文章编号: 1001-0920 (2013) 10-1513-07

一种基于相角映射的改进多目标粒子群优化算法

李 婷, 吴 敏, 何 勇

(中南大学 a. 信息科学与工程学院，b. 先进控制与智能自动化湖南省工程实验室，长沙 410083)

摘 要: 提出一种相角粒子群优化算法求解多目标优化问题. 该算法采用相角映射实现了粒子在相角空间上仅依赖于归一化多目标函数的快速搜索, 在粒子飞行信息共享机制上引入共享池概念, 提出基于关联支配排序和相似度排序的共享池更新策略, 提高了 Pareto 解的多样性. 采用 Sigma 领导策略和混沌变异操作, 平衡了算法的快速搜索能力和全局寻优能力. 标准多目标测试函数和电力系统广域阻尼控制多目标优化算例表明了所提出算法的可行性和有效性.

关键词: 多目标优化；粒子群优化算法；混沌；多目标粒子群优化算法

中图分类号: TP273 **文献标志码:** A

Improved multi-objective particle swarm optimization algorithm based on phase angle reflection

LI Ting, *WU Min*, *HE Yong*

(a. School of Information Science and Technology，b. Hu'nan Engineering Laboratory for Advanced Control and Intelligent Automation，Central South University，Changsha 410083，China．Correspondent：LI Ting，E-mail: litingcsu@163.com)

Abstract: Phase angle particle swarm optimization(PAPSO) algorithm is proposed to solve multi-objective optimization problems. The algorithm adopts phase angle reflection, such that particles can rapidly search solutions in phase angle space, which is only dependent on normalized multi-objective functions. Sharing pool concept is introduced into the particle flight information sharing mechanism, and the sharing pool update strategy based on the related predominance sorting and similarity degree sorting is presented to improve the diversity of Pareto solutions. Sigma leading strategy and chaos variation operation balance its ability of finding optimal solutions rapidly and extensively. Cases related to benchmark multi-objective test functions and wide area damping control multi-objective optimization in power systems verify the feasibility and effectiveness of the proposed algorithm.

Key words: multi-objective optimization；particle swarm optimization；chaoes；multi-objective particle swarm optimization

0 引 言

研究多目标优化理论和方法具有重要的工程应用价值, 如兼顾发电燃料成本和NO_x排放控制的电力系统优化调度[1]、计及节能减排目标和配料成本的钢铁烧结配料优化[2]、铅锌烧结过程中烧结矿产量和质量指标控制[3]等. 由于各目标之间相互制约, 多目标优化问题一般不存在使各优化目标均为最优的解, 但是存在一个称为 Pareto 最优解集的非劣解集合. 衡量多目标优化问题求解方法的性能包括两方面: 一是要尽可能快尽可能多地求解到 Pareto 最优解, 二是要使求得的解在 Pareto 最优解集中均匀分布. 进化计算通过种群间协同机制可以实现并行计算, 并能在一次进化中同时得到多个解, 非常适合求解多目标优化问题.

由 Kennedy 等[4]提出的粒子群优化算法 (PSO) 具有并行计算、收敛速度快、易于编程实现等特点, 是一种有效的优化工具. 以标准 PSO 算法为基础, 根据特定问题设计各种改进的 PSO 算法已在诸多领域取得成功应用[5-6]. 但是标准 PSO 算法并不能直接用于求解多目标优化问题, Eberhart 等[7]通过将其与遗

收稿日期: 2012-06-04；**修回日期:** 2012-10-20.

基金项目: 国家自然科学基金重大国际合作研究项目(61210011)；国家自然科学基金项目(60974045).

作者简介: 李婷(1978−), 女, 副教授, 博士, 从事电力系统分析与控制、智能控制的研究；吴敏(1963−), 男, 教授, 博士生导师, 从事鲁棒控制、智能控制等研究.

传算法比较给出了原因: 遗传算法通过染色体双向共享信息引导种群向最优区域进化; 标准PSO算法通过个体经验和群体经验单向共享信息引导粒子向最优值进化, 因此不能找到Pareto解集. 为此, Eberhart提出了两种改进方法: 1) 使用动态邻域[7], 即根据第1个优化目标计算当前粒子与其他粒子的距离确定动态邻域, 根据第2个目标选择邻域内粒子作为群体的领导粒子, 但这种方法只能处理双目标优化问题; 2) 引入扩展内存存储Pareto最优解[8], 并将其作为群体的领导粒子, 与第1种方法相比可显著减少计算时间. 与第2种方法类似, 将容器[9]和档案[10-11]的思想引入PSO算法, 用于解决多目标优化中个体或群体领导粒子的产生、存储、更新、选择问题. 除了根据动态邻域概念和档案策略进行算法改进外, 还有学者试图将PSO算法与遗传算法[3]、差分算法[12]等其他优化算法相结合来求解多目标优化问题. 但是这些算法均没有考虑粒子空间分布和粒子演化过程的特征, 在求解实际工程中的复杂多目标优化问题时, 仍然存在求解效率不高或Pareto最优解多样性不足等问题.

