文章编号: 1001-0920(2018)02-0226-09 DOI: 10.13195/j.kzyjc.2016.1451

基于融合多策略改进的多目标粒子群优化算法

杨景明[1,2†], 侯新培[1], 崔慧慧[1], 呼子宇[1], 穆晓伟[1]

(1. 燕山大学 工业计算机控制工程河北省重点实验室，河北 秦皇岛 066004；
2. 国家冷轧板带装备及工艺工程技术研究中心，河北 秦皇岛 066004)

摘 要: 为进一步提高多目标粒子群算法的收敛性和多样性,提出一种多策略融合改进的多目标粒子群优化算法. 首先,引入分解思想以增加Pareto解集的多样性;然后,在速度和位置更新时,引入“多点”变异,即随着迭代次数的递增,根据相应判据对位置的更新作出不同的变异,避免算法早熟现象的发生;最后,将更新后种群和最优解集进行非支配排序,最优解放入精英外部存档. 仿真实验结果表明,与另外4种进化算法对比,所提出算法表现出良好的整体性能.

关键词: 多目标优化；粒子群算法；多策略改进；非支配排序

中图分类号: TP273 **文献标志码:** A

Improved multi-objectiive particle swarm optimzation algorithm based on integrating multiply strategies

YANG Jing-ming[1,2†], *HOU Xin-pei*[1], *CUI Hui-hui*[1], *HU Zi-yu*[1], *MU Xiao-wei*[1]

(1. Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China; 2. National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Qinhuangdao 066004，China)

Abstract: In order to improve the convergence and diversity of the multi-objective particle swarm optimization, an improved multi-objective particle swarm optimzation algorithm integrating multiply strategies is proposed. Firstly, decompositions are introduced to increase the diversity of the pareto sets. Then, in the updating of the velocity and location, the“multi-point”variation is introduced. With the increase of the number of iterations, the updating of the location varies according to the corresponding criterion, which avoids the premature phenomenon of the algorithm. Finally, the updated population and the optimal solution set are subjected to non-dominated sorting and stored in the elite external archive. Simulation experiment results show that the proposed algorithm has better overall performance than other four multi-objective optimizers.

Keywords: multi-objective optimization; particle swarm optimization algorithm; multiple strategies improved; non-dominated ranking

0 引 言

在当今的工程实践和科研中,决策者面对的问题越来越复杂,往往需要同时处理多个目标的问题. 这种在优化设计中要求多个目标达到最优的问题被称为多目标优化问题(MOP),即目标个数一般为2或3. 由于多个目标是相互制约或是相互影响的,处理起来复杂程度较高,传统的算法很难处理这些复杂的问题. 粒子群优化算法[1]是一种启发式群智能随机算法,具有收敛速度快、参数设置简单、易于编程实现的特点. 因其在单目标优化中,具有良好的优化性能,使得一些学者开始将其应用在多目标优化问题中. Coello等[2-3]提出多目标粒子群优化算法,具有里程碑意义. 他们利用网格技术对外部存档进行维护,利用最优粒子库指导粒子更新,并对粒子以及粒子的飞行区域进行变异. 这样可以加快收敛速度和解集多样性,但面对多峰问题,多目标粒子群则表

收稿日期: 2016-11-16；**修回日期:** 2017-04-18.

基金项目: 河北省高等学校创新团队领军人才培育计划项目(LJRC013)；国家冷轧板带及装备工程研究中心开放课题项目(2012006)；河北省自然科学基金面上项目(F2016203249)；河北省博士研究生创新资助项目(CXZZBS2017049).

作者简介: 杨景明(1957−), 男, 教授, 博士生导师, 从事冶金机械综合自动化、先进控制及工程应用等研究；侯新培(1991−), 男, 硕士生, 从事冶金机械综合自动化、多目标决策的研究.

†通讯作者. E-mail: yangjm@ysu.edu.cn

现出陷入局部最优和多样性差的弊端. 部分学者将分解思想引入标准的多目标粒子群中. Ai等[4]提出D^{2}MOPSO算法,将非支配排序加入分解中,将总的问题划分为一系列子问题,并且靠非支配关系建立起领导粒子存档,通过一个新的存档技术更好地保持解的多样性和快速的收敛性.邱飞越等[5]提出了一种大规模变量分解的多目标粒子群算法. 该算法主要针对高维度变量问题,应用随机分解策略,增加关联变量分配到同组的概率,最后将协同进化策略加到算法框架中. 分解的思想可以很好地降低时间复杂度,但是处理一些离散的问题效果不佳. 还有一些学者借鉴了遗传算法等优秀算法的更新变异策略,将不同的策略融合到标准的粒子群算法中加以改进. 谢承旺等[6]提出了一种多策略融合的多目标粒子群算法,他们采用一种简化的K-最邻近方法维持外部存档,并且对档案粒子采取一种有生存期的淘汰机制,优势互补,提高了算法的性能. Hu等[7]提出一种多目标粒子群算法,采用二阶段策略和平行细胞坐标系的方法,提高了算法在收敛速度和解的多样性上的综合性能,能够较好地缓解平衡收敛性和Pareto前沿的多样性的巨大困境. Hu等[8]提出一种增广多目标粒子群算法,将算法分为粒子群优化模块、智能多搜索方法模块和外部存档模块,多模块结合,使得所提出算法的性能在解的质量和收敛速度上得到了提高. Kumar等[9]提出了一种自学习粒子群优化算法来处理多目标优化问题. 此算法在粒子速度和位置更新上依靠4种运算方式,不同方式对应不同的学习来源,独立地提高每一个粒子搜索的有效性. Cheng等[10]提出一种新的最优粒子的搜索策略,即引用局部最优粒子策略,这个混合策略有效地将局部搜索方法引用到基本的多目标粒子群优化算法中,在一定程度上提高了算法的收敛性能. 然而,现有大多数改进的多目标粒子群算法仅仅依靠一种搜索策略来更新粒子的速度或位置,这样的方法在面对复杂的多目标问题时很难得到最优解集,尤其是面对“多峰”问题很容易陷入局部最优,过早收敛.

