

DOI: 10.11992/tis.201711015

网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.TP.20180426.1712.011.html>

基于目标空间分解和连续变异的多目标粒子群算法

钱小宇^{1,2}, 葛洪伟^{1,2}, 蔡明³

(1. 江南大学 轻工过程先进控制教育部重点实验室, 江苏 无锡 214122; 2. 江南大学 物联网工程学院, 江苏 无锡 214122; 3. 江南大学 信息化建设与管理中心, 江苏 无锡 214122)

摘要: 针对当前多目标粒子群优化算法收敛性和多样性不佳等问题, 提出了一种基于目标空间分解和连续变异的多目标粒子群优化算法。利用目标空间分解方法将粒子群分配到预先设定好的子区域中, 在该过程中, 通过一种新适应值公式来对每个子区域中的粒子进行择优筛选, 该适应值公式融入了支配强度因素; 在全局搜索过程中, 使用差分变异、高斯变异和柯西变异对全局引导粒子的位置进行连续变异操作。将该算法与当前主流的一些多目标优化算法进行对比实验, 结果表明, 本文提出的算法在提高粒子收敛性的同时, 多样性也得到了提升。

关键词: 多目标优化; 粒子群优化算法; 分解; 子区域; 变异; 差分; 高斯变异; 柯西变异

中图分类号: TP391.4 **文献标志码:** A **文章编号:** 1673-4785(2019)03-0464-07

中文引用格式: 钱小宇, 葛洪伟, 蔡明. 基于目标空间分解和连续变异的多目标粒子群算法[J]. 智能系统学报, 2019, 14(3): 464-470.

英文引用格式: QIAN Xiaoyu, GE Hongwei, CAI Ming. Decomposition and continuous mutation-based multi-objective particle swarm optimization[J]. CAAI transactions on intelligent systems, 2019, 14(3): 464-470.

Decomposition and continuous mutation-based multi-objective particle swarm optimization

QIAN Xiaoyu^{1,2}, GE Hongwei^{1,2}, CAI Ming³

(1. Ministry of Education Key Laboratory of Advanced Process Control for Light Industry, Jiangnan University, Wuxi 214122, China; 2. School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China; 3. Information Construction and Management Center, Jiangnan University, Wuxi 214122, China)

Abstract: In light of the poor convergence problems and the diversity of current multi-objective optimization algorithms, in this paper, we propose an objective-space decomposition and continuous mutation-based multi-objective particle-swarm-optimization algorithm. Its innovations are as follows: we use a space decomposition method to distribute the particle swarm into a predefined sub-region. During this process, we apply a new adaptive value formula to select and filter the particles in each sub-region and incorporate a fitness formula into the dominance factor. In the global search process, we apply differential, Gaussian, and Cauchy mutations to continuously mutate the position of the global guide particle. We compare the performance of this algorithm with those of current multi-objective optimization algorithms, and the results show that the proposed algorithm improves the convergence and diversity of the particles.

Keywords: multi-objective optimization; particle swarm optimization algorithm; decomposition; sub-region; mutation; differential; Gaussian mutation; Cauchy mutation

收稿日期: 2017-11-13. 网络出版日期: 2018-04-27.

基金项目: 江苏省普通高校研究生科研创新计划项目 (KYLX16_0781, KYLX16_0782); 江苏高校优势学科建设工程资助项目 (PAPD).

通信作者: 葛洪伟. E-mail: ghw8601@163.com.

粒子群算法自 1995 年被 Kennedy 和 Eberhart 提出后^[1], 由于其简单高效, 收敛速度快, 逐渐在优化算法中脱颖而出。随之粒子群算法被 Coelloran 应用到多目标优化问题上^[2-3], 得到了各界的

广泛认可。后来更多学者对多目标粒子群算法(MOPSO)进行了更深入的研究: Raquel等^[4]使用了拥挤距离作为适应值对粒子进行排序; Li等^[5]通过整合粒子的GMR(global margin ranking)和粒子密度信息的方法来对粒子进行排序,能高效快速地选择出 p_{best} 和 g_{best} 粒子;除了对排序选择的方法进行优化外,有的利用聚合函数来对多个目标函数进行处理^[6];有的使用了目标空间分解的方法来保障粒子最优解的多样性^[7];以及将MOPSO和其他优化算法进行交叉使用,如教与学方法^[8]、差分方法^[9]等,通过一定的比例调用MOPSO。