针对多目标PSO算法(MOPSO)的不足, 本文提出一种基于相角映射的PSO算法(PAPSO)用于求解多目标优化问题. 在该方法中, PSO算法中的粒子视作具有位置和方向的相角矢量, 并且该相角矢量的位置在各个子优化目标函数中具有均等的上界和下界, 确保了相角位置初始化时空间分布的等概率性和搜索过程的紧凑性, 客观上加快了算法的搜索速度. 同时, 还设计了粒子飞行信息共享机制和粒子混沌变异操作, 使得PAPSO算法可以快速搜索到Pareto前沿. 该算法用于求解标准多目标函数优化和电力系统广域阻尼控制多目标优化问题, 取得了较好的优化结果, 从而验证了算法的有效性和可行性.

1 相角多目标粒子群优化算法

1.1 PSO算法信息共享机制

PSO算法将待优化问题的每个可能解视作“粒子”, 该粒子具有位置和速度两个属性. 算法的寻优过程是: 首先通过初始化操作产生一群随机粒子即随机解, 然后通过种群内粒子间信息共享规则更新粒子的速度和位置, 指导粒子在搜索空间中以一定的速度向更优的目标飞行. 在第k次飞行时, 种群中第i个粒子的状态更新方程为

$$v_{i,k+1}=\omega v_{i,k}+c_1r_1(p_{i,k}-x_{i,k})+ c_2r_2(g_k-x_{i,k}), \tag{1}$$

$$x_{i,k+1}=x_{i,k}+v_{i,k+1}. \tag{2}$$

其中: $v_{i,k}$为粒子当前速度; $v_{i,k+1}$为更新后的粒子速度; $p_{i,k}$为第i个粒子目前搜索到的最好位置; g_k为种群目前搜索到的最好位置; $x_{i,k}$为当前粒子位置; $x_{i,k+1}$为更新后的粒子位置; r_1和r_2为[0,1]之间的随机数; c_1和c_2为加速因子; ω为惯性权重因子.

由式(1)和(2)可知, PSO算法能搜索到最优值的根本原因是该算法通过共享粒子个体飞行经验$p_{i,k}$和群体飞行经验g_k, 使整个种群向最优值进化. 这是一种单向信息共享机制, 因为群体飞行经验g_k对于整个种群是惟一的, 种群中其他粒子将向单一更优目标飞行, 所以PSO算法在求解单目标优化问题时是十分高效的. 但是对于多目标优化问题, 由于Pareto最优解往往不是惟一的, 基于单一飞行目标的PSO算法不能直接用于求解Pareto最优解集[7], 必须根据多目标优化问题的特点, 对标准PSO算法信息共享机制进行重新设计, 使之能够直接高效求解多目标优化问题的Pareto最优解集.

1.2 相角PSO

多目标优化问题可以描述如下:

$$\begin{aligned}&\min\ f(x)=[f_1(x),f_2(x),\cdots,f_m(x)],\\ &\quad x=(x^1,x^2,\cdots,x^d);\\ &\text{s.t. } y_n(x)\leqslant 0.\end{aligned} \tag{3}$$

其中: $f(x)\subset R^m$为含m个优化子目标的目标函数; x为d维决策变量; $y_n(x)$为含n个约束的不等式组. 定义解x_1 Pareto支配x_2如下:

$$\begin{aligned}x_1\prec x_2\Leftrightarrow f_1(x_1)<f_1(x_2),\ f_2(x_1)<f_2(x_2),\\ \cdots,f_m(x_1)<f_m(x_2).\end{aligned} \tag{4}$$

则Pareto最优解集X为

$$X=\{x|\ \nexists x_i\prec x\}. \tag{5}$$

将PSO算法中粒子的状态映射为相角矢量$\boldsymbol{\psi}$, 该矢量可以通过初始位置ψ和方向$\Delta\psi$惟一确定. 对于第i个粒子有

$$\begin{aligned}x_{i,k}=0.5[x^{\max}(1+\sin(\psi_{i,k}))+\\ x^{\min}(1-\sin(\psi_{i,k}))].\end{aligned} \tag{6}$$

其中: $x^{\max}$和$x^{\min}$分别为决策变量x的最大值和最小值, $\psi_{i,k}$为第k次迭代时第i个粒子的相角. 相角矢量更新方程为

$$\begin{aligned}\Delta\psi_{i,k+1}=\omega\Delta\psi_{i,k}+c_1r_1(p^{\psi}_{i,k}-\psi_{i,k})+\\ c_2r_2(g^{\psi}_{i,k}-\psi_{i,k}),\end{aligned} \tag{7}$$

$$\psi_{i,k+1}=\psi_{i,k}+\Delta\psi_{i,k+1}, \tag{8}$$

$$f_j^{'}(\psi_{i,k})=\frac{f_j(x_{i,k})-f_j^{\min}}{f_j^{\max}-f_j^{\min}},\ j=1,2,\cdots,m. \tag{9}$$

其中: $\Delta\psi_{i,k}$为第k次迭代时第i个粒子的相角增量, $p^{\psi}_{i,k}$和$g^{\psi}_{i,k}$分别为相角矢量坐标下第i个粒子和种群目前搜索到的最好位置; $f_j^{'}(\psi_{i,k})$为相角矢量坐标

下第 j 个子目标函数. PAPSO 的寻优过程具备以下特征: 1) 决策变量 $\psi_{i,k}$ 各维分量的取值均在 $[-\pi/2, \pi/2]$ 之间, 决策变量构成的解空间为 d 维对称空间; 2) 目标函数各子目标函数 $f_j^{'}(\psi_{i,k})$ 的取值均在 [0,1] 之间, 目标函数空间为 m 维对称空间; 3) 相对标准 PSO 算法而言, PAPSO 在解空间上初始化时具有均等的概率, 在目标函数空间上搜索过程更加紧凑和快速; 4) 粒子不再向种群中单个更优目标 g_k 飞行, 而是向种群的 Pareto 最优解和非劣解 $g_{i,k}^{\psi}$ 飞行, 使得 PAPSO 算法具有多目标寻优能力.

