

新型全局优化蝙蝠算法

李煜^{1,2} 马良¹

(上海理工大学管理学院 上海 200093)¹ (河南大学管理科学与工程研究所 开封 475004)²

摘要 通过对生物智能机理的借鉴,许多解决复杂问题的新方法不断涌现。最近, Yang 基于蝙蝠的回声定位行为,提出了一种新的全局优化算法——蝙蝠算法,同时将一些现有算法的优点引入到该算法中。首先讨论了蝙蝠算法的生物学动机,从原理上描述了蝙蝠回声定位行为和算法实现流程,随后求解了函数极值优化问题。仿真结果表明,蝙蝠算法的性能优于粒子群算法。最后,对进一步研究作了展望。

关键词 蝙蝠算法,全局优化,自然计算,函数优化

中图分类号 TP301.6 **文献标识码** A

Bat-inspired Algorithm: A Novel Approach for Global Optimization

LI Yu^{1,2} MA Liang¹

(School of Management, University of Shanghai for Science and Technology, Shanghai 200093, China)¹

(Research Institute of Management science and Engineering, Henan University, Kaifeng 475004, China)²

Abstract The study of bionics bridges the functions, biological structures and organizational principles found in nature with our modern technologies, and numerous mathematical and meta-heuristic algorithms have been developed along with the knowledge transferring process from the life forms to the human technologies. Recently, a new global optimization algorithm, called Bat-inspired Algorithm(BA), has been developed by Yang. The presented algorithm is inspired by fundamentals of echolocation of micro bats, and intends to combine the advantages of existing algorithms into the new bat algorithm. The first part of the paper was devoted to the detailed description of the existing algorithm. Subsequent sections concentrated on the performed experimental parameter studies and a comparison with efficient particle swarm optimizer based on existing benchmark functions. Finally the implication of the results and potential topics for further research was discussed.

Keywords Bat-inspired algorithm, Global optimization, Nature-inspired computation, Function optimization

大自然经过亿万年继承创造的演化,展现出完美的高效性和神奇性,复杂多样的生物界在长期物竞天择、优胜劣汰的进化过程中不断适应复杂环境的变化得到了生存和发展。人们从自然系统的研究中学到许多知识,并用这些知识发展新的算法和模型来求解复杂问题。正是模仿生物的智能行为,借鉴其智能机理,使得许多解决复杂问题的新方法不断涌现。通过对自然智能的建模,我们取得了许多成功从而创建出许多的智能系统,这些智能算法包括遗传算法^[1]、蚁群算法^[2]、粒子群算法^[3,4]、中心引力搜索算法^[5,6]和量子计算^[7]等。每种智能算法都对应一种实际的启发源,例如, DNA 计算^[8]是基于诺贝尔生理学或医学奖获得者 Watson 和 Crick 提出的双螺旋结构和诺贝尔化学奖获得者 Mullis 提出的聚合酶链反应;人工蜂群算法^[9]是基于诺贝尔生理学或医学奖获得者 Frisch 破译的蜜蜂跳舞行为;而人工免疫算法^[10]是基于获诺贝尔生理学或医学奖获得者 K. Jerne 提出的免疫网络理

论。智能优化算法奇妙的思路和广阔的应用领域吸引了大量的研究人员不断探索,不断创新。

2010年,剑桥大学学者 YANG 通过模拟蝙蝠回声定位行为,提出一种新的智能优化算法——蝙蝠算法(Bat-inspired Algorithm)^[11]。该算法是一种基于种群的随机寻优算法,蝙蝠个体是蝙蝠算法的基本单元,整个群体的运动在问题求解空间中产生从无序到有序的演化过程,从而获取最优解。蝙蝠算法中,每个优化问题的解都是搜索空间中的一个蝙蝠,所有蝙蝠都有一个优化问题决定的适应值,每个蝙蝠通过调整频率、响度、脉冲发射率,追随当前的最优蝙蝠在解空间中搜索。蝙蝠算法结合了现有成功算法的主要优点和回声定位的新特性,本文讨论了蝙蝠算法的生物学动机,从原理上描述了蝙蝠回声定位行为和算法实现流程,随后给出了蝙蝠算法的数学模型,最后结合函数极值优化问题给出了蝙蝠算法的具体应用,并将该算法与粒子群优化算法进行了比较。仿真结

