Computer Science and Application 计算机科学与应用, 2022, 12(11), 2619-2632
Published Online November 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.1211266

Hans汉斯

基于折射反向学习的人工蜂群算法

葛孟珂[1]，王　冰[1*]，王　剑[1]，何　浩[2]

[1]牡丹江师范学院，黑龙江 牡丹江
[2]哈尔滨商业大学，黑龙江 哈尔滨

收稿日期：2022年10月25日；录用日期：2022年11月23日；发布日期：2022年11月30日

摘　要

人工蜂群算法(ABC)作为一种简单而有效的计算技术，被广泛应用于解决工程问题，为克服算法的局部搜索能力弱，收敛速度慢的缺点，提出了基于折射反向学习的人工蜂群算法(CRABC)。在引领蜂阶段，根据当前最优解指导产生下一次迭代的候选解；在引领蜂阶段后，计算当前种群的折射反向解，根据适应度值对种群择优选择组成下次迭代的候选解，极大提高了算法的收敛速度。为验证所提CRABC算法的性能，采用了12个基准测试函数进行测试，将CRABC算法与ABC算法，仅加入差分变异的ABC算法(CABC)，仅加入折射反向学习的ABC算法(RABC)进行比较，来验证综合两种策略后的CRABC算法改进效果；同时将CRABC算法，RABC算法与GABC算法，ABC/best/2算法在5个基准测试函数上进行比较。实验结果表明：所提的CRABC算法可以提高ABC算法的开发和探索能力。

关键词

人工蜂群算法，差分变异，折射反向学习

Artificial Bee Colony Algorithm Based on Refracted Opposition-Based Learning

Mengke Ge[1], BingWang[1*], JianWang[1], Hao He[2]

[1]Mudanjiang Normal University, Mudanjiang Heilongjiang
[2]Harbin University of Commerce, Harbin Heilongjiang

Received: Oct. 25th, 2022; accepted: Nov. 23rd, 2022; published: Nov. 30th, 2022

Abstract

Artificial bee colony algorithm (ABC) as a simple and effective computing technology, is widely used to solve engineering problems, in order to overcome the drawbacks of the algorithm's weak local search ability, slow convergence speed, Artificial bee colony algorithm based on refracted

*通讯作者。

文章引用：葛孟珂，王冰，王剑，何浩. 基于折射反向学习的人工蜂群算法[J]. 计算机科学与应用，2022，12(11): 2619-2632. DOI: 10.12677/csa.2022.1211266

opposition-based learning (CRABC) was proposed. In the employed bee phase, the candidate solution of the next iteration is generated according to the current optimal solution. After the employed bee phase, calculate the refracted opposite solution of the current population, according to the fitness value, excellent individuals are selected to form the candidate solution of the next iteration, which greatly improves the convergence speed of the algorithm. To verify the optimization performance of the proposed algorithm, 12 benchmark functions were utilized to investigate the algorithm, the CRABC algorithm was compared with ABC algorithm, ABC algorithm with only differential variation (CABC), and ABC algorithm (RABC) with only refracted opposition-based learning to verify the improvement effect of the CRABC algorithm after the two strategies. Meanwhile, CRABC algorithm, RABC algorithm, GABC algorithm and ABC/best/2 algorithm are compared on five benchmark functions. The experimental results show that the proposed CRABC algorithm can improve the ability to develop and explore of the ABC algorithm.

Keywords

Artificial Bee Colonyalgorithm (ABC), Differential Variation, Refracted Opposition-Based Learning

1. 引言

大数据时代的来临，使得工程系统中的计算模型越来越复杂，如何获得更高计算精度成为一个亟待解决的问题，优化算法则是解决的重要工具。经典优化算法的计算过程复杂，且对目标函数的性质有严格要求，并不适用于解决高维复杂的优化问题。智能优化算法因其操作简单，鲁棒性强，对解要求不高，且仅以目标函数为适应度值的特点，而比经典优化算法应用范围更广。

2005 年土耳其学者 Karaboga [1]受蜂群个体之间分工和交流完成采蜜行为的启发提出了人工蜂群算法(Artificial Bee Colony, ABC)。与经典的优化算法相比，ABC 算法对目标函数和约束几乎没有要求，操作简单，控制参数少，同时拥有较好的灵活性和鲁棒性，在解决一些优化问题上有得天独厚的优势。然而 ABC 算法存在局部搜索能力弱，收敛速度慢的缺点[2]，因此，学者们提出了各种不同方式的改进算法。反向学习(Opposition-Based Learning, OBL)策略是其中的一个改进手段，Tizhooshs 和 Pahnamayan 等[3] [4] [5]已将反向学习用于遗传算法(Genetic Algorithm, GA)，差分进化算法(Differential Evolution, DE)和粒子群算法(Particle Swarm Ootimization, PSO)等多种智能优化算法，并取得较好结果。王等在 ABC 算法的跟随蜂阶段采用依概率反向学习策略更新种群[6]。

为进一步提高 ABC 算法的性能，本文提出了基于折射反向学习的人工蜂群算法。该算法在引领蜂阶段利用差分变异搜索策略生成候选解，在引领蜂阶段后，将扰动向量加入折射反向学习(Refracted Opposition-Based Learning, ROBL)策略[7] [8]生成折射反向种群。通过基准测试函数对所提算法的性能进行测试，同时与其他算法进行对比分析，以验证所采用改进策略的可行性和有效性。

2. 人工蜂群算法介绍

在 ABC 算法中，蜂群分为引领蜂，跟随蜂和侦察蜂 3 种类型，其中引领蜂和跟随蜂各占蜂群的一半，数量等于蜜源的数量，且每个蜜源同一时间内只有一只引领蜂采蜜[9] [10] [11]，蜜源位置表示候选解，蜜源 i 的质量用适应度 fit 表示。