1.3 多样性保持的共享池策略

为了求解多目标优化问题, $p_{i,k}^{\psi}$ 和 $g_{i,k}^{\psi}$ 是决定 PAPSO 算法能否找到 Pareto 最优解集的关键. 本文采用共享池存储在计算过程中产生的 Pareto 最优解集和非劣解集信息, 粒子从共享池中根据一定策略选择领导粒子 $g_{i,k}^{\psi}$ 以指导下一次飞行. 共享池大小、共享池中解的多样性和领导粒子更新策略等对 PAPSO 算法影响较大. 定义解 ψ_i 关联支配 ψ_l 如下:

$$\psi_i \bowtie \psi_l \Leftrightarrow (\psi_i \prec \psi_l) \bigcup (\psi_i \sim \psi_l), \tag{10}$$

$$\psi_i \sim \psi_l \Leftrightarrow \psi_l \sim \psi_i \Leftrightarrow \exists j,\ |f_j^{'}(\psi_i) - f_j^{'}(\psi_l)| < \varepsilon_j, \tag{11}$$

$$\varepsilon_j = (\varepsilon_j^{\max} - \varepsilon_j^{\min}) \frac{k_{\max} - k}{k_{\max}} + \varepsilon_j^{\min}, \tag{12}$$

其中 ε_j 为自适应关联支配系数. 当 $\varepsilon_j = 0$ 时, 关联支配关系退化为支配关系, 共享池中的粒子变得过于紧密; 当 ε_j 取值过大时, 关联支配关系变强, 共享池中的粒子变得过于稀疏. 在 PAPSO 算法的初始阶段, ε_j 取值为最大值 $\varepsilon_j^{\max}$, 使得粒子可以在关联支配集的较大邻域范围内搜索 Pareto 最优解; 在 PAPSO 算法的最后阶段, ε_j 取值为最小值 $\varepsilon_j^{\min}$, 确保粒子在 Pareto 最优解集的较小邻域范围内搜索 Pareto 最优解. 对于优化子目标数 $m > 2$ 的多目标优化问题, $\varepsilon_j^{\max}$ 和 $\varepsilon_j^{\min}$ 的经验取值为 [0.1,0.3], 并且随着 m 的增加, $\varepsilon_j^{\max}$ 和 $\varepsilon_j^{\min}$ 的取值相对增大.

在第 k 次迭代中, 共享池中两个粒子 $\psi_{i,k}$ 和 $\psi_{l,k}$ 之间的距离 δ 定义为

$$\delta(i,l) = \sqrt{\sum_{j=1}^{m} (f_j^{'}(\psi_{i,k}) - f_j^{'}(\psi_{l,k}))^2}. \tag{13}$$

定义相似度函数 ρ 为

$$\rho(i,l) = \begin{cases} 1 - \dfrac{\delta(i,l)}{\delta_0},\ \delta(i,l) \leqslant \delta_0; \\ 0,\ \delta(i,l) > \delta_0. \end{cases} \tag{14}$$

$$\rho(i) = \sum_{l} \rho(i,l),\ l = 1, 2, \cdots, s,\ l \neq i. \tag{15}$$

其中: δ_0 为预定的相似度阈值, s 为共享池中粒子的个数. 共享池中粒子的更新方式如下: 对共享池和迭代产生的新粒子进行联合排序, 首先根据关联支配关系 $\psi_i \bowtie \psi_l$ 进行非劣排序; 然后根据相似度函数 ρ 进行逆序排序; 最后用具有更好关联支配关系和相似度的新粒子替换共享池中的粒子. 这种更新方式可以确保共享池中粒子的领导性和多样性, 共享池的大小 s 根据计算复杂度和计算精度确定. 对于适应值函数计算时间较长的多目标优化问题, 可以选择较小的种群规模和较大的共享池以提高算法计算速度.

为了进一步提高粒子搜索速度, 每个粒子选择共享池中 Sigma 值最接近的粒子作为 $g_{i,k}^{\psi}$ [13]. 二维 Sigma 值定义为

$$\sigma = \frac{f_1^{'2}(\psi_i) - f_2^{'2}(\psi_i)}{f_1^{'2}(\psi_i) + f_2^{'2}(\psi_i)}. \tag{16}$$

三维 Sigma 值定义为

$$\sigma = \begin{bmatrix} f_1^{'2}(\psi_i) - f_2^{'2}(\psi_i) \\ f_2^{'2}(\psi_i) - f_3^{'2}(\psi_i) \\ f_3^{'2}(\psi_i) - f_1^{'2}(\psi_i) \end{bmatrix} [f_1^{'2}(\psi_i) + f_2^{'2}(\psi_i) + f_3^{'2}(\psi_i)]^{-1}. \tag{17}$$

采用 Sigma 值作为领导粒子的选择策略, 本质上是使在一定扇形区域内的粒子以相同的斜角飞行, 直至飞行到 Pareto 最优解集或目标函数空间的边界. Sigma 加速技术使得 PAPSO 算法具有更快的速度, 可以确保其他粒子以极快的速度飞向 Pareto 前沿.