到稿日期:2012-11-14 返修日期:2013-02-25 本文受国家自然科学基金项目(70871081),河南省科技攻关重点项目(102102210022,122102210201)资助。

李煜(1969—),女,博士生,副教授,主要研究方向为智能优化、电子商务;马良(1964—),男,博士,教授,博士生导师,主要研究方向为系统工程、智能优化。

果表明,蝙蝠算法明显优于粒子群算法。

1 蝙蝠算法

1.1 微型蝙蝠的行为

蝙蝠是哺乳动物中分布最广泛、数量最多的类群之一,高度进化的回声定位系统使它们成为进化最成功的类群之一^[12-14]。1793年,意大利科学家 Lazzaro Spallanzani 首次提出蝙蝠依靠听觉进行空中定位和搜索猎物。20世纪30年代后期,美国动物学家 Donald R. Griffin 提出回声定位的概念,并且利用声纳探测器对飞行的蝙蝠进行探测。蝙蝠是哺乳类中第二大目,全世界有 977 种,我国有 105 种,占兽类种类的 17.59%。蝙蝠种类繁多,大多数为食虫类。蝙蝠擅长夜间活动,其视觉较差,但听觉异常发达,因为能够有效地利用回声定位,在夜间或十分昏暗的环境中也能自由飞翔并准确无误地捕捉猎物。蝙蝠的大小差别很大,最大的蝙蝠是犛斯麦飞狐蝠,翼展达 1.65m,体重达 1kg。最小的蝙蝠是一种蝶蝠,体重仅 4g,翼展仅 2cm。蝙蝠能听到频率高达 212kHz 的超声波,人耳只能听到 20kHz 以下的声音。微型蝙蝠约有 780 个物种,几乎都有回声定位系统,多能自口腔发出从 7000Hz 至 70000Hz 不同频率的声波,并用发达的耳接收返回声波。蝙蝠发出速度低(约 10 次/秒)而频率高(60000 至 70000Hz)的声波,当检测到有昆虫等猎物存在时,频率下降;随着向目标靠近,回声定位频率也越来越低,听到的回声频率也是逐步下降的,这种改变频率的特点使得蝙蝠能在嘈杂的外界环境中,迅速辨认出自己的声波。

1.2 回声定位的声学原理

回声定位是一个复杂的、高度进化的过程,是指动物通过对自身发射声波的回声的分析来建立其周围环境的声学图像。蝙蝠对自己发出的超声波的回声进行分析,不仅能判断附近障碍物的距离,还能发现昆虫以便捕捉。蝙蝠通过回声可以知道昆虫的大小、形状及其运动方向,因此捕捉极为准确。这一过程就叫做回声定位。

许多研究认为,蝙蝠根据自己发出的声波与接收的声波的时间差确定物体的距离,依据回声到达两耳的不同时间与不同强度来判断物体方位,利用回声定位波形变化来识别物体的性质,判断是敌害、猎物还是建筑物。这一切活动都在极短的时间内发生,因为它们接近物体时,随着频率降低,声波速度越来越大,可达 100~200 次/s,即每个脉冲所需时间仅为 1/100~1/200s,检测速度十分惊人。在一般情况下,蝙蝠平均每分钟能捕到 10 只蚊子或 14 只果蝇大小的昆虫。蝙蝠的声纳系统非常精巧,动物学家在 200 年前首次进行了实验,结果显示蝙蝠在吊挂着许多悬线的暗室中往返飞行,没有一个会撞线。蝙蝠通过嘴或鼻孔发出高频声波,以此探路。一只蝙蝠接近一个静止的昆虫时接收的回声的频率将会升高,当昆虫飞向蝙蝠时频率的升高幅度将更大。有些蝙蝠能利用相对于其周围物体的多普勒变化来计算它们的飞行速度。

