

基于 Pareto 熵的多目标粒子群优化算法^{*}

胡旺^{1,2}, Gary G. YEN², 张鑫¹

¹(电子科技大学 信息与软件工程学院, 四川 成都 610054)

²(School of Electrical and Computer Engineering, Oklahoma State University, USA)

通讯作者: 胡旺, E-mail: scuhwang@126.com

摘要: 粒子群优化算法因形式简洁、收敛快速和参数调节机制灵活等优点,同时一次运行可得到多个解,且能逼近非凸或不连续的 Pareto 最优前端,因而被认为是求解多目标优化问题最具潜力的方法之一.但当粒子群优化算法从单目标问题扩展到多目标问题时, Pareto 最优解集的存储与维护、全局和个体最优解的选择以及开发与开采的平衡等问题亦随之出现.通过目标空间变换方法,采用 Pareto 前端在被称为平行格坐标系统的新目标空间中的分布熵及差熵评估种群的多样性及进化状态,并以此为反馈信息来设计进化策略,使得算法能够兼顾近似 Pareto 前端的收敛性和多样性.同时,引入格占优和格距离密度的概念来评估 Pareto 最优解的个体环境适应度,以此建立外部档案更新方法和全局最优解选择机制,最终形成了基于 Pareto 熵的多目标粒子群优化算法.实验结果表明:在 IGD 性能指标上,与另外 8 种对等算法相比,该算法在由 ZDT 和 DTLZ 系列组成的 12 个多目标测试问题集中表现出了显著的性能优势.

关键词: 多目标优化问题;粒子群优化;平行格坐标系统;Pareto 熵;自适应参数

中图法分类号: TP301

中文引用格式: 胡旺, Yen GG, 张鑫. 基于 Pareto 熵的多目标粒子群优化算法. 软件学报, 2014, 25(5): 1025–1050. <http://www.jos.org.cn/1000-9825/4496.htm>

英文引用格式: Hu W, Yen GG, Zhang X. Multiobjective particle swarm optimization based on Pareto entropy. Ruan Jian Xue Bao/Journal of Software, 2014, 25(5): 1025–1050 (in Chinese). <http://www.jos.org.cn/1000-9825/4496.htm>

Multiobjective Particle Swarm Optimization Based on Pareto Entropy

HU Wang^{1,2}, Gary G. YEN², ZHANG Xin¹

¹(School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

²(Oklahoma State University, School of Electrical and Computer Engineering, USA)

Corresponding author: HU Wang, E-mail: scuhwang@126.com

Abstract: Due to its concise formation, fast convergence, and flexible parameters, particle swarm optimization (PSO) with the ability to gain multiple solutions at a run and to approximate the Pareto front of those non-convex or discontinuous multiobjective optimization problems (MOPs) is considered to be one of the most promising techniques for MOPs. However, several challenges, such as maintaining the archive, selecting the global and personal best solutions, and balancing the exploration and exploitation, occur when extending PSO from single-objective optimization problems to MOPs. In this paper, the distribution entropy and its difference of an approximate Pareto front in a new objective space, named parallel cell coordinate system (PCCS), are proposed to assess the diversity and evolutionary status of the population. The feedback information from evolutionary environment is served in the evolutionary strategies to balance the convergence and diversity of an approximate Pareto front. Meanwhile, the new concepts, such as cell dominance and individual density based on cell distance in the PCCS, are introduced to evaluate the individual environmental fitness which is the metric using in updating the archive and selecting the global best solutions. The experimental results illustrate that the proposed algorithm in this paper

* 基金项目: 中央高校基本科研业务费专项资金(2672013ZYGX2013J078)

收稿时间: 2013-04-19; 修改时间: 2013-06-18; 定稿时间: 2013-09-09

significantly outperforms the other eight peer competitors in terms of IGD on 12 test instances chosen from the ZDT and DTLZ test suites.

Key words: multiobjective optimization problem (MOP); particle swarm optimization (PSO); parallel cell coordinate system (PCCS); Pareto entropy; adaptive parameter

大多数工程和科学问题都可以归结为多目标优化问题(multiobjective optimization problem,简称 MOP),其求解方法一直都是学术界和工程界共同关注的焦点.多目标优化问题通常存在多个彼此冲突的目标,其优化结果为 Pareto 最优解集.与数学规划方法相比,进化算法因一次运行可得到多个解,且能逼近非凸或不连续的 Pareto 最优前端,从而被认为是更适合求解多目标优化问题的智能方法.

粒子群优化算法(particle swarm optimization,简称 PSO)^[1]是由 Kennedy 和 Eberhart 在 1995 年提出的仿生算法,它是受到飞鸟集群活动规律的启发,根据社会学和心理学而建立的群体智能模型.粒子群优化算法是一种具有形式简洁、收敛快速和参数调节机制灵活等优点的进化算法,并且已经成功应用于单目标优化问题,被认为是求解多目标优化问题最具潜力的方法之一.但当粒子群优化算法从单目标问题扩展到多目标问题而形成多目标粒子群优化(multiple objective particle swarm optimization,简称 MOPSO)时,新的技术问题随之出现.多目标优化问题的 Pareto 最优解集特征和粒子群优化算法的快速收敛特征,使得多目标粒子群优化算法面临以下技术挑战:① 维护存储 Pareto 最优解集的档案,使算法最终输出的 Pareto 前端能够兼顾收敛性和多样性,这是第二代进化多目标算法的共性问题;② 从群体和个体的 Pareto 最优解集中分别选举全局最优解(gBest)和个体最优解(pBest),使这两个最优解引导粒子尽最大可能地发现高质量的新解;③ 平衡进化过程中的开采(exploration)和开发(exploitation),使得种群既保持快速收敛的优点,但又尽可能地避免陷入局部 Pareto 前端或者收敛到单一解.后面两个挑战是多目标粒子群优化算法区别于其他多目标进化算法的特有问题.

本文将根据多目标优化问题和粒子群优化算法的特征,采用近似 Pareto 前端的分布熵及其变化来估计种群进化状态,并依据这些进化过程反馈信息设计具有动态平衡开采和开发能力的进化策略,形成基于 Pareto 熵的多目标粒子群优化算法.

为了内容相对独立以及统一本文术语和符号,本文第 1 节简要介绍多目标优化问题和粒子群优化算法的基本概念.第 2 节综述与本文内容相关的多目标粒子群优化算法的研究现状和存在的问题.第 3 节阐述 Pareto 熵的计算方法以及基于 Pareto 熵的种群进化状态检测方法.第 4 节描述本文算法的主要进化策略和完整的算法流程.第 5 节分析对比实验结果.最后,在第 6 节中给出全文结论.

1 预备知识

1.1 多目标优化问题

对于最小化无约束连续多目标优化问题(在优化计算领域,最大化问题与最小化问题是一组对偶问题.在无特别说明的情况下,本文所指最优优化均为最小化问题),可以定义如下:

$$\min F(\mathbf{x})=[f_1(\mathbf{x}),f_2(\mathbf{x}),\dots,f_M(\mathbf{x})]^T \text{ for all } \mathbf{x} \in \mathcal{R}^D,$$

其中, $\mathbf{x}=[x_1,x_2,\dots,x_D]^T \in \mathcal{S}, D$ 为决策变量个数, \mathcal{S} 是 D 维决策空间.一组目标函数 f_1,f_2,\dots,f_M 将决策空间映射到目标空间, $\mathcal{S} \rightarrow \Omega \in \mathcal{R}^M, M$ 为目标个数, Ω 是 M 维目标空间.

下面给出与本文使用了 Pareto 最优的相关定义.

定义 1(Pareto 占优). 对于任意两个向量 $\mathbf{u}, \mathbf{v} \in \Omega$, 称 \mathbf{u} 占优 \mathbf{v} , 或称 \mathbf{v} 被 \mathbf{u} 占优, 记作 $\mathbf{u} > \mathbf{v}$, 当且仅当

$$\forall i=1,2,\dots,m, u_i \leq v_i \wedge \exists j=1,2,\dots,m, u_j < v_j.$$

定义 2(Pareto 最优解). 一个解 $\mathbf{x}^* \in \Omega$ 被称为 Pareto 最优解或非占优解, 当且仅当

$$\neg \exists \mathbf{x} \in \Omega: \mathbf{x} > \mathbf{x}^*.$$

定义 3(Pareto 最优解集和 Pareto 前端). 所有 Pareto 最优解的集合 $PS=\{\mathbf{x}^* | \neg \exists \mathbf{x} \in \Omega: \mathbf{x} > \mathbf{x}^*\}$ 称为 Pareto 最优解集.所有 Pareto 最优解对应目标函数值所形成的区域 $PF=\{F(\mathbf{x}^*) | \mathbf{x}^* \in PS\}$ 称为 Pareto 前端, 或 Pareto 均衡面.

多目标优化算法的目标是:生成数量尽可能多的 Pareto 最优解集,同时,尽可能地使对应的 Pareto 前端分布多样化,包括均匀性(uniformity)和延展性(extensiveness).

1.2 粒子群优化算法

在粒子群优化算法中,一个无质量的粒子 i 可以由位置向量 \mathbf{x}_i 和速度向量 \mathbf{v}_i 表示,其中, $\mathbf{x}_i=[x_{i,1},x_{i,2},\dots,x_{i,D}]^T \in \mathcal{R}^D$, $\mathbf{v}_i=[v_{i,1},v_{i,2},\dots,v_{i,D}]^T \in \mathcal{R}^D$, $i=1,2,\dots,N$, N 是种群中的粒子个数, D 是决策变量个数.种群中的每一个粒子,在进化过程中根据公式(1)更新速度和位置:

$$\begin{cases} \mathbf{v}_i(t+1) = \omega \mathbf{v}_i(t) + c_1 r_1 (\mathbf{pBest}_i - \mathbf{x}_i(t)) + c_2 r_2 (\mathbf{gBest} - \mathbf{x}_i(t)) \\ \mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \end{cases} \quad (1)$$

其中, t 表示迭代次数, $\omega \geq 0$ 表示惯性权重系数, $c_1, c_2 \geq 0$ 表示加速系数, r_1 和 r_2 是在 $[0,1]$ 上均匀分布的随机数, \mathbf{pBest}_i 表示第 i 个粒子的个体最优解, \mathbf{gBest} 表示整个群体的全局最优解.

在公式(1)的速度更新方程中,第 1 部分为粒子当前速度乘以惯性权重进行加速,表示粒子对当前自身运动的信任,依据自身的速度进行惯性运动;第 2 部分为自我认知部分,表示粒子对自身历史的思考;第 3 部分为社会认知部分,表示粒子在群体中的信息共享和相互合作.

2 相关研究工作

自 Coello 等人^[2]于 2002 年提出采用粒子群算法来求解多目标优化问题以来,多目标进化领域中的成功经验被迅速借鉴到多目标粒子群优化算法中,出现了大量的阶段性研究成果,其中以 Coello 等人^[3]采用超网格和变异方法的多目标粒子群优化算法最为经典.Reyes-Sierra^[4]对 2006 年之前的多目标粒子群优化算法做了综述总结,随后,Padhye^[5,6]做了一些专题比较研究.关于多目标进化算法的最新综述可参见文献[7,8].

对多目标粒子群优化算法的研究主要集中在档案维护、全局最优解选择和种群多样性增加等方面,也有少部分工作对个体最优解选择和粒子变异方面进行了研究.

在外部档案维护策略方面,主要借鉴多目标遗传算法中档案维护的有效策略,如采用聚类、拥挤距离(crowding distance)、自适应超网格(adaptive grid)、 \square -占优(\square -dominance)、最大化最小适应度(maximin fitness)、 σ 值等形成相应的多目标粒子群优化算法中的档案维护策略^[9,10].这些不同的策略都是直接或间接地评估档案中每个个体的密度(拥挤度)或者占优强度.当档案中的 Pareto 最优解个数超出容量限制时,通常直接将个体适应度较好的新解来更新个体适应度较差的旧解,从而使得 Pareto 前端具有更好的分布性能.

在全局最优解选择策略方面,基于随机选择、 σ 值、拥挤距离、动态邻居、自适应超网格、聚类、占优树、非占优排序(non-dominated sorting)、Pareto 占优、超体积(hyper-volume)、最大化最小适应度、小生境(niche)、综合学习(comprehensive learning)等策略多目标粒子群优化算法被相继提出来^[9-12].在这些方法中,大多数通过评估外部档案中的个体密度来选择那些处于相对稀疏区域的 Pareto 最优解作为全局最优解以降低选择压力,避免收敛到单一解,以及增加 Pareto 前端的多样性;而另一部分基于“占优”概念的最优解选择策略则偏重算法的收敛性能,但容易导致过高的选择压力而容易收敛到局部极值.

在个体最优解选择策略方面,虽然绝大部分算法为了降低计算开销而仅保留一个 Pareto 最优解,但仅一个个体最优解不足以描述个体 Pareto 最优前端.Branke^[13]和 Abido^[14]分别研究了采用个体外部档案保存个体 Pareto 最优解的方法(类似种群外部档案保存种群 Pareto 最优解),获得了比单一个体 Pareto 最优解更好的性能.另外,最大化最小个体间距离和加权和^[13]等策略被用来作为选择个体最优解的标准.

在平衡开发与开采策略方面,如果不合理利用粒子群算法快速收敛的优点,则容易导致早熟而陷入局部极值.一些算法从种群角度提出平衡策略,如动态调整种群大小策略^[15]和子群策略^[16-18];一些算法采用文化进化框架提出平衡策略^[19,20];一些算法从粒子运动参数选择角度提出平衡策略,如线性下降惯性权重参数和自适应参数等^[21];一些算法从粒子扰动(变异)角度提出平衡策略,自 Coello^[3]在多目标粒子群算法中提出变异方法后,后续多目标粒子群优化算法几乎都采用变异方法来增强种群多样性;还有一些算法采用混合平衡策略^[22,23].

但是,多目标粒子群优化算法仍然面临一些技术挑战:

- 一方面,个体适应度评估决定了多目标粒子群优化算法中选择全局最优解和维护档案这两个关键策略.但当前的个体适应度都是单一度量标准,要么是基于密度的评估来考察个体的多样性潜力,要么是基于 Pareto 占优关系的评估来考察个体的收敛性潜力.因而导致在维护外部档案时不能兼顾 Pareto 最优前端的多样性和收敛性,而在选择全局最优解时不能兼顾平衡开发和开采.
- 另一方面,全局最优解频繁更换和快速收敛特征,使得开发与开采的平衡问题在多目标粒子群优化算法中更为突出:过度地开发将会导致收敛性不足而影响优化精度,而过度地开采将会导致多样性匮乏而陷入局部极值.虽然已经存在自适应单目标粒子群优化算法^[24],但多目标问题的 Pareto 最优解集特征,使得在多目标空间中监测种群进化环境更为复杂.而当前已经存在的多目标粒子群优化算法缺乏种群进化环境的动态监测机制来获取反馈信息,难以决定算法“在何时调节何种进化策略到何种程度”.

熵在热力学中表示系统混乱状态,而在生态学中表示生物的多样性.本文将通过对目标空间变换来获得 Pareto 前端的熵,以此来度量多目标粒子群优化算法中种群的多样性,并利用差熵来估计种群的进化状态,从而为算法提供实时的进化环境反馈信息.同时,在变换后的目标空间中评估 Pareto 最优解的个体密度和格占优强度,为算法中外部档案更新和全局最优解选择提供决策依据.

3 Pareto 熵及进化状态检测

本节先介绍 Pareto 熵的计算方法,包括将采用平行格坐标系统映射目标空间映射的方法和 Pareto 熵及差熵的定义;然后,根据外部档案更新算法中 Pareto 熵的变化情况,分析不同进化状态的临界阈值;最后,归纳出种群进化状态的判定条件.

