

doi:10.19665/j.issn1001-2400.2021.04.016

一种自适应模拟退火粒子群优化算法

闫群民^{1,2}, 马瑞卿¹, 马永翔³, 王俊杰³

(1. 西北工业大学 自动化学院, 陕西 西安 710072;

2. 陕西省工业自动化重点实验室, 陕西 汉中 723001;

3. 陕西理工大学 电气工程学院, 陕西 汉中 723001)

摘要: 为了提高粒子群算法的寻优速度和精度, 避免陷入局部最优解, 提出一种自适应模拟退火粒子群优化算法。采用双曲正切函数来控制惯性权重系数, 进行非线性自适应变化; 利用线性变化策略控制社会学习因子和自我学习因子, 达到改变不同阶段寻优重点的目的; 引入模拟退火操作, 根据种群的初始状态设置一个温度, 根据米特罗波利斯准则和温度指导种群以一定的概率接受差解, 保证了算法跳出局部最优解的能力。为验证这种算法的效果, 选择7种典型测试函数与已有文献中提出的5种粒子优化算法进行对比实验, 根据寻优结果的平均值、标准差以及迭代次数等数据, 证明文中所提算法在迭代精度、收敛速度以及稳定性上都有很大的提升, 有效地弥补了经典粒子群算法的缺陷。

关键词: 粒子群优化; 模拟退火; 惯性权重系数; 自适应调整策略

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1001-2400(2021)04-0120-08

Adaptive simulated annealing particle swarm optimization algorithm

YAN Qunmin^{1,2}, MA Ruiqing¹, MA Yongxiang³, WANG Junjie³

(1. School of Automation, Northwestern Polytechnical University, Xi'an 710072, China;

2. Shaanxi Key Laboratory of Industrial Automation, Hanzhong 723001, China;

3. Department of Electrical Engineering, Shaanxi University of Technology, Hanzhong 723001, China)

Abstract: Particle swarm optimization is widely used in various fields because of the few parameters to be set and the simple calculation structure. In order to improve the optimization speed and accuracy of the PSO, and to avoid falling into the local optimal solution, an adaptive simulated annealing PSO is proposed, which uses the hyperbolic tangent function to control the inertia weight factor for nonlinear adaptive changes, uses linear change strategies to control 2 learning factors, introduces the simulation annealing operation, set a temperature according to the initial state of the population, guide the population to accept the difference solution with a certain probability according to the Metropolis criterion, and ensure the ability to jump out of the local optimal solution. To verify the effect of the algorithm proposed in this paper, 7 typical test functions and 5 algorithms proposed in the literature are selected for comparison and testing. According to the average value, standard deviation and number of iterations of the optimization results, the algorithm proposed in this paper has greatly improved the iteration accuracy, convergence speed and stability so as to overcome the shortcomings of particle swarm optimization.

收稿日期: 2020-06-14

网络出版时间: 2021-03-03

基金项目: 陕西省教育厅重点科学研究计划项目(20JS018)

作者简介: 闫群民(1980—), 男, 教授, 西北工业大学博士研究生, E-mail: yanqunm@163.com

马瑞卿(1963—), 男, 教授, 博士, E-mail: marq@nwpu.edu.cn

马永翔(1965—), 男, 教授, 硕士, E-mail: mayx@snut.edu.cn

王俊杰(1996—), 男, 陕西理工大学硕士研究生, E-mail: wangjunjie9675@163.com

网络出版地址: <https://kns.cnki.net/kcms/detail/61.1076.TN.20210303.1129.011.html>

Key Words: particle swarm optimization; simulated annealing; inertia weight factor; self-adaptive adjust tactics