声音在空气中的速度通常为 $v=340\text{m/s}$,而超声波在 f 频率下的波长为:

$$\lambda=v/f \quad (1)$$

通常频率为 25kHz 到 150kHz,波长的范围在 2mm 到 14mm 之间。这样的波长符合它的猎物的大小。发出的在超声波范围内的声波,其响度能达到 110dB。响度可以从搜索

猎物时的最高变化到靠近猎物时的最静音。根据实际的频率,这样短暂的脉冲的传送距离通常只有几米。微型蝙蝠通常能够设法避开障碍物,哪怕障碍物只有发丝大小。研究表明,微型蝙蝠利用发出和探测回声的时间延迟,利用双耳的时间差,利用回声的响度变化去建立周围环境的三维场景。蝙蝠能够探测目标物的距离与方向、猎物的种类、猎物的移动速度,哪怕猎物只是一只小昆虫。

1.3 蝙蝠算法

模拟微型蝙蝠回声定位理论进而形成一种新的智能计算模式,该算法基于如下基本假设:

(1)蝙蝠运用回声定位感应距离,并能感知食物和障碍物之间的区别。

(2)蝙蝠在位置 X_i 以速度 V_i 随机飞行,以固定频率 f_{\min} (或 λ)、可变化波长 λ (或 f) 和响度 A_0 搜索猎物,它们根据猎物与自己的距离调节发射出的脉冲波长(或频率),并在靠近猎物时调整发射脉冲的频度 $r \in [0, 1]$ 。

(3)假定响度变化从最大值(正值) A_0 到最小值 A_{\min} 。

通常频率的范围为 $[f_{\min}, f_{\max}]$,其对应的波长范围为 $[\lambda_{\min}, \lambda_{\max}]$,如频率范围 $[20\text{kHz}, 500\text{kHz}]$ 对应的波长范围便是 0.7mm 到 17mm。实际应用中,可以根据不同的问题使用任何的波长,通过调整波长(或频率)来调整范围,先选择相应的区域,然后慢慢缩小。我们可以假定 $f \in [0, f_{\max}]$,频率越高,它的波长越短,飞行距离越短,通常飞行范围在几米以内。发射脉冲的频度在 $[0, 1]$ 范围内,其中 0 表示没有脉冲,1 意味着最大发射频度。

基于以上分析,蝙蝠搜索算法的主要步骤可以描述如下:

Step 1 参数初始化,目标函数 $f(X)$, $X=(x_1, \dots, x_d)^T$, 初始的蝙蝠种群 $X_i (i=1, 2, \dots, n)$ 和 V_i , 脉冲频率 f_i at X_i , 初始化脉冲速率 r_i 和声音响度 A_i ;

Step 2 通过调整频率产生新的解并更改速度和位置;

Step 3 如果 $(\text{rand} > r_i)$ 从最佳解集中选一个解,在选择的最佳解附近形成一个局部解;

Step 4 通过随意飞行产生一个新解;

Step 5 If $(\text{rand} < A_i \ \& \ f(X_i) < f(X^*))$, 接受这个新的解,增大 r_i 减小 A_i ;

Step 6 排列蝙蝠并找到当前最佳 X^* ;

Step 7 如未满足结束条件,则返回 Step 2;

Step 8 输出全局最优位置。

在一个 d 维搜索空间中,蝙蝠新位置 X_i^t 和新速度 V_i^t 在步骤 t 时的更新公式如下:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \beta \in [0, 1] \quad (2)$$

$$V_i^t = V_i^{t-1} + \epsilon(X_i^t - X^*)f_i \quad (3)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (4)$$

其中, $\beta \in [0, 1]$ 是一个随机向量, X^* 是当前全局最佳位置。根据所求解的问题类型,在固定 λ_i (或 f_i) 的同时改变 f_i (或 λ_i)。在实际求解过程中,我们可以根据问题的领域大小确定 f_i 的取值,比如使用 $f_{\min}=0$ 和 $f_{\max}=100$ 。开始时每只蝙蝠随机分配频率,频率是从 $[f_{\min}, f_{\max}]$ 平均得出的。局部搜索时,每只蝙蝠的更新公式如下:

$$X_{\text{new}} = X_{\text{old}} + \epsilon A', \epsilon \in [-1, 1] \quad (5)$$

