

自适应变异粒子群优化算法及在新冠肺炎疫情传播预测中的应用

季伟东¹, 徐浩天¹, 林平²

¹ (哈尔滨师范大学 计算机科学与信息工程学院, 哈尔滨 150025)

² (哈尔滨医科大学, 哈尔滨 150086)

E-mail: 15189871428@163.com

摘要: 为克服粒子群算法容易陷入局部最优和全局寻优精度不高的缺点, 通过对算法的局部寻优和全局寻优的特点进行分析, 首先使用正态分布衰减策略改进惯性权重; 同时基于算法运行的时间自适应采用不同的基于高斯分布及柯西分布的变异优化策略, 解决全局搜索和局部开发能力的不平衡问题, 实现了局部寻优和全局寻优的双重优化, 满足了提高寻优速度和寻优精度的目的. 为验证算法有效性及实用性, 将改进算法用于预测新冠肺炎疫情传播情况. 选取 100 天新冠肺炎疫情每日新确诊人数的数据, 利用算法优化前馈神经网络的参数, 使训练好的神经网络模型预测新冠肺炎疫情的性能提升. 实验部分首先使用测试函数将改进的算法与其它五种算法进行对比, 验证算法的良好性能, 最后应用到神经网络完成新冠肺炎疫情传播预测.

关键词: 粒子群算法; 自适应; 变异优化; 新冠肺炎; 传播预测

中图分类号: TP301

文献标识码: A

文章编号: 1000-1220(2021)03-0472-06

Adaptive Mutation Particle Swarm Optimization and Its Application in Predicting the COVID-19 Epidemic Transmission

Ji Wei-dong¹, XU Hao-tian¹, LIN Ping²

¹ (School of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China)

² (Harbin Medical University, Harbin 150086, China)

Abstract: Aiming at the shortcomings that particle swarm optimization is easy to fall into local optimization and low precision. By analyzing the characteristics of algorithm's local and global optimization, firstly, the Gaussian distribution decay strategy is used to improve the inertia weight. Based on the running time, Gaussian distribution and Cauchy distribution are adopted on mutation strategies adaptively. The algorithm solves the imbalance between of global search and local search, realizes the dual optimization of local optimization and global optimization, and improves the convergence speed and accuracy. COVID-19 epidemic situation is predicted by the improved algorithm in order to verify the effectiveness and practicability. The 100 days' data of COVID-19 daily newly diagnosed patients are collected and the algorithm is used to optimize the parameters of feedforward neural network, so that the trained neural network model can improve the prediction performance. The experiment first compares the improved algorithm with the other five algorithms, and shows the good performance of the improved algorithm. Finally, the algorithm is applied to the prediction of the COVID-19.

Key words: particle swarm optimization; adaptively; mutation optimization; COVID-19; transmission prediction

1 引言

群体智能算法通常是指模拟自然过程来解决复杂的工程问题的算法, 这些算法具有群体搜索性、多进程并行性、自适应、自学习等特质, 代替传统算法解决含有多个极值和高维复杂优化问题. 经过多年的发展, 在参数优化、神经网络训练等领域得到广泛运用. 而其中的粒子群算法原理简明, 参数少而受研究者们青睐. 为了解决大规模高维复杂函数优化问题, 出现很多改进粒子群优化算法, 这些算法通常针对某一方面进行改进来强化全局搜索或者局部搜索能力. 譬如对粒子群算法参数进行改进, Shi^[1] 等将惯性权重因子加入速度位移公

式, 得到线性递减权重的粒子群算法, 算法的全局和局部搜索得到了较好的平衡. Yan^[2] 等引入收缩因子, 该因子控制粒子的移动轨迹, 提升算法全局开发能力, 使得算法能够得到及时收敛. 从算法的拓扑结构角度研究, Cheng^[3] 等设计的 SLPSO 将社会学习机制引入粒子群优化, 每个粒子向种群中任何一个更优的粒子学习, 将社会学习机制的能力即全局搜索能力最大化. 梁静等^[4] 提出随机动态协同进化策略, 将其应用到多种群粒子群优化算法中, 加强算法局部搜索能力. 邓先礼^[5] 通过赋予不同子种群不同的搜索特性, 实现子种群间的协作和综合性能的提升. 除参数和拓扑结构, 重新设计速度位移更新的规则也可以对提升算法的搜索能力, 如经典的 CLP-

SO 算法^[6]中,粒子以不同维度从各个粒子学习经验来更新自身的速度.黄洋等^[7]在算法的位移更新公式引入 S 型函数,利用粒子与种群之间的适应度比值关系自适应调整搜索步长,提高了算法的搜索效率.而当今较为流行的是将粒子群优化算法与智能方法结合,反向学习策略是其中的佼佼者,Tizhoosh H R^[8]中提出了反向学习(Opposition-based learning, OBL).夏学文^[9]提出一种具备方向学习和局部学习能力的粒子群算法,反向学习保证全局探测能力,利用差分策略提升局部探测能力,较好的实现全局和局部双重优化.周凌云^[10]将邻域重心引入反向学习 PSO 中,保持种群多样性的同时充分利用种群搜索经验,增强全局搜索能力,有效提高了算法性能.

