

自适应协同进化蝙蝠算法

刘 振[†], 鲁华杰, 刘文彪

(海军航空大学 岸防兵学院, 山东 烟台 264001)

摘要: 蝙蝠算法作为一种新型元启发式进化算法, 不可避免在进化过程中存在陷入局部极值的危险. 为了提高蝙蝠算法的进化性能, 提出一种自适应协同进化的蝙蝠算法 (ACEBA). 为保证算法具有良好的进化结构, 提出采用自适应进化种群结构, 使得种群结构能够依据种群多样性在集中式结构与分布式结构之间进行切换. 为协调实现主种群的勘探和子种群的开采, 引入优良个体解对速度和位置进行更新, 并在主种群和子种群内采用相适应的更新方式, 同时将原有固定参数推广到自适应变化, 并对蝙蝠行为的多普勒效应进行补偿. 最后对所提出的算法进行收敛性分析和仿真验证, 并与相关算法进行对比分析, 充分验证了算法的正确性和有效性.

关键词: 蝙蝠算法; 协同进化; 自适应; 搜索结构; 收敛性

中图分类号: TP18

文献标志码: A

Adaptive cooperation evolutionary bat algorithm

LIU Zhen[†], LU Hua-jie, LIU Wen-biao

(College of Coastal Defense Force, Naval Aeronautical University, Yantai 264001, China)

Abstract: The bat algorithm is a novel meta-heuristic nature-inspired algorithm, and also easy to trap into local optimum inevitably, therefore, the paper proposes an adaptive cooperation evolutionary bat algorithm (ACEBA). In order to ensure the proper framework for the algorithm, the evolutionary framework can be switched between the centralized and distributed framework according to the diversity judgment criteria in order to ensure the favorable evolutionary framework for the algorithm. In order to ensure the exploration ability of the main population and the exploitation ability of the sub-population, the position and velocity for the bat are updated, and the update way in main population is different from the sub-population. The compensation for Doppler effect in echoes is considered and the former fixed constant can change adaptively. Finally, the convergence of the algorithm is also deduced and verified by simulation results show the effectiveness and correctness of the proposed algorithm.

Keywords: bat algorithm; cooperation evolutionary; adaptive; search framework; convergence

0 引言

蝙蝠算法^[1-3]是基于蝙蝠超声定位的智能优化算法. 目前蝙蝠算法已经被广泛应用于各种约束和非约束优化问题, 如航迹规划^[4]、TSP问题^[5]、函数优化^[6]、视觉追踪^[7]. 相比传统进化算法, 蝙蝠算法具有较高的进化效率、较好的鲁棒性能, 其中文献[8]提出一种改进的蝙蝠算法用于优化神经网络模型, 蝙蝠个体保留当前最优解和最优路径, 并依据历史最优解、当前最优解和当前位置的多种线性组合更新速度, 同时在进化算法中采用分段线性混沌和Logistic混沌, 便于算法进行局部开采. 文献[9]提出一种改进的自适应蝙蝠算法, 引入了交叉和变异方式, 同时对参数进行了自适应设计. 文献[10]提出一种混合多

目标蛙跳蝙蝠算法, 将蛙跳算法与蝙蝠算法进行有效的融合, 并应用于不同电力负载下的电力单元布局和规模问题. 文献[11]在蝙蝠算法中融入了杂草算法与遗传算法, 利用遗传算法的变异和交叉对个体进行操作, 并依据轮盘赌方式选择个体. 通过对蝙蝠算法的研究和分析, 当前对其主要改进方式可概括为:

- 1) 引入局部寻优算子和搜索方式, 如将遗传进化中的交叉和变异^[9,11]引入蝙蝠算法中, 在速度和位置的更新过程中引入Levy飞行.
- 2) 与其他算法的有机融合, 引入蛙跳算法^[10]、遗传算法和杂草算法^[11]等优化算法.
- 3) 通过寻优结构和种群规模的调整, 利用多种群协同进化方式, 提高种群和个体之间的协同性能. 如

收稿日期: 2018-01-04; 修回日期: 2018-07-22.

