

基于 S 型函数的自适应粒子群优化算法

黄 洋¹ 鲁海燕^{1,2} 许凯波¹ 胡士娟¹

(江南大学理学院 江苏 无锡 214122)¹ (无锡市生物计算工程技术研究中心 江苏 无锡 214122)²

摘 要 针对粒子群算法求解精度低和后期收敛速度慢等问题,提出了一种基于 S 型函数的自适应粒子群优化算法 SAPSO (S-shaped function based Adaptive Particle Swarm Optimization)。该算法利用倒 S 型函数的特点,实现了对惯性权重的非线性调整,从而更好地平衡算法的全局搜索能力和局部搜索能力;同时,在算法的位置更新公式中引入 S 型函数,并利用个体粒子自身的适应度值与群体平均适应度值的比值自适应地调整搜索步长,从而提高算法的搜索效率。在若干经典测试函数上的仿真实验结果表明,与已有的几种改进粒子群算法相比,SAPSO 在收敛速度和求解精度方面均有较大优势。

关键词 S 型函数,惯性权重,位置更新,粒子群优化算法

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.01.038

S-shaped Function Based Adaptive Particle Swarm Optimization Algorithm

HUANG Yang¹ LU Hai-yan^{1,2} XU Kai-bo¹ HU Shi-juan¹

(School of Science, Jiangnan University, Wuxi, Jiangsu 214122, China)¹

(Wuxi Engineering Technology Research Center for Biological Computing, Wuxi, Jiangsu 214122, China)²

Abstract Aiming at the problems of low solution precision and slow convergence speed in the later stage of particle swarm optimization algorithm, this paper presented an S-shaped function based adaptive particle swarm optimization algorithm (SAPSO). This algorithm takes advantage of the characteristics of upside-down S-shaped function to adjust the inertia weight nonlinearly, better balancing the global search ability and local search ability. In addition, an S-shape function is introduced into the position updating equation, and the ratio of the individual particle's fitness value to the swarm's average fitness value is used to adaptively adjust the step size in the search, thus enhancing the efficiency of the algorithm. Simulation results on a set of typical test functions show that SAPSO is superior to several existing improved PSO algorithms significantly in terms of the convergence rate and solution accuracy.

Keywords S-shaped function, Inertia weight, Position updating, Particle swarm optimization algorithm

1 引言

近年来,随着现实生活中优化问题复杂程度的增加,粒子群优化算法^[1]、遗传算法^[2]、蚁群算法^[3]等各种智能算法相继被提出来解决这些问题。粒子群优化(Particle Swarm Optimization, PSO)算法作为一种基于群体的智能优化算法^[1],由于其实现简单、参数少,已经被广泛应用于参数优化^[4-5]、电磁优化^[6-7]等领域。但是 PSO 算法存在容易早熟收敛以及易陷入局部最优等不足,尤其是对于多模态函数来说,其容易陷入局部最优。随着迭代次数的增加,算法可能无法逃出局部最优而出现早熟收敛,算法的求解精度也难以提高。而对于大多数改进粒子群算法来说,在达到指定精度之后,如何提高算法的收敛速度也是粒子群算法主要研究的问题之一。因此,许多学者从算法参数、速度和位置公式以及与其他算法相结合等不同的角度来提高算法的性能。Shi 等^[8]提出了惯性权

重线性递减的粒子群算法,算法的全局和局部搜索能力得到提高。文献^[9]提出了一种变加速因子的粒子群算法,该算法更好地平衡了算法的全局与局部搜索能力。文献^[10]通过在算法中引入压缩因子,来控制粒子的运动轨迹,保证算法的收敛。朱童等^[11]提出了一种位置加权的改进粒子群算法,该算法有效地提高了算法的搜索效率。文献^[12]将高斯扰动策略加入到粒子群算法中,增强了算法逃离局部最优的能力,提高了算法的收敛速度。文献^[13]将高斯扰动策略、自适应调节余弦函数递减策略以及异步变化学习因子调整方法与粒子群算法相结合,有效地平衡了算法的全局和局部寻优性能。Zhan 等^[14]通过引入多种学习策略,提出了自适应粒子群优化算法,该算法的性能有很大的提高。文献^[15]把模拟退火算法引入粒子群算法中,改进后的算法增强了摆脱局部最优解的能力。文献^[16]将差分演化算法与粒子群算法相融合,加快了算法的收敛速度。上述不同的改进策略从不同方面对

到稿日期:2017-11-24 返修日期:2018-03-02 本文受国家自然科学基金项目(61772013,61402201),中央高校基本科研业务费专项资金项目(114205020513526)资助。