本文基于部分学者所提出算法的启示,提出一种融合了多种策略的改进多目标粒子群优化算法(MSIMOPSO). 该算法在标准的多目标粒子群优化算法中,引入分解和非支配排序的思想,在初始化时依靠分解的思想,将多目标问题分解为一系列单目标的标量子问题,对粒子的个体极值和全局极值进行随机初始化;然后在粒子的位置和速度更新上引入“多点”变异,即在位置的更新上依据多种变异判据进行高斯扰动,很好地处理多峰问题,加快收敛速度和解的多样性. 在此基础上,算法中将更新后的种群和最优解存档进行非支配排序,然后保存在外部存档中,合理维护外部存档,以其为最终输出结果. 外部存档靠支配关系进行更新可以促进Pareto前沿的多样性和均匀性的保持. 该算法具有快速的收敛性,同时在解的多样性和分布性上也表现出很好的性能. 在解决复杂的多目标问题时,算法的整体性能得到较大提高.

1　基本概念

1.1　多目标优化问题

一般将无约束多目标问题的数学模型描述为如下形式:

$$\begin{aligned}&\min F(x)=\{f_1(x),f_2(x),\cdots,f_m(x)\},\\&\text{s.t. } x\in R^n.\end{aligned}\tag{1}$$

其中:$x=(x_1,x_2,\cdots,x_n)$为n维决策变量;R^n为决策变量的可行域;$F:R^n\to R^m$为由决策空间到目标空间的映射;m为目标空间的维度. 在多目标优化问题中,Pareto最优解的定义如下.

定义1 (Pareto支配)　设x_1,x_2分别为式(1)中的两个决策变量,如果x_1 Pareto x_2支配($x_1\prec x_2$),当且仅当$\forall i\in\{1,2,\cdots,m\},\exists j\in\{1,2,\cdots,m\}$时,使得$f_i(x_1)\leqslant f_i(x_2)\bigwedge f_j(x_1)<f_j(x_2)$.

定义2 (Pareto最优解)　当且仅当$\neg\exists x\in R^n:x\prec x^*$,即$R^*$中不存在支配$x^*$的向量,则称$x^*$为Pareto最优解.

定义3 (Pareto最优解集)　决策空间R^n中所有的Pareto最优解构成的集合称为Pareto最优解集(PS),即:$\mathrm{PS}=\{x|\neg\exists x\in R^n:x\prec x^*\}$.

定义4 (Pareto前沿)　Pareto最优解集PS中,所有的Pareto最优解对应的目标函数值构成的曲线(面),称为Pareto前沿,即$\mathrm{PF}=\{F(x)=(f_1(x),f_2(x),\cdots,f_m(x))^{\mathrm{T}}|x\in\mathrm{PS}\}$.

1.2　标准粒子群算法

标准的PSO算法[11]是受鸟类觅食这种社会行为的启发提出的,模拟鸟类觅食时个体之间自我认知和相互协作,相互交换信息来寻找最优解的一种随机优化的智能技术. 假设种群中的每一个粒子所在搜索空间为n维,每一个粒子的位置矢量和速度矢量分别被表示为$x_i=(x_{i1},x_{i2},\cdots,x_{in})$和$v_i=(v_{i1},v_{i2},\cdots,v_{in})$,粒子$i$在第$d(d=1,2,\cdots,n)$维由$t$时刻到$t+1$时刻的速度和位置更新如下:

$$\begin{aligned}v_{id}(t+1)=&wv_{id}(t)+C_1r_1(\mathrm{pbest}_{id}(t)-x_{id}(t))+\\&C_2(\mathrm{gbest}_d(t)-x_{id}(t)),\end{aligned}\tag{2}$$

$$x_{id}(t+1)=v_{id}(t)+x_{id}(t). \tag{3}$$

其中:w为惯性系数,表示前一时刻的速度对下一时刻的影响,较大的w,全局搜索能力较强,局部搜索能力弱,较小的w反之;C_1和C_2分别为个体自身学习因子和社会学习因子;r_1和r_2为[0,1]的随机数;pbest_{id}为粒子i最好位置pbest_i的第d维分量;gbest_d为群体最好位置gbest的第d维分量.

本文运用的是动态变化的惯性权重,公式如下:

$$w=(T_{\max}-t)^{\rho}(w_{\max}-w_{\min})+w_{\min}. \tag{4}$$

其中:$T_{\max}$为算法的最大迭代次数;t为当前的迭代次数;ρ为一个适当的常数;$w_{\max}$为初始时的惯性权重;$w_{\min}$为到最大迭代次数的惯性权重. 文献[12]中的研究得出:取$w_{\max}=0.9, w_{\min}=0.4$,这样处理惯性权重可以在算法运行前期具有较大的惯性,有利于全局搜索,加大解的多样性;算法运行后期,权重减小有利于局部搜索,加大解的精确度.

2 MSIMOPSO算法

MSIMOPSO算法是一种多策略的粒子群算法,不同策略相互融合,优势互补.本章对MSIMOPSO算法的策略,流程和时间复杂度作出详细介绍.

2.1 算法中的4大策略

2.1.1 分解与支配并存的策略

标准的多目标粒子群算法在处理优化问题时,往往会出现多样性差、收敛速度慢的缺点.分解可以降低问题的复杂度,提高解集的多样性,因此在粒子群算法中引入分解思想,用来提高解集的多样性. Zhang等[13]提出了一种基于分解的多目标进化算法,将多目标优化问题分解为若干个标量子问题加以处理. 他们提出了3种方法用于分解多目标问题,分别是权重和法、切比雪夫分解方法、处罚边界交叉法. 本文中采用切比雪夫分解方法,即

$$\min\ g^{te}(x|\lambda,z^*)=\max_{1\leqslant i\leqslant m}\{\lambda_i|f_i(x)-z_i^*|\},$$
$$\text{s.t. } x\in R^n. \tag{5}$$

其中$z^*=(z_1^*,\cdots,z_m^*)^{\mathrm{T}}$为参考点,$z_i^*=\min\{f_i(x)|x\in R^n\}$. 本文利用均匀分布法选取$N$个(与种群规模相同)权重向量,将要优化的多目标问题分解成N个子问题同时处理.