基金项目: 国家自然科学基金项目(51605487).

责任编辑: 刘德荣.

[†]通讯作者. E-mail: hylz1008@126.com.

文献[12]提出利用两个种群协同分工进行勘探和开采的操作;文献[13]提出多种不同的种群协同结构,提高蝙蝠算法的进化能力。

由蝙蝠算法的基本特性和相关改进情况可以看出,蝙蝠算法作为一种新兴仿生进化算法,不可避免地会出现陷入局部极值,无法收敛到全局最优解等缺点。尽管蝙蝠算法已经有了诸多改进版本,并且算法性能也有了一定程度的提高,但自始至终采用单一的进化结构,往往不能满足进化需求,同时算法局部搜索能力也有待提高。因此,为提高蝙蝠算法的进化能力,本文提出一种进化框架结构可变的自适应协同进化蝙蝠算法(Adaptive cooperation evolutionary bat algorithm, ACEBA),依据种群多样性适时变换种群进化结构,在不同的种群内部采用了相应的速度和位置更新方式,并采用参数自适应方法,从全局和局部搜索方式提高蝙蝠算法的进化能力。

1 基本蝙蝠算法及其特性分析

在蝙蝠算法中,包含3个重要的因素,分别是搜索脉冲频率、脉冲音量和脉冲频度。在 d 维搜索空间中,第 i 只蝙蝠在 t 时刻的位置和速度分别为 x_i^t 和 v_i^t , $t + 1$ 时刻的位置和速度为

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{1}$$

$$v_i^{t+1} = v_i^t + (x_i^t - x^*)f_i, \tag{2}$$

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta. \tag{3}$$

其中: f_{\max} 、 f_{\min} 和 f_i 分别表示第 i 只蝙蝠在 t 时刻声波的最大频率、最小频率以及当前的频率; β 为 $[0, 1]$ 区间范围内的随机数; x^* 为当前的最优值。

从当前的最优解集中选定某个优良个体后,该蝙蝠个体位置的更新可以按照下式进行:

$$x_{\text{new}}(i) = x_{\text{old}} + \varepsilon A(t). \tag{4}$$

其中: x_{old} 表示从当前最优解集中选出的优良个体; $A(t)$ 表示 t 时刻所有蝙蝠个体的平均响度; ε 为 d 维随机变量, $\varepsilon \in [-1, 1]$ 。

随着迭代过程的进行,音量和脉冲频度会实时更新。通常在靠近猎物的过程中,音量会逐渐降低,脉冲发射的频度会逐渐增高, $A_i = 0$ 时意味着蝙蝠 i 刚刚找到一只猎物,暂时停止发音,更新方程可以按照下式进行:

$$A_i^{t+1} = \alpha A_i^t, \tag{5}$$

$$R_i^{t+1} = R_i^0 [1 - \exp(-\gamma t)]. \tag{6}$$

其中: $0 < \alpha < 1, \gamma > 0$, 且均为常数。可以看到,当 $t \rightarrow \infty$ 时, $A_i^t \rightarrow 0, R_i^t \rightarrow R_i^0$ 。

由当前蝙蝠算法的研究现状和基本特性可以看出,虽然蝙蝠算法已经有了较多的改进方法,但蝙蝠算法进化方式单一,进化性能有待提高,其存在的主要问题可以概述为:

1) 传统蝙蝠算法往往是单一种群进化,无法有效地进行勘探和开采的协调,当发现优良解集时,不能保证算法在进化过程中始终在优良解区域进行开采;

2) 缺少局部搜索方式,从而导致种群多样性降低,使得算法易陷入局部极值,响度和脉冲频率以一定概率决定了更新位置,但不可避免地受到局部极值吸引,导致缺乏深度开采能力;