自然界中的生物均具有一定程度的群体行为。在计算机中建模分析这些行为是人工生命研究的一个重要方面,其背后隐藏的规律将对人类的生活产生很大的启发。粒子群算法(Particle Swarm Optimization, PSO)^[1]是 KENNEDY 博士和 EBERHART 教授通过对鸟群觅食模型进行改进所提出的。因为粒子群算法需设置的参数少且计算结构简单,目前已经广泛地应用在实际工程中,如多目标控制、神经网络研究、复杂多峰值优化求解问题等。众多的学者针对粒子群算法早熟和收敛速度慢等问题,做了大量的优化研究,主要可以分为3个方向:(1)最早出现的是针对粒子群算法自身迭代策略的改进,将原本固定不变的学习因子和惯性权重系数改进,为其设置随着迭代次数线性或非线性动态变化的调整策略^[2-6]。在搜索的初期、中期及末期等阶段有着不同的搜索重点,为达到平衡全局和局部搜索能力的要求,在保证搜索精度的情况下提升搜索速度。文献[6]中针对不同的迭代策略进行对比,经过仿真证明在初始权值和最终权值相同的情况下,正弦函数递减策略优于线性策略,而线性策略优于正切函数策略。(2)将粒子群算法的优势和其他优化算法进行融合,结合其他算法的优势来改进粒子群算法^[7-8]。文献[8]将牛顿最速下降因子引入粒子群算法中,提高了粒子群体的全局收敛性和收敛速度,克服了这种算法易“早熟”的缺点。(3)针对粒子种群的拓扑结构进行改进^[9-12]。文献[11]将原本的 D 维空间转化到混沌空间,根据混沌具有遍历性等特点进行迭代,提升了优化效果;文献[12]结合两种邻居拓扑结构,将种群分为3个子群,并采用不同的学习因子组合不同的搜索侧重点,经过一定迭代次数后对子种群的性能进行评估,指导粒子在不同子群间迁移,提升了算法的综合性。笔者将前两种类型进行改进,采用线性变化的策略控制社会及自我认知因子,利用双曲正切函数来控制惯性权重系数,在此基础上融合模拟退火算法提升了算法跳出局部最优解的能力。选出5种近几年被提出的改进粒子群算法的优化算法^[12-16],与笔者提出的算法共同通过 CEC2017 中选出的7种测试函数进行寻优测试(其中包含1种单峰函数和6种典型复杂多峰函数),结果证明该优化算法在寻优速度、收敛精度及算法稳定性等方面有显著的提升。

1 相关工作

1.1 粒子群算法

粒子群算法的寻优过程通过粒子在搜索空间的飞行完成。粒子每次迭代中的移动由3个部分组成,分别是对上一次速度的继承、自身学习以及种群的信息交互,其表达式为

$$v_i(k+1) = \omega v_i(k) + c_1 r_1 (P_{\text{best},i}(k) - x_i(k)) + c_2 r_2 (G_{\text{best}} - x_i(k)) \quad , \quad (1)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad , \quad (2)$$

其中, ω 为惯性权重系数; c_1 、 c_2 分别表示自身认知因子和社会认知因子,是控制粒子群算法的迭代最重要的参数; $x_i(k)$ 和 $v_i(k)$ 分别代表第 i 个粒子第 k 次迭代时的位置和速度; r_1 和 r_2 为随机数; $P_{\text{best},i}$ 为第 i 个粒子个体最优位置; G_{best} 为种群最佳位置。飞行粒子的搜索能力不仅受到粒子之前的速度影响,还受到两个学习因子指导。通过 c_1 控制自我认知部分,指导粒子向该粒子自身的历史最优解飞行; c_2 控制社会认知部分,体现种群中各粒子的信息交换,指导粒子向种群最优解的位置飞行。实际问题的寻优的终止条件设置为全局最优解满足适应度的要求,而实验中为比较不同算法的效果,设置迭代终止条件为达到最大迭代次数时终止。

每次粒子飞行前,先判断 $v_i(k)$ 是否越过设置的速度范围。若越过,则取速度边界值替代当前速度。飞行后再判断 $x_i(k)$ 是否超出最大搜索空间。若超出,同样取边界值替代当前值。根据相应的适应度值的变化来更新粒子群中 $P_{\text{best},i}$ 和 G_{best} ,更新方程为

$$P_{\text{best},i}(k+1) = \begin{cases} P_{\text{best},i}(k), & f(x_i(k)) \leq f(P_{\text{best},i}(k)) \\ x_i(k+1), & f(x_i(k)) > f(P_{\text{best},i}(k)) \end{cases} \quad , \quad (3)$$

$$G_{\text{best}}(k+1) = \begin{cases} G_{\text{best}}(k), & f(P_{\text{best},i}(k)) \leq f(G_{\text{best}}(k)) \\ P_{\text{best},i}(k), & f(P_{\text{best},i}(k)) > f(G_{\text{best}}(k)) \end{cases} \quad . \quad (4)$$

1.2 模拟退火法

模拟退火算法(Simulated Annealing, SA)是由米特罗波利斯在 20 世纪 50 年代根据对固态物质退火过程的研究而提出的一种算法,后在组合优化领域中得到大规模的应用推广,如现代的玻尔兹曼机、图像恢复处理、VLSI 最优设计、机器学习等科学领域中的组合优化问题。该算法是一种基于蒙特卡罗思想设计的近似求解最优化问题的方法,结构简单易实现,初始条件限制少,使用灵活且运行效率高,与其他算法的结合能力较强,有着广阔的前景。模拟退火算法的核心源自固体降温热力学过程,退火指物体的能量随着温度的降低达到最低值的过程。米特罗波利斯准则是模拟退火算法中的核心准则,其特别之处在于温度下降过程中不仅能够接受优值,也会根据温度变量产生的概率接受较差的值,增大了算法在寻优过程中跳出局部最优解的概率。将粒子群算法和模拟退火算法相结合,可以有效地减少粒子群算法陷入局部最优解的概率。