3.1 平行格坐标系统

平行坐标(parallel coordinates)^[25]是一种广泛使用的多维数据分析和可视化方法.多目标优化问题的目标向量正是一种多维数据(在进化多目标优化计算社区中,通常将 2~3 个目标的优化问题称为 multi-objective optimization problem,简称 MOP,而将 4 个以上目标的优化问题称为 many-objective optimization problem,简称 MaOP).受这一方法启发,将存储在外部档案中的多维 Pareto 前端按照平行坐标方式转化到二维平面中,并将 Pareto 前端的笛卡尔坐标值映射成整数值坐标,从而多维 Pareto 前端被转换成一张二维平面网格.对应于笛卡尔坐标系统,在此将这种二维平面网格称为平行格坐标系统(parallel cell coordinates system,简称 PCCS).下面采用数学形式来描述这种映射方法.

对于 Pareto 最优解集中第 k 个非占优解对应的第 m 个目标值 $f_{k,m}$ 可以按照公式(2)映射到一个具有 $K \times M$ 个格子(cell)的二维平面网格中:

$$L_{k,m} = \left\lceil K \frac{f_{k,m} - f_m^{\min}}{f_m^{\max} - f_m^{\min}} \right\rceil \quad (2)$$

其中, $\lceil x \rceil$ 是向上取整函数,返回不小于 x 的最小整数; $k=1,2,\dots,K$, K 为外部档案在当前迭代中的成员个数; $m=1,2,\dots,M$, M 为待优化问题的目标个数; $f_m^{\max} = \max_k f_{k,m}$ 和 $f_m^{\min} = \min_k f_{k,m}$ 分别是当前 Pareto 前端上第 m 个目标的最大值和最小值; $L_{k,m} \in \{1,2,\dots,K\}$ 是 $f_{k,m}$ 被映射到 PCCS 中的整数标号,表示第 k 个非占优解的第 m 个格坐标分量.在特殊情况下,当 $f_{k,m} = f_m^{\min}$ 时,为了避免 $L_{k,m}$ 处于 PCCS 之外,令 $L_{k,m}=1$.

任何一个笛卡尔坐标形式的多维数据集(本文仅指 Pareto 前端的点数据集)都可以通过公式(2)映射到 PCCS 中,如图 1 所示.图 1(a)中的 7 个圆圈表示多目标优化问题测试函数 DTLZ2(包含 3 个目标)的 Pareto 前端中 7 个目标向量;图 1(b)是由 7 行 3 列构成的二维网格,其行数和列数分别对应于图 1(a)中向量个数和目标个数,图 1(a)中每一个点的每一个坐标分量都被公式(2)映射到图 1(b)中对应目标列的某一个格子中,用 $P_k.f_m$ 的形式表示第 k 个目标向量的第 m 个格坐标分量.例如,图 1(a)中 P_7 点的平行格坐标是(6,2,4),在图 1(b)中分别用

$P_7, f_1=6, P_7, f_2=2$ 和 $P_7, f_3=4$ 表示.为了清晰起见,用点划线连接了图 1(b)中同一个目标向量在不同列的格坐标分量.

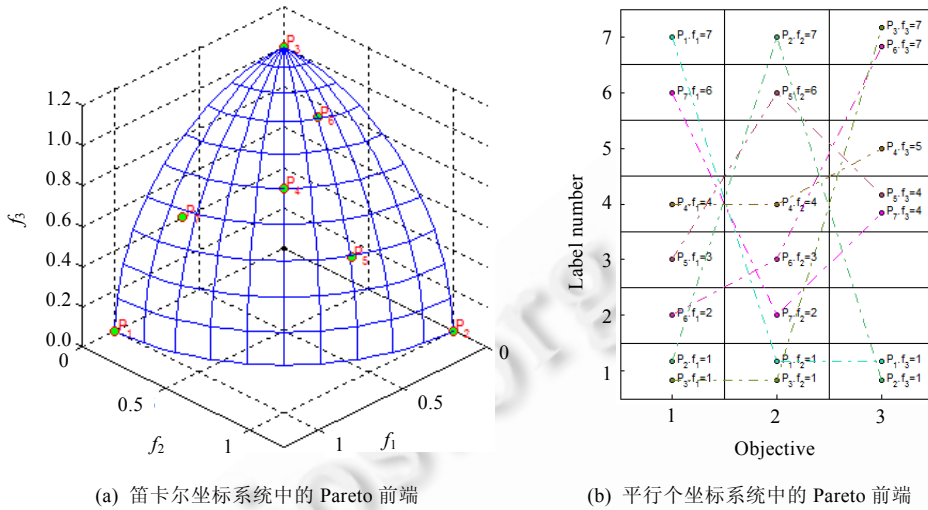


Fig.1 A mapping example for Pareto front from Cartesian coordinate system to parallel cell coordinate system

图 1 Pareto 前端从笛卡尔坐标系向平行格坐标系统映射的示例

尤其值得一提的是, K 不是一个需要用户指定的参数,而是自动随外部档案中 Pareto 最优解集的成员个数变化而变化的目标维度分割数.每一个目标列上的格子边长将随 f_m^{max}, f_m^{min} 和 K 的变化而自动变化.这与 Adaptive Grid^[26], □-Box^[27] 和 GrEA-grid^[28] 中需要用户指定每一个维度上的网格划分数(divisions)或盒子边长(□)不同, PCCS 无需依赖用户对待优化多目标问题的领域先验知识.同时,公式(2)在每个维度上的最大值与最小值之间进行了归一化变换操作.

3.2 Pareto熵及其差熵

熵是一种度量微观分布均匀性的方法.在热力学中,熵表示系统混乱状态,而在生态学中熵表示物种的多样性.当存储在外部档案中的近似 Pareto 前端被映射到 PCCS 后,可以采用熵来度量近似 Pareto 前端的分布均匀性,间接地表达当前种群的多样性特征.如果近似 Pareto 前端发生变化,则会引起 PCCS 中的对应的格坐标分量重新分布,从而引起熵发生变化.因此,可以采用差熵来表示近似 Pareto 前端在相邻迭代时刻的熵的变化大小:

- 差熵越大,意味着近似 Pareto 前端重新分布的范围越大,这可能由于:① 算法生成的新解占优了大量的旧解而引起 Pareto 前端发生大范围变化;② 新解引起了 f_m^{max} 或者 f_m^{min} 发生变化而导致第 k 列上的格坐标分量重新分布.
- 相反,差熵越小,近似 Pareto 前端重新分布的范围越小,这是因为在外部档案维护过程中,质量更好(如个体密度较小)的新解替换了质量较差(个体密度较大)的旧解而引起个别格坐标分量的重新分布.

从而,通过差熵可以推测当前种群的发现新解的潜力,进而估计种群所处的进化状态,如收敛状态、多样化状态和停滞状态.这种进化状态是设计开发和开采进化策略的依据.

由于采用熵来度量近似 Pareto 前端的分布特征,间接地评估对应的 Pareto 最优解集的性能.本文为了强调在 PCCS 中这种分布熵评估的对象是近似 Pareto 最优解集所对应的近似 Pareto 前端,故将其称为 Pareto 熵.

在第 t 次迭代过程中,外部档案中存储的近似 Pareto 前端的 Pareto 熵 $Entropy(t)$ 可以根据公式(3)计算:

$$Entropy(t) = - \sum_{k=1}^K \sum_{m=1}^M \frac{Cell_{k,m}(t)}{KM} \log \frac{Cell_{k,m}(t)}{KM} \tag{3}$$

其中, $Cell_{k,m}(t)$ 表示近似 Pareto 前端被映射到 PCCS 后,落在第 k 行第 m 列的格子中的格坐标分量的个数.

在相邻两次迭代 $t-1$ 和 t 中,表示近似 Pareto 前端变化程度的差熵 $\Delta Entropy(t)$ 可以通过公式(4)进行计算:

$$\Delta Entropy(t) = Entropy(t) - Entropy(t-1) \quad (4)$$

Pareto 熵及其差熵在进化过程中的变化趋势与待优化多目标问题的性质和优化算法的求解能力紧密相关.对于单模态多目标优化问题,在进化过程的前期,优化算法通过发现的新解占优较多的旧解而推动种群向真实 Pareto 前端方向收敛,从而引起 Pareto 熵及其差熵发生较大变化;而在进化过程的后期,当种群逼近真实 Pareto 前端后,算法发现的新解只能占优其邻近的个别旧解,或者替换个别适应度较差(如个体密度较大)的旧解,仅能引起 Pareto 熵及其差熵发生很小的变化.但对于多模态多目标优化问题,当优化算法发现的新解跳出局部极值时,新解将会占优处于局部 Pareto 前端上的大量旧解,从而引起 Pareto 熵及其差熵发生急剧变化,直到种群重新聚集在新的局部 Pareto 前端附近为止.根据这一特点,利用 Pareto 熵及其差熵可以识别待优化问题的模态性质和估计种群所处的进化状态,从而动态地调整优化算法的进化策略.

为了说明 Pareto 熵及差熵在不同模态下的变化特征,图 2 给出了本文算法(详细内容见第 4 节)采集的 Pareto 熵及差熵变化曲线.其中,存储 Pareto 最优解集的外部档案的最大容量设定为 100.图 2(a)是由具有 3 个目标的单模态测试问题 DTLZ2 产生的 Pareto 熵及差熵变化曲线,图 2(b)是由具有 2 个目标的多模态测试问题 ZDT4 产生的 Pareto 熵及差熵变化曲线.在图 2(a)中的进化前期阶段(大约 $Iteration < 40$,不同算法对同一优化问题可能会有不同的分界值),Pareto 熵及差熵发生较大幅度变化,表明算法产生的新解在不断占优外部档案中的旧解,导致近似 Pareto 前端发生大范围变化;而在图 2(a)中的进化后期阶段($Iteration > 40$),Pareto 熵及差熵仅发生小幅度变化,表明近似 Pareto 前端只发生局部小范围变化,算法产生的新解仅占优个别就解或替换个别质量较低的旧解.对于单模态多目标优化问题,Pareto 熵及差熵在进化过程中只经历了一个周期的剧烈变化.而在图 2(b)中的进化前期阶段(约 $Iteration < 100$),Pareto 熵及差熵经历了大约 8 个周期的较为明显的剧烈变化,表明算法在前期阶段产生的新解频繁地跳出局部 Pareto 前端,反映了算法具有很好的发现高质量新解的能力.

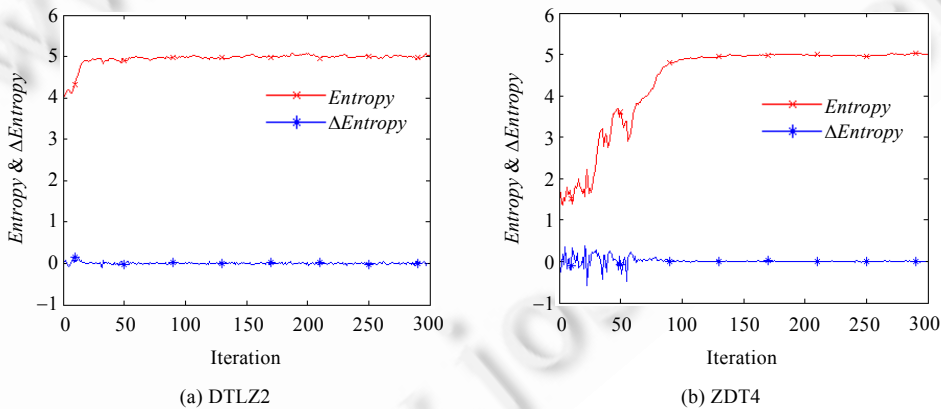


Fig.2 Curves of Pareto entropy and its difference generated by uni-modal and multi-modal MOPs

图 2 单模态和多模态多目标优化问题的 Pareto 熵及差熵进化曲线

需要特别指出的是,Pareto 熵及差熵的剧烈变化周期个数并不等同于局部 Pareto 前端的个数,因为一个高质量的新解有时可以跳过多个局部 Pareto 前端;同时,具有极端边界值的新解有时可能引起 PCCS 的对应目标列表中大量格坐标分量重新分布,而使 Pareto 熵及差熵产生一个剧烈变化周期.

定理 1. 对于具有 M 个目标的最优化问题,当外部档案中存储的 Pareto 最优解的个数为 K 时,对应的近似 Pareto 前端的最小和最大 Pareto 熵分别为 $\log M$ 和 $\log KM$.

证明:

(1) 最小 Pareto 熵

当 Pareto 前端上所有目标向量在 PCCS 中对应的格坐标分量分布处于最不均匀状态时,即,每一目标列的

所有格坐标分量聚集在一个格子中,而该列的其他格子中均为 0 个格坐标分量.在公式(3)中,令第 c 行第 m 列所在的格子聚集了该列的所有格坐标分量,即 $Cell_{k=c,m}(t)=K$,而 $Cell_{k\neq c,m}(t)=0$.且令 $0\log(0)=0$,则最小 Pareto 熵 $Entropy_{\min}$ 为

$$\begin{aligned} Entropy_{\min} &= -\sum_{m=1}^M \frac{Cell_{k=r,m}(t)}{KM} \log \frac{Cell_{k=r,m}(t)}{KM} - \sum_{k=1}^{K-1} \sum_{m=1}^M \frac{Cell_{k\neq r,m}(t)}{KM} \log \frac{Cell_{k\neq r,m}(t)}{KM} \\ &= -\sum_{m=1}^M \frac{K}{KM} \log \frac{K}{KM} - \sum_{k=1}^{K-1} \sum_{m=1}^M \frac{0}{KM} \log \frac{0}{KM} \\ &= \log M. \end{aligned}$$

(2) 最大 Pareto 熵

当 Pareto 前端上所有目标向量在 PCCS 中对应的格坐标分量分布达到最均匀状态时,即,每个格子当且仅当容纳一个格坐标分量时,Pareto 前端对应的 Pareto 熵将达到最大值.在公式(3)中,令 $Cell_{k,m}(t)=1$,则最大 Pareto 熵 $Entropy_{\max}$ 为

$$Entropy_{\max} = -\sum_{k=1}^K \sum_{m=1}^M \frac{Cell_{k,m}(t)}{KM} \log \frac{Cell_{k,m}(t)}{KM} = -\sum_{k=1}^K \sum_{m=1}^M \frac{1}{KM} \log \frac{1}{KM} = \log KM. \quad \square$$

定理 1 表明了近似 Pareto 前端的 Pareto 熵的有界性.

定理 2. 如果具有相同目标个数为 M 的两个近似 Pareto 前端在 PCCS 中对应的格坐标分量分布都达到最均匀状态,则 Pareto 最优解个数越多的近似 Pareto 前端对应的最大 Pareto 熵越大.

证明:用 K_1 和 K_2 分别表示两个近似 Pareto 前端 PF_1 和 PF_2 的 Pareto 最优解的个数,根据定理 1,则 PF_1 和 PF_2 的最大 Pareto 熵分别为 $Entropy_{\max}^{PF_1} = \log K_1M$ 和 $Entropy_{\max}^{PF_2} = \log K_2M$. 令 $0 < K_1 < K_2$, 则

$$Entropy_{\max}^{PF_1} - Entropy_{\max}^{PF_2} = \log K_1M - \log K_2M = \log \frac{K_1}{K_2} < 0. \quad \square$$

根据定理 2,随着进化算法的迭代过程不断推进,存储在外部档案中的 Pareto 最优解个数逐渐增加,并且外部档案更新算法将促使 Pareto 最优解尽可能地均匀分布,因此,Pareto 熵会不断地增加.虽然在某些时刻算法产生的新解将占优 Pareto 前端上的旧解而使外部档案中的 Pareto 最优解的个数减少,暂时性地降低 Pareto 熵,但从整个进化过程的长周期来看,Pareto 熵将保持整体增加的趋势.外部档案更新算法的目标是促使外部档案中的 Pareto 熵朝着最大值 $\log KM$ 不断增大,以使得 Pareto 前端的均匀性逐渐增强.图 2 的 Pareto 熵曲线可以证实这一结论,其中,图 2(a)的最大熵为 5.704,图 2(b)的最大熵为 5.298.在图 2 中,虽然不同模态的多目标优化问题会造成 Pareto 熵曲线具有不同的周期性波动,但 Pareto 熵的总体趋势是不断增加的,并逐渐逼近最大 Pareto 熵.值得一提的是,在多目标优化问题中,随着目标个数的增加,占优阻力(dominance resistance)^[29-31]将逐渐增大,抗占优解(dominance resistant solution,简称 DRS)^[29-32]的数量将会逐渐增加,从而使得 Pareto 熵逼近最大值的难度逐渐增加.例如,图 2(a)中 3 目标 DTLZ2 的 Pareto 熵逼近其最大值 5.704 的差距,比图 2(b)中 2 目标 ZDT4 的 Pareto 熵逼近其最大值 5.298 的差距要大得多.