1.4 混沌变异

PSO 算法的典型特点是搜索速度快, 但是对于进化算法而言, 过快的搜索速度可能会导致算法早熟. 为此, 本文引入混沌思想对种群中的粒子进行混沌寻优, 基于混沌遍历的规律性和不重复性用混沌寻优得到的更优位置替换种群中的粒子. 采用 Logistic 动力方程产生混沌序列, 有

$$\theta^{(t+1)} = \alpha \theta^{(t)} (1 - \theta^{(t)}),\ t = 0, 1, \cdots, T-1. \tag{18}$$

其中: T 为混沌迭代次数; α 为控制参数, 当 $\alpha = 4$ 时, 由初值 $\theta^{(0)} \in (0,1)$ 即可迭代出一个确定的混沌序列 $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(T)}$. 由 Logistic 动力方程构造的混沌序列具有结构简单、计算量小的优点. 将 ψ_i 的各维展开如下:

$$\psi_i = (\psi_{i1}, \psi_{i2}, \cdots, \psi_{id}). \tag{19}$$

则 ψ_i 的混沌序列 $\psi_i^{(t)}$ 描述为

$$\theta^{(0)} = 0.5 + \psi_{id}/\pi, \tag{20}$$

$$\psi_{id}^{(t)} = \pi \theta^{(t)} - \pi/2, \tag{21}$$

$$\psi_i^{(t)} = (\psi_{i1}^{(t)}, \psi_{i2}^{(t)}, \cdots, \psi_{id}^{(t)}). \tag{22}$$

若 $\theta^{(0)}$ 取值为 (0.25, 0.5, 0.75), 则设 $\theta^{(0)} = \theta^{(0)} + \mathrm{rand}(0, 0.1)$, 通过附加随机扰动可使 $\theta^{(0)}$ 跳出小周期点或不动点重新进入混沌. 上述混沌序列替换本质上

是通过混沌扰动使粒子跳出局部最优, 从而避免算法早熟, 有助于提高算法的收敛速度和精度. 给定迭代次数Δk和目标函数变化范围Δf, 当PAPSO算法中第i个粒子ψ_i在Δk中搜索到的Pareto最优解或非劣解的变化小于Δf时, 触发混沌变异, 即

$$\| f^{'}(\psi_{i,k+\Delta k}) - f^{'}(\psi_{i,k}) \| \leqslant \Delta f. \tag{23}$$

1.5 算法流程

PAPSO算法的具体计算流程如下.

Step 1: 将决策变量x通过相角映射转变为ψ变量; 将定义在$x \in R^d$空间上的目标函数$f(x) \subset R^m$映射为定义在$\psi \in [-\pi/2, \pi/2]^d$空间上的目标函数$f^{'}(\psi) \subset [0,1]^m$.

Step 2: 初始化PAPSO算法参数, 包括种群大小$i_{\max}$、共享池大小s、最大迭代次数$k_{\max}$、c_1、c_2、相似度阈值δ_0、$k = 0$、$i = 0$、惯性权重的最大值$\omega_{\max}$和最小值$\omega_{\min}$等, 初始化种群并产生共享池初值.

Step 3: 开始迭代计算$k = k + 1$.

Step 4: 开始遍历种群中的粒子$i = i + 1$.

Step 5: 根据Sigma方法从共享池中选择σ相近的粒子作为当前粒子的$g_{i,k}^{\psi}$, 根据$\omega = (\omega_{\max} - \omega_{\min}) \times (k_{\max} - k)/k_{\max} + \omega_{\min}$确定粒子的自适应惯性权重.

Step 6: 更新当前粒子的相角$\psi_{i,k}$和相角增量$\Delta\psi_{i,k}$.

Step 7: 对种群中部分粒子实施混沌变异.

Step 8: 若找到新的Pareto支配粒子$\psi_{i,k}$, 则更新当前粒子$p_{i,k}^{\psi}$.

Step 9: 根据关联支配关系和相似度函数更新共享池, 若$i < i_{\max}$, 则返回Step 4.

Step 10: 若$k < k_{\max}$, 则返回Step 3; 否则输出当前粒子或(和)共享池中的粒子作为PAPSO算法所能求得的Pareto解集.

