

# 基于捕食搜索策略的模拟退火优化算法\*

张慕雪, 张达敏<sup>†</sup>, 杨菊靖, 朱陈柔玲

(贵州大学大数据与信息工程学院, 贵阳 550025)

**摘要:** 针对传统模拟退火算法初始温度和降温函数难以确定以及接收劣质解同时容易遗失当前最优解等缺陷, 将禁忌搜索算法的禁忌表功能引入 SA 算法, 避免遗失最优解和对某个解进行多次重复搜索; 根据函数的复杂程度确定初始温度, 并定义新的降温函数, 提高算法的搜索效率和精度; 引入捕食搜索策略, 平衡算法搜索能力和开发能力, 避免陷入局部最优。通过对五个典型的基准测试函数的仿真表明, 改进算法具有较强的全局搜索能力, 同时寻优精度和收敛速度比原算法也有较大的提高。

**关键词:** 模拟退火; 捕食搜索策略; 禁忌表; 初始温度; 降温函数

**中图分类号:** TP301.6

**文献标志码:** A

**文章编号:** 1001-3695(2018)09-2628-04

doi:10.3969/j.issn.1001-3695.2018.09.016

## Simulated annealing algorithm based on predatory search strategy

Zhang Muxue, Zhang Damin<sup>†</sup>, Yang Juqing, Zhu Chenrouling

(College of Big Data & Information Engineering, Guizhou University, Guiyang 550025, China)

**Abstract:** The traditional simulated annealing algorithm is difficult to determine the initial temperature and the cooling function as well as easy to receive the inferior solution at the same time losing the current optimal solution. In order to improve the global search capability and the computational efficiency of the simulated annealing, this paper considered to take the tabu table of tabu search algorithm into the simulated annealing to avoid the loss of the optimal solution and searching repeatedly for a solution. At the same time by determining the initial temperature according to the complexity of the function and defined the new cooling function to improve the efficiency and accuracy of the algorithm. Then it introduced predator search strategy to balance the search ability and development ability to avoid getting trapped into local optima. The results on five typical standard test functions show that the improved simulated annealing algorithm not only improves the global search ability, but also the search efficiency, search accuracy and convergence rate are better than the traditional simulated annealing algorithm.

**Key words:** simulated annealing; predatory search strategy; tabu table; initial temperature; cooling function

## 0 引言

模拟退火算法(simulated annealing, SA)是一种基于局部搜索的优化算法,算法出发点是利用物理中固体物质的退火过程与一般组合优化问题之间的相似性,达到求解全局最优解的目的<sup>[1,2]</sup>。但是 SA 算法对整个搜索空间的状况了解不多,不能使搜索过程进入最有希望的搜索区域,容易陷入局部最优<sup>[3]</sup>。捕食搜索(predatory search, PS)是在模拟动物群体捕食行为的基础上提出来的,用于解决组合优化问题的空间搜索策略。捕食动物在搜寻猎物时会集中搜索猎物集中的区域,而忽略猎物较少的区域,该算法具有较好的搜索效果<sup>[4,5]</sup>。

禁忌搜索算法(tabu search, TS)是一种启发式全局搜索算法,因其高效的局域搜索能力而著称。算法引入灵活的存储结构和相应的禁忌准则以避免重复搜索<sup>[3,6]</sup>。

SA 具有算法简单、能以一定的概率接收恶化解跳出局部最优、易于实现的特点,但同时该算法仍然有初始温度等参数难以控制、退火过程所需计算时间较长、没有记忆功能导致重复计算的缺陷<sup>[6]</sup>。为改善算法性能,文献[3]根据函数复杂度自适应调整步长控制参数,然后根据调整后的步长求得函数的粗糙解,在此基础上再使用初始步长求得全局最优解,保证算法最优解的精度。文献[6]加入禁忌表的功能,提出一种新的

温度控制机制,增加了搜索结束阶段进行爬坡移动的概率。文献[7]将粒子群算法与模拟退火算法相结合,利用粒子群算法的收敛速度、全局搜索能力强以及 SA 算法的具有跳出局部极值点区域能力的特点来解决约束优化问题。文献[8]提出 SASEA(simulated annealing steady-state evolutionary algorithm)算法,利用交叉算子构造模拟退火算法的邻域函数,并且将候选解嵌入到种群进化过程,利用 Metropolis 接受准则得到质量较好的子代解。文献[9]将布朗运动和模拟退火思想结合在一起,根据布朗运动提出新的邻域函数,并定义布朗运动温度下降函数。然而,目前研究表明,传统模拟退火算法计算效率较低、退火过程慢、收敛条件苛刻,不能使搜索过程进入最有希望的搜索区域,改进算法操作过于复杂,需要设定和控制的参数个数较多,并且各参数之间的相互影响也会对算法结果产生较大的影响,因此在应用上存在局限性。

