DOI: 10.14135/j.cnki.1006-3080.20191125003

基于强化学习和角度惩罚距离的冰晶连续优化算法

许 毅[1], 冯 翔[1,2], 虞慧群[1,2]
(1. 华东理工大学信息科学与工程学院, 上海 200237; 2. 上海智慧能源工程技术研究中心, 上海 200237)

摘要: 针对全局连续优化问题, 提出了一种基于强化学习的概率更新和角度惩罚距离偏差策略的冰晶连续优化算法。首先, 通过模拟湖水结冰的自然现象, 提出了冰晶连续优化算法, 实现对连续极值问题的求解; 其次, 在选择湖水中心时, 加入的角度惩罚距离能更好地平衡收敛性和多样性, 消除临时湖水中心带来的能量计算误差; 然后, 基于强化学习的概率更新可以对新生晶体的位置有更好的引导效果, 加快湖水的结冰过程, 更快地逼近湖水中心——全局最优点; 最后, 为了验证概率更新和角度惩罚距离的有效性, 对加入概率更新策略前后的算法进行了比较。将本文算法与其他4种算法在12个基准函数上进行了比较, 验证了算法的有效性。

关键词: 冰晶连续优化算法; 角度惩罚距离; 强化学习; 优化问题
中图分类号: TP18 **文献标志码:** A

全局优化问题广泛存在于工程、经济、生物、网络交通等领域, 目前主要有两大类优化算法用于解决全局优化问题, 一类是基于梯度的算法, 一类是启发式优化算法。然而许多基于梯度的算法, 如拉格朗日乘子法、共轭方向法、投影罚函数法以及黄金分割法等, 主要依赖于目标函数的梯度信息, 而实际问题中的目标函数往往是不可导的, 所以该类算法在解决实际问题时会受到很大的限制。

元启发式算法的优势在于它的灵活性。由于其仅需要通过观察输入和输出来解决优化问题, 而无需考虑搜索空间的梯度, 所以在解决各式各样的问题上具有高度的灵活性, 同时, 在算法中加入的随机因子也有助于避免陷入局部最优。基于这些优势, 元启发式算法可应用于更广泛的领域中。

元启发式算法分为两大类: 群体智能算法[1]和进化算法[2]。群体智能算法的主要基础来源于一些群体生物的集体智慧, 将这些群体行为构建而成的模型作为解决复杂现实问题的框架, 其中最常用的算法是粒子群算法(PSO)[3-4]和蚁群算法(ACO)[5]。此外, 还有其他群体智能算法, 如: 灰狼群算法(GWO)[6-8]、人工蜜蜂群算法(ABC)[9]、樽海鞘群算法(SSA)[10]等。进化算法模拟了自然进化的过程, 最著名的是遗传算法(GA)[11], 其他的进化算法有差分演化算法(DE)[12]、生物地理优化算法(BBO)[13]等。

冬季湖水结冰是一种常见的自然现象。当湖水温度低至凝固点时, 水分子就会凝结成冰晶。通常, 影响湖水温度的因素有湖水的深度、大气的温度、湖面风的大小等。本文以找出湖水中心为目标, 通过模拟湖水结冰的过程实现对连续极值问题的求解, 提出了冰晶连续优化算法来解决全局优化问题。

冰晶连续优化算法在解决极值问题时会极度依赖于每次迭代时中心点的选取, 这样算法容易陷入局部最优。受文献[14]中参考向量的启发, 引入了角度惩罚距离策略来综合考虑中心点位置的选取, 以达到平衡收敛性和分布性的目的。同时在新生冰晶的位置选择上, 为了加速算法的收敛速度, 受文献[15]中组选择策略的影响, 引入了基于强化学习的概率更新策略, 利用已有的先验知识对新生冰晶的位

收稿日期: 2019-11-25
基金项目: 国家自然科学基金(61772200, 61772201, 61602175); 上海市浦江人才计划(17PJ1401900); 上海市经信委信息化发展专项资金(201602008)
作者简介: 许 毅 (1995—): 硕士生, 主要研究方向为演化计算、人工智能。E-mail: 783393048@qq.com
通信联系人: 冯 翔, E-mail: xfeng@ecust.edu.cn

置作出指导,有效地提高了收敛速度。

1 基于强化学习和角度惩罚距离的冰晶连续优化算法

1.1 冰晶连续优化算法

1.1.1 *初始化* 算法开始时,湖水中会分布 S 个维度为 n 的晶体,即 $\boldsymbol{c}_i=(c_{i1},c_{i2},\ldots,c_{in})\in\mathbf{R}^n$,初始化每个晶体的能量值 $E_{i0}=f(x)$,其中 $f(x)$ 为目标函数。

1.1.2 *冰晶壳的生成* 第一次初始化后,能量值最低的 N_S 个晶体将会链接在一起,形成一层冰晶壳,形成冰晶壳的晶体不再加入到沉淀过程中。同时为了保证湖水中未析出晶体个数不变,将会重新生成 N_S 个晶体,以替代形成冰晶壳的晶体。

1.1.3 *湖水晶体的能量变化* 未形成冰晶壳的晶体会发生能量变化,其能量随时间的变化公式为

$$P_i(t)=\lambda_1 Q_i(t)+\lambda_2 S_i(t)+\lambda_3 W_i(t) \tag{1}$$

其中:$Q_i(t)=\dfrac{\alpha\mathbb{P}}{\|L_i-L_{\text{center}}\|}$ 为从湖水中心获取的能量,$\|L_i-L_{\text{center}}\|$ 为晶体到湖水中心的距离。由于湖水中心位置不确定性,所以选择将当前位置最好的晶体定义为临时的中心点 L_{center}。$S_i(t)=\dfrac{\beta\mathbb{S}}{d_{\min}}$ 为晶体从冰晶壳处失去的能量,$\mathbb{S}$ 为一个能量定值,β 为比例因子,$d_{\min}$ 为晶体与冰晶壳间的最短距离。$W_i(t)=\gamma\mathbb{C}$ 为被风带走的能量,γ 为比例因子,$\mathbb{C}$ 为常数。

1.1.4 *冰晶壳的再生长* 未结成冰晶壳的晶体在能量变化后,能量值最低的 N_p 个晶体开始沉淀,沉淀过程中会被已结成冰晶壳的晶体吸收能量,从而加速沉淀并加入到已结成的冰晶壳中,这样扩大了冰晶壳的范围,而未沉淀的晶体将会参与到下次的能量变化和沉淀过程中。

图 1 为冰晶连续优化模型收敛示意图,其伪代码如下:

输入:冰晶群大小 S,最大迭代次数 $T_{\max}$
输出:湖水中心的位置
begin
(1)初始化 S 个冰晶,$T=0$
(2)while $T<T_{\max}$
(3) 根据式(1)更新晶体的能量
(4) 对于每一个晶体
(5) if 晶体能量到达阈值
(6) if 晶体能量为最高
该晶体设为临时湖水中心
(7) 冰晶壳收缩
(8) 出现新的晶体
(9)输出湖水中心的位置
end