上述的这些方法不断地对MOPSO的多样性和全局收敛进行改进和优化,虽然取得了较好的研究成果,但是这些方法的改进策略只对MOPSO某一性能效果进行改进,其他的性能指标并没有同时得到很好的改进。例如:文献^[7]中,多样性方面得到很大的提升,但是其收敛性方面仍有很大不足。所以,本文借鉴Pareto支配强度^[10]和连续变异的方法^[11]对基于目标空间分解方法的多目标粒子群算法^[7]进行优化,在给每个子区域分配粒子时,删除没有归属的粒子,给没分到粒子的区域初始化新的粒子,增加获得较优粒子的可能性。这样既从整体和局部两方面提升了粒子的收敛性,同时多样性也得到了一定的优化提升,这种新的算法被称为基于目标空间分解和连续变异的多目标粒子群优化算法(multi-objective particle swarm optimization based on decomposition and continuous mutation, MOPSO/DC)。

1 多目标优化问题

1.1 基本概念

$$\begin{cases} \min y = F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s.t. } g_i(x) \leq 0, i = 0, 1, 2, \dots, q \\ h_j(x) = 0, j = 1, 2, \dots, p \end{cases} \quad (1)$$

式中: $x = (x_1, x_2, \dots, x_n)$ 为 n 维决策变量; m 为目标函数的个数; $g(x)$ 函数为目标函数的 q 个不等式约束; $h(x)$ 为目标函数的 p 个等式约束。所有这些满足条件的决策变量用集合 Ω 表示, $Y = \{F(x) | x \in \Omega\}$ 为目标空间。接下来介绍4个关于多目标问题的重要定义。

定义1 Pareto 支配。解 $d, e \in \Omega$, d 支配 e , 记为 $d > e$, 满足下面的两个关系式:

$$\begin{cases} \forall i \in \{1, 2, \dots, m\}, f_i(d) \leq f_i(e) \\ \exists i \in \{1, 2, \dots, m\}, f_i(d) < f_i(e) \end{cases} \quad (2)$$

定义2 Pareto 最优。如果 x 是 Pareto 最优解, 则在 Ω 中, $\neg \exists z \in \Omega$ 使 $z > x$ 成立。

定义3 Pareto 最优解集 (PS):

$$PS = \{x \in \Omega | \neg \exists z \in \Omega, z > x\} \quad (3)$$

定义4 Pareto 前沿 (PF):

$$PF = \{F(x) | x \in PS\} \quad (4)$$

1.2 粒子群优化

粒子群算法中的粒子由速度信息和位置信息组成, 更新公式为

$$V_k^{t+1} = WK_k^t + C_1 R_1 (P - X_k^t) + C_2 R_2 (G - X_k^t) \quad (5)$$

$$X_k^{t+1} = V_k^{t+1} + X_k^t \quad (6)$$

式中: k 指粒子群中第 k 个粒子; t 为当前迭代次数; W 为权衡局部搜索和全局搜索的参数, $W \in [0.1, 0.9]$; C_1 和 C_2 为学习因子, 其大小都为 2; R_1 和 R_2 都是 $[0, 1]$ 之间的随机数; P 指当前粒子最好位置 p_{best} ; G 指引导粒子 g_{best} 的位置; V_k^t 指在第 t 次迭代中第 k 个粒子的速度; X_k^t 指在第 t 次迭代中第 k 个粒子的位置。