本文提出一种基于捕食搜索策略的模拟退火优化算法(simulated annealing algorithm based on predatory search strategy, SAPS)。传统的模拟退火算法初始温度和降温函数难以确定,本文根据函数的复杂程度设定初始温度,并定义一种新的降温函数,在确保算法搜索效率和精度的前提下减少算法运行时间;引入禁忌搜索算法禁忌表的功能,增加记忆能力,避免因为接受劣质解而遗失最优解和对某个解进行多次重复的搜索;结

**收稿日期:** 2017-04-24; **修回日期:** 2017-06-02 **基金项目:** 贵州省合作计划项目(黔科合计省合[2014]7002); 贵州大学研究生创新基金项目(研理工 2016069)

**作者简介:** 张慕雪(1992-),女,贵州安顺人,硕士,主要研究方向为智能算法、数据挖掘(514330031@qq.com); 张达敏(1967-),男(通信作者),贵州贵阳人,教授,博士,主要研究方向为优化计算、网络拥塞控制; 杨菊靖(1993-),女,贵州贵阳人,硕士,主要研究方向为数据挖掘; 朱陈柔玲(1993-),女,江西抚州人,硕士,主要研究方向为数据挖掘。

合捕食搜索策略,使算法在有希望的区域内进行搜索,平衡算法的搜索能力和开发能力,避免盲目搜索陷入局部最优解。

## 1 模拟退火算法

### 1.1 模拟退火算法的原理

模拟退火算法在搜索区间随机游走,利用 Metropolis 抽样准则,使随机游走逐渐收敛于局部最优解。而温度是 Metropolis 算法中的一个重要控制参数,可以认为这个参数的大小控制了随机过程向局部或全局最优解移动的快慢<sup>[1,9,10]</sup>。Metropolis是一种有效的重点抽样法,其算法为:系统从一个能量状态变化到另一个状态时,相应的能量从  $E_1$  变化到  $E_2$ ,其概率为

$$p = \exp\left(-\frac{E_2 - E_1}{T}\right) \quad (1)$$

如果  $E_2 < E_1$ ,系统接收此状态;否则,以一个随机的概率接收或丢弃此状态。状态 2 被接收的概率为

$$p(1 \rightarrow 2) = \begin{cases} 1 & E_2 < E_1 \\ \exp\left(-\frac{E_2 - E_1}{T}\right) & E_2 \geq E_1 \end{cases} \quad (2)$$

这样经过一定次数的迭代,系统会逐渐趋于一个稳定的分布状态。重点抽样时,如果新状态向下,则接受;若向上,则以一定的概率接受。温度是一个重要的控制参数,模拟退火可视为递减控制参数  $T$  时的 Metropolis 算法的迭代。开始时  $T$  值大,可以接受较差的恶化解;随着  $T$  的减小,只能接受较好的恶化解;最后趋于 0 时,就不再接受任何恶化解了。

### 1.2 模拟退火算法的步骤

- 初始化:设置初始温度  $T_0$ 、初始解状态  $X_0$ 、每个  $T$  值的迭代次数  $L$ ;
- 对  $k = 1, \dots, L$  进行步骤 c) ~ f);
- 产生新解  $X'$ ;
- 计算增量  $\Delta E = E(X') - E(X)$ ,其中  $E(X)$  为评价函数;
- 若  $\Delta E < 0$ ,则接受  $X'$  作为新的当前解,否则以概率  $\exp(-\Delta E/T)$  接受  $X'$  作为新的当前解;
- 如果满足终止条件,则输出当前解作为最优解,结束程序;
- $T$  逐渐减小,且  $T \rightarrow 0$ ,然后转步骤 b)。

### 1.3 传统模拟退火算法的局限

传统模拟退火算法在迭代过程中不仅接受使目标函数变“好”的点,而且还能够以一定的概率接受使目标函数值变“差”的点,增加搜索过程的灵活性,而且还具有算法简单、易于实现等优点<sup>[10]</sup>。但是也明显存在以下缺点:

- 求解时间太长。在变量多、目标函数复杂时,为了得到一个较好的近似解,控制参数  $T$  需要从一个较大的值开始,并在每一个温度值  $T$  下执行多次 Metropolis 算法,因此迭代运算速度慢。
- 初始温度和温度下降函数难以确定。如果初始温度较高,而温度下降速度较慢,虽然能得到较好的解,但算法收敛速度太慢;如果初始温度较低,温度下降速度较快,则得不到全局最优解。
- 没有记忆功能。搜索过程由于以一定的概率接收劣质解而遗失当前遇到的最优解;对某个已经访问过的解进行多次重复的搜索,增加运行时间。