1) 种群初始化。假设维空间 D 中，蜜源数量 SN，蜜源被废弃的条件参数为 $limit$，根据式(1)确定第 i 个蜜源的位置 $X_i=\{x_{i1},x_{i2},\cdots,x_{ij},\cdots,x_{iD}\}$。

$$x_{ij}=x_j^{\min}+rand(0,1)\left(x_j^{\max}-x_j^{\min}\right),i=1,2,\cdots,SN;j=1,2,\cdots,D \tag{1}$$

其中，x_{ij} 为第 i 个蜜源的第 j 维的值，$x_j^{\max}$ 和 $x_j^{\min}$ 分别表示 x_{ij} 的最大值和最小值。每个蜜源位置相当于一个问题的可行解，可行解的适应度值 $fit(x_i)$，可根据式(2)计算，同时，$fit(x_i)$ 是人工蜂群算法的目标函数值。

$$fit(x_i)=\begin{cases}1/(1+f(x_i)) & f(x_i)\geq 0\\ 1+abs(f(x_i)) & f(x_i)<0\end{cases} \tag{2}$$

2) 引领蜂阶段。引领蜂采取“贪婪选择”将新旧蜜源适应度比较，当新蜜源优于旧蜜源时，将旧蜜源替换，否则保持不变。在进行邻域搜索时，根据式(3)由旧蜜源产生新蜜源 v_i。

$$v_{ij}=x_{ij}+\varphi(x_{ij}-x_{kj}) \tag{3}$$

其中，$k\neq i\in\{1,2,\cdots,SN\}$，$j$ 是随机选择的维度，$j\in\{1,2,\cdots,d\}$，φ 是 $[0,1]$ 中是一个随机数，用来控制搜索范围。

3) 跟随蜂阶段。引领蜂通过跳摇摆舞将蜜源信息传递给跟随蜂。此时，跟随蜂通过"轮盘赌"的方式来选择开采的蜜源。适应度值大越能吸引蜜蜂跟随，跟随蜂开采的蜜源由概率 $P(x_i)$ 来确定。

$$P(x_i)=\frac{fit(x_i)}{\sum_{i=1}^{SN}fit(x_i)} \tag{4}$$

跟随蜂选择的蜜源也将根据式(3)进行邻域搜索，同样采取贪婪选择。

4) 侦察蜂阶段。在搜索的过程中，如果蜜源经过 $trial$ 次迭代搜索到达阈值 $limit$ 后，还没更新为更好的蜜源，该蜜源就会被放弃。引领蜂转为侦察蜂，侦察蜂就会带领找到一个新的蜜源，侦察蜂找到的新的蜜源的位置根据式(1)随机生成。

3. 改进人工蜂群算法

ABC 算法在处理单峰问题时，不能充分利用种群等信息来确定最有效的搜索方向；在求解复杂的多峰问题时，也很容易陷入局部最优[12]，这些缺点限制了 ABC 算法的更广泛应用，因此本文在引领蜂阶段和跟随蜂阶段进行改进，提出了基于折射反向学习的人工蜂群算法。

3.1. 反向学习策略

在智能优化算法中，需要在种群初始化和搜索过程中产生随机数以供使用，反向学习策略考虑随机性的同时，也考虑反向性[3]。Rahnamayan 在文献[13]中从理论和实验上证明了反向学习比纯随机产生的解更好，Wang [14]提出了一种基于广义反向学习的新算法来提高差分进化算法的性能；暴励等[15]在改进 ABC 算法时利用反向学习产生初始解；毕晓君等[16]将基于反向学习的变异策略代替侦察蜂；杨小健等[17]用反向学习策略来改进引领蜂搜索阶段。其中，反向学习的相关定义如下：

定义 1 若 x 是定义在 $[a,b]$ 上的实数，则 x 的反向解 x^{op} 定义如下：

$$x^{op}=a+b-x \tag{5}$$

定义 2 若 $X=\{x_1,x_2,\cdots,x_i,\cdots,x_n\}$ 是定义在 n 维空间中的一点，且 $x_i\in[a_i,b_i]$，则 x_i 的反向解 x_i^{op} 定义如下：

$$x_i^{op} = a_i + b_i - x_i, i = 1, 2, \cdots, n \tag{6}$$

反向学习机制的主要思想是：对所有解求其反向解，并在其中选取较优解参与到下次迭代。

3.2. 光的折射原理

几何光学将光的传播和成像简化，笛卡尔在《折光学》中首次公布了具有现代形式正弦之比的规律，邵等[7]将折射原理用于改进反向学习，并应用于改进粒子群算法，有效地改进了全局搜索能力，其中光的折射相关定义和定律如下：

定义 3 光从一种介质斜射入另一种介质时，传播方向发生改变，从而使光线在不同介质的交界处发生偏折，这种现象称为光的折射。如图 1 所示。

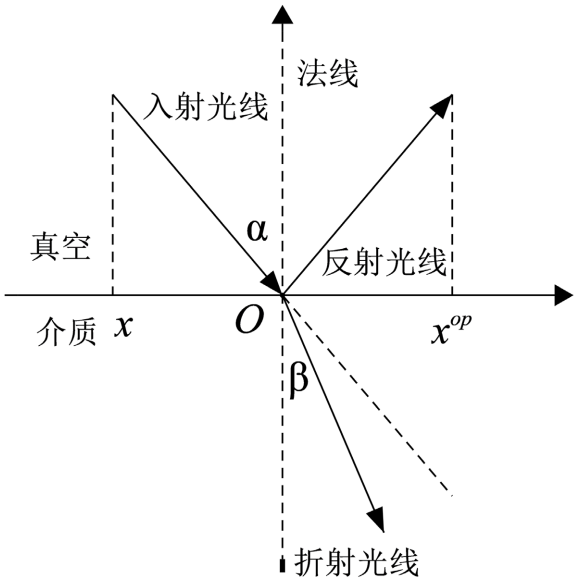

Figure 1. The phenomenon of refraction of light
图 1. 光的折射现象

定律 1 折射光线，入射光线和法线在同一平面内；折射光线与入射光线分居法线两侧；当光从空气斜射入其他介质中时，折射角小于入射角；当光从其他介质中斜射入空气时，折射角大于入射角；折射角随着入射角的增大而增大；当光线垂直射向介质表面时，传播方向不改变，这时入射角与折射角均为 0˚，称为光的折射定律。