在初始化种群时,将分解的思想引入标准多目标粒子群算法,依据切比雪夫不等式法将多目标问题分解为一系列单目标标量子问题加以处理. 权重向量的个数为粒子的个体数目N. 为每一个单目标优化问题分配一个互不相同的权重向量,且在当前种群中有唯一个体与之对应.空间被均匀地分解为N个区域,即N个标量子问题. 示意图在两目标问题上以某3个权重向量为例进行说明,如图1所示.

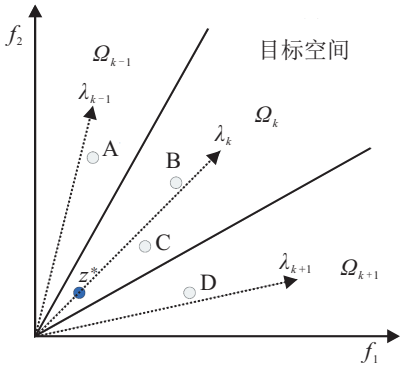

图1 分解与支配共存示意图

图1中3个权重向量将目标空间均匀分为3部分,以空间中的4个粒子为例进行说明,其中$z^*=(f_1,f_2)$是各个目标函数的最小值,所以需要解空间中的解向接近z^*方向飞行. 采用分解的方法,主要依据粒子距权重向量的距离大小进行保留,A、B、D粒子均会保留下来,保持了解的多样性. 若是在分解后加入非支配排序,依靠支配关系,C粒子支配B粒子,将B粒子移除,加入支配关系后A、C、D解保留下来,使得解集更加接近前沿,提高了收敛性. 显然分解和非支配排序具有不同的特点,分解使得多目标问题生成大量的单目标标量子问题,每个子问题都有其领域子问题,产生的新解不仅要求自身更新还需要领域子问题的更新,这样使得算法获得的前沿具有更好的多样性;而非支配排序虽然收敛性能较好,但很难保持多样性. 融合分解和支配,使得PF快速收敛的同时又保持了解集的多样性.结合上面讨论,本文引用先分解后支配的方法来平衡获得前沿的收敛性和多样性问题.

2.1.2 个体极值和全局极值的更新策略

粒子的个体极值和全局极值的初始化都引用分解的思想.算法运行初期,粒子并没有先前的飞行经验,所以将粒子的初始位置当作粒子的个体最优位置,即$\mathrm{pbest}_{id}=x_{id}$. 随机打乱初始化的最优个体位置,分配给$N$个子问题,作为初始全局最优极值gbest,然后依靠速度和位置公式对粒子的速度和位置进行更新,再更新个体极值和全局极值.

个体极值更新主要依据式(5)进行比较,具体比较$g^{te}(\mathrm{pop}(i,d)|\lambda,z^*)$与$g^{te}(pb(i,d)|\lambda,z^*)$的大小,如果$g^{te}(\mathrm{pop}(i,d)|\lambda,z^*)$小于$g^{te}(pb(i,d)|\lambda,z^*)$的计算值,则保留$\mathrm{pop}(i,d)$;反之不变. 使得单个子问题中的$g^{te}(x|\lambda,z^*)$最小的个体保留下来,实际上每一个粒子

趋向于一个更好的位置,使$g^{te}(x|\lambda,z^*)$变得最小.

在粒子群优化算法中,全局最优粒子的作用对于整个种群举足轻重. 多目标问题最困难的就是合理地选择全局最优粒子,不同于单目标问题全局最优解只有一个,而是一个集合. 在全局最优粒子的更新中,邻居粒子起到了重要的作用. 在初始化时,粒子的全局最优gbest是随机打乱其初始位置随机分配给N个子问题的,每一个粒子的位置更新对其周围的邻居粒子都会造成影响. 与邻居粒子作比较,选择离参考点更近的粒子作为全局最优点gbest.

2.1.3　位置更新的“多点”变异

在粒子个体极值和全局极值的更新中,标准的多目标粒子群优化算法粒子会经常出现早熟现象,使得种群过早地结束进化,陷入局部最优. 研究发现,在粒子位置更新时,加些扰动,很容易跳出局部最优. 因此本文在粒子位置更新时引入多点变异,目的是使粒子快速地跳出局部最优点,更好地处理多峰问题. 文献[14]提出了一种水平混合变异操作,受此启发本文采用多点变异,即在位置的更新上加入多准则判据进行变异.将迭代次数引入变异因子中,使得变异因子呈动态形式运作.根据迭代次数的变化,粒子依据不同的判据进行动态变异. 首先对每一个粒子在每一代的变异能力进行分析,决定每一代粒子的变异因子如下所示:

$$Mu(t)=0.05+0.45\frac{\exp\left(\dfrac{5(t-1)}{T_{\max}-1}\right)-1}{\exp(5)-1}. \tag{6}$$

其中:t为当前迭代次数;$T_{\max}$为算法最大迭代次数. 本文算法中粒子的变异采用如下变异思路.

Step 1: 迭代次数t从1到最大迭代次数$T_{\max}$,并且计算式(6)中$Mu(t)$.

Step 2: 判断unifrnd(0, 1, 1, 1)与0.5的大小关系,若$\mathrm{unifrnd}(0,1,1,1)<0.5$,则按下式:

$$\begin{aligned}\mathrm{pop}(i,d)=&\mathrm{normrnd}(((1-\mathrm{rand})gb(1,d)+\\&\mathrm{rand}pb(1,d))/2,|gb(1,d)-pb(i,d)|\end{aligned} \tag{7}$$

更新位置,反之$\mathrm{pop}(i,d)=pb(i,d)$.