2 MOPSO/DC 算法

2.1 目标空间分解

把目标空间 Y 分成 M 个子区域 Y_1, Y_2, \dots, Y_M 。令 $j \in \{1, 2, \dots, M\}$, 对任一给定的第 j 个子区域, 每个目标函数在所有目标函数中所占有的权重 $a_i (i = 1, 2, \dots, m, \text{且} \sum_{i=1}^m a_i = 1)$ 所组成的向量 (a_1, a_2, \dots, a_m) 定义为该子区域的中心向量 A_j 。当目标函数个数 $m = 2$ 时, 则第 j 子区域的中心向量 A_j 表示为 $\left[\frac{j-1}{M-1} \quad 1 - \frac{j-1}{M-1} \right]$; 当 $m = 3$ 时, 进行两层循环, 令 k_1 为第 1 层 (外层) 循环变量, k_2 为第 2 层 (内层) 循环变量, k_1 从 0 取到 h , k_2 从 0 取到 $h - k_1$, 每次循环令 $u = k_2 + 1 + \sum_{k_1=1}^h (h + 2 - k_1)$, 其中 h 和 M 满足: 当 $C_{h+2}^2 \geq M$ 时 h 取最小值, 则第 u 个子区域的中心向量为 $\left[\frac{k_1}{h} \quad \frac{k_2}{h} \quad 1 - \frac{k_1}{h} - \frac{k_2}{h} \right]$; 当 $m > 3$ 时, 进行 $m - 1$ 层循环, k_1 为第一层循环变量 (最外层), 依次由外向内, 令 k_i 为第 i 层循环变量, k_{m-1} 为第 $m - 1$ 层 (最内层) 循环变量, k_1 从 0 取到 h , k_2 从 0 取到 $h - k_1$, k_i 从 0 取到 $(h - k_1 - k_2 - \dots - k_i)$, 最里层循环变量 k_{m-1} 从 0 取到 $(h - k_1 - k_2 - \dots - k_{m-2})$, 每次循环令

$$u = \sum_{k_1=0}^h \sum_{k_2=0}^{h-k_1} \dots \sum_{k_{m-3}=0}^{h-k_1-\dots-k_{m-4}} \sum_{k_{m-2}=0}^{h-k_1-\dots-k_{m-4}} \left[\sum_{k_{m-1}=1}^{h-k_1-\dots-k_{m-4}} (h + 2 - \sum_{l=1}^{m-2} k_l) + k_{m-1} + 1 \right]$$

其中参数 h 、 m 和子区数 M 满足关系: h 取满足 $C_{h+m-1}^{m-1} \geq M$ 时的最小值, 则第 u 个子区域的中心向量为 $\left[\frac{k_1}{h} \quad \frac{k_2}{h} \quad \dots \quad 1 - \frac{1}{h} \sum_{l=1}^{m-1} k_l \right]$ 。当 $m \geq 3$ 时, 一共产

生 C_{h+m-1}^{m-1} 个中心向量, 这些向量的下标为 $u = 1, 2, \dots, C_{h+m-1}^{m-1}$ 。若 $C_{h+m-1}^{m-1} > M$, 则从下标 $u = 2$ 开始, 以 $\frac{M-2}{C_{h+m-1}^{m-1}-M}$ 为步长依次删除 $C_{h+m-1}^{m-1} - M$ 个下标对应的中心向量, 然后将剩下的 M 个中心向量依次作为 $A_j, j=1, 2, \dots, M$ 。通过每个子区域的中心向量找出该子区域的 T 个相邻的子区域, 参考指标为两个中心向量的余弦值。

2.2 粒子的分类与更新

在对粒子进行分配时, 通过参考点确定每个粒子的方向向量^[7]。令参考点为 $R(r_1, r_2, \dots, r_m)$, 其中 $r_i = \min\{f_i(x) | x \in \Omega\}, i = 1, 2, \dots, m$, m 为目标函数个数。例如: 如图1所示, f_1 和 f_2 表示两个目标函数, $m=2$, 则图中粒子 C 的方向向量为从点 $R(r_1, r_2)$ 指向点 C 的向量 $X, X=C-R$ 。通过比较粒子方向向量和所有子区域中心向量的余弦值, 确定该粒子属于最大余弦值所对应的中心, 向量子区域。

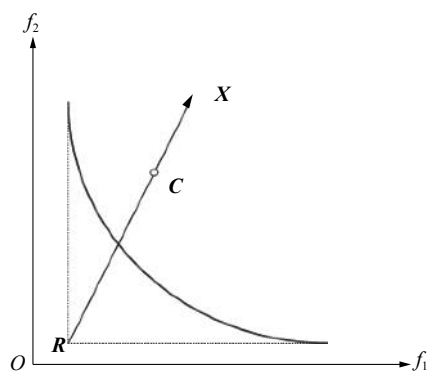


图1 粒子的方向向量

Fig. 1 Direction vector of the particle

在对粒子进行分配时会有以下两种特殊情况:

1) 有的子区域粒子数大于子区域的容量 Vol 时, 通过适应值进行取舍, 适应值公式为

$$f = aT + CD \quad (7)$$

式中: T 是 Pareto 支配强度, 其数值为当前粒子支配的粒子数, 在适应值计算中加入了 Pareto 支配强度 T , 增强每个子区域中的粒子趋向真实 PF 的能力; 参数 a 表示支配强度对适应值的影响程度, $a = \frac{2}{M}$, M 指目标空间子区域数; CD 为拥挤距离, 通过每个目标函数值对粒子进行排序, 序列两端粒子在当前目标函数中的拥挤距离设为 5, 其他粒子在当前目标函数中的拥挤距离为在序列中该粒子前后两粒子的目标函数值之差的绝对值, 最后将求出的该粒子在每个目标函数中计算的拥挤距离之和作为当前该粒子的拥挤距离 CD 。

计算该子区域包含的粒子的适应值, 从大到小排序, 选择序列中前 30% 的粒子, 再从这些粒

子中选择离所在子区域中心向量最近的 Vol 个粒子, 多余的劣质粒子删除。

2) 当子区域粒子数少于该子区域容量 Vol 时, 由于情况 1) 中已经对劣质的粒子进行删除, 为了尽可能保证整体的优越性, 所以重新初始化该子区域所缺数目的粒子作为该区域的粒子, 增加获得较优粒子的可能性, 增加粒子多样性。然后计算新粒子的目标函数值。

2.3 连续变异操作

当只采用一种方法进行变异时, 不能兼顾全局粒子和局部粒子的特性。在 MOPSO/DC 中提出差分+柯西^[12]+高斯^[13]连续的变异策略, 公式为

$$\begin{cases} X_{\text{diff}} = X + 0.5(X_1 - X_2) \\ X_{\text{Cauchy}} = \frac{c(t)^2}{\pi(X^2 + c(t)^2)} \\ X_{\text{Gauss}} = \frac{1}{g(t)\sqrt{2\pi}} \exp\left(\frac{-X^2}{2}\right) \end{cases} \quad (8)$$

式中: X 为当前引导粒子 g_{best} 的位置; X_1 和 X_2 是从 EPOP 中随机选出的两个不同粒子的位置。这样在变异时可保留全局粒子的一部分性质, 起到一定信息交流的作用, 增加了多样性。 t 为当前迭代的次数, g_{max} 为最大迭代次数, $g(1) = 2, g(t+1) = \frac{g(t)-1}{g_{\text{max}}}$; $c(1) = 1, c(t+1) = \frac{c(t)-1}{g_{\text{max}}}$; X_{diff} 、 X_{Cauchy} 和 X_{Gauss} 分别是对 g_{best} 的位置进行变异操作之后产生的新粒子。每次变异后让产生的新粒子和当前的 g_{best} 进行比较, 选择支配权优先的作为 g_{best} , 然后进行后续变异操作, 直到 3 次变异结束, 最终确定引导粒子 g_{best} 。

高斯变异步长较短, 能很好地吸取局部粒子性质, 柯西变异具有相对大的步长, 能进行较大范围的变异, 具有全局范围变异的特性, 这样能很好地产生全新的粒子。

下面对标准柯西分布 y_c 和标准高斯分布 y_g 进行对比分析, 根据 y_c 和 y_g 的单调性和对称性, 对其第一象限进行分析: $y_c = \frac{1}{\pi(x^2+1)}$, $y_g = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$, 求反函数 $x_c^2 = \frac{1}{\pi y_c} - 1$ 和 $x_g^2 = -2 \ln(\sqrt{2\pi} y_g)$, 求两反函数之差 $(x_c^2 - x_g^2)$, 并用 y 同时替代 y_c 和 y_g , 即 $T = \frac{1}{\pi y} - 1 + 2 \ln(\sqrt{2\pi} y)$; 对 T 求关于 y 的导数, 即 $T' = -\frac{1}{\pi y^2} + \frac{2}{y}$, T 在 $(0, \frac{1}{2\pi})$ 上单调减, 在 $(\frac{1}{2\pi}, +\infty)$ 上单调增, 所以 $T(0) > T(\frac{1}{5\pi}) > T(\frac{1}{2\pi})$; 又因为 $T(\frac{1}{5\pi}) > 0$, 所以 $T(0) > 0$; 所以在 y_c 和 y_g 同时趋向于 0 时, $x_c^2 > x_g^2$, 即 $x_c > x_g$, 所以柯西分布在 x 轴上覆盖的面积比高斯分布大。从图2中可直