## 2 捕食搜索策略

捕食搜索算法是受到自然界中很多动物捕食策略启发而诞生的。生物学家发现,尽管很多生物的身体构造千差万别,

但是这些生物,包括鸟类、蜥蜴、昆虫等,捕食的搜索策略都惊人地相似。当这些生物在搜索猎物时,首先快速地沿着某一方向搜索,直到它们捕捉到猎物。此后,它们就会放慢速度,在发现猎物的地点附近一个很小的范围内继续搜索,试图找到更多的猎物。如果过了一段时间仍然没有发现新的猎物,它们就会放弃目前精密搜索的范围,转向其他区域,重新进行新的大范围搜索。如此循环,直到找到最优解为止<sup>[4,5]</sup>。整个过程如图 1 所示。生物的这种捕食搜索策略可以总结为以下两大搜索步骤:

搜索 1 广泛搜索(general search)。在整个搜索空间内进行全面搜索,如果找到猎物或者有猎物存在的迹象,就转到搜索 2。

搜索 2 区域精密搜索(area-restricted search)。在找到猎物地方的领域内进行精密搜索,如果在很长一段时间内没有进展,则转到搜索 1。

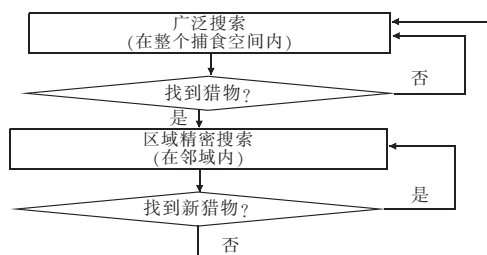


图 1 捕食搜索策略

生物学家发现动物的这种捕食搜索策略是非常高效的,这种看似普通而又简单的搜索策略,平衡了解空间开拓(exploration,在整个区域的广泛搜索)和解空间探索(exploitation,在某一地区进行精密搜索),而这种平衡恰恰正是优化算法所追求的目标。广泛搜索可以帮助找到一块很有可能猎物众多、适合精密搜索的区域,并且可以避免整个搜索过程被限制在一块很小的区域内;另一方面,精密搜索可以在广泛搜索所确定的限定区域内快速地找到很多猎物。这也正是这种策略能够被诸多物种高效使用的原因。

在捕食搜索中,使用限制(restriction)来表征较优解的邻域大小,通过限制的调节,实现搜索空间的增大和减小,从而达到搜索能力和开发能力的平衡。

## 3 改进的模拟退火算法

### 3.1 初始温度改进

温度  $T$  在算法中具有决定性的作用,它直接控制着退火的走向<sup>[1]</sup>。 $T$  的初值越大,获得高质量解的几率就越大; $T$  的初值过高,会使计算时间增加。而在传统的模拟退火算法中, $T$  的初值的确定是由大量的实验和试算得到的。由大量实验发现函数自身的复杂情况对算法步长、 $T$  的初值等相关参数有较大的影响<sup>[3]</sup>,一般希望  $T$  的初值根据函数的复杂情况来设定,既不过低而得不到全局最优解,也不过高增加计算时间。

为解决上述问题,使算法具有更高的搜索效率与搜索精度,本文参考文献[3]中函数复杂度的计算方法,对  $T$  的初值进行如下定义:

$$T = T_0 \times \ln(1 + \rho) \quad (3)$$

$$\rho = \log_{10}(f - g) + (m + n) \quad (4)$$

其中: $T_0$  为初始温度; $\rho$  为函数复杂度; $f$  是最大函数值; $g$  是最小函数值; $m$  是函数波峰数; $n$  是函数波谷数。

### 3.2 降温函数改进

降温函数用来控制温度的下降方式,温度的大小决定着 SA 进行广域搜索还是局域搜索。若温度下降过快,SA 将很快

从广域搜索转变为局域搜索,这就可能使算法陷入局部最优状态,为了跳出局部最优,只能通过增加内循环次数来实现,这会大大增加算法的运行时间;如果温度下降过慢,虽然可以减少内循环次数,但由于外循环次数增加,同样也会影响算法运行时间<sup>[11]</sup>。可见,合理的降温函数能提高算法的性能。在迭代初期,本文希望退火速率较快,而在退火的后期,具有下降平缓的特性来更好地进行精细搜索。本文定义一种新的降温函数,公式如下:

$$T(n+1) = T(n) \times \alpha^{1/\sqrt{\text{sq}(j)}} \quad (5)$$

其中:令  $k = \alpha^{1/\sqrt{\text{sq}(j)}}$ ,  $0 < \alpha < 1$  为降温系数,  $j$  为当前迭代次数。

从式(5)可以看出,迭代次数  $j$  与  $k$  成正比,也就是说,随着迭代次数的不断增加,  $k$  的值也不断增大,但仍然在  $[0, 1]$  内,这就满足了在迭代初期降温快,而在退火后期下降平缓的要求。指数函数在初期虽然有较快的下降速率,但其下降速率是恒定的,无法满足后期下降平缓的要求。与指数函数相比,新的降温函数有更优的寻优效率。

