

文章编号: 1001-0920(2009)12-1851-05

## 自适应进化多目标粒子群优化算法

陈民铀, 张聪誉, 罗辞勇

(重庆大学 a. 电气工程学院, b. 输配电装备及系统安全与新技术国家重点实验室, 重庆 400044)

**摘要:** 提出一种自适应进化粒子群优化算法以求解多目标优化问题. 采用非支配排序策略和动态加权法选择最优粒子, 引导种群飞行, 提高 Pareto 解的多样性. 采用动态惯性权重, 提高其全局寻优能力. 当种群的寻优能力减弱时, 采用变异操作以引导粒子群跳出局部最优. 通过  $ZDT_1 \sim ZDT_4$  基准函数验证, 该算法能够在保持优化解多样性的同时实现较好的收敛性. 与其他多目标进化算法和多目标粒子群优化算法相比, 该算法具有较好的性能.

**关键词:** 多目标优化; 粒子群算法; 非支配排序; 拥挤距离; 动态加权法

中图分类号: TP18

文献标识码: A

## Adaptive evolutionary multi-objective particle swarm optimization algorithm

CHEN Min-you, ZHANG Cong-yu, LUO Ci-yong

(a. School of Electrical Engineering, b. State Key Laboratory of Power Transmission Equipment and System Security and New Technology, Chongqing University, Chongqing 400044, China. Correspondent: LUO Ci-yong, E-mail: luociyong@cqu.edu.cn)

**Abstract:** Adaptive evolutionary particle swarm optimization (AEPSO) algorithm is proposed for multi-objective problem. Non-dominated sorting and dynamic aggregate method are used to guide the flight of particles and improve the diversity of the Pareto solutions, which incorporates adaptive inertia weight and a special mutation operation into PSO to enhance the global exploratory capability and avoid pre-mature convergence. Four well-known benchmark test functions  $ZDT_1 \sim ZDT_4$  are used to test the performance of the proposed algorithm. Simulation results show that the AEPSO algorithm can converge to the global optimal with good accuracy while keeps the diversity of the Pareto solutions.

**Key words:** Multi-objective optimisation; PSO; Nondominated sorting; Crowding distance; Dynamic aggregate method

### 1 引言

粒子群优化(PSO)是近年来提出的一种启发式搜索技术,通过种群粒子中个体的交互作用来寻找复杂问题空间中的优化解. PSO 算法随机产生一个初始种群并赋予每个微粒一个随机速度,在飞行过程中,微粒的飞行速度和轨迹根据自己及同伴的飞行经验来动态调整,整个群体有飞向更好搜索区域的能力. 目前 PSO 算法已成功应用于众多领域的优化问题,并被证明能够以较小的计算代价获得良好的优化解<sup>[1]</sup>.

2006 年, Sierra 和 Colleo<sup>[1]</sup> 全面归纳总结了各种多目标 PSO(MOPSO)算法,指出尽管 PSO 算法

在单目标领域已获得很大的成功,但它并不能直接用于多目标问题. 他们认为,将 PSO 扩展为 MOPSO 需要解决以下问题:1)如何在搜索过程中产生非支配解并构成 Pareto 解集,这些解不仅对当前种群是非支配的,而且对所有的过去种群也是非支配的;2)采用什么策略从当前非支配解集中选择全局最优(或局部最优)粒子;3)如何保持 Pareto 前沿上优化解的多样性.

Parsopoulos 等<sup>[2]</sup> 率先采用加权法(固定权重、动态改变权重和 bang-bang 加权)将多目标问题转化为单目标问题. 其后 Parsopoulos 等<sup>[3]</sup> 又受向量评价遗传算法(VEGA)启发,提出了应用于多目标

收稿日期: 2009-02-08; 修回日期: 2009-09-09.

基金项目: 国家 111 引智计划项目(B08036); 重庆市自然科学基金项目(CSTC2006BB2238).

作者简介: 陈民铀(1954—),男,重庆人,教授,博士生导师,从事智能控制、数据建模等研究; 张聪誉(1982—),男,重庆人,硕士生,从事多目标优化、电力系统优化的研究.