黄 洋(1991-),男,硕士,CCF 会员,主要研究方向为最优化与控制;鲁海燕(1970-),女,博士,副教授,主要研究方向为组合最优化、智能算法,E-mail:luhaiyan@jiangnan.edu.cn(通信作者);许凯波(1992-),男,硕士,主要研究方向为最优化与控制;胡士娟(1994-),女,硕士,主要研究方向为最优化与控制。

粒子群算法的性能进行了提升,但是有些算法的改进策略比较复杂,并且部分改进算法的收敛速度较慢。

为了使算法改进简单、加快收敛速度以及提高求解精度,本文提出了一种基于S型函数的自适应粒子群优化算法。通过将不同的S型函数引入惯性权重和位置更新公式,来增强粒子逃离局部最优解的能力,从而提高算法的收敛速度和求解精度。与其他改进算法相比,本文改进算法的性能有所提高。

2 粒子群优化算法

在PSO算法中,群体中的每个粒子都是优化问题的一个可行解,每个粒子的适应度值是由目标函数来确定的。粒子在搜索空间中以一定的速度飞行并根据自身和群体的飞行经验对粒子的速度和位置进行动态调整,个体之间通过协作与竞争实现对问题最优解的搜索^[17]。

假设PSO算法在一个 N 维的搜索空间中,种群有 m 个粒子,那么第 i 个粒子在第 t 次迭代时的位置表示为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})$,粒子的速度记为 $V_i = (v_{i1}, v_{i2}, \dots, v_{iN})$,其中 $i=1, 2, \dots, m$ 。粒子自身在迭代过程中搜索到的最优位置记为 p_{best} ,整个粒子群搜索到的最优位置记为 g_{best} 。粒子更新公式如下:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (p_{best} - X_i^t) + c_2 r_2 (g_{best} - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

其中, ω 为惯性权重; c_1 和 c_2 为学习因子,通常取值为2; $\omega_{max}=0.9$ 和 $\omega_{min}=0.4$ 为分布在 $[0,1]$ 内的随机数; t 为粒子当前的迭代次数。

3 S型函数的特征

S型曲线多存在于逻辑回归(Logistic Regression)模型中,在种群增长、植物生长以及病毒流行等众多领域有着广泛的应用。S型函数的表达式为:

$$y = \frac{k}{1 + e^{-rx}}$$

其中, a 和 r 为常数。令 $x = \frac{1}{t}$,则得到倒S型函数的表达式为:

$$y = \frac{k}{1 + e^{-\frac{r}{t}}}$$

S型函数和倒S型函数的图像如图1和图2所示。由S型函数图像可知,随着 x 的增大, y 随之增大, y 在前期的增长速度较快,在后期的增长速度较慢。

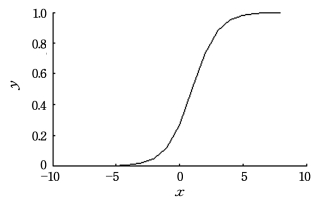


图1 S型函数

Fig. 1 S-shaped function

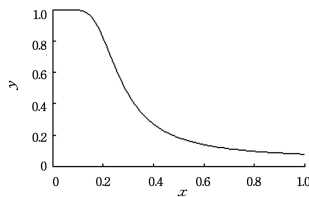


图2 倒S型函数

Fig. 2 Upside-down S-shaped function

4 改进的粒子群优化算法

4.1 惯性权重

在粒子群算法中,Shi等^[8]将惯性权重引入到速度更新

公式当中,并提出了惯性权重线性递减策略的粒子群算法,其改进的惯性权重公式为:

$$\omega = \omega_{start} - (\omega_{start} - \omega_{end}) \frac{t}{t_{max}} \quad (3)$$

其中, t 为粒子的当前迭代次数, t_{max} 为粒子的最大迭代次数。

文献[18]对4种惯性权值函数(线性递减函数、开口向下的抛物线函数、开口向上的抛物线函数和递减的指数曲线函数)策略的改进粒子群算法进行了实验对比分析。结果表明,惯性权值为开口向上的抛物线函数和递减的指数曲线函数时,算法的寻优性能均能得到不同程度的提高。结合上述惯性权值函数的改进策略,近年来,高斯函数惯性权重递减策略、惯性权重对数递减策略、柯西密度函数惯性权重递减策略等^[19-20]也相继被提出并用以改善粒子群算法的性能。上述惯性权重改进策略虽然都能在一定程度上提高粒子群算法的性能,但是由于在前期搜索阶段惯性权值递减速度过快,相应算法容易使得相应种群中的粒子过快丧失多样性而导致早熟收敛。