定义 4 光从真空射入介质发生折射时，入射角 α 正弦值与折射角 β 正弦值的比值为一固定数值 n，称为该介质的绝对折射率，简称折射率，公式表示如下：

$$n = \sin\alpha / \sin\beta \tag{7}$$

式(7)又称为斯涅尔公式。

3.3. 折射反向学习策略

反向学习作为一种改进方式，在迭代初期可以提高算法的寻优能力，但在后期易陷入局部最优，导致收敛精度和速度变差。折射反向学习机制[8]是在反向学习的基础上引入光的折射原理来寻找最优解。折射反向学习的原理如图 2 所示。

在一维空间中，x 轴上方表示真空，下方表示其他介质，x 的搜索空间为$[a,b]$，y 为法线。为描述折

射反向学习机制做以下假设：在 A 点(入射点)有一光源，向 O 点发出入射光线 AO，设 AO 的长度为 l，由定义 3 与定律 1 可知，光线 AO 在 O 处发生折射现象，折射光线为 OA^r，A^r 为折射点，设 OA^r 的长度为 l^r，入射角和折射角分别为 α，β，O 为区间 $[a,b]$ 的中点。

定义 5 设入射点 A 在 x 轴的投影为 x，入射光线发生折射后产生的折射点 A^r 在 x 轴的投影为 x^r，称 x^r 为 x 的基于折射原理的反向解。

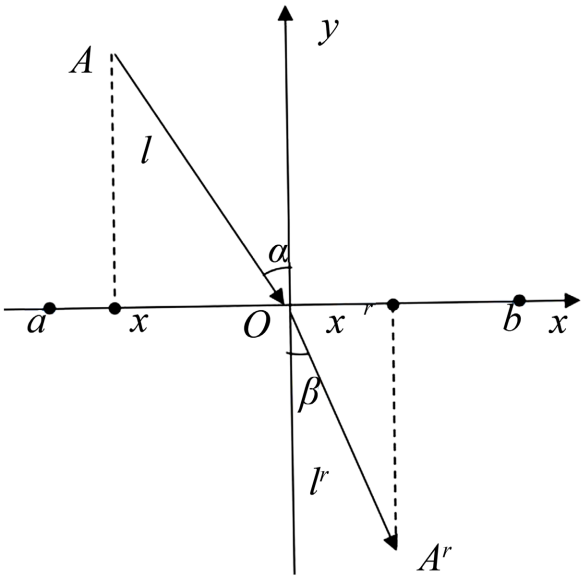

Figure 2. Refracted opposition-based learning mechanism
图 2. 折射反向学习机制

由图 2 中各线段的几何关系可得：

$$\sin\alpha = \left((a+b)/2 - x\right)/l \tag{8}$$

$$\sin\beta = \left(x^r - (a+b)/2\right)/l^r \tag{9}$$

由定义 4 与上两式可得：

$$n = \frac{(a+b)/2 - x}{x^r - (a+b)/2} \cdot \frac{l^r}{l} \tag{10}$$

令 $k = l/l^r$，则上式可化为：

$$kn = \frac{(a+b)/2 - x}{x^r - (a+b)/2} \tag{11}$$

由(11)可得折射反向学习解的计算公式：

$$x^r = \frac{a+b}{2} + \frac{a+b}{2kn} - \frac{x}{kn} \tag{12}$$

当 $k=1$，$n=1$ 时，式(12)可化为式(5)，ROBL 可以通过调整参数 k、n 获得动态候选解，而 OBL 只能得固定解。实际上，由图 1 可知，入射光线在发生折射的同时，还会出现光的反射现象。当入射光线等于反射光线长度时，反射点在 x 轴的投影恰好是 x 的反向解 x^{op}。由于优化问题都是多维的，折射反向

解可按下式计算：

$$x_{i,j}^{r}=\frac{a_j+b_j}{2}+\frac{a_j+b_j}{2kn}-\frac{x_{i,j}}{kn} \tag{13}$$

其中，$x_{i,j}$ 表示当前种群中第 i 个个体在第 j 维的值，$x_{i,j}^{r}$ 是求得的折射反向解，a_j 和 b_j 表示第 j 维的最小值和最大值，动态边界更新有利于的到得到更好的解，加速收敛。

3.4. 基于折射原理的反向学习人工蜂群算法

3.4. 1. 差分变异搜索策略

Storn 和 Price [18]为求解多项式问题提出了 DE 算法，它是模拟群体生物进化的一种随机模型，根据“适者生存”的原则，通过迭代得到最优解。DE 算法需要进行变异和交叉操作，变异操作计算公式如下：

$$V=X_{r1}+F\left(X_{r2}-X_{r3}\right) \tag{14}$$

其中，V 表示变异后的个体，F 表示变异算子，X_{r1},X_{r1},X_{r3} 表示随机选择的三个个体。变异操作有不同的变体，不同的变异操作对算法性能产生不同影响。

引领蜂根据(2)进行位置更新时，在邻域中随机选择一个蜜源进行操作产生候选解，此种选择方式具有良好的全局搜索能力，忽略了局部搜索能力。在算法寻优过程中，希望尽可能向最优解靠近，因此引入当前最优解指导的差分变异搜索方程：

$$v_{ij}=x_{ij}+\varphi\left(x_{ij}-x_{kj}\right)+\psi\left(x_{best,j}-x_{kj}\right) \tag{15}$$