从中可以发现,目前的改进粒子群算法一般针对于局部搜索能力或全局搜索能力其中一方面进行强化.兼顾强化局部和全局搜索能力的算法也很容易会失去平衡搜索能力的控制力,最终导致算法本质上只强化部分能力,并未充分发挥粒子群算法的特点.针对该系列问题,受深度学习萤火虫算法^[11]和二阶振荡粒子群算法^[12]的启发.本文提出了一种自适应变异优化的学习策略,并将之应用到粒子群优化算法.利用算法优化神经网络参数,实现新冠肺炎疫情传播预测模型的构建.其创新点主要体现在 4 方面:

1. 设计了柯西变异和高斯变异的方法.

2. 设计了新的速度位移更新规则.

3. 设计了基于算法进度的自适应变异优化策略.

4. 算法应用到前馈神经网络完成新冠肺炎疫情预测模型.
- 最后通过实验结果验证了本文算法的性能优于其他改进算法,且具有较高的实用性.

2 相关工作

2.1 基于正态分布衰减的惯性权重

为使自适应变异优化策略性能最大化,针对传统粒子群算法的惯性权重因子,本文引入基于正态分布衰减的惯性权重^[13],惯性权重公式为式(1),惯性权重变化曲线如图 1 所示.由图 1 可见,正态分布衰减惯性权重变化规律可分为前后两个时期.

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) * \frac{1}{\sqrt{2\pi}\theta} e^{-\frac{(\frac{t}{t_{\max}})^2}{2\theta^2}}$$

(1)

其中 ω_{\max} , ω_{\min} 分别为最大和最小加权系数; t_{\max} 为最大迭代次数; t 为当前迭代次数; θ 描述了正态分布数据分布的离散程度,控制图形曲线,文献[13]提出满足最佳衰减权重曲线的 θ 等于 0.4433.

这样的惯性权重衰减策略可以满足以下规律:前期惯性权重处于最大值,使得算法全局搜索能力最大化,提高算法收敛速度;中期快速过渡,惯性权重衰减迅速,全局搜索能力减弱,局部开发能力增强;后期局部开发能力最大化,提高算法收敛精度.达到全局寻优和局部寻优共同优化的效果.

2.2 自适应变异优化

对粒子进行变异操作可以拓展算法的搜索范围.若进行变异操作后,采用贪心选择策略,粒子的适应度值优于之前的粒子的适应度值,算法便将之前粒子取而代之,从而获得更

优解.

表 1 8 个标准测试函数
Table 1 Eight benchmark test functions

| 测试函数 | 函数名 | 定义域 | 最优 |
|-------|------------|----------------|----|
| f_1 | Sphere | $[-100,100]$ | 0 |
| f_2 | Quartic | $[-1.28,1.28]$ | 0 |
| f_3 | Rosenbrock | $[-30,30]$ | 0 |
| f_4 | Step | $[-100,100]$ | 0 |
| f_5 | Ackley | $[-32,32]$ | 0 |
| f_6 | Rastrigin | $[-5.12,5.12]$ | 0 |
| f_7 | Griewank | $[-600,600]$ | 0 |
| f_8 | Schwefel | $[-10,10]$ | 0 |

本文设计柯西变异和高斯变异用于自适应变异优化.其中柯西变异参考柯西分布函数是一种数学期望不存在的连续型概率分布函数.其分布图如图 2 中的柯西分布所示,柯西变异操作如式(2)所示:

$$x(i)^{*} = x(i) + Cauchy * (pbest(i) - x(i))$$

(2)

其中, $x(i)$ 是第 i 个粒子的位置; $x(i)^{*}$ 是更新后的第 i 个粒子的位置; $pbest(i)$ 是第 i 个粒子的个体最优; $Cauchy$ 是基于柯西分布产生的随机数.

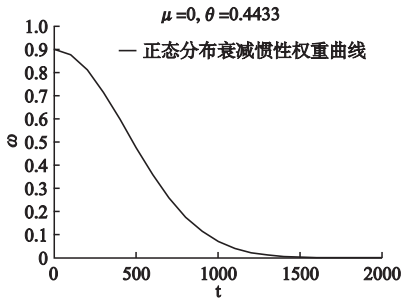


图 1 惯性权重曲线

Fig. 1 Inertial weight curve

高斯变异则参考高斯分布,高斯分布是一种连续随机型变量概率分布函数.其分布图如图 2 中的高斯分布所示,高斯变异操作如式(3)所示:

$$x(i)^{*} = x(i) + Gaussian * (pbest(i) - x(i))$$

(3)

其中, $x(i)$ 是第 i 个粒子的位置; $x(i)^{*}$ 是更新后的第 i 个粒子的位置; $pbest(i)$ 是第 i 个粒子的个体最优; $Gaussian$ 是基于柯西分布产生的随机数.