Step 3: 判断$\mathrm{unifrnd}(0,1,1,1)$与$Mu(t)$的大小,若$\mathrm{unifrnd}(0,1,1,1)<Mu(t)$,则按下式:

$$\mathrm{pop}(i,d)=(1+\mathrm{rand})pb(i,d)+\mathrm{normrnd}(\mu_1,\sigma_1) \tag{8}$$

更新位置,反之按下式:

$$\mathrm{pop}(i,d)=\mathrm{normrnd}(\mu_2,\sigma_2)\mathrm{pop}(i,d) \tag{9}$$

更新.

Step 4: 结束迭代,输出变异后的新位置. 其中:$Mu(t)$为一个与迭代次数呈一定关系的变异因子,随着t的变化而变化;$\mathrm{normrnd}(\mu,\sigma)$为生成服从均值为$\mu$、标准差为$\sigma$的正态(高斯)分布;$\mathrm{unifrnd}(a,b,c,d)$指生成$c$行$d$列$a$到$b$的随机数.

通过上述的变异操作,可以使得粒子更加快速地收敛到真实的PF,更好地跳出局部最优,尤其对于ZDT4这样的多峰函数,伪前沿非常多,标准的粒子群算法优化时速度和位置更新经常出现陷于局部最优的情况. 在位置更新处引入上述变异后,能够很好地克服早熟收敛的弊端,加快收敛,并且能够很好地覆盖真实的PF.

2.1.4　外部存档的更新与维护

种群经过上述更新变异后,生成新的种群,其与父代种群结合成一个组合种群. 这个组合种群与最优解集存档进行非支配排序产生新的精英外部存档,在下一代更新时替代上一代最优解存档解集. 如此循环迭代,当满足终止条件后以最终的精英外部存档解集作为输出结果. 具体如图2所示.

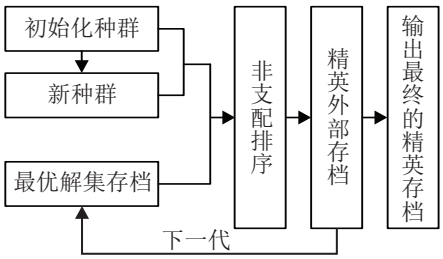

图 2　外部存档更新示意图

外部存档的规模是一个实验参数,可由实验者自己确定,大多数算法中都将外部存档规模设置为与种群规模一致,因此需要对外部存档规模作出合理地维护. 本文采用一种合理的处理机制,即外部存档初始化为空,当存档为空时,新得到的解直解存档;当外部存档中有解时,新得到的解与外部存档中的解进行排序,将等级高的存到外部存档中,若是新解被存档中所有解支配,则外部存档不变;当存储解的数量超出外部存档的规模时,计算拥挤距离,删除拥挤距离最小的解,直至满足规模要求. 将最终的外部存档解集作为最后的输出.

2.2　算法流程

MSIMOPSO算法流程如下.

Step 1: 初始化种群规模N,外部存档规模ArchiveSize,精英外部存档解集archive,总的评价次数FEAS_SUM,以及对粒子的位置和速度进行限定. 初始化与种群规模一致的权重向量,以及将粒子

速度v初始设为0,将粒子原始的位置初始化为粒子的个体极值pbest,将每一个子问题中的个体极值随机打乱重新分配给各个子问题,即初始化的全局极值gbest,初始化参考点$z^* = (z_1^*, z_2^*, \cdots, z_m^*)^{\mathrm{T}}$.

Step 2: 根据式(2) ~ (4)对种群粒子的速度和位置进行更新,根据2.1.3节中提到的变异方法对粒子的位置更新作出合理的变异.

Step 3: 根据上一步得到的速度和位置,对粒子作出越界处理. 根据适应度函数对粒子作出评价,然后对新种群以及原种群结合的种群和外部存档解集进行非支配排序,将得到的最优解存入新的精英外部存档中.

Step 4: 更新精英外部存档,并根据2.1.4节中提到的外部存档维护策略对外部存档的规模进行维护.

Step 5: 根据2.1.2节中提到的分解方法更新个体极值pbest,全局极值gbest,并对参考点z^*作出更新.

Step 6: 判断是否达到最大评价次数. 若满足最大评价次数,则停止运行输出精英外部存档解集;反之,则进入Step2重复进行.

算法流程如图3所示.