观地看出,柯西分布具有较宽的变异范围,粒子具有较大范围的变异。

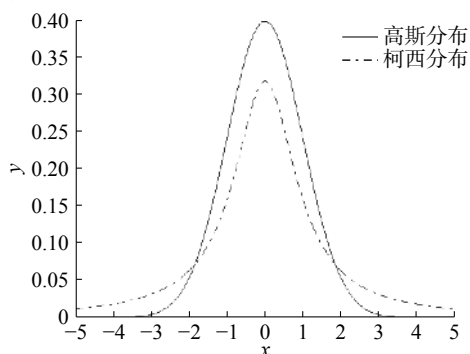


图2 标准高斯和分布柯西分布对比

Fig. 2 Comparison of standard Gaussian and Cauchy distributions

用这3种方法进行变异操作,既增强了 g_{best} 的引导能力,同时 g_{best} 吸取局部或全局粒子的性质,促进了粒子之间信息交流共享,增加了多样性。

2.4 MOSPO/DC 步骤

1) 初始化 $2N$ 个粒子的位置,这些粒子的位置集合记为POP,初始化 N 个速度,这些速度集合记为 V ,设目标函数的个数为 m ,当前迭代次数为 t ,初始化最大迭代次数 g_{max} 、每个子区域容量 $Vol \in [1, 3]$ 、邻域个数 T 以及子区域数目 M 。

2) 进行目标空间分解操作。

3) 计算粒子群POP目标函数值,选出每个目标函数的最小值作为参考点 R 用于计算每个粒子的方向向量,方向向量的计算参考图1。

4) 进行粒子的分类与更新操作,接着将当前所有子区域粒子的位置信息存入集合EPOP中,最后更新参考点 R ,并重新计算每个粒子的方向向量,清空POP备用。

5) g_{best} 的选择:产生 $[0, 1]$ 之间的随机数 r_1 ,当 $r_1 > 0.8$ 时,随机从EPOP集合中选择一个粒子作为当前粒子的 g_{best} ;否则,在该粒子所在区域的邻域内操作。首先计算该粒子邻域中每个子区域的中心向量和该子区域中粒子方向向量的余弦值,然后选出余弦值最大的子区域中的粒子作为当前粒子的 g_{best} 。过多地使用变异会让算法变得更加随机,降低了算法的效率,所以产生 $[0, 1]$ 之间的随机数 r_2 ,当 $r_2 < 0.6$ 时,对选出的引导粒子位置进行连续变异操作,确定最终的引导粒子 g_{best} 。否则,不对 g_{best} 进行变异操作。

6) p_{best} 的选择:产生 $[0, 1]$ 之间的随机数 r_3 ,当 $r_3 > 0.8$ 时,随机从EPOP中选择一个粒子;否则,

从该粒子的所在区域的邻域中随机选择一个粒子。让当前粒子和这个随机选出的粒子进行比较,选出支配权优先的作为当前粒子的 p_{best} 。

7) 通过上述选出的 g_{best} 和 p_{best} ,以及粒子速度更新式(5)和位置更新式(6)产生下一代新的粒子群体,这些新粒子位置集合记为NPOP,下一代新的速度集合重新覆盖集合 V 。

8) 如果当前迭代次数 t 大于最大迭代次数 g_{max} ,则循环结束,输出EPOP作为最优解集,否则把NPOP和EPOP合并放入POP中,然后跳转至3),继续循环。

3 实验分析

为了测试所提出的MOPSO/DC算法的性能,将其与目前较流行的MOPSO/D^[7]、MOPSOTL^[8]、MOPCODE^[9]和NNIA^[14]算法进行对比,实验中同样选择了文献[7]中的6个测试函数,依次为F1, F2...F6。为了定量比较5种算法的性能,采用以下3种广泛使用的性能指标:HV^[15],测量了粒子收敛于真实PF的效果以及最优解集的多样性,所获得值越大越好;GD^[16],指算法获得的PF到真实PF的距离,数值越小越接近真实PF,效果越好;IGD^[15],显示真实PF到算法所得到PF的距离,值越小越能说明算法得到的PF上的点可以分散地均匀地收敛于真实PF上。