2 性能测试

为了验证PAPSO算法的性能, 选择如下两个标准多目标优化函数F_1和F_2进行性能测试:

$$\begin{aligned} &\min\ f(x) = [f_1(x_1), f_2(x)];\\ &\text{s.t. } f_2(x) = y_1(x_2, x_3, \cdots, x_d) y_2 \times \\ &\qquad (f_1(x), y_1(x_2, x_3, \cdots, x_d)). \end{aligned} \tag{24}$$

F_1中的f_1, y_1和y_2分别为

$$\begin{aligned} &f_1(x_1) = x_1,\\ &y_1 = 1 + 9\sum_{j=2}^{d} x_j \Big/ (d-1),\\ &y_2(f_1, y_1) = 1 - \sqrt{f_1/y_1} - (f_1/y_1)\sin(10\pi f_1). \end{aligned} \tag{25}$$

F_2中的f_1, y_1和y_2分别为

$$\begin{aligned} &f_1(x_1) = x_1,\\ &y_1 = 1 + 10(d-1) + \sum_{j=2}^{d}(x_j^2 - 10\cos(4\pi x_j)),\\ &y_2(f_1, y_1) = -\sqrt{f_1/y_1}. \end{aligned} \tag{26}$$

其中: F_1的维数$d = 30$, $x_1, x_2, \cdots, x_d \in [0,1]$, 由于在$y_2$中存在$\sin(10\pi f_1)$项, F_1的Pareto前沿是不连续的; F_2的维数$d = 10$, $x_1 \in [0,1]$, $x_2, x_3, \cdots, x_d \in [-5, 5]$, F_2共有21^9个Pareto前沿, 其中只有1个全局最优Pareto前沿, 其余为局部最优Pareto前沿. F_1和F_2中Pareto前沿的不连续性和多局部极值的特点客观上增加了多目标优化算法寻优的难度.

为了评估多目标优化算法的性能, 定义收敛度指标和均匀度指标分别为

$$\lambda = \frac{1}{s}\sum_{i=1, x_i^{'} \in X^{'}}^{s} \min\{\| f(x_i^{'}) - f(x) \|; x \in X\}; \tag{27}$$

$$\phi = \Big[l_0 + l_s + \sum_{i=1}^{s-1} |l_i - \bar{l}|\Big] \Big/ [l_0 + l_s + (s-1)l_i], \tag{28}$$

$$l_i = \| f(x_i^{'}) - f(x_{i+1}^{'}) \|,\ i = 1, 2, \cdots, s-1,$$

$$\bar{l} = \sum_{i=1}^{s-1} l_i \Big/ (s-1).$$

其中: $X^{'}$为多目标优化算法的解集, $x_i^{'}$为$X^{'}$中的第i个解; l_i为第i个和第$i+1$个解之间的距离, l_0和l_s分别为第1个解和第s个解与Pareto前沿两端边界解之间的距离, $\bar{l}$为$X^{'}$中解之间的平均距离. 上述距离均在目标函数空间上度量. $\lambda = 0$表示$x_i^{'}$全部为Pareto最优解. 当l_0和l_s分别为0且$l_i = \bar{l}$时, 多目标优化算法的解集具有最好的均匀分布形态.

计算中PAPSO算法的种群规模为50, 共享池大小为100, 最大迭代次数为200, 运行30次, 计算结果的收敛度指标和均匀度指标见表1和表2. 为了比较PAPSO算法的性能, 表1和表2还列出采用NSGA-II[14]、SPEA[15]和MOPSO[15]算法求解F_1和F_2时计算结果的统计指标. NSGA-II和SPEA算法的种群规模和迭代次数均为100和250; MOPSO算法的种群规模和迭代次数为50和500. NSGA-II(R)和NSGA-II(B)分别为实数编码和二进制编码的NSGA-II算法.

表 1 算法收敛度指标比较

算法	λ均值		λ方差	
	F_1	F_2	F_1	F_2
NSGA-II(R)	0.114 50	0.513 053	0.007 94	0.118 46
NSGA-II(B)	0.043 411	3.227 636	0.000 042	7.307 63
SPEA	0.047 517	7.340 299	0.000 047	6.572 516
MOPSO	0.004 18	7.374 29	0.000 00	5.482 86
PAPSO	0.009 71	1.913 41	0.000 00	2.476 01

表 2　算法均匀度指标比较

算 法	ϕ均值		ϕ方差	
	F_1	F_2	F_1	F_2
NSGA-II(R)	0.738 54	0.702 612	0.019 706	0.064 648
NSGA-II(B)	0.575 606	0.479 475	0.005 078	0.009 841
SPEA	0.672 938	0.798 463	0.003 587	0.014 616
MOPSO	0.831 95	0.961 94	0.008 92	0.001 14
PAPSO	0.610 41	0.732 34	0.001 83	0.055 42

由表 1 可见, PAPSO 算法在求解 F_1 时收敛度指标整体上优于 NSGA-II 和 SPEA, 在求解 F_2 时收敛度指标整体上优于 NSGA-II(B)、SPEA 和 MOPSO, 这表明 PAPSO 算法由于采用了共享池共享粒子间寻优信息具备较大的概率找到 Pareto 最优解, 并且对于多局部极值优化问题具有较好的跳出局部极值的能力. 由表 2 可知, PAPSO 算法在求解 F_1 和 F_2 时均匀度指标均值优于 SPEA 和 MOPSO, 但部分指标差于 NSGA-II; 均匀度指标方差与 NSGA-II、SPEA 和 MOPSO 大体接近. 这是由遗传算法和 PSO 算法信息共享机制本质上不同决定的. 遗传算法中的染色体可以实现极为细微的最优值调整, 所以 NSGA-II(B) 算法可能产生最好的均匀度指标; 而 PSO 算法的粒子飞行受历史速度、最优目标等影响较大, 尽管可以采用各种措施提高算法均匀度, 但也难以达到 NSGA-II(B) 算法的优秀指标. 但是, PAPSO 算法通过对混沌变异触发条件 (式(23)) 的参数 Δk 和 Δf 进行合理设置, 可以极大地减少由混沌序列迭代产生的计算开销, 能够用相对较少的适应值函数评价次数获得较好的优化结果.