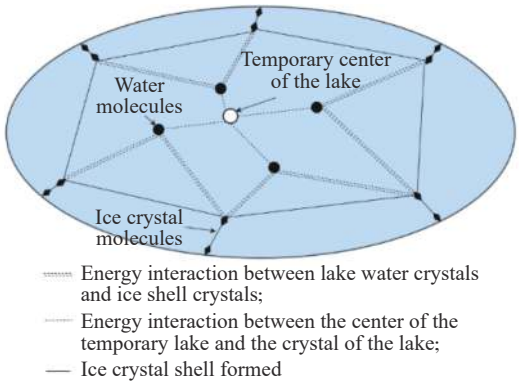

图 1 冰晶连续优化模型收敛示意图

Fig. 1 Convergence diagram of ice crystal continuous optimization model

1.2 基于角度惩罚距离的偏差策略

由于湖水中心位置的不确定性,所以选择当前位置最佳的点作为临时湖水中心,但这势必会导致晶体能量的变化产生偏差,即不能准确地计算出晶体从湖水中心获得的能量值。本文采用角度惩罚距离策略来抵消临时湖水中心带来的偏差值。

1.2.1 *多样性度量* 受临时湖水中心影响,晶体的沉淀过程和结冰结果易陷入局部最优。本文采用角度信息来评价晶体的位置分布,从而消除临时湖水中心带来的收敛程度的影响。

对于晶体 c_i,其多样性度量是指其与未结冰晶体中其他晶体最小夹角,计算公式为

$$\theta(\mathbf{c}_i)=\min_{\mathbf{c}_j\in\mathbf{R},i\neq j}\operatorname{angle}(\mathbf{c}_i,\mathbf{c}_j) \tag{2}$$

$$\operatorname{angle}(\mathbf{c}_i,\mathbf{c}_j)=\arccos\left(\frac{\mathbf{c}_i\cdot\mathbf{c}_j}{\|\mathbf{c}_i\|_2\cdot\|\mathbf{c}_j\|_2}\right) \tag{3}$$

其中:$\cdot$ 表示向量的内积;$\|\|_2$ 表示向量的二阶范式。$\theta(\mathbf{c}_i)$ 的值越大,表示该晶体与其他晶体分隔的越明显,其多样性也越好。

1.2.2 *角度惩罚距离* 在算法前期,应该使晶体的分布更为分散,即让冰晶壳的形成总是从外围向内部扩展,使之与实际湖水中心带来的能量变化的影响相符;而到了算法后期,在结冰范围越来越靠近湖水中心时,减少晶体的多样性,以期在较小的区域中更快地逼近湖水中心。为了消除湖水中心带来的偏差以及满足以上要求,针对 1.1.3 节中晶体的能量变化公式,本文引入了角度惩罚距离策略。随着晶体结冰的进程,动态地调整算法在前后期晶体的分布和抵消临时湖水中心带来的能量偏差。角度惩罚距离公式为

$$\mathrm{APD}(\boldsymbol{c}_i) = (1 + Z(\theta)) \cdot F(x) \tag{4}$$

其中：$F(x) = f(x) + P_x(t)$ 为晶体当前的能量值；$Z(\theta)$ 为角度惩罚因子。

$$Z(\theta) = \left(\frac{t_{\max} - t}{t_{\max}}\right)^{\alpha} \cdot \cos(\theta(x)) \tag{5}$$

式中：$t_{\max}$ 表示预设的最大迭代次数；t 表示当前迭代次数；α 用来控制惩罚因子随迭代次数变化的快慢；$\cos(\theta(x))$ 是对多样性度量 $\theta(x)$ 的一个归一化，并同样保证了度量值其原本的变化规律。

基于角度惩罚距离的偏差策略示意图如图 2 所示。晶体 a、b、c、d、e 分布在湖水的不同位置，其中晶体 c 和 d 的收敛性是一致的，但与 c 最近的晶体为 b，其夹角为 $\angle cb$；与 d 最近的晶体为 a，其夹角为 $\angle da$。由于 $\angle cb < \angle da$，并通过式(4)可以得出 $Z(\angle cb) < Z(\angle da)$，即在下一次迭代中会将以晶体 d 作为临时湖水中心点。所以临时湖水中心点为

$$L_{\text{center}} = \{c_i | \max(\mathrm{APD}(\boldsymbol{c}_i))\} \tag{6}$$

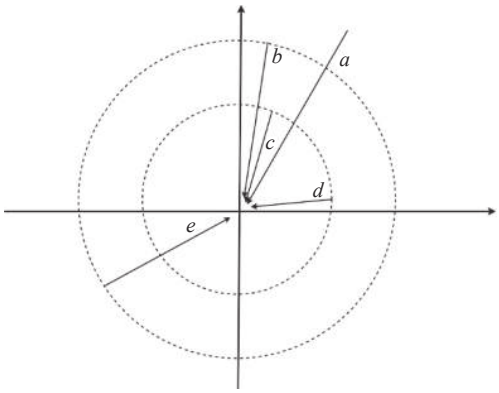

图 2　基于角度惩罚距离的偏差策略示意图

Fig. 2　Deviation strategy based on angle penalty distance

通过角度惩罚距离选择出迭代过程中的临时湖水中心，与原本从目标函数得出的湖水中心相比，角度惩罚距离会考虑到多样性的影响，且是一个随迭代过程动态变化的值，能更好地体现实际湖水中心的位置。

1.3　基于强化学习的概率更新

经过能量交互后，能量较低的晶体将会沉淀并析出，析出的晶体则会链接在一起形成冰晶壳，冰晶壳的结冰范围为

$$c_{j\_\text{down}} = c_{j\_\min} - \left|c_{j\_\max} - c_{j\_\min}\right| \tag{7}$$

$$c_{j\_\text{up}} = c_{j\_\max} + \left|c_{j\_\max} - c_{j\_\min}\right| \tag{8}$$

其中：$c_{j\_\min}$ 和 $c_{j\_\max}$ 分别为析出晶体在第 j 维上的最大值和最小值，则冰晶壳在第 j 维上的结冰范围是 $[c_{j\_\text{down}}, c_{j\_\min}]$ 和 $[c_{j\_\max}, c_{j\_\text{up}}]$，该范围内的晶体都会沉淀并链接在一起形成冰晶壳。

为了保证湖水中的未析出晶体总数不变，在冰晶壳的范围内会生成新的晶体分子，以参与到下一次的迭代沉淀中。在 $[c_{j\_\min}, c_{j\_\max}]$ 范围内随机生成新的晶体，旨在后续迭代中不断形成冰晶壳，从而不断地逼近湖水中心。对于整个湖水能量体系来说，冰晶壳总是位于远离湖水中心的位置，随机生成的晶体极有可能会同样被分布到远离湖水中心的位置，很大程度地降低了湖面结冰的速度。为了让湖面尽快结冰，即加快冰晶算法的寻优速度，本文提出了基于强化学习的概率更新来加快晶体的沉淀过程。

