别采用两种不同的函数评价次数,运行结果如表2所示.

由表2可以看出,针对所有测试问题,HNSGA-II算法的IGD值都低于NSGA-II算法,所求得的PF更接近于理想的PF,且所求得的非支配解具有良好

表2 HNSGA-II和NSGA-II对8个测试问题的测试结果

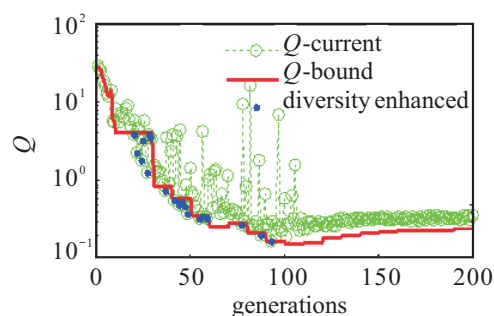
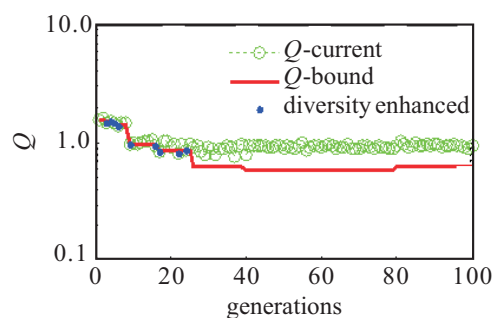
test problems	FE	NSGA-II			HNSGA-II		
		best	median	worst	best	median	worst
ZDT4(2-obj)	20 000	0.006 2	0.014 1	0.100 1	0.004 5	0.005 2	0.014 5
	21 000	0.005 4	0.010 9	0.068 1	0.004 4	0.005 0	0.008 6
DTLZ1(3-obj)	20 000	0.462 5	1.315 5	3.003 7	0.082 3	0.365 8	1.328 0
	30 000	0.048 3	0.276 2	0.812 1	0.018 1	0.054 2	0.782 3
DTLZ2(3-obj)	5 000	0.053 4	0.061 1	0.067 9	0.047 8	0.055 1	0.064 3
	10 000	0.047 8	0.050 1	0.052 6	0.047 1	0.049 6	0.053 5
DTLZ3(3-obj)	30 000	1.114 6	3.042 8	5.910 7	0.058 5	0.678 9	3.712 6
	40 000	0.175 6	0.834 3	2.756 2	0.050 8	0.058 7	1.011 2
DTLZ4(3-obj)	5 000	0.056 5	0.065 1	0.082 7	0.049 6	0.052 9	0.067 8
	10 000	0.046 1	0.049 8	0.051 9	0.045 7	0.048 9	0.050 6
DTLZ5(3-obj)	5 000	0.010 1	0.012 9	0.015 8	0.002 6	0.004 5	0.008 5
	10 000	0.003 1	0.003 9	0.004 4	0.002 7	0.002 8	0.003 3
DTLZ6(3-obj)	15 000	1.880 2	2.005 1	2.217 3	0.003 2	0.236 6	2.101 2
	20 000	1.242 5	1.333 5	1.512 6	0.002 8	0.055 8	0.467 1
DTLZ24(4-obj)	20 000	0.113 4	0.196 8	0.131 8	0.114 5	0.118 5	0.125 6
	30 000	0.114 6	0.117 1	0.123 4	0.114 3	0.116 9	0.123 8

的分布性,说明HN算法优于NSGA-II算法。在多峰的ZDT4、DTLZ1和DTLZ3测试问题中,HN算法获得的IGD值也明显低于NSGA-II算法,这充分表明了基于所提混合框架的HN算法在处理多峰问题时的有效性。由表2可以看出,即使经过较多的函数评价次数,NSGA-II算法的IGD值仍高于HNSGA-II算法,HNSGA-II算法经过较少的函数评价次数FE1就能获得优于FE2下NSGA-II算法的收敛性和分布性,HNSGA-II算法较快的收敛性归功于局部搜索模块的作用。上述结果表明,基于所提出多目标进化算法框架的HNSGA-II算法在保证种群多样性的前提下可以快速逼近理想PF。