### 3.3 算法步骤

本文将捕食搜索策略的思想与模拟退火算法相结合,并将禁忌搜索算法的记忆功能应用到算法中,改进后的算法步骤为:

- 初始化各参数,并产生初始解;
- 在当前温度下根据初始解产生解集,并升序排列,加入限制,令  $\text{best} = \text{restriction}(0)$ , 将  $\text{best}$  加入禁忌表;
- 判断是否满足限制结束条件,满足转步骤 h), 不满足转步骤 d);
- 根据当前限制产生邻域解,选择邻域最优解,判断该解是否在禁忌表中出现过,出现过则重新产生邻域解,计算  $\Delta E = E(X') - E(X)$  判断接受新解还是以概率  $\exp(-\Delta E/T)$  接受或不接受,接受转步骤 e), 不接受转步骤 f);
- 更新限制,令  $\text{level} = 0$ , 将新解加入禁忌表,转步骤 d);
- 令  $\text{counter} = \text{counter} + 1$ , 判断是否满足限制循环次数,满足转步骤 g), 不满足转步骤 d);
- 令  $\text{level} = \text{level} + 1$ ,  $\text{counter} = 0$ , 转步骤 c);
- 温度衰减,产生新解,判断该解是否在禁忌表中出现过,出现过则重新产生新解,如果多次产生的解都在禁忌表中出现过,则认为已经没有更优解,跳出循环;
- 重复步骤 c) ~ h), 直到满足收敛条件;
- 找到禁忌表中的最优解,输出优化结果。

传统模拟退火算法有初温恒定,降温函数以恒定的降温率进行降温,重复搜索,容易陷入局部最优的缺陷。算法经过改进后,有四个优势:a)根据函数的复杂程度自适应设定函数的初始温度,避免增加算法运行时间;b)定义新的降温函数,使降温率随着迭代次数的增加而增大,达到迭代初期温度下降快,退火后期下降平缓的要求;c)增加记忆功能,避免对某个解进行多次重复的搜索,提高算法效率;d)引入捕食搜索策略,使算法在有希望的区域进行搜索,平衡算法搜索能力和开发能力,避免盲目搜索,陷入局部最优。

## 4 实验仿真

为了测试改进算法 SAPS 的有效性和客观性,本文进行了 SA、SAPS、BMSA<sup>[9]</sup>、SAGBA<sup>[14]</sup>、SAPSO<sup>[15]</sup> 和 ADSA<sup>[16]</sup> 在五个典型的基准测试函数中的优化性能比较。本文所用的五个典型基准测试函数均来自全局优化测试函数库<sup>[17]</sup>。函数形式、搜索区间、理论极小值如表 1 所示。其中:Sphere 和 Rosenbrock 为连续单峰函数,通常用于衡量算法的收敛速度;函数 Rastrigin、Griewank 和 Schaffor 是复杂的非线性多峰函数,存在大量

的局部极值,通常用于衡量算法的全局搜索性能。

表 1 算法测试函数

函数名称	函数形式	搜索区间	最小值
$f_1(x)$ Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-5.12, 5.12]$	0
$f_2(x)$ Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-5.12, 5.12]$	0
$f_3(x)$ Rastrigin	$f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-100, 100]$	0
$f_4(x)$ Griewank	$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0
$f_5(x)$ Schaffor	$f_5(x) = 0.5 + \frac{\sin^2(\sqrt{\frac{1}{2}(x_1^2 + x_2^2)}) - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$	$[-100, 100]$	0

在对算法进行测试时本文拟从以下三个方面进行:a)在  $T$  的初值、步长、衰减因子一定的情况下,比较寻优精度、收敛速度和稳定性;b)在固定的精度下,比较算法的运行时间和迭代次数;c)将 SAPS 与参考文献中的几种算法进行寻优性能对比。

### 4.1 初始温度、步长和衰减因子固定时性能测试

在测试 SA 和 SAPS 两种算法的寻优精度、收敛速度和稳定性时,设定  $T$  的初值为 100,步长为 0.02,衰减因子为 0.98,将 SA 和 SAPS 两种算法分别对选用的五个函数进行独立实验 50 次,测试结果如表 2 所示。表 2 分别从最差解、最优解、平均解和方差四个方面来进行对比,对两种算法的性能可以有更直观的认识。从实验结果可以看出,SAPS 对五个函数的寻优精度均比 SA 高出了 3、4 个数量级,这说明 SAPS 无论是对较简单的  $f_1(x)$  还是复杂度较高的其余四个函数,在寻优能力上都远远高于 SA,其极小值也更接近理论极小值。另外,SAPS 对较为复杂的四个函数( $f_2(x) \sim f_5(x)$ )的方差比 SA 的方差低很多,而对  $f_1(x)$  的方差只低了一个数量级,说明传统的 SA 在对较为简单的函数寻优时较为稳定,但当函数较为复杂存在多个极值点时,传统 SA 不仅得不到理想的寻优结果,而且算法稳定性也会大幅降低。总体来说,SAPS 比 SA 不仅在寻优能力方面更好,而且也具有更高的稳定性。