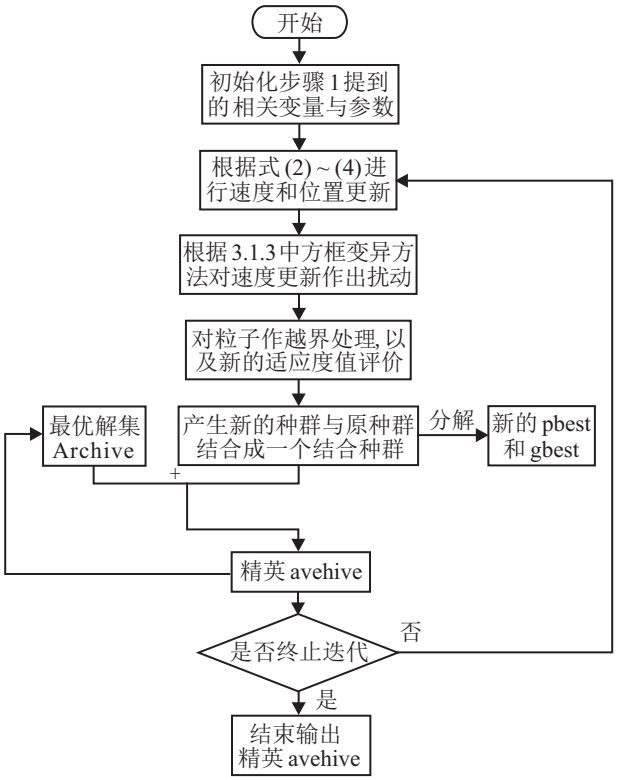

图 3 MSIMOPSO算法流程

2.3 算法时间复杂度分析

本文提出的算法MSIMOPSO,融合了分解、非支配排序和外部精英存档等方法. 测试函数目标维度为m,决策变量维度为n,假定种群大小为N,外部存档大小ArchiveSize为Ar. 分解中算法的时间复杂度为$O(mNT)$,其中T为权重向量λ的邻居向量个数;非支配排序的时间复杂度为$O(mN^2)$. 将非支配解集存到外部存档,对外部存档维护的时间复杂度为$O(m(N+Ar)^2\log(N+Ar))$. 输出最优解即最后精英外部存档解集输出的时间复杂度为$O(mAr)$. 综上可知,所提出的MSIMOPSO算法的时间复杂度为$O(mNT)+O(mN^2)+O(m(N+Ar)^2\log(N+Ar))+O(mAr)$. 可以看出对时间复杂度影响最大的是非支配排序和外部存档的维护. MSIMOPSO的时间复杂度数量级大于MOPSO和MOEA/D,但是与NSGAII和MIMOPSO在同一数量级上. 整体上看,算法的时间复杂度主要与种群规模和外部存档规模有关.

3 性能测试及实验结果分析

3.1 测试函数与评价指标

本文主要针对目标个数为2或3的多目标优化问题进行测试和仿真,采用的测试函数集有ZDT[15]、DTLZ[16]和UF[17]3个系列测试函数. 在本文中,选择ZDT和部分UF函数作为2个目标的测试函数,选择DTLZ部分函数和部分UF函数作为3个目标的测试函数,并采用综合性指标Inverted Generational Distance(IGD)[18]作为本文算法的评价指标. IGD作为一个双重指标反映的是真实的Pareto前沿到多目标算法获得近似Pareto前沿的距离,IGD值越小,说明算法获得的近似Pareto前沿与真实前沿越接近,算法的收敛性和解的分布度越好. IGD的计算公式如下:

$$\mathrm{IGD}(P_{\mathrm{true}}, P) = \frac{1}{|P_{\mathrm{true}}|}\sum_{i=1}^{|P_{\mathrm{true}}|} d(P_{\mathrm{true},i}, P). \quad (10)$$

其中:$d(P_{\mathrm{true},i}, P)$表示的是$P_{\mathrm{true},i}$和$P$在目标空间的欧氏距离,$P_{\mathrm{true}}$是理想的真实PF上的一组均匀采样;$P$为多目标优化算法在决策空间上求得的Pareto解集;$|P_{\mathrm{true}}|$为种群的规模,即为N.

3.2 算法参数设置与性能分析

为了验证本文算法MSIMOPSO的有效性,本文设置了4种应用于多目标优化算法的对比算法,其中包括标准的多目标粒子群算法MOPSO[2],以及MIMOPSO[19]、MOEA/D[13]、NSGAII[20]. MSIMOPSO算法的参数设置如下:惯性权重按式(4)计算,其中$w_{\max} = 0.9, w_{\min} = 0.4, \rho = 2.5$,学习因子$C_1 = C_2 = 1.454\,49$. ZDT系列函数所有算法均取种群规模$N = 100$,外部存档规模ArchiveSize = 100,最高评价次数FEAS = 3×10^4;DTLZ系列函数,所有算法均取种群规模$N = 150$,外部存档规模ArchiveSize = 150,最高评价次数FEAS = 10^5;UF系列函数所有算法均取种群规模N=200,外部存档规模ArchiveSize =200,

表 **1**　指标IGD值统计

函数	MSIMOPSO		MOPSO		MIMOPSO		MOEA/D		NSGAII	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ZDT1	**2.605e-03**	**6.024e-04**	1.704e-02	3.867e-03	4.043e-03	1.145e-03	1.144e-02	4.785e-02	3.330e-03	2.529e-03
ZDT2	**2.609e-03**	**1.098e-03**	1.603e-02	4.647e-03	3.974e-03	5.111e-02	1.807e-01	1.753e-01	3.332e-03	1.933e-02
ZDT3	**2.607e-03**	**8.160e-04**	2.514e-02	4.608e-03	4.650e-03	6.091e-03	1.271e-01	3.665e-02	5.636e-03	3.813e-03
ZDT4	**2.568e-03**	**9.281e-05**	5.747e+00	1.272e+00	4.935e-03	1.102e-04	5.012e-01	2.145e-01	1.937e-02	2.162e-01
ZDT6	**1.567e-03**	**1.413e-04**	2.805e-01	2.830e-02	2.411e-03	2.460e-03	3.231e-03	2.134e-03	1.874e-03	8.261e-04
DTLZ2	**5.085e-02**	**2.679e-04**	1.204e-01	8.223e-04	6.130e-02	2.932e-03	6.735e-02	1.872e-03	6.208e-02	5.390e-03
DTLZ4	1.283e-01	1.559e-04	1.396e-01	6.022e-03	1.270e-01	2.730e-02	7.465e-02	1.674e-03	**6.295e-02**	**4.112e-05**
DTLZ7	**6.171e-02**	**3.323e-04**	8.704e-02	1.664e-03	8.248e-02	1.335e-02	1.483e-01	3.815e-04	1.064e-01	7.894e-04
UF2	2.439e-02	1.816e-03	9.889e-02	2.721e-03	**1.637e-02**	**1.604e-03**	8.470e-02	8.331e-03	2.242e-02	1.913e-03
UF7	**3.885e-02**	**1.420e-03**	8.923e-02	5.731e-03	5.516e-02	1.381e-02	3.042e-01	5.068e-03	8.116e-02	1.331e-02
UF8	2.228e-01	3.620e-03	3.098e-01	1.381e-02	**2.062e-01**	**3.615e-03**	2.129e+00	2.651e-01	2.737e-01	6.554e-03
UF9	**9.745e-02**	**2.630e-02**	3.801e-01	1.464e-01	1.722e-01	3.736e-02	1.383e-01	3.665e-02	4.233e-01	8.717e-02
UF10	**8.643e-02**	**1.709e-03**	3.848e+00	4.221e-02	1.810e-01	4.255e-03	7.671e-01	4.694e-03	1.696e-01	6.128e-02

最高评价次数FEAS $= 3 \times 10^5$. 以上每种算法均采用Intel(R)core i5-4590处理器, 8G RAM, 实验环境为Matlab 2014 a, 独立运行30次. 表1列出了5种算法IGD评价指标的平均值 (Mean) 和标准差 (Std), 其中加黑的为对应算法在同一测试函数上的最好值.