的向量评价粒子群算法(VEPSO),每个子种群只根据单个目标函数产生,而不考虑其他目标函数,各种种群间通过交换最优粒子相互通信. 2002年,Deb等<sup>[4]</sup>提出一种非支配排序遗传算法(NSGA-II),对MOPSO的研究产生了重大影响. Li<sup>[5]</sup>把非支配排序的思想引入PSO算法中,并从非支配解集中随机选择粒子作为最优粒子来引导粒子群飞行. 为了放宽非支配排序,Laumanns等<sup>[6]</sup>提出了 $\epsilon$ 非支配排序的概念. Mostaghim等<sup>[7]</sup>的研究工作表明, $\epsilon$ 非支配排序可以提高计算速度,多数情况下可以获得更好的收敛性和多样性. Sierra和Colleo<sup>[8]</sup>在MOPSO中引入了 $\epsilon$ 支配概念,同时采用变异操作和小生境技术,从而提高了算法的收敛性能. Colleo等<sup>[1,8-10]</sup>在PSO和MOPSO领域做了大量的研究工作. 目前MOPSO算法主要采用NSGA-II非支配排序策略,并根据拥挤程度对粒子进行筛选;在选择最优粒子方面,有的采用随机选择,有的采用轮盘赌的方式,有的则根据非支配解的疏密程度进行选择;在保持种群多样性方面,主要采用变异操作(在速度或位置上变异),但不恰当的变异往往会带来不确定性<sup>[1-13]</sup>.

本文针对多目标优化问题,提出一种自适应进化粒子群优化(AEPSO)算法,主要作了如下改进:1)采用动态加权法选择最优粒子,引导种群飞行,提高Pareto解的多样性;2)采用动态惯性权重,提高全局寻优能力;3)当种群飞行速度减弱时采用变异操作以引导粒子群跳出局部最优. 通过ZDT<sub>1</sub>~ZDT<sub>4</sub>测试函数验证了AEPSO算法在多样性情况下具有较好的收敛性能.

## 2 自适应多目标PSO算法

### 2.1 Pareto最优解评价

解决多目标优化问题的方法之一是将多个目标函数 $f_i(x)$ 表示成一个目标函数 $h(x)$ . 这个转化过程称为聚合. 加权法就是其中常见的方法. 对于一个具有 $k$ 个目标的最小化问题,目标函数可表示为

$$h(x) = \min \sum_{i=1}^k w_i f_i(x). \quad (1)$$

采用该方法最大的好处在于,不用对粒子群算法作较大改动. 但加权法存在的问题是,优化结果与权重的选择有很大关系,如果Pareto前沿为非凸集,则很难得到Pareto最优集<sup>[2]</sup>.

在多目标向量评价粒子群算法中,每个子种群只根据单个目标函数产生,而不考虑其他目标函数,因此,VEPSO得到的解是局部非劣的,但不一定是全局非劣的. 即不同子种群中的个体仅在子种群范围内对某一单个目标进行优化. 这就导致了一种非

常不利的情形:在进化过程中,可能存在一个中等性能的个体对获得最后的折中解有利,但在子种群的选择中却被淘汰,因为它们在单个目标评价中不是最优的<sup>[3]</sup>.

非支配排序遗传算法(NSGA-II)是目前公认高效的多目标算法. NSGA-II中非支配排序思想已成为目前MOPSO构成Pareto最优解的主流方法. 文献<sup>[6]</sup>引入 $\epsilon$ 支配的概念,提高了算法的收敛速度. 但实验表明,采用 $\epsilon$ 支配并不能提高最终Pareto解集距离真实Pareto前沿的接近程度,有时甚至会使用这一指标变差<sup>[7]</sup>.

NSGA-II中的非支配排序策略,其操作过程如图1所示<sup>[4]</sup>. 设种群规模为 $N$ . 首先由当前支配解集 $P_t$ 和当前的子代 $Q_t$ 共同组合种群 $R_t$ (种群大小为 $2N$ ).  $R_t$ 根据支配关系得到一系列非支配Pareto解集,依次为 $F_1, F_2, F_3$ 等.  $F_1$ 级别最高,如果 $F_1$ 的数量小于 $N$ ,则把 $F_1$ 的成员全部选择到种群 $P_{t+1}$ 中.  $P_{t+1}$ 的剩下成员将在 $F_2, F_3$ 等中选择,直到数量为 $N$ 时为止.