为了改善上述不足,本文利用倒S型函数在整个区间内递减而在递减初期和后期变化缓慢的特点对惯性权重进行改进。这种改进策略可使得在迭代初期惯性权重能较长时间地取得较大的值,从而加强算法的全局搜索能力;在迭代后期能较长时间地取得较小的值,从而加强算法的局部搜索能力^[21]。本文改进的惯性权重公式为:

$$\omega = \omega_{start} - (\omega_{start} - \omega_{end}) \frac{1}{1 + e^{a-bt}} \quad (4)$$

其中, $a=3.4$, $b=0.07$, t 为粒子的当前迭代次数。惯性权重曲线如图3所示。

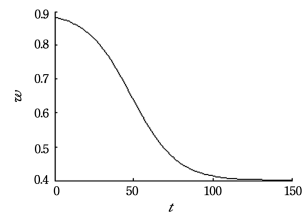


图3 惯性权重递减曲线

Fig. 3 Decline curve of inertia weight

由图3可知,这种改进策略使得惯性权重在整个迭代过程中呈一个倒S型的递减趋势。这样的改进策略可以更好地平衡算法的全局搜索能力和局部搜索能力。

4.2 位置公式的更新

为了有效避免粒子群算法在搜索后期陷入局部最优,文献[22]结合位置更新式(2),利用数学中的外推法给出两个新的粒子位置更新公式,对粒子位置更新加以引导,提出了一种引导型粒子群算法,最后得到粒子的位置更新公式为:

$$X_i^{t+1} = X_i^t + [\text{rand} + k]V_i^{t+1} + 10^{-6} \text{rand} \quad (5)$$

文献[11]结合文献[22]中对位置公式的改进方式,提出了一种位置加权粒子群算法,该算法主要是在式(2)的基础上增加一个式子:

$$\tilde{X}_i^{t+1} = \alpha X_i^t + (1-\alpha)V_i^{t+1} \quad (6)$$

将式(2)和式(6)联立,得到粒子新位置的更新公式为:

$$X_i^{t+1} = X_i^t + (1-\alpha)V_i^{t+1} \quad (7)$$

仿真实验结果表明:上述两种对位置更新公式的改进方法能够有效避免算法在搜索过程中由于步长过长而错过全局

最优值,减少了粒子在搜索过程中的盲目性,提高了算法的搜索效率。

上述两种对位置更新公式的改进方法的本质是:在原来的位置更新公式中通过设置随机步长因子或固定步长因子来对粒子的搜索步长进行调整。但在粒子飞行过程中,如果每一个粒子都采用固定搜索步长因子对粒子的搜索步长进行调整,那么当固定搜索步长因子的取值较大时,可能导致粒子远离最优解,算法收敛速度变慢;当固定搜索步长因子的取值较小时,可能导致算法在搜索过程中过早地陷入局部最优解,最终求得的解的精度较低。当每一个粒子都采用随机搜索步长因子对粒子的搜索步长进行调整时,可能使得算法在前期搜索过程中取得较小的搜索步长因子,从而导致算法过早地陷入局部最优;也可能使得算法在后期搜索过程中取得较大的搜索步长因子,从而导致算法远离最优解。上述两种对搜索步长的改进方式都缺乏自适应性,不能在算法的不同搜索阶段自适应地调整搜索步长。

在粒子群算法中,粒子自身的适应度值与群体平均适应度值的比值可用于描述个体粒子在群体中的相对搜索状态。对于极小化问题,若上述两个适应度值的比值较大,则表示当前个体距离群体最优位置较远,因此可适当增大粒子的搜索步长以增强算法的全局搜索能力;若两个适应度值的比值较小,则说明当前粒子距离群体最优位置较近,从而可适当减小粒子的搜索步长进行精度搜索。基于上述分析,本文根据 S 型函数前期快速增加、后期缓慢增加的特点,同时将搜索过程中两个适应度值比值的变化的变化作为自变量,引入 Sigmoid 函数(一种 S 型函数)来构造动态调整因子,其表达式如式(8)所示。采用式(9)对粒子群算法的位置更新公式进行改进,使粒子的搜索步长随着适应度值比值的变化的变化而自适应改变。

$$\varphi = \frac{1}{1 + \exp(-f(x_i)/f(x)_{\text{avg}})} \quad (8)$$

$$X_i^{t+1} = X_i^t + \alpha \varphi V_i^{t+1} \quad (9)$$

其中, $f(x_i)$ 为个体粒子的适应度值, $f(x)_{\text{avg}}$ 为当前整个群体的平均适应度值。

为了进一步提高算法的搜索效率,本文结合文献[10]在速度更新公式中加入压缩因子的方法,对式(9)做进一步改进,可得到新的位置更新公式:

$$X_i^{t+1} = \alpha \varphi (X_i^t + V_i^{t+1}) \quad (10)$$

综合上述分析,本文提出的基于 S 型函数的自适应粒子群优化算法(S-shaped function based Adaptive Particle Swarm Optimization, SAPSO)的计算步骤如下:

步骤 1 设定粒子群算法中的相关参数;

步骤 2 初始化种群中粒子的位置和速度;

步骤 3 计算粒子群算法中每个粒子的适应值,同时计算出粒子的最优位置以及整个群体的最优位置;

步骤 4 按照式(1)、式(4)和式(10)对粒子的速度和位置进行更新;

步骤 5 若达到最大迭代次数,则算法停止,输出最优值;否则,返回步骤 3。

5 数值实验和结果分析

5.1 测试函数以及参数设置

为了测试本文提出的 SAPSO 算法的有效性,本文选取

了 9 个基本测试函数^[14,23]来对所提出的改进算法进行验证。函数参数设置如表 1 所列,其中 $f_1 - f_5$ 为单模态函数, $f_6 - f_9$ 为多模态函数。

表 1 9 个基本测试函数

Table 1 Nine benchmark test functions

函数序号	函数名	范围	最优值
f_1	Sphere	$[-100, 100]$	0
f_2	Schwefel P2.21	$[-100, 100]$	0
f_3	Schwefel P2.22	$[-10, 10]$	0
f_4	Rosenbrock	$[-10, 10]$	0
f_5	Quadric Noise	$[-1.28, 1.28]$	0
f_6	Schaffers	$[-100, 100]$	0
f_7	Rastrigin	$[-5.12, 5.12]$	0
f_8	Griewank	$[-600, 600]$	0
f_9	Ackley	$[-32, 32]$	0

实验中不同的 PSO 算法设置了相同的群体规模 $N=40$, 搜索维数 $D=30$, 最大迭代次数 $t_{\text{max}}=500$, 学习因子 $c_1=c_2=2$, 其他参数设置与原文献保持一致, 在 SAPSO 算法中 $\omega_{\text{max}}=0.9$, $\omega_{\text{min}}=0.4$ 。

5.2 参数 α 的选取实验

在本文的改进粒子群算法的位置更新公式(见式(10))中,当参数 α 取不同值时,粒子群优化算法会得到不同的寻优结果。为了确定参数 α 的最优取值,本文选取表 1 中 9 个标准测试函数进行测试,算法运行 50 次,取最优值的平均值作为比较结果,具体结果如表 2 所列。

表 2 参数 α 取不同值时 SAPSO 算法的平均最优值

Table 2 Average optimal values of SAPSO algorithm under different values of parameter α

α	0.1	0.2	0.3	0.4
f_1	0/452	0/370	0/407	0/440
f_2	1.97E-241	5.63E-301	2.02E-286	4.51E-260
f_3	6.00E-181	1.45E-180	3.27E-196	1.09E-188
f_4	8.9454	8.9409	8.9536	8.9186
f_5	5.55E-05	7.55E-05	5.36E-05	6.35E-05
f_6	0/452	0/454	0/423	0/436
f_7	0/60	0/56	0/60	0/58
f_8	0/56	0/58	0/60	0/59
f_9	8.88E-16	8.88E-16	8.88E-16	8.88E-16

由表 2 的实验结果可知,当参数 α 取不同值时,本文改进算法在函数 $f_1, f_6 - f_9$ 上的最优值的平均值结果相同,且在 $f_1, f_6 - f_8$ 这 4 个函数上都能够找到函数的理论最优值,说明算法的寻优差异不明显。但在函数 f_2 和 f_4 上,当参数 $\alpha=0.2$ 时,本文算法的寻优性能较优。为了进一步确定参数 α 的最优值,在表 2 中给出了本文算法在函数 $f_1, f_6 - f_8$ 上达到最优值时的平均迭代次数(最优值平均值/平均迭代次数)。由上述实验结果可知,当参数 $\alpha=0.2$ 时,本文改进算法的整体性能最优。

5.3 与相关改进算法的比较

为了验证本文算法的有效性,利用 5.1 节中的 9 个测试函数对惯性权重线性递减的粒子群算法 LWPSO^[8](Linear Decreasing Inertia Weight Particle Swarm Optimization Algorithm)、带压缩因子的粒子群算法 YPSO^[10](Particle Swarm Optimization Algorithm with Compression Factor)、文献[22]中的引导型粒子群算法 IEPSO(Particle Swarm Optimization Algorithm with Induction-enhanced)、随机惯性权重粒子群算

法 RandPSO^[24] (Random Weighting Particle Swarm Optimization Algorithm) 与本文改进算法 (SAPSO) 分别进行测试, 各算法分别运行 50 次, 取最优值的平均值 (*mean*) 和标准差

