一种多策略融合的多目标粒子群优化算法

谢承旺[1,2],邹秀芬[1],夏学文[2],王志杰[2]
(1.武汉大学数学与统计学院,湖北武汉 430072;2.华东交通大学软件学院,江西南昌 330013)

摘 要: 为提高多目标粒子群算法在解决复杂多目标优化问题中的整体性能,提出一种多策略融合的多目标粒子群算法.该算法采用均匀化与随机化相结合的方式初始化种群,在粒子速度更新中新增一扰动项,运用简化的 k-最近邻方法维持档案以及对档案个体赋予生存期属性并动态调整生存期值.实验结果表明,在 GD 和 SP 性能指标上,本文算法与另外 5 种对等算法在 ZDT 和 DTLZ 系列测试问题上进行对比,其表现出了总体显著性的性能优势.

关键词: 粒子群优化; 多策略融合; 多目标优化问题; 多目标粒子群优化算法
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2015)08-1538-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2015.08.011

A Multi-Objective Particle Swarm Optimization Algorithm Integrating Multiply Strategies

XIE Cheng-wang[1,2], ZOU Xiu-fen[1], XIA Xue-wen[2], WANG Zhi-jie[2]
(1. *School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei* 430072, *China*;
2. *School of Software, East China Jiaotong University, Nanchang, Jiangxi* 330013, *China*)

Abstract: In order to improve the overall performance of multi-objective particle swarm optimization algorithm (MOPSO) in solving complicated multi-objective optimization problems, a multi-objective particle swarm optimization algorithm integrating multiply strategies (MSMOPSO) was proposed in the paper. A new initialization approach of combining uniformization and randomization was adopted in the MSMOPSO. Secondly, a disturbance item was added to the particle's velocity updating formula. Thirdly, a simplified k-nearest neighbor approach was applied to preserve the diversity of external archive. Finally, every non-dominated particle in the external archive was assigned the property of lifespan and the lifespan value would be adjusted dynamically during the run of the MSMOPSO. The experimental results illustrate that the proposed algorithm significantly outperforms the other five peer competitors in terms of GD, SP on ZDT and DTLZ test instances set.

Key words: particle swarm optimization; integrating multiply strategies; multi-objective optimization problem; multi-objective particle swarm optimization algorithm

1 引言

粒子群优化算法(Particle Swarm Optimization, PSO)是由 Kennedy 和 Eberhart[1]提出的一种基于群体智能的随机优化方法,它具有结构简单、参数设置少和易于实现的特点,并且在单目标问题中表现出了收敛速度快、解题效率高的良好特性.鉴于 PSO 算法群体搜索的特性以及它在单目标问题中良好的解题性能,促使研究者尝试将 PSO 算法应用于多目标优化领域.由于多目标问题和单目标问题之间存在本质差别,使得标准 PSO 算法并不能直接用于求解多目标优化问题.为了利用 PSO 算法的良好特性解决 MOP 问题,必须根据多目标优化问题的特点,对标准 PSO 算法的模式进行适应性改进和扩展,并由此产生了一些经典的多目标粒子群优化算法(Multi-Objective Particle Swarm Optimization, MOPSO).Coello C 等[2]提出的 MOPSO 算法具有里程碑意义,他们利用自适应网格的机制对外部档案进行维护,并对粒子和粒子的飞行区域进行变异.Sierra 等[3]提出了基于拥挤距离和 ε 占优机制的多目标粒子群算法 OMOPSO.Leong 等[4]通过动态调整种群大小和采用子种群等策略从种群角度平衡算法的勘探与开采能力.由于目前大多数 MOPSO 算法只是在算法的局部,而未能从算法的整

收稿日期:2014-11-20;修回日期:2015-03-24;责任编辑:李勇锋
基金项目:国家自然科学基金(No.61165004);国家自然科学基金重大研究计划培育项目(No.91230118);江西省自然科学基金(No.20114BAB201025);江西省教育厅科技项目(No.GJJ12307, No.GJJ14373)

体上改善其性能,它们对算法的一些重要组件并未实施协同式的改进,因而使得这些算法在解决复杂 MOP 问题时尚有较大的性能提升空间.