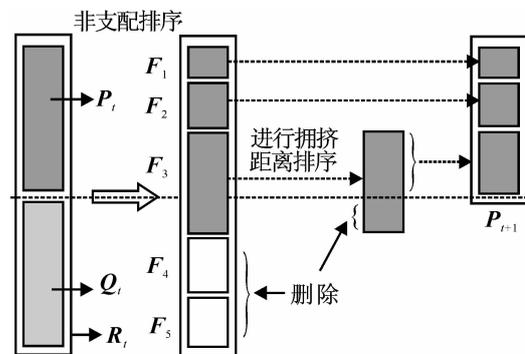


图1 NSGA-II排序过程

在图1中, $F_3$ 集合的第1个成员次序小于 $N$ ,而最后一个成员次序大于 $N$ . 为了保持种群多样性,NSGA-II中需要对 $F_3$ 进行拥挤距离排序. 拥挤距离大的个体优先,其拥挤距离的计算复杂度为 $O(M(2N)\log(2N))$ , $M$ 是目标的数量.

本文通过对比,选择NSGA-II中的非支配排序策略来构成Pareto最优解集.

### 2.2 最优粒子的选择

应用PSO求解多目标问题时,如何选择最优引导粒子需要仔细考虑. 在当前Pareto非支配解集中(图1中的 $P_{t+1}$ )采用随机方式选择最优粒子,使得较低前沿的非支配粒子和最前沿的非支配粒子拥有相同的概率. 而只在最前沿粒子中进行选择,不利于种群的多样性.

如前所述,加权法在构成Pareto解集时存在一些缺陷. 但对于已形成的Pareto解集,动态加权法可以形成动态的评价. 本文采用动态加权法,按下式

动态计算 Pareto 解集中各粒子适应度:

$$\text{fitness} = 1 / \sum_{i=1}^M \omega_i f_i(\mathbf{x}), \quad \sum_{i=1}^M \omega_i = 1. \quad (2)$$

当前动态适应度最大的粒子就是全局最优粒子。

个体最优粒子, 根据支配和非支配关系来决定。如果当前粒子支配个体最优粒子, 则更新个体最优粒子; 如果个体最优粒子不支配当前粒子, 则按 50% 的概率更新最优粒子; 如果个体最优粒子支配当前粒子, 则不更新。

### 2.3 保持种群多样性策略

1995 年, Kenney 和 Eberhart<sup>[14]</sup> 提出了 PSO 算法, 随后 Shi 和 Eberhart<sup>[15]</sup> 引入惯性权重对算法进行了改进, 参数  $\omega$  根据迭代次数线性减小。引入惯性权重  $\omega$  来更好地控制开拓和探索能力。在每次迭代过程中, 粒子的位置由下式确定:

$$v_i(t) = \omega v_i(t-1) + c_1 r_1 (p_i - x_i(t-1)) + c_2 r_2 (p_g - x_i(t-1)), \quad (3)$$

$$x_i(t) = v_i(t) + x_i(t-1). \quad (4)$$

式中:  $x_i$  为粒子位置;  $v_i$  为粒子飞行速度;  $\omega$  为惯性权重;  $c_1, c_2$  为加速度常数;  $r_1, r_2$  为在  $[0, 1]$  范围变化的随机数;  $p_i$  为个体最优粒子的位置;  $p_g$  为全局最优粒子的位置。

PSO 算法存在一个固有问题, 即按式(3), (4) 计算时, 粒子群总是向全局最优、个体最优飞行, 经过若干次迭代后, 可能发生所有粒子的位置都趋于同一个局部最优解的情况。这时所有粒子的速度趋于 0, 粒子的位置不再发生变化, 即发生早熟收敛。为克服早熟收敛, 提高 PSO 算法的开拓能力并保持快速的收敛速度, 特别考虑多目标优化问题, 本文将自适应惯性权重和变异操作引入粒子群优化算法。将式(3) 修改如下:

$$v_i(t+1) = \omega v_i(t) + \alpha [r_1 (p_i - x_i(t)) + r_2 (p_g - x_i(t))]. \quad (5)$$

惯性权重  $\omega$  是控制先前速度对当前速度的冲击, 其目的是平衡全局和局部搜索性能。大的惯性权重有助于增强全局开拓(搜索更大范围), 而小的惯性权重则倾向于在当前位置进行局部搜索。本文采用动态变化惯性权重, 在每一次迭代时, 惯性权重由下式自适应调整:

$$\omega = \omega_0 + r(\omega_1 - \omega_0). \quad (6)$$

式中:  $\omega_0 \in [0, 1], \omega_1 > \omega_0$  均为常数;  $r$  是在  $[0, 1]$  分布的随机数。  $\omega_0$  建议的范围是  $[0, 0.5]$ 。式(6) 使得惯性权重在  $[\omega_0, \omega_1]$  之间随机变化。