3.3　对两目标优化问题的指标分析

表1给出了MSIMOPSO算法以及4个对比算法优化13个测试函数得到的IGD评价指标的统计. 可以看出, 在两个目标问题上ZDT系列函数的值均获得了较好的结果, 并且在部分函数上比其他测试函数有较明显的优势. 由图4绘制的对比算法 (MOPSO收敛性不佳未绘制) 部分测试函数的Pareto前沿收敛图可以清晰直观地看出, MSIMOPSO和MIMOPSO以及NSGAII在处理多峰问题ZDT4时均能收敛到前沿, MOEA/D和MOPSO很难收敛到前沿, 但通过评价指标的值和前沿图共同表明MSIMOPSO在ZDT4上具有更好的收敛性和分布性. UF2测试函数上MSIMOPSO的IGD值劣于MIMOPSO和NSGAII, 但是本文算法要优于MOPSO和MOEA/D. 表1给出了性能指标IGD的均值和标准差, 能更有效地识别数据的特征, ZDT4函数的指标值更直观地体现出了MSIMOPSO在多峰问题上的优势. 虽然在UF2上MSIMOPSO不如MIMOPSO和NSGAII, 但是通过IGD均值和标准差可看出, MSIMOPSO的均值和标准差也较小. 总体而言, 本文提出的MSIMOPSO算法在两目标问题上能很好地跳出局部最优, 保持良好的分布性和收敛性.

3.4　对三目标优化问题的指标分析

在三目标优化问题上, 由表1给出的数据可清晰地看出, MSIMOPSO算法在DTLZ4上评价指标IGD的值差于NSGAII, 在UF8上评价指标IGD劣于MIMOPSO. 在其他4个测试函数上MSIMOPSO的IGD值均优于其他4个测试函数. 图4中绘制的DTLZ系列是评价10^5次的前沿图, UF系列是评价3×10^5万次的前沿图. 根据前沿图的对比可以发现, MSIMOPSO在DTLZ系列函数上除DTLZ4外均比较完整地覆盖了前沿面, 在UF测试函数上虽不是完全覆盖真实的Pareto面, 但与其他测试函数对比, MSIMOPSO算法在其他测试函数的优化中获得的PF分布性均优于另外4个对比算法 (除UF8外). 由表1中的三目标函数对应的性能指标IGD均值和标准差可以清晰地看出, MSIMOPSO的IGD, MSIMOPSO算法的IGD在大多数测试函数上更加稳定, 而且小于同一测试函数下其他对比算法的IGD值, 更加确切地表明MSIMOPSO算法在三目标测试函数上的优秀性, 而且具有很好的收敛性和解集的多样性.

3.5　IGD收敛性分析

图5给出的是各对比算法在不同测试函数下的指标收敛性曲线. 为了观测清晰, 纵轴取IGD以10为底的对数. 采用IGD值为30次独立实验的平均值. ZDT系列函数总评价3×10^4次, 每评价1 000次记录一次IGD值; DTLZ函数评价10^5次, 每评价2 000次记录一次IGD值, UF系列函数评价3×10^5次, 每评价10 000次记录一次IGD值. 通过分析图5性能指标收敛曲线, 在ZDT1～ZDT4, 以及ZDT6测试函数上, MSIMOPSO的IGD值均为最小, 除ZDT3函数下降速度缓慢外其他测试函数下降非常快速稳定. 在测试函数DTLZ2, DTLZ4和DTLZ7对比中, MSIMOPSO在DTLZ2和DTLZ7上的IGD值下降最