3.3 进化状态检测

收敛速度和优化精度是基于概率寻优的进化算法的两个相互冲突目标:过分强调收敛速度往往会引起算法早熟而陷入局部极值,相反,过分注重收敛精度将会增加大量的函数评估次数而使得算法复杂度急剧提高.因此,需要有效地设计开发与开采的平衡策略来控制进化算法的多样性和收敛性.但在设计这种平衡策略之前,算法研究者需要回答“在何时采用何种策略到何种程度”的问题.然而在算法设计之前,通常缺乏对待优化问题的先验知识,并且算法还需要兼顾不同类型的待优化问题.这要求具有平衡开发与开采能力的算法在迭代进化过程中能够检测算法自身所处状态,如收敛状态(convergence status)、多样化状态(diversity status)和停滞状态(stagnation status),然后根据不同进化状态来动态调节进化策略,使得算法能够适应不同特征的待优化问题,并在收敛速度和优化精度之间达到最佳平衡.

本质上,检测进化状态就是从种群所处的进化环境中获得实时反馈信息,从而为控制进化过程提供决策依

据,并形成“检测→调节→检测→...”的闭环控制过程.由于多目标优化问题的优化结果不再是单一解,而是由一组非占优解构成的 Pareto 最优解集,因此,多目标优化问题比单目标优化问题具有更加复杂的进化环境,不能简单地通过比较目标值的大小来判断进化状态,从而使具有自适应进化环境的多目标优化算法设计变得更加困难.有的多目标进化算法通过性能评估指标(如 hyper-volume)来评估进化环境,但这类综合性度量指标难以反映进化迭代中的占优与多样化过程细节.

通过上一小节的分析可知,Pareto 熵的变化是由于进化算法获得的新解占优了 Pareto 前端上的旧解,或者替换了质量较差的旧解而引起的.在采用外部档案存储 Pareto 最优解集的多目标进化算法中,通常认为个体密度较大的 Pareto 解的质量较差,在外部档案更新过程中,往往采用个体密度较小的 Pareto 新解替换个体密度较大的 Pareto 旧解,从而增强近似 Pareto 前端的多样性,包括分布均匀性和延展性.因此,通过评估进化过程中的 Pareto 差熵可以推测进化迭代过程中的占优与多样化细节,从而估计多目标进化算法所处的进化状态,为动态调整进化策略提供反馈信息.但是划分不同进化状态的 Pareto 差熵的分界值是这种方法的关键因素,不合理的分界值会导致进化状态的误判,从而可能诱导进化算法采用相反的进化策略.在此,本小节将从理论上分析划分进化状态的 Pareto 差熵临界阈值.

3.3.1 Pareto 熵在外部档案更新算法中的变化分析

在采用外部档案存储 Pareto 最优解集的多目标进化算法,有一个外部档案更新的关键操作.为了清晰表达 Pareto 前端发生变化的各种情况,有必要先介绍外部档案更新操作,如算法 1 所示.

算法 1, 外部档案更新算法.

- 输入:1) 待更新的外部档案 A ;
 2) 外部档案的最大容量 K ;
 3) 进化算法获得的一个新解 P .

输出:更新后的外部档案 A' .

- Step 1. 如果 $A=\emptyset$,则 $A'=\{P\}$,返回 A' . /*情形 I:外部档案为空集*/
 Step 2. 如果 P 被 A 中的任意一个成员 $a_i \in A$ 占优,则直接返回 A . /*情形 II:新解被旧解占优*/
 Step 3. 对任意 $a_i \in A$,如果 a_i 被 P 占优,则 $A=A/\{a_i\}$.
 Step 4. 如果 A 的成员个数 $|A|<K$,则 $A'=A \cup \{P\}$,返回 A' . /*情形 III:新解占优旧解*/
 Step 5. 令 $B=A \cup \{P\}$,对所有 B 的成员 $b_i \in B$,评估 b_i 的个体密度.
 Step 6. 查找 B 中具有最大个体密度的成员 b_{\max} .
 Step 7. 如果 P 就是 b_{\max} ,则 $A'=A$,返回 A . /*情形 IV:新解质量最差,被拒绝进入档案*/
 Step 8. 令 $A'=B/\{b_{\max}\} \cup \{P\}$,返回 A' . /*情形 V:新解替换质量最差的旧解*/

说明:(1) 算法中的“占优”可以是 Pareto 占优、 \square -占优、模糊占优,或者其他定义形式的占优操作算子.

- (2) 算法中的“个体密度”可以是基于空间距离的密度、基于邻居数量的密度,或者其他定义形式的密度度量.

下面根据算法 1 分析各种情形下的 Pareto 熵的变化.

情形 I. 算法刚启动时,外部档案为空集,此时,Pareto 熵 $Entropy(0)=0$.当算法产生一个新解后,则新解进入外部档案.在外部档案映射到 PCCS 后,仅有 1 行 M 列,且每一个格子均容纳一个格坐标分量,即 $Cell_{k,m}(1)=1$,则外部档案更新后的 Pareto 熵变化 $\Delta Entropy_1$ 为

$$\Delta Entropy_1 = Entropy(1) - Entropy(0) = -\sum_{k=1}^1 \sum_{m=1}^M \frac{Cell_{k,m}(1)}{KM} \log \frac{Cell_{k,m}(1)}{KM} - 0 = \sum_{m=1}^M \frac{1}{M} \log \frac{1}{M} = \log M.$$

在情形 I 下,算法具有最小 Pareto 熵.而在图 2 中,因为算法经过种群初始化以后,外部档案中已经存储了一些 Pareto 最优解,因此,图 2 中的 Pareto 熵曲线不是以 $Entropy_{\min}$ 作为起点,而是以种群初始化后近似 Pareto 前端对应的 Pareto 熵作为起点.

情形 II. 当算法产生的新解被外部档案中的旧解占优时,新解不能进入外部档案,此时,Pareto 熵没有发生

变化,即 $\Delta Entropy_{II}=0$.这种情况称为停滞状态.

情形 III. 当算法产生的新解没有被外部档案中的任何旧解占优,但新解占优了外部档案中 $0 \sim |A|$ 个旧解,且从外部档案中移去被占优的旧解后,外部档案仍然有剩余空间容纳新解.下面按照被占优的旧解的个数为 $0, 1, r$ ($r \geq 2$) 时,分别讨论 Pareto 熵的变化.

(a) 被占优的旧解的个数为 0

在这种情况下,新解和外部档案中的旧解互为非占优解,外部档案中的成员个数 $|A| < K$. 因此,新解可以直接进入外部档案,使得外部档案的成员个数增加为 $|A|+1$, Pareto 熵也将增加.假设在 Pareto 前端分布最均匀的情况下,根据定理 2, Pareto 熵的变化 $\Delta Entropy_{III-a}$ 可以表示为

$$\Delta Entropy_{III-a} = Entropy_{\max}(t) - Entropy_{\max}(t-1) = \log M(|A|+1) - \log M|A| = \log \frac{|A|+1}{|A|} \quad (5)$$

(b) 被占优的旧解的个数为 1

在这种情况下,被占优的旧解将被移出外部档案,新解将进入外部档案,属于占优型替换.外部档案的成员个数在更新前后没有变化,但近似 Pareto 前端的分布发生了局部变化, Pareto 熵也将随之发生改变.令外部档案在此时的成员个数为 $H=|A|$, p 和 q 分别表示外部档案更新前后的旧解和新解索引号.则发生占优型替换前后的 Pareto 熵 $Entropy(t-1)$ 和 $Entropy(t)$ 分别为

$$Entropy(t-1) = E - \sum_{m=1}^M \frac{Cell_{h=p,m}(t-1)}{HM} \log \frac{Cell_{h=p,m}(t-1)}{HM} - \sum_{m=1}^M \frac{Cell_{h=q,m}(t-1)}{HM} \log \frac{Cell_{h=q,m}(t-1)}{HM},$$

$$Entropy(t) = E - \sum_{m=1}^M \frac{Cell_{h=p,m}(t)-1}{HM} \log \frac{Cell_{h=p,m}(t)-1}{HM} - \sum_{m=1}^M \frac{Cell_{h=q,m}(t)+1}{HM} \log \frac{Cell_{h=q,m}(t)+1}{HM},$$

其中, $E = - \sum_{h=1}^{H-2} \sum_{m=1}^M \frac{Cell_{h \neq p \wedge h \neq q,m}(t)}{HM} \log \frac{Cell_{h \neq p \wedge h \neq q,m}(t)}{HM}$, 表示在这种情况下,外部档案更新过程中不涉及 p 和 q 的坐标分量所在格子的 Pareto 熵. E 在外部档案更新前后不会发生变化,因此, Pareto 差熵 $\Delta Entropy_{III-b}$ 为

$$\Delta Entropy_{III-b} = Entropy(t) - Entropy(t-1)$$

$$= \left(\sum_{m=1}^M \frac{Cell_{h=p,m}(t)-1}{HM} \log \frac{Cell_{h=p,m}(t)-1}{HM} - \sum_{m=1}^M \frac{Cell_{h=p,m}(t-1)}{HM} \log \frac{Cell_{h=p,m}(t-1)}{HM} \right) +$$

$$\left(\sum_{m=1}^M \frac{Cell_{h=q,m}(t)+1}{HM} \log \frac{Cell_{h=q,m}(t)+1}{HM} - \sum_{m=1}^M \frac{Cell_{h=q,m}(t-1)}{HM} \log \frac{Cell_{h=q,m}(t-1)}{HM} \right) \quad (6)$$

(c) 被占优的旧解的个数为 r ($r \geq 2$)

在这种情况下,外部档案在更新过程中需要移除 r 个旧解,并增加一个新解,外部档案更新后的成员个数为 $|A|-r+1 < |A|$. 根据定理 2, 外部档案更新后的 Pareto 熵将减少.在这种情况下, Pareto 差熵 $\Delta Entropy_{III-c}$ 可以近似等价于以下两个部分:① 一个旧解发生占优型替换引起的 Pareto 差熵,近似等于公式(6)中的 $\Delta Entropy_{III-b}$;② 移除 $r-1$ 个旧解引起的 Pareto 熵的变化 $\Delta Entropy_{Removal}$. 假设在 Pareto 前端分布最均匀的情况下,根据定理 1, 第②部分产生 Pareto 熵的变化量 $\Delta Entropy_{Removal}$ 近似为

$$\Delta Entropy_{Removal} = Entropy_{\max}(t) - Entropy_{\max}(t-1) = \log M(|A|-1) - \log M(|A|-1-(r-1)) = \log \frac{|A|-1}{|A|-r} \quad (7)$$

因此,在被占优的旧解的个数为 r ($r \geq 2$) 的情况下, Pareto 熵发生变化的总和 $\Delta Entropy_{III-c}$ 可以近似表示为

$$\Delta Entropy_{III-c} = \Delta Entropy_{Removal} + \Delta Entropy_{III-b} \quad (8)$$

由公式(7)可知,被占优的旧解个数 r 越大, Pareto 熵减少得越多,从而造成 Pareto 熵发生急剧下降的变化.例如,图 2(b) 中 Pareto 熵曲线发生了很多周期性急剧下降的变化,这是因为进化算法发现的高质量新解跳出了局部 Pareto 前端,引发其占优了大量的旧解,从而引起外部档案的成员个数急剧下降,导致了 Pareto 熵的急剧减少.

情形 IV. 当算法生成的新解与外部档案中的所有旧解互为非占优解,且外部档案的成员个数达到限定值,即 $|A|=K$ 时,将启动个体密度评估操作.如果新解的个体密度在由这个新解与所有旧解组成的新集合中具有最大密度值,即,新解替换外部档案中的任何成员都会引起 Pareto 前端分布更不均匀.此时,新解被禁止替换已有的旧

解, Pareto 熵的变化 $\Delta Entropy_{IV}=0$.

情形 V. 在情形 IV 下, 如果新解的个体密度不是最大值, 则用新解替换具有最大个体密度的旧解, 称为择优型替换. 这时, Pareto 熵的变化与占优型替换相同, 可以根据公式(6)计算, 即 $\Delta Entropy_V = \Delta Entropy_{III-b}$.

3.3.2 进化状态

针对算法 1 的 5 种情况分析 Pareto 熵变化的目的是, 寻找划分进化状态(收敛状态、多样化状态和停滞状态)的 Pareto 差熵的临界阈值. 下面先给出各个进化状态的定义, 然后估计划分进化状态的临界阈值.

定义 4(收敛状态). 在第 t 次迭代过程中, 如果算法获得的近似 Pareto 前端 PF_{appr} 向真实 Pareto 前端 PF_{true} 在目标空间中发生了距离逼近时, 则称该算法在时刻 t 处于收敛状态(convergence status).

推动 PF_{appr} 逼近 PF_{true} 的收敛过程动力包括两种: ① PF_{appr} 上最优解的数量增加; ② 算法产生的新解占优的 PF_{appr} 上的旧解. 根据对算法 1 中各种情形的分析, 情形 I 和情形 III 将推动 PF_{appr} 向 PF_{true} 逼近.

定义 5(多样化状态). 在第 t 次迭代过程中, 如果算法获得的新解替换了 PF_{appr} 上质量较低的旧解, 则称该算法在时刻 t 处于多样化状态(diversity status).

根据算法 1, 新解替换旧解需要满足 3 个条件: ① PF_{appr} 的容量已经饱和; ② 新解与 PF_{appr} 上的所有解互为非占优解; ③ 新解至少比 PF_{appr} 上的某一个旧解具有更好的个体环境适应度(如个体密度). 情形 V 表示的择优型替换将促进 PF_{appr} 上的多样性增加.

定义 6(停滞状态). 在第 t 次迭代过程中, 如果算法获得的新解被拒绝进入 PF_{appr} , 则算法在时刻 t 处于停滞状态(stagnation status).

由算法 1, 当算法产生的新解被拒绝进入 PF_{appr} 时, 可能由于: ① 新解被 PF_{appr} 上的部分旧解占优; ② 新解具有比 PF_{appr} 上的任何一个旧解都差的个体环境适应度(如个体密度). 情形 II 和情形 IV 将保持 PF_{appr} 不发生变化.

3.3.3 阈值估计

根据定义 4~定义 6, 可以从进化算法产生的一个新解来判断算法所处的微观进化状态. 从微观上分析, 不同的新解将使算法处于不同(甚至相互冲突)的进化状态. 但种群在一次迭代过程中将产生 N (种群的个体数量)个新解. 因此, 需要从宏观统计上判断算法所处状态. 在 PCCS 中, 可以分别计算两次进化迭代 $t-1$ 和 t 的 Pareto 熵 $Entropy(t-1)$ 和 $Entropy(t)$, 对应的 Pareto 差熵 $\Delta Entropy = Entropy(t) - Entropy(t-1)$ 可以从宏观上反映算法所处的状态. 为了区分 3 种进化状态, 需要两个临界阈值: 在此, 用 δ_c 表示划分收敛状态与多样化状态的临界阈值, 用 δ_s 表示划分多样化状态与停滞状态的临界阈值.