模拟退火算法在迭代初始阶段需要根据种群的初始状态设置一个初始温度。每次迭代对模拟固体内部粒子在温度下降情况下的移动,根据米特罗波利斯准则判断是否由于干扰产生的新解替代全局最优解,其表达式如下:

$$p_i(k) = \begin{cases} 1, & E_i(k) \geq E_g \\ \exp\left(-\frac{E_i(k) - E_g}{T_i}\right), & E_i(k) < E_g \end{cases}, \quad (5)$$

其中, $E_i(k)$ 表示第 i 个粒子在第 k 次迭代时的内能,即当前粒子的适应度值; E_g 表示当前种群最优点的内能; T_i 表示当前温度。温度每次迭代会以一定程度线性衰减,寻优过程是不断寻找新解和缓慢降温的交替过程。 $E_i(k)$ 完全决定了其下一次产生的新状态 $E_i(k+1)$,与之前的 $E_i(0)$ 至 $E_i(k-1)$ 无关,这个过程是一个马尔可夫过程。使用马尔可夫过程分析模拟退火步骤,经过有限次的转换在温度 T_i 下的平衡态的分布如下:

$$p_i(k) = \exp\left(\frac{-E_i(k-1)}{T}\right) / \sum_{j \in S} \exp\left(\frac{-E_j(k)}{T_i}\right), \quad (6)$$

$$p_i = \begin{cases} \frac{1}{|S_{\min}|}, & x_i \in S_{\min} \\ 0, & \text{其他} \end{cases} \quad \text{且} \quad \sum_{x_i \in S_{\min}} p_i = 1, \quad (7)$$

其中, S_{\min} 是最优值的搜索空间集合。当温度降到 0 时, p_i 的分布见式(7)。温度下降的同时伴随着大量的状态转移,达到热平衡的状态,则找到全局最优的概率为 1。模拟退火算法最大的优点是跳出局部最优解的能力突出。

2 自适应模拟退火粒子群算法

以往应用经典的粒子群算法设置 ω 、 c_1 和 c_2 的值时,大多依赖经验的判断,或者根据大量的仿真实验来确定一个固定的值。但通过上述的分析知,如果这 3 个参数能够随着优化的进行不断变化的话,粒子群算法将会有更加优秀的效果。惯性权重系数 ω 直接影响着算法搜索能力的强弱,控制粒子继承以往运动趋势,即搜索飞行的惯性。若需要算法的全局搜索能力较强,则设置 ω 的值比较大,但受前一次迭代速度的影响较大,相应的粒子飞行距离较大,不利于局部寻优。若需要进行局部精细解的搜索,提升局部区域的寻优能力,则设置 ω 的值较小,但陷入局部最优的概率会变大。 ω 值的选取与算法收敛速度和全局搜索能力成正比,与局部搜索能力成反比。笔者选取 $[-4, 4]$ 之间负的双曲正切曲线来控制惯性权重系数的变化。双曲正切曲线是一个非线性的控制策略,在搜索初期其递减速度较慢,给粒子充分的时间进行大范围的全局搜索,减小陷入局部最优的情况;中期近似线性递减,逐渐加强局部搜索的能力;后期变化率再次减小,着重细致的局部搜索,精准确定全局最优解。惯性权重函数如下:

$$\omega = (\omega_{\max} + \omega_{\min})/2 + \tanh(-4 + 8 \times (k_{\max} - k)/k_{\max}) (\omega_{\max} - \omega_{\min})/2, \quad (8)$$

其中, ω_{\max} , ω_{\min} 是惯性权重系数的最大值和最小值,文中取 $\omega_{\max} = 0.95$, $\omega_{\min} = 0.40$ 。经大量实验证明,采用如上取值时算法性能会大幅度提升^[2]。 k 是当前迭代次数, k_{\max} 是最大迭代次数,其图像见图 1。

当 $c_1 > c_2$ 时,粒子的运动更偏向个体最优的方向;反之,则更偏向群体最优方向。笔者所提的优化算法

初始阶段注重全局搜索,着重突出粒子的自我认知能力,注重粒子运动的遍历性,减小陷入局部最优解的概率。随着迭代的进行,加强粒子间的交流,使种群最优解的位置对每个粒子的运行起到更大的影响作用,着重对 G_{best} 的附近进行局部搜索。对应着这样的搜索策略,随着迭代次数的增加, ω 不断减小, c_1 逐渐减小, c_2 逐渐增大。笔者采取如下参数变化策略:

$$c_1 = c_{1max} - k(c_{1max} - c_{1min})/k_{max}, \quad (9)$$

$$c_2 = c_{2min} - k(c_{2min} - c_{2max})/k_{max}, \quad (10)$$

其中, c_{1max} , c_{1min} 分别是自我学习因子的最大值和最小值; c_{2max} , c_{2min} 分别是社会学习因子的最大值和最小值。参照文献[16]并进行