MOPSO/DC参数设置:当目标函数个数 $m=2$ 时, $M=500$ 。当 $m=3$ 时, $M=1035$ 。 $Vol=1$, $N=MVol$, $T=[0.1N]$,粒子维度 $n=10$,最大循环次数 $g_{max}=1000$,一共进行了30次实验求各性能指标的均值和标准差。其他算法参数请参考文献[7]。

MOPSO/DC与其他算法关于这6个测试函数的3种性能指标对比结果见表1,表中给出了通过MOPSO/DC和其他4个算法获得关于这6个问题的IGD、GD和HV的平均值和标准差。从表1中可以看出:1)MOPSO/DC获得的IGD平均值远小于其他4种算法;2)MOPSO/DC获得的GD平均值都是最佳的,除F3函数外,不过关于F3的GD均值,本算法也优于MOPSO/D算法^[7];3)通过MOPSO/DC获得的HV值总体来说比其他算法获得的HV值好(除F1函数),说明MOPSO/DC的收敛性以及解的多样性也相当好。整体来说,MOPSO/DC的这3个性能指标优于其他4个算法,对MOPSO/D算法进行了很好的改进。

表 1 各算法效果指标对比
Table 1 Comparison of the effect of each algorithm

函数	算法	IGD		GD		HV	
		平均值	标准差	平均值	标准差	平均值	标准差
F1	MOPSO/DC	0.000 0	0.000 0	0.000 1	0.000 0	0.332 4	0.000 0
	MOPSO/D	0.004 5	0.000 1	0.001 8	0.000 1	0.659 8	0.000 2
	MOPSOTL	0.143 3	0.137 5	0.001 4	0.002 0	0.274 1	0.116 5
	MOPSONDE	0.119 0	0.076 9	0.002 1	0.001 8	0.264 1	0.057 6
	NNIA	0.383 0	0.014 9	0.061 5	0.134 7	0.125 1	0.026 5
F2	MOPSO/DC	0.000 0	0.000 0	0.000 0	0.000 0	0.665 7	0.000 0
	MOPSO/D	0.005 9	0.000 4	0.002 2	0.000 2	0.325 2	0.000 3
	MOPSOTL	0.321 4	0.297 7	0.000 4	0.000 6	0.374 3	0.227 7
	MOPSONDE	0.309 2	0.226 2	0.000 7	0.001 2	0.370 9	0.170 5
	NNIA	0.354 6	0.000 0	0.000 0	0.000 0	0	0
F3	MOPSO/DC	0.000 0	0.000 0	0.000 2	0.000 0	0.784 5	0.000 0
	MOPSO/D	0.005 3	0.000 2	0.001 5	0.000 2	0.202 7	0.000 4
	MOPSOTL	0.422 0	0.329 7	0.000 5	0.000 6	0.468 0	0.245 9
	MOPSONDE	0.369 6	0.321 4	0.001 0	0.002 0	0.501 4	0.232 9
	NNIA	0.387 3	0.000 0	0.000 0	0.000 0	0	0
F4	MOPSO/DC	0.000 0	0.000 0	0.000 2	0.000 1	0.596 1	0.014 5
	MOPSO/D	0.004 7	0.000 1	0.001 7	0.000 2	0.513 1	0.000 6
	MOPSOTL	0.327 8	0.253 9	0.000 7	0.000 7	0.147 1	0.168 0
	MOPSONDE	0.224 8	0.190 0	0.002 6	0.003 8	0.334 5	0.217 6
	NNIA	0.356 2	0.003 9	0.007 5	0.001 6	0.018 7	0.008 4
F5	MOPSO/DC	0.000 1	0.000 0	0.000 7	0.000 0	0.809 9	0.001 2
	MOPSO/D	0.041 0	0.000 6	0.021 9	0.000 6	0.764 5	0.003 7
	MOPSOTL	0.038 9	0.024 9	0.004 3	0.000 8	0.737 4	0.024 7
	MOPSONDE	0.296 2	0.249 7	0.005 0	0.000 9	0.572 5	0.059
	NNIA	0.299 4	0.000 0	0.017 2	0.000 3	0.494 0	0.000 3
F6	MOPSO/DC	0.000 2	0.000 0	0.000 7	0.000 0	0.459 8	0.001 5
	MOPSO/D	0.062 7	0.003 4	0.026 9	0.001 8	0.391 2	0.005 7
	MOPSOTL	0.078 7	0.006 6	0.006 9	0.001 3	0.334 6	0.010 3
	MOPSONDE	0.322 2	0.049 5	0.012 4	0.003 7	0.216 8	0.035 3
	NNIA	0.369 4	0.000 1	0.018 7	0.000 4	0.210 2	0.000 1

