

带收缩与发散操作的自适应粒子群优化算法

赵志刚 尹兆远 林玉娇

(广西大学计算机与电子信息学院 南宁 530004)

摘要 提出一种结合收缩与发散操作的自适应粒子群算法,其在运行过程中通过判断种群收敛情况与进化情况,自适应地选择粒子的运动行为。通过收缩操作使群体向极值点快速靠拢,通过发散操作保证粒子有能力跳出局部极值点,并根据种群进化状况在两种操作间转换。实验表明,该算法具有较强的跳出局部极值、逼近全局最优解的能力,在高维多峰函数上有较出色的表现。

关键词 粒子群优化算法,收缩操作,发散操作,自适应检测

中图法分类号 TP18 文献标识码 A

Adaptive Particle Swarm Optimization Algorithm with Shrink and Expansion Operation

ZHAO Zhi-gang YIN Zhao-yuan LIN Yu-jiao

(College of Computer and Electronics Information, Guangxi University, Nanning 530004, China)

Abstract An adaptive particle swarm optimization algorithm with shrink and expansion operation was presented, which can adaptively choose the behavior of the particle by detecting the degree of convergence during running. These two operations make particle swarm converge to extreme point and jump out of it quickly, and the evolutionary status of the population make them convert between these two operations adaptively. Experimental data show this algorithm has strong ability to get rid of the local optima and approach to global optima, especially in tackling the problem of high dimension multimodal function.

Keywords Particle swarm optimization, Shrink operation, Expansion operation, Adaptive detection

粒子群优化(Particle Swarm Optimization, PSO)算法于1995年提出以来^[1],以结构简单、需要控制和调整的参数较少、收敛速度较快、算法容易实现等优点而受到广泛关注,但也因为收敛精度不高、易陷入局部最优等问题,使许多学者一直致力于对其进行改进。Shi等提出惯性权重的概念^[2],以权重线性递减的方式平衡全局探索与局部发掘之间的能力。文献^[3-5]在算法中引入变异因子,通过变异因子的扰动操作使粒子群在陷入局部最优解时有一定概率跳出局部最优解继续进化。文献^[6,7]通过引入吸引和排斥机制,使种群多样性维持在一定限度之内,并根据种群收敛情况控制发散和收缩的时机,使算法能逐渐往全局最优逼近。以上所述的改进主要集中在平衡全局搜索与局部搜索能力上。本文提出一种粒子群改进算法,较大程度克服了粒子群局部收敛的缺点,将该算法用于高维多峰函数求解,获得了较好的效果。

1 标准粒子群优化算法

PSO算法模拟鸟群飞行行为,将其中飞行的鸟看作“粒子”,每个粒子都有速度和位置两个参数,并使用适应度函数值来评价当前位置的优劣,粒子通过跟踪个体极值和全局极值两个吸引极来更新自己的速度和位置,通过迭代向最优解逼近。

在Kennedy和Eberhart最早提出的PSO算法^[1]的速度项中,Shi等增加惯性权重^[2],将速度与位置更新公式作如下改进:

$$v_i(t+1) = \omega(t)v_i(t) + c_1r_1(pBest_i(t) - x_i(t)) + c_2r_2(gBest(t) - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

其中, v 为速度项, x 为位置项, i 为当前粒子编号, t 为当前迭代次数; c_1 和 c_2 为学习因子; r_1 、 r_2 是分布在 $[0,1]$ 内的随机数; $pBest$ 为单个粒子本身找到的个体极值; $gBest$ 为整个粒子群当前找到的全局极值。 $\omega(t)$ 随着迭代次数的增加逐渐减小。以上算法一般被称为标准粒子群算法(PSO)。

2 改进的粒子群优化算法

本文提出的算法采用一种较激进的搜索策略,核心思想为:“快速收敛至极优解”和“频繁跳出局部最优解”两种操作交替进行,通过多次发散与收敛来逐步逼近全局最优解。详细描述如下。

2.1 收缩操作

此操作的主要作用为向最有可能存在极优解的区域快速聚集粒子,进行集中搜索,以期尽早发现该区域内更好的解。触发收缩操作的条件为粒子群收敛至给定阈值 $avgGPT$ 。计