大量实验后,取 $c_{1max} = 2.50$, $c_{2max} = 1.25$, $c_{1min} = 1.25$, $c_{2min} = 2.50$ 。随着迭代次数 k 的增加, c_1 由 2.50 线性减小至 1.25,而 c_2 则由 1.25 逐步线性增加到 2.50,这样设置就满足了初期注重粒子在空间上的遍历性要求,增强了全局搜索的能力。在迭代次数过半时, $c_1 < c_2$,且差距越来越大,局部搜索的能力也随之增强。

将模拟退火算法中的米特罗波利斯准则引入迭代中。根据最初的粒子最优值设置初始温度,并且每次迭代后以一定的降温系数 μ 衰减。具体操作如下:

$$T(k) = \begin{cases} E(G_{best})/\log(0.2), & k=1 \\ T(k-1)\mu, & k>1 \end{cases}, \quad (11)$$

其中, T 为初始温度。每次迭代后,计算更新后位置的内能(适应度)与种群最优点内能的差距,根据式(5)计算得出的概率与 $\text{rand}()$ 进行对比,判断是否接受较差的解。取降温系数 $\mu = 0.95$ 。笔者所提出的自适应模拟退火粒子群算法对粒子群算法的 3 个重要参数进行了自适应变化且加入了模拟退火操作,增加了寻优的精度和速度。具体步骤如下:

步骤 1 设置搜索空间及搜索速度的边界值,设置种群规模及最大迭代次数 k_{max} 。

步骤 2 随机产生种群中所有粒子的初始位置和初始速度。

步骤 3 评价全局粒子的适应度值并记录 G_{best} ,根据式(11)设置模拟退火的初始温度。

步骤 4 根据式(8)~式(10)自适应地改变 ω 、 c_1 和 c_2 。

步骤 5 根据式(1)改变粒子速度,根据式(2)进行一次迭代寻优。

步骤 6 计算移动后粒子的适应度。

步骤 7 根据式(3)更新粒子的自身的历史最优位置。

步骤 8 根据式(5)计算接受新解的概率 $p_i(k)$ 。

步骤 9 以米特罗波利斯准则为依据,对比概率 $p_i(k)$ 与 $\text{rand}()$,判断是否由产生的新解替代全局最优解进行退火操作,更新温度。

步骤 10 判断是否达到最大迭代次数 k_{max} ,若未达到则返回步骤 4。

步骤 11 输出当前最优粒子,即寻优结果,算法终止。

3 实验及结果分析

3.1 测试函数及参数设置

为了验证笔者提出的模拟退火粒子群优化算法的效果,现从 CEC2017 测试集中选取出 1 种单峰函数 f_1 及 6 种多峰函数 $f_2 - f_7$ 来对笔者所提出的算法进行测试,根据要求设置其搜索空间。具体的函数表达式见表 1。

同时不同类型的粒子群算法的优化算法中选取 5 种,分别是 MSMP SO^[12] (Multi-population based Self-adaptive Migration PSO), RPPSO^[13] (PSO with Random Parameters), PSO-DAC^[14] (PSO based on Dynamic Acceleration Coefficients), NDPSO^[15] (PSO with inertia weight decay by Normal Distribution) 和 APSO-DA^[16] (Adaptive PSO via Disturbing Acceleration coefficients) 进行仿真。通过对比笔者所提算法与

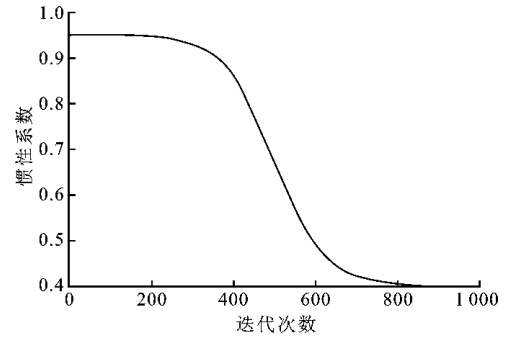


图1 惯性系数自适应变化图

这 5 种算法在搜索空间维数分别是 $D=10$ 和 $D=30$ 时,粒子群包含的粒子数相同,最大迭代次数相同的情况下的运行效果,来体现笔者所改进的算法的优越性。