表 2 SA 与 SAPS 的性能测试对比

测试函数	算法名称	最差解	最优解	平均解	方差
$f_1(x)$	SAPS	4.3585E-07	3.1452E-08	6.8315E-08	1.7437E-10
	SA	1.8655E-04	5.9384E-06	5.6648E-05	3.0492E-09
$f_2(x)$	SAPS	7.9491E-07	1.1886E-08	2.1866E-07	4.2757E-13
	SA	1.2400E-02	3.4000E-03	1.0100E-03	2.0736E-07
$f_3(x)$	SAPS	6.9786E-05	1.5288E-07	2.8491E-06	7.5674E-10
	SA	4.0330E-01	1.7300E-02	6.7800E-02	1.4400E-02
$f_4(x)$	SAPS	7.6335E-05	1.2850E-07	8.2214E-07	1.4644E-10
	SA	1.6200E-02	9.8299E-05	7.6000E-03	4.4043E-05
$f_5(x)$	SAPS	8.2349E-05	8.4119E-06	3.5039E-06	7.7141E-10
	SA	1.5200E-02	9.7000E-03	1.1300E-02	4.2290E-06

### 4.2 固定精度下性能测试

设置目标精度为  $10^{-5}$ ,在固定测试精度的情况下,分别对 SAPS 和 SA 进行 20 次独立的实验,比较两种算法的运行时间和迭代次数。由于 Sphere 较为简单,所以目标精度设置为  $10^{-6}$ 。为使比较结果更具有说服力,运行时间和迭代次数均取 20 次实验的平均值。

从表 3 可以看出,在确定目标精度的情况下,SAPS 的迭代次数比 SA 的迭代次数降低了 1~2 个数量级,但 SAPS 的运行时间明显比 SA 的运行时间增加了,这是由于 SAPS 在一定程度上增加了代码的复杂度造成的,在得到更好的寻优精度、收

收敛速度和稳定性的情况下,适当增加算法的运行时间这是可以理解和忽略的。

表3 目标精度下算法的性能测试

函数名称	目标精度	SA		SAPS	
		迭代次数	耗时 $t_1/s$	迭代次数	耗时 $t_2/s$
$f_1(x)$	$10^{-6}$	1 283	0.222 5	59	1.368 7
$f_2(x)$	$10^{-5}$	9 875	0.982 0	153	13.820 1
$f_3(x)$	$10^{-5}$	1.763E+05	4.459 4	2 379	27.369 2
$f_4(x)$	$10^{-5}$	7.103E+04	7.283 0	1 697	25.532 9
$f_5(x)$	$10^{-5}$	2.854E+05	11.372 1	2 261	32.564 1

由图2可知,SAPS和SA对 $f_1(x)$ 的迭代次数都为53时,SAPS比SA的寻优精度高1个数量级;SAPS和SA对 $f_2(x)$ 的迭代次数都为167时,SAPS比SA的寻优精度高4个数量级;SAPS和SA对 $f_3(x)$ 的迭代次数都为2 332时,SAPS比SA的寻优精度高1个数量级;SAPS和SA对 $f_4(x)$ 的迭代次数都为1 765时,SAPS比SA的寻优精度高3个数量级;SAPS和SA对 $f_5(x)$ 的迭代次数都为2 138时,SAPS比SA的寻优精度高2个数量级。由此可见,不管是固定寻优精度比较迭代次数,还是固定迭代次数比较寻优精度,SAPS的性能都优于SA。

#### 4.3 与参考文献的算法性能对比

为比较本文算法与其他改进算法的优劣,将本文算法与参考文献中的几种算法进行比较,引用文献[9,14~16]的几组数据,如表4所示(“—”表示参考文献未给出)。

表4 与参考文献中算法性能的对比

函数	平均最优值					
名称	SAPS	SA	BMSA <sup>[9]</sup>	SAGBA <sup>[14]</sup>	SAPSO <sup>[15]</sup>	ADSA <sup>[16]</sup>
$f_1(x)$	6.8315E-08	5.6648E-05	5.00E-13	4.6101E-05	4.8602E-02	—
$f_2(x)$	2.1866E-07	1.0100E-03	4.00E-13	4.6779E-06	3.5383E-03	1.0500E-03
$f_3(x)$	2.8491E-06	6.7800E-02	—	1.5840E-07	1.9425E-04	2.9330E-05
$f_4(x)$	8.2214E-07	7.6000E-03	2.35E-06	3.6363E-08	5.7773E-05	1.7000E-03
$f_5(x)$	3.5039E-06	1.1300E-02	3.00E-06	—	1.6800E-10	—