强化学习[16-17]被定义为一个智能体在未知环境中如何采取动作以期获得最大的累计收益，每一步的选择都不仅仅只是影响到当前的收益。强化学习潜在的思想就是那些能使累计收益最大的动作有更大的可能被选择。

晶体的生成依赖于未沉淀晶体的位置。每次沉淀后，会保留有 $S - N_s$ 个晶体继续参与下次迭代，新生的晶体会依概率 G_t 从保留的 $S - N_s$ 个晶体中选择一个，出现在其附近。初始时，保留的晶体被选择的概率是相同的，但随着迭代，上一次保留的晶体可能会在这一次迭代中依旧存在，这时将会奖励这个晶体，并通过式(9)更新概率：

$$p_j^t = \begin{cases} \dfrac{p_k}{N - G_s} + p_j^{t-1}, & j \text{ exsited} \\ 0, & j \text{ precipitated} \\ \dfrac{p_k}{N - G_s}, & j \text{ newborned} \end{cases} \tag{9}$$

其中：p_j^t 为第 t 迭代时第 j 个未沉淀晶体被选择的概率；p_k 和 G_s 分别为上一次迭代中保留而在这次迭代中沉淀的晶体的概率之和晶体数量。

新生晶体依概率选择后，其在第 j 维的值计算公式如下：

$$c_{ij} = \omega c_{pj} + r(1 - \omega) \cdot c_{\text{center}\_j} \tag{10}$$

其中：ω 是权重因子；r 是 $(0,1)$ 中间的一个随机数；c_{pj} 为被选择的晶体在第 j 维的值；c_{center_j} 为临时湖水中心晶体在第 j 维的值。

1.4　算法流程

为了尽量消除由于临时湖水中心的位置引起的能量变化误差及加快湖面结冰的速度，本文分别对临时湖水中心的选择和晶体的生成进行了改进，提出了基于强化学习和角度惩罚距离的冰晶优化算法(APD-CEO)。算法的伪代码如下：

输入：冰晶群大小 S，最大迭代次数 $T_{\max}$

输出：湖水中心的位置

begin

(1)初始化 S 个冰晶, $T=0$

(2)while $T<T_{\max}$

(3) 根据式(2)~式(5)计算角度惩罚距离,得出临时湖水中心

(4) 根据式(1),晶体能量发生变化

(5) 对于每个晶体:

(6) if 晶体能量达到阈值

(7) 晶体开始沉淀,形成冰晶壳

(8) 根据式(9)计算晶体被选择的概率 Gt

(9) 根据式 10 计算出新生晶体的位置

(10)输出湖水中心的位置

end

2 算法收敛性证明

湖水能量体系可以看作是一个能量消耗系统,其整体的能量会随着时间而减少。APD-CEO 算法中的每个晶体都会在空气中散发能量,并且有可能会被冰晶吸收能量,因此构成了一个能量衰减的动态系统。本文通过定义一个正定的 Lyapunov 函数来判定该动态系统的稳定性。

定理 1 Lyapunov 稳定性定理。

如果一个函数 $L(V)$ 满足 $L(V)>0$,为正定的,而 $\frac{\mathrm{d}L(V(t))}{\mathrm{d}t}<0$,为负定的,则函数 $L(V)$ 被看作是一个 Lyapunov 候选项, V 为渐进稳定的。

定理 2 APD-CEO 算法会收敛到一个稳定的状态。

证明 对于 APD-CEO 算法,会存在 3 种状态的晶体,即湖水中心的晶体、处于湖水边沿的晶体和已经析出的晶体。对于已经析出的晶体,其能量稳定,状态不再变化,已变为稳定的状态。

对于处于湖水边沿的晶体,其能量变化公式为 $P_i(t)$,构建 Lyapunov 函数:

$$L(E_i(t))=\sum_{i=0}^{n-1}E_i(t)=\sum_{i=0}^{n-1}(E_i(t)+P_i(t))$$

因为处于湖水边沿的晶体处于要被析出的状态,即 $\sum_{i=0}^{n-1}(P_i(t))<0$。所以 $|L(E_i(t))|\leqslant\sum_{i=0}^{n-1}E_i(t)$,有界。

将式(1)代入可得,

$$L(E_t(t))=\sum_{i=0}^{n-1}(E_i(t)+\lambda_1Q_i(t)+\lambda_2S_i(t)+\lambda_3W_i(t))$$

代入相应的计算公式,有

$$L(E_t(t))=\sum_{i=0}^{n-1}\left(E_i(t)+\lambda_1\frac{\alpha P}{\|L_i-L_{\text{center}}\|}+\lambda_2S_i(t)+\lambda_3W_t(t)\right)$$

$$\frac{\partial L(E_t(t))}{\partial t}=\sum_{i=0}^{n-1}\left(\lambda_1\frac{\alpha}{\|L_i-L_{\text{center}}\|}\times\frac{\partial E_{\text{center}}(t)}{\partial t}+\lambda_2\times\frac{\partial S_i(t)}{\partial t}+\lambda_3\gamma C\right)$$

在晶体析出的过程中,主要受到沉淀晶体对其能量吸收的影响,而湖水中心晶体对其的影响可忽略不计,即令 $\lambda_1=0$。则简化可得

$$\frac{\partial L(E_t(t))}{\partial t}=\sum_{i=0}^{n-1}(\lambda_2\times\frac{\partial S_i(t)}{\partial t}+\lambda_3\gamma C)$$

其中: $\lambda_2<0, S_i(t)>0; \lambda_3<0, C$ 为一大于 0 的常量,因此,有 $\frac{\partial L(E_t(t))}{\partial t}\leqslant 0$,即湖水边沿的能量 $L(E_t(t))$ 随时间单调减少。

湖水中心晶体通过式(4)来判断,湖水中心 L_{center} 由角度惩罚距离和初始能量共同决定。当算法步入迭代后期,式(5)中 $Z(\theta)$ 的值趋近于 0,此时的湖水中心总是选取当前能量最大的湖水晶体,即湖水中心分子总是选取每次迭代中的最大值。而湖水体系是一个能量衰减的系统,所以湖水中心分子最终会收敛到一个稳定的值。

由上可知,基于 Lyapunov 稳定性定理,APD-CEO 算法将会收敛到一个平衡的状态。

3 实验与分析

实验环境: Matlab R2014b on the HPCC of Lenovo Shenteng 6800, 该集群拥有 8 个计算节点和 1 个总控节点。每一个计算节点是一台高性能服务器,内存为 24 GB, 四核 2.4 GHz 中央处理器。

3.1 算法的有效性比较

12 个基准函数的描述如表 1 所示。将 APD-CEO 算法与 RGA[18]、GSO[19]、SSO[20] 和 LAS[21] 算法进行对比来验证算法的性能。在所有的比较示例中,群体规模均为 100, 最大迭代次数均为 1000。每个算法的参数设置如表 2 所示。