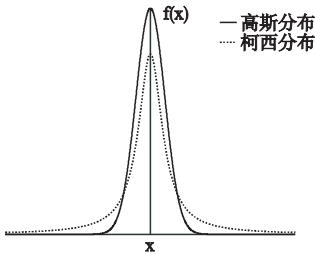


图 2 柯西分布与高斯分布

Fig. 2 Cauchy distribution and Gaussian distribution

从图 2 中可以看出,柯西分布在 x 轴上具有更宽的取值范围,意味着对应的柯西变异能帮助扩大粒子搜索的解空间

范围. 经过柯西变异的粒子能够搜索到更多更优的可行解, 由此可得知柯西变异适合前期的全局搜索. 而高斯变异在 y 轴上具有更深的取值范围, 经过高斯变异的粒子可以提升算法的收敛精度, 可以得知高斯变异适合后期的局部搜索.

自适应变异优化则是将算法整个过程分为两个部分: 前期与后期. 前期对粒子更新完成的位置进行柯西变异, 扩大全局搜索范围; 后期对粒子的位置进行高斯变异, 提升算法收敛精度.

3 自适应变异优化粒子群算法

3.1 速度公式的更新

为了避免改进的粒子群算法在搜索过程中陷入局部最优和收敛精度过低的问题, 给出了一种自适应速度更新公式, 用来控制个体最优和全局最优对粒子速度的影响度. 前期个体最优影响度更大, 后期全局最优影响度逐渐变大, 以此达到全局搜索和局部开发共同优化的效果. 给出粒子速度更新公式见公式(4).

$$v_{id}(t+1) = \omega * v_{id}(t) +$$
$$c_1 * r_1 * (pbest(t) - rand * (\frac{1}{2} - \frac{t}{t_{max}}) * x_{id}(t)) +$$
$$c_2 * r_2 * (gbest(t) - rand * (\frac{t}{t_{max}} - \frac{1}{2}) * x_{id}(t)) \quad (4)$$

其中 $v_{id}(t+1)$ 为更新后的粒子速度; ω 为正态分布衰减惯性权重; $pbest(t)$ 是粒子在第 t 代所得的最优位置信息; $gbest(t)$ 是种群在第 t 代所得的最优位置信息; t/t_{max} 为当前迭代次数与最大迭代次数的比值; $x_{id}(t)$ 为第 t 代粒子的位置信息; r_1 、 r_2 和 $rand$ 为 $[0,1]$ 之间均匀分布的随机数.

式(4)本质是由当前迭代次数与最大迭代次数的比值控制影响度. 由于算法的惯性权重采用正态分布惯性权重衰减策略, 并已证明其有效性, 结合此策略, 将算法整个迭代过程作为正态分布惯性权重衰减策略的 x 轴部分, 其中衰减曲线由前期缓慢衰减至中期加速收敛, 直到后期缓慢收敛, 除了中期的衰减过程的过渡, 前期和后期都分别给予粒子进行全局搜索和局部开发的必要的优化成本. 并且由图 1 中曲线特性可知, 整个迭代过程的 $1/2$ 处是衰减速度的转折点, 根据此特性结合个体最优与全局最优对速度的影响度, 设计出自适应变异优化策略速度更新公式. 算法前期, 当前 t/t_{max} 小于 $1/2$ 时, $pbest(t) - rand * (\frac{1}{2} - \frac{t}{t_{max}}) * x_{id}(t)$ 部分的影响度小于 $gbest(t) - rand * (\frac{t}{t_{max}} - \frac{1}{2}) * x_{id}(t)$; 当前 t/t_{max} 大于 $1/2$ 时, 则相反. 此公式自适应的控制了个体最优信息和全局最优信息对粒子速度更新的影响度. 其中 ω 的衰减策略则采用基于正态分布权重衰减策略.

3.2 自适应变异优化策略

由于柯西变异适合前期全局搜索, 高斯变异适合后期局部搜索^[14]. 根据算法迭代次数进行自适应选择变异方式, 前期选择柯西变异, 后期选择高斯变异. 如式(2)和式(3)中, 粒子利用自身找到的最优信息完成相应的变异优化. 这样的优化方式可以结合式(4)中的局部和全局最优的影响度, 前期让全局最优的影响度最大化, 使得算法前期能很好的拓展搜索范围, 寻得更多更优的可行解; 后期让局部最优的影响度最大化, 大大提升算法的收敛速度与收敛精度.

3.3 算法流程

将新的速度更新公式和自适应变异优化应用于粒子群优化算法, 得到本文的算法—自适应变异优化粒子群算法(ADVPSO).