本文受国家自然科学基金项目(61363067)资助。

赵志刚(1973—),男,博士,副教授,主要研究方向为智能优化计算,E-mail:zzg@mail2002@163.com;尹兆远(1988—),男,硕士生,主要研究方向为粒子群优化算法,E-mail:zacchues@qq.com;林玉娇(1989—),女,硕士生,主要研究方向为粒子群优化算法,E-mail:273565629@qq.com。

算公式如下:

$$avgGP = \frac{1}{n} \sqrt{\sum_{i=1}^n (gBest - x_i)^2} \quad (3)$$

其中, n 为粒子总数。若 $avgGP < avgGPT$, 则启动收缩操作。收缩操作首先计算收缩半径 cR , 并以当前 $gBest$ 为圆心, 计算落在此超圆范围内的粒子, 称在此超圆内的粒子为中心粒子, 超圆外的粒子为外围粒子。本轮迭代的收缩半径 cR 根据上一轮迭代中的中心粒子 x_i^c 与 $gBest$ 的平均距离计算得出, 公式如下:

$$cR = \frac{1}{m} \sqrt{\sum_{i=1}^m (gBest - x_i^c)^2} \quad (4)$$

其中, m 为上轮中心粒子数量。中心粒子的速度和位置更新操作为在式(1)与式(2)的基础上增加式(5)与式(6):

$$v_i(t+1) = v_i(t+1) * r \quad (5)$$

$$x_i(t+1) = x_i(t) + (gBest(t) - x_i(t)) * (1-r) \quad (6)$$

其中, 式(1)中的惯性权重 $\omega(t')$ 采用文献[8]提出的修正公式:

$$\omega(t') = \omega_{min} + \exp(-(a * t' / T_{max})^2) * (\omega_{max} - \omega_{min}) \quad (7)$$

其中, a 为经验参数, 一般取值在 45 至 65 之间。 T_{max} 为一个与算法最大迭代次数 I_{max} 相区别的独立参数, 一般设置为 500。在粒子群覆盖极优解的前提下, 中心粒子群通过式(5)与式(6)的收缩操作快速向收缩中心塌缩, 增大收缩中心的粒子密度, 以提高寻找到新的最优解的概率; 而外围粒子只依据式(1)与式(2)飞行, 在 $gBest$ 的吸引下按正常速度向其靠拢。此时将会有两种情况: (1) 极优解落在收缩半径划出的超球范围内, $gBest$ 将由某个中心粒子找到并更新, 这会使 $gBest$ 与极优解进一步靠近, 由此说明搜索方向正确。则在下一轮迭代中, 继续根据式(4)计算选择半径, 在上轮中心粒子与接近 $gBest$ 的外围粒子中选择出本轮中心粒子, 重复进行收缩操作, 使得 $gBest$ 以尽可能快的速度接近极优解。此操作意味着中心粒子数量将逐渐减少, 在收缩比 r 较高时尤为突出。(2) 更优解落在收缩半径之外, 这有可能是因为在第一种情况下收缩半径因中心粒子的减少变得越来越小造成的。由此需要对收缩中心和收缩半径进行重新调整。假若外围粒子在 $gBest$ 的吸引下向收缩中心飞行的过程中“意外”发现更优的解, 说明当前收缩方向已经偏离了正确方向, 应当调整粒子群搜索方向使其往新发现的更优位置方向。此时算法将所有粒子标记为未选, 使本轮迭代剩下的粒子继续根据式(1)与式(2)飞行, 并在下一轮迭代按照所有粒子对 $gBest$ 的平均距离重新计算收缩半径。这意味着收缩半径的扩张, 重新选择大量粒子进行下一轮的收缩操作。

2.2 发散操作

因为收缩操作是以牺牲全局搜索能力为代价来增强局部发掘能力的, 当收缩操作开始后, 若经连续若干次迭代后全局最优解变化不大, 则可认为种群已经收敛至相当小的范围, 进化趋于停滞。为弥补收缩操作的不足, 启动发散操作, 以当前 $gBest$ 为期望中心, 全局最大搜索范围 X_{max} 为方差, 使用高斯分布重新初始化所有粒子的位置, 并更新 $pBest$ 为新位置, 速度随机生成。进化停滞判断公式根据实际问题中解的值域变化区间设置。本文如下:

$$deValue = |\lg(gBest(t)) - \lg(gBest(t-t_f))| \quad (8)$$

其中, $deValue$ 为第 $t-t_f$ 代与 t 代最优适应度的差值, 当低于

某一阈值 $deValueT$ 时, 则认为粒子群陷入局部最优。 $deValueT$ 一般统一取 0.001。发散操作的公式如下:

$$x_i(t) = gBest + N(\mu, \sigma) \quad (9)$$

$$pBest_i(t) = x_i(t) \quad (10)$$

其中, N 为高斯分布随机数生成器, 期望 $\mu=0$, 方差 $\sigma=X_{max}$ 。每次发散操作时需重新初始化惯性权重 $\omega(t')$ 的参数 t' 为 1。发散的粒子将重新围绕在 $gBest$ 周围, 在其吸引下从各个方向重新靠近 $gBest$, 并在满足收敛条件时重新进行收缩操作。在这一重新靠近的过程中, 所有粒子都成为外围粒子, 按照标准粒子群算法重新进行全局寻优, 从而使得跳出局部最优解的概率显著增加。

2.3 自适应检测

在迭代的过程中, 需要对粒子群的运算结果进行监测, 以随时调整粒子群的行为。需要进行识别的地方有: 全局探索至局部精细发掘的切换、对收缩半径的调整、局部精细发掘至全局探索的切换。全局探索至局部发掘切换时机的确定, 需要对粒子收敛情况进行判断, 以确定粒子群是否已收敛至可以开启收缩操作进行局部发掘的程度。如果转换过早, 会损失过多全局搜索能力, 增大粒子陷入局部最优解的概率; 反之则会浪费迭代次数, 影响最终结果。收缩半径 cR 调整的作用是保证同时存在合理数量的中心粒子和外围粒子协同参与局部发掘, 并能及时修正收缩方向以保证寻优效果。局部精细发掘至全局探索的切换, 需要对种群进化是否停滞进行判断, 以确定是否需要进行发散操作, 保证粒子不在局部最优解中耗费过多搜索次数。收缩和发散次数根据所优化问题特征自动调整。收缩半径调整操作和进化停滞判断具体操作在论述收缩和发散操作时均有提及, 此处限于篇幅, 不再赘述。

综合 2.1—2.3, 本文提出一种带收缩与发散操作的自适应粒子群优化算法 (Adaptive PSO with shrink and expansion, seAPSO)。该算法的步骤如下:

步骤 1 算法初始化。转步骤 2。

步骤 2 计算 $gBest$ 与所有粒子均距 $avgGP$, 若 $avgGP < avgGPT$, 转步骤 3; 否则按照式(1)与式(2)更新粒子速度与位置, 并更新个体最优 $pBest_i$ 与全局最优 $gBest$, 转步骤 8。

步骤 3 若当前第 t 代 $gBest$ 适应度对数值与第 $t-t_f$ 代 $gBest$ 适应度对数值差值小于给定阈值 $deValueT$, 转步骤 4; 否则转步骤 5。

步骤 4 若第 $i-1$ 轮迭代中 $gBest$ 被更新过, 则按式(9)、式(10)执行发散操作。转步骤 5。

步骤 5 若上一轮被选择的粒子个数小于 1, 则标记所有粒子为已选择, 并按式(4)计算收缩选择半径 cR , 之后标记所有粒子为外围粒子, 将惯性权重迭代次数 t' 初始化为 1, 由当前迭代开始重新递增。转步骤 6。

步骤 6 按式(4)计算 cR 并判断当前粒子与 $gBest$ 距离是否小于 cR , 若是则对当前粒子按式(1)、式(5)、式(6)、式(2)的顺序更新, 否则只按式(1)与式(2)操作。更新本轮个体最优与全局最优, 转步骤 7。

步骤 7 若 $gBest$ 被外围粒子找到, 重置选择半径为 0。转步骤 8。

步骤 8 若满足精度, 算法结束, 否则迭代次数加 1, 转步骤 2。

3 仿真实验

本文算法受到了标准粒子群算法(PSO)^[1]、自适应变异粒子群算法(AMPSO)^[3]、基于吸引与排斥操作的粒子群算法(ARPSO)^[4]的启发。ARPSO的吸引排斥策略使粒子群在收缩与扩张两种操作中来回切换,以此保持种群多样性,防止早熟,这是本文发散操作的思想来源,但算法因为设定收缩固定最小阈值而导致精度不能进一步提高。AMPSO最致命的缺陷是其扰动因子 $gBest = gBest * (1 + 0.5 * \eta)$ 的扰动幅度 $gBest * \eta$ 随着 $gBest$ 接近原点(零点)而逐渐减小,假若理论最优值在非原点处(如假若位于 $[1, 1, \dots, 1]$ 处),其优化效果将大打折扣。但假若最优点在原点,其扰动策略将会使 $gBest$ 在进化趋于停滞时有更大概率朝原点跳跃,而非向各个方向等概率随机扰动。本文收缩操作的快速逼近极优点思想即是受 AMPSO 算法在计算理论最优值时在原点的特例情况下所表现出的优异性能这一特点启发。用 5 个基准测试函数 f_1 : Sphere、 f_2 : Rosenbrock、 f_3 : Ackley、 f_4 : Griewank、 f_5 : Rosenbrock(理论最小值均为 0)进行测试:

$$f_1(x-1) = \sum_{i=1}^D x_i^2$$

$$f_2(x) = \sum_{i=1}^D [(1-x_i)^2 + 100(x_{i+1}-x_i^2)^2]$$

表 1 4 种算法在 Sphere 函数上的优化结果统计

f_1	D=30, M=100, $I_{max}=500$			D=50, M=500, $I_{max}=500$			D=100, M=1000, $I_{max}=1000$		
	max	min	sr	max	min	sr	max	min	sr
A ₁	-28.7026	-Inf	1	2.0000	-Inf	0.98	2.6990	-30.3294	0.36
A ₂	2	-11.6476	0.96	2.0000	-11.2443	0.94	2.6990	-11.0295	0.30
A ₃	-23.2884	-25.7042	1	-30.5112	-Inf	1	-29.0099	-Inf	1
A ₄	-17.0088	-25.7763	1	-9999	-Inf	1	2	-Inf	0.94

表 2 4 种算法在 Rosenbrock 函数上的优化结果统计

f_2	D=30, M=1000, $I_{max}=1500$			D=50, M=1000, $I_{max}=2000$			D=100, M=2000, $I_{max}=5000$		
	max	min	sr	max	min	sr	max	min	sr
A ₁	3.9094	-0.3497	0	5.7924	0.6006	0	6.6851	3.9126	0
A ₂	3.9099	0.4949	0	5.7924	-2.5726	0.02	6.6851	3.9136	0
A ₃	2.1141	0.2493	0	2.3057	1.2974	0	2.7007	1.6930	0
A ₄	3.0419	-14.4825	0.60	2.2248	-10.8465	0.78	3.0874	-27.5120	0.58

表 3 4 种算法在 Ackley 函数上的优化结果统计

f_3	D=30, M=200, $I_{max}=500$			D=50, M=500, $I_{max}=500$			D=100, M=1500, $I_{max}=1500$		
	max	min	sr	max	min	sr	max	min	sr
A ₁	0.4246	-13.6536	0.46	0.7435	-13.3983	0.28	0.9705	0.4423	0
A ₂	0.3817	-9.3522	0.74	0.7364	-6.8102	0.68	0.9580	0.1381	0
A ₃	0.3649	-14.0973	0.70	0.4626	-13.6536	0.44	0.9438	-13.1882	0.02
A ₄	-12.3215	-13.6536	1	0.2892	-13.4833	0.90	0.6234	-13.2956	0.82

表 4 4 种算法在 Griewank 函数上的优化结果统计

f_4	D=30, M=1000, $I_{max}=500$			D=50, M=1000, $I_{max}=500$			D=100, M=1500, $I_{max}=1000$		
	max	min	sr	max	min	sr	max	min	sr
A ₁	-1.2236	-Inf	0.24	-0.3388	-Inf	0.66	0.0591	-15.3525	0.12
A ₂	-0.9929	-15.9546	0.36	-0.3435	-Inf	0.76	0.0601	-13.0687	0.24
A ₃	-1.0248	-Inf	0.36	-0.1277	-Inf	0.30	0.0169	-Inf	0.36
A ₄	-0.9773	-Inf	0.32	-1.0471	-Inf	0.54	-1.0959	-Inf	0.62