根据定义 4, 当算法在一次迭代过程中, 种群生成的 N 个新解对 PF_{appr} 产生的宏观综合效果至少相当于占优 PF_{appr} 上的一个旧解时, 则认为算法在此次迭代过程中处于收敛状态.

情形 III-b 分析了占优型替换的 $\Delta Entropy_{III-b}$, 因此, 可以根据公式(6)估计收敛状态下最小 Pareto 差熵, 即, 判断收敛状态的 Pareto 差熵的临界阈值 δ_c .

占优型替换可能导致 Pareto 熵增加, 也可能导致 Pareto 熵减少. 这是因为新解占优旧解时, 新解需要替换旧解, 而新解的在 Pareto 前端中的个体密度可能比旧解大, 也可能比旧解小. 在公式(6)中, 为了使 $\Delta Entropy_{III-b}$ 达到临界状态, 在 PCC 中设计一个极端情况下的例子来分析 Pareto 熵的变化, 如图 3 所示. 假设被替换的旧解 p (如图 3(a)中 P_5)处于最不理想的状态, 即 p 的每一个格坐标分量都与至少 1 个其他旧解对应列的格坐标分量共享格子(如图 3(a)中, $P_5.f_1$ 与 $P_4.f_1$ 共享格子(5,1), $P_5.f_2$ 与 $P_3.f_2$ 共享格子(3,3), $P_5.f_3$ 与 $P_1.f_3$ 共享格子(1,4)); 同时, 假设新解 q (如图 3(b)中 P_5)处于最理想状态, 即 q 的每一个格坐标分量都独占对应列上的一个格子(如图 3(b)中, $P_5.f_1$, $P_5.f_2$ 和 $P_5.f_3$ 的格坐标分量分别独占格子(3,1), (2,2)和(3,3)).

在图 3 中, 令旧解 p 和新解 q 的格坐标分量所在的格子在外部档案更新前容纳的格坐标分量个数分别为 $Cell_{h=p,m}(t-1)=2$, $Cell_{h=q,m}(t-1)=0$. 因此, 根据图 3 表示的极端例子, 采用公式(6)的 Pareto 差熵可以近似估计算法在收敛状态下的临界阈值 δ_c 为

$$\delta_c = \left(\sum_{m=1}^M \frac{1}{HM} \log \frac{1}{HM} - \sum_{m=1}^M \frac{2}{HM} \log \frac{2}{HM} \right) + \left(\sum_{m=1}^M \frac{1}{HM} \log \frac{1}{HM} - \sum_{m=1}^M \frac{0}{HM} \log \frac{0}{HM} \right) = \frac{2}{H} \log 2 \quad (9)$$

例如,在图 3 中,新解 q 占优了旧解 p ,此时,根据公式(9)计算得到的收敛阈值 $\delta_c=0.2772$.而外部档案更新前后(对应图 3(a)和图 3(b))的 Pareto 熵分别为 2.4308 和 2.7080,其二者差值等于此时的 δ_c .

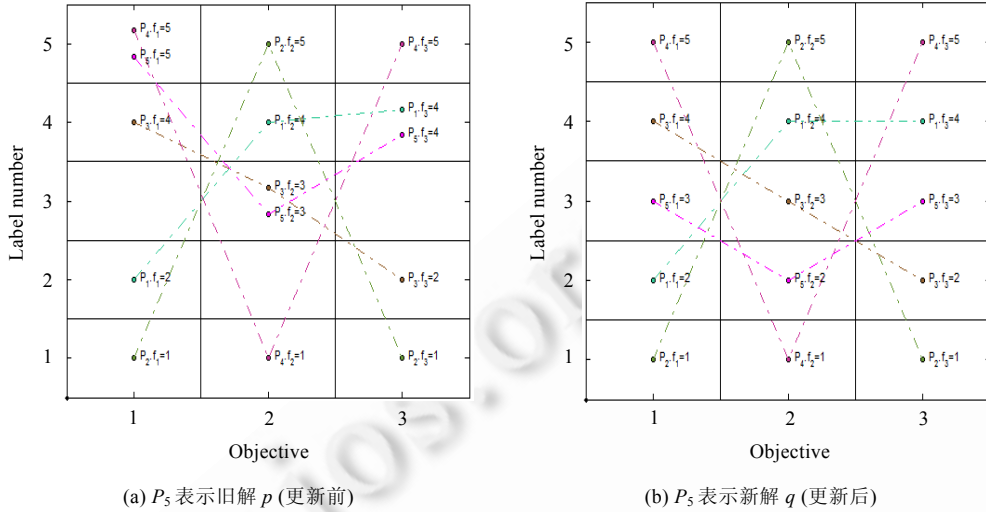


Fig.3 An example in PCCS for replacing an old solution by a new solution
图 3 PCCS 中新解替换旧解的例子

根据定义 5 和定义 6,当算法在一次迭代过程中,种群生成的 N 个新解对 PF_{app} 产生的宏观综合效果至少相当于一个格坐标分量的均匀化分布所引起的 Pareto 差熵时,则认为算法在此次迭代过程中处于多样化状态;否则,认为算法处于停滞状态.采用公式(6)同样可以计算情形 V 描述的择优型替换产生的 Pareto 差熵.但 Pareto 前端在最小化的多样化变动下,仅有 1 个格坐标分量从相对拥挤的格子调整了相对宽松的格子中.在公式(9)中,有 M 个格坐标同时发生了均匀化分布.在多样化状态下,外部档案中的成员数量必定达到了最大容量 K ,即 $H=K$.因此,根据公式(9),停滞状态阈值 δ_s 可以估计为

$$\delta_s = \frac{2}{MK} \log 2 \tag{10}$$

3.3.4 判定条件

根据以上对外部档案更新算法 1 中 Pareto 熵的变化分析、进化状态的定义以及阈值估计分析的结果,可以归纳出种群所处进化状态的判定条件.令外部档案 A 的最大容量为 K ,其在相邻两次迭代 $t-1$ 和 t 时刻的成员数量分别为 $H(t-1)$ 和 $H(t)$,且在更新前后的 Pareto 熵的变化量为 $\Delta Entropy=Entropy(t)-Entropy(t-1)$,则

- 1) 收敛状态的判定条件: $|\Delta Entropy| > \delta_c$; 或 $|H(t)-H(t-1)| > 0$.
- 2) 多样化状态的判定条件: $\delta_c < |\Delta Entropy| < \delta_s$ 且 $H(t-1)=H(t)=K$.
- 3) 停滞状态的判定条件: $|\Delta Entropy| < \delta_s$ 且 $H(t-1)=H(t)$.

除了算法 1 中描述的 5 种情况以外,算法发现的新解获准进入外部档案以后,可能会引起某些目标维度上的最大值 f_m^{\max} 或最小值 f_m^{\min} 发生变化,从而引起 PCCS 中对应目标列上的格坐标分量发生大范围重新分布,导致 Pareto 熵发生较大变化.这时,虽然是择优型替换,但 $|\Delta Entropy| > \delta_c$.因此,这种情况仍然可以归结为收敛状态.

4 基于 Pareto 熵的多目标粒子群优化算法

在多目标粒子群优化算法中,维护外部档案、选择全局最优解 $gBest$ 、选择个体最优解 $pBest$ 、选择粒子运动参数和局部极值扰动是算法的关键组成部分.本节先分别描述这些主要算法部件的策略,然后通过算法流程进行总体集成.

4.1 外部档案维护策略

第二代多目标进化算法以外部档案保留精英解为主要特征,因此需要外部档案更新算法来保存高质量的 Pareto 最优解集.算法 1 描述了外部档案维护方法,但在此需要阐述在 PCCS 中个体密度的评估方法.

在外部档案中的 Pareto 最优解集根据公式(2)被映射到 PCCS 后,其中的任意一个解 $P_i(i=1,2,\dots,K,K$ 为外部档案中成员个数的)个体密度 $Density(P_i)$ 可以根据公式(11)计算得到:

$$Density(P_i) = \sum_{\substack{j=1 \\ j \neq i}}^K \frac{1}{PCD(P_i, P_j)^2} \quad (11)$$

其中, $P_j(j=1,2,\dots,K,j \neq i)$ 是外部档案中不同于 P_i 的任意其他非占优解; $PCD(P_i, P_j)$ 表示 P_i 与 P_j 之间的平行格距离(parallel cell distance),可以由公式(12)计算:

$$PCD(P_i, P_j) = \begin{cases} 0.5, & \text{if } \forall m, L_{i,m} = L_{j,m} \\ \sum_{m=1}^M |L_{i,m} - L_{j,m}|, & \text{if } \exists m, L_{i,m} \neq L_{j,m} \end{cases} \quad (12)$$

在公式(12)中, $L_{i,m}$ 由公式(2)计算.当 P_i 与 P_j 的所有格坐标分量都相同时,意味着这两个解在 PCCS 中完全相同,二者共享对应的 M 个格子,因此它们之间的平行格距离被赋予 0.5.

获得了外部档案中每一个 Pareto 最优解的个体密度后,算法 1 中 5 种情形的执行过程如下:

情形 I. 新解直接进入外部档案.

情形 II. 如果新解被外部档案中的任意一个旧解占优,则算法 1 将阻止新解进入外部档案.

情形 III. 新解不被任意一个旧解占优,但新解占优了 $0 \sim r$ 个旧解,则这些被占优的旧解将被移出外部档案,此时,如果外部档案还有空间,则新解进入外部档案而被保留.

情形 IV. 如果新解与当外部档案中的所有旧解均为非占优解,且外部档案已经达到饱和,则根据公式(11)计算评估新解和所有旧解的个体密度,如果新解具有最大个体密度,则新解被拒绝进入外部档案.

情形 V. 如果新解的个体密度不是最大值,则用新解替换具有最大个体密度的旧解,从而使得近似 Pareto 前端的均匀性得到不断提高.

算法 1 在多目标优化算法中具有通用性.个体密度是个体环境适应度的一种表达形式,不同的个体环境适应度在算法 1 中可以形成不同的外部档案更新策略.

算法 1 在最坏的情况下(即新解与所有旧解互为非占优解,且外部档案已经饱和)需要计算 $K+1$ 个解(新解与外部档案中所有旧解)的个体密度.根据公式(11)计算每一个解的个体密度时,需要计算这个解与另外 K 个解之间的平行格距离.而在公式(12)中,平行格距离涉及 M 个目标.因此,算法 1 的时间复杂度为 $O(MK^2)$.在多目标进化算法中,由于个体适应度评估处于算法的最内层循环,因此,其复杂度决定了整个算法的复杂度.

4.2 全局最优解选择策略

在粒子群优化算法中,全局最优解引导着整个群体的进化方向.在单目标粒子群优化算法中,全局最优解通过比较目标函数就可以确定,因而无需专门处理.但在多目标粒子群优化算法中,全局最优解是一组 Pareto 最优解集,如果不设计有效的选择策略,则可能造成选择压力过小而不能有效推进进化过程.同时,快速收敛是粒子群优化算法的最重要特征,如果不合理利用这一特征,则可能造成算法早熟收敛而失去多样性.通常,评价多目标问题的优化结果包括收敛性和多样性两个方面.因此,在多目标粒子群优化算法的全局最优解选择策略上,需要有效地平衡控制算法的开发和开采过程.全局最优解的选择问题是多目标粒子群优化算法区别于单目标粒子群优化算法和其他多目标优化算法的特有问题之一.

为了使全局最优解能够引导种群平衡开发和开采目标空间以获得兼顾收敛性和多样性的近似 Pareto 前端,需要从外部档案中选择能够代表收敛性和多样性的非占优解作为全局最优解.由于收敛性和多样性通常是相互冲突的,因此需要根据不同的度量标准分别评估外部档案中的 Pareto 最优解的收敛性和多样性.

在 PCCS 中,根据公式(11)计算的个体密度可以作为外部档案中 Pareto 最优解的多样性度量指标,亦即个体密度最小的 Pareto 最优解可以被评选为代表多样性的全局最优解,从而增加 Pareto 前端的多样性.

为了评估外部档案中 Pareto 最优解的收敛性,需要在 PCCS 中引入格占优(cell dominance)和格占优强度(strength of cell dominance)的新概念.

定义 7(格占优). 假设 $L_{x,m}$ 和 $L_{y,m}(m=1,2,\dots,M,M$ 为目标个数)分别是外部档案 A 中任意两个 Pareto 最优解 x 和 y 在 PCCS 中根据公式(2)的格坐标向量,则称 x 格占优 y ,或 y 被 x 格占优,当且仅当

$$\forall m_1=1,2,\dots,M, L_{x,m_1} \leq L_{y,m_1} \wedge \exists m_2=1,2,\dots,M, L_{x,m_2} < L_{y,m_2} \quad (13)$$

记作 $x \succ_c y$.一个 Pareto 最优解 $x^* \in A$ 被称为 PCCS 最优解(或 PCCS 非占优解),当且仅当 $\neg \exists x \in A: x \succ_c x^*$.

定义 8(格占优强度). 一个 Pareto 最优解 $x \in A$ 的格占优强度 $S_c(x)$ 是 x 在 PCCS 中格占优 A 中其他 Pareto 最优解的数目,即

$$S_c(x) = |\{y | y \in A \wedge x \succ_c y\}| \quad (14)$$

格占优强度反映了一个 Pareto 最优解在所有目标上逼近真实 Pareto 前端的综合程度,因此可以作为外部档案中 Pareto 最优解的收敛性度量指标.如果粒子群体围绕具有最大格占优强度的 Pareto 最优解搜索,那么算法获得的近似 Pareto 前端逼近真实 Pareto 前端的机会就很大.图 4 给出外部档案中存储了 8 个 3 目标 Pareto 最优解的格占优及格占优强度的例子.表 1 列出了与图 4 对应的 8 个 Pareto 最优解的笛卡尔坐标($f_1 \sim f_3$)、平行格坐标($L_1 \sim L_3$)和格占优强度(S_c).

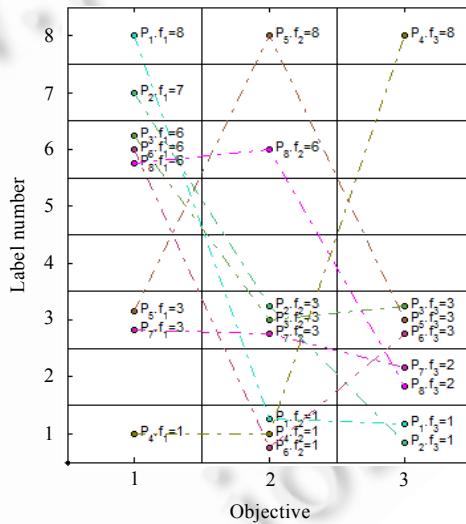


Fig.4 An example of cell dominance and strength of cell dominance

图 4 格占优与格占优强度示例

Table 1 Cartesian coordinates and the parallel cell coordinates and their strength of cell dominance of the eight nondominated solutions in Fig.4

表 1 图 4 中 8 个非占优解的笛卡尔坐标和格坐标及其格占优强度

Code	f_1	f_2	f_3	L_1	L_2	L_3	S_c
P_1	0.601 0	0.000 3	0.004 7	8	1	1	0
P_2	0.527 0	0.030 9	0.023 2	7	3	2	0
P_3	0.471 5	0.030 9	0.065 5	6	3	3	0
P_4	0.208 1	0.003 1	0.200 2	1	1	8	0
P_5	0.329 5	0.093 3	0.071 7	3	8	3	0
P_6	0.502 0	0.007 8	0.074 9	6	1	3	1
P_7	0.331 4	0.033 5	0.039 7	3	3	2	3
P_8	0.478 0	0.069 1	0.036 3	6	6	2	0

从表 1 的笛卡尔坐标值($f_1 \sim f_3$)可以看出, $P_1 \sim P_8$ 互为 Pareto 非占优解. 从图 4 和表 1 的平行格坐标值($L_1 \sim L_3$)可以看出, P_6 格占优了 P_3, P_7 格占优了 P_3, P_5 和 P_8 . 因此, P_6 和 P_7 的格占优强度分别为 1 和 3, 其他 Pareto 最优解的格占优强度均为 0. 在这种情况下, 从推进近似 Pareto 前端逼近真实 Pareto 前端的角度来看, P_7 更加适合担任代表收敛性的全局最优解, 引导种群快速向真实 Pareto 前端快速收敛.