3) 参数的更新方式往往是固定不变或者线性变化的,使得蝙蝠算法在搜索进化过程中缺乏灵活性,不能根据当前种群进化过程动态调整,无法保证种群具有良好的多样性。

由于基本蝙蝠算法存在诸多缺点,使得算法寻优性能不高,不可避免地容易陷入局部极值,为此本文提出一种自适应协同进化蝙蝠算法(ACEBA)。下面详细阐述算法流程。

2 自适应协同进化蝙蝠算法

2.1 搜索结构自适应变化

传统的蝙蝠算法,往往采用单一种群,进化方式单一,不能实现个体之间的协同进化。为了有效提高种群进化的多样性,增强算法的全局勘探和局部开采的能力,避免算法采用单一种群陷入局部极值问题,本文提出一种多种群协同的蝙蝠算法。将蝙蝠种群划分为主种群和子种群,其进化结构如图1所示。

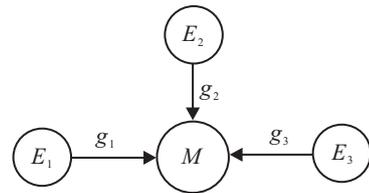


图1 集中式结构拓扑图

在图1的进化结构中,主种群为 M , 子种群分别设定为 E_1 、 E_2 和 E_3 , g_1 、 g_2 和 g_3 分别表示每个子种群的最优值,各子种群将最优个体与主种群进行交流。但是,如果进化过程中都采用同样的框架结构,即采用图1所示的集中式结构,虽然能保证全局勘探能力的提高,但无法有效地发挥局部开采能力,限制了算法的寻优性能,无法保证算法具备较高的进化效率。为有效提高进化过程中各种群的小生境进化能力,种群结构可在集中式结构与分布式结构之间适时进行切换,其中分布式结构如图2所示。

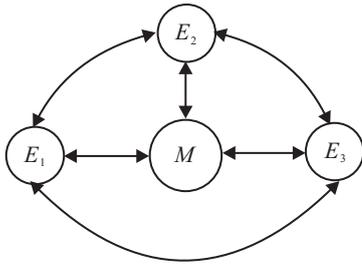


图2 分布式框架结构拓扑图

本文提出种群结构能够在图1所示的集中式结构与图2所示的分布式结构之间实现搜索框架的切换. 结构的自适应切换^[12]可以有效地实现勘探和开采性能的统一. 各个子种群分别进行独立进化, 有效地提高了主种群的全局勘探能力. 由于将每个子种群单独进行进化, 提高了进化种群的进化速度, 有效地加强了种群的开采能力. 以下详细阐述种群进化结构切换的依据和方法.

1) 集中式结构向分布式结构转换.

集中式进化结构向分布式进化框架切换的依据是将分布熵作为判断准则. 分布熵可以有效度量种群进化的多样性. 以往较多文献利用方差度量种群多样性, 与方差度量种群多样性进行对比发现, 当少数个体明显偏离群体多样性时, 方差不能有效地度量该多样性, 而其在多模态的情况下, 方差和种群多样性往往并不一致. 在协同进化框架结构中, 种群规模为 N , 主种群和3个子种群的个体数目分别为 $c_i (i = 0, 1, 2, 3)$, 则种群 P 此时的分布熵可以表示为

$$H(P) = - \sum_{i=0}^3 (c_i/N) \ln(c_i/N). \quad (7)$$

依据分布熵确定整个种群的分布程度^[14], 并依据如下两个准则实现集中式框架结构向分布式种群框架结构的切换:

i) 当分布熵 $H(P)$ 低于设定阈值 σ_1 时, 说明种群个体的多样性降低, 个体出现明显的集聚性, 因此将集中式框架结构变换为分布式框架结构;

ii) 当分布熵连续 τ 代, 其变化范围保持在 τ_2 以内, 此时说明种群进化性能连续 τ 代变化不大, 原有框架结构已经不能满足种群进化要求, 需要增加种群多样性, 因此需要将集中式框架结构变换为分布式框架结构.

2) 分布式结构向集中式结构转换.