表 5 4 种算法在 Rastrigin 函数上的优化结果统计

f_5	D=30, M=1500, $I_{max}=2500$			D=50, M=2500, $I_{max}=5000$			D=100, M=5000, $I_{max}=10000$		
	max	min	sr	max	min	sr	max	min	sr
A ₁	2.2087	1.2283	0	2.7138	1.8611	0	3.2447	2.7401	0
A ₂	2.004	-10.6553	0.02	2.7159	0.4749	0	3.1646	2.0452	0
A ₃	2.0733	1.0770	0	2.5331	1.4892	0	2.8699	2.2856	0
A ₄	1.7054	-Inf	0.66	2.0043	-Inf	0.32	2.5889	0.5999	0

$$f_3(x-1) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^D \cos 2\pi x_i) + 20 + \exp(1)$$

$$f_4(x-1) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D (\cos \frac{x_i}{\sqrt{i}}) + 1$$

$$f_5(x-1) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10]$$

以上函数经调整后理论全局最优值均位于 $[1, 1, \dots, 1]$ 处,当适应度对数值 $\lg(\text{fitness}(X))$ 小于 -3 时认为算法寻优成功,其中 D 为问题维数。参数设置如下:搜索区域为 $[-9, 11]$, $c_1 = c_2 = 1.4962$, $\omega_{\max} = 0.95$, $\omega_{\min} = 0.4$, ARPSO 中的 D_{high} 在 f_1, f_2 中设置为 3, 其它函数为 2; D_{low} 在 f_2 中为 $10^{\wedge}(-2)$, f_5 中为 $10^{\wedge}(-3)$, 其它函数为 $10^{\wedge}(-6)$; AMP-SO 中的变异概率值 k 统一为 0.3, 方差阈值 σ_d^2 在 f_2 中为 3, 其它函数为 $10^{\wedge}(-3)$; seAPSO 中收缩比 r 统一为 0.5, avg-GP 在 f_1 中为 $10^{\wedge}(-3)$, f_5 中为 0.1, 其它函数为 0.3。M 为种群数量, $I_{\max}$ 为最大迭代次数。各算法在各优化问题上独立运行 50 次。分别将 SPSO、ARPSO、AMPSO、seAPSO 记为 A₁、A₂、A₃、A₄, max 代表 50 次运算中获得的结果最大值, min 代表最小值, sr 代表寻优成功率。仿真统计数据见表 1—表 5, 计算 100 维函数时的进化曲线见图 1—图 4。