(*std*) 进行比较。最优值的平均值和标准差分别反映了算法的求解精度和稳定性, 具体实验结果如表 3 所列, 其中的最好结果用黑体表示。

表 3 5 种算法的性能测试结果

Table 3 Performance test results of five algorithms

函数		LWPSO	YSPSO	RandPSO	IEPSO	SAPSO
f_1	<i>mean</i>	0.6179	2.1925E+01	0.2241	1.5791E+01	0
	<i>std</i>	0.2347	1.8041E+01	0.1083	8.6774E+00	0
f_2	<i>mean</i>	3.8389E-29	2.4958E-26	5.5068E-25	1.3691E-10	8.7710E-301
	<i>std</i>	1.6565E-28	5.7501E-26	3.2818E-24	1.3542E-10	0
f_3	<i>mean</i>	3.812	11.5068	4.3946	3.7073	1.1156E-180
	<i>std</i>	1.5587	2.4167	1.7982	1.1504	0
f_4	<i>mean</i>	8.8522E+01	6.4030E+02	9.3154E+01	8.3264E+01	8.9494
	<i>std</i>	5.5335E+01	4.1075E+02	4.5555E+01	5.0079E+01	0.0330
f_5	<i>mean</i>	0.1698	0.7387	0.2549	0.0884	4.8460E-005
	<i>std</i>	0.3877	0.7174	0.1191	0.0487	5.0276E-005
f_6	<i>mean</i>	9.1822E+01	9.7070E+01	9.2791E+01	9.6414E+01	0
	<i>std</i>	1.6433E+01	1.1262E+01	1.0290E+01	9.3614E+00	0
f_7	<i>mean</i>	4.8808E+01	8.4733E+01	1.5749E+01	4.2885E+01	0
	<i>std</i>	1.2322E+01	1.5749E+01	1.2522E+01	1.1958E+01	0
f_8	<i>mean</i>	3.4069E+01	3.0512E+01	2.7488E+01	1.7258E+02	0
	<i>std</i>	4.7708	4.2794	4.7931	1.3729E+01	0
f_9	<i>mean</i>	4.2162	5.9271	5.3272	4.5519	8.8818E-16
	<i>std</i>	0.9958	0.9992	1.1515	0.8005	0

对表 3 中的数据进行对比可以看出, 与其他 4 种改进的粒子群算法相比, SAPSO 算法在单模态函数 $f_1 - f_3$ 以及多模态函数 $f_6 - f_9$ 上的标准差都为 0, 这说明该算法的稳定性比其他 4 种算法更高。同时, SAPSO 算法在单模态函数 f_1 和多模态函数 $f_6 - f_8$ 这 4 个测试函数上的最优值的平均值都为 0, 说明算法的求解精度更高。但是 SAPSO 算法在较难

寻优的典型病态单模态函数 f_4 上的最优值的平均值与实际最优值相比, 求解精度较差, 说明在此函数的寻优上, 该算法陷入了局部最优。总的来说, SAPSO 算法在单模态和多模态函数上的寻优性能均优于其他几种算法。