在分布式寻优框架结构中, 利用方差判断种群多样性, 无法利用个体中的每一维数的信息, 不能保证个体基因位进化过程趋向于全局最优, 无法确保优良积木块的积累, 进而破坏优良模式的不断增长^[15]. 因

此, 引入多样性判断准则

$$\text{div}(P) = \frac{1}{NL} \sum_{i=1}^N \sqrt{\sum_{j=1}^N (x_{ij} - \bar{x}_j)^2}. \quad (8)$$

其中: L 为种群编码长度, \bar{x}_j 为第 j 维的平均值. 当 $\text{div}(P)$ 连续 μ 代, 其变化范围保持在阈值 σ_3 以内时, 原有框架结构已经不能满足种群进化要求, 将分布式框架结构变换为集中式框架结构.

2.2 引入优良解的速度和位置更新

为提高种群进化速度和全局勘探能力, 并保证种群多样性的提高, 在集中式还是分布式进化框架结构中, 可充分利用历史最优解和当前第 t 代最优解指导进化方向, 以提高种群进化方向, 便于主种群进行有效勘探^[16]. 本文利用历史最优解和当前第 t 代最优解的线性组合确定更新位置, 并设置不同的频率. 主种群 M 内的个体位置更新方式为

$$x_i^{t+1} = x_i^t + (x_g^t - x_i^t)f_1 + (x_b^t - x_i^t)f_2 + v_i'. \quad (9)$$

其中: X_g^t 为截止到第 t 代时的全局最优解, x_b^t 为第 t 代的最优解.

依据历史最优解和第 t 代最优解进行勘探寻优, 其中 f_1 和 f_2 更新过程如下所示:

$$f_1 = f_{\min} + (f_{\max} - f_{\min}) \left| \frac{x_g^t - x_i^t}{x_g^t - x_w^t} \right| \text{rand}_1, \quad (10)$$

$$f_2 = f_{\min} + (f_{\max} - f_{\min}) \left| \frac{x_b^t - x_i^t}{x_b^t - x_w^t} \right| \text{rand}_2. \quad (11)$$

在式(10)和(11)中, x_w^t 为第 t 代产生的最差解.

在子种群内, 为了充分进行开采操作, 利用第 t 代平均解值和历史最优解值, 并设置不同的权重系数, 子种群中速度更新方式为

$$v_i^{t+1} = v_i^t + (x_{\text{av}}^t - x_i^t)f_3w_1 + (x_g^t - x_i^t)f_3w_2. \quad (12)$$

其中: x_{av}^t 为第 t 代的平均位置值; w_1 和 w_2 为权重系数, $w_1 + w_2 = 1$.

2.3 参数自适应调整

在传统蝙蝠算法中, A_i^{t+1} 和 R_i^{t+1} 一般按照式(5)和(6)进行迭代循环更新, 其中参数 α 和 γ 是固定常数. 因此, 不能根据当前群体进化情况进行自适应更改, 导致进化尺度单调, 种群多样性不高. 为提高种群进化的多样性, 使得参数能够根据迭代次数动态更改, 本文采用自适应进化参数, 其中

$$\alpha^{t+1} = \left(\frac{1}{2t} \right)^{1/t} \alpha^t. \quad (13)$$

定义 $\delta^t = g_b^t - g_b^{t-1}$, 如果 $\delta^t < 0$, 则说明全局最优解性能下降, 需要加强局部搜索能力, 更新方式可变化为

$$\gamma^{t+1} = \begin{cases} 2\gamma^t, & \delta^t < 0; \\ \gamma^t, & \text{otherwise.} \end{cases} \quad (14)$$