在这里 $\epsilon \in [-1, 1]$ 是随机数, $A' = \langle A_i \rangle$ 是所有蝙蝠在这一代里的平均响度。蝙蝠的速度和位置更新与标准粒子群算

法有些相似,在这里, f_i 基本上控制了蝙蝠运动的节奏和范围。所以,在某种程度上可以认为 BA 是标准粒子群算法和由响度和脉冲率控制的局部搜索的一种均衡组合。

然后,脉冲发射的响度 A_i 和速率 r_i 也要随着迭代过程进行更新。一旦蝙蝠发现了猎物,响度会逐渐降低,脉冲速率会逐渐提高。响度可以选为任何便利值,例如,我们可以用 $A_0 = 100$ 和 $A_{\min} = 1$,或者我们也可以用 $A_0 = 1$ 和 $A_{\min} = 0$ 。我们假设 $A_{\min} = 0$ 表示一只蝙蝠刚刚发现猎物,然后暂时停止发出声音。脉冲发射的响度 A_i 和速率 r_i 的更新公式如下:

$$A_i = \alpha A_i^{-1}, r_i = r_i^0 [1 - e^{-\gamma}] \quad (6)$$

在这里 α 和 γ 是恒量, α 类似于模拟退火算法中冷却进程表中的冷却因素。对于任何 $0 < \alpha < 1$ 和 $\gamma > 0$ 的量都有

$$A_i \rightarrow 0, r_i \rightarrow r_i^0, \text{ as } t \rightarrow \infty \quad (7)$$

初始化时,每只蝙蝠所发出的响度和脉冲速率随机给出。一般情况下,我们定义初始的响度 A_i^0 通常在 $[1, 2]$ 之间,初始的发射速率 r_i^0 一般在 0 左右。蝙蝠的响度和发射率将随搜索过程不断更新,从而逐渐飞向最优解。

蝙蝠算法的伪代码表示如下:

Initialize the bat population $x_i (i=1, 2, \dots, n)$ and v_i

Initialize frequencies f_i , pulse rates r_i and the loudness A_i

While($t < \text{maximum number of iterations}$)

For $i=1:n$

Generate new solutions using(2),(3) and(4)

If $\text{rand} > r_{\text{new}}$

Select one among the best solutions and generate a local solution around this one, using(5)

Else

Select randomly a solution and generate a local solution around this one, using(5)

End if

Evaluate the bats

If($\text{rand} < A_i$) and ($f(x_i) < f(x^*)$)

$F(x_i) = f(x^*)$

Increase r_i and reduce A_i , using(6)

End if

End for

Rank bats to find the current best solutions

End while

2 仿真实验

在文献[15-17]中有很多基准测试函数,从中选择 8 个函数来验证算法的性能,并与 PSO 算法进行比较。模拟中,仿真环境为 Windows XP 操作系统、Intel 处理器 2.40GHz、1G 内存、仿真软件 Matlab7.0。

(1)Hansen Function,函数图形如图 1 所示。

$$f_1(x) = \sum_{i=1}^5 i \cdot \cos((i-1) \cdot x_1 + i) \cdot \sum_{j=1}^5 j \cdot \cos((j+1) \cdot x_2 + j), x_i \in [-10, 10]$$

该函数是一个多峰函数,局部极值点有 760 个。运行蝙蝠算法很快就可以收敛到最优解,可得全局最优值 -176.541793 ,并经多次反复运行,可搜索到全部最优解: $(-7.589893, -7.708314)$, $(-7.589893, -1.425128)$, $(-7.589893, 4.858057)$, $(-1.306708, -7.708314)$,