ψ 是 $[0,1.5]$ 中是一个随机数，$x_{best,j}$ 是当前种群中最好个体的第 j 维的值。

3.4.2. 折射反向学习搜索策略

随着迭代的进行，差分变异搜索策略会降低种群多样性，也容易陷入局部最优，因此在引领蜂阶段后进行折射反向学习，产生当前种群的折射反向解，并在当前解和折射反向解中选择较优的作为下次迭代种群，此过程相当于在 $2*SN$ 个个体中选择出较优的 SN 个个体，在保证种群多样性的同时，可以提高跳出局部最优的概率，折射反向解计算方式如下：

$$X_{i,j}^{r}=\phi\left(\frac{\min_j+\max_j}{2}+\frac{\min_j+\max_j}{2kn}-\frac{X_{i,j}}{kn}\right) \tag{16}$$

其中，$X_{i,j}$ 表示当前种群中第 i 个个体在第 j 维的值，$\min_j,\max_j$ 表示为当前种群第 j 维的下限和上限，$\phi=rand\left(0,h\right)$ 表示 $[0,h]$ 上的均匀分布随机数，计算公式如下：

$$h=\begin{cases}h_1,t\le T/3\\ h_2,T/3<t<2T/3\\ h_3,t\ge 2T/3\end{cases} \tag{17}$$

其中，t 表示当前迭代次数，T 表示最大迭代次数。由式(16)可知，当 k 变化时，折射反向解的位置也会发生变化，所以可以通过调整 k 的值提高算法跳出局部最优解的概率，这里 k 是随着迭代次数的增加从 0 到 2 非线性递减，计算公式如下：

$$k=2\mathrm{e}^{-\left(\frac{3t}{T}\right)^2} \tag{18}$$

k 随 t 的变化曲线如图 3 所示。n 表示折射率，根据式(7)可知，实际跟入射角有关，但是在(16)中仅把 n 看作参数，n 的不同取值对算法性能产生不同影响。

以二维空间为例，如图 4 所示，在式(16)中不考虑扰动向量 ϕ，即 $\phi=1$，此时是一般的折射反向学习计算公式。当 k，n 都为 1 时，根据式(16)产生的反向解 X^{op} 距离最优解 X^* 较远，说明此时处于局部最优解区域；当 k，n 任一参数不为 1 时，产生的折射反向解 X_1^r 距离最优解 X^* 仍然较远，说明此时仍处于局部最优解区域；调整参数 k，产生的折射反向解 X_2^r 向最优解靠近，说明调整参数可以帮助跳出局部最优，再对 n 进行调整，得到折射反向解 X_3^r，此时距离最优解较近。

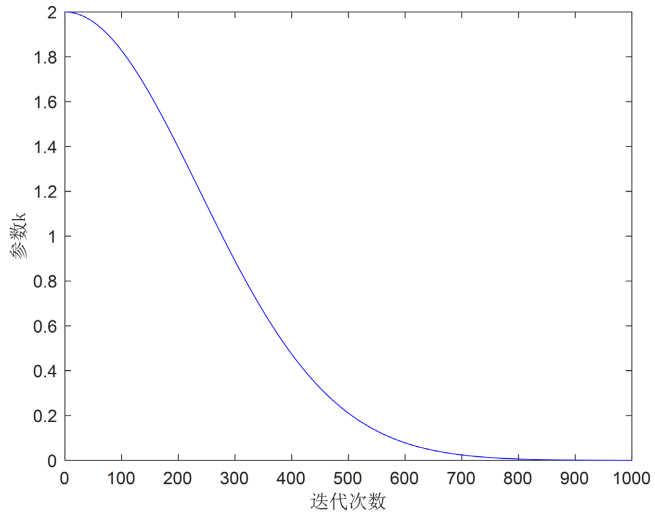

Figure 3. Variation curve of parameter k
图 3. 参数 k 的变化曲线

一般的折射反向学习产生的折射反向解不能利用折射反向解所在邻域的信息，因此通过加入扰动向量 ϕ 扩大搜索邻域，有助于跳出局部最优，这样产生的折射反向解在保证种群多样性的基础上，利用一般折射反向学习信息。在图 4 中阴影部分表示加入扰动向量 ϕ 后，反向解或折射反向解所在的区域，通过随机扰动后得到折射反向解 X_4^r，说明到达全局最优解 X^* 所在区域，经历多次迭代后。扰动向量 ϕ 由 h 取值决定，在迭代初期，需要保持算法的全局搜索能力，因此 h 取值较大，随着迭代次数的增加，在全局最优解区域搜索时，需要提高算法精度，因此 h 取值变小，采用随机扰动的方式可以提高产生的折射反向参与下次迭代的可能性。

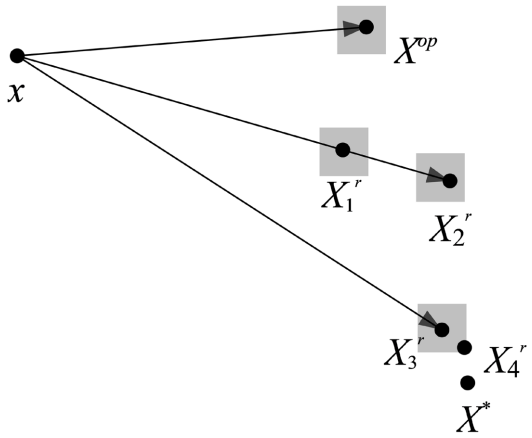

Figure 4. Two-dimensional space refraction of the reverse learning process
图 4. 二维空间折射反向学习过程

综上所述，折射反向学习搜索策略是通过调整参数 h，k，n 的值来得到折射反向解，通过比较当前解与折射反向解的适应度值选出较优的个体组成新的种群参与到下一次迭代，在保证种群多样性的同时，使算法跳出局部最优到达全局最优区域。