式(5) 右边的第 2 项为加速项, 由当前粒子位置与个体最优位置和全局最优位置的距离决定。加速因子  $\alpha$  定义为

$$\alpha = \alpha_0 + t/N_t, \quad t = 1, 2, \dots, N_t. \quad (7)$$

式中:  $N_t$  为总的迭代次数,  $t$  为当前代数,  $\alpha_0$  的范围是  $[0.5, 1]$ 。式(7) 中加速因子随迭代次数增加而增加, 在运行后期会提高全局搜索能力, 特别是对于多峰问题有助于跳出局部最优解。

采用动态惯性权重和线性加速因子, 尽管可以增强搜索能力, 但经过一定次数的迭代后, 粒子的飞行速度越来越小, 仍会发生早熟收敛。为了克服这一问题, 本文采用变异操作来保持粒子群的多样性: 当整群粒子飞行速度很小时(低于  $V_{\text{limit}}$ ), 在粒子位置上进行变异。变异操作是在随机选择粒子某一维位置上进行, 具体操作如下:

$$v_m = 2(r_3 - 1)\beta V_{\text{max}},$$

$$x_i^d(t) = x_i^d(t) + v_m. \quad (8)$$

式中:  $v_m$  为变异值;  $\beta \in [0, 1]$  为变异系数, 用于调节变异程度;  $r_3$  为在  $[0, 1]$  范围变化的随机数;  $x_i^d$  为第  $i$  个粒子的随机选中的第  $d$  维。

变异操作不仅能够增强粒子群的全局搜索能力, 而且有助于保持种群的多样性。通过变异操作引导粒子群在邻域内随机飞行, 一旦某个粒子成为新的支配粒子, 便会吸引其他粒子飞过去。

由于本文采用动态惯性权重, 并且在寻优过程中群体速度降低时引入变异操作, 使得种群具有自适应进化功能, 从而构成自适应进化粒子群优化(AEPSO) 算法。

### 2.4 完整的 AEPSO 算法流程

AEPSO 多目标优化算法可归纳如下:

Step1: 初始化。设定初始种群  $\mathbf{P}$ , 规模为  $N$ , 迭代次数  $N_t$ 。用预定义的值域随机初始化粒子位置。  $V_{\text{max}}$  设置为最大允许速度。初始化个体最优  $p_i$  和全局最优  $p_g$ 。迭代计数器  $t = 0$ 。

Step2: 评价。评价当前种群  $\mathbf{P}$  中所有粒子的各目标值。

Step3: 更新领导者。  $t = t + 1$ 。按照 2.2 节的策略更新种群  $\mathbf{P}$  中各粒子的个体最优位置  $p_i$ , 并更新全局最优位置  $p_g$ 。

Step4: 产生新的粒子。拷贝  $\mathbf{P}$  以形成新种群  $\mathbf{Q}$ , 按照式(5) 和(4) 计算种群  $\mathbf{Q}$  中各粒子新的速度和位置, 并计算  $\mathbf{Q}$  中每个粒子的各目标值。组合种群  $\mathbf{P}$  和  $\mathbf{Q}$  以构成种群  $\mathbf{R}$  (共  $2N$  个粒子)。

Step5: 种群  $\mathbf{R}$  排序。采用 NSGA-II 的非支配排序策略法, 对种群  $\mathbf{R}$  进行非支配排序和拥挤距离排序。

Step6: 选择下一代粒子。从种群  $\mathbf{R}$  中按顺序选择  $N$  个粒子到种群  $\mathbf{P}$  中。

Step7: 变异操作。对种群  $\mathbf{P}$  中粒子, 判断其整

体速度是否小于临界值  $V_{\text{limit}}$ . 若是, 则按下列步骤执行变异操作:

- 1) 将种群  $P$  备份到  $P_{\text{temp}}$ ;
- 2) 按式(8) 计算  $P_{\text{temp}}$  中粒子的  $x(t)$ ;
- 3) 计算  $P_{\text{temp}}$  中粒子各目标值, 判别是否为非支配解, 若是, 则用变异粒子取代种群  $P$  中对应的原粒子.