表 3 示出了本文算法与其他 4 种算法在函数维度 D 为 30 维的比较结果,实验运行次数为 30 次。性能评价指标为平均最佳解 (AB)、最佳解的标准差 (SD)。从表中可以看出本文算法只有在 f_5 函数、f_6 函数和 f_9 函数上的 AB 指标不理想;在 f_2 和 f_3 函数上,算法性能排第二;在其余的 7 个函数中都能取得最优的 AB 和 SD 指标。表 4 示出了算法在 50 维基

表 1　基准测试函数表

Table 1　Benchmark function table

Benchmark	Function	D	Range	Minimum
Sphere	$f_1(x)=\sum_{i=1}^{D}x_i^2$	30,50,100	[−100, 100]	0
SumSquares	$f_2(x)=\sum_{i=1}^{D}ix_i^2$	30,50,100	[−10, 10]	0
Rosenbrock	$f_3(x)=\sum_{i=1}^{D-1}(100(x_i^2-x_{i+1})^2+(1-x_i)^2)$	30,50,100	[−30, 30]	0
Schwefel 2.22	$f_4(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	30,50,100	[−10,10]	0
Rastrigin	$f_5(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$	30,50,100	[−10,10]	0
Schwefel	$f_6(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$	30,50,100	[−500, 500]	$-418.9829\times D$
Ackley	$f_7(x)=20+\exp-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))$	30,50,100	[−32, 32]	0
Griewank	$f_8(x)=\frac{1}{4\,000}(\sum_{i=1}^{D}x_i^2)-(\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}))+1$	30,50,100	[−600, 600]	0
F4	$f_9=418.9829n+\sum_{i=1}^{D}-x_i\sin(\sqrt{\lvert x_i\rvert})$	30,50,100	[−100, 100]	0
Step	$f_{10}=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$	30,50,100	[−100, 100]	0
Zakharov	$f_{11}(x)=\sum_{i=1}^{n}x_i^2+(\sum_{i=1}^{n}0.5ix_i)^2+(\sum_{i=1}^{n}0.5ix_i)^4$	30,50,100	[−5, 10]	0
Salomon	$f_{12}=-\cos(2\pi\sqrt{\sum_{i=1}^{D}x_i^2})+0.1\sqrt{\sum_{i=1}^{D}x_i^2+1}$	30,50,100	[−100, 100]	0

表 2　各个算法的参数设置

Table 2　Parameter settings of each algorithms

Algorithm	Parameter setting
RGA	The initial number of root tips is 1, and the number of branches is 3, the parameter α is a random number ranges from 0 to 1 and β is a random number ranges from 5 to 10.
GSO	The initial angle is $\pi/4$, the parameter a is a random number ranges from 0 to $\sqrt{n+1}$ and the maximum angle of pursuit is π/a^2.
SSO	The threshold parameter PF is 0.7.
LSA	The channel time is set as 10.
APD-CEO	The number of precipitated molecules is 80, parameter α,β and γ are random numbers range from 0 to 1. At the stage of energy change, the parameter λ_1 =100 and λ_2 =0. At the regrowth stage, the parameter λ_1 =0, λ_2 =-100 and λ_3 =-50.

准函数下的实验结果, 与 30 维基准函数相比, 本文算法表现得更好, 除了上述的 7 个函数外, 还在 f_2 和 f_3 函数上取得了最好的结果。

为了测试本文算法在高维度上的性能表现, 提升基准函数的维度到 100 维, 实验结果如表 5 所示。与 50 维的结果类似, 本文算法在 9 个基准函数上取得了最好的效果, 同时在 f_8 和 f_{10} 函数中依旧取得了标准最小值 0。

图 3 为不同优化算法在 30 维的基准函数上的搜索演化曲线图, 可以更直观地看出算法的优劣性。基准函数分别为 f_1、f_3、f_4、f_7、f_8、f_{10}、f_{11}、f_{12}, 从图中可以看出, APD-CEO 的收敛速度最快。

为了更进一步分析算法的有效性, 以 Friedman 排名来进一步地说明, 表 6 列出了不同维度下基准函数上的 Friedman 平均排名。可以看出 5 种算法的优

表3 APD-CEO、RGA、GSO、SSO和LSA在基准函数上的实验结果 (D=30)

Table 3 Experimental results of APD-CEO, RGA, GSO, SSO and LSA on benchmark functions (D=30)

Function	Index	APD-CEO	RGA	GSO	SSO	LSA
f_1	AB	**$8.537\,0\times10^{-99}$**	$5.774\,4\times10^{-8}$	$4.468\,5\times10^{-5}$	$1.965\,3\times10^{-3}$	$3.213\,8\times10^{-30}$
	SD	**$2.103\,9\times10^{-98}$**	$6.487\,1\times10^{-9}$	$4.260\,9\times10^{-2}$	$9.960\,1\times10^{-4}$	$4.628\,5\times10^{-31}$
f_2	AB	$6.140\,2\times10^{-100}$	$3.709\,5\times10^{-5}$	**0**	$4.446\,5\times10^{-4}$	$2.471\,5\times10^{-20}$
	SD	$1.020\,3\times10^{-99}$	$1.942\,6\times10^{-6}$	**0**	$2.901\,9\times10^{-4}$	$5.815\,3\times10^{-21}$
f_3	AB	$1.898\,8\times10$	$2.307\,2\times10$	$8.004\,9\times10$	$1.148\,3\times10^{2}$	**1.164 8**
	SD	**$1.387\,1\times10^{-2}$**	1.339 6	$2.428\,8\times10$	$3.942\,8\times10$	$2.381\,8\times10^{-1}$
f_4	AB	**$2.261\,8\times10^{-78}$**	$5.417\,7\times10^{-3}$	$5.608\,7\times10^{-1}$	$1.371\,0\times10^{-2}$	$4.395\,1\times10^{-20}$
	SD	**$1.590\,6\times10^{-78}$**	$1.775\,3\times10^{-4}$	$7.340\,9\times10^{-2}$	$3.117\,9\times10^{-3}$	$3.781\,9\times10^{-21}$
f_5	AB	$1.117\,7\times10^{2}$	$1.694\,2\times10$	$2.113\,6\times10$	**8.594 8**	$2.167\,8\times10$
	SD	$9.919\,2\times10$	8.539 8	3.233 8	**1.114 3**	4.032 6
f_6	AB	$-3.099\,1\times10^{3}$	$-6.942\,3\times10^{3}$	-7.224×10^{2}	$-9.364\,3\times10^{2}$	**$-9.211\,5\times10^{3}$**
	SD	$4.732\,0\times10^{2}$	$8.684\,9\times10^{2}$	**1.343 5×10**	$1.614\,9\times10$	$4.085\,6\times10^{2}$
f_7	AB	**$4.440\,2\times10^{-16}$**	$1.864\,9\times10^{-8}$	$8.660\,6\times10^{-4}$	$1.360\,5\times10^{-2}$	$2.614\,8\times10^{-10}$
	SD	**$4.440\,8\times10^{-16}$**	$2.200\,8\times10^{-9}$	$3.669\,0\times10^{-5}$	$2.361\,2\times10^{-3}$	$4.258\,4\times10^{-11}$
f_8	AB	**0**	$7.394\,7\times10^{-6}$	$6.422\,9\times10^{-1}$	$3.294\,0\times10^{-3}$	$2.806\,5\times10^{-16}$
	SD	**0**	$1.493\,2\times10^{-7}$	$7.920\,5\times10^{-2}$	$5.490\,1\times10^{-4}$	$1.398\,6\times10^{-17}$
f_9	AB	$1.207\,5\times10^{4}$	$7.112\,9\times10^{2}$	$1.184\,7\times10^{4}$	$6.789\,3\times10^{3}$	**$1.892\,5\times10^{2}$**
	SD	$1.429\,0\times10^{2}$	$1.259\,8\times10^{2}$	**1.535 8×10**	$1.013\,2\times10^{3}$	$3.948\,6\times10$
f_{10}	AB	**0**	$5.263\,7\times10^{-42}$	**0**	$2.689\,2\times10^{-3}$	$4.754\,5\times10^{-16}$
	SD	**0**	$2.253\,2\times10^{-43}$	**0**	$6.059\,8\times10^{-4}$	$9.843\,5\times10^{-17}$
f_{11}	AB	**$5.188\,2\times10^{-98}$**	$6.542\,3\times10^{-5}$	$2.322\,9\times10^{-6}$	$6.810\,3\times10$	$8.549\,8\times10^{-7}$
	SD	**$1.275\,4\times10^{-97}$**	$2.168\,3\times10^{-5}$	$1.532\,5\times10^{-6}$	$3.008\,4\times10$	$7.924\,8\times10^{-8}$
f_{12}	AB	**$1.873\,6\times10^{1}$**	$3.326\,3\times10^{-1}$	$5.800\,2\times10^{-1}$	$2.740\,9\times10^{-1}$	$3.405\,5\times10^{-1}$
	SD	**$1.490\,9\times10^{-2}$**	$1.965\,3\times10^{-2}$	$6.310\,2\times10^{-2}$	$5.177\,5\times10^{-2}$	$4.540\,5\times10^{-2}$