$(-1.306708, -1.425128)$, $(-1.306708, 4.858057)$,

$(4.976478, -7.708314)$, $(4.976478, -1.425128)$, $(4.976478, 4.858057)$ 。

收敛性态如图 2 所示。其余函数运算结果在表 1 中给出。

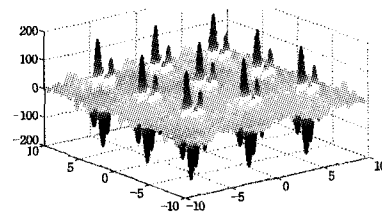


图 1 $f_1(x)$ 函数的空间特征

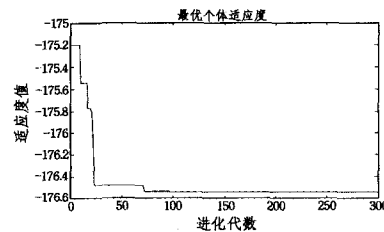


图 2 BA 算法 $f_1(x)$ 函数收敛图

(2)Schaffer Function

$$f_2(x, y) = \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1 + 0.001(x^2 + y^2)]^2} - 0.5, x, y \in [-4, 4]$$

该函数的最优解为 $-1, X^* = (0, 0)$,运行蝙蝠算法在迭代 20 次内每次均能得到最优值。

(3)Branin Function

$$f_3(x) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10, -5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$$

该问题的两个极值点为: $(-3.142, 2.275)$, $(3.142, 2.275)$,全局最优值 $Z^* = 0.398$ 。

(4)Bohachevsky Function

$$f_4(x, y) = 0.3 \cos 3\pi x - 0.3 \cos 4\pi x - x^2 - y^2 - 0.3, x, y \in [-1, 1]$$

该函数的最优解为 $0.2400, X^* = (0, -0.23)$; $(0, 0.23)$ 。

(5)Six-Hump Camel-Back Function

$$f_5(x) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1 x_2 - 4x_2^2 + 4x_2^4$$

该函数的全局最优解 $-1.0316285, X^* = (0.08983, -0.7126)$; $(-0.08983, 0.7126)$ 。

(6)多峰函数

$$f_6(x) = [0.01 + \sum_{i=1}^5 \frac{1}{i + (x_i - 1)^2}]^{-1}, -10 \leq x_i \leq 10$$

该函数极限位置为 $(1, 1, 1, 1)$,全局最小值为 0.436。

(7)Goldstein-Price Function

$$f_7(x, y) = [1 + (x + y + 1)^2 (19 - 14x + 3x^2 - 14y + 6xy + 3y^2)] \times [30 + (2x - 3y)^2 (18 - 32x + 12x^2 + 48y - 36xy + 27y^2)], |x| \leq 2, |y| \leq 2$$

此函数有 4 个极小值点: $(1.2, 0.8)$, $(1.2, 0.2)$, $(-0.6, -0.4)$, $(0, -1)$ 全局极小值点为 $(0, -1)$,全局极小值为 3。

(8)Camel Function

$$f_8(x, y) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2$$

此函数全局极小值为 $\min f_3(x^*) = -1.031628$ 。

最近的研究表明, PSO 算法可以超越遗传算法(GA)和其它许多优化问题的现有算法。Shilane 等已经详细提出了一个评估算法统计性能的总体框架^[18]。为了比较新算法的性能,我们针对上述各种标准测试函数,将蝙蝠算法与 PSO 算法进行比较。在使用 Matlab 实施这些算法之后,为了进行有意义的统计分析,每一种算法至少运行 100 次,循环次数设定为 1000 次。计算结果如表 1 所列。

表 1 BA 与粒子群算法的比较

函数	算法	最优值	最差值	平均值
F1	PSO	-176.5418	-176.5412	-176.5415
	BA	-176.5418	-176.5418	-176.5418
F2	PSO	-0.99999	-0.99667	-0.99966
	BA	-1	-0.99687	-0.99999
F3	PSO	0.398	0.398	0.398
	BA	0.398	0.398	0.398
F4	PSO	0.2400	0.2394	0.2399
	BA	0.2400	0.2400	0.2400
F5	PSO	-1.0316	-1.0316	-1.0316
	BA	-1.0316	-1.0316	-1.0316
F6	PSO	0.436	0.4367	0.4362
	BA	0.436	0.436	0.436
F7	PSO	3	3.0003	3.0001
	BA	3	3	3
F8	PSO	-1.031628	-1.031610	-1.031623
	BA	-1.031628	-1.031628	-1.031628