3) 投影聚类模块。

在所提混合多目标优化算法框架中,投影聚类模块采用聚类质量指标判别种群多样性的好坏。图5和图6给出了测试函数ZDT4和DTLZ2的聚类质量指标进化过程。ZDT4拥有多个局部最优前沿,DTLZ2则没有,这两个测试函数分别代表了两种特征的优化问题。第 t 代的聚类质量指标 Q -current用圆圈表示,质量指标下限阈值 Q -bound用加粗实线表示,实心圆点表示种群多样性模块被激活。

由图5和图6可以看出,初始阶段 Q -current有较高的取值,表示初始种群具有较好的多样性,随着

图5 ZDT4聚类质量指标 Q 进化曲线图6 DTLZ2聚类质量指标 Q 进化曲线

种群逐渐逼近PF, Q -current逐渐减小并稳定。图5中ZDT4的 Q -current进化曲线类似“之”字型,中间有很多峰值突起,这归因于种群增强模块的多样性增强作用。ZDT4为多峰函数,具有多个局部最优,局部搜索和局部最优前沿的“欺诈”均可导致种群多样性丧失和最优解的丢失。在混合算法中,多样性增强模块通过多样性增强策略增加种群多样性。研究表明,HN算法能有效防止种群多样性的丢失。由图6可以看出,与ZDT4不同,DTLZ2不具有多个Pareto局部最优前沿,所以其 Q -current进化曲线与ZDT4不同,在大多数进化过程中,HN算法都保持良好的种群多样性:在前25代,在实心圆点处, Q -current低于 Q -bound,种群多样性增强模块多次被激活以增加种群多样性;在25代以后, Q -bound低于 Q -current,多样性增强模块在进化25代以后没有被激活。

4) 局部搜索模块作用。

为理解局部搜索模块的作用,本文提出一个对比算法:将局部搜索策略作用于以概率 P_{local} 随机从种群中选取的个体上代替作用于分别从不同聚类中以概率 P_{local} 选择的个体上,称之为HN-r。3种算法采用相同的参数设置,测试结果如表3所示。

针对上述测试问题,由表3可以看出,HN算法均优于HN-r和NSGA-II算法,在DTLZ1、DTLZ3和DTLZ6测试函数上表现尤为明显,这说明从不同的聚类中选择个体进行局部搜索能有效加快算法收敛。对于上述测试函数,种群能逼近理想PF归功于混合算法中的投影聚类模块和局部搜索模块的作用。

表3 局部搜索模块作用

test problem	FE	NSGA-II	HN-r	HN
ZDT4(2-obj)	20 000	0.014 1	0.006 9	0.005 1
DTLZ1(3-obj)	20 000	1.315 5	0.475 1	0.373 5
DTLZ2(3-obj)	5 000	0.061 1	0.056 6	0.055 1
DTLZ3(3-obj)	40 000	0.834 3	0.294 4	0.058 7
DTLZ4(3-obj)	5 000	0.065 1	0.055 7	0.052 9
DTLZ5(3-obj)	5 000	0.012 9	0.008 3	0.004 5
DTLZ6(3-obj)	20 000	1.333 5	0.103 7	0.055 8
DTLZ24(4-obj)	20 000	0.196 8	0.119 8	0.118 5

5) 种群多样性增强模块作用.

本节将研究多样性增强模块在维护算法多样性方面的作用. 采用HN算法, 选用两个优化目标的ZDT4测试函数, 分别在添加多样性增强模块与去掉该模块两种情形下进行测试, 图7~图10分别为两种情形下第1、5、8、10代的种群多样性分布图.