算法流程如下:

Step 1. 采用均匀分布随机初始化种群, 生成由 N 个 D 维的粒子组成的种群 P ;

Step 2. 计算评价种群 P 中各个粒子的适应度值. 根据优劣更新个体历史最优解 $Pbest$ 、全局历史最优解 $Gbest$;

Step 3. 计算算法迭代次数与最大迭代次数的比值结果, 用式(1)更新惯性权重 ω ;

Step 4. 按照式(4)更新每个粒子的速度 v_i ;

Step 5. 根据算法迭代次数与最大迭代次数的比值判断, 比值如果小于或等于 $1/2$, 用式(2)优化更新粒子位置 x_i ; 否则用式(3)优化更新粒子的位置 x_i .

Step 6. 计算评价种群 P 中各个粒子的适应度值. 将其与个体历史最优 $Pbest$ 和全局历史最优 $Gbest$ 比较, 更新最优解.

Step 7. 判断算法是否满足终止条件, 若满足, 算法停止; 否则返回执行 Step 3.

4 实验对比与结果分析

4.1 参数设置与测试函数

为进一步验证 ADVPSO 的性能和优势, 实验选取了 LDWPSO (1998)^[1], RLP SO (2015)^[9], MSMP SO (2018)^[5], SAPSO (2019)^[7] 和 NDP SO (2020)^[13] 对比. LDWPSO 是原始粒子群算法, 其中引入了线性衰减惯性权重参数; 与 RLP SO 和 MSMP SO 对比, 是为了验证自适应变异优化策略对比其他智能方法策略的优越性; SAPSO 和 NDP SO 是近几年提出的新型改进 PSO 算法. 为了比较的公平性, 实验设置在同一硬件环境下, 6 个算法的参数设置如下: 种群大小 N 为 30; 维数 D 为 500 维; 迭代优化次数为 2000 次; 运行算法次数为 20 次; $c_1 = c_2 = 2$; ω_{max} 为 0.9, ω_{min} 为 0.4.

表2 各算法在 8 个标准测试函数上 20 次独立运行的结果

Table 2 Comparison of convergence results of six algorithms

| 算法 | $f_1(x)$ | $f_2(x)$ | $f_3(x)$ | $f_4(x)$ | $f_5(x)$ | $f_6(x)$ | $f_7(x)$ | $f_8(x)$ |
|---------|--|---|--------------------|--------------------|--|--------------------|--------------------|--|
| LDWPSO | 2.09×10^5 | 6.73×10^6 | 2.19×10^8 | 2.02×10^5 | 1.70×10 | 4.52×10^3 | 1.91×10^3 | 1.05×10^2 |
| RLPSO | 4.50×10^4 | 2.33×10^6 | 6.04×10^7 | 4.48×10^4 | 1.10×10 | 7.66×10^2 | 4.02×10^2 | 2.69×10 |
| MSMP SO | 7.20×10^2 | 1.20×10^3 | 4.74×10^4 | 7.98×10^2 | 0.3781 | 1.11×10^2 | 4.9879 | 0.9749 |
| SAPSO | 3.42×10^5 | 1.52×10^7 | 5.38×10^8 | 3.34×10^5 | 1.79×10 | 5.58×10^3 | 2.93×10^3 | 1.20×10^2 |
| NDP SO | 5.3847 | 6.8856 | 8.4334 | 5.3776 | 1.2369 | 3.6885 | 3.3507 | 2.0482 |
| ADVPSO | 9.04 $\times 10^{-39}$ | 1.10 $\times 10^{-3}$ | 4.99×10^2 | 1.20×10^2 | 8.88 $\times 10^{-16}$ | 0 | 0 | 9.12 $\times 10^{-22}$ |

为了客观的对 ADVPSO 算法的性能做出评价,选用两组基准测试函数^[15,16]进行测试.第1组是单峰测试函数: Sphere, Quartic, Rosenbrock 和 Step, 分别记为 f_1 - f_4 .第1组是复杂的多峰测试函数: Ackley, Rastrigin, Griewank, Schwefel, 分别记为 f_5 - f_8 .这些测试函数是群智能算法最常用的基准测试函数.8个典型标准测试函数定义如表1所示.其中测试函数的定义域为粒子的每一维度的数值取值范围,最优值为理论上对应测试函数的最优解,而算法的目的就是要求得此最优值.

4.2 实验结果与分析

将6个算法根据以上要求设定初始值,运行20次后取平均最优值进行比较,结果如表2所示.加粗值表示各算法中最优值.第1组测试函数 f_1 - f_4 是较为简单的单峰测试函数.从表2结果来看,由于算法初始设定 $D=500$ 维,而 LDWPSO、RLPSO、MSMPSO、SAPSO 由于高维度的复杂性,几乎不能找到单峰函数的最优解,并且过早的陷入局部最优. ADVPSO 在 f_1 和 f_2 上的收敛精度是最高的.除了 f_3 和 f_4 , ADVPSO 解

的平均最优值都远远小于其他算法.而 f_3 是单峰测试函数中较为复杂的病态测试函数,故而大部分算法在此函数上性能表现较差,但 ADVPSO 的解的精度仍要高于其他算法.在 f_4 函数中,由于其特殊的阶梯形状,使得算法容易陷入局部最优. ADVPSO 是根据迭代次数分别在前后期进行柯西变异, 高斯变异,虽然可以跳出部分阶梯,但仍会陷入局部最优,而其他算法几乎无法跳出局部最优,所以尽管精度都不高,但 ADVPSO 仍然有较强的性能.