为提高多目标粒子群算法的整体性能以更好地解决复杂 MOP 问题,本文提出一种多策略融合的多目标粒子群优化算法(multi-objective particle swarm optimization integrating multiply strategies, MSMOPSO). MSMOPSO 算法利用均匀化与随机化相结合方法初始化种群,并对粒子的飞行施加扰动.此外,该算法采用了一种简化的 k-最近邻方法维持外部档案,并对档案粒子实施一种带生存期的淘汰机制. MSMOPSO 算法将上述几种有利于全局勘探或局部开采的策略有机融合起来,其目的在于发挥各种策略的长处,进行优势互补,以改善算法在解空间搜索的效率和解题的效果,提高算法解决复杂 MOP 问题时的整体性能.

2　基本概念

不失一般性,一个具有 k 个决策变量, m 个目标函数, $(p+q)$ 个约束的 MOP 问题可以定义如下:

$$\begin{cases} \min\ \boldsymbol{F}(x)=(f_1(x),f_2(x),\cdots,f_m(x))^{\mathrm{T}}, \\ \text{s.t.}\quad g_i(x)\geq 0\quad (i=1,2,\cdots,p) \\ \qquad\ h_j(x)=0\quad (j=1,2,\cdots,q) \end{cases} \tag{1}$$

其中, $\boldsymbol{x}=[x_1,x_2,\cdots,x_k]^{\mathrm{T}}\in X\subset\mathbb{R}^k$ 是 k 维决策向量; $\boldsymbol{X}$ 表示由决策向量 $\boldsymbol{x}$ 形成的 k 维决策空间; $y=(y_1,y_2,\cdots,y_m)\in Y\subset\mathbb{R}^m$ 为 m 维的目标空间;目标函数 $\boldsymbol{F}(x)$ 定义了由决策空间向目标空间映射的函数, $g(x)$ 定义了 p 个不等式约束, $h(x)$ 定义了 q 个等式约束,约束函数 $g(x)$ 和 $h(x)$ 共同确定决策向量 $\boldsymbol{x}$ 的可行域.本文讨论连续型多目标优化问题,以下给出几个重要的概念.

定义 1　(Pareto 支配)假设 x 和 y 是上述 MOP 问题的可行解,称 x Pareto 支配 y(记作 $x\prec y$),当且仅当 $\forall i\in[1:m]:f_i(x)\leqslant f_i(y)\wedge\exists j\in[1:m]:f_j(x)<f_j(y)$ 成立.

定义 2　(Pareto 最优解)对于决策空间 $\boldsymbol{X}$ 中的任一点 x^*,若在 $\boldsymbol{X}$ 中不存在支配 x^* 的向量,则称 x^* 为 Pareto 最优解,即: $\neg\exists x\in X:x\prec x^*$.

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

定义 4　(Pareto 前沿)集合 POS 对应的目标向量的集合称为 Pareto 前沿(POF),即: $\mathrm{POF}=\{f(x^*)\mid x^*\in\mathrm{POS}\}$.

在标准 PSO 算法中,第 t 代的粒子 i 在搜索空间中的速度和位置根据式(2)和式(3)确定:

$$\begin{aligned} v_i(t+1)=\omega\cdot v_i(t)+c_1r_1(pbest_i-x_i(t)) \\ +c_2r_2(gbest_i-x_i(t)) \end{aligned} \tag{2}$$

$$x_i(t+1)=x_i(t)+v_i(t+1) \tag{3}$$

其中, ω 为惯性权重, c_1, c_2 为学习因子, r_1, r_2 为 $[0,1]$ 范围内均匀分布的随机数. $pbest_i$ 为粒子 i 自身经历的最好位置, $gbest_i$ 为粒子 i 的全局最好位置,它是整个种群所经历的最好位置.式(2)表示粒子的速度由 3 部分决定:惯性部分、认知部分和社会部分,它们共同改变粒子飞行速度.

3　MSMOPSO 算法

3.1　种群多样性的初始化策略

目前,大多数 MOPSO 算法通过随机方式产生初始群体,这种初始化方法由于存在随机误差并不能保证初始种群较好地覆盖问题的决策空间,从而较难获得多样性的初始群体.鉴于此,本文提出一种均匀化与随机化相结合的初始化方法,该方法的原理是将决策向量 x 中任一维决策变量 $x_i(i=1,2,\cdots,k)$ 均匀划分成与种群规模相等的若干等长的子区间,初始代个体在任一基因位上的取值需首先随机选定子区间,然后再在选定的子区间内产生随机值而获得.均匀划分后的各个子区间有且仅有一次产生基因值的机会,这样可在最大程度上保证初始群体基因的多样化.算法 1 给出了种群多样性的初始化算法的流程.