初始种群如图7所示, 种群集中在 f_1 取值较小的左侧区域, 因此, 在以后迭代中目标空间中产生的解有聚集在该区域的趋势. 由图8和图9中可以看出: 在

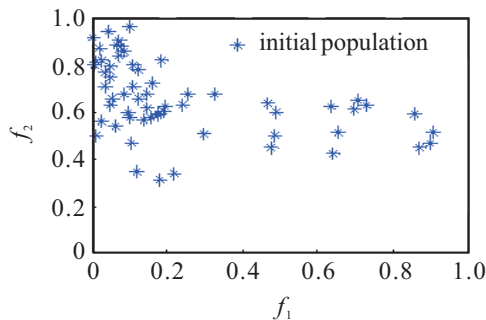


图7 ZDT4第1代种群分布

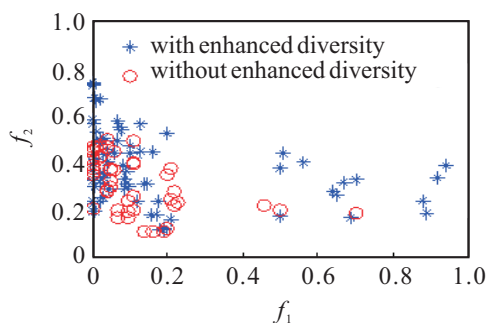


图8 ZDT4第5代种群分布

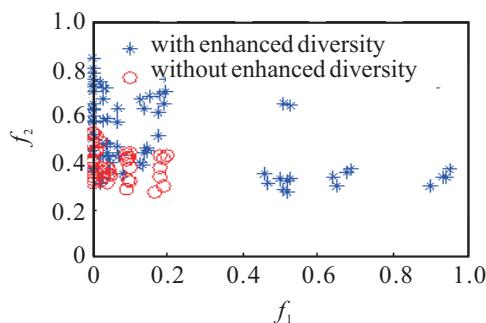


图9 ZDT4第8代种群分布

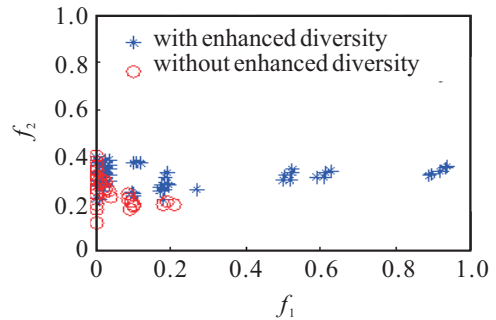


图10 ZDT4第10代种群分布

没有多样性增强模块的情况下, 种群在左侧区域分布较密集, 种群多样性较低, 这是因为在该区域操作因子更容易产生同一区域的解以及局部搜索模块的作用; 在有多多样性增强模块的情况下, 种群比较均匀地分布在目标空间内, 具有较好的种群多样性和分布性. 由图9也可以看出, 进化第8代在种群多样性增强模块作用下HN算法保持了较好的多样性, 但是同时也降低了算法收敛至PF的速度, 因此, 在进化第10代在种群多样性增强模块作用的同时, HN算法加快了算法收敛速度. 至此, 以上结果演示了种群多样性增强模块在所提混合框架中的重要作用.

4.3 基于MOEA/D的算法框架验证

为验证所提混合框架同样适用于其他多目标进化算法, 本文采用MOEA/D算法作为混合框架中的多目标进化算法模块算法, 对比算法为HMOEA/D. 选用多目标优化问题CEC09测试集UF1-UF13来检测所提HMOEA/D算法的性能^[22], 在所选的13个测试问题中, UF1-UF7为2个目标的优化问题, UF8-UF10为3个目标的优化问题, UF11-UF13为5个目标的优化问题, 因CEC09测试集包含可微与非可微多目标优化问题, 故HMOEA/D算法中局部搜索模块采用文献[23]中的简化二次逼近策略, 算法其他模块采用与HNSGA-II相同策略. 算法参数设置如下.