第2组测试函数 f_5 - f_8 是较为复杂的多峰测试函数.这些测试函数往往有多个局部极小值,且分布位置特殊,极其容易引导算法陷入局部极小值. ADVPSO 算法在多峰测试函数中表现良好,在 f_6 - f_7 中, ADVPSO 都取得了最优解0.在 f_5 和 f_8 中, ADVPSO 取得的解精度也远远高于其他算法,因为 f_8 具有很深且距离最优解较远的局部极小值,相比其他算法, ADVPSO 的正态分布衰减权重策略能扩大算法前期的全局搜索范围,后期也有较好的局部搜索能力,很好的平衡了局部与全局搜索.

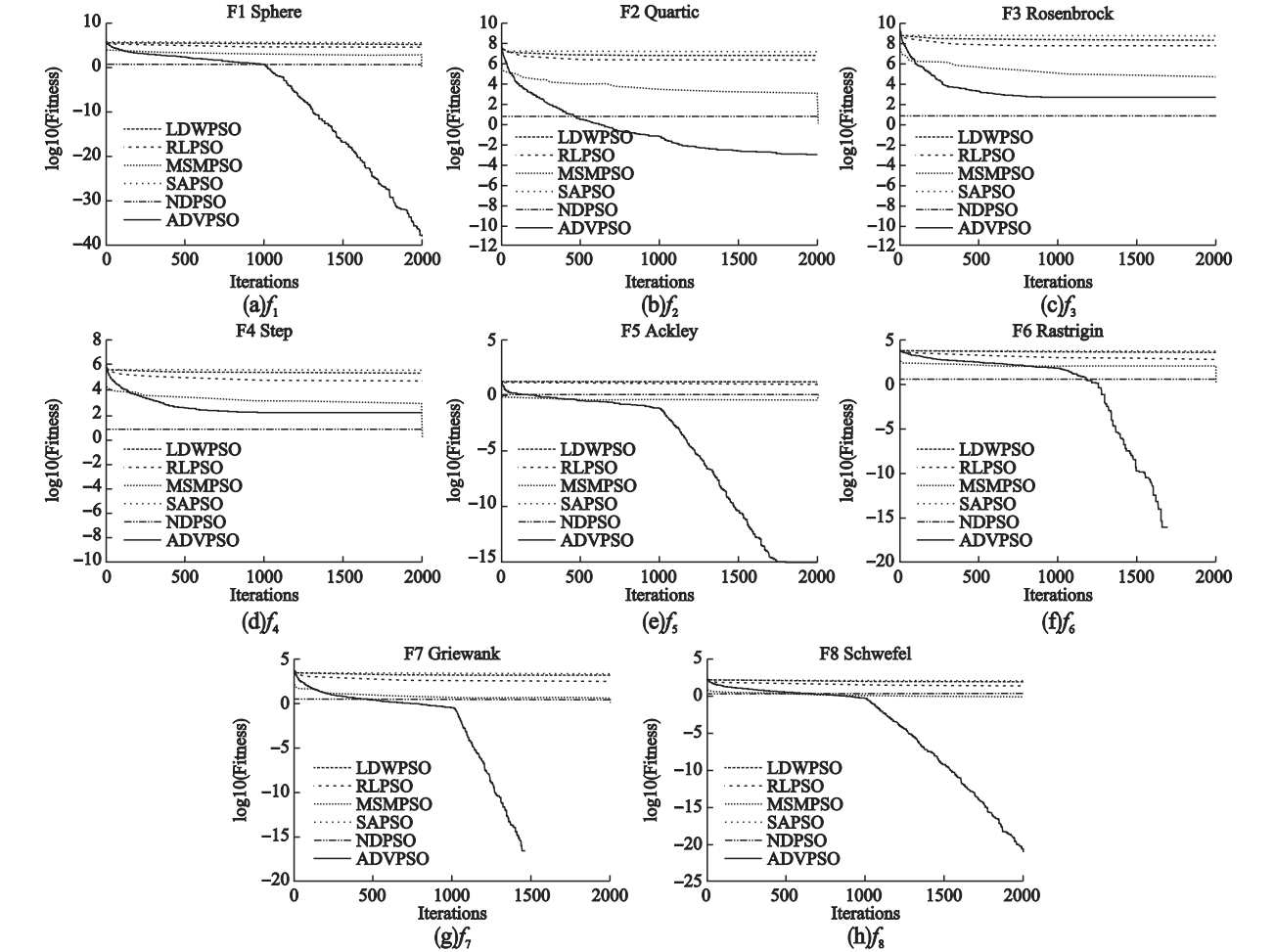


图3 各算法在 f_1 - f_8 上的收敛曲线
Fig. 3 Convergence curves of each algorithm on f_1 - f_8