算法 1　种群多样性的初始化算法

输入:种群的规模 n,决策向量的维度 k,各决策变量 $x_j(j\in[1:k])$ 的区间 $[a_j,b_j]$.

输出:多样性的初始群体.

Step1 FOR $j=1$ TO k

Step2　$\triangle_j=(b_j-a_j)/n$ //将决策变量 x_j 的区间均匀划分成 n 等份.

Step3　$\Lambda_j=\{[a_j,a_j+\triangle_j],[a_j+\triangle_j,a_j+2\triangle_j],\cdots,[a_j+(n-2)\triangle_j,a_j+(n-1)\triangle_j],[a_j+(n-1)\triangle_j,b_j]\}$

Step4　FOR $i=1$ TO n

Step5　从集合 Λ_j 中随机选择一个子区间,并在该子区间内随机产生一个基因值赋给 x_j^i

Step6　更新集合 Λ_j:从集合 Λ_j 中删除在 Step5 中已选中的子区间

Step7　END FOR

Step8　END FOR

Step9 输出多样性的初始种群 $\{x^1,x^2,\cdots,x^n\}$.

这种初始化方法既保持了选取子区间以及从子区间中产生基因值的随机性,又保证了决策变量区间划分的均匀性,算法执行的结果可获得一组在决策空间中均匀分布的初始解点,这就为算法的搜索提供了一个良好的开端.

3.2　增加扰动的粒子速度更新策略

与现有大多数 MOPSO 算法所采用的变异策略所不同,本文提出增加扰动的粒子速度更新策略,其原理如

下:假设在 k 维搜索空间中,粒子 i 搜索至第 t 代时的个体历史最优位置为 $P_i^t=(p_{i,1}^t,p_{i,2}^t,\cdots,p_{i,k}^t)$,搜索至第 t 代时整个粒子群的全局最优位置为 $P_g^t=(p_{g,1}^t,p_{g,2}^t,\cdots,p_{g,k}^t)$,而 $P_d^t=(p_{d,1}^t,p_{d,2}^t,\cdots,p_{d,k}^t)$则是从第 t 代粒子群中随机选出的扰动粒子 P_d^t,并要求 P_d^t 支配当前粒子 i 或者与粒子 i 彼此非劣,则在第$(t+1)$代中,粒子 i 的第$j(j\in[1:k])$维的速度和位置的更新公式可做如下定义:

$$v_{i,j}^{t+1}=\omega\cdot v_{i,j}^t+c_1\cdot r_1\cdot(p_{i,j}^t-x_{i,j}^t)+c_2\cdot r_2\cdot(p_{g,j}^t-x_{i,j}^t)+c_3\cdot r_3\cdot(p_{d,j}^t-x_{i,j}^t) \tag{4}$$

$$x_{i,j}^{t+1}=x_{i,j}^t+v_{i,j}^{t+1} \tag{5}$$

这里的 ω 为惯性权重,衡量着前一时刻的速度对下次移动的影响,c_1和 c_2为学习因子,c_3为扰动因子,r_1,r_2为[0,1]范围内均匀分布的随机数,r_3为[-1,1]区间内的均匀随机数,式(4)右边的第 4 项为粒子速度更新中新增的扰动项.

本文采用的 ω 动态更新公式如下:

$$\omega^t=(\omega_{\max}-\omega_{\min})\times(T-t)/T+\omega_{\min} \tag{6}$$

其中,t 为当前迭代次数,T 为最大迭代次数,$\omega_{\max}$为初始惯性权重,$\omega_{\min}$为粒子群在最大迭代次数时的惯性权重,根据文献[5]的研究,本文取 $\omega_{\min}=0.4$,$\omega_{\max}=0.9$.

3.3 简化的多样性保持策略

在使用渐进方法决定解个体进入档案的过程中,档案成员的多样性不断增强,而渐进方法所采用的个体密度估计方法对档案的维护和解群多样性的保持都很重要.在目前已定义的密度估计方法中,SPEA2 算法[6]的 k-最近邻方法在保持解群的多样性方面表现突出,该算法将个体的密度值定义为个体与它的第 k 个最邻近个体间的距离,并且取 $k=\sqrt{N+N'}$,其中 N 为种群的规模,N'为外部档案的最大规模.