表 4　APD-CEO、RGA、GSO、SSO 和 LSA 在基准函数上的实验结果 (D=50)

Table 4　Experimental results of APD-CEO, RGA, GSO, SSO and LSA on benchmark functions (D=50)

Function	Index	APD-CEO	RGA	GSO	SSO	LSA
f_1	AB	**$9.789\,0\times10^{-99}$**	$2.907\,4\times10^{-6}$	$1.451\,3\times10^{-4}$	$1.642\,4\times10^{-1}$	$5.671\,5\times10^{-16}$
	SD	**$2.274\,5\times10^{-98}$**	$4.790\,8\times10^{-7}$	$8.946\,4\times10^{-5}$	$3.218\,4\times10^{-2}$	$2.114\,8\times10^{-17}$
f_2	AB	**$9.781\,3\times10^{-99}$**	$3.213\,2\times10^{-3}$	$3.801\,3\times10^{-3}$	4.768 4	$8.484\,3\times10^{-9}$
	SD	**$1.964\,0\times10^{-98}$**	$2.998\,6\times10^{-3}$	$1.167\,8\times10^{-2}$	1.445 6	$3.481\,6\times10^{-10}$
f_3	AB	**4.898 5×10**	$1.945\,2\times10^{2}$	$1.723\,7\times10^{2}$	7.349 5×10	5.538 4×10
	SD	**$1.836\,3\times10^{-2}$**	1.567 8×10	1.057 4×10	2.011 9×10	2.859 9×10
f_4	AB	**$1.038\,5\times10^{-77}$**	$1.515\,3\times10^{-2}$	$2.357\,4\times10^{-1}$	2.034 6	$3.915\,8\times10^{-11}$
	SD	**$7.593\,2\times10^{-78}$**	$1.205\,4\times10^{-3}$	$6.328\,4\times10^{-2}$	$1.722\,4\times10^{-1}$	$2.191\,5\times10^{-12}$
f_5	AB	9.054 4×10	**5.224 6×10**	$1.008\,5\times10^{2}$	7.945 6×10	5.804 5×10
	SD	9.782 3×10	6.951 3	1.531 5×10	**1.435 9**	4.439 5
f_6	AB	$-4.130\,6\times10^{3}$	$-9.596\,1\times10^{3}$	$-1.204\,1\times10^{3}$	**$-1.456\,2\times10^{4}$**	$-1.195\,3\times10^{4}$
	SD	$7.035\,4\times10^{2}$	$4.265\,4\times10^{3}$	**$8.132\,4\times10^{-6}$**	$1.624\,0\times10^{3}$	$2.178\,5\times10^{3}$
f_7	AB	**$4.235\,2\times10^{-14}$**	$2.137\,9\times10^{-4}$	$5.187\,4\times10^{-2}$	$3.384\,6\times10^{-1}$	$1.486\,2\times10^{-5}$
	SD	**$1.846\,3\times10^{-15}$**	$6.644\,5\times10^{-5}$	$7.165\,8\times10^{-3}$	$3.023\,4\times10^{-2}$	$2.018\,9\times10^{-6}$
f_8	AB	**0**	$8.214\,4\times10^{-5}$	6.226 1	$2.488\,3\times10^{-2}$	$7.415\,8\times10^{-8}$
	SD	**0**	$2.237\,8\times10^{-6}$	$4.502\,9\times10^{-2}$	$7.085\,4\times10^{-3}$	$2.784\,6\times10^{-9}$
f_9	AB	$2.029\,6\times10^{4}$	$1.895\,6\times10^{4}$	$1.974\,5\times10^{4}$	$6.794\,5\times10^{3}$	**$1.075\,4\times10^{3}$**
	SD	**$1.562\,9\times10^{2}$**	$2.349\,5\times10^{3}$	$3.001\,8\times10^{2}$	$1.332\,4\times10^{3}$	$2.008\,1\times10^{2}$
f_{10}	AB	**0**	$3.387\,9\times10^{-26}$	$1.764\,1\times10^{-7}$	**0**	$4.675\,1\times10^{-13}$
	SD	**0**	$3.246\,4\times10^{-27}$	$5.161\,5\times10^{-8}$	0	$1.456\,8\times10^{-14}$
f_{11}	AB	**$5.149\,1\times10^{-98}$**	$3.913\,4\times10^{-3}$	$1.387\,6\times10^{-2}$	6.214 7	$9.198\,7\times10^{-4}$
	SD	**$1.950\,8\times10^{-97}$**	$3.240\,0\times10^{-4}$	$1.135\,4\times10^{-3}$	3.684 5	$4.528\,3\times10^{-4}$
f_{12}	AB	**$1.772\,1\times10^{-1}$**	$6.791\,3\times10^{-1}$	$9.299\,9\times10^{-1}$	$4.802\,5\times10^{-1}$	$5.219\,8\times10^{-1}$
	SD	$4.023\,2\times10^{-2}$	$4.362\,5\times10^{-2}$	$3.154\,6\times10^{-2}$	**$4.004\,6\times10^{-2}$**	$4.159\,8\times10^{-2}$