首先,SAPS与SA的性能对比,SAPS比SA具有更优的性能。从表4可知,除Griewank函数外,BMSA的平均最优解比SAPS的平均最优解要低,这是在牺牲迭代次数的前提下得出的结果,当固定精度为 $10^{-6}$ 时,BMSA对 $f_1(x)$ 迭代次数的数量级为102,而SAPS的迭代次数为59,对其余四个函数,当固定精度为 $10^{-5}$ 时,BMSA的迭代次数分别为276、3 478和9 475,而SAPS的迭代次数分别为153、1 697和2 261,这表明SAPS比BMSA具有更优寻优能力。对比本文算法和文献[14]算法,SAPS在对 $f_1(x)$ 和 $f_2(x)$ 进行优化时,平均最优值比SAGBA高出了1~3个数量级,明显SAPS比SAGBA具有更优的优化性能。但对 $f_3(x)$ 和 $f_4(x)$ 进行优化时,SAGBA比SAPS高了1个数量级,分析SAGBA的进化曲线可知,SAGBA在对 $f_3(x)$ 和 $f_4(x)$ 进行优化时分别在第800次和第200次时很快收敛,但在进行到第1 350次和第1 190次迭代时出现拐点,以跳出局部最优解,从而可以说明在稳定性上SAGBA并不比SAPS更优。对比本文算法和文献[15]算法,本文算法除了对 $f_5(x)$ 函数的寻优能力低于SAPOS算法之外,对其他函数的寻优能力都高于SAPOS算法。其原因是SAPOS在对 $f_5(x)$ 进行优化时设置了较大的种群数和进化代数,这在一定程度上增加了算法的搜索范围,但同时也增加了算法的时间和复杂度,从对函数 $f_1(x) \sim f_4(x)$ 的优化可以看出在搜索能力上SAPS比SAPSO更优。最后,对比本文算法和文献[16]算法,SAPS的平均最优解比ADSA的平均最优解低了1~3个数量级,这是因为ADSA对算法增加了记忆功能和自适应降温的策略,这在本文的算法中也作了相应的改进,且本文算法加入了捕食搜索策略增加了算法的搜索和开发能力,使得SAPS的寻优能力比ADSA的寻优能力要好。