在 SPEA2 运行过程中,如果档案规模达到预设的最大值时,档案中每加入一个新解则要从中移出一个个体,比如个体 A,个体 A 应满足:对于档案中所有个体 B,$A<_dB$ 成立.这里的 $A<_dB$ 定义为当且仅当对于所有的 $l=1,2,\cdots,N'$:$D_{lA}=D_{lB}$;或者对于某个 l,$D_{lA}<D_{lB}$,而对任意的 $l'<l$:$D_{l'A}=D_{l'B}$.这里的 $D_{lA}(D_{lB})$为个体 $A(B)$与它的第 l 个最邻近个体的距离.不难看出,这种解个体密度的估计过程比较复杂,而且需要对所有的个体和所有的 $l=1,2,\cdots,N'$ 比较个体间 D_l 上的差异,时间开销很大.

鉴于此,本文提出一种简化的 k-最近邻方法,与原始的 k-最近邻方法相比,简化后的方法取 $k=\log(N+N')$,时间开销要小于原始方法.简化的档案个体密度估计方法可定义如下:$A<_dB$ 当且仅当对于 $l=1,2,\cdots,k$:$D_{lA}=D_{lB}$;或者对于某个 $l\leqslant k$:$D_{lA}<D_{lB}$,而对任意的 $l'<l$:$D_{l'A}=D_{l'B}$.由于简化的 k 值为 $\log(N+N')$,其阶要低于原始的 k 值$\sqrt{N+N'}$,因此,简化后的方法在时间开销上有所改善.

3.4 带生存期的全局最优位置的淘汰机制

本文尝试将生存期(lifespan)的概念引入到 MOPSO 算法中,并对档案中的非劣个体赋予生存期属性,该属性刻画了档案成员充当全局最优位置角色引导粒子寻优的存活周期,并用进化的代数表征存活期的长短,即生存期值.带生存期的全局最优位置的淘汰机制的基本思路是:如果粒子群中某粒子 i 选中档案中某非支配个体 j 作为全局最优解,则成员 j 被赋予一个初始的生存期值 L_0,一旦当 j 粒子的生存期值 L 减至预设的最小阈值 $L_{\min}$,甚至更小时,j 的存活期结束,进而从档案中重新为粒子 i 随机选择一个非劣解作为它的全局最优位置并引导粒子 i 继续飞行.这里的生存期值需要根据粒子间的支配关系进行调整,例如对于档案个体 j,其生存期值将按照以下 3 种情况进行调整:

(1)如果粒子 i 的试验粒子 i'(i'由式(4)和式(5)产生)与 i 的历史最优位置之间相互非支配,则当前的全局最优解 j 的生存期值 L 减少△$L/2$;

(2)如果粒子 i 的试验粒子 i'被 i 的历史最优位置所支配,则当前的全局最优解 j 的生存期值 L 减少△L;

(3)如果粒子 i 的试验粒子 i' 支配 i 的历史最优位置,则当前的全局最优解 j 的生存期值 L 增加△L.

以上对生存期值的调整步长依公式△$L=(L_0-L_{\min})/l$ 执行,此外,由本文的调参实验表明,$l=5$ 时通常能获得较好的效果.

3.5 MSMOPSO 算法流程

在前面 3.1 至 3.4 小节的基础上给出本文算法 MSMOPSO 的流程,如算法 2 所示.

算法 2 多策略融合的多目标粒子群优化算法 MSMOPSO

输入: 种群规模 N,外部档案最大规模 N',多项式变异的分布指数 η_m,变异概率 p_m,学习因子 c_1,c_2和 c_3,最大迭代次数 $T_{\max}$,档案成员初始生存期值 L_0,最小生存期阈值 $L_{\min}$.

输出: 最末代外部档案中所有解.

Step1 种群初始化:利用算法 1 的初始化策略产生初始粒子群,并置粒子速度 $v=0$,粒子的历史最优位置设为粒子本身;外部档案初始化为空,将初始粒子群中非支配粒子复制到档案中,并利用 3.4 小节的规则对档案个体赋予生存期值;设迭代计数器 $t=0$.

Step2 利用 3.2 小节的增加扰动的粒子速度更新,计算粒子的适应度,采用基于 Pareto 支配的关系确定粒子自身的历史最优位置,并运用 3.4 小节的带生存期的全局最优位置淘汰机制更新粒子的全局最优位置.

Step3　如果粒子的生存期值 L 小于 L_{min},则随机地从档案中选出一个非支配粒子作为粒子的全局最优位置.

Step4　对粒子群中所有的粒子以一定的概率执行多项式变异.