表 1 测试函数信息表

函数	函数表达式	搜索空间
Sphere	$f_1 = \sum_{i=1}^D x_i^2$	$[-100, 100]$
Levy	$f_2 = \sin^2(\pi\omega_i) + \sum_{i=1}^{D-1} (\omega_i - 1)^2 [1 + 10\sin^2(\pi\omega_i + 1)] + (\omega_D - 1)^2 [1 + \sin^2(2\pi\omega_D)]$, $\omega_i = 1 + (x_i - 1)/4$	$[-10, 10]$
Ackley	$f_3 = -20\exp(-0.2(1/D \sum_{i=1}^D x_i^2)^{1/2}) - \exp((1/D) \sum_{i=1}^D \cos(2\pi x_i^2)) + 20 + e$	$[-30, 30]$
Griewank	$f_4 = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$
Rastrigin	$f_5 = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$
Rosenbrock	$f_6 = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-2.408, 2.408]$
Schwefel	$f_7 = 418.9829D - \sum_{i=1}^D x_i \sin x_i ^{1/2}$	$[-500, 500]$

3.2 算法精度对比

使用 MATLAB 2016b 环境进行仿真验证。所选取的 5 种对比算法的惯性权重系数和认知因子等参数都按照文献[12-16]中的设置,粒子种群规模统一设置为 30,最大迭代次数 $k_{\max} = 1000$ 。每个测试函数的维数在 $D=10$ 和 $D=30$ 的情况下分别运行 30 次,结果记录如表 2 及表 3 所示。从表 2~表 3 中根据 30 次寻优结果的平均值和标准差可以看出,无论是针对单峰函数还是多峰函数,本文算法针对多种测试函数的搜索精度都优于以往的算法。 $f_2 \sim f_5$ 这 4 个复杂的多峰函数的寻优精度,优于以往优化算法数十倍。 $f_1 \sim f_5$ 的寻优结果的标准差最小,有极强的搜索稳定性。

表 2 $D=10$ 时 6 种优化算法的运行结果对比

函数	准则	MSMPSO	RPPSO	PSO-DAC	NDPSO	APSO-DA	文中算法
f_1	avg	2.48×10^{-2}	8.24×10	5.72×10^2	4.24×10^{-2}	2.14×10^{-4}	0.00
	std	4.16×10^{-2}	8.99×10	3.65×10^2	8.67×10^{-2}	1.14×10^{-3}	0.00
f_2	avg	3.91×10^{-1}	1.36	2.47	3.87×10^{-2}	8.56×10^{-2}	3.70×10^{-2}
	std	6.88×10^{-1}	1.23	1.80	9.86×10^{-2}	1.01×10^{-1}	4.34×10^{-2}
f_3	avg	1.78	4.87	7.90	3.89×10^{-1}	8.88×10^{-16}	8.88×10^{-16}
	std	9.44×10^{-1}	2.02	1.99	5.12×10^{-1}	9.86×10^{-32}	9.86×10^{-32}
f_4	avg	3.10×10^{-1}	2.17	7.05	1.67×10^{-1}	3.36×10^{-3}	0.00
	std	1.70×10^{-1}	2.25	4.65	2.05×10^{-1}	1.78×10^{-2}	0.00
f_5	avg	9.76	2.14×10	2.19×10	2.47	1.33×10^{-1}	0.00
	std	5.38	9.54	8.32	5.11	7.03×10^{-1}	0.00
f_6	avg	6.92	1.56×10	5.58×10	7.14	7.41	6.81
	std	1.76	1.30×10	3.41×10	1.65	2.60	3.17
f_7	avg	9.38×10^2	1.58×10^3	1.72×10^3	8.81×10^2	8.30×10^2	8.22×10^2
	std	2.87×10^2	4.25×10^2	3.07×10^2	4.05×10^2	2.86×10^2	3.50×10^2

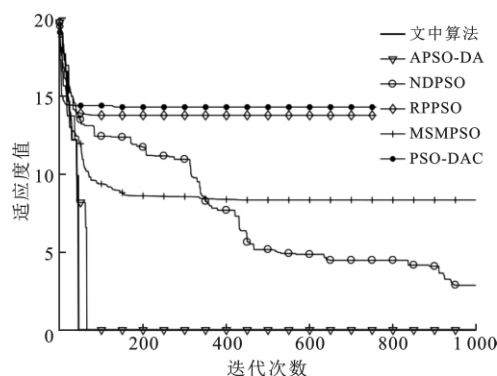
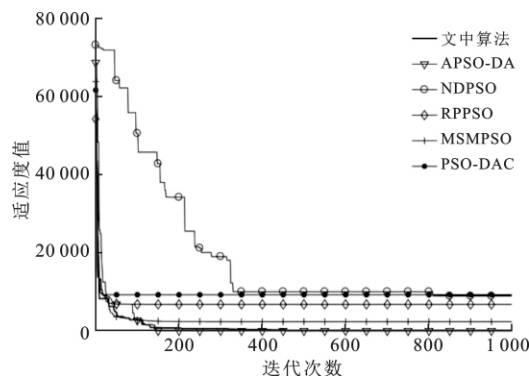
表3 $D=30$ 时6种优化算法的运行结果对比