图 4 给出了 5 种算法在 9 个测试函数上的适应度平均值进化曲线。

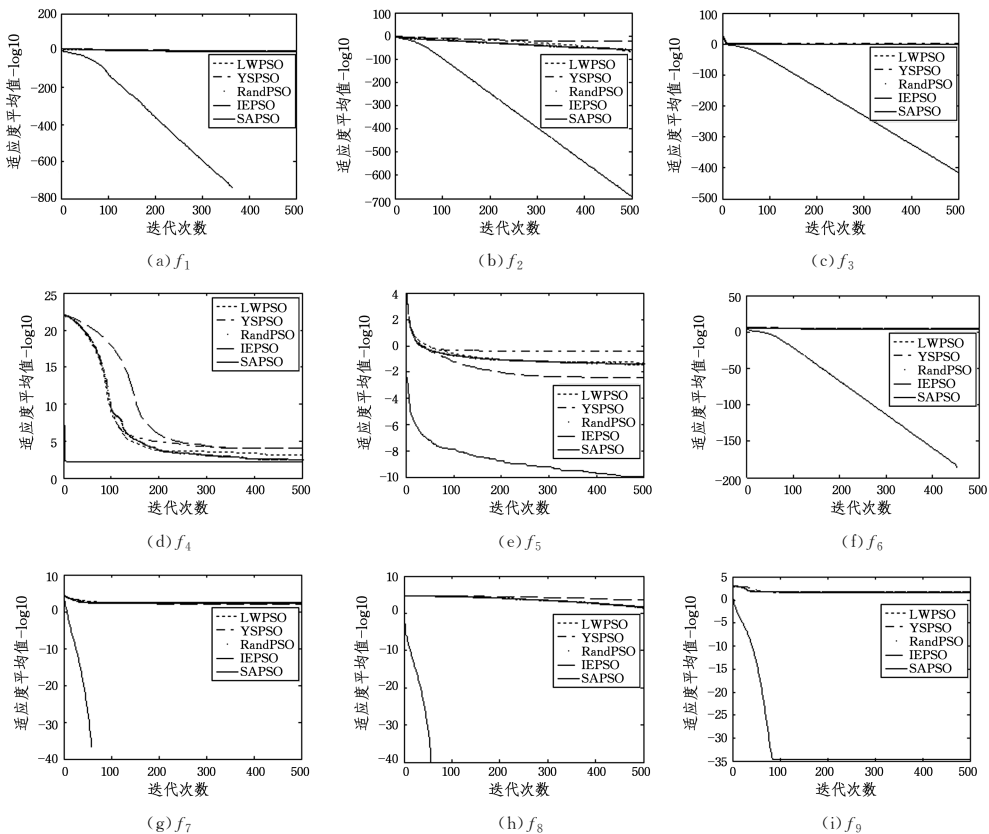


图 4 5 种算法在 9 个测试函数上的适应度平均值曲线

Fig. 4 Average fitness value curve of five algorithms on nine benchmark test functions

由图 4 可以很直观地看出,随着迭代次数的增加,SAPSO 算法的进化曲线下降明显,且在处理单模态函数 f_1 和多模态函数 f_6-f_8 问题上,都能够快速收敛到最优解。此外,由图 4(a)~图 4(c)和图 4(e)可知,SAPSO 算法在单峰函数上不仅收敛速度快,而且求解精度高。特别是对于图 4(a)中的复杂非线性对称单模态函数 f_1 来说,其他 4 种改进的粒子群算法并不能找到其最优解,而 SAPSO 算法不仅收敛速度快,而且能找到其最优解。从图 4(f)~图 4(i)可看出,本文改进的算法不仅在处理单模态函数问题上有很好的寻优性能,在处理多模态函数问题上,其收敛速度和求解精度也更加明

表 4 在不同改进策略下 SAPSO 算法的测试结果比较

Table 4 Comparison of test results of SAPSO under different improved strategies

测试函数	仅有惯性改进策略		仅有位置更新策略		组合策略	
	mean	std	mean	std	mean	std
f_1	4.4207E+02	2.920E+02	0.0062	0.0030	0	0
f_2	7.8214	2.307	4.0387E-046	2.0304E-046	8.7710E-301	0
f_3	0.2771	0.2008	0.7394	0.1167	1.1156E-180	0
f_6	77.0938	17.1885	8.0917	0.8604	0	0
f_7	35.7593	6.0125	1.3904	0.8077	0	0
f_8	5.4535	1.0292	2.2205E-004	1.4762E-004	8.8818E-16	0

由表 4 的实验结果可知,仅有惯性权重改进策略的 SAPSO 算法,得到的最优值平均值和标准差都较差。但其在函数 f_2, f_3 和 f_8 上的求解精度相对较好,这说明改进惯性权重策略虽然能够提高算法的全局搜索能力,但是在寻优的过程中缺乏引导,导致算法容易陷入局部最优;而仅有位置更新策略的 SAPSO 算法的寻优结果较好,说明对位置公式的改进方法减少了搜索过程中的盲目性,因此能提高算法的搜索效率,同时也能提高算法的求解精度,但是并未寻找到最优解;而两个策略结合的 SAPSO 算法不管在 3 个单模态函数上还是在 3 个多模态函数上都能取得较好的求解精度,同时 SAPSO 算法在函数 f_1, f_6 和 f_7 上的最优值的平均值和标准差都为 0。上述实验结果表明,算法在改进惯性权重策略和位置更新策略的共同作用下,能够有效地避免算法陷入局部最优,同时又加快了收敛速度,提高了求解精度。

5.5 算法时间复杂度分析

标准粒子群算法的时间复杂度为 $O(nmD)^{[25]}$,其中 n 为粒子个数, m 为算法的迭代次数, D 为优化问题的维数。本文改进的粒子群算法利用式(1)、式(4)和式(10)对算法的速度和位置进行更新,与标准粒子群算法中的式(1)和式(2)相比,其实质相当于在标准粒子群算法的每次迭代过程中,在速度和位置更新公式中线性地乘上一个常数。因此,改进的策略并未增加算法的时间复杂度,且本文改进算法 SAPSO 的时间复杂度仍为 $O(nmD)$ 。由 5.2 节中对参数 α 最优值选取的实验结果可知,在求得相同精度解的情况下,当参数 $\alpha=0.2$ 时,所需要的迭代次数更少;由 5.3 节中的 5 种算法在 9 个测试函数上的进化曲线可以看出,相较于其他 4 种算法,SAPSO 算法的收敛速度更快。综上可知,在种群规模和问题求解空间相同的情况下,SAPSO 算法的时间复杂度并未增加且算法的收敛性能均优于其他 4 种算法。