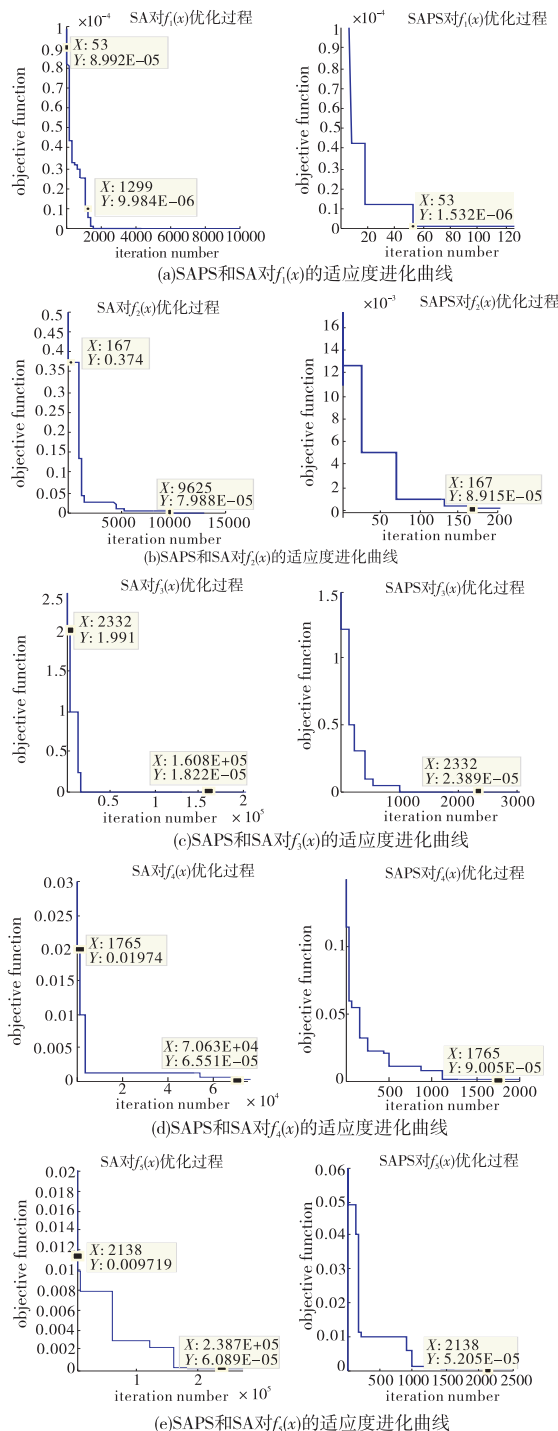


图2 两种算法适应度进化曲线对比

## 5 结束语

对于简单的函数,传统的SA算法能得到函数的最优解,但对于较为复杂的函数,标准的SA算法容易陷入局部最优解。本文充分考虑到初始温度和降温函数对算法的影响,根据函数的复杂情况设定初始温度,定义新的降温函数,增加记忆功能,并将捕食搜索策略引入算法中,平衡算法的搜索能力和开发能力,增加算法寻优能力,在保证求解精度的情况下,减少算法的迭代次数,提高了收敛速度和稳定性。

#### 参考文献:

- [1] 包子阳. 智能优化算法及其MATLAB实例[M]. 北京:电子工业出版社,2016.
- [2] 杨卫波,赵燕伟. 求解TSP问题的改进模拟退火算法[J]. 计算机工程与应用,2010,46(15):34-36.

(下转第2637页)



根据表3得到的误差柱状图如图9所示。

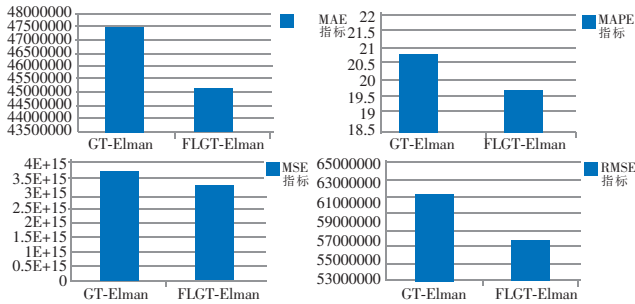


图9 GT-Elman 与 FLGT-Elman 预测指标

#### 4.4 实验结论

本实验一方面首先通过对原始时序数据进行特征提取操作,得到312维的特征集合。然后将该特征集合分别通过Clamping网络与DS-Clamping进行筛选,将两个特征选择模型筛选到的特征用于时间序列预测,分别得到对应的误差指标,通过计算DS-Clamping网络相对于Clamping网络在MAE、MAPE、MSE、RMSE指标方面分别下降了3.4%、3.29%、5.53%、2.9%。即改进的DS-Clamping相对于Clamping网络更能从特征集合中挑选出有利于时间序列预测的特征,从而能更好地拟合实际值。

另一方面本实验还通过直接使用原始时序数据特征作为GT-Elman模型的输入来预测时序数据与将原始时序数据通过特征学习模型得到的特征作为GT-Elman模型的输入来预测时序数据,通过对比两个模型的预测值与实际值之间的误差指标,得到互联网金融风险预测模型(特征学习模型+GT-Elman神经网络预测模型)相对于未使用特征学习模型的GT-Elman神经网络在MAE、MAPE、MSE、RMSE指标方面分别下降了4.77%、5.23%、12.89%、6.67%,即本章提出的互联网金融风险预测模型在时序数据预测性能上更优。

#### 5 结束语

本文通过对时间序列进行特征学习得到的特征通过神经网络来进行预测。通过改进的Elman神经网络,提出了GT-Elman神经网络预测模型应用于金融时序数据的预测;较于基础的Elman神经网络,GT-Elman神经网络模型在时序预测中具有更好的准确度。针对Clamping网络只考虑到了单一特征参数对结果的影响,改进了Clamping网络,提出了DS-Clamping应用于时序数据特征选择;通过使用改进型的Clamping网络,

能使时序数据预测的精度更高。虽然在通过改进的Elman与改进的Clamping网络能使预测准确度提高,但还是有一些尖峰数据没能很好地预测到,这主要的原因是只使用了原始的时序数据特征,而未用到用户的特征如用户的性别、城市、消费等一些用户特征,后续研究将考虑用户特征从而提高预测准确率。