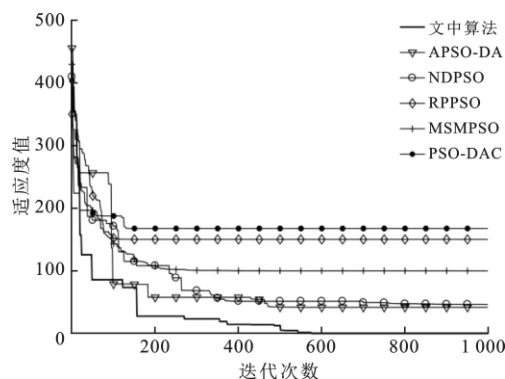
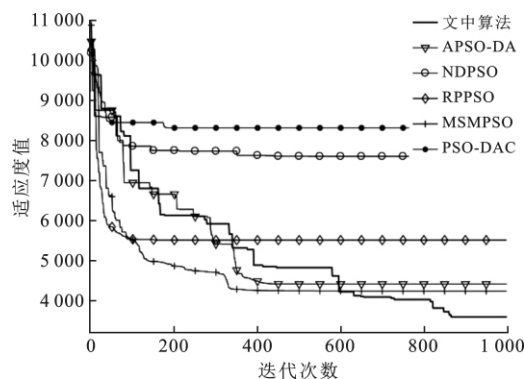
函数	准则	MSMPSO	RPPSO	PSO-DAC	NDPSO	APSO-DA	文中算法
f_1	avg	1.93×10^3	6.46×10^3	9.74×10^3	7.86×10^2	2.04	0.00
	std	9.43×10^2	2.48×10^3	2.79×10^3	2.53×10^3	7.13	0.00
f_2	avg	8.68	2.05×10	2.55×10	1.10×10	2.16	2.04
	std	3.48	6.89	9.43	7.70	2.48	2.31
f_3	avg	8.74	1.30×10	1.44×10	6.32	1.83×10^{-1}	8.88×10^{-16}
	std	1.88	1.05	1.07	5.63	5.16×10^{-1}	5.07×10^{-22}
f_4	avg	1.85×10	5.82×10	9.99×10	1.92×10	6.47×10^{-2}	4.83×10^{-2}
	std	8.19	1.79×10	3.71×10	4.97×10	2.47×10^{-1}	2.64×10^{-1}
f_5	avg	1.05×10^2	1.49×10^2	1.72×10^2	7.98×10	1.16×10	3.63
	std	2.06×10	2.49×10	2.77×10	3.86×10	1.70×10	8.68
f_6	avg	1.55×10^2	4.54×10^2	6.58×10^2	4.05×10	2.87×10	2.87×10
	std	7.72×10	1.95×10^2	1.96×10^2	2.41×10	2.13×10^{-1}	4.03×10^{-1}
f_7	avg	4.74×10^3	6.78×10^3	7.65×10^3	4.40×10^3	4.13×10^3	4.05×10^3
	std	9.14×10^2	5.80×10^2	6.10×10^2	1.02×10^3	7.36×10^2	6.62×10^2

搜索空间维度的增长,表示着每次迭代所需要处理的数据成倍提升,全局最优点位置信息也更加复杂,寻优任务变得十分困难。在实际工况下,面对高维问题首先要做的是降低维数,减少计算量。 $D=30$ 的这类高维数情况较少,但为证明优化算法面对高维问题时的性能,进行了针对 $D=30$ 时的寻优实验。实验中面对搜索空间维数从 10 增长到 30,大部分算法的收敛精度都大幅度下降,而笔者所提出的算法依然有着精准的寻优精度,针对高维数的复杂多峰测试函数有很强的寻优结果。