此外,我们尝试变换蝙蝠种群的数量 n , 使用了 $n=5, 10, 15, 20, 50, 100, 150, 250$ 。从模拟中发现, n 取 15 至 50 时对于大多数优化问题效果良好, 因此在所有仿真模拟中使用固定值 $n=40$ 。其中 $\alpha=\gamma=0.9$, α 类似于模拟退火算法中冷却进程表中的冷却因素, 参数的选择需要进行试验。初始化时, 随机给出每只蝙蝠所发出的响度和脉冲速率的值。粒子群算法有许多的变种算法, 如平均算法比标准粒子群算法更好用^[19]。然而, 标准粒子群算法是到目前为止使用最普遍的算法。因此, 我们使用标准粒子群算法来做比较。PSO 算法参数设置采用文献^[20]推荐的设置值: $w=0.729, c1=c2=1.49445$ 。

从表 1 可以看到, BA 在寻找全局最优时是更有效的, 它的成功率较高, 蝙蝠算法明显优于粒子群算法, 每个函数的计算在个人电脑上可以瞬间完成。蝙蝠算法一方面吸取现有算法的优点, 另一方面结合了蝙蝠惊人的回声定位能力。如果使 $A_i=0$ 及 $r_i=1$ 替换频率的变化, 那么蝙蝠算法本质上就是标准粒子群算法。同样, 如果使用固定响度和速率, 如 $A_i=r_i=0.7$, 由于波长(频率)的改变实质上是音高的改变, 蝙蝠算法此时就是和声算法。蝙蝠算法和粒子群算法一样采用了种群的方式进行搜索, 这使得它可以同时搜索待优化目标函数解空间中的多个区域。另一方面, 需要在个性和社会性之间寻找平衡, 也就是说, 既希望个体具有个性化, 又希望其指导其它个体已经找到的好解并向它们学习, 即社会性。

结束语 蝙蝠算法完美结合了粒子群优化算法和遗传算法以及和声搜索这些算法的主要优点, 粒子群算法及和声搜索是简化的蝙蝠算法的特殊情况。事实上, 参数 α 所起的作用类似于模拟退火中冷却进程表的作用。虽然具体操作要比其他元启发式算法更加复杂, 但蝙蝠算法完美结合了现有成功算法的主要优点和回声定位基础上的新特色。蝙蝠算法是通过调整频率、响度、脉冲发射率产生的, 其质量受制于响度

和脉冲速率。我们对典型的函数优化问题进行了仿真测试, 并和粒子群算法进行了比较, 结果表明 BA 明显优于粒子群算法。

蝙蝠算法是一种很有前途的算法, 在以后的工作中可以使用不同的波长或频率变化方案代替现在的线性实施。此外, 脉冲发射的速率和响度也可以以更复杂的方式变化。另一个扩展是使用脉冲发射和回声弹回的时间延迟, 例如, 对于旅行商问题, 随着时间的延迟两个相邻节点之间的距离可以被编码。微型蝙蝠使用双耳获取信息的时间不同, 因此它们可以识别猎物的类型和速度。进一步延伸的蝙蝠算法可以利用方向性的回声定位和多普勒效应, 从而会有多种变异和新的算法。更多的研究可以用于灵敏度分析, 分析算法的收敛性, 进一步提高收敛速度, 与更多的现有算法进行更严谨的测试比较, 将进一步揭示算法的优缺点。