2.4 对多普勒效应进行补偿修正

在基本蝙蝠算法中,往往没有考虑多普勒效应的影响,而在蝙蝠个体实际觅食过程中,这种现象是存在的.因此,为了有效地补偿多普勒效应对脉冲频率的影响^[7],需对蝙蝠位置和速度进行适应性更新.此时频率值不仅要考虑 f_{\max} 和 f_{\min} , 同时还需要考虑多普勒效应的影响,进行有效的补偿措施.依据多普勒公式

$$f^0 = \frac{v \pm v_0}{v \mp v_s} f^s, \quad (15)$$

当蝙蝠在捕食的过程中,不断靠近猎物,同时猎物会逃脱追捕,式(15)可变更为

$$f^0 = \frac{v + v_0}{v - v_s} f^s, \quad (16)$$

此时经过补偿修正后的频率值为

$$f' = \frac{c + v_g^t}{c + v_g^t} f^0 \left(1 + C \frac{(x_g^t - x_i^t)}{|x_g^t - x_i^t + \varepsilon|} \right). \quad (17)$$

利用式(17)可以分别对 f_1 、 f_2 和 f_3 进行补偿修正,当式(17)中的 f^0 分别替换为 f_1 、 f_2 和 f_3 时,即完成了对原有频率值的更新.其中: v_g^t 为第 t 代全局最优值时的速度值; c 为声音在空气中的传播速度, $c = 340 \text{ m/s}$; $C \in [0, 1]$ 为补偿因子,当 $C = 0$ 时,表示蝙蝠不能在回声定位中对多普勒效应进行补偿,当 $C = 1$ 时,表示能完全补偿.

依据以上的描述,可以对本文提出的自适应协同进化蝙蝠算法(ACEBA)流程进行描述.在算法运行初始阶段,初始化种群 $X(t)$, 设置算法的运行参数,包括种群规模 N , 最大循环迭代次数 T_{\max} , 脉冲频率范围 f_{\max} 和 f_{\min} , 音量的衰减系数 α , 以及 μ 、 τ 、 σ_1 、 σ_2 和 σ_3 等参数,则算法的主要进化流程可表述如下.

Step 1: 主种群设定为 M , 子种群分别设定为 E_1 、 E_2 和 E_3 , 各子种群分别独立进化,初始进化结构为集中式进化框架;

Step 2: 在集中式进化框架中,子种群将最优个体传递给主种群,主种群进行贪婪选择,在分布式进化框架中,子种群和主种群可以进行个体交流,并进行贪婪选择;

Step 3: 利用式(10)和(11)更新获得 f_1 和 f_2 , 并依据式(17)进行多普勒补偿修正;

Step 4: 利用式(9)获得主种群内个体的位置信息,利用式(12)并结合式(1)和(2)获得子种群内个体的位置信息;

Step 5: 产生随机数 rand , 如果 $\text{rand} > R_i^t$, 则接

受新产生的解,否则,当 $\text{rand} < A_i^t$ 并且 $f(x_i(t)) < f(x_b)$ 时,更新当前最优解,并依据式(5)、(6)、(13)和(14)更新 A_i^{t+1} 和 R_i^{t+1} ;

Step 6: 判断当前进化种群结构,若当前为集中式进化结构,则依据式(7),当分布熵低于设定阈值 σ_1 时,或者在连续 τ 代变化范围保持在阈值 σ_2 以内时,由集中式进化框架变换为分布式进化框架;

Step 7: 若当前为分布式进化结构,则依据式(8), $\text{div}(P)$ 在连续 μ 代,其分布熵的变化范围保持在 σ_3 阈值以内,由分布式进化框架变换为集中式进化框架;

Step 8: 判断是否满足结束条件,不满足则转 Step 2, 否则结束输出.

3 收敛性分析

对本文提出的自适应协同进化蝙蝠算法的收敛性能和进化稳定性进行分析.本文所提出的ACEBA,其核心进化过程可表示为

$$x_i^{t+1} = x_i^t + (x_g^t - x_i^t)f_1 + (x_b^t - x_i^t)f_2 + v_i^t,$$

其中

$$v_i^{t+1} = v_i^t + (x_{\text{av}}^t - x_i^t)f_3w_1 + (x_g^t - x_i^t)f_3w_2.$$

则可得到

$$\begin{aligned} &x(t+1) - (2 - f_1 - f_2 - f_3)x(t) + \\ &(1 - f_1 - f_2)x(t-1) = \\ &f_3(w_1x_{\text{av}}^t + w_2x_g^t) + f_1(x_g^t - w_2x_g^{t+1}) + f_2(x_b^t + x_b^{t-1}). \end{aligned} \quad (18)$$