在多目标粒子群优化算法中, 整个群体可以共享一个全局最优解, 但容易导致单点搜索而产生过高的选择压力; 种群中的每一个粒子也可以独享不同的全局最优解, 但将导致分散搜索而出现过低的选择压力. 因此, 算法 2 中为种群粒子提供 $2M$ 个候选的全局最优解, 从而产生适中的选择压力. 算法 2 还通过进化状态控制候选的全局最优解数量来动态适应进化环境, 从而利用进化状态平衡种群的开发和开采过程.

根据个体密度和格占优强度, 可以分别选择代表多样性和收敛性的全局最优解, 如算法 2 所示.

算法 2. 全局最优解选择算法.

输入: 1) 外部档案 A 及目标个数 M ;

2) 算法所处进化状态 $Status \in \{Convergence, Diversity, Stagnation\}$.

输出: 全局最优解 $gBest$.

Step 1. 对 A 中每一个 Pareto 最优解 $k=1, 2, \dots, |A|$, 根据公式(11)和公式(14)分别计算其个体密度 $Density(k)$ 和个体格占优强度 $S_c(k)$.

Step 2. 令全局最优解的候选者集合 $C=\emptyset, D=\{Density(k)\}, S=\{S_c(k)\}$.

Step 3. 如果 $Status=Stagnation$, 则 $C=Top(D, M, 'ASC') \cup Top(S, M, 'DESC')$;

Step 4. 否则, 如果 $Status=Convergence$, 则 $C=Top(D, M-1, 'ASC') \cup Top(S, M+1, 'DESC')$;

Step 5. 否则, 如果 $Status=Diversity$, 则 $C=Top(D, M+1, 'ASC') \cup Top(S, M-1, 'DESC')$.

Step 6. $gBest=RandSelect(C)$.

说明: ① $Top(D|S, n, 'ASC'|'DESC')$ 函数表示从集合 D 或 S 中按照 'ASC(升序)' 或 'DESC(降序)' 方式返回最前面 n 个成员;

② $RandSelect(C)$ 函数表示从集合 C 中随机选择一个成员.

4.3 个体最优解选择策略

大多数已经存在的多目标粒子群优化算法仅保存 1 个个体最优解, 其更新方法是: 当粒子发现的新解被旧解占优时, 则个体最优解保持不变; 当新解占优旧解时, 则采用新解作为个体最优解; 而当新解与旧解互为 Pareto 最优解时, 则随机选择一个解作为个体最优解. 这种单一个体最优解的更新方法简单高效, 但一个个体最优解不足以表达个体粒子所发现和经历的个体近似 Pareto 前端, 损失了过多的个体历史信息, 丧失了个体最优解选择的多样性. 仅有少数多目标粒子群优化算法^[14]采用个体外部档案(personal archive)存储个体 Pareto 最优解. 在本文中, 为了有效表示个体近似 Pareto 前端, 采用个体外部档案保存个体粒子发现的 Pareto 最优解, 其更新过程仍然采用算法 1. 但为了降低计算复杂性, 个体外部档案的最大容量设为全局外部档案最大容量的 1/4.

众所周知, 在粒子群优化算法中, 个体粒子的飞行方向由全局最优解和个体最优解共同决定, 并飞向 $\frac{c_1 r_1 \cdot pBest + c_2 r_2 \cdot gBest}{c_1 r_1 + c_2 r_2}$ 所确定的吸引点^[33]. 在多目标粒子群优化算法中, 全局最优解的更新频率比单目标粒子群优化算法要快得多, 几乎每次迭代都会更新全局最优解. 为了使个体粒子快速跟踪全局最优解的变化, 本文将个体外部档案中选择与全局最优解(此时已经通过算法 2 选择确定)距离最近的成员作为个体最优解.

4.4 自适应参数策略

粒子群优化算法可以通过调节运动方程的参数 ω, c_1 和 c_2 来灵活控制开发和开采过程, 但对于不同性质(如单模态和多模态)的待优化问题, 需要获得进化环境反馈信息来动态决定“在何时将何种参数调节到何种程度”. 第 3.2 节中获得了进化环境中的 Pareto 差熵, 第 3.3 节中分析了进化状态的判定条件. 在粒子群优化算法中, 较大的 ω 和 c_2 以及较小的 c_1 有利于种群的开发过程, 而较小的 ω 和 c_2 以及较大的 c_1 有利于种群的开采过程. 因此, 根

据种群所处进化状态和 Pareto 差熵,可以设计粒子运动参数 ω, c_1 和 c_2 的调节策略,如公式(15)~公式(17)所示:

$$\omega(t) = \begin{cases} \omega(t-1), & \text{如果种群处于停滞状态} \\ \omega(t-1) - 2Step_{\omega}(1 + |\Delta Entropy(t)|), & \text{如果种群处于收敛状态} \\ \omega(t-1) + Step_{\omega}|\Delta Entropy(t)|, & \text{如果种群处于多样化状态} \end{cases} \quad (15)$$

$$c_1(t) = \begin{cases} c_1(t-1), & \text{如果种群处于停滞状态} \\ c_1(t-1) - 2Step_{c_1}(1 + |\Delta Entropy(t)|), & \text{如果种群处于收敛状态} \\ c_1(t-1) + Step_{c_1}|\Delta Entropy(t)|, & \text{如果种群处于多样化状态} \end{cases} \quad (16)$$

$$c_2(t) = \begin{cases} c_2(t-1), & \text{如果种群处于停滞状态} \\ c_2(t-1) + 2Step_{c_2}(1 + |\Delta Entropy(t)|), & \text{如果种群处于收敛状态} \\ c_2(t-1) - Step_{c_2}|\Delta Entropy(t)|, & \text{如果种群处于多样化状态} \end{cases} \quad (17)$$

其中, $Step_{\omega}$, $Step_{c_1}$ 和 $Step_{c_2}$ 分别是 ω, c_1 和 c_2 的调节步长,并分别等于对应参数的最大和最小取值之间的区间长度除以最大迭代次数 T_{max} .根据文献[34], ω 的最小和最大取值范围分别为0.4和0.9,初始值为0.9; c_1 和 c_2 的最小和最大值均分别为0.5和2.5,二者的初始值均为最大值与最小值之间的中点值1.5.

本文算法根据自适应参数策略获得的 ω, c_1 和 c_2 参数进化曲线如图5所示.图5(a)表示单模态3目标DTLZ2的自适应参数进化曲线,其对应的 Pareto 熵和 Pareto 差熵如图2(a)所示;图5(b)表示多模态2目标的自适应参数进化曲线,其对应的 Pareto 熵和 Pareto 差熵如图2(b)所示.不同模态性质的多目标问题具有不同形态的自适应参数曲线,这说明基于 Pareto 熵的自适应参数策略是有效的.

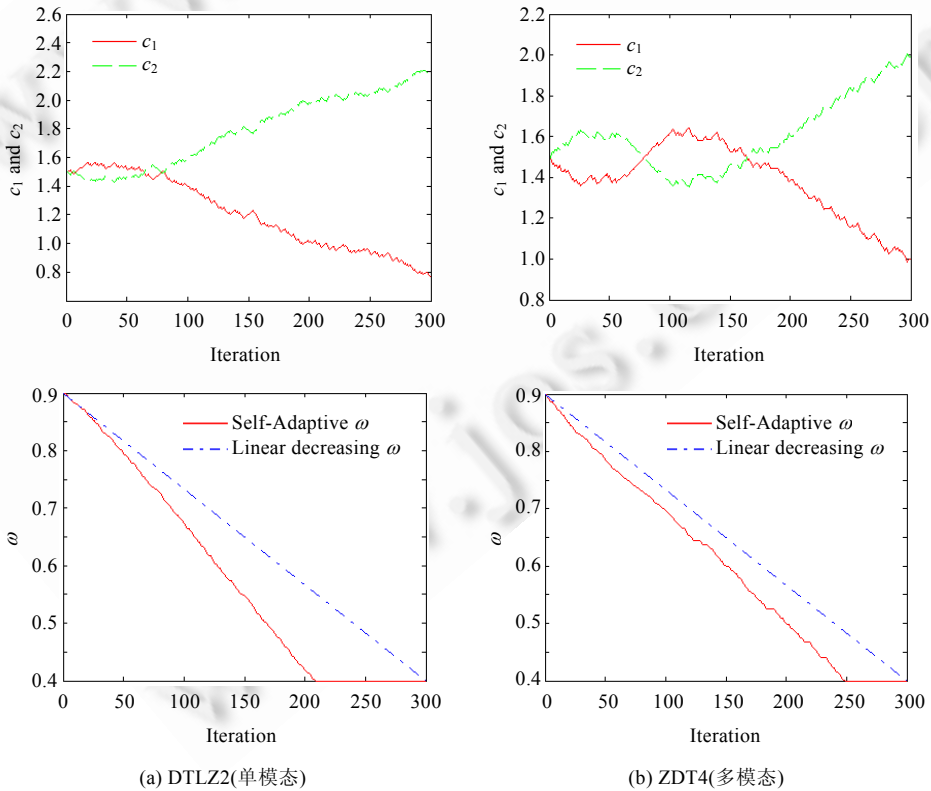


Fig.5 Evolutionary curves of adaptive parameters based on Pareto entropy

图5 基于 Pareto 熵的自适应参数进化曲线

在公式(15)~公式(17)中,当种群处于收敛状态时, ω 和 c_2 的下降速度以及 c_1 的上升速度相对较快;而当种群处于多样化阶段时, ω 和 c_2 的上升速度以及 c_1 的下降速度相对较慢.本文称这种参数调节方法为快降慢升策略,它有利于个体粒子快速收敛到全局最优解,且有充分的机会进行多样化搜索过程.

4.5 局部极值扰动策略

粒子群优化算法的快速收敛特征容易使算法陷入局部极值.不仅在有的单目标粒子群优化算法中采用变异或扰动方法来增强种群多样性,而且几乎所有的多目标粒子群优化算法都采用了扰动(或称为变异)算子来辅助算法跳出局部极值^[24,35].本文采用文献[24]中提出的精英学习策略(elitism learning strategy,简称 ELS)作为扰动方法.在多目标优化粒子群优化算法中,外部档案中存储的 Pareto 最优解都可以看作为精英粒子.与全局最优解面临相同的选择压力问题,如果从外部档案中随机选择一个 Pareto 最优解作为精英粒子,则会导致过小选择压力而降低的扰动成功率.因此,本文利用算法 2 中生成的候选全局最优集 C ,并从 C 中随机选择一个成员作为参与学习的精英粒子.精英学习率 L_r 根据公式(18)计算:

$$L_r(t) = \begin{cases} L_r(t-1) + 2Step_{L_r}(1 + |\Delta Entropy(t)|), & \text{如果种群处于停滞状态} \\ L_r(t-1) - Step_{L_r}|\Delta Entropy(t)|, & \text{如果种群处于收敛状态} \\ L_r(t-1), & \text{如果种群处于多样化状态} \end{cases} \quad (18)$$

在公式(18)中, $Step_{L_r}$ 表示精英学习率步长,等于最大和最小精英学习率之间的区间长度除以最大迭代次数 T_{max} .在文献[24]描述的单目标优化问题中,最大和最小精英学习率分别取值为 1.0 和 0.1,初始值为最大精英学习率;而本文针对多目标优化问题,为了降低参与精英学习的比率,最大精英学习率设定为 0.5.

当种群陷入停滞状态时,公式(18)快速提升学习速率,使得算法有更多的扰动机会来突破局部极值;相反,当种群处于收敛状态时,公式(18)降低了学习速率,使得种群有更多的机会抓住当前有利状态来推动 Pareto 前端向真实 Pareto 前端逼近.在局部极值扰动策略中,扰动幅度为对应决策变量的取值范围乘以高斯变异因子,如公式(19):

$$x_d(t) = x_d(t-1) + (x_d^{max} - x_d^{min}) \times Gaussian(0, r^2) \quad (19)$$

其中, x_d 是参与精英学习粒子的被随机选中的第 d 个决策变量, x_d^{max} 和 x_d^{min} 分别是第 d 个决策变量取值范围的最大值和最小值, r 是均匀分布在 $[0,1]$ 上的随机数, $Gaussian(0, r^2)$ 是以 0 和 r 分别为均值和方差的高斯函数.

4.6 总体算法流程

根据多目标粒子群优化算法的框架以及本文的外部档案维护策略、全局最优解选择策略、个体最优解选择策略、自适应参数策略和局部极值扰动策略,基于 Pareto 熵的多目标粒子群优化算法的总体流程如算法 3 所示.

算法 3. 基于 Pareto 熵的多目标粒子群优化算法(peMOPSO).

输入:1) 具有 M 个目标的待优化问题 MOP;

2) 具有 D 个决策变量的搜索空间 S^D ;

3) 初始化参数:最大外部档案容量 K ;种群粒子个数 N ;最大迭代次数 T_{max} .

输出:存储在外部档案中的近似 Pareto 最优解集 $gArchive$.

1. (初始化种群)

1.1. 令迭代计数器 $t=0$;

1.2. 在搜索空间 S^D 中按均匀分布随机生成具有初始位置和零速度的 N 个粒子;

1.3. 根据 MOP 对 N 个粒子分别计算 M 个目标函数值;

1.4. 为每个粒子初始化个体外部档案 $pArchive_i, i=1,2,\dots,N$,并将第 i 个粒子加入到对应的 $pArchive_i$;

1.5. 令全局外部档案 $gArchive=\emptyset$,对每一个粒子调用算法 1,更新 $gArchive$;

2. (更新迭代计数器) $t=t+1$.

3. (评估进化环境)

- 3.1. 根据公式(2),将 $gArchive$ 映射到平行格坐标系统 PCCS 中;
- 3.2. 根据公式(3)和公式(4)分别计算 $gArchive$ 的 Pareto 熵和差熵;
- 3.3. 根据第 3.3.4 节的判定条件评估种群进化状态 $Status$;
- 3.4. 根据公式(11)和公式(14)分别计算 $gArchive$ 中每个 Pareto 最优解的个体密度和个体格占优强度.
4. (自适应调节粒子运动参数)
 - 4.1. 根据种群进化状态 $Status$ 以及公式(15)~公式(17),分别计算当前粒子运动参数 $\alpha(t)$, $c_1(t)$ 和 $c_2(t)$.
5. (更新种群) For $i=1$ to N
 - 5.1. 根据算法 2 选择粒子 i 的全局最优解 $gBest_i$;
 - 5.2. 从中 $pArchive_i$ 中选择一个与 $gBest_i$ 空间距离最近的成员作为粒子 i 的个体最优解 $pBest_i$;
 - 5.3. 根据粒子运动方程式(1)更新粒子 i 的速度和位置;
 - 5.4. 如果一个随机数小于公式(18)确定的精英学习速率 L_r ,则根据公式(19)对粒子 i 进行极值扰动;
 - 5.5. 根据 MOP 评估粒子 i 的 m 个目标函数值;
 - 5.6. 根据算法 1 更新粒子 i 的个体外部档案 $pArchive_i$;
 - 5.7. 根据算法 1 更新全局外部档案 $gArchive$.
6. (检测算法终止条件)
 - 6.1. 如果 $t > T_{max}$,则输出 $gArchive$;否则,跳转第 2 步.