表5 APD-CEO、RGA、GSO、SSO和LSA在基准函数上的实验结果(D=100)

Table 5 Experimental results of APD-CEO, RGA, GSO, SSO and LSA on benchmark functions (D=100)

Function	Index	APD-CEO	RGA	GSO	SSO	LSA
f_1	AB	**$3.474\,3\times10^{-98}$**	$1.974\,2\times10^{-1}$	$3.093\,2\times10^{-1}$	2.244 2	$5.238\,4\times10^{-6}$
	SD	**$1.058\,6\times10^{-97}$**	$2.463\,2\times10^{-2}$	$3.161\,8\times10^{-2}$	2.943 2	$4.187\,8\times10^{-6}$
f_2	AB	**$2.412\,1\times10^{-99}$**	5.427 9	4.606 4×10	1.932 4×10	$2.018\,6\times10^{-6}$
	SD	**$7.604\,7\times10^{-99}$**	2.379 2	1.851 7	1.103 7	$1.268\,7\times10^{-7}$
f_3	AB	**9.898 7×10**	$3.715\,3\times10^{2}$	$7.898\,4\times10^{2}$	$3.352\,3\times10^{2}$	$1.608\,9\times10^{2}$
	SD	**$1.105\,7\times10^{-2}$**	4.174 2×10	3.681 6	$1.574\,8\times10^{2}$	2.168 6×10
f_4	AB	**$3.924\,3\times10^{-77}$**	4.224 3	5.610 5	8.843 2	$2.678\,2\times10^{-4}$
	SD	**$2.343\,3\times10^{-77}$**	$2.197\,8\times10^{-1}$	3.167 5	$7.231\,5\times10^{-1}$	$1.925\,8\times10^{-5}$
f_5	AB	$1.233\,6\times10^{2}$	$1.398\,8\times10^{2}$	$4.231\,2\times10^{2}$	$2.385\,7\times10^{2}$	**9.108 6×10**
	SD	$1.014\,4\times10^{2}$	1.747 9×10	3.123 1×10	8.544 2×10	**3.371 7**
f_6	AB	$-5.779\,8\times10^{3}$	$-1.668\,3\times10^{4}$	$-2.407\,9\times10^{3}$	$-2.895\,3\times10^{2}$	**$-2.256\,7\times10^{4}$**
	SD	$8.555\,7\times10^{2}$	$4.324\,6\times10^{3}$	**3.131 8**	$4.084\,6\times10^{3}$	$1.861\,7\times10^{3}$
f_7	AB	**$7.440\,8\times10^{-13}$**	2.479 8	4.920 5	1.571 2	$2.946\,8\times10^{-1}$
	SD	**$1.845\,3\times10^{-14}$**	$4.254\,2\times10^{-1}$	$8.921\,6\times10^{-1}$	$8.271\,2\times10^{-1}$	$2.061\,8\times10^{-2}$
f_8	AB	**0**	$1.717\,6\times10^{-3}$	$1.236\,4\times10^{2}$	$2.184\,7\times10^{-2}$	$4.821\,6\times10^{-3}$
	SD	**0**	$2.839\,7\times10^{-4}$	$0.598\,4\times10^{-2}$	$4.290\,3\times10^{-3}$	$1.815\,8\times10^{-4}$
f_9	AB	$4.098\,8\times10^{4}$	$8.214\,3\times10^{4}$	$3.949\,1\times10^{4}$	$1.618\,2\times10^{4}$	$9.149\,5\times10^{3}$
	SD	$1.787\,3\times10^{2}$	$4.512\,3\times10^{3}$	**1.468 4**	$1.347\,8\times10^{3}$	$1.128\,4\times10^{3}$
f_{10}	AB	**0**	5.263 4×10	1.715 6×10	2.108 3	1.042 6
	SD	**0**	2.637 4×10	$3.187\,9\times10^{-3}$	1.589 3	2.361 8
f_{11}	AB	**$3.292\,8\times10^{-99}$**	8.324 5×10	3.384 8×10	6.199 5×10	5.124 8
	SD	**$1.119\,7\times10^{-98}$**	4.386 6	6.165 7	7.104 7×10	$1.801\,6\times10^{-4}$
f_{12}	AB	**$1.708\,5\times10^{-1}$**	$9.180\,6\times10^{-1}$	2.299 9	1.081 0	$8.317\,6\times10^{-1}$
	SD	$4.498\,7\times10^{-2}$	$2.479\,2\times10^{-2}$	$1.264\,8\times10^{-1}$	$1.479\,3\times10^{-1}$	**$1.286\,4\times10^{-2}$**

化性能表现排名为APD-CEO、LSA、RGA、GSO和SSO。图4示出了不同算法100维下在12个基准函数上的排名,可以看出,除了函数f_5、f_6和f_9之外,APD-CEO是5种算法中表现最好的。