- 1) 群体规模: UF1-UF7(2-obj) 设为 600; UF8-UF10(3-obj) 设为 1000; UF11-UF13(5-obj) 设为 1500.
- 2) 向量邻居个数: $T = 0.01N$, $\eta_r = 0.01N$.
- 3) 选择交叉/更新区域概率 $\delta = 0.9$.
- 4) DE操作算子中: DE交叉概率 $CR = 1.0$; DE比例系数 $F = 0.5$; 变异分布指数 $\eta_m = 20$; 变异概率 $p_m = 1/n$.
- 5) 在局部搜索中, $\eta = 10$, $\varepsilon = 10^{-6}$.
- 6) 停止准则: 当函数评价次数达到 100 000 时算法终止.
- 7) 算法对每个测试问题均独立运行 30 次.

另外, 聚类算法采用基于距离的聚类方法, 聚类数为6. 用于性能评价的Pareto解的个数为: UF1-

UF7(2-obj) 设为 100; UF8-UF10(3-obj) 设为 150; UF11-UF13(5-obj) 设为 400. 算法性能评价采用 IGD 指标, 最好的指标值用加粗表示, 统计结果如表 4 所示. 由表 4 可以看出: 对于 CEC09 测试集, HMOEA/D 算法求得的近似 PF 的多样性和收敛性优于 MOEA/D 算法; 对于测试问题 UF7, MOEA/D 算法虽优于 HMOEA/D 算法, 但两者相差不大.

HMOEA/D 算法通过局部搜索模块加速算法收敛, 利用多样性增强模块增加种群多样性, 理论上会

增加算法求解测试问题所需的运行时间. 为此, 本文对两种算法求解效率进行比较. 在相同的进化代数下, 表 5 给出了两种算法求解测试问题所需的运行时间(单位 s). 由表 5 可以看出, HMOEA/D 算法的运行时间均值大于 MOEA/D 算法, 这是因为当种群经投影聚类模块后, 若种群具有良好的种群多样性, 则进行一次局部搜索, 否则, 进行种群多样性增强. 这虽然会增加算法的运行时间, 但同时局部搜索模块加速了算法收敛, 所以两种算法计算效率相差不大.

表 4 HMOEA/D 和 MOEA/D 对 UF1-UF13 测试问题的测试结果

problems	HMOEA/D				MOEA/D				SS
	best	worst	mean	std	best	worst	mean	std	
UF1	0.004 16	0.004 62	0.004 31	0.000 09	0.003 93	0.005 17	0.004 37	0.000 29	+
UF2	0.004 45	0.010 17	0.005 65	0.001 33	0.004 78	0.010 83	0.006 81	0.001 82	+
UF3	0.004 02	0.007 82	0.004 69	0.000 96	0.003 92	0.024 36	0.007 43	0.005 18	+
UF4	0.040 51	0.047 18	0.043 87	0.001 72	0.056 86	0.081 37	0.063 85	0.005 32	+
UF5	0.043 15	0.553 22	0.145 06	0.121 44	0.082 27	0.306 19	0.180 74	0.068 11	+
UF6	0.002 12	0.004 87	0.003 17	0.000 78	0.003 51	0.010 04	0.005 89	0.001 77	+
UF7	0.004 09	0.315 71	0.014 5	0.056 82	0.003 25	0.010 65	0.004 47	0.001 18	-
UF8	0.048 45	0.086 42	0.057 62	0.008 57	0.050 72	0.065 61	0.058 43	0.003 31	+
UF9	0.030 37	0.145 83	0.085 26	0.057 02	0.035 06	0.149 78	0.088 97	0.058 13	+
UF10	0.107 37	0.543 11	0.289 04	0.109 03	0.364 06	0.649 58	0.474 16	0.007 35	+
UF11	0.100 92	0.187 89	0.106 43	0.015 47	0.106 90	0.115 16	0.110 24	0.002 29	+
UF12	73.176 43	206.055 8	119.715 5	33.596 6	66.168 3	214.226 4	145.788 7	41.827 6	+
UF13	1.184 17	1.855 05	1.837 28	0.003 81	1.834 66	1.899 22	1.856 65	0.019 78	+