5.6 与其他相关改进粒子群算法的比较实验分析

为了使算法的性能进一步得到验证,将本文改进的算法

显,尤其是对于函数 f_6-f_8 来说,SAPSO 算法都能够快速收敛到最优解。由此可得:SAPSO 算法比其他 4 种算法具有更好的全局搜索能力以及更快的收敛速度,尤其是在多模态函数问题上,本文算法表现更优。

5.4 改进惯性权重策略和位置更新策略的对比实验

为了进一步明确改进惯性权重策略和位置更新策略在 SAPSO 算法中的作用,在只有惯性权重改进策略、位置更新策略和两种组合策略都有的条件下,选取 3 个单模态函数 f_1-f_3 和 3 个多模态函数 f_6-f_8 ,对 SAPSO 算法的性能进行测试,实验结果如表 4 所列。

(SAPSO)与文献[26]中的基于试探的变步长自适应粒子群算法在其所测试的 3 个函数(Rosenbrock, Rastrigin, Griewank)上的最优值和方差进行比较,各个算法独立运行 20 次,具体实验结果如表 5 所列,其中的最好结果用黑体表示。另外,SAPSO 算法中的相关参数设置与文献[26]中的参数设置保持一致,文献[26]中算法的实验数据取自文献本身。

表 5 文献[26]中的算法和 SAPSO 算法的实验结果

Table 5 Experimental results of reference[26]'s algorithm and SAPSO algorithm

函数	文献[26]中的算法		SAPSO 算法	
	mean	std	mean	std
Rosenbrock	14.6624	12.3836	8.9172	0.0011
Rastrigin	6.1126	4.1532	0	0
Griewank	0.0561	0.0231	0	0

文献[26]中的改进算法和 SAPSO 算法都是对惯性权重和粒子的搜索步长进行改进,并且都利用了 S 型函数的相关特点,但由表 5 中的实验数据结果对比可知,不管是在单峰函数还是在多峰函数问题上,SAPSO 算法的求解精度和稳定性均优于文献[26]的算法,并且 SAPSO 算法在 2 个多峰函数上都能够寻找到理论最优值。因此,SAPSO 算法的总体寻优效果更优。

结束语 本文针对粒子群优化算法存在早熟收敛、求解精度低等问题,提出了一种基于 S 型函数的自适应粒子群优化算法。该算法利用倒 S 型函数和 S 型函数的特点对惯性权重和位置更新公式进行改进,既提高了算法的全局搜索能力和局部搜索能力,又提高了算法的搜索效率。仿真实验结果表明,与已有文献中的改进算法相比,本文改进算法具有收敛速度快、求解精度高等特点,并且改进算法易编程实现,在实际问题中具有一定的应用潜力。