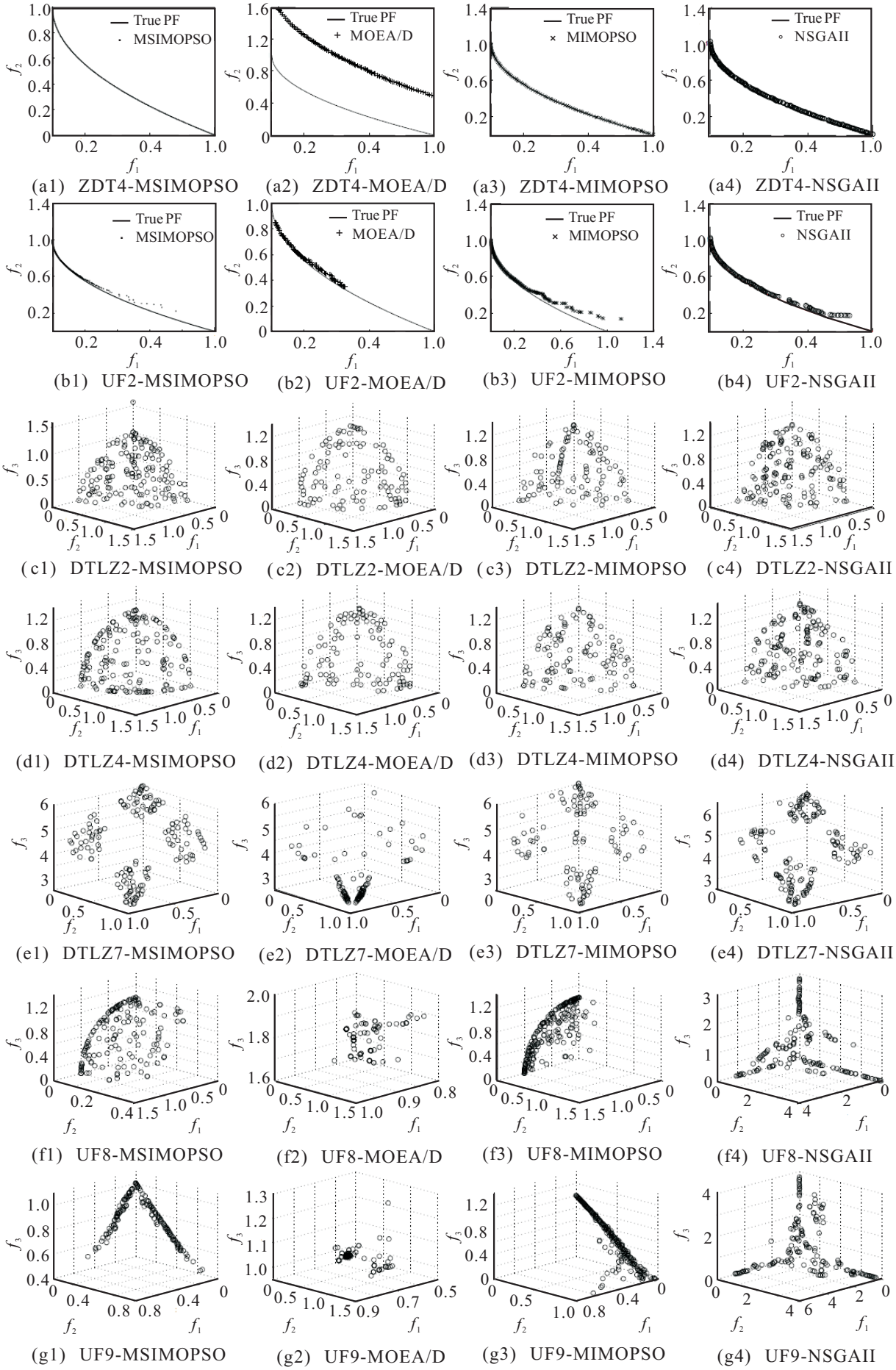

图 4 各算法部分测试函数PF图

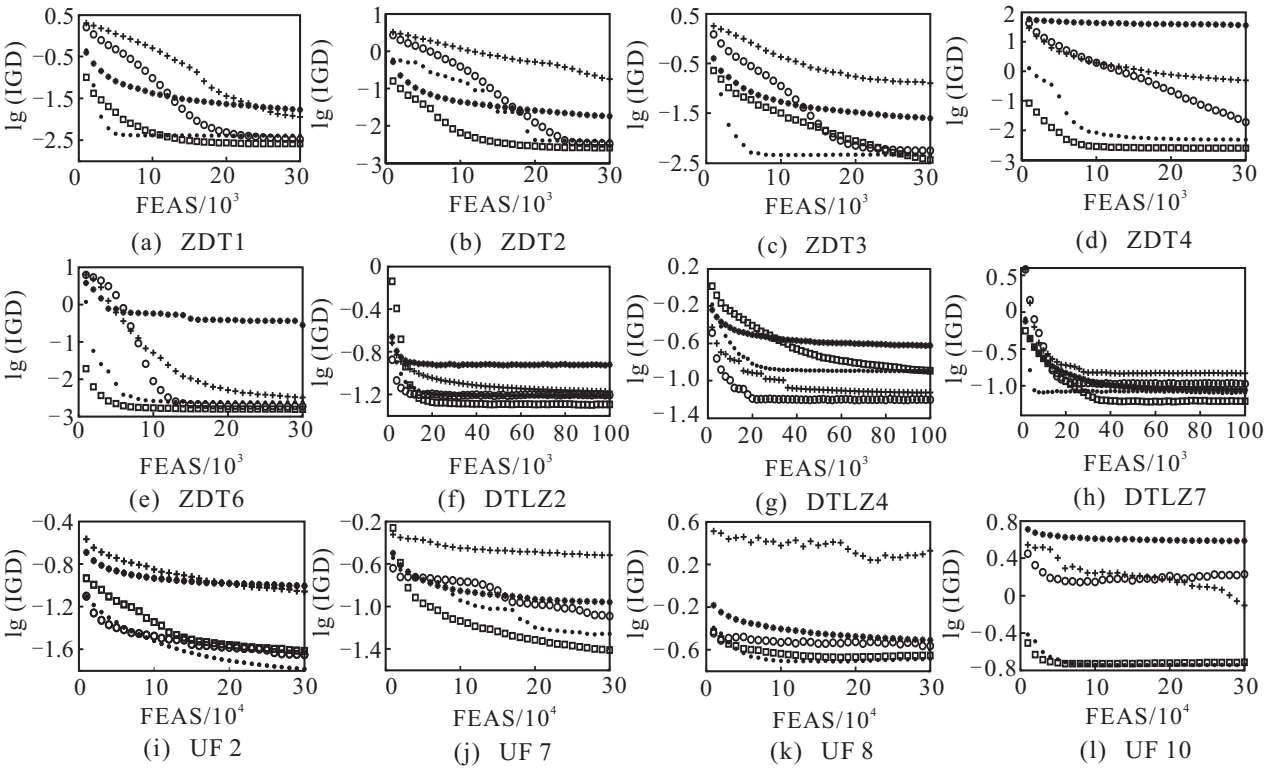

图 5　各测试函数IGD收敛图

快,而且快速趋于稳定;在测试函数DTLZ4上效果不是很好,变化缓慢. 观察图5,在UF系列函数的IGD趋势图上,MSIMOPSO除了在UF2上变化速度劣于MIMOPSO和NSGAII,在UF7, UF8和UF10上均能快速收敛,尤其在UF7上快速收敛且优于其他各个对比算法. 结合表1和图5, MSIMOPSO在13个测试函数中完全优于MOSPO,在DTLZ4劣于MIMOPSO, MOEA/D和NSGAII,在UF2上劣于MIMOPSO,其余均取得了较好的结果.