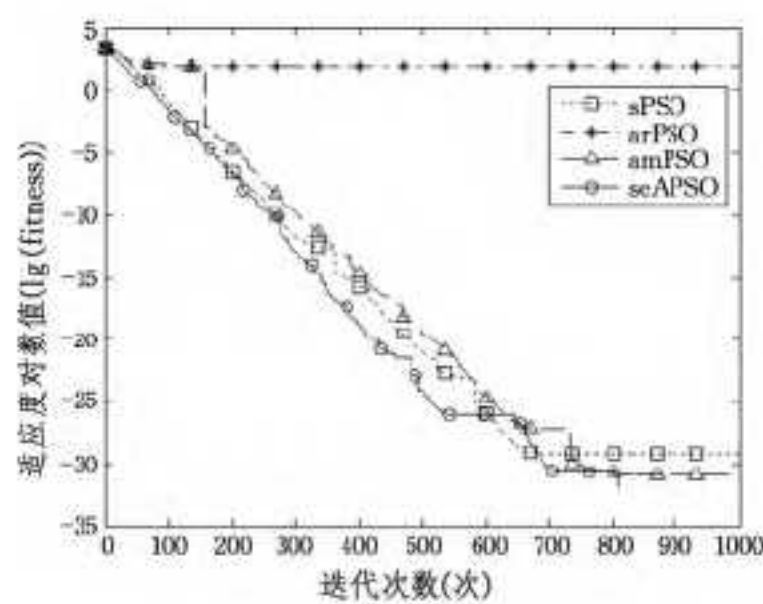


图1 100维 Sphere 函数寻优曲线

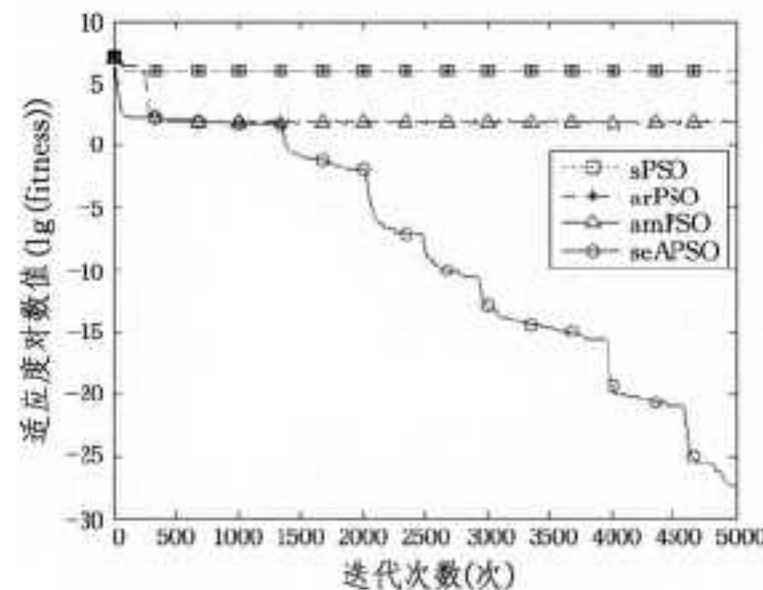


图2 100维 Rosenbrock 函数寻优曲线

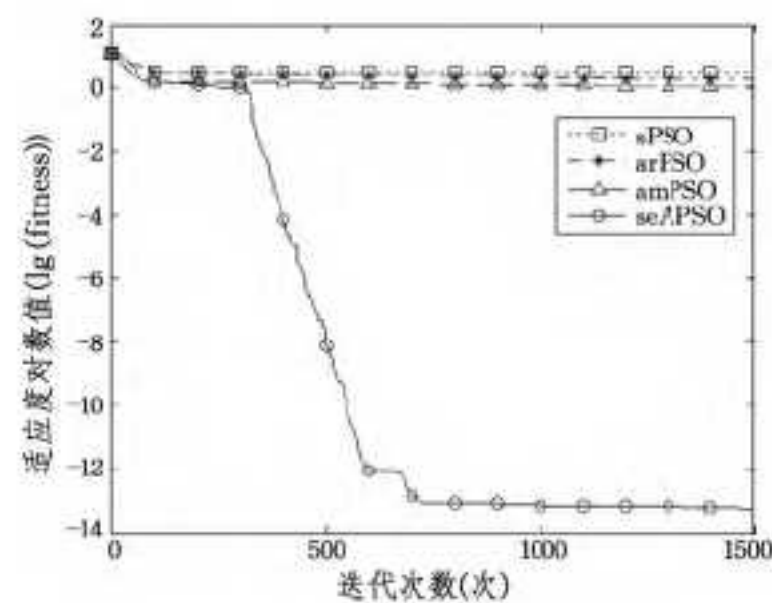


图3 100维 Ackley 函数寻优曲线

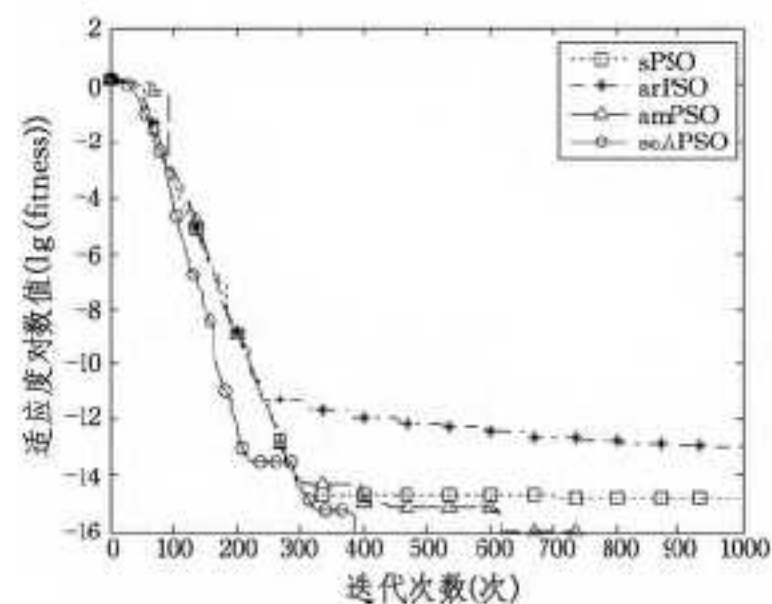


图4 100维 Griewank 函数寻优曲线

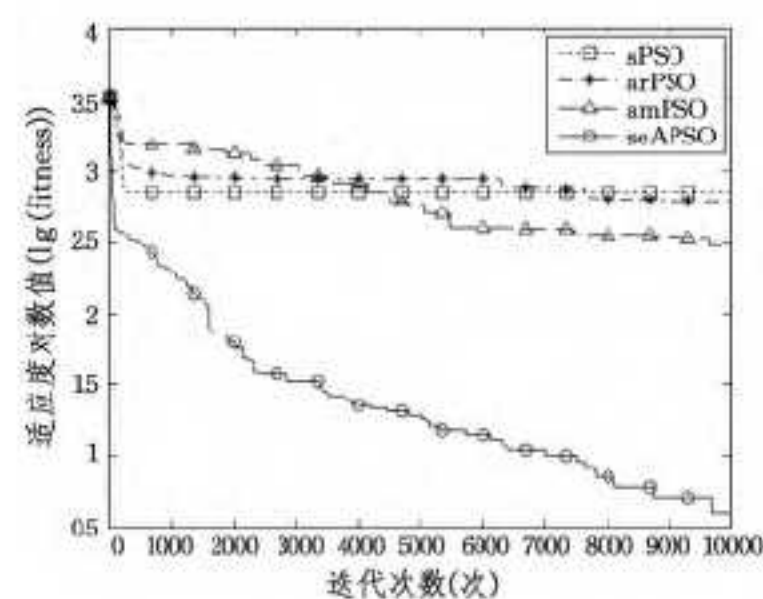


图5 100维 Rastrigin 函数寻优曲线

f_1 为简单的单峰函数。在这个函数上 seAPSO 在保证

收敛精度的前提下,提高了寻优成功率。函数 f_2 的最优值隐藏在一条平滑而狭长的抛物线形状的山谷底部,为优化算法提供的信息很少,普通算法在在计算高维数中容易丢失种群进化方向信息。而 seAPSO 可以通过多次发散操作逐次将粒子群重新分布至山谷附近,并通过收缩操作将发散后的新一代种群从各个方向拉拢至山谷底部。在收缩到一定程度、粒子密度足够大时,总可以发现新的最优解,从而获得新的进化方向信息,能沿山谷底部较小的适应度变化方向逐步逼近全局最优值,表现出很强的跳出局部最优解的能力,优化效果出众。函数 f_3 、 f_4 、 f_5 均是具有大量局部极小值的多峰函数,普通算法极易陷入局部最优值而停止进化。从进化曲线图可以看出,seAPSO 因具有从各个方向重新寻优的特点,能在每次判断出种群陷入局部最优解时及时跳出该解并从全局视野上重新寻优,使每次进行发散操作后寻优曲线都有一个呈阶梯状的下降变化,保证了收敛成功率,并且在寻优精度上也优于其它算法。相比而言,seAPSO 在处理高维复杂问题上更具优势。

结束语 本文提出一种兼顾收敛精度和全局收敛性的粒子群改进算法。实验表明,本文提出的算法在保证收敛精度的前提下提高了算法收敛成功率,并在处理高维问题时相比于其它算法表现出较明显的优势,具有较强的跳出局部最优解继续进化的能力。但该算法性能的提升依赖于粒子数与迭代次数的增加,开销较大。未来研究中,将会把解决解空间膨胀与维度之间的耦合性问题及保证精度与成功率的前提下减少迭代次数与粒子数作为研究重点。

参考文献