参考文献

[1] KENNEDY J, EBERHART R C. Particle swarm optimization

- [C]//Proceedings of IEEE International Conference on Neural Networks, 1995:1942-1948.
- [2] HOLLAND J H. Adaptation in Natural and Artificial Systems [D]. Ann Arbor:University of Michigan press,1975.
- [3] COLORNI A,DORIGO M,MANIEZZO V,et al. Distributed optimization by ant colonies[C]//Proceedings of European Conference on Artificial Life, Paris,1991:134-142.
- [4] LIU Z H,WEI H W,ZHONG Q C,et al. Parameter Estimation for VSI-Fed PMSM based on a Dynamic PSO with Learning Strategies [J]. IEEE Transactions on Power Electronics,2017,32(4):3154-3165.
- [5] LIU Z H,LI X H,ZHANG H Q,et al. An Enhanced Approach for Parameter Estimation Using Immune Dynamic Learning PSO Based on Multi-core Architecture [J]. IEEE Systems, Man, and Cybernetics Magazine,2016,2(1):26-33.
- [6] LIU Z H,WEI H W,ZHONG Q C,et al. GPU Implementation of DPSO-RE Algorithm for Parameters Identification of Surface PMSM Considering VSI Nonlinearity [J]. IEEE Journal of Emerging and Selected Topics in Power Electronics,2017,5(3):1334-1345.
- [7] LIU Z H,ZHANG J,ZHOU S W,et al. Coevolutionary Particle Swarm Optimization Using AIS and Its Application in Multi-parameter estimation of PMSM[J]. IEEE Transactions on Cybernetics,2013,43(6):1921-1935.
- [8] SHI Y,EBERHART R C. A modified particle swarm optimizer [C]//Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC'98). NJ: IEEE Press,1998:69-73.
- [9] RATNAWEERAA,HALGAMUGE S K,WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Transactions on Evolutionary Computation,2004,8(3):240-255.
- [10] CLERC M,KENNEDY J. The particle swarm-explosion, stability and convergence in a multidimensional complex space[J]. IEEE Transactions on Evolutionary Computation,2002,6(1):58-73.
- [11] ZHU T,LI X F,LU M W. Improved particle swarm optimization with position weighted[J]. Computer Engineering and Applications,2011,47(5):4-6. (in Chinese)
朱童,李小凡,鲁明文. 位置加权的改进粒子群算法[J]. 计算机工程与应用,2011,47(5):4-6.
- [12] MELO H,WATADA J. Gaussian-PSO with fuzzy reasoning based on structural learning for training a Neural Network [J]. Neurocomputing,2016,172:405-412.
- [13] AI B,DONG M G. Improved particle swarm optimization algorithm based on Gaussian disturbance and natural selection [J]. Journal of Computer Applications,2016,36(36):687-691. (in Chinese)
艾兵,董明刚. 基于高斯扰动和自然选择的改进粒子群算法[J]. 计算机应用,2016,36(36):687-691.
- [14] ZHAN Z H,ZHANG J,LI Y,et al. Adaptive particleswarm optimization[J]. IEEE Transactions on Systems Man & Cybernetics,2009,39(6):1362-1381.
- [15] JAVIDRAD F,NAZARI M. A new hybrid particle swarm and simulated annealing stochastic optimization method [J]. Applied Soft Computing,2017,60:634-654.
- [16] GOU J,LEI Y X,GUO W P,et al. A novel improved particle swarm optimization algorithm based on individual difference evolution [J]. Applied Soft Computing,2017,57:468-481.
- [17] CHENG T,CHEN M,YANG Z,et al. A novel hybrid teaching learning based multi-objectiveparticle swarm optimization [J]. Neurocomputing,2016,222(C):11-25.
- [18] CHEN G M,JIA J Y,HAN Q. Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm[J]. Journal of Xi'an Jiaotong University,2006,40(1):53-56. (in Chinese)
陈贵敏,贾建援,韩琪. 粒子群优化算法的惯性权值递减策略研究[J]. 西安交通大学学报,2006,40(1):53-56.
- [19] DAI W Z,YANG X L. Particle swarm optimization algorithm based on inertia weight logarithmic decreasing [J]. Computer Engineering and Application,2015,51(17):14-19. (in Chinese)
戴文智,杨新乐. 基于惯性权重对数递减的粒子群优化算法[J]. 计算机工程与应用,2015,51(17):14-19.
- [20] LI H L,LUO L,PU D M,et al. Improved particle optimization algorithm based on Cauchy distribution [J]. Electronic Science and Technology,2016,29(1):33-35. (in Chinese)
黎红玲,罗林,蒲冬梅,等. 基于柯西分布的粒子群优化算法改进[J]. 电子科技,2016,29(1):33-35.
- [21] JIANG J G,TIAN W,WANG X Q,et al. Adaptive particle swarm optimization via disturbing acceleration coefficients[J]. Journal of Xidian University,2012,39(4):74-80. (in Chinese)
姜建国,田旻,王向前,等. 采用扰动加速因子的自适应粒子群优化算法[J]. 西安电子科技大学学报,2012,39(4):74-80.
- [22] ZHANG J K,LIU S Y,ZHANG X Q. Improved particle swarm optimization [J]. Computer Engineering and Design,2007,28(17):4215-4216. (in Chinese)
张建科,刘三阳,张晓清. 改进的粒子群算法[J]. 计算机工程与设计,2007,28(17):4215-4216.
- [23] LIU J S,HE J J,LI P F. Improved particle swarm optimization algorithm based on theory of complex adaptive system[J]. Computer Engineering and Application,2017,53(5):57-63. (in Chinese)
刘举胜,何建佳,李鹏飞. 基于CAS理论的改进PSO算法[J]. 计算机工程与应用,2017,53(5):57-63.
- [24] 龚纯,王正林. 精通MATLAB最优化计算[M]. 北京:电子工业出版社,2001.
- [25] WU R X,SUN H,ZHU D G,et al. Particle swarm optimization algorithm based on optimal particle guidance and Gauss perturbation[J]. Journal of Chinese Computer Systems,2016,37(1):146-151. (in Chinese)
吴润秀,孙辉,朱德刚,等. 具有高斯扰动的最优粒子引导粒子群优化算法[J]. 小型微型计算机系统,2016,37(1):146-151.
- [26] ZHENG C Y,ZHENG Q D,WANG X D,et al. Self-adaptive particle swarm optimization algorithm based on tentative adjusting step factor [J]. Computer Science,2009,36(11):193-195. (in Chinese)
郑春颖,郑全第,王晓丹,等. 基于试探的变步长自适应粒子群算法[J]. 计算机科学,2009,36(11):193-195.