从图3中可以看出,ADVPSO 的收敛性能远远优于 LDWPSO、RLPSO、MSMPSO、SAPSO、NDPSO 得益于 ADVPSO 算法的自适应变异优化策略.图3中(a,e,f,g,h)有一个明显的曲线转折点,当迭代次数为1000次时(即最大迭代次数的

一半),解的精度大幅度提升.自适应变异优化策略依靠局部和全局寻优的影响度,图2中可知柯西分布使粒子拥有较宽的取值范围,意味着对应的柯西变异能帮助扩大粒子搜索的解空间范围.经过柯西变异的粒子能够搜索到更多更优的可

行解,由此可得知柯西变异适合前期的全局搜索.而高斯分布在 y 轴上具有更深的取值范围,经过高斯变异的粒子可以提升算法的收敛精度,可以得知高斯变异适合后期的局部搜索.当算法前期陷入局部最优时利用正态分布惯性权重衰减策略的前期粒子收敛步长较大的特性,结合柯西变异拓展解空间,让粒子有更多的可行解进行搜索,及时跳出局部最优,后期利用正态分布惯性权重衰减策略后期粒子步长较小的特性,结合高斯变异加强局部搜索能力,提高解的精度.

实验分析后,ADVPSO 算法性能优于其他改进 PSO 算法,证明了自适应变异优化策略的有效性.

5 算法在新冠肺炎疫情传播预测中的应用

在全世界抗击新冠肺炎的过程中,涌现出大量对新冠肺炎疫情传播的研究.国内高校及科研院所利用模型对新冠肺炎疫情传播进行预测分析,其中张琳^[17]采用了次指数增长方式拟合了早期国内新冠肺炎疫情爆发的情况,并在拟合好的次指数增长模型基础上,预测疫情当前阶段的发展情况.盛华雄^[18]对武汉封城前后的新冠肺炎疫情传播进行建模分析,采用经典的 SIR 模型和差分递推方法预测疫情.以此为列,本文爬取中华人民共和国国家卫生健康委员会疫情通报中的每日新确诊人数数据¹,使用神经网络进行预测分析.其中神经网络中的各个权重参数(初始权值和阈值)由本文提出的 ADVPSO 算法进行优化.

将 2020 年 3 月 14 日-6 月 21 日共 100 天的每日新确诊人数进行数据与处理,处理结果如图 4 所示.从图中可以看出疫情爆发前期,每日新确诊人数居高不下,后期即使减少也偶尔有人数爆发,由于政府的有效控制以及人口流动等诸多未知因素,新确诊人数呈现非线性、非平稳和非高斯分布特性,因此数据的拟合存在困难且预测每日确诊人数准确率较低.如果采用普通的神经网络,则需要前期不断调试网络节点中的初始权值及阈值,这会花费大量的人力时间成本.本文因此采用改进的粒子群算法对神经网络中的权重参数及隐含层节点数进行迭代优化,使模型能更准确的预测,提高模型准确率.

本应用将采用三输入单输出的 3 层前馈神经网络预测新冠肺炎新确诊人数,网络输入 $z_i(t)$ 定义为 $z_1(t) = y_1^d(t-1)/200$, $z_2(t) = y_1^d(t-2)/200$, $z_3(t) = y_1^d(t-3)/200$, 其中 t 表示天数, $y_1^d(t)$ 表示为第 t 天的新确诊人数.利用前 64 天(2020 年 3 月 14 日-2020 年 5 月 16 日)的新确诊人数数据训练神经网络,神经网络输出表达式为 $\sum_{j=1}^4 w_{ji} \cdot \text{logsig}[\sum_{i=1}^3 (v_{ij}z_i(t) - b_j^1)] - \text{logsig}(b_j^2)$, 其中 v_{ij} 、 b_j^1 、 w_{ji} 、 b_j^2 为权值和阈值,搜索范围都设为 $[-10,10]$, sig 为 sigmoid 函数.

实验参数设置为:模型独立预测次数为 20 次,分析所得数据后,将训练集与测试集按 2:1 的比例进行数据分割.训练集为 2020 年 3 月 14 日-5 月 16 日,共 64 天的新确诊人数数据,测试集为 2020 年 5 月 17 日-6 月 14 日,共 36 天的新确诊人数数据,模型输入层节点数为 3,隐含层节点数为 4,输出层节点数为 1,其中粒子群优化权重参数的参数为:迭代次数为

500 次 $c_1 = c_2 = 2$; ω_{\max} 为 0.9, ω_{\min} 为 0.4.