参考文献

- [1] Goldberg D. Genetic Algorithms in Search, Optimization and Machine Learning, Reading[M]. Mass: Addison-Wesley, 1989
- [2] Dorigo M, Maniezzo V, Colnari A. The ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996, 26(1): 29-41
- [3] Kennedy J, Eberhart R. Particle swarm optimization[C]// Proceedings of IEEE International Conference on Neural Networks. 1995: 1942-1948
- [4] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]// Proceedings of the 6th International Symposium on Micro-Machine and Human Science. 1995: 39-43
- [5] Formato R A. Central force optimization: a new metaheuristic—With a applications in applied eletromagnetics[J]. Progress in Electromagnetics Research, 2007, 77: 425-491
- [6] 钱伟懿, 张桐桐. 自适应中心引力优化算法[J]. 计算机科学, 2012, 39(6): 207-209
- [7] Shor P W. Algorithms for quantum computation: discrete logarithms and factoring [C] // Proceedings of the 35th Annual Symp. on Foundations of Computer Science. New York, USA: IEEE Computer Society Press, 1994, 11: 124-134
- [8] Adleman L. Molecular computation of solutions to combinatorial problems[J]. Science, 1994, 266(5187): 1021-1024
- [9] Teodorovic D, Dell'Orco M. Bee colony optimization—a cooperative learning approach to complex transportation problems. Advanced OR and AI Methods in Transportation [C] // 10th EWGT Meeting and 16th Mini-EVRO Conference. 2005: 51-60
- [10] Bersini H, Varela F. The immune recruitment mechanism: A selective evolutionary strategy[C]// Proceedings of the 4th International Conference on Genetic Algorithms. 1991: 520-526
- [11] Yang X S. A new metaheuristic bat-inspired algorithm[C]// Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence 284. Berlin Heidelberg: Springer-Verlag, 2010: 65-74
- [12] 张树义, 赵辉华, 冯江, 等. 蝙蝠回声定位与捕食对策的研究[J]. 动物学杂志, 1999, 34(6): 47-50
- [13] Altringham J D. Bats, Biology and Behaviour[M]. Oxford University Press, 1996
- [14] 陈敏. 7 种蝙蝠回声定位行为生态研究[D]. 长春: 东北师范大学, 2003

[15] Chattopadhyay R. A study of test functions for optimization algorithms[J]. Opt. Theory Appl., 1971, 8: 231-236

[16] Schoen F. A wide class of test functions for global optimization [J]. Global Optimization, 1993, 3: 133-137

[17] Shang Y W, Qiu Y H. A note on the extended rosenbrock function[J]. Evolutionary Computation, 2006, 14: 119-126

[18] Shilane D, Martikainen J, Dudoit S, et al. A general framework for statistical performance comparison of evolutionary computa-

tion algorithms[J]. Information Sciences: an Int. Journal, 2008, 178: 2870-2879

[19] Deep K, Bansal J C. Mean particle swarm optimisation for function optimisation[J]. Int. J. Comput. Intel. Studies, 2009, 1: 72-92

[20] Eberhart R C, Shi Y H. Comparing inertia weights and constriction factors in particle swarm optimization[C]// Proceedings of the Congress on Evolutionary Computing. La Jolla, 2000: 84-89

(上接第 215 页)

表 3 是算法 4 所得到的全局最稳态结构的二面角解。表 4 给出的是算法 4 同 CSA 等算法所得的全局最稳态结构的精度对比。尽管这几种算法都能得到脑啡肽的全局最稳态结构,但是算法 4 所求的最优解同公认稳定结构之间的标准差为 0.606° ,最接近公认的脑啡肽稳定结构。图 5 所示的是算法 4 所得到的脑啡肽全局稳定结构和其它 5 个高质量的局部稳定结构的三维图。从图中可以直观地看出,尽管能量值相差很小,但 6 个三维结构图有很大的差别。图 5 中还给出了这 6 种高质量脑啡肽稳定结构与公认稳定结构相比较的 C_α -RMSD。如图 5 中所示,能量值为 -10.8748kcal/mol 所对应的脑啡肽结构尽管能量值比较大,但是其结构和公认稳定结构的误差为 0.471\AA 。由此表明,优化算法所得到的脑啡肽结构虽然其能量值较小,但是其三维结构和脑啡肽的全局最稳定结构可能有较大的误差。

表 3 算法 4 运行所得最稳定结构解

氨基酸	符号	二面角($^\circ$)	氨基酸	符号	二面角($^\circ$)
	ϕ_1	-83.499		ϕ_4	-136.847
	φ_1	155.793	苯丙	φ_4	19.097
酪氨酸	ω_1	-177.127	氨酸	ω_4	-174.09
Try ¹	χ_1	-173.177	Phe ⁴	χ_1	58.857
	χ_2	79.384		χ_2	-85.476
	χ_3	-166.329		ϕ_5	-163.446
	ϕ_2	-154.261		φ_5	160.899
甘氨酸	φ_2	85.817	甲硫	ω_5	-179.79
Gly ²	ω_2	168.503	氨酸	χ_1	52.863
	ϕ_3	82.918	Met ⁵	χ_2	175.293
甘氨酸	φ_3	-75.029		χ_3	-179.868
Gly ³	ω_3	-169.97		χ_4	-58.589