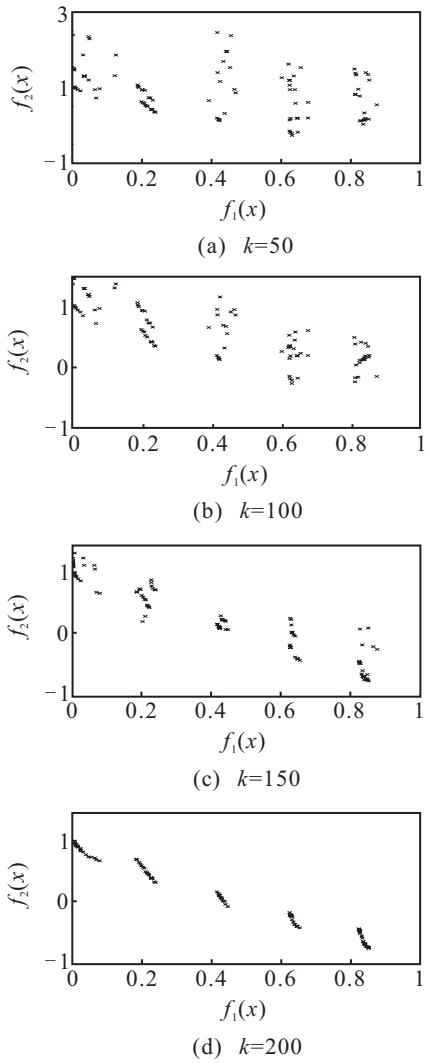

图 1　PAPSO 算法求解 F_1 时解的分布

图 1 为 PAPSO 算法在求解 F_1 时不同迭代次数下粒子在目标函数空间的分布情况. 从图 1 可以看到, PAPSO 算法在不到 200 次的迭代次数内已经搜索到 Pareto 前沿, 具备快速求解多目标优化问题的能力.

图 2 为 PAPSO 算法在求解 F_2 时的计算结果, 并与 SPEA 算法进行了比较. 图 2 表明 PAPSO 算法可以获得比 SPEA 算法更好的优化结果.

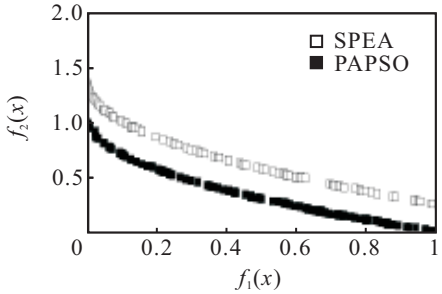

图 2　F_2 的多目标优化结果

3　算例分析

将 PAPSO 算法应用于电力系统广域阻尼控制多目标优化问题. 在广域测量环境下, 设计基于统一潮流控制器 (UPFC) 的辅助阻尼控制器 (SDC) 需要满足双重目标, 既能以更大的阻尼比 (ξ) 抑制主导振荡模式 ($\delta + j\varpi$), 又能具备更大的时滞稳定裕度使得 SDC 镇定信号即使被延迟一小段时间仍能使电力系统维持稳定. 电力系统常用的 SDC 数学模型为

$$\text{output} = K_{\text{sdc}}\frac{T_w s}{1+T_w s}\frac{1+T_1 s}{1+T_2 s}\frac{1+T_3 s}{1+T_4 s}\text{input}. \tag{29}$$

其中: K_{sdc} 为增益; T_w 为隔直参数; T_1, T_2, T_3 和 T_4 为超前滞后参数; input 和 output 分别为 SDC 的镇定输入和输出信号. 计及信号传输时滞的 UPFC 阻尼控制模型包括开环电力系统、SDC 和时滞环节 3 部分, 在平衡点展开后, 可转化为如下时滞电力系统模型:

$$\dot{u} = Au(t) + A_d u(t - d(t)). \tag{30}$$

其中: u 为系统状态, $d(t)$ 为时滞. 调节 SDC 的增益 (K_{sdc})、隔直 (T_w) 和超前滞后 (T_1, T_2, T_3, T_4) 等参数将改变时滞电力系统模型的系数矩阵 A 和 A_d, 进而影响电力系统的阻尼性能 ξ 和时滞稳定裕度 τ. 因此 SDC 的设计目标为

$$\max\ \xi = -\delta/\sqrt{\delta^2+\varpi^2},\quad \max\ \tau, \tag{31}$$

其中τ采用基于LMI的时滞依赖稳定条件计算[16]. 根据PSO算法, SDC中待优化的变量x为

$$x=(K_{\rm sdc},T_w,T_1,T_2,T_3,T_4). \tag{32}$$

通过相角映射并根据SDC设计经验有$T_3=T_1$, $T_4=T_2$, $T_w\in[5,10]$, 则PAPSO算法待优化的变量ψ简化为

$$\psi=(\psi_{K_{\rm sdc}},\psi_{T_1},\psi_{T_2}). \tag{33}$$

ψ的3个分量与x第1个、第3个、第4个分量的关系由式(6)给出.