3.2 角度惩罚距离的有效性

为了验证角度惩罚距离在冰晶连续优化算法中的有效性,将加入角度惩罚距离策略后的冰晶连续优化算法与未加入任何策略的冰晶连续优化算法进

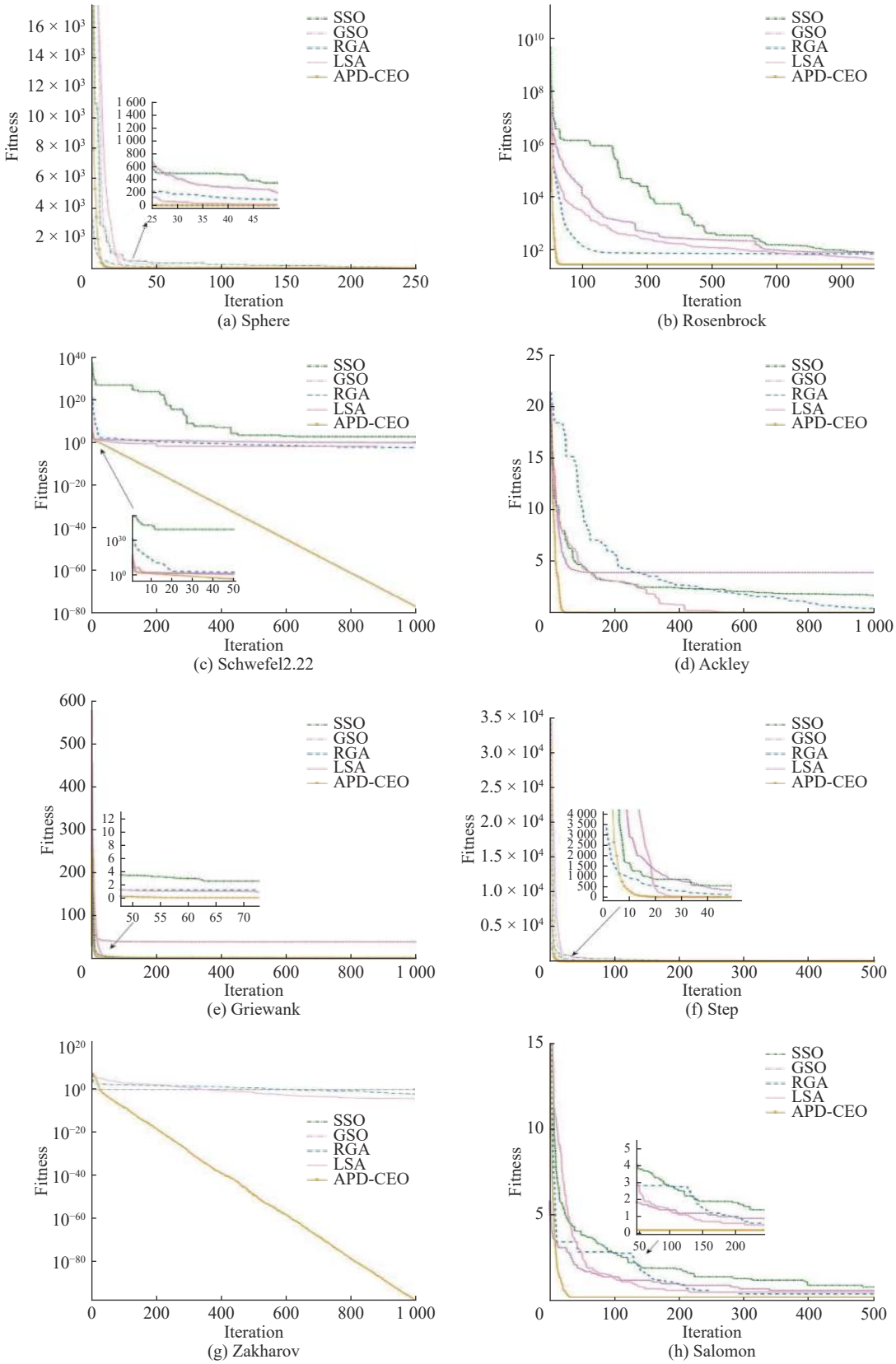

图 3　5 种算法在 8 个 30 维基准函数上的演化曲线

Fig. 3　Evolution curves of the five algorithms on eight benchmark functions with 30 dimensions as examples

行了性能比较。实验结果如表 7 所示。

3.3　基于强化学习的概率更新策略的有效性

为了研究基于强化学习的概率更新策略对算法的影响, 本文对加入基于强化学习的概率更新策略前后的算法进行了比较, 实验结果如图 5 所示。After 表示加入概率更新后的优化曲线, Before 表示

表 6 各优化算法在 30、50 和 100 维下的平均排名

Table 6 Average ranking of each optimization algorithms

D	APD-CEO	RGA	GSO	SSO	LSA
30	**2.125**	2.92	3.75	3.71	2.25
50	**1.875**	3.17	4.42	3.46	2.08
100	**1.50**	3.50	4.42	3.75	1.83

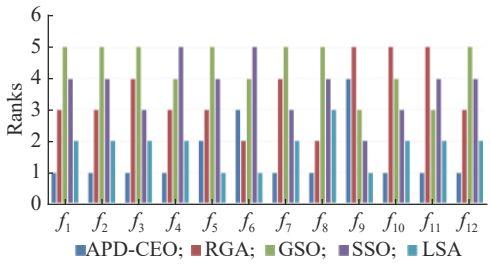

图 4 各算法的 Friedman 排名

Fig. 4 Friedman ranks of each optimization algorithms

表 7 角度惩罚距离策略的有效性

Table 7 Effectiveness of angle penalty distance strategy

Function	Index	Before	After
Sphere	AB	$9.435\,3\times10^{-29}$	$5.353\,5\times10^{-83}$
	SD	$1.469\,7\times10^{-44}$	$1.666\,1\times10^{-84}$
SumSquares	AB	$1.736\,2\times10^{-28}$	$5.840\,1\times10^{-82}$
	SD	$1.617\,9\times10^{-28}$	$1.639\,2\times10^{-81}$
Rastrigin	AB	$7.194\,3\times10^{2}$	$7.117\,7\times10^{2}$
	SD	4.239 53×10	1.919 2×10
Ackley	AB	$1.234\,9\times10^{-5}$	$7.372\,3\times10^{-14}$
	SD	$5.047\,8\times10^{-9}$	$9.832\,7\times10^{-15}$

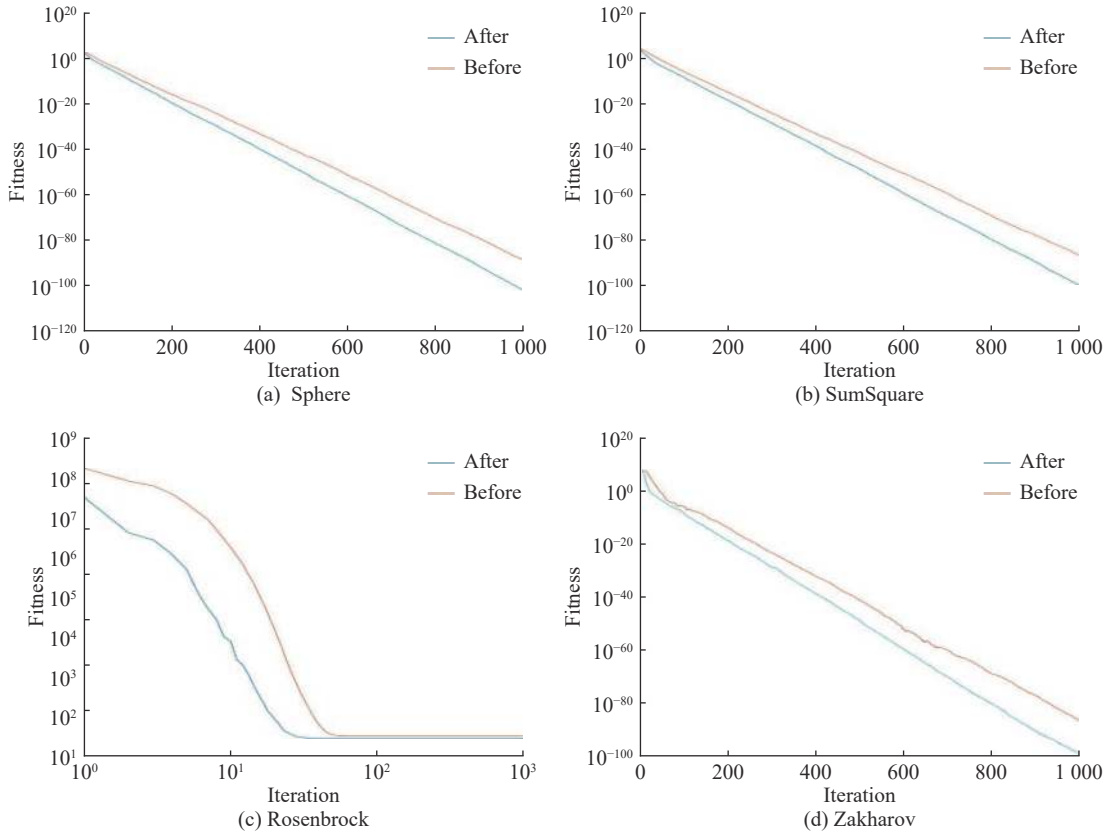

图 5 加入概率更新策略前后算法的演化曲线

Fig. 5 Evolution curves of the algorithm before and after joining probabilistic update atrategy