Step5　选择非支配粒子进入外部档案,并利用简化的多样性保持策略维护外部档案.

Step6　终止条件判定:如果 $t < T_{max}$, $t = t + 1$,并转至 Step2;否则执行 Step7.

Step7　输出外部档案中所有的解.

假定优化问题的目标数目为 M,搜索空间的维度为 K,粒子群的规模为 N,外部档案最大容量为 N',则 MSMOPSO 算法的时间复杂度可估计如下:(1)Step1 中生成多样性的初始群体的时间为 $O(KN)$,粒子飞行速度的初始化所需时间为 $O(KN)$,粒子历史最优位置的初始化时间为 $O(N)$,从初始种群中选择非支配粒子进入外部档案的时间为 $O(MN^2)$,为档案中的粒子赋予生存期值的时间为 $O(N')$,因此 Step1 总的时间复杂度为 $O(MN^2)$;(2) Step2 中更新粒子的时间复杂度为 $O(KN)$,计算种群中粒子适应度的时间为 $O(MN)$,对粒子历史最优位置更新的时间亦为 $O(MN)$,而且在一般的优化问题中存在 $K > M$,因此 Step2 中总的时间复杂度为 $O(KN)$;(3)Step3 中从外部档案中随机选择全局最优解所需的时间为 $O(MN')$;(4)Step4 中对粒子群中所有粒子执行变异操作的时间复杂度为 $O(KN)$;(5) Step5 中选择非支配粒子加入档案,并对档案进行维护的时间复杂度为 $O(M(N+N')^2\log(N+N'))$;(6)Step7 中输出档案中所有解个体的时间为 $O(N')$. 因此,循环迭代过程的时间复杂度为 $O(KN) + O(MN') + O(KN) + O(M(N+N')^2\log(N+N')) = O(M(N+N')^2\log(N+N'))$,而整个迭代过程所需的时间为 $O(T_{max}M(N+N')^2\log(N+N'))$.

综上,MSMOPSO 算法的时间复杂度应为 $O(MN^2) + O(T_{max}M(N+N')^2\log(N+N')) + O(N') = O(T_{max}M(N+N')^2\log(N+N'))$. 由此可见,MSMOPSO 算法的时间复杂性取决于对外部档案进行维护的复杂性.

4　实验与结果分析

为了比较本文算法 MSMOPSO 的性能,选取 5 种常用于比较多目标优化的对等算法,这些算法可分为两组,其中一组为多目标粒子群优化算法,包括:(1)Coello C 等[2]提出的具有里程碑意义的多目标粒子群算法 MOPSO,为了区别多目标粒子群算法的简写,这里将该算法标识为 CMOPSO;(2)Sierra 等[3]提出的改进型多目标粒子群算法 OMOPSO. 另一组为代表性多目标遗传进化算法,包括:(1)Zitzler 等[6]提出的改进型强度 Pareto 进化算法 SPEA2;(2)Deb 等[7]在原始 NSGA 算法的基础上改进的 NSGA-II 算法;(3)Zhang 和 Li[8]将传统的数学规划方法与进化算法结合起来的一种基于分解的多目标进化算法 MOEA/D. 本文选取的 5 个对比算法的实验参数都按照对应的参考文献设置.

测试函数集由 ZDT[9]和 DTLZ[10]两个系列函数组成. 其中,ZDT 系列是 2-目标的测试问题,DTLZ 系列是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-目标优化测试问题集. 这些测试问题的函数表达式、决策变量数目和决策变量的取值范围等采用相应的参考文献的设置. 本文采用的测试函数的 Pareto 前沿具有不同的特征,构成了不同的难度类型,是检验多目标优化算法解题性能的基准测试问题. 此外,这些测试函数的 Pareto 最优解集在理论上均是已知的,这就为算法的性能评估提供了极大的便利.

本文采用 Van Veldhuizen 与 Lamont[11]提出的世代距离(Generational Distance,GD)评价算法的收敛性,采用 Schott[12]提出的空间评价方法(Spacing,SP)来评价解集的分布性.