3.3 收敛速度对比

为了更加直观地对比不同算法在精度上的差别以及迭代速度的优劣性,选出部分测试函数的优化曲线图。图 2~图 5 分别显示在高维数 $D=30$ 时,笔者所提算法与 5 种算法的对比。图 2 显示在面对简单的单峰函数 f_1 时,文中算法的速度优势并不突出,个别算法也具有较快的速度和精度。由图 3~图 5 可知,在面对 f_3 、 f_5 和 f_7 复杂多峰函数时,笔者所提算法在收敛速度上的优势得到显著的体现,寻优速度快的同时也保证了收敛的精度。观察图 5,面对搜索空间极大、峰值极多且分布极不规则的 Schwefel 测试函数,并且这个函数的各个局部优解的差值极小,寻优难度较大。可以看出在迭代到中期(500 次左右)时大部分算法已经陷入局部优解,因为这些算法的寻优过程在这个阶段,种群中的大部分粒子已经聚集在局部最优处。随着越来越多粒子受种群最优的影响被引向局部最优解处,所以跳出局部最优解的能力逐步减弱至丧失;但笔者所提的算法虽然在初始迭代的前 400 次,其精度优势并不突出,但其优势突出体现在其因为融合了模拟退火操作,极大程度地增强了跳出局部最优的能力,在迭代 800 多次的后期,仍然可以不断地更新全局最优值,保证跳出局部最优的能力,提升搜索结果的精准性。综上所述可知,笔者所提的算法在收敛速度上也具有一定程度的优势。

图2 f_1 函数的6种方法优化曲线对比图图3 f_3 函数的6种方法优化曲线对比图

图 4 f_5 函数的 6 种方法优化曲线对比图图 5 f_7 函数的 6 种方法优化曲线对比图

4 结束语

采用双曲正切函数控制惯性权重系数随着迭代进行非线性变化,用线性变化策略控制认知因子 c_1 、 c_2 ,从而保证了 3 个控制搜索性能参数都能随着迭代进行自适应变化;同时,在粒子群算法寻优过程中加入模拟退火操作,通过米特罗波利斯准则引导种群以一定的概率接受差解,弥补了粒子群算法易陷入局部最优的缺陷,极大地提升了粒子群算法的搜索能力,满足了算法在不同阶段的搜索要求。面对多种高维多峰的测试函数,与以往的粒子群优化算法的寻优效果进行对比,其寻优精度提升数十倍,能精准地收敛在全局最优处,以较快的收敛速度完成寻优目标。

参考文献:

- [1] KENNEDY J, EBERHART R. Particle Swarm Optimization[C]//Proceedings of the IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.
- [2] SHI Y, EBERHART R. A Modified Particle Swarm Optimizer[C]//Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings. Piscataway: IEEE, 1998: 69-73.
- [3] TANG Y, WANG Z, FANG J A. Feedback Learning Particle Swarm Optimization[J]. Applied Soft Computing Journal, 2011, 11(8): 4713-4725.
- [4] YAN C, LU G, LIU Y, et al. A Modified PSO Algorithm with Exponential Decay Weight[C]//Proceedings of the 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. Piscataway: IEEE, 2017: 239-242.
- [5] 全秋娟, 李萌, 赵岂. 基于分类思想的改进粒子群优化算法[J]. 现代电子技术, 2019, 42(19): 11-14.
TONG Qiujuan, LI Meng, ZHAO Qi. An Improved Particle Swarm Optimization Algorithm Based on Classification[J]. Modern Electronics Technique, 2019, 42(19): 11-14.
- [6] 周敏, 李太勇. 粒子群优化算法中的惯性权重非线性调整策略[J]. 计算机工程, 2011, 37(5): 204-206.
ZHOU Min, LI Taiyong. Nonlinear Adjustment Strategy of Inertia Weight in Particle Swarm Optimization Algorithm[J]. Computer Engineering, 2011, 37(5): 204-206.
- [7] CHEN W, ZHANG N. Particle Swarm Optimization with An Aging Leader and Challengers[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(2): 241-258.
- [8] 李江杰, 常安定, 陈童, 等. 融合牛顿-最速下降算子的自适应粒子群算法[J]. 微电子学与计算机, 2020, 37(2): 1-7.
LI Jiangjie, CHANG Anding, CHEN Tong, et al. An Adaptive Particle Swarm Optimization Algorithm with Newton-steepest Descent Operator[J]. Microelectronics and Computer, 2020, 37(2): 1-7.
- [9] XIA X W, XIE C W, WEI B, et al. Particle Swarm Optimization Using Multi-level Adaptation and Purposeful Detection Operators[J]. Information Sciences, 2017, 385: 174-195.
- [10] 齐欣, 靳雁霞, 张晋瑞, 等. 大数据环境中简化粒子群算法的改进研究[J]. 微电子学与计算机, 2020, 37(2): 25-29.
QI Xin, JIN Yanxia, ZHANG Jinrui, et al. Improvement of Simplified Particle Swarm Optimization in Big Data Environment[J]. Microelectronics and Computer, 2020, 37(2): 25-29.