Step8: 如果  $t < N_t$ , 则转 Step2.

Step9: 输出种群  $P$  作为最终的 Pareto 支配解.

### 3 性能验证

#### 3.1 测试函数和指标

为验证 AEPSo 多目标优化算法的性能, 选择 4 种具有代表性的多目标优化标准测试函数  $ZDT_1 \sim ZDT_4$  [4,16]. 其中:  $ZDT_1$  的 Pareto 前沿为凸函数,  $ZDT_2$  的为非凸函数,  $ZDT_3$  的为非连续函数, 而  $ZDT_4$  则是一个多峰函数.

与单目标优化不同, 多目标优化需同时满足两点: 1) 收敛到真实的 Pareto 解集; 2) 保持 Pareto 解的多样性. 本文采用 Deb [4] 提出的收敛度  $\gamma$  和多样性  $\Delta$  来评价这两个性能.

用收敛度  $\gamma$  测算 Pareto 解集  $P$  和已知 Pareto 最优解  $P^*$  的距离. 首先从真实的 Pareto 前沿上选取均匀分布的 500 个解构成  $P^*$ ; 然后计算 Pareto 解集  $P$  每个解与  $P^*$  中最近解之间的欧氏距离, 这些距离的平均值即为收敛度  $\gamma$ .

多样性  $\Delta$  定义如下:

$$\Delta = \frac{d_f + d_l + \sum_{j=1}^{N-1} |d_j - \bar{d}|}{d_f + d_l + (N-1)\bar{d}}. \quad (9)$$

式中:  $d_i$  是在 Pareto 解集  $P$  中相邻解的距离;  $\bar{d}$  是所有  $d_i$  的平均值;  $d_f$  和  $d_l$  是 Pareto 解集  $P$  的边界解与  $P^*$  中最极端解之间的欧氏距离.

#### 3.2 用于对比的多目标优化算法及其参数设置

用于对比的多目标优化算法包括: NSGA-II [4], PESA-II [17], NSPSO [5], MOPSO [9], RM-MEDA [18]. NSGA-II 和 NSPSO 种群大小是 100, 迭代次数是 250 次; PESA-II 种群大小是 10, 迭代次数是 2500 次; MOPSO 种群大小是 50, 迭代次数是 500 次; 函数评价都是 25000 次. 输出的 Pareto 解集大小均是 100. NSGA-II 和 PESA-II 中  $p_c$  取 0.9,  $p_m = 1/n$ . NSPSO 中  $c_1 = 1, c_2 = 2, \omega$  是从 1.0 线性下降到 0.4.

本文 AEPSo 算法参数经验地选取为:  $\omega_0 = 0.35, \omega_1 = 1.0, \alpha_0 = 0.5, V_{\text{limit}} = 0.2, \beta = 0.1$ , 种群规模  $N = 100$ , 迭代次数  $N_t = 250$ , 算法重复运行 30 次. 算法采用 C++ 编程, 仿真实验在 CPU 为 1.6 G

的 PC 机上进行.

#### 3.3 实验结果

各种多目标优化算法的实验结果见表 1 和表 2 (表中: M 为平均值, VAR 为方差). NSGA-II 的结果来自于文献 [4]; PESA-II, NSPSO, MOPSO 的实验结果来自于 [11]; RE-MEDA 的计算结果是采用 [18] 提供的程序运算后获得的.

表 1 收敛度  $\gamma$  比较

算法		ZDT <sub>1</sub>	ZDT <sub>2</sub>	ZDT <sub>3</sub>	ZDT <sub>4</sub>
NSGA-II	M	0.03348	0.07239	0.11450	0.51305
	VAR	0.00475	0.03168	0.00794	0.11846
PESA-II	M	0.00105	0.00074	0.00789	9.98254
	VAR	0.00000	0.00000	0.00011	20.1340
NSPSO	M	0.00642	0.00951	0.00491	4.95775
	VAR	0.00000	0.00000	0.00000	7.43601
MOPSO	M	0.00133	0.00089	0.00418	7.37429
	VAR	0.00000	0.00000	0.00000	5.48286
RM-MEDA	M	0.02121	0.02919	0.04418	53.70333
	VAR	0.00002	0.00017	0.00014	5.37964
AEPSo	M	0.00100	0.00078	0.00462	0.49800
	VAR	0.00000	0.00000	0.00000	1.93606