3.4.3. 所提算法

综合上述策略对基本 ABC 算法改进后，得到的 CRABC 算法基本思想是：引领蜂阶段利用差分变异向最优解靠近；引领蜂阶段后进行折射反向学习，参数 h 和 k 随迭代次数而变化，随着迭代的增加，折射反向学习搜索范围逐渐变小，提高了算法的收敛速度和精度。CRABC 算法实现流程如下：

Step 1 设置算法参数：种群规模，最大迭代次数等，随机生成初始种群；

Step 2 引领蜂根据式(15)搜索新的蜜源，根据贪婪选择对更好的蜜源进行替代；

Step 3 根据式(16)生成折射反向种群，在当前种群与折射反向种群选择优秀的个体组成新种群；

Step 4 根据式(3)计算引领蜂被跟随的概率；

Step 5 根据轮盘赌规则选择跟随，并搜索一个新的蜜源，如果更好则替代；

Step 6 如果一个蜜源在阈值 *limit* 内没有找到更好的蜜源，则被遗弃，侦察蜂根据式(1)产生新蜜源，去寻找更好的蜜源；

Step 7 判断是否满足终止条件，若满足则停止，否则返回 Step 2。

4. 实验与分析

4.1. 基准测试函数

为了验证 CRABC 算法在求解全局优化问题的性能，本文选择 12 个基准测试函数进行测试，基准测试维数为 30 维，表 1 给出了基准函数的序号，名称，类型，取值范围以及理论最优值。除 F_7 理论最优值为 −78.33236 外，其他都为 0, U, M 分别表示单峰函数，多峰函数，S, N 分别表示可分函数，不可分函数。

Table 1. Benchmark functions
表 1. 基准测试函数

No	Function	Type	Range	Global Optimum
F_1	Sphere	US	[−100, 100]	0
F_2	Schwefel2.22	UN	[−10, 10]	0
F_3	Schwefel2.21	UN	[−100, 100]	0
F_4	Rosenbrock	US	[−30, 30]	0
F_5	Step	US	[−100, 100]	0
F_6	Quartic	US	[−1.28, 1.28]	0
F_7	Himmelblau	MS	[−5, 5]	−78.33236
F_8	Rastrigin	MS	[−5, 12, 5.12]	0
F_9	Ackley	MN	[−32, 32]	0
F_{10}	Griewank	MN	[−600, 600]	0
F_{11}	Alpine	MS	[−10, 10]	0
F_{12}	Weierstrass	MN	[−0.5, 0.5]	0

4.2. 参数设置

采用 CRABC 算法对 12 个基准测试函数求解，并与 ABC 算法、仅采用差分变异搜索策略的 ABC 算

法(CABC)、仅加入折射反向学习搜索策略的 ABC 算法(RABC)进行比较，来验证综合两种策略后的 RABC 算法改进效果。

为了实验结果的公平性，将 4 种算法的参数设置一致。函数评价次数为 100,000，最大迭代次数为 1000，种群规模为 100 ($SN=50$)，蜜源迭代阈值为 $limit=0.6*dim*SN$ (dim 表示函数维数)，这与文献[19]设置一样，其中 RABC 算法与 CRABC 算法中的参数 $h_1=1$，$h_2=10^{-4}$，$h_3=10^{-8}$，$n=2$。

4.3. 实验结果与分析

4.3.1. 算法精度结果分析

为减少随机性带来的误差，表 2 给出 4 种算法在相同环境下独立运行 30 次后 12 个基准测试函数的最优值，平均值，方差，最差值以及一次实验的平均时长，其中黑体表示最佳结果。平均值和方差能够反映算法的收敛精度和稳定性。

从表 2 中可以看出，对于 12 个 30 维的单峰基准测试函数(F_1~F_6)和多峰基准测试函数(F_6~F_{12})除 F_4 和 F_8 外，RABC 算法和 CRABC 算法的平均值和方差都优于 ABC 算法，且在 F_{10} 上收敛到最小值，标准差也小于 ABC 算法。由于 F_4 为 Rosenbrock 函数，在低维时是单峰函数，高维情形具有多个理论最优解，所以四个算法得到的结果相差不大；而对于 F_8，虽然 CABC 算法精度更高也更稳定，但是 RABC 算法和 CRABC 算法却可以找到理论最优值。综上所述，提出的两种策略对提升算法性能是有效的，尤其是在 F_8、F_{10}、F_{12} 这三个函数找到了理论最优解。

表 3 给出了 CRABC，RABC 与文献[20]中提出的 GABC (C = 1.5)算法和文献[19]提出的 ABC/best/1 算法在 5 个基准测试函数上进行比较，Sphere，Rastrigin，Ackley，Griewank 这 4 个函数是在 30 维上测试，Rosenbrock 函数在 3 维上测试，在 Sphere，Griewank，Ackley，Rosenbrock 这 4 个函数上 CRABC 算法最优，尤其是 Griewank 函数可以找到理论最优解。

4.3.2. 算法运行时间分析

ABC 算法没有加入其他策略，而 CABC 算法只是在引领蜂阶段变化了搜索方程，所以这两种算法的运行时间差不多且少于 RABC 算法和 CRANC 算法；而 RABC 算法和 CRABC 算法引入了折射反向学习策略，耗时花费在折射反向学习，当前解与折射反向解比较上，所以在运行时间上要长于另两种算法，这也与表 2 中所列的 30 次平均运行时间吻合，由于运行环境影响，实验允许存在误差。

Table 2. Comparison of ABC, CABC, RABC and CRABC experimental results
表 2. ABC, CABC, RABC 与 CRABC 实验结果比较