表 5 两种算法 30 次独立运行所需时间

problems	iteration	HMOEA/D				MOEA/D			
		max	min	mean	std	max	min	mean	std
UF1	2 500	100.058	93.864	95.443 9	0.865 481	92.128	84.276	89.516 2	1.102 541
UF2	2 500	110.358	95.036 2	98.835 8	1.1077 25	91.526	87.252	89.057 6	0.582 751
UF3	2 500	108.102 4	90.235 2	93.021 6	1.8452 62	88.317	81.528	85.604 5	1.194 172
UF4	2 500	126.258	95.305 6	101.361 2	1.5242 36	90.762	86.586	88.725 4	0.834 201
UF5	2 500	82.241	72.167 1	76.782 3	2.416 434	74.386	65.525	71.826 6	1.8363 12
UF6	2 500	104.145	91.328	96.641 7	1.756 276	84.426	78.153	81.538	1.294 225
UF7	2 500	120.538	90.366	98.389 1	1.935 218	89.012	86.158	88.089	0.474 431
UF8	1 500	362.578	351.354 1	355.882 3	2.487 524	309.551	304.121	306.515 8	1.142 627
UF9	1 500	410.328	346.128	358.423 6	12.635 64	383.712	311.442	336.872 2	22.092 16
UF10	1 500	374.455 6	310.435	325.311 9	12.852 12	341.656	269.558	296.316 8	23.245 01
UF11	1 000	1247.258	1029.135	1079.263	51.678 21	1 446.01	1 000.26	1 078.586	120.265 7
UF12	1 000	918.352 2	809.506	841.627 8	22.088 16	829.018	734.162	778.546 3	26.312 3
UF13	1 000	1 372.643	1 013.138	1 088.065	90.462 86	1 271.25	1 029.38	1 058.429	42.658 64

5 结 论

本文提出了一种模块化的基于种群的全局搜索与局部搜索相结合的多目标进化算法混合框架, 该混合框架包含进化算法模块、投影聚类模块、局部搜索模块和种群多样性增强模块. 进化算法模块可以采用现有多目标进化算法及其改进算法; 投影聚类

模块维持和增加进化群体的多样性和分布性, 聚类质量指标作为种群多样性的好坏指标来进行局部搜索或种群多样性增强操作; 局部搜索模块在种群具有较好多样性时加快算法收敛; 多样性增强模块在保证收敛的同时增加种群多样性以保证解的均匀性和分布性. 以经典多目标优化算法 NSGA-II 算法和

MOEA/D 算法为例,以所提框架为基础,设计了各模块采用的策略,并采用数值实验方法详细研究了所提框架各模块的作用,验证了所提框架的有效性.结果表明,该框架具有模块化通用性特点,能提升多目标进化算法的性能.

提出更有效的局部搜索策略和多样性增强策略,以及种群多样性指示指标,研究局部搜索概率的自适应取值方法是未来的研究工作.

参考文献(References)