设开环电力系统为KUNDUR的两区四机系统, UPFC安装于两区的联络线上. SDC输入信号为线路7-8的有功功率偏差$\Delta P_{7\text{-}8}$和线路6-7的电流偏差$\Delta I_{6\text{-}7}$. UPFC分别采用U_p, U_q和I_q三种控制方式实现线路有功、无功潮流控制和UPFC输入端电压控制. PAPSO算法的种群规模为25, 迭代次数为100, 共享池大小为50, $\omega_{\max}$和$\omega_{\min}$分别为0.9和0.4, $K_{\rm sdc}\in[0.1,1]$, $T_1,T_2\in[0.01,1]$, $T_w=10$. 由于计算τ十分耗时, PAPSO算法采用较小的种群规模以减少τ的计算次数, 同时PAPSO算法采用较大的共享池以保留较多的Pareto最优解和非劣解. 针对UPFC三种控制方式和SDC两种输入方式, 采用PAPSO算法可以得到满足式(31)的不同SDC参数. 图3为UPFC采用I_q控制方式且SDC输入信号为ΔP_{7-8}时PAPSO算法和SPEA算法的计算结果. 由图3可见, 阻尼比ξ和时滞稳定裕度τ为互相制约的指标, 当SDC参数$(K_{\rm sdc},T_1,T_2)$使ξ增大时会减少系统所能承受的输入信号传输延时, 且PAPSO算法能够获得比SPEA算法更好的优化结果.

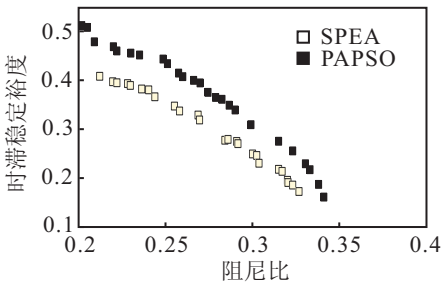

图 3 UPFC辅助阻尼控制多目标优化结果

由于在通信线路阻塞时SDC输入信号传输时滞可高达500 ms, 选取$(\xi_1,\tau_1)=(0.201\,8,0.513\,4)$进行仿真分析验证PAPSO算法计算结果的正确性, 并选取SPEA算法所能获得的具有最大时滞稳定裕度值$(\xi_2,\tau_2)=(0.212\,1,0.409\,5)$进行仿真分析. 设置扰动模式为联络线在1.0 s时三相短路, 50 ms后线路重新合闸成功. 在不同时滞影响下的联络线有功功率ΔP响应如图4所示. 从图4可以看出, 当信号传输时滞h不大于0.405 9 s时, 通过PAPSO算法和SPEA算法得到的SDC控制器均能使ΔP稳定, 且前者在时滞情况下稳定性能更好. 当信号传输时滞h达到0.513 4 s时, 由于$\tau_2<h\leqslant\tau_1$, 通过SPEA算法得到的SDC控制器已不能维持ΔP的稳定, 但根据PAPSO算法得到的SDC控制器仍然可以使系统在略多于10 s的时间内稳定. 当信号传输时滞h为0.6 s时, 由于$h>\tau_1$, ΔP失去稳定. 图4表明, PAPSO算法可以找到比SPEA算法更优的解.