No.	Algorithm	Best	Mean	SD	Worst	Time(s)
F_1	ABC	2.27e−11	2.60e−10	1.54e−10	6.86e−10	0.4462
	CABC	1.78e−17	2.06e−16	1.34e−16	5.34e−16	0.4537
	RABC	1.68e−33	1.27e−32	1.46e−32	7.79e−32	0.6073
	CRABC	**1.34e−37**	**6.97e−37**	**4.18e−37**	**6.97e−37**	0.6310
F_2	ABC	6.32e−07	1.29e−06	3.18e−07	1.89e−06	0.7552
	CABC	1.31e−09	2.65e−09	7.27e−10	4.04e−09	0.7669
	RABC	1.18e−24	6.87e−24	9.64e−24	4.61e−23	1.0908
	CRABC	**1.40e−26**	**1.45e−25**	**3.19e−25**	**1.763e−24**	1.1240
F_3	ABC	9.51e−00	1.61e+01	3.38e−00	2.11e+01	0.4845
	CABC	6.90e−00	1.14e+01	2.28e−00	1.51e+01	0.4896
	RABC	**1.06e−18**	2.53e−02	1.31e−01	7.18e−01	0.6808
	CRABC	4.18e−18	**7.87e−03**	**4.29e−02**	**2.35e−01**	0.6981

Continued

F_4	ABC	4.90e−03	1.52e−00	1.75e−00	**7.08e−00**	0.9228
	CABC	7.73e−04	**0.77e−00**	**1.63e−00**	8.58e−00	0.9208
	RABC	3.32e−16	2.09e+01	1.28e+01	2.86e+01	1.4459
	CRABC	**9.71e−17**	1.43e+01	1.43e+01	2.86e+01	1.4454
F_5	ABC	**0**	**0**	**0**	**0**	0.6308
	CABC	**0**	**0**	**0**	**0**	0.6536
	RABC	**0**	**0**	**0**	**0**	1.0017
	CRABC	**0**	**0**	**0**	**0**	0.9529
F_6	ABC	5.16e−02	9.66e−02	2.06e−02	1.39e−01	1.1588
	CABC	2.89e−02	6.39e−02	1.43e−02	9.24e−02	1.1987
	RABC	7.91e−06	**5.33e−05**	**4.49e−05**	**1.70e−04**	1.6393
	CRABC	**1.33e−06**	7.49e−05	5.89e−05	2.32e−04	1.6959
F_7	ABC	**−78.33233**	**−78.33233**	8.393e−14	**−78.33233**	1.1682
	CABC	**−78.33233**	**−78.33233**	4.25e−14	**−78.33233**	1.0862
	RABC	**−78.33233**	**−78.33233**	**1.69e−14**	**−78.33233**	1.4216
	CRABC	**−78.33233**	**−78.33233**	1.87e−14	**−78.33233**	1.7591
F_8	ABC	2.12e−10	3.31e−02	1.81e−01	9.95e−01	1.0795
	CABC	3.98e−13	**8.82e−11**	**3.18e−10**	**1.69e−09**	1.1049
	RABC	**0**	6.06e−09	2.26e−08	1.18e−07	1.5440
	CRABC	**0**	1.63e−09	8.88e−09	4.87e−08	1.4962
F_9	ABC	1.66e−06	4.52e−06	1.60e−06	7.59e−06	1.0678
	CABC	9.39e−09	1.76e−08	5.96e−09	3.95e−08	1.0404
	RABC	**8.88e−16**	1.01e−15	6.49e−16	4.44e−15	1.5265
	CRABC	8.88e−16	**8.88e−16**	**0**	**8.88e−16**	1.5537
F_{10}	ABC	2.87e−11	9.09e−09	1.46e−08	6.24e−08	0.9956
	CABC	4.55e−15	2.72e−09	1.11e−08	5.56e−08	1.0998
	RABC	**0**	**0**	**0**	**0**	1.3693
	CRABC	**0**	**0**	**0**	**0**	1.4010
F_{11}	ABC	4.90e−05	1.29e−04	9.17e−05	5.05e−04	0.6.473
	CABC	2.64e−06	3.31e−05	2.51e−05	9.50e−05	0.6772
	RABC	1.13e−19	**1.58e−10**	**7.27e−10**	**3.99e−09**	0.9935
	CRABC	**3.64e−26**	4.87e−10	1.61e−09	8.19e−09	0.9954
F_{12}	ABC	2.57e−04	4.22e−04	9.28e−05	6.55e−04	37.7078
	CABC	1.56e−06	6.01e−06	2.40e−06	1.15e−05	37.9231
	RABC	**0**	4.68e−11	2.02e−10	1.06e−09	57.2289
	CRABC	**0**	**1.30e−14**	**7.13e−14**	**3.91e−13**	56.6982

Table 3. Comparison of GABC (C = 1.5), ABC/best/1, RABC and CRABC experimental results
表 3. GABC (C = 1.5), ABC/best/1, RABC 与 CRABC 实验结果比较

Function		GABC(C=1.5)	ABC/best/1	RABC	CRABC
Sphere	Mean	4.18e−16	1.57e−27	9.70e−33	**1.39e−37**
	SD	7.26e−17	1.14e−27	6.35e−32	**1.67e−37**
Rastrigin	Mean	1.37e−14	**0**	4.25e−09	2.12e−09
	SD	2.45e−14	**0**	3.12e−08	3.26e−09
Ackley	Mean	3.22e−14	1.26e−13	1.01e−15	**8.88e−16**
	SD	3.25e−15	3.48e−14	6.48e−16	**0**
Griewank	Mean	2.96e−17	4.23e−11	**0**	**0**
	SD	4.99e−17	2.16e−11	**0**	**0**
Rosenbrock (D = 3)	Mean	2.65e03	9.06e−06	2.12e−04	**6.28e−20**
	SD	2.22e−03	1.41e−05	1.17e−03	**1.34e−19**
算法排名		4	3	2	1