比较实验中各算法在 2-目标测试问题中的种群规模均设为 $N = 100$,全局外部档案的最大容量均设为 $N' = 100$,所有测试函数的最大评估次数均设为 50000,所有算法的最大迭代次数 $T_{max} = 500$;3-目标测试问题中种群的大小均设为 $N = 200$,全局外部档案的最大容量均设为 $N' = 200$,所有测试函数的最大评估次数均为 200000,所有算法的最大迭代次数为 $T_{max} = 2000$. 此外,MSMOPSO 算法中 L_0在不同测试函数中均取值为 2.5, L_{min}均取值为 0. 为减少性能分析中随机因素的影响,每种算法在所有的测试函数上均独立运行 30 次. 本文的仿真实验是在 Think Pad X200 笔记本电脑上运行,电脑配置 5G 内存和 2.4GHz 双核 CPU,安装 Windows 7 X64 操作系统,算法运用C++ 编程实现,并利用 Matlab 2010a 环境出图.

表 1 列出了 6 个对等比较算法在 7 个测试问题上的 GD 指标的均值(Mean)和标准差(Std.),这些值的获得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 GD 值在显著水平为 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 7 个测试问题中的净胜得分,即得“+”与得“-”的测试问题个数之差. 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 GD 值.

从表 1 可以看出,MSMOPSO 在 7 个测试问题中获

得了 3 个最优的 GD 值,MOEA/D 亦获得了 3 个最好的 GD 值,NSGA-II 算法获得 1 个最好的 GD 值,其他算法未能获得最优的 GD 值.表 1 的 t-检验结果反映了本文算法对其他 5 种对比算法的净胜分均为正数,其中 MSMOPSO 对比 OMOPSO 和 SPEA2 两个算法时的净胜分为 7,对比 NSGA-II 为 5,对比 MOEA/D 为 3,对比 CMOPSO 为 1.这表明本文算法在所有测试问题中获得的总体 GD 性能要显著优于另外 5 种对比算法.当然,根据没有免费午餐定律,不能期望本文算法在每一个测试函数上都能获得最好的 GD 值.由于 GD 性能能够反映算法的收敛性,因此,从收敛性度量来看,MSMOPSO 的表现较为突出.

表 1 6 种对等算法在 7 个测试实例上的 GD 性能对比结果

Test instance		MSMOPSO	CMOPSO	OMOPSO	NSGA-II	MOEA/D	SPEA2
ZDT1	Mean	1.073e-4	1.087e-4	1.379e-4	3.044e-4	**6.628e-5**	2.649e-4
	Std.	2.476e-5	3.780e-5	2.408e-5	1.469e-4	4.363e-6	1.638e-5
	t-test		-1.111e-1	-3.812e0	-4.025e0	2.821e1	-2.886e1
			-	+	+	+	+
ZDT2	Mean	9.028e-5	7.198e-5	9.352e-5	1.755e-4	**6.060e-5**	2.382e-4
	Std.	1.837e-5	7.741e-6	3.962e-6	4.104e-5	1.564e-6	2.922e-5
	t-test		7.921e0	-2.453e0	-6.230e0	5.693e1	-1.519e1
			-	+	+	-	+
ZDT3	Mean	**3.535e-4**	3.549e-4	3.622e-4	3.887e-4	3.665e-4	4.274e-4
	Std.	1.502e-5	3.029e-5	1.564e-5	1.946e-5	1.398e-6	6.879e-5
	t-test		-1.387e-1	-1.669e0	-5.427e0	-2.790e1	-3.223e0
			+	+	+	+	+
ZDT4	Mean	**1.379e-4**	1.519e-1	6.974e-1	6.494e-4	1.779e-4	1.652e-3
	Std.	2.804e-5	6.466e-2	3.888e-1	2.864e-4	8.604e-5	1.320e-3
	t-test		-7.041e0	-5.380e0	-5.358e0	-1.394e0	-3.441e0
			+	+	+	+	+
DTLZ1	Mean	4.339e-4	9.616e-1	3.198e-1	**3.530e-4**	7.153e-4	2.400e-3
	Std.	1.986e-4	8.020e-1	3.530e-4	1.408e-4	1.943e-5	2.500e-3
	t-test		-3.595e0	-2.714e3	1.724e0	-4.345e1	-2.359e0
			+	+	-	+	+
DTLZ2	Mean	8.316e-4	7.830e-4	9.402e-4	8.471e-4	**6.707e-4**	1.600e-3
	Std.	3.047e-5	2.566e-5	8.643e-5	3.302e-5	1.700e-6	1.202e-4
	t-test		5.682e0	-3.770e0	-1.408e0	2.839e3	-1.918e1
			-	+	+	-	+
DTLZ3	Mean	**5.354e-4**	7.563e-4	2.043e1	8.534e-4	1.51e+3	5.170e-2
	Std.	1.780e-2	4.156e-5	4.658e0	7.339e-5	5.838e1	7.630e-2
	t-test		-1.595e1	-1.316e1	-1.300e1	-7.760e1	-2.012e0
			+	+	+	+	+
Better(+)			4	7	6	5	7
Same(=)			0	0	0	0	0
Worse(-)			3	0	1	2	0
Score			1	7	5	3	7