算法 3 的特点是:将全局外部档案映射到平行格坐标系统 PCCS 中,通过其 Pareto 熵及差熵评估群体进化状态以及通过个体密度和格占优强度评估个体适应度,并以此设计具有自适应平衡开发与开采能力的多目标粒子群进化策略.

5 实验及结果分析

5.1 实验设置

(1) 多目标测试问题

本文采用 ZDT^[36]和 DTLZ^[37]两个系列共 12 个函数构成多目标测试问题集,其中,ZDT 系列是 5 个 2 目标优化测试问题,由 ZDT1~ZDT4 和 ZDT6 组成.由于 ZDT5 是布尔函数,需要二进制编码,本文省去了该测试问题.DTLZ 系列是由 DTLZ1~DTLZ7 组成的一个可以改变搜索变量和优化目标个数的可扩展多目标测试问题集.本文选择 DTLZ 系列作为 3 目标优化测试问题.表 2 列出了 12 个多目标优化测试问题的函数表达式、Pareto 前端特征、决策变量和目标数目、决策变量取值范围以及用于评估性能而在真实 Pareto 前端上采取的样本量.

(2) 对等比较算法

为了比较本文算法(peMOPSO)的性能,选取了 8 种常用于比较多目标优化的对等算法.这些算法可以分为两组,其中一组为多目标粒子群优化算法(MOPSO),包括 σ MOPSO^[38],agMOPSO^[3],cdMOPSO^[39],cMOPSO^[40]和 pdMOPSO^[41](为了按名称区分各算法,本文根据关键特征命名了各对比算法). σ MOPSO 算法通过比较目标向量之间的 σ 值来选择全局最优解.agMOPSO 采用自适应网格方法来更新外部档案和选择全局最优解. cdMOPSO 采用拥挤距离来更新外部档案和选择全局最优解.cMOPSO 采用聚类方法划分子群,子群之间交换共享信息. pdMOPSO 算法提出了 3 种策略选择全局最优解:ROUND 策略给与支配粒子群中的个体数量最少的档案个体最高优先权;RANDOM 策略为每个粒子随机地从支配它的档案成员中选取一个最为全局最优解;而 PROB 策略则结合 ROUND 和 RANDOM 策略.另一组为多目标遗传进化算法(MOEA),包括 NSGA-II(NSGAI 的源代码网址: <http://www.iitk.ac.in/kangal>)^[42],SPEA2(SPEA2 的源代码网址: <http://www.tik.ee.ethz.ch/pisa>)^[43]和 MOEA/D (MOEA/D 的源代码网址: <http://dces.essex.ac.uk/staff/zhang/webfmoad.htm>)^[44].各算法的实验参数都按照对应参考文献设置.

Table 2 Test suits for comparing peer algorithms on multiobjective optimization problems

表 2 用于对等算法性能比较的多目标优化测试问题集

Code	Objective functions to be minimized	Features of PF	Variables	Objectives	Range	Sample size in PF
ZDT1	$\begin{cases} \min f_1(\mathbf{x}) = x_1 \\ \min f_2(\mathbf{x}) = g(\mathbf{x})(1 - \sqrt{x_1/g(\mathbf{x})}) \\ \text{s.t. } g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^n x_i\right)/(n-1) \end{cases}$	Convex	30	2	$x_i \in [0, 1]$	1 000
ZDT2	$\begin{cases} \min f_1(\mathbf{x}) = x_1 \\ \min f_2(\mathbf{x}) = g(\mathbf{x})(1 - (x_1/g(\mathbf{x}))^2) \\ \text{s.t. } g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^n x_i\right)/(n-1) \end{cases}$	Concave	30	2	$x_i \in [0, 1]$	1 000
ZDT3	$\begin{cases} \min f_1(\mathbf{x}) = x_1 \\ \min f_2(\mathbf{x}) = g(\mathbf{x})(1 - \sqrt{x_1/g(\mathbf{x})} - x_1 \sin(10\pi x_1)/g(\mathbf{x})) \\ \text{s.t. } g(\mathbf{x}) = 1 + 9\left(\sum_{i=2}^n x_i\right)/(n-1) \end{cases}$	Disconnected multi-modal	30	2	$x_i \in [0, 1]$	536
ZDT4	$\begin{cases} \min f_1(\mathbf{x}) = x_1 \\ \min f_2(\mathbf{x}) = g(\mathbf{x})(1 - \sqrt{x_1/g(\mathbf{x})}) \\ \text{s.t. } g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \end{cases}$	Convex multi-modal	10	2	$x_1 \in [0, 1];$ $x_i \in [-5, 5]$ $i=2, \dots, n$	1 000
ZDT6	$\begin{cases} \min f_1(\mathbf{x}) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ \min f_2(\mathbf{x}) = g(\mathbf{x})(1 - (x_1/g(\mathbf{x}))^2) \\ \text{s.t. } g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \end{cases}$	Concave non-uniform disconnected	10	2	$x_i \in [0, 1]$	420
DTLZ1	$\begin{cases} \min f_1(\mathbf{x}) = 0.5(1 + g(\mathbf{x}_M)) \prod_{i=1}^{M-1} x_i \\ \min f_{m=2:M-1}(\mathbf{x}) = 0.5(1 + g(\mathbf{x}_M))(1 - x_{M-m+1}) \prod_{i=1}^{M-m} x_i \\ \min f_M(\mathbf{x}) = 0.5(1 + g(\mathbf{x}_M))(1 - x_1) \\ \text{s.t. } g(\mathbf{x}_M) = 100 \left[\sum_{x_i \in x_M} x_i - 0.5 - \cos(20\pi(x_i - 0.5)) \right] \end{cases}$	Linear multi-modal	10	3	$x_i \in [0, 1]$	5 000
DTLZ2	$\begin{cases} \min f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \prod_{i=1}^{M-1} \cos(x_i \pi/2) \\ \min f_{m=2:M-1}(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_{M-m+1} \pi/2) \prod_{i=1}^{M-m} \cos(x_i \pi/2) \\ \min f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_1 \pi/2) \\ \text{s.t. } g(\mathbf{x}_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \end{cases}$	Concave	10	3	$x_i \in [0, 1]$	5 000
DTLZ3	$f_{1:M}$ 与 DTLZ2 相同, $g(\mathbf{x}_M)$ 与 DTLZ1 相同	Concave multi-modal	10	3	$x_i \in [0, 1]$	4 000
DTLZ4	$f_{1:M}$ 和 $g(\mathbf{x}_M)$ 均与 DTLZ2 相同, 但在 $f_{1:M}$ 中用 x_i^α 替代 $x_i, i=1, \dots, n; \alpha=100$	Concave non-uniform	10	3	$x_i \in [0, 1]$	4 000
DTLZ5	$f_{1:M}$ 和 $g(\mathbf{x}_M)$ 均与 DTLZ2 相同, 但在 $f_{1:M}$ 中用 $\frac{1+2g(\mathbf{x}_M)x_i}{2+(1+g(\mathbf{x}_M))}$ 替换 $x_i, i=2, \dots, n$	-	10	3	$x_i \in [0, 1]$	5 100
DTLZ6	$f_{1:M}$ 与 DTLZ5 相同, $g(\mathbf{x}_M) = \sum_{x_i \in x_M} x_i^{0.1}$	Non-Uniform	10	3	$x_i \in [0, 1]$	5 100
DTLZ7	$\begin{cases} \min f_{m=1:M-1}(\mathbf{x}) = x_m \\ \min f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \left\{ M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1 + g(\mathbf{x}_M)} (1 + \sin(3\pi f_i)) \right] \right\} \\ \text{s.t. } g(\mathbf{x}_M) = 1 + \frac{9}{ \mathbf{x}_M } \sum_{x_i \in x_M} x_i \end{cases}$	Disconnected multi-modal	20	3	$x_i \in [0, 1]$	4 700

(3) 实验参数

在比较实验中,各算法的种群大小均设为 $N=100$,全局外部档案的最大容量均设为 $K=100$,所有测试函数的最大评估次数均设为 $FE=30000$,所有算法的最大迭代次数为 $T_{\max}=300$.在 cMOPSO 算法的参数设置中,子群数

量为 5,每个子群的粒子数为 20,总粒子数仍为 100.为减少性能分析中随机的影响,每种算法在所有测试函数上均独立运行 30 次.本文的仿真实验在 1.2GHz 双 CPU 和 4GB 内存的笔记本电脑上运行.

(4) 性能指标

为了评估算法获得的近似 Pareto 前端的收敛性和均匀性,本文采用反转世代距离(inverted generational distance,简称 IGD)^[44]作为性能评估指标.IGD 是度量真实 Pareto 前端到算法获得的近似 Pareto 前端之间的距离指标.该指标值越低,表明算法获得的近似 Pareto 前端的收敛性和多样性越好,越接近真实 Pareto 前端.令 \mathbf{P} 为真实 Pareto 最优解集, \mathbf{A} 是进化算法获得的近似 Pareto 最优解集,则 IGD 可以根据公式(20)计算:

$$IGD = \frac{1}{|\mathbf{P}|} \sum_{i=1}^{|\mathbf{P}|} Dist_i \quad (20)$$

其中, $Dist_i = \min_{j=1}^{|\mathbf{A}|} \sqrt{\sum_{m=1}^M \left(\frac{f_m(\mathbf{p}_i) - f_m(\mathbf{a}_j)}{f_m^{\max} - f_m^{\min}} \right)^2}$ 为最小归一化欧式距离; f_m^{\max} 和 f_m^{\min} 分别为 \mathbf{P} 中第 m 目标上的最大值和最小值, $m=1,2,\dots,M$, M 为目标个数; $\mathbf{p}_i \in \mathbf{P}, i=1,2,\dots,|\mathbf{P}|$; $\mathbf{a}_j \in \mathbf{A}, j=1,2,\dots,|\mathbf{A}|$.在本实验中,12 个测试函数的真实 Pareto 最优解集的采样样本数 $|\mathbf{P}|$ 见表 2 的最后一列.

5.2 实验结果及比较分析

表 3 列出了 9 个对等比较算法在 12 个多目标测试问题上的 IGD 性能指标值,其中,IGD 的平均值(mean)和方差值(std.)是同一算法在同一测试问题上独立运行 30 次的统计结果; t -test 值是本文算法与其他对等算法在同一测试问题上进行 t 检验时的 t 值;“+”、“=”和“-”表示本文算法获得的 IGD 值在显著性水平为 5%的双尾 t 检验中分别优于、等于和劣于对应列的对等算法在对应行的测试问题上的显著性区分结果;“Score”表示本文算法显著优于对应列的对等算法在 12 个测试问题中的净胜得分,即得“+”与得“-”的测试问题个数之差.例如,本文算法(peMOPSO)与 SPEA2 进行性能对比时,在 9 个测试问题上显著优于 SPEA2,在 1 个测试问题上与 SPEA2 相当,而在另外 2 个问题上显著劣于 SPEA2,因此,本文算法在所有的 12 测试问题上的净胜得分(列举在 SPEA2 所在列的最后一行)为 $9-2=7$.同时,采用粗体字表示所有对比算法在每一个测试问题中的最小 IGD 值.

从表 3 中可以看出,本文算法在 12 个测试问题中获得了 6 个最优 IGD 值,cdMOPSO 和 MOEA/D 分别获得了 3 个最优 IGD 值,其他对比算法的最优 IGD 值个数为 0.这从总体上表明,本文算法在 ZDT 和 DTLZ 系列测试问题上具有最好的 IGD 性能.尤其值得注意的是,本文算法在 6 个 3 目标 DTLZ 系列上获得了 5 个最优 IGD 值,说明本文算法更加擅长于优化较高目标维数的问题,这主要是因为 PCCS 在外部档案维护操作中引入了“格占优”的个体适应度度量方法.

从表 3 中进一步分析,虽然本文算法在 2-目标的 ZDT 系列测试函数上没有获得最优的 IGD 值,但与在 ZDT 系列函数上获得了最优 IGD 值的对比算法单独比较,本文算法在 ZDT 系列函数上不差于其他对比算法.例如,cdMOPSO 在 ZDT3 和 ZDT6 上胜过 peMOPSO,但 peMOPSO 在 ZDT1,ZDT2 和 ZDT4 上优于 cdMOPSO.同理,MOEA/D 在 ZDT2 和 ZDT4 上优于 peMOPSO,但在 ZDT3 和 ZDT6 上差于 peMOPSO,而二者在 ZDT1 上的性能在统计意义上相当;SPEA2 在 ZDT2 和 ZDT3 上优于 peMOPSO,但在 ZDT4 和 ZDT6 上差于 peMOPSO,而二者在 ZDT1 上的性能在统计意义上相当.

在表 3 的 t 检验结果中,本文算法对其他 8 种测试算法的净胜得分均为正数,其中,本文算法在对比 agMOPSO,cMOPSO,pdMOPSO 和 NSGAI 这 4 种算法时净胜得分均为 12,对比 σ MOPSO 时净胜得分为 11,对比 cdMOPSO 时净胜得分为 5,而对比 SPEA2 和 MOEA/D 时净胜得分均为 7,这表明本文算法在所有测试问题集中总体上显著优于另外 8 种测试算法.

从以上分析可知,peMOPSO 在整个 ZDT 系列和 DTLZ 系列测试函数集上具有明显统计意义上的性能优势,并且在较多目标(3 目标 DTLZ 系列)问题上比较少目标(2 目标 ZDT 系列)问题上具有更加优越的性能.当然,根据没有免费午餐定律(no free lunch,简称 NFL)^[45],我们不能期望本文算法在每一个测试函数上都获得最好的 IGD 值.