4.3.3. 相关参数确定和说明

1) 参数 h 和 k 的影响

在迭代初期，算法需要较好的全局搜索能力，因此 h 和 k 取值较大，相应的搜索范围也大，增加了找到潜在最优解所在区域的可能；随着迭代的进行，h 和 k 取值变小，这样保证在折射反向解附近搜索；当 h 和 k 取值足够小时，折射反向解在当前解的足够小邻域内，提高了开发能力。

2) 参数 n 的影响

n 实际表示折射率，跟入射角有关，在算法中仅将其看作一个参数，因此需要找到合适的 n 值，来保证算法性能。以 F_1 为例，选取不同的 n 值做 30 次独立实验，取平均值。从表 4 可以看出，n 取 2 时取值最优，过小会导致性能下降，因此，参数 n 取 2。

Table 4. The test function value corresponding to the different values of parameter n
表 4. 参数 n 的不同取值对应的测试函数值

参数 n 的取值	最优值
0.002	7.51e−14
0.02	2.50e−10
0.2	7.25e−15
2	6.97e−37

4.3.4. 算法收敛分析

为进一步说明算法的收敛性，图 5 给出了 4 种算法在 12 个 30 维基准测试函数上的收敛曲线。从图中可以看出，RABC 算法和 CRABC 算法在迭代初期就可以保持较高的收敛速度。对于 F_5 这一非连续的单峰函数，RABC 算法和 CRABC 算法可以在不到 20 次迭代找到最优值。对于 F_4，F_6，F_{11} 这三个函数，可以看到 RABC 算法和 CRABC 算法的收敛曲线呈阶段式下降，这与参数 h 有关，h 的变化使得搜索区域在最优解潜在区域更小的范围进行。F_8，F_{10}，F_{12} 在某次迭代后收敛速度迅速提升，这是因为折射反向

学习能很快搜索到最优解的潜在区域。

通过分析表 2 和图 5，可以得到结论：引入差分变异搜索策略和折射反向学习搜索策略的 CRABC 算法在总体上优于 ABC 算法以及两种单策略改进的 ABC 算法，能更好地平衡探索和开发能力，提高了算法收敛速度和精度。