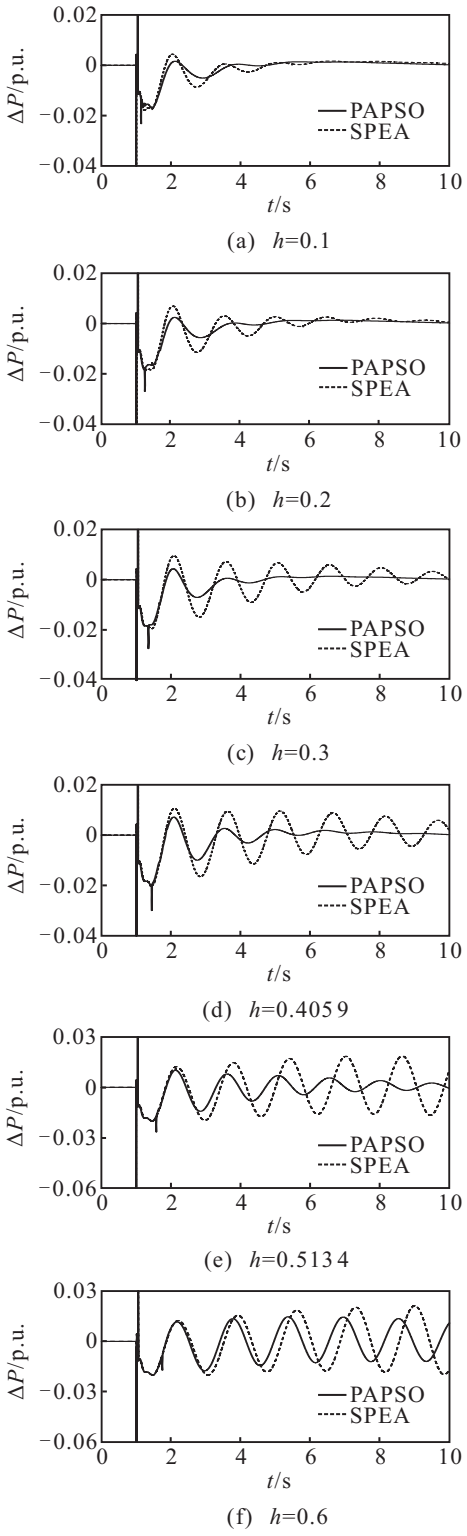

图 4 三相短路时在不同时滞下联络线有功功率响应

4 结　论

本文提出了一种基于相角映射的改进多目标粒子群优化算法, 该算法将标准PSO算法的位置映射为相角, 实现了粒子在解空间上仅依赖于归一化多目标函数的快速搜索; 采用共享池信息交换和混沌变异策略增加了算法计算结果的多样性, 避免了算法的早熟问题. 标准测试函数的数值实验结果表明, 本文所提出的PAPSO算法能快速逼近Pareto前沿并能保持解的高度多样性. 电力系统广域阻尼控制多目标优化算例表明了该算法的可行性和有效性. 尽管与其他算法相比, PAPSO算法可以取得更优的解, 但PAPSO算法为了保持种群多样性采用的各种改进策略也增加了计算开销. 因此, 在求解复杂的实际工程多目标优化问题时, 如何提高PAPSO算法计算效率和计算结果的均匀度值得进一步研究.

参考文献(References)

[1] Panigrahi B K, Tiwari M K. Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch[J]. IEEE Trans on Evolutionary Computation, 2008, 12(5): 529-541.

[2] 李勇, 吴敏, 曹卫华, 等. 基于线性规划和遗传-粒子群算法的烧结配料多目标综合优化方法[J]. 控制理论与应用, 2011, 28(12): 1740-1746.
(Li Y, Wu M, Cao W H, et al. A multi-objective optimization algorithm for sintering proportion based on linear programming and genetic algorithm particle swam optimization[J]. Control Theory & Applications, 2011, 28(12): 1740-1746.)

[3] 丁雷, 吴敏, 曹卫华, 等. 基于混合粒子群算法的铅锌烧结过程产量质量优化[J]. 中国有色金属学报, 2008, 18(6): 1152-1158.
(Ding L, Wu M, Cao W H, et al. Quantity and quality optimization for Lead-Zinc sintering process based on hybrid PSO algorithm[J]. The Chinese J of Nonferrous Metals, 2008, 18(6): 1152-1158.)

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

[5] 吴敏, 丁雷, 曹卫华, 等. 一种克服粒子群早熟的混合优化算法[J]. 控制与决策, 2008, 23(5): 511-514.
(Wu M, Ding L, Cao W H, et al. A kind of hybrid optimization algorithm with prevention of premature convergence of particle swarm[J]. Control and Decision, 2008, 23(5): 511-514.)

[6] 龙文, 梁昔明, 肖金红, 等. 一种动态分级的混合粒子群优化算法[J]. 控制与决策, 2009, 24(10): 1513-1516.
(Long W, Liang X M, Xiao J H, et al. Dynamic hierarchical hybrid particle swarm optimization algorithm[J]. Control and Decision, 2009, 24(10): 1513-1516.)

[7] Hu X, Eberhart R. Multiobjective optimization using dynamic neighborhood particle swarm optimization[C]. Proc of the 2002 Congress on Evolutionary Computation. New York: IEEE Press, 2002: 1677-1681.

[8] Hu X, Eberhart R, Shi Y. Particle swarm with extended memory for multiobjective optimization[C]. Proc of the 2003 IEEE Swarm Intelligence Symposium. New York: IEEE Press, 2003: 193-197.

[9] Coello 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.

[10] Agrawal S, Dashora Y, Tiwari M K, et al. Interactive particle swarm: A Pareto-adaptive metaheuristic to multiobjective optimization[J]. IEEE Trans on Systems, Man and Cybernetics, Part A: Systems and Humans, 2008, 38(2): 258-277.

[11] Limbourg P, Mehnen J, Schmitt K, et al. Particle swarm optimizers for Pareto optimization with enhanced archiving techniques[C]. Proc of the 2003 Congress on Evolutionary Computation. New York: IEEE Press, 2003: 1780-1787.

[12] Zhang Y, Gong D W, Ding Z H. A bare-bones multi-objective particle swarm optimization algorithm for environmental economic dispatch[J]. Information Sciences, 2009, 192(4): 213-227.

[13] Mostaghim S, Teich J. Strategies for finding good local guides in multi-objective particle swarm optimization[C]. Proc of the 2003 IEEE Swarm Intelligence Symposium. New York: IEEE Press, 2002: 26-33.

[14] 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.

[15] Tripathi P K, Bandyopadhyay Sanghamitra, Pal S K. Adaptive mufti-objective particle swarm optimization algorithm[C]. Proc of the 2007 Congress on Evolutionary Computation. New York: IEEE Press, 2007: 2281-2288.

[16] He Y, Wang Q G, Lin C, et al. Delay-range-dependent stability for systems with time-varying delay[J]. Automatica, 2007, 43(2): 371-376.