Table 3 Experimental results in term of IGD between the nine peer algorithms on 12 test instances

表 3 9种对等算法在 12 个测试实例上的 IGD 性能对比结果

Test instance	peMOPSO (this paper)	σ MOPSO Ref.[38]	agMOPSO Ref.[3]	cdMOPSO Ref.[39]	cMOPSO Ref.[40]	pdMOPSO Ref.[41]	NSGAI1 Ref.[42]	SPEA2 Ref.[43]	MOEA/D Ref.[44]	
ZDT1	Mean	4.08E-03	6.32E-03	1.19E-01	4.24E-03	1.25E-02	5.57E-01	5.06E-01	4.15E-03	4.03E-03
	Std.	2.41E-04	1.16E-03	8.03E-02	2.58E-04	1.78E-03	1.95E-01	7.37E-02	1.77E-18	5.59E-05
	<i>t</i> -test		-9.43E+00	-7.15E+00	-2.33E+00	-2.34E+01	-1.42E+01	-3.41E+01	-1.50E+00	8.76E-01
		+	+	+	+	+	+	=	=	
ZDT2	Mean	4.19E-03	3.68E-01	1.08E-02	4.28E-03	1.78E-02	6.88E-02	7.57E-01	4.10E-03	3.84E-03
	Std.	1.77E-04	3.02E-01	7.47E-03	1.14E-04	5.09E-03	4.38E-02	1.44E-01	2.66E-18	4.34E-05
	<i>t</i> -test		-6.02E+00	-4.40E+00	-2.06E+00	-1.34E+01	-7.37E+00	-2.62E+01	2.63E+00	9.57E+00
		+	+	+	+	+	+	+	-	
ZDT3	Mean	3.39E-03	6.19E-03	3.61E-01	3.06E-03	1.05E-01	3.05E-01	3.57E-01	3.16E-03	8.42E-03
	Std.	1.04E-04	2.90E-03	9.84E-02	7.13E-05	7.05E-02	1.00E-01	3.95E-02	0.00E+00	7.02E-03
	<i>t</i> -test		-4.84E+00	-1.82E+01	1.29E+01	-7.18E+00	-1.50E+01	-4.48E+01	1.09E+01	-3.59E+00
		+	+	-	+	+	+	-	+	
ZDT4	Mean	1.84E-02	1.29E+00	5.80E+00	5.91E+00	3.99E+00	4.03E+00	2.53E+01	2.50E+01	4.87E-03
	Std.	2.26E-02	2.55E+00	2.98E+00	4.52E+00	2.62E+00	1.65E+00	7.21E+00	7.25E-15	8.41E-04
	<i>t</i> -test		-2.50E+00	-9.70E+00	-6.52E+00	-7.59E+00	-1.22E+01	-1.75E+01	-5.52E+03	2.98E+00
		+	+	+	+	+	+	+	-	
ZDT6	Mean	3.67E-03	3.85E-03	4.68E-01	2.99E-03	4.39E-01	2.47E+00	1.65E+00	5.32E-03	3.99E-03
	Std.	4.00E-04	1.20E-03	7.67E-01	1.54E-04	2.37E-02	8.17E-01	9.80E-01	2.66E-18	6.02E-05
	<i>t</i> -test		-7.33E-01	-3.03E+00	7.92E+00	-9.20E+01	-1.51E+01	-8.40E+00	-2.07E+01	-4.03E+00
		=	+	-	+	+	+	+	+	
DTLZ1	Mean	6.19E-02	2.98E+01	2.13E+01	2.75E+01	4.15E+01	6.01E+00	1.41E+01	3.77E+01	7.44E-02
	Std.	6.67E-03	8.75E+00	8.18E+00	9.34E+00	2.29E+01	3.62E+00	6.65E+00	1.45E-14	8.77E-05
	<i>t</i> -test		-1.70E+01	-1.30E+01	-1.47E+01	-9.05E+00	-8.21E+00	-1.05E+01	-2.82E+04	-9.36E+00
		+	+	+	+	+	+	+	+	
DTLZ2	Mean	6.21E-02	9.81E-02	1.03E-01	1.02E-01	1.27E-01	2.07E-01	1.06E-01	8.22E-02	6.63E-02
	Std.	1.94E-03	6.87E-03	9.60E-03	1.34E-02	1.68E-02	4.13E-02	8.38E-03	2.83E-17	7.27E-05
	<i>t</i> -test		-2.52E+01	-2.11E+01	-1.45E+01	-1.90E+01	-1.75E+01	-2.55E+01	-5.19E+01	-1.09E+01
		+	+	+	+	+	+	+	+	
DTLZ3	Mean	7.48E-02	3.83E+01	2.87E+01	4.46E+01	4.76E+01	2.16E+01	1.65E+01	4.88E+01	9.57E-02
	Std.	1.29E-02	1.51E+01	1.11E+01	1.02E+01	2.87E+01	1.09E+01	7.56E+00	0.00E+00	1.87E-04
	<i>t</i> -test		-1.26E+01	-1.29E+01	-2.19E+01	-8.27E+00	-9.87E+00	-1.08E+01	-1.89E+04	-8.08E+00
		+	+	+	+	+	+	+	+	
DTLZ4	Mean	4.43E-02	1.07E-01	1.10E-01	1.02E-01	2.28E-01	5.05E-02	7.30E-02	7.29E-02	7.17E-02
	Std.	3.72E-03	2.28E-02	2.95E-02	3.66E-02	8.45E-02	3.46E-03	5.09E-02	1.42E-17	6.19E-04
	<i>t</i> -test		-1.37E+01	-1.11E+01	-7.88E+00	-1.09E+01	-6.12E+00	-2.81E+00	-3.84E+01	-3.63E+01
		+	+	+	+	+	+	+	+	
DTLZ5	Mean	7.05E-03	1.65E-02	1.48E-02	6.05E-03	1.62E-02	2.73E-02	8.05E-03	1.41E-02	9.56E-03
	Std.	7.56E-04	5.34E-03	3.28E-03	7.29E-04	2.48E-03	1.34E-02	1.63E-03	3.54E-18	1.07E-05
	<i>t</i> -test		-8.77E+00	-1.15E+01	4.75E+00	-1.76E+01	-7.57E+00	-2.77E+00	-4.68E+01	-1.66E+01
		+	+	-	+	+	+	+	+	
DTLZ6	Mean	5.07E-03	1.82E-01	2.46E-01	5.23E-03	7.38E-02	1.56E+00	1.47E+00	2.49E-01	9.57E-03
	Std.	2.75E-04	1.61E-01	3.66E-01	3.90E-04	1.55E-01	6.04E-01	6.09E-01	5.67E-17	8.73E-06
	<i>t</i> -test		-5.47E+00	-3.30E+00	-1.66E+00	-2.21E+00	-1.29E+01	-1.20E+01	-4.44E+03	-8.17E+01
		+	+	=	+	+	+	+	+	
DTLZ7	Mean	4.12E-02	1.55E-01	3.17E-01	5.78E-02	2.52E-01	1.03E-01	6.14E-01	6.24E-02	9.23E-02
	Std.	1.15E-03	2.75E-01	1.55E-01	8.49E-03	1.88E-01	1.89E-02	1.16E-01	0.00E+00	3.83E-03
	<i>t</i> -test		-2.08E+00	-8.89E+00	-9.65E+00	-5.60E+00	-1.63E+01	-2.47E+01	-9.18E+01	-6.38E+01
		+	+	+	+	+	+	+	+	
Better (+)		11	12	8	12	12	12	9	9	
Same (=)		1	0	1	0	0	0	1	1	
Worse (-)		0	0	3	0	0	0	2	2	
Score		11	12	5	12	12	12	7	7	

注：“+”、“=”和“-”表示本文算法获得的 IGD 值在显著性水平为 5% 的双尾 *t*-test 测试中分别优于、等于和劣于对应对列的对等算法在对应对应的测试问题上的 IGD 值。“Score”表示本文算法显著优于对应对列的对等算法在 12 个测试问题中的净胜得分,即,得“+”与得“-”的测试问题个数之差。

由于 IGD 性能指标能够同时反映算法的收敛性和多样性,因此本文算法在所有测试算法中获得了最好的

收敛性和多样性的折中效果,说明本文算法在外部档案维护策略、最优解选择策略和自适应参数调整策略等方面能够很好地平衡开发与开采进化过程.

文献[46]通过分析测试算法在所有测试问题上的性能指标排名均值和方差来比较算法的准确性(accuracy)和稳定性(stability),本文也借助这一方法进一步比较 9 种测试算法.每一种测试算法在 12 个测试问题上均有一个 IGD 性能排名值,令 $R_A = \{r_1, r_2, \dots, r_{|R|}\}$ 为算法 A 在 $|R|$ 个测试问题中排名集合,则算法 A 的准确性可以用排名均值 $\mu_A = \frac{1}{|R|} \sum_{r \in R} r$ 表示,而其稳定性可以用排名方差 $\sigma_A = \frac{1}{|R|} \sum_{r \in R} (r - \mu_A)^2$ 表示.如果一个算法 A_1 的排名均值和排名方差均小于另一种对比算法 A_2 ,则可以认为 A_1 的性能优于 A_2 .

本文的 9 种对比算法在 12 个测试函数上的排名均值和排名方差如图 6 所示.从图 6 中可以看出,本文算法 peMOPSO 的排名均值和排名方差确定的点位于左下角,即,本文算法的排名均值和排名方差占优了其他 8 种对比算法,这表明本文算法在所有测试算法中具有最好的准确性和稳定性.

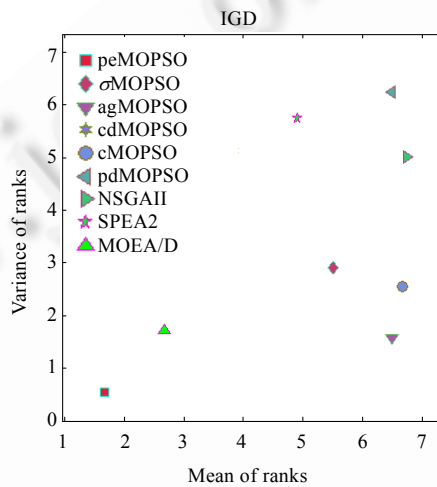


Fig.6 Plots of mean and variance of ranks in term of IGD for 9 algorithms on 12 test instances

图 6 9 个对比算法在 12 个测试实例上的 IGD 综合排名均值与方差对比图

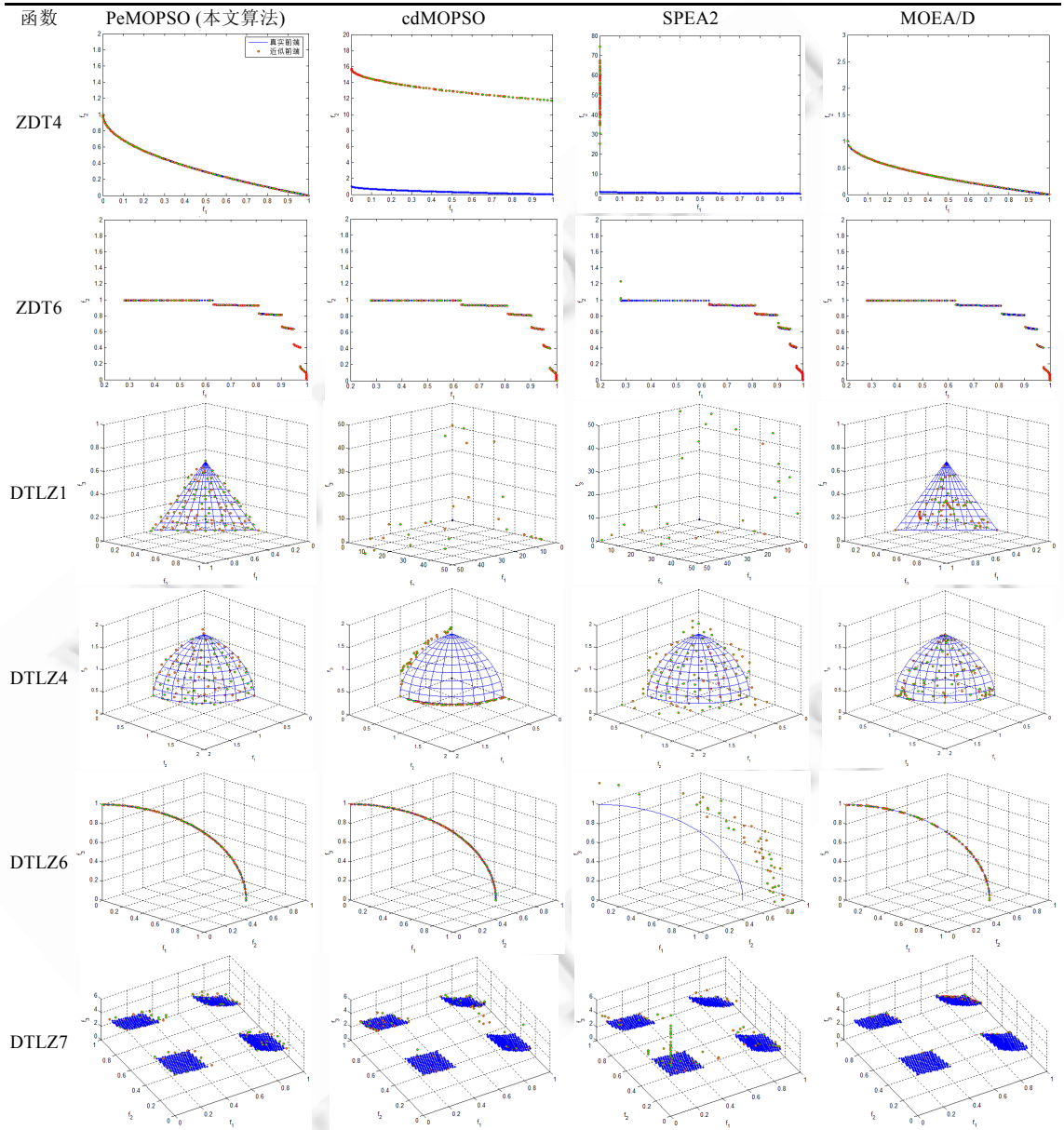
为了直观地展示对比算法的收敛性和均匀性,表 4(表中所有图例均与第 1 个图相同)中给出了本文算法和表 3 中性能较好的 cdMOPSO, SPEAII 和 MOEA/D 算法在相对复杂的测试函数 ZDT4,ZDT6,DTLZ1,ZDLZ3, DTLZ6 和 DTLZ7 上的 Pareto 前端.因为表 3 的实验数据对应了 9 种算法在 12 个测试函数上共 108 个近似前端,为了缩短篇幅,表 4 中省略了那些性能较差的对比算法和相对简单的测试函数上的近似 Pareto 前端.表 4 中的近似,均选自每种算法在每个测试函数上 30 次独立运行实验的第 1 个实验结果.

在表 4 中,对于 2 目标多模态 ZDT4 函数,MOEA/D 具有最好的收敛性和均匀性,本文算法次之,cdMOPSO 虽然也具有较好的均匀性但没有收敛到真实前端,而 SPEA2 则效果最差.对于 2 目标非均匀且不连续 ZDT6 函数,本文算法和 cdMOPSO 具有基本相当的直观收敛性和均匀性,MOEA/D 虽然收敛于真实 Pareto 前端但均匀性相对较差,而 SPEA2 的收敛性和均匀性均最差.对于 3 目标多模态 DTLZ1,MOEA/D 具有最好的收敛性但均匀性较差,部分区域没有被 Pareto 最优解覆盖,本文算法虽然有个别点没有完全收敛到真实 Pareto 前端,但具有很好的分布均匀性,而 cdMOPSO 和 SPEA2 完全没有收敛到真实 Pareto 前端.对于 3 目标非均匀 DTLZ4,本文算法具有很好的收敛性和均匀性,而 MOEA/D 具有较好的收敛性但均匀性次之,cdMOPSO 和 SPEA2 的效果最差.本文算法和 cdMOPSO 在 3 目标 DTLZ6 上具有基本相当的最好收敛性和均匀性,MOEA/D 的均匀性稍差,而 SPEA2 在不能收敛到真实 Pareto 前端.对于 3 目标不连续的多模态 DTLZ7 函数,MOEA/D 虽然收敛到真实 Pareto 前端但极不均匀,cdMOPSO 和 SPEA2 均不能收敛到 4 个簇上,本文算法虽然有个别 Pareto 最优解未能完

全收敛到真实 Pareto 前端,但所有解均收敛到 4 个簇之一,并且具有最好的均匀性.

Table 4 Pareto fronts of some peer algorithms on some relatively hard test instances

表 4 部分对比算法在部分较难测试实例上的 Pareto 前端



通过对表 4 中各个近似 Pareto 前端的收敛性和均匀性分析,本文算法由于采用 PCCS 中 Pareto 熵作为反馈信息,自适应调节了算法的收敛性和均匀性,因而表现出最好的平衡收敛性和均匀性效果.

同时,为了进一步考察算法的单一性能指标,本文还分别采用世代距离(generation distance,简称 GD)^[47]和空间度量(spacing)^[48]指标来度量算法的收敛性和均匀性性能.为了节省篇幅,这里仅给出 GD 和 Spacing 的综合排名均值与方差图,如图 7 所示,其中,综合排名均值与方差的含义与图 6 相同.从图 7 可以看出,MOEA/D 获得了最好的 GD 收敛性,本文算法仅次于但非常接近 MOEA/D 的 GD 性能.而在空间均匀性方面,本文算法大大优于其

他算法.从图 6 和图 7 可以看出,本文算法具有很好的平衡收敛性和均匀性的能力.