#### 参考文献:

- [1] Brockwell P J, Davis R A. Introduction to time series and forecasting [M]. Berlin: Springer, 2009.
- [2] Yan Juan, Zhao Xiaodong, Li Kang. On temporal resolution selection in time series wind power forecasting [C]//Proc of the 11th International Conference on Control. Piscataway, NJ: IEEE Press, 2016: 1-6.
- [3] Jiang Shan, Wang Shuofeng, Li Zhiheng, et al. Fluctuation similarity modeling for traffic flow time series: a clustering approach [C]//Proc of IEEE International Conference on Intelligent Transportation Systems. Piscataway, NJ: IEEE Press, 2015: 848-853.
- [4] Nguyen T T, Huu Q N, Li M J. Forecasting time series water levels on Mekong river using machine learning models [C]//Proc of the 7th International Conference on Knowledge and Systems Engineering. Piscataway, NJ: IEEE Press, 2015: 292-297.
- [5] Liu Yingying, Thulasiraman P, Thulasiram R K. Parallelizing active memory ants with MapReduce for clustering financial time series data [C]//Proc of IEEE International Conferences on Big Data and Cloud Computing. Piscataway, NJ: IEEE Press, 2016: 137-144.
- [6] 聂淑媛. 时间序列分析的早期发展 [D]. 西安: 西北大学, 2012.
- [7] Zhan Shu, Li Weihao, Zhuang Xuan. A novel data mining algorithm based on BP neural network and its applications on stock price prediction [C]//Proc of International Conference on Materials Engineering, Manufacturing Technology and Control. 2016: 1688-1693.
- [8] Falat L, Marcek D, Durisova M. Intelligent soft computing on forex: exchange rates forecasting with hybrid radial basis neural network [J]. The Scientific World Journal, 2016, 2016(4): 3460293.
- [9] Li Ming, Wang Limin, Liu Yang, et al. An improved OIF Elman neural network model with direction profit factor and its applications [C]//Proc of the 2nd International Conference on Machine Vision. 2009: 208-211.
- [10] Elman J L. Finding structure in time [J]. Cognitive Science, 1990, 14(2): 179-211.
- [11] Niu Hongli, Wang Jun. Financial time series prediction by a random data-time adaptive RBF neural network [J]. Soft Computing, 2014, 18(3): 497-508.
- [12] Wang W, Jones P, Partridge D. Assessing the impact of input features in a feedforward neural network [J]. Neural Computing & Applications, 2000, 9(2): 101-112.
- [13] Rodriguez F J, Garcia M C, Lozano M. Hybrid meta-heuristics based on evolutionary algorithm and simulated annealing: taxonomy comparison and synergy test [J]. IEEE Trans on Evolutionary Computation, 2012, 16(6): 787-800.
- [14] 姚明海, 王娜, 赵连朋. 改进的模拟退火和遗传算法求解TSP问题 [J]. 计算机工程与应用, 2013, 49(14): 60-65.
- [15] Liu Minghua, Shi Yong, Yan Jiashu, et al. Lattice Boltzmann simulation of flow and heat transfer in random porous media constructed by simulated annealing algorithm [J]. Applied Thermal Engineering, 2016, 115(3): 1348-1356.
- [16] 贺兴时, 丁文静, 杨新社. 基于模拟退火高斯扰动的蝙蝠优化算法 [J]. 计算机应用研究, 2014, 31(2): 392-397.
- [17] 刘爱军, 杨育, 李斐. 混沌模拟退火粒子群优化算法研究及应用 [J]. 浙江大学学报: 工学版, 2013, 47(10): 1722-1730.
- [18] 孙士平, 吴建军. 直接搜索模拟退火算法的自适应改进 [J]. 计算机工程与应用, 2015, 51(23): 31-37.
- [19] Hedar J. Test functions for unconstrained global optimization [EB/OL]. [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar\\_files/Test\\_GO\\_files/Page364.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar_files/Test_GO_files/Page364.htm).

(上接第2631页)

- [3] 徐鹏飞, 苗启广. 基于函数复杂度的自适应模拟退火和禁忌搜索新算法 [J]. 电子学报, 2012, 40(6): 1218-1222.
- [4] 曹秀爽, 姚明林, 李兵. 具有捕食策略的混合随机优化算法及其多极值函数优化 [J]. 计算机应用, 2014, 34(S2): 162-165.
- [5] 张顶学, 关治洪, 刘新芝. 基于捕食搜索策略的遗传算法研究 [J]. 计算机应用研究, 2008, 25(4): 1006-1012.
- [6] 刘毅, 熊盛武. TSP问题的禁忌模拟退火求解 [J]. 计算机工程与应用, 2009, 45(31): 43-45.
- [7] 焦巍, 刘光斌, 张艳红. 求解约束优化的模拟退火 PSO 算法 [J]. 系统工程与电子技术, 2010, 32(7): 1532-1536.
- [8] Garcia-Martinez C, Lozano M, Rodriguez-Diaz F J. A simulated annealing method based on specialized evolutionary algorithm [J]. Applied Soft Computing, 2012, 12(2): 573-588.
- [9] 付文洲, 凌朝东. 布朗运动模拟退火算法 [J]. 计算机学报, 2014, 37(6): 1301-1308.
- [10] Zhi Jianzhuang, Yu Guibo, Deng Shijie, et al. Modeling and simulation about TSP based on simulated annealing algorithm [J]. Applied Mechanics and Materials, 2013, 384(8): 1109-1112.