- [11] 刘道华,原思聪,兰洋,等.混沌映射的粒子群优化方法[J].西安电子科技大学学报,2010,37(4):764-769.
LIU Daohua, YUAN Sicong, LAN Yang, et al. Method of Particle Swarm Optimization Based on the Chaos Map[J]. Journal of Xidian University, 2010, 37(4): 764-769.
- [12] 邓先礼,魏波,曾辉,等.基于多种群的自适应迁移 PSO 算法[J].电子学报,2018,46(8):1858-1865.
DENG Xianli, WEI Bo, ZENG Hui, et al. A Multi-Population Based Self-Adaptive Migration PSO[J]. Acta Electronica Sinica, 2018, 46(8): 1858-1865.
- [13] 黄少荣.基于随机参数的粒子群优化算法[J].重庆师范大学学报:自然科学版,2013,30(6):123-127.
HUANG Shaorong. A New Particle Swarm Optimization with Random Parameters[J]. Journal of Chongqing Normal University: Natural Science, 2013, 30(6): 123-127.
- [14] 滕志军,吕金玲,郭力文,等.基于动态加速因子的粒子群优化算法研究[J].微电子学与计算机,2017,34(12):125-129.
TENG Zhijun, LV Jinling, GUO Liwen, et al. Research on Particle Swarm Optimization Based on Dynamic Acceleration Coefficients[J]. Microelectronics and Computer, 2017, 34(12): 125-129.
- [15] 徐浩天,季伟东,孙小晴,等.基于正态分布衰减惯性权重的粒子群优化算法[J].深圳大学学报:理工版,2020,37(2):208-213.
XU Haotian, JI Weidong, SUN Xiaoqing, et al. A PSO Algorithm with Inertia Weight Decay by Normal Distribution[J]. Journal of Shenzhen University: Science and Engineering, 2020, 37(2): 208-213.
- [16] 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.
JIANG Jianguo, TIAN Min, WANG Xiangqian, et al. Adaptive Particle Swarm Optimization Via Disturbing Acceleration Coefficients[J]. Journal of Xidian University, 2012, 39(4): 74-80.

(编辑:郭 华)

(上接第 119 页)

- [11] CARNS P, HARMS K, JENKINS J, et al. Impact of Data Placement on Resilience in Large-Scale Object Storage Systems [C]//Proceedings of the 2016 32nd Symposium on Mass Storage Systems and Technologies (MSST). Piscataway: IEEE, 2016: 1-12.
- [12] ZHANG M, HAN S J, LEE P P C. SimEDC: A Simulator for the Reliability Analysis of Erasure-Coded Data Centers[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(12): 2836-2848.
- [13] 穆飞,薛巍,舒继武,等.一种面向大规模副本存储系统的可靠性模型[J].计算机研究与发展,2009,46(5):756-761.
MU Fei, XUE Wei, SHU Jiwu, et al. Reliability Model for Large-Scale Copy Storage System[J]. Journal of Computer Research and Development, 2009, 46(5): 756-761.
- [14] HUANG C, SIMITCI H, XU Y, et al. Erasure Coding in Windows Azure Storage [C]//Proceedings of the 2012 USENIX Annual Technical Conference (USENIX ATC 12). Piscataway: IEEE, 2012: 15-26.
- [15] SHAHABINEJAD M, KHABBAZIAN M, ARDAKANI M. A Class of Binary Locally Repairable Codes[J]. IEEE Transactions on Communications, 2016, 64(8): 3182-3193.
- [16] TAMO I, BARG A, FROLOV A. Bounds on the Parameters of Locally Recoverable Codes[J]. IEEE Transactions on Information Theory, 2016, 62(6): 3070-3083.
- [17] HAFNER J L. WEAVER Codes: Highly Fault Tolerant Erasure Codes for Storage Systems [C]//Proceedings of the 4th USENIX Conference on File and Storage Technologies. Piscataway: IEEE, 2005: 16.
- [18] HU Y, LIU Y, LI W, et al. Unequal Failure Protection Coding Technique for Distributed Cloud Storage Systems[J]. IEEE Transactions on Cloud Computing, 2021, 9(1): 386-400.
- [19] 郝晓慧,车书玲,张欣瑜. LRC 码最小距离限的深入分析[J].西安电子科技大学学报,2018,45(5):75-79.
HAO Xiaohui, CHE Shuling, ZHANG Xinyu. In-depth Analysis of the Minimum Distance Limit of LRC Codes[J]. Journal of Xidian University, 2018, 45(5): 75-79.
- [20] HAFNER J L, RAO K K. Notes on Reliability Models for Non-MDS Erasure Codes[R]. New York: IBM, 2006.

(编辑:牛姗姗)