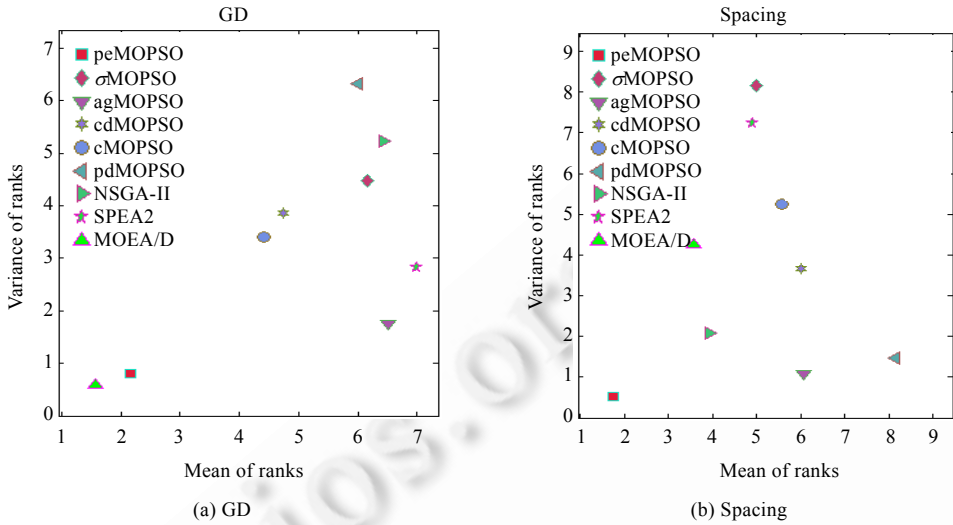


Fig.7 Plots of mean and variance of ranks in term of GD and Spacing for 9 algorithms on 12 test instances
图 7 9 个对比算法在 12 个测试实例上的 GD 和 Spacing 综合排名均值与方差对比图

6 结 论

本文通过目标空间变换方法将笛卡尔形式的近似 Pareto 前端映射到平行格坐标形式的二维网格中,采用 Pareto 熵及其差熵估计种群的多样性和进化状态,并以此为反馈信息来设计具有自适应调节开发与开采过程的进化策略.同时,在新的目标空间 PCCS 中引入格占优和基于格坐标距离的个体密度来评估外部档案中 Pareto 最优解的个体环境适应度,并以此作为外部档案更新和全局最优解选择的度量指标.在最优解选择策略上,本文采用格占优强度和个体密度分别评估个体的收敛性和多样性潜力,并采用多个全局最优解来领导种群的进化方向,有效地兼顾了开发和开采过程.而在粒子飞行参数调节上,依据种群进化状态自适应调节 ω, c_1 和 c_2 ,使得算法能够适应无先验知识的待优化问题.对比实验结果表明:与其他 8 种对等算法(包括 5 种 MOPSO 算法和 3 种 MOGA 算法)比较,本文算法获得的 IGD 指标值在由 ZDT 和 DTLZ 系列构成的 12 个测试问题上具有 t 检验意义上的显著性能优势,表明本文算法在总体上具有最好的收敛性和多样性性能.同时,通过排名均值和排名方差比较分析,本文算法具有最好的准确性和稳定性.

本文算法在较多目标(3 目标 DTLZ 系列)函数上比在较少目标(2 目标 ZDT 系列)函数上具有更好的优化性能,这表明 PCCS 可能更适合于高维多目标优化问题(many-objective optimization problem,简称 MaOP).因此,我们下一步的工作是尝试将 PCCS 与其他策略相结合,以解决高维多目标优化问题.另外,进一步从不同角度分析和判别种群进化状态,将有利于建立多样化的自适应多目标优化算法.

致谢 本文第一作者于 Oklahoma State University 访问研究期间得到了国家留学基金管理委员会提供的资金支持,在此表示感谢.

References:

[1] Kennedy J, Eberhart R. Particle swarm optimization. In: Proc. of the IEEE Int'l Conf. on Neural Networks. Piscataway: IEEE Service Center, 1995. 1942-1948. [doi: 10.1109/ICNN.1995.488968]

- [2] Coello Coello CA, Lechuga MS. MOPSO: A proposal for multiple objective particle swarm optimization. In: Fogel DB, ed. Proc. of the IEEE Congress on Evolutionary Computation (CEC 2002). Piscataway: IEEE Service Center, 2002. 1051–1056. [doi: 10.1109/CEC.2002.1004388]
- [3] Coello Coello CA, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. IEEE Trans. on Evolutionary Computation, 2004,8(3):256–279. [doi: 10.1109/TEVC.2004.826067]
- [4] Reyes-Sierra M, Coello Coello CA. Multi-Objective particle swarm optimizers: A survey of the state-of-the-art. Int'l Journal of Computational Intelligence Research, 2006,2(3):287–308.
- [5] Padhye N, Branke J, Mostaghim S. Empirical comparison of MOPSO methods—Guide selection and diversity preservation. In: Tyrrell A, ed. Proc. of the IEEE Congress on Evolutionary Computation (CEC 2009). Piscataway: IEEE Service Center, 2009. 2516–2523. [doi: 10.1109/CEC.2009.4983257]
- [6] Padhye N. Comparison of archiving methods in multi-objective particle swarm optimization (MOPSO): Empirical study. In: Rothlauf F, ed. Proc. of the Genetic and Evolutionary Computation Conf. (GECCO 2009). New York: ACM Press, 2009. 1755–1756. [doi: 10.1145/1569901.1570143]
- [7] Gong MG, Jiao LC, Yang DD, Ma WP. Evolutionary multi-objective optimization algorithms. Ruan Jian Xue Bao/Journal of Software, 2009,20(2):271–289 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.03483]
- [8] Zhou AM, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang QF. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation, 2011,1(1):32–49. [doi: 10.1016/j.swevo.2011.03.001]
- [9] Reyes-Sierra M, Coello Coello CA. Improving PSO-based multiobjective optimization using crowding, mutation and \square -Dominance. In: Coello Coello CA, Aguirre AH, Zitzler E, eds. Proc. of the 3rd Int'l Conf. on Evolutionary Multi-Criterion Optimization (EMO 2005). Berlin: Springer-Verlag, 2005. 505–519. [doi: 10.1007/978-3-540-31880-4_35]
- [10] Li XD. Better spread and convergence: particle swarm multiobjective optimization using the maximin fitness. In: Deb K, Poli R, *et al.*, eds. Proc. of the Genetic and Evolutionary Computation Conf. (GECCO 2004). New York: ACM Press, 2004. 117–128. [doi: 10.1007/978-3-540-24854-5_11]
- [11] Lei DM, Wu ZM. Pareto archive multi-objective particle swarm optimization. Pattern Recognition and Artificial Intelligence, 2006,19(4):475–480 (in Chinese with English abstract). [doi: 10.3969/j.issn.1003-6059.2006.04.008]
- [12] Huang VL, Suganthan PN, Liang JJ. Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. Int'l Journal of Intelligence System, 2006,21(2):209–226.
- [13] Branke J, Mostaghim S. About selecting the personal best in multiobjective particle swarm optimization. In: Runarsson TP, Beyer HG, Burke E, Merelo-Guervos JJ, Whitley LD, Yao X, eds. Proc. of the Parallel Problem Solving from Nature (PPSN IX). LNCS, Berlin: Springer-Verlag, 2006. 523–532. [doi: 10.1007/11844297_53]
- [14] Abido A. Multiobjective particle swarm optimization with nondominated local and global sets. Natural Computation, 2010,9(1): 747–766. [doi: 10.1007/s11047-009-9171-7]
- [15] Leong WF, Yen GG. PSO-Based multiobjective optimization with dynamic population size and adaptive local archives. IEEE Trans. on System, Man, Cybernetics, Part B, 2008,38(5):1270–1293. [doi: 10.1109/TSMCB.2008.925757]
- [16] Yen GG, Leong WF. Dynamic multiple swarms in multiobjective particle swarm optimization. IEEE Trans. on System, Man, Cybernetics, Part A, 2009,39(4):890–911. [doi: 10.1109/TSMCA.2009.2013915]
- [17] Liang JJ, Qu BY, Suganthan PN, Niu B. Dynamic multi-swarm particle swarm optimization for multi-objective optimization problems. In: Soule T, Moore JH, eds. Proc. of the Genetic and Evolutionary Computation Conf. (GECCO 2012). New York: ACM Press, 2012. 1–8. [doi: 10.1109/CEC.2012.6256416]
- [18] Daneshyari M, Yen GG. Constrained multiple-swarm particle swarm optimization within a cultural framework. IEEE Trans. on System, Man, Cybernetics, Part A, 2012,42(2):475–490. [doi: 10.1109/TSMCA.2011.2162498]
- [19] Daneshyari M, Yen GG. Cultural-Based multiobjective particle swarm optimization. IEEE Trans. on System, Man, Cybernetics, Part B, 2011,41(2):553–567. [doi: 10.1109/TSMCB.2010.2068046]
- [20] Kadkol AA, Yen GG. A cultural-based particle swarm optimization framework for dynamic, constrained multi-objective optimization. Int'l Journal of Swarm Intelligence Research, 2012,3(1):1–29. [doi: 10.4018/jsir.2012010101]

- [21] Cooren Y, Clerc M, Siarry P. MO-TRIBES, an adaptive multiobjective particle swarm optimization algorithm. *Computational Optimization and Applications*, 2011,49(2):379–400. [doi: 10.1007/s10589-009-9284-z]
- [22] Tang LX, Wang XP. A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. *IEEE Trans. on Evolutionary Computation*, 2013,17(1):20–45. [doi: 10.1109/TEVC.2012.2185702]
- [23] Liu DS, Tan KC, Goh CK, Ho WK. A multiobjective memetic algorithm based on particle swarm optimization. *IEEE Trans. on System, Man, Cybernetics, Part B*, 2007,37(1):42–50. [doi: 10.1109/TSMCB.2006.883270]
- [24] Zhan ZH, Zhang J, Li Y, Chung SH. Adaptive particle swarm optimization. *IEEE Trans. on System, Man, Cybernetics, Part B*, 2009,39(6):1362–1381. [doi: 10.1109/TSMCB.2009.2015956]
- [25] Inselberg A. The plane with parallel coordinates. *Visual Computer*, 1985,1(2):69–91. [doi: 10.1007/BF01898350]
- [26] Knowles JD, Corne DW. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 2000,8(1):149–172. [doi: 10.1162/106365600568167]
- [27] Deb K, Mohan M, Mishra S. A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. KanGAL Report, 2003002, Kanpur: Indian Institute of Technology (Kanpur), Kanpur Genetic Algorithms Laboratory (KanGAL), 2003.
- [28] Yang SX, Li MQ, Liu XH, Zheng JH. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2013, 17(5):721–736. [doi: 10.1109/TEVC.2012.2227145]
- [29] Purshouse RC, Fleming JF. On the evolutionary optimization of many conflicting objectives. *IEEE Trans. on Evolutionary Computation*, 2007,11(6):770–784. [doi: 10.1109/TEVC.2007.910138]
- [30] Adra SF, Fleming PJ. Diversity management in evolutionary many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2011,15(2):770–784. [doi: 10.1109/TEVC.2010.2058117]
- [31] Lopez A, Coello Coello CA, Oyama A, Fujii K. An alternative preference relation to deal with many-objective optimization problems. In: Purshouse RC, Fleming PJ, Fonseca CM, Greco S, Shaw J, eds. *Proc. of the 3rd Int'l Conf. on Evolutionary Multi-Criterion Optimization (EMO 2013)*. Berlin: Springer-Verlag, 2013. 291–306. [doi: 10.1007/978-3-642-37140-0_24]
- [32] Batista LS, Campelo F, Guimaraes FG, Ramirez JA. A comparison of dominance criteria in many-objective optimization problems. In: *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2011)*. Piscataway: IEEE Service Center, 2011. 2359–2366. [doi: 10.1109/CEC.2011.5949909]
- [33] Van den Bergh F. An analysis of particle swarm optimizers [Ph. D. Thesis]. Pretoria: University of Pretoria, 2002.
- [34] Ratnaweera A, Halgamuge S, Watson H. Self-Organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. on Evolutionary Computation*, 2004,8(3):240–255. [doi: 10.1109/TEVC.2004.826071]
- [35] Hu W, Li ZS. A simpler and more effective particle swarm optimization algorithms. *Ruan Jian Xue Bao/Journal of Software*, 2007, 18(4):861–868 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/861.htm> [doi: 10.1360/jos180861]
- [36] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 2000,8(2):173–195. [doi: 10.1162/106365600568202]
- [37] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multiobjective optimization test problems. In: Fogel DB, ed. *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2002)*. Piscataway: IEEE Service Center, 2002. 825–830. [doi: 10.1109/CEC.2002.1007032]
- [38] Mostaghim S, Teich J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: *Proc. of the IEEE Swarm Intelligence Symp. (SIS 2003)*. Los Alamitos: IEEE Computer Society Press, 2003. 26–33. [doi: 10.1109/SIS.2003.1202243]
- [39] Raquel CR, Nava PC. An effective use of crowding distance in multiobjective particle swarm optimization. In: Beyer HG, O'Reilly UM, eds. *Proc. of the Genetic and Evolutionary Computation Conf. (GECCO 2005)*. New York: ACM Press, 2005. 257–264.
- [40] Pulido GT, Coello Coello CA. Using clustering techniques to improve the performance of a particle swarm optimizer. In: Deb K, Poli R, *et al.*, eds. *Proc. of the Genetic and Evolutionary Computation Conf. (GECCO 2004)*. New York: ACM Press, 2004. 225–237. [doi: 10.1007/978-3-540-24854-5_20]

- [41] Alvarez-Benitez JE, Everson RM, Fieldsend JE. A MOPSO algorithm based exclusively on Pareto dominance concepts. In: Coello Coello CA, Aguirre AH, Zitzler E, eds. Proc. of the 3rd Int'l Conf. on Evolutionary Multi-Criterion Optimization (EMO 2005). Berlin: Springer-Verlag, 2005. 459–473. [doi: 10.1007/978-3-540-31880-4_32]
- [42] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation, 2002,6(2):182–197. [doi: 10.1109/4235.996017]
- [43] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report, TIK-Report 103, Zurich: Swiss Federal Institute of Technology (ETH), 2001.
- [44] Zhang QF, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. on Evolutionary Computation, 2007,11(6):712–731. [doi: 10.1109/TEVC.2007.892759]
- [45] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation, 1997,1(1): 67–82. [doi: 10.1109/4235.585893]
- [46] Chow CK, Yuen SY. A multiobjective evolutionary algorithm that diversifies population by its density. IEEE Trans. on Evolutionary Computation, 2012,16(2):149–172. [doi: 10.1109/TEVC.2010.2098411]
- [47] Deb K, Jain S. Running performance metrics for evolutionary multi-objective optimization. Technical Report, 2002004, Kanpur: Indian Institute of Technology Kanpur, 2002.
- [48] Schott JR. Fault tolerant design using single and multicriteria genetic algorithm optimization [MS. Thesis]. Cambridge: Massachusetts Institute of Technology, 1995.

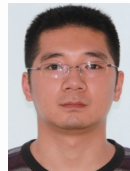
附中文参考文献:

- [7] 公茂果,焦李成,杨咚咚,马文萍.进化多目标优化算法研究.软件学报,2009,20(2):271–289. <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.03483]
- [11] 雷德明,吴智铭.Pareto 档案多目标粒子群优化.模式识别与人工智能,2006,19(4):475–480.
- [35] 胡旺,李志蜀.一种简单而高效的粒子群优化算法.软件学报,2007,18(4):861–868. <http://www.jos.org.cn/1000-9825/18/861.htm> [doi: 10.1360/jos180861]



胡旺(1974—),男,湖南湘潭人,博士,讲师, CCF 会员,主要研究领域为进化计算,信息系统.

E-mail: scuhuwang@126.com



张鑫(1986—),男,硕士生,主要研究领域为进化计算,软件工程.

E-mail: 422231937@qq.com



Gary G. YEN(1962—),男,博士,教授,博士生导师,主要研究领域为进化计算.

E-mail: gyen@okstate.edu