表 2 多样性  $\Delta$  比较

算法		ZDT <sub>1</sub>	ZDT <sub>2</sub>	ZDT <sub>3</sub>	ZDT <sub>4</sub>
NSGA-II	M	0.39031	0.43077	0.73854	0.70261
	VAR	0.00187	0.00472	0.01971	0.06465
PESA-II	M	0.84816	0.89292	1.22731	1.01136
	VAR	0.00287	0.00574	0.02925	0.00072
NSPSO	M	0.90695	0.92156	0.62072	0.96462
	VAR	0.00000	0.00012	0.00069	0.00156
MOPSO	M	0.68132	0.63922	0.83195	0.96194
	VAR	0.01335	0.00114	0.00892	0.00114
RM-MEDA	M	0.38590	0.36560	0.81103	0.76447
	VAR	0.00057	0.00531	0.00404	0.00415
AEPSo	M	0.55743	0.51901	0.55182	0.80367
	VAR	0.00142	0.00095	0.00087	0.07078

图 2 显示了 AEPSo 与 RM-MEDA 算法各自在  $ZDT_1 \sim ZDT_4$  测试函数上重复运行 30 次所得到的 Pareto 解.

### 4 讨 论

表 1 显示, 与 PESA-II, NSPSO, MOPSO 和 RM-MEDA 相比, NSGA-II 提高了  $ZDT_1 \sim ZDT_3$  的收敛度, 但在  $ZDT_4$  问题上算法的收敛度较差; RM-MEDA 的平均收敛度达到了 53.7. 而 AEPSo 在  $ZDT_4$  的收敛指标与 NSGA-II 的类似, 但在  $ZDT_1 \sim ZDT_3$  则优于 NSGA-II. 在多样性指标方面, AEPSo 的性能适中, 优于 PESA-II, NSPSO 和 MOPSO, 而低于 NSGA-II 和 RM-MEDA.

文献 [18] 为了对比多目标优化输出结果, 提出了将重复实验的多组 Pareto 解叠加放在一起显示, 以对比算法的收敛性和多样性. 以往的文献通常显示最佳的一组 Pareto 前沿. RM-MEDA 是一种启发

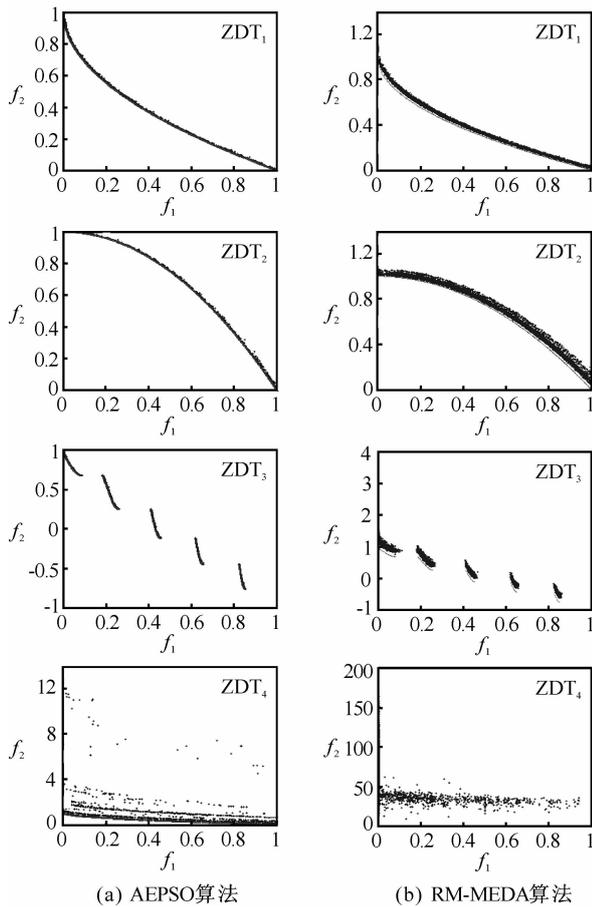


图 2 ZDT1 ~ ZDT4 的 30 组 Pareto 优化解

式多目标优化算法。图 2 表明：RM-MEDA 在 ZDT<sub>1</sub> ~ ZDT<sub>3</sub> 上的多样性较好（分布均匀），但收敛度较差；AEPSo 在 ZDT<sub>1</sub> ~ ZDT<sub>3</sub> 上收敛性较好（更接近真实 Pareto 前沿），收敛方差也小，但存在的问题是多样性不好，30 组 Pareto 前沿叠加后其稀疏程度不一致。