加入之前的优化曲线。可以看出，加入概率更新后，曲线下降得更快，优化算法能更快地收敛，并且保持在 1 000 次迭代后拥有更好的结果。实验表明基于强化学习的概率更新能给算法带来优势。

4 结束语

通过模拟湖水结冰的过程来实现对连续极值问题的求解，提出了冰晶连续优化算法以解决全局优化问题，并在其基础上加入了角度惩罚距离策略和基于强化学习的概率更新策略，使得中心点的选择能同时考虑到收敛性和分布性，以及晶体的生成能借鉴已有的先验知识，加速了算法的收敛速度和准确度。将本文算法与其他 4 种对比算法在 12 个基准函数上进行测试，并分别在 30 维、50 维和 100 维上检验了算法性能。实验结果表明，与其他 4 种算法相

比, APD-CEO 能表现出良好的性能, 且在高维度上能表现得更好。为了更进一步测试算法性能, 使用 Friedman Test 非参数统计方法分析了本文算法的表现。本文主要探讨的是算法在单目标极值优化问题上的性能表现, 在接下来的工作中, 我们将更多地研究算法在多目标优化问题中的表现。

参考文献:

[1] PHAN H D, ELLIS K, BARCA J C, *et al*. A survey of dynamic parameter setting methods for nature-inspired swarm intelligence algorithms[J]. Neural Computing and Applications, 2019, 8: 1-22.

[2] BÄCK T.Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms[M]. UK: Oxford University Press, 1996.

[3] M R reza BONYADI, Z MICHALEWICZ. Particle swarm optimization for single objective continuous space problems: A review[J]. Evolutionary Computation, 2017, 25(1): 1-54.

[4] 赖兆林, 冯翔, 虞慧群. 基于逆向学习行为粒子群算法的云计算大规模任务调度[J]. 华东理工大学学报(自然科学版), 2020, 46(2).

[5] MAVROVOUNIOTIS M, YANG S, YAO X. Multi-colony ant algorithms for the dynamic travelling salesman problem[C]//2014 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE). Orlando, FL: IEEE, 2014: 9-16.

[6] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.

[7] KUMAR V, CHHABRA J K, KUMAR D. Grey wolf algorithm-based clustering technique[J]. Journal of Intelligent Systems, 2017, 26(1): 153-168.

[8] LONG W, CAI S H, JIAO J J. Hybrid grey wolf optimization algorithm for high-dimensional optimization[J]. Control and Decision, 2016, 31(11): 1991-1997.

[9] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 9(3): 459-471.

[10] SEYEDALI M, AMIR H G, SEYEDEH Z M, *et al*. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114: 163-191.

[11] GOLDBERG D E, HOLLAND J H. Genetic algorithms and machine learning[J]. Machine Learning, 1988, 3: 95-99.

[12] BREST J, ZAMUDA A, BOSKOVIC B, *et al*. Dynamic optimization using self-adaptive differential evolution[C]//2009 IEEE Congress on Evolutionary Computation. Norway: IEEE, 2009: 415-422.

[13] SIMON D. Biogeography-based optimization[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(6): 702-713.

[14] CHENG R, JIN Y, OLHOFER M, *et al*. A reference vector guided evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5): 773-791.

[15] ZHOU Y, HAO J K, DUVAL B. Reinforcement learning based local search for grouping problems: A case study on graph coloring[J]. Expert Systems with Applications, 2016, 64: 412-422.

[16] WATKINS C J C H, DAYAN P. Q -learning[J]. Machine Learning, 1992, 8(3/4): 279-292.

[17] LEIBO J Z, ZAMBALDI V, LANCTOT M, *et al*. Multi-agent reinforcement learning in sequential social dilemmas[C]//16th Conference on Autonomous Agents and Multiagent Systems. Sao Paulo: [s.n.], 2017: 464-473.

[18] ZHANG H, ZHU Y L, CHEN H N. Root growth model: A novel approach to numerical function optimization and simulation of plant root system[J]. Soft Computing, 2014, 18(3): 521-537.

[19] HE S, WU Q H, SAUNDERS J R. Group search optimizer: An optimization algorithm inspired by animal searching behavior[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 973-990.

[20] CUEVAS E, CIENFUEGOS M, DANIEL Z, *et al*. A swarm optimization algorithm inspired in the behavior of the social-spider[J]. Expert Systems with Applications, 2013, 40(16): 6374-6384.

[21] SHAREEF H, IBRAHIM A A, MUTLAG A H. Lightning search algorithm[J]. Applied Soft Computing, 2015, 36: 315-333.

Ice Crystal Continuous Optimization Algorithm Based on Reinforcement Learning and Angle Penalty Distance

XU Yi[1], FENG Xiang[1,2], YU Huiqun[1,2]

(1. School of Information Science and Engineering, East China University of Science and Technology, Shanghai

200237, China; 2. Shanghai Engineering Research Center of Smart Energy, Shanghai 200237, China）

Abstract: The global optimization problem has been widely used in various fields, but the traditional method relies on the gradient information of the objective function too much. The meta heuristic search algorithms have better flexibility and it can be used to solve practical problems. Hence, a ice crystal continuous optimization algorithm based on reinforcement learning and angle penalty distance (APD-CEO), which introduces probabilistic update strategy based on reinforcement learning and deviation strategy based on angle penalty distance, is proposed for the global continuous optimization problem. Firstly, ice crystal continuous optimization algorithm is proposed to solve the continuous extremum problem by simulating the freezing process of lake water. Secondly, in order to eliminate the error in calculating the energy from the temporary center of the lake, the Angle penalty distance strategy is introduced, and the convergence and diversity are better balanced. Meanwhile, probabilistic update strategy based on reinforcement learning can better guide the position of the newly formed crystals, accelerate the freezing process of the lake, and approach the center of lake faster (the global optimum). Finally, in order to verify the validity of probabilistic update strategy and angle penalty distance strategy, the algorithm before and after joining the strategy are compared. Moreover, APD-CEO has better performance than other algorithms in most benchmark functions, and the contrast effect is more obvious in the high dimensions. And Friedman test also shows that the APD-CEO ranks better among the five algorithms.

Key words: continuous ice crystal optimization algorithm; angle penalty distance; reinforcement learning; optimization problem