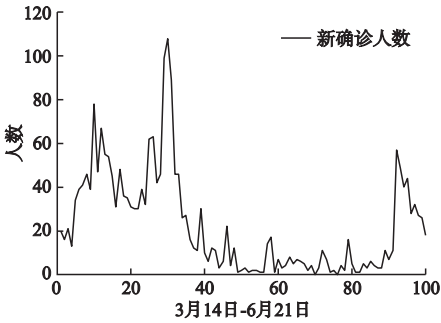


图 4 预处理后的新确诊人数
Fig. 4 Number of newly diagnosed patients after pretreatment

实验结果如图 5 所示,在进化初期,曲线上升速度快,说明神经网络具有较快的收敛速度,能有效的减弱常规进化算法的早熟收敛、易陷入局部最优解的现象.算法适应度值随着次数的增加而逼近最优解,平均适应度值最佳为 0.9546.

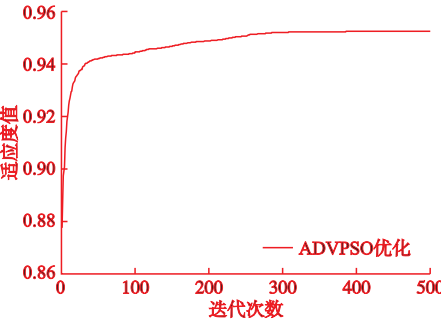


图 5 适应度值变化曲线
Fig. 5 Fitness curve

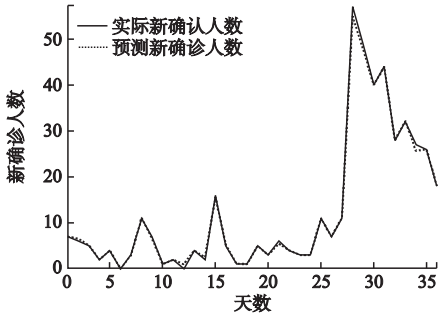


图 6 用 ADVPSO 预测新确诊人数曲线与实际新确诊人数曲线比较图
Fig. 6 Comparison between predicted and actual newly diagnosed cases

使用 ADVPSO 算法优化的神经网络预测模型在训练后,预测 2020 年 5 月 17 日-2020 年 6 月 21 日总计 36 天的新确诊人数,其中实际新确诊人数为实线,预测新确诊人数为虚线,如图 6 所示.

¹ http://www.wa.gov.cn/xcs/yqtb/list_gzbd.shtml

从图 6 中可以看出,使用 ADVPSO 算法优化的神经网络预测模型在训练后,预测新确诊人数与实际新确诊人数基本相同,由此可见本文使用的 ADVPSO 优化神经网络具有良好的预测新冠肺炎新确诊人数的能力,通过对神经网络的权值和阈值进行调整优化,使得算法适应度值不断逼近最优解,与此同时,神经网络预测模型的预测性能随着适应度值的逼近

也得到提升。
为了方便描述 ADVPSO 算法优化神经网络的优势,采用传统粒子群算法与 ADVPSO 进行实验比较,通过比较适应度值、误差(训练误差即训练集结果与元数据集的误差,预测误差为模型训练完毕后新预测的结果与元数据集的误差)和所获得的节点连接数,完成性能比较,实验结果如表3所示,加

表 3 两种算法实验结果比较

| 算法 | 最佳适应度值 | 平均适应度值 | 最劣适应度值 | 标准差 | 训练误差 | 预测误差 | 节点连接数 |
|--------|---------------|---------------|---------------|---|---------------|----------------|-----------|
| PSO | 0.9495 | 0.9457 | 0.9475 | 2.2236×10^{-3} | 13.5857 | 13.4134 | 9 |
| ADVPSO | 0.9566 | 0.9546 | 0.9477 | 2.1692×10^{-3} | 9.1045 | 11.4287 | 14 |

粗值为比较后的最优值。

6 结 语

本文提出了一种自适应变异优化粒子群算法,以算法运行进度作为参考,将整个寻优过程分为两个阶段.在此基础上,采用高斯分布惯性权重衰减策略动态更新 ω ,在不同阶段引导粒子的搜索行为.当粒子寻找到自身的最优解,便利用个体最优信息以及自适应变异机制优化粒子,实现了算法的高效搜索及计算资源的合理利用.实验结果表明 ADVPSO 具有良好的综合性能,在单峰及高峰函数中都表现出较好的鲁棒性.结合 2020 年新冠疫情爆发情况,本文将该算法应用至神经网络来建立预测模型,用 ADVPSO 算法优化神经网络权重参数,提高预测模型准确率.为验证 ADVPSO 算法的优势,将其与传统 PSO 算法优化的模型进行比较,ADVPSO 仍具有较好的鲁棒性和实用性.

其中,ADVPSO 对寻优过程的自适应划分并非最优划分方式.前期的全局搜索和后期的局部搜索对整个寻优过程的影响度还有待分析.这些研究将在后续工作中进一步解决.

References:

[1] Shi Y,Eberhart R C. A modified particle swarm optimizer [C]// Proceedings of the IEEE International Conference on Evolutionary Computation(IEEE ICEC), 1998:69-73.