3.6　变异策略分析

变异策略在本文中起到了至关重要的作用,尤其在算法中引入了2.1.3节中的多点变异,即在位置更新加入多判据变异,在测试函数ZDT4的变异效果尤为明显. 图6绘制了加多点变异和未加变异的IGD趋势图,其中MSIMOSPO-1算法是未引入变异的算法. 可以看出,算法未引入变异时,在多峰问题ZDT4上

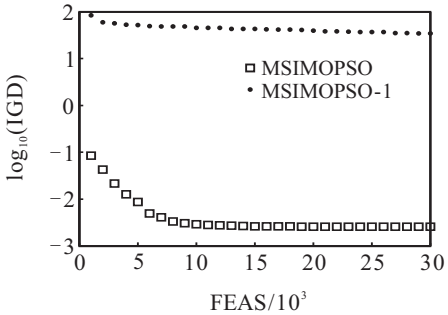

图 6　变异策略对比IGD收敛图

算法无法收敛前沿,加入变异策略后IGD值在稳定的基础上变得非常小,说明运用该策略所获得的解更加接近真实的PF,且具有更好的收敛性和稳定性.

4　结　论

为了更好地处理多峰问题和提高算法的收敛性,本文提出了多策略融合改进的多目标粒子群算法,所提出算法融合了各策略的优点,使优化的效果更佳. MSIMOPSO首先融合分解的思想,将多目标问题分解为多个子问题,降低了问题的复杂度,提高了算法的效率,更有效地保持了解的多样性. 应用邻居最优粒子的信息选择和更新全局极值,提高算法的全局性,有利于算法快速地找到全局最优解,加快收敛速度. 为了避免早熟,及时跳出局部最优,在速度和位置处增加多点变异,即多判断准则的位置变异,在多峰问题上取得了良好的收敛效果. 最后在更新后种群和最优解集存档中引入非支配排序思想,选择等级高的解存入精英外部存档,在迭代终止时将新的精英解集作为最终的输出结果,使获得的PF分布更加均匀,更加贴近真实的PF. 引用分解增加了解的多样性,再加入非支配排序增加了解集的收敛性,相互结合优势互补. 通过与对比算法在同一环境下的实验对比,验证了MSIMOPSO算法在大多数测试函数中均表现出了很好的收敛性和分布性,获得的PF分布均匀,并且更加接近真实的PF.

参考文献(References)

[1] Kennedy J. Particle swarm optimization[C]. Encyclopedia of Machine Learning. Springer, 2011: 760-766.

[2] Coello C C, Lechuga M S. MOPSO: A proposal for multiple objective particle swarm optimization[C]. Proc of the 2002 Congress on Evolutionary Computation. Honolulu: IEEE, 2002: 1051-1056.

[3] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.

[4] Ai M N, Petrovski A, Mccall J. MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces[J]. Evolutionary Computation, 2014, 22(1): 47-77.

[5] 邱飞岳,莫雷平,江波,等.基于大规模变量分解的多目标粒子群优化算法研究[J]. 计算机学报, 2016, 39(12): 2598-2613.
(Qiu F Y , Mo L P, Jiang B, et al. Multi-objective particle swarm optimization algorithm using large scale variable decomposition[J]. Chinese J of Computers, 2016, 39(12): 2598-2613.)

[6] 谢承旺,邹秀芬,夏学文,等.一种多策略融合的多目标粒子群优化算法[J]. 电子学报, 2015, 43(8): 1538-1544.
(Xie C W, Zou X F, Xia X W, et al. A multi-objective particle swarm optimization algorithm integrating multiply strtegies[J]. Acta Electronica Sinca, 2015, 43(8): 1538-1544.)

[7] Hu W, Yen G G, Luo G. Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system[J]. IEEE Trans on Cybernetics, 2016, 47(6): 1446-1459.

[8] HU M Q, WEIR J D, WU T. An augmented multi-objective particle swarm optimizer for building cluster operation decisions[J]. Applied Soft Computing, 2014, 25(C): 347-359.

[9] Kumar R S, Kondapaneni K, Dixit V, et al. Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach[J]. Computers & Industrial Engineering, 2016, 99(C): 29-40.

[10] Cheng S, Zhan H, Shu Z. An innovative hybrid multi-objective particle swarm optimization with or without constraints handling[J]. Applied Soft Computing, 2016, 47: 370-388.

[11] Kennedy J, Eberhart R. Particle swarm optimization[C]. 1995 Proc of IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.

[12] Xia X, Liu J, Hu Z. An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space[J]. Applied Soft Computing, 2014, 23(23): 76-90.

[13] Zhang Q, Li H. A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Trans on Evolutionary Computation, 2007, 11(6): 712-731.

[14] 刘衍民, 赵庆祯, 牛奔. 基于自适应动态邻居和广义学习的改进粒子群算法[J]. 计算机应用, 2010, 30(10): 2578-2581.
(Liu Y M, Zhao Q Z, Niu B. Improved particle swarm optimization based on adaptive dynamic neighborhood and generalized learning[J]. J of Computer Applications, 2010, 30(10): 2578-2581.)

[15] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. Evolutionary computation, 2000, 8(2): 173-195.

[16] Deb K, Ttiele L, Laumanns M, et al. Scalable multi-objective optimization test problems[C]. Proc of the 2002 Congress on Evolutionary Computation. Honolulu, 2002: 825-830.

[17] Zhang Q, Zhou A, Zhao S, et al. Multiobjective optimization test instances for the CEC 2009 special session and competition[R]. Essex: University of Essex, 2008.

[18] Ishibuchi H, Masuda H, Tanigaki Y, et al. Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems[C]. IEEE Symposium on Computional Int Muili-criteria Decision-making. Orlando, 2014: 170-177.

[19] 杨景明,穆晓伟,车海军,等.一种多策略改进的多目标粒子群优化算法[J]. 控制与决策, 2017, 32(3): 435-442.
(Yang J M, Mu X W, Che H J, et al. An improved multi-objective particle swarm optimization algorithm based on multiple strategies[J]. Control and Decision, 2017, 32(3): 435-442.)

[20] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.

（责任编辑：孙艺红）