表 2　6 种对等算法在 7 个测试实例上的 SP 性能对比结果

Test instance		MSMOPSO	CMOPSO	OMOPSO	NSGA-II	MOEA/D	SPEA2
ZDT1	Mean	**1.386e-3**	9.191e-3	1.441e-3	7.371e-3	9.744e-3	3.179e-3
	Std.	2.773e-4	6.181e-4	1.646e-4	6.641e-4	7.187e-5	3.176e-4
	t-test		-3.788e1	-1.002e0	-2.704e1	-3.489e2	-1.694e1
			+	+	+	+	+
ZDT2	Mean	**1.481e-3**	1.047e-2	1.612e-3	7.688e-3	4.277e-3	3.251e-3
	Std.	2.681e-4	1.527e-3	1.179e-4	8.206e-4	9.027e-5	3.747e-4
	t-test		-1.766e1	-3.333e0	-2.269e1	-9.292e1	-1.417e1
			+	+	+	+	+
ZDT3	Mean	3.132e-3	9.501e-3	**2.999e-3**	7.980e-3	2.295e-2	3.791e-3
	Std.	3.782e-4	9.841e-4	2.476e-4	7.967e-4	6.594e-5	6.033e-4
	t-test		-1.942e1	1.612e0	-1.826e1	-9.016e2	-3.277e0
			+	-	+	+	+
ZDT4	Mean	**1.517e-3**	6.339e-3	2.832e-2	7.398e-3	9.992e-3	1.210e-2
	Std.	2.480e-4	7.491e-3	3.195e-2	9.061e-4	1.984e-4	1.173e-2
	t-test		-1.931e0	-3.517e0	-1.947e1	-1.282e2	-2.707e0
			+	+	+	+	+
DTLZ1	Mean	**5.317e-3**	1.032e0	1.546e0	9.749e-3	2.499e-2	2.203e-2
	Std.	2.239e-3	1.043e0	3.378e-1	3.917e-4	3.963e-5	2.973e-2
	t-test		-2.953e0	-1.368e1	-3.394e1	-1.489e3	-1.687e0
			+	+	+	+	=
DTLZ2	Mean	2.734e-2	**1.485e-2**	2.525e-2	2.577e-2	3.820e-2	1.671e-2
	Std.	1.298e-3	2.188e-3	1.095e-3	1.535e-3	1.565e-4	1.820e-3
	t-test		1.713e1	5.726e0	3.068e0	-2.082e2	1.752e1
			-	-	-	+	-
DTLZ3	Mean	1.682e-1	**1.351e-2**	1.086e1	2.619e-2	8.18e+2	7.063e-1
	Std.	2.314e-1	5.866e-3	4.529e0	6.745e-4	3.229e1	1.079e0
	t-test		7.911e1	-7.082e0	6.316e2	-7.598e1	-1.496e0
			-	+	-	+	=
Better(+)			5	5	5	7	4
Same(=)			0	0	0	0	2
Worse(-)			2	1	2	0	1
Score			3	4	3	7	3

表 2 给出了 6 种对等比较算法在 7 个多目标测试问题上的 SP 性能的对比结果.从表 2 可以看出,MSMOPSO 在 7 个测试函数上获得了 4 个最优的 SP 值,CMOPSO 获得了 2 个最优的 SP 值,OMOPSO 获得了 1 个最优的 SP 值,而 NSGA-II、MOEA/D 和 SPEA2 未能获得最优的 SP 值.这从总体上表明,本文算法在 7 个测试问题上具有最好的 SP 性能.

在表 2 的 *t*-检验结果中,本文算法对比其他 5 种算法的净胜得分均为正数,其中 MSMOPSO 对比 MOEA/D 的净胜得分为 7,对比 OMOPSO 的净胜得分为 4,对比 CMOPSO、NSGA-II 和 SPEA2 三种算法的净胜得分均为 3.这表明 MSMOPSO 算法在所有测试问题中的 SP 性能总体上显著优于另外 5 种比较算法.