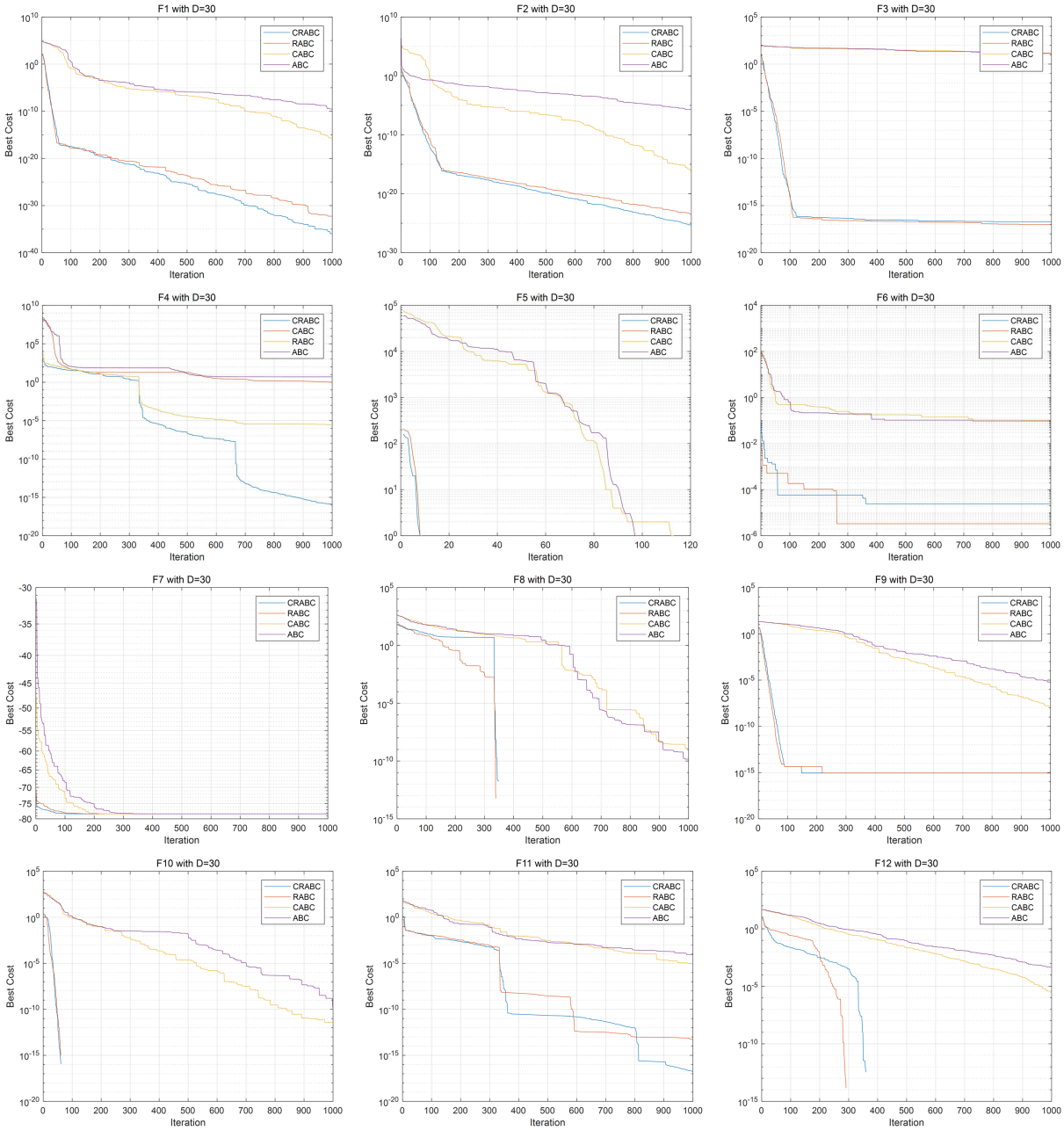

Figure 5. Convergence curves of each algorithm
图 5. 各算法收敛曲线

5. 结束语

本文将差分变异搜索策略和折射反向学习搜索策略引入 ABC 算法，提出了基于折射反向学习的人工

蜂群算法(CRABC)。在引领蜂阶段，根据当前最优解来指导候选解生成；然后计算当前种群的折射反向解，并根据适应度值在折射反向种群和当前种群选择较好的个体参与下一次迭代。通过对 12 个基准测试函数以及实验研究，分析了算法的探索和开发能力，同时与 GABC (C = 1.5)算法，ABC/best/1 算法进行比较实验。结果表明：差分变异搜索策略和折射反向学习搜索策略对提高算法性能是有效的，CRABC 算法的求解精度和收敛速度是有优越性的，未来可考虑用于求解实际优化问题。

基金项目

牡丹江师范学院国家级课题培育项目(GP2020003)；国家自然科学基金项目(11871097)；牡丹江师范学院科技创新项目(kjcx2020-009mdjnu)。

参考文献

[1] Karaboga, D. (2005) An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Erciyes University, Kayseri.

[2] Li, G.Q., Niu, P.F. and Xiao, X.J. (2012) Development and Investigation of Efficient Artificial Bee Colony Algorithm for Numerical Function Optimization. *Applied Soft Computing*, **12**, 320-332. https://doi.org/10.1016/j.asoc.2011.08.040

[3] Tizhoosh, H. (2005) Opposition-Based Learning: A New Scheme for Machine Intelligence. *Proceedings of International Conference on Intelligent Agent, Web Technologies and Internet Commerce*, Vol. 1, 695-701. https://doi.org/10.1109/CIMCA.2005.1631345

[4] Rahnamayan, S., Tizhoosh, H.R. and Salama, M.M. (2006) Opposition-Based Differential Evolution Algorithm. *IEEE Congress on Evolutionary Computation*, Vancouver, 16-21 July 2006, 2010-2017. https://doi.org/10.1109/CEC.2006.1688554

[5] Zhao, J., Lv, L., Fan, T.H., Wang, H., Li, C.X. and Fu, P. (2014) Particle Swarm Optimization Using Elite Opposition-Based Learning and Application in Wireless Sensor Network. *Sensor Letters*, **12**, 404-408. https://doi.org/10.1166/sl.2014.3257

[6] 王剑, 王冰, 葛孟珂. 基于反向学习的人工蜂群算法[J]. 牡丹江师范学院学报(自然科学版), 2022(1): 23-30.

[7] 邵鹏, 吴志健, 周炫余, 邓长寿. 基于折射原理反向学习模型的改进粒子群算法[J]. 电子学报, 2015, 43(11): 2137-2144.

[8] 范千, 陈振健, 夏樟华. 一种基于折射反向学习机制与自适应控制因子的改进樽海鞘群算法[J]. 哈尔滨工业大学学报, 2020, 52(10): 183-191.

[9] Karaboga, D. and Basturk, B. (2008) On the Performance of Artificial Bee Colony (ABC) Algorithm. *Applied Soft Computing*, **8**, 687-697. https://doi.org/10.1016/j.asoc.2007.05.007

[10] Karaboga, D. and Basturk, B. (2007) A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *Journal of Global Optimization*, **39**, 459-471. https://doi.org/10.1007/s10898-007-9149-x

[11] Tereshko, V. and Loengarov, A. (2005) Collective Decision Making in Honey-Bee Foraging Dynamics. *Computing and Information Systems*, **9**, 1.

[12] Karaboga, D. and Akay, B. (2009) A Comparative Study of Artificial Bee Colony Algorithm. *Applied Mathematics and Computation*, **214**, 108-132. https://doi.org/10.1016/j.amc.2009.03.090

[13] Rahnamayan, S., Tizhoosh, H.R. and Salama, M.M. (2008) Opposition versus Randomness in Soft Computing Techniques. *Applied Soft Computing*, **8**, 906-918. https://doi.org/10.1016/j.asoc.2007.07.010

[14] Wang, H., Wu, Z. and Rahnamayan, S. (2011) Enhanced Opposition-Based Differential Evolution for Solving High-Dimensional Continuous Optimization Problems. *Soft Computing*, **15**, 2127-2140. https://doi.org/10.1007/s00500-010-0642-7

[15] 暴励, 曾建潮. 一种双种群差分蜂群算法[J]. 控制理论与应用, 2011, 28(2): 266-272.

[16] 毕晓君, 王艳娇. 加速收敛的人工蜂群算法[J]. 系统工程与电子技术, 2011, 33(12): 2755-2761.

[17] 杨小健, 董毅伟. 基于反向学习的自适应快速人工蜂群算法[J]. 系统仿真学报, 2016, 28(11): 2684-2691+2700. https://doi.org/10.16182/j.issn1004731x.joss.201611006

[18] Storn, R. and Price, K. (1997) Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, **11**, 341-359. https://doi.org/10.1023/A:1008202821328

[19] Gao, W.F., Liu, S.Y. and Huang, L.L. (2012) A Global Best Artificial Bee Colony Algorithm for Global Optimization. *Journal of Computational and Applied Mathematics*, **236**, 2741-2753. https://doi.org/10.1016/j.cam.2012.01.013

[20] Zhu, G.P. and Kwong, S. (2010) Gbest-Guided Artificial Bee Colony Algorithm for Numerical Function Optimization. *Applied Mathematics and Computation*, **217**, 3166-3173. https://doi.org/10.1016/j.amc.2010.08.049