- [1] James K, Russell E. Particle swarm optimization [C]// Proceedings of 1995 IEEE International Conference on Neural Networks. 1995:1942-1948
- [2] Shi Y, Eberhart R. A modified particle swarm optimizer [C] // The 1998 IEEE International Conference on Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. IEEE, 1998: 69-73
- [3] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报, 2004, 32(3): 416-420
- [4] Li C, Yang S, Korejo I. An adaptive mutation operator for particle swarm optimization [C] // The 2008 UK Workshop on Computational Intelligence. 2008: 165-170
- [5] Pant M, Thangaraj R, Abraham A. Particle swarm optimization using adaptive mutation [C] // 19th International Workshop on Database and Expert Systems Application, 2008 (DEXA'08). IEEE, 2008: 519-523
- [6] Vesterström J S, Riget J. A Diversity-Guided Particle Swarm Optimizer-the ARPSO[R]. EVALife Technical Report, 2002
- [7] Pant M, Radha T, Singh V P. A simple diversity guided particle swarm optimization [C] // IEEE Congress on Evolutionary Computation, 2007 (CEC 2007). IEEE, 2007: 3294-3299
- [8] 赵志刚, 张振文, 石辉磊, 等. 带扰动因子的自适应粒子群优化算法[J]. 计算机科学, 2013, 40(12): 68-69, 103