MOPSO/DC 算法的仿真图如图 3, 结果显示该算法得到的 PF 和真实 PF 完全重合。

通过这些对比实验得出结论: MOPSO/DC 对

原算法^[7]的收敛性和多样性进行了很好地优化, 增强了粒子趋向真实 PF 的效果, 并且具有很好的稳定性。

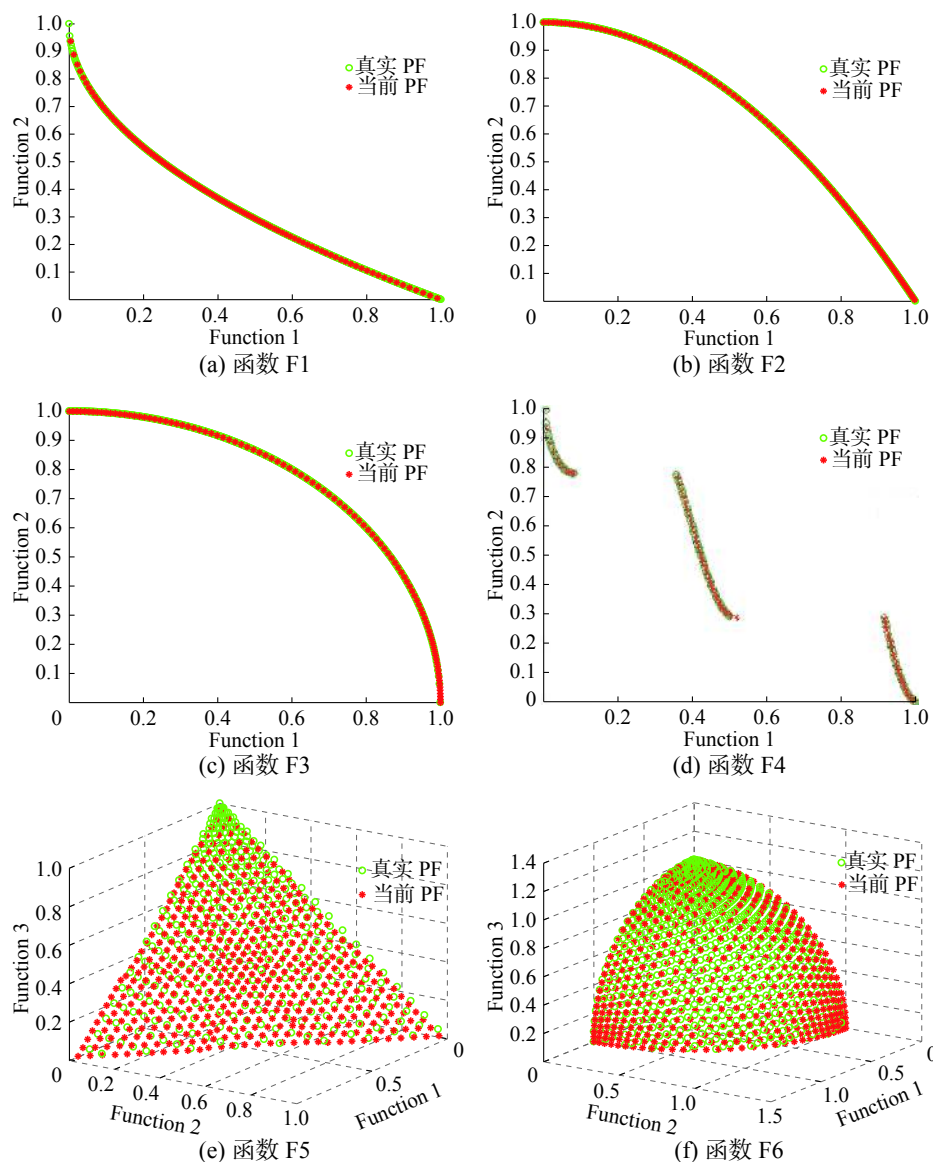


图3 测试函数的仿真图

Fig. 3 Test function's simulations

4 结束语

本文提出的算法采用目标空间分解与子区域粒子更新和对引导粒子 g_{best} 的位置进行连续变异的方法来提高粒子多样性和收敛性, 对原算法的不足之处进行了很好地弥补和优化。通过和4个不同种类的优化算法对不同种测试函数进行多种性能的测试对比实验证明, MOPSO/DC 整体效果最好, 具有很高收敛性, 多样性方面也有提升。总之, MOPSO/DC 对多目标优化问题的多个性能指标同时进行了很好的优化。未来的研究重点是, 将该算法更好地适应更多目标的优化问题, 增强其通用性, 用于解决实际问题。

参考文献:

- [1] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95-International Conference on Neural Networks. Perth, WA, Australia, Australia: IEEE, 1995: 1942-1948.
- [2] COELLO C A C, LECHUGA M S. MOPSO: a proposal for multiple objective particle swarm optimization [C]//Proceedings of the 2002 Congress on Evolutionary Computation. Honolulu, HI, USA: IEEE, 2002: 1051-1056.
- [3] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization[J]. IEEE transactions on evolutionary computation, 2004, 8(3): 256-279.
- [4] RAQUEL C R, NAVAL P C JR. An effective use of crowding distance in multiobjective particle swarm optimization[C]//Proceedings of the 7th Annual Conference Genetic and Evolutionary Computation Washington, DC,

- USA: ACM, 2005: 257–264.
- [5] LI Li, WANG Wanliang, XU Xinli. Multi-objective Particle swarm optimization based on global margin ranking[J]. Information sciences, 2017, 375: 30–47.
- [6] LIN Qiuzhen, LI Jiangqiang, DU Zhihua, et al. A novel multi-objective particle swarm optimization with multiple search strategies[J]. European journal of operational research, 2015, 247(3): 732–744.
- [7] DAI Cai, WANG Yuping, YE Miao. A new multi-objective particle swarm optimization algorithm based on decomposition[J]. Information sciences, 2015, 325: 541–557.