- [1] Coello C A, Lamont G B. Applications of multi-objective evolutionary algorithms[M]. Singapore: World Scientific, 2004: 2-16.
- [2] Medaglia A L, Villegas J G, Rodriguez-Coca D M. Hybrid bi-objective evolutionary algorithms for the design of a hospital waste management network[J]. *Heuristics*, 2009, 15(2): 153-176.
- [3] Tapia C, Coello C A. Applications of multi-objective evolutionary algorithms in economics and finance: A survey[C]. *Proc of CEC*. Singapore, 2007: 532-539.
- [4] Qi Y T, Hou Z T, Li H H, et al. A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows[J]. *Computers & Operations Research*, 2015, 62: 61-67.
- [5] Lin Q Z, Zhu Q L, Huang P Z, et al. A novel hybrid multi-objective immune algorithm with adaptive differential evolution[J]. *Computers & Operations Research*, 2015, 62: 95-111.
- [6] Liu C A, Jia H M. Dynamic multi-objective evolutionary algorithm with two stages evolution operation[J]. *Intelligent Automation and Soft Computing*, 2015, 4(21): 575-588.
- [7] Aimin Z, Jin Y C, Zhang Q F. A population prediction strategy for evolutionary dynamic multi-objective optimization[J]. *IEEE Trans on Cybernetics*, 2014, 44(1): 40-53.
- [8] Deb K, Lele S, Datta R. A hybrid evolutionary multi-objective and SQP based procedure for constrained optimization[C]. *Advances in Computation and intelligence*. Wuhan: Springer, 2007: 36-45.
- [9] Rafiul Hassan M, Nath B, Kirley M, et al. A hybrid of multi-objective evolutionary algorithm and HMM-fuzzy model for time series prediction[J]. *Neuro Computing*, 2012, 81: 1-11.
- [10] Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling[J]. *IEEE Trans on Systems, Man, and Cybernetics: Part C*, 1998, 28(3): 392-403.
- [11] Goel T, Deb K. Hybrid methods for multi-objective evolutionary algorithms[C]. *Proc of the 4th Asia-Pacific Conf on Simulated Evolution and Learning(SEAL 2002)*. Singapore, 2002: 188-192.
- [12] Sindhya K, Sinha A, Deb K, et al. Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems[C]. *Proc of Congress on Evolutionary Computation*. Norway: IEEE, 2009: 2919-2926.
- [13] Sindhya K, Deb K, Miettinen K. A local search based evolutionary multi-objective optimization approach for fast and accurate convergence[C]. *Parallel Problem Solving from Nature(PPSN X)*. Berlin: Springer, 2008: 815-824.
- [14] 贾树晋, 杜斌, 岳恒. 基于局部搜索与混合多样性策略的多目标粒子群算法[J]. *控制与决策*, 2012, 27(6): 813-826.
(Jia S J, Du B, Yue H. Local search and hybrid diversity strategy based multi-objective particle swarm optimization algorithm[J]. *Control and Decision*, 2012, 27(6): 813-826.)
- [15] Wierzbicki A P. The use of reference objectives in multi-objective optimization[C]. *Multiple Criteria Decision Making Theory and Application*. Berlin: Springer, 1980: 468-486.
- [16] Miettinen K. *Nonlinear multi-objective optimization*[M]. Boston: Kluwer Academic Publishers, 1999: 19-21.
- [17] Luque M, Miettinen K, Eskelinen P, et al. Incorporating preference information in interactive reference point methods for multi-objective optimization[J]. *Omega*, 2009, 37(2): 450-462.
- [18] Luque M, Miettinen K, Ruiz A B, et al. A two-slope achievement scalarizing function for interactive multiobjective optimization[J]. *Computers and Operations Research*, 2012, 39: 1673-1681.
- [19] 董宁. 求解约束优化和多目标优化问题的进化算法研究[D]. 西安电子科技大学数学与统计学院, 2014: 90-93.
(Dong N. Study on evolutionary algorithms for constrained optimization and multi-objective optimization problems[D]. Xi'an: School of Mathematics and Statistics, Xidian University, 2014: 90-93.)
- [20] Wilson R B. *A simplicial method for canver programming*[D]. Cambridge: Havverd University, 1963: 37-46.
- [21] Byrd R H, Nocedal J, Waltz R A. *Large-scale nonlinear optimization*[M]. Berlin/Heidelberg: Springer, 2006: 126-131.
- [22] Zhang Q F, Zhou A, Zhao S, et al. Multi-objective optimization test instances for the CEC 2009 special session and competition[R]. UK: School of Computer Science and Electronic Engineering, University of Essex, 2008.
- [23] 谭艳艳. 几种改进的分解类多目标进化算法及其应用[D]. 西安电子科技大学电子工程学院, 2013: 64-65.
(Tan Y Y. Several modified decomposition-based multiobjective evolutionary algorithms and their applications[D]. Xi'an: School of Electronic Engineering, Xidian University, 2013: 64-65.)

(责任编辑: 李君玲)