[2] Yan Chun-man, Lu Gen-yuan, Liu Ying-ting, et al. A modified PSO algorithm with exponential decay weight [C]//Proceedings of International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery(ICNC-FSKD), 2018:239-242.

[3] Cheng Ran, Jin Yao-chu. A social learning particle swarm optimization algorithm for scalable optimization [J]. Information Sciences, 2015, 291 (2015): 43-60.

[4] Liang Jing, Liu Rui, Yu Kun-jie, et al. Dynamic multi-swarm particle swarm optimization with cooperative coevolution for large scale global optimization [J]. Journal of Software, 2018, 29 (9): 2595-2605.

[5] Deng Xian-li, Wei Bo, Zeng Hui, et al. A multi-population based self-Adaptive migration PSO [J]. Acta Electronica Sinica, 2018, 46 (8): 1858-1865.

[6] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal function [J]. IEEE Transactions on Evolutionary Computation, 2006, 10 (3): 281-295.

[7] Huang Yang, Lu Hai-yan, Xu Kai-bo, et al. S-shaped function based adaptive particle swarm optimization algorithm [J]. Computer Science, 2019, 46 (1): 245-250.

[8] Tizhoosh H R. Opposition-based learning; a new scheme for machine intelligence [C]//Proceedings of International Computational Intelligence for Modelling Control and Automation (CIMCA), 2005: 695-701.

[9] Xia Xue-wen, Liu Jing-nan, Gao Ke-fu, et al. Particle swarm opti-

mization algorithm with reverse-learning and local-learning behavior [J]. Chinese Journal of Computers, 2015, 38 (7): 1397-1407.

[10] Zhou Ling-yun, Ding Li-xin, Peng Hu, et al. Neighborhood centroid opposition-based particle swarm optimization [J]. Acta Electronica Sinica, 2017, 45 (11): 2815-2824.

[11] Zhao Jia, Xie Zhi-feng, Lv Li, et al. Firefly algorithm with deep learning [J]. Acta Electronica Sinica, 2018, 46 (11): 2633-2641.

[12] Jiang Li, Ye Run-zhou, Liang Chang-yong, et al. Improved second-order oscillatory particle swarm optimization [J]. Computer Engineering and Application, 2019, 55 (9): 130-138.

[13] Xu Hao-tian, Ji Wei-dong, Sun Xiao-qing, et al. A PSO algorithm with decay weight based on normal distribution [J]. Journal of Shenzhen University Science and Engineering, 2020, 37 (2): 208-213.

[14] Su Tong-fen. A modified immune particle swarm algorithm and research on cascaded reservoir optimal scheduling [D]. Taiyuan: Taiyuan University of Technology, 2010.

[15] Suganthan P N, Hansen N, Liang J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization [R]. Singapore: Nanyang Technological University, 2005.

[16] Yao Xin, Liu Yong, Lin Guang-ming. Evolutionary programming made faster [J]. IEEE Transactions on Evolutionary Computation, 1999, 3 (2): 82-102.

[17] Zhang Lin. Fitness of the generalized growth to the COVID-19 data [J]. Journal of University of Electronic Science and Technology of China. 2020, 49 (3): 345-348.

[18] Sheng Hua-xiong, Wu Lin, Xiao Chang-liang. Modeling analysis and prediction on NCP epidemic transmission [J]. Journal of System Simulation, 2020, 32 (5): 759-766.

附中文参考文献:

[4] 梁 静, 刘 睿, 于坤杰, 等. 求解大规模问题协同进化动态粒子群优化算法 [J]. 软件学报, 2018, 29 (9): 2595-2605.

[5] 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.

[7] 黄 洋, 鲁海燕, 许凯波, 等. 基于 S 型函数的自适应粒子群优化算法 [J]. 计算机科学, 2019, 46 (1): 245-250.

[9] 夏学文, 刘经南, 高柯夫, 等. 具备反向学习和局部学习能力的粒子群算法 [J]. 计算机学报, 2015, 38 (7): 1397-1407.

[10] 周凌云, 丁立新, 彭 虎, 等. 一种邻域重心反向学习的粒子群优化算法 [J]. 电子学报, 2017, 45 (11): 2815-2824.

[11] 赵 嘉, 谢智峰, 吕 莉, 等. 深度学习萤火虫算法 [J]. 电子学报, 2018, 46 (11): 2633-2641.

[12] 蒋 丽, 叶润舟, 梁昌勇, 等. 改进的二阶振荡粒子群算法 [J]. 计算机工程与应用, 2019, 55 (9): 130-138.

[13] 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.

[14] 苏同芬. 改进的免疫粒子群算法及梯级水库优化调度问题的研究 [D]. 太原: 太原理工大学, 2010.

[17] 张 琳. 新冠肺炎疫情传播的一般增长模型拟合与预测 [J]. 电子科技大学学报, 2020, 49 (3): 345-348.

[18] 盛华雄, 吴 琳, 肖长亮. 新冠肺炎疫情传播建模分析与预测 [J]. 系统仿真学报, 2020, 32 (5): 759-766.