ZDT<sub>4</sub> 有 2<sup>19</sup> 个 Pareto 前沿<sup>[4]</sup>，其中只有 1 个是全局最优 Pareto 前沿，而其他均为局部优化 Pareto 前沿。图 2 显示较多的 AEPSo 最终解接近真实 Pareto 前沿，在 30 组 Pareto 解中有 13 组（43%）的收敛度  $\gamma$  小于 0.01；而 RM-MEDA 的收敛性能明显劣于 AEPSo。文献[4]给出了 NSGA-II 在 ZDT<sub>4</sub> 上的最优 Pareto 前沿（见图 3），其收敛度  $\gamma$  的值约为 0.5。

表 3 给出了动态惯性权重、固定惯性权重和递减惯性权重的实验结果对比。其中：迭代次数为 250，线性递减惯性权重参数  $\omega$  取 [0.9 ~ 0.4]，固定权重参数  $\omega$  取 0.5，所列数据均为 30 组实验数据的平均值，其他算法参数与前相同。

表 3 中，动态惯性权重的收敛性和多样性比其他策略稍好一些，证实了动态惯性权重对全局寻优的贡献。由于其他策略如变异、拥挤距离、非支配排

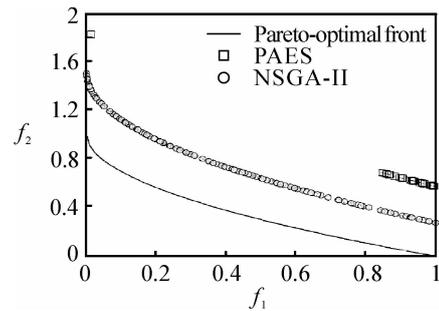


图 3 NSGA-II 和 PAES 在 ZDT<sub>4</sub> 的 Pareto 解

表 3 不同惯性权重的寻优结果比较

惯性权重		ZDT <sub>1</sub>	ZDT <sub>2</sub>	ZDT <sub>3</sub>
动态惯性权重	$\gamma$	0.00100	0.00078	0.00462
	$\Delta$	0.55743	0.51901	0.55182
固定惯性权重	$\gamma$	0.00102	0.00081	0.00485
	$\Delta$	0.56136	0.52773	0.55281
递减惯性权重	$\gamma$	0.00108	0.00081	0.00479
	$\Delta$	0.55983	0.52264	0.57842

序等策略的作用，几种惯性权重的结果都很接近。由此证明了 AEPSo 对于惯性权参数变化不敏感，进而显示了该算法对惯性权重参数的鲁棒性。

### 5 结 论

本文提出了 AEPSo 算法，并将其用于求解多目标问题。AEPSo 采用 NSGA-II 的非支配排序策略，用动态加权法选择最优粒子，引导种群飞行；用动态权重避免粒子早熟；当种群的搜索能力下降时，采用变异操作引导粒子群跳出局部最优。通过 ZDT<sub>1</sub> ~ ZDT<sub>4</sub> 函数验证了 AEPSo 算法能够在保证多样性的同时具有较好的收敛性，能有效地解决多目标优化问题。

尽管 AEPSo 算法比其他算法具有较好的收敛性，但在 ZDT<sub>4</sub> 上尚未获得理想、稳定的收敛解。因此改进在 ZDT<sub>4</sub> 上的收敛性能，值得进一步研究。另外 AEPSo 算法的多样性也有待于提高。

### 参考文献 (References)

[1] Sierra M R, Coello C A C. Multi-objective particle swarm optimizers: A survey of the state-of-the-art[J]. Int J of Computational Intelligence Research, 2006, 2 (3): 287-308.

[2] Parsopoulos K E, Vrahatis M N. Particle swarm optimization in multiobjective problems[C]. Proc of the ACM 2002 Symposium on Applied Computing. Madrid, 2002: 603-607.

[3] Parsopoulos K E, Tasoulis D K, Vrahatis M N. Multi-objective optimization using parallel vector evaluated particle swarm optimization[C]. Proc of the IASTED Int Conf on Artificial Intelligence and Applications, Innsbruck, 2004: 823-828. (下转第 1864 页)