- [8] CHENG Tingli, CHEN Minyou, FLEMING P J, et al. A novel hybrid teaching learning based multi-objective particle swarm optimization[J]. Neurocomputing, 2017, 222: 11–25.
- [9] SU Yixin, CHI Rui. Multi-objective particle swarm-differential evolution algorithm[J]. Neural Computing and applications, 2017, 28(2): 407–418.
- [10] ZITZLER E, LAUMANN M, THIELE L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization[M]//GIANNAKOGLU K C, TSAHALIS D T, PÉRIAUX J, et al. Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems. Athens, Greece: International Center for Numerical Methods in Engineering, 2002: 95–100.
- [11] JORDEHI A R. Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems[J]. Applied Soft Computing, 2015, 26: 401–417.
- [12] 陈明杰, 黄佰川, 张旻. 混合改进蚁群算法的函数优化[J]. 智能系统学报, 2012, 7(4): 370–376.
- CHEN Mingjie, HUANG Baichuan, ZHANG Min. Function optimization based on an improved hybrid ACO[J]. CAAI transactions on intelligent systems, 2012, 7(4): 370–376.
- [13] CHELLAPILLA K, FOGEL D B. Two new mutation operators for enhanced search and optimization in evolutionary programming[C]//Proceedings Volume 3165, Applications of Soft Computing. San Diego, CA, United States: SPIE, 1997: 260–269.
- [14] GONG Maoguo, JIAO Licheng, DU Haifeng, et al. Multiobjective immune algorithm with nondominated neighbor-based selection[J]. Evolutionary computation, 2008, 16(2): 225–255.
- [15] STORN R, PRICE K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of global optimization, 1997, 11(4): 341–359.
- [16] ZITZLER E, THIELE L, LAUMANN M, et al. Performance assessment of multiobjective optimizers: an analysis and review[J]. IEEE transactions on evolutionary computation, 2003, 7(2): 117–132.

作者简介:



钱小宇, 男, 1992 年生, 硕士研究生, 主要研究方向为人工智能与模式识别。



葛洪伟, 男, 1967 年生, 教授, 博士, 博士生导师, 主要研究方向为人工智能与模式识别、机器学习、图像处理与分析等。



蔡明, 男, 1962 年生, 高级工程师, 主要研究方向为计算机软件、网络应用的研究。