表 4 各算法所得最优解的精度对比

算法	CSA	1	2	3	4
方差(\AA)	0.672	0.625	0.618	0.628	0.606
标准差(\AA)	0.812	0.791	0.786	0.793	0.779

结束语 本文针对现在药物设计对蛋白质结构多模态的需求,提出了一种新的多模态混合算法 BECDE-SL,新算法可较好地解决多模态问题,能优化存在的无法有效地平衡收敛性和多样性的问题,特别是针对蛋白质结构预测这种高维非凸函数的优化。算法在 CrowdingDE-SL 算法的框架下,对种群个体进行能量极小化处理,同时集结过程和空间局部原理大大加快了算法的收敛速度,不同的交叉策略又增加了种群的多样性。仿真结果表明,算法 BECDE-SL 同其它算法相比,不仅具有较好的收敛特性,而且保持了种群的多样性,能够得到众多高质量的脑啡肽稳定结构。

参考文献

[1] 顾培亮. 系统分析与协调[M]. 天津:天津出版社,1998

[2] Bradley P, Misura K M S, Baker D. Toward high-resolution de novo structure prediction for small proteins[J]. Science, 2005, 309(5742): 1868-1871

[3] 许忠能. 生物信息学[M]. 北京:清华大学出版社,2008: 361-363

[4] Unger R, Moulton J. Genetic algorithms for protein folding simulations[J]. Journal of Molecular Biology, 1993, 231(1): 75-81

[5] Kalegari D H, Lopes H S. A differential evolution approach for protein structure optimization using a 2D off-lattice model[J]. International Journal of Bio-Inspired Computation, 2010, 2(3/4): 242-250

[6] Wong K C, Wu C H, Peng C B, et al. Evolutionary multimodal optimization using the principle of locality[J]. Information Sciences, 2012, 194: 138-170

[7] Jooyoung L, Harold A S, Shalom R. New optimization method for conformational energy calculations on polypeptides; conformational space annealing[J]. Journal of Computational Chemistry, 1997, 18(9): 1222-1232

[8] Klepeis J L, Pieja M J, Floudas C A. A new class of hybrid global optimization algorithms for peptide structure prediction; integrated hybrids [J]. Computer Physics Communications, 2003 (151): 121-140

[9] Wong K C, Leung K S, Wong M H. Protein structure prediction on a lattice model via multimodal optimization techniques[C]// GECCO'10 Genetic and Evolutionary Computation Conference. Portland, USA; ACM, 2010: 155-162

[10] Juyong L, Jinhyuk L, Takeshi N S, et al. De novo protein structure prediction by dynamic fragment assembly and conformational space annealing[J]. PROTEINS: Structure, Function, and Bioinformatics, 2011, 79(8): 2403-2417

[11] 来鲁华. 蛋白质结构预测与分子设计[M]. 北京:北京出版社, 1993

[12] Anfinsen C B. Principles that govern the folding of protein[J]. Science, 1973, 181(96): 223-230

[13] George N, Kenneth D G, Kathleen A P, et al. Energy parameters in polypeptides, 10. Improved geometrical parameters and non-bonded interactions for use in the ECEPP/3 algorithm, with application to proline-containing peptides[J]. Journal of Physical and Chemical, 1992, 96(15): 6472-6484

[14] Maranas C D, Floudas C A. Global minimum potential energy conformations of small molecules[J]. Journal of Global Optimization, 1994, 4(2): 135-170

[15] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997(11): 341-359

[16] Thomsen R. Multimodal optimization using crowding-based differential evolution[C]// CEC2004, IEEE Congress on Evolutionary Computation. Portland Marriott Downtown, USA; IEEE, 2004(2): 1382-1389