由于 SP 指标能够反映算法所获解集分布性,因此,从解群的分布性来看,MSMOPSO 算法在 6 个对等比较算法中是最好的.究其原因,MSMOPSO 运用了种群多

样性的初始化策略为算法提供了多样性的初始迭代点;其次,MSMOPSO 对全局最优解赋予生存期属性,并依据其引导效果动态调整生存期值,这种机制能有效防止粒子间聚集,改善了解群的分布性;第三,简化的 k-最近邻策略改善了算法的时间复杂性,同时亦能有效地维持档案群体的多样性.这些策略有机结合显著地提高了解群的分布性能.

综上,MSMOPSO 算法在所有测试函数中获得了最好的收敛性和多样性的折中效果,表明了本文多种策略融合的模式能够更好地平衡算法的勘探与开采的进化过程,进而发挥了各种策略协同改进的整体优势,并在复杂多目标优化的测试问题中取得了明显的效果.

5 结论

论文提出一种多策略融合的多目标粒子群优化算法 MSMOPSO,以集成各种策略的长处,进行优势互补.MSMOPSO 使用均匀化与随机化相结合的初始化方法,在粒子速度更新中新增了扰动项,运用简化的 k-最近邻方法维持档案等.实验结果表明了这种融合多种策略的 MSMOPSO 算法的整体性能明显优于其他几种对比算法,由此表明了多策略融合模式的有效性,同时也表明了 MSMOPSO 是一种颇具前途的多目标粒子群算法.

未来将利用更多、更难的测试函数以及一些工程实践中的 MOP 问题较全面地检验 MSMOPSO 算法的性能,并进一步考察 MSMOPSO 算法各组件之间的协合作用,寻找更有效的多策略融合模式,不断提高算法在解决复杂多目标优化问题中的整体性能.

参考文献

[1] Kennedy J, Eberhart R C. Particle swarm optimization[A]. Proc of the IEEE International Conference on Neural Networks[C]. Piscataway: IEEE Press, 1995. 1942 – 1948.

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

[3] Sierra M R, Coello Coello C A. Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance[A]. Proceedings of 3rd International Conference on Evolutionary Multi-criterion Optimization [C]. Berlin: Springer, 2005. 505 – 519.

[4] Yen GG, Leong WF. Dynamic multiple swarms in multiobjective particle swarm optimization[J]. IEEE Transacations on System, Man, Cybernetics, Part A, 2009, 39(4): 890 – 911.

[5] Xuewen Xia, Jingnan Liu, Zhongbo Hu. An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space[J]. Applied Soft Computing, 2014, 23: 76 – 90.

[6] Zitlzer E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[A]. Proceedings of International Conference on Evolutionary Method for Design, Optimization and Control with Applications to Industrial Problems[C]. Berlin: Springer, 2002. 95 – 100.

[7] Deb K, Pratab A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182 – 197.

[8] Qingfu Zhang, Hui Li. MOEA/D: A multi-objective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712 – 731.

[9] Zitler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8: 173 – 195.

[10] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems[A]. Proc of the IEEE Congress on Evolutionary Computation (CEC 2002)[C]. Piscataway: IEEE Service Center, 2002. 825 – 830.

[11] Van Veldhuizen D A, Lamont G B. On measuring multi-objective evolutionary algorithm performance[A]. Congress on Evolutionary Computation[C]. Piscataway: IEEE Press, 2000. 204 – 211.

[12] Schoot J R. Fault tolerant design using single and multicriteria genetic algorithm optimization[D]. Massachusetts: Cambridge Institute of Technology, 1995.

作者简介

谢承旺 男,1974 年出生,湖北武汉人.副教授、硕士生导师、中国计算机学会高级会员.2005 年、2010 年分别在武汉理工大学、武汉大学获得工学硕士和工学博士学位.现为武汉大学数学与统计学院博士后,主要从事智能计算与智能信息处理方面的研究工作.
E-mail: chengwangxie@163.com

邹秀芬 女,1966 年出生,湖北云梦人.教授、博士生导师.1986 年、1991 年和 2003 年分别在武汉大学获理学学士、理学硕士和工学博士学位.主要从事并行与智能计算方面的研究工作.
E-mail: xfzou@whu.edu.cn

夏学文 男,1974 年出生,湖北孝感人.副教授、中国计算机学会会员.主要从事智能计算及其应用方面的研究.
E-mail: laughkid@163.com

王志杰 男,1989 年出生,山西朔州人.硕士研究生.主要从事智能计算方面的研究.
E-mail: wzjwhtur@126.com