式(18)为一个二阶差分方程,其中

$$f_3(w_1x_{\text{av}}^t + w_2x_g^t) + f_1(x_g^t - w_2x_g^{t+1}) + f_2(x_b^t + x_b^{t-1})$$

可视作控制系统的外部输入,并不影响系统的稳定性,因此在分析系统稳定性过程中可以忽略.

定义1 对于离散时间序列 $\{x(t), t = 0, 1, \dots\}$, $x(t+1)$ 收敛于最优解集 x^* , 当且仅当 $\lim_{t \rightarrow \infty} x(t) = x^*$.

对于式(18)而言, $\{x(t)\}$ 收敛于最优解 x^* 的充要条件是 $\lim_{t \rightarrow \infty} E(x(t)) = x^*$.

当前对进化算法稳定性的分析,大多将时变线性系统视作线性定常系统,即利用系数矩阵特征值根的模小于1,但该判据适用于线性定常系统,对线性时变系统并不一定适用,推理结果的可信性降低.本文利用控制系统中的Jury稳定判据判断当系统稳定时参数的限制条件.

引理1 (劳斯-胡尔维茨稳定判据) 若线性系统其对应的特征方程为

$$D(\lambda) = a_0 + a_1\lambda + \dots + a_{n-1}\lambda^{n-1} + a_n\lambda^n = 0,$$

其中 $a_n > 0$, 则该线性系统稳定的充要条件为: 在特

征方程中,各项系数为正且不缺项.

连续系统的劳斯-胡尔维茨稳定判据,是利用特征方程的系数和符号来判断系统的稳定性能,而在离散线性系统中,则不能直接套用,因为离散线性系统是以特征根是否在 z 平面的单位圆内进行判断的,即如果特征值 $|\lambda_i| < 1$,则判断该系统是稳定的,否则不稳定^[18].

引理2 (Jury 稳定判据) 设 n 阶离散系统特征方程可以表示为

$$D(\lambda) = a_0 + a_1\lambda + \dots + a_{n-1}\lambda^{n-1} + a_n\lambda^n = 0.$$

利用特征方程的系数,构造 $(2n-3)(n+1)$ 列朱利矩阵,其中矩阵中 $2k+2$ 行与 $2k+1$ 行互为反向排列,有

$$b_k = \begin{bmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{bmatrix}, k = 0, 1, \dots, n-1;$$

$$c_k = \begin{bmatrix} b_0 & b_{n-k-1} \\ b_{n-1} & b_k \end{bmatrix}, k = 0, 1, \dots, n-2;$$

$$d_k = \begin{bmatrix} c_0 & c_{n-k-2} \\ c_{n-2} & c_k \end{bmatrix}, k = 0, 1, \dots, n-3;$$

$$q_0 = \begin{bmatrix} p_0 & p_3 \\ p_3 & p_0 \end{bmatrix}, q_1 = \begin{bmatrix} p_0 & p_2 \\ p_3 & p_1 \end{bmatrix}, q_2 = \begin{bmatrix} p_0 & p_1 \\ p_3 & p_2 \end{bmatrix}.$$

该离散系统特征根全部位于 z 平面单位圆内,即系统稳定的充要条件为

$$D(1) > 0;$$

$$D(-1) \begin{cases} > 0, n \text{ 为偶数;} \\ < 0, n \text{ 为奇数;} \end{cases}$$

并且 $|a_0| < a_n, |b_0| > |b_{n-1}|, |c_0| > |c_{n-2}|, |d_0| > |d_{n-3}|, \dots, |q_0| > |q_2|$.也就是说,当以上条件都满足时,离散系统才是稳定的,否则系统是不稳定的.

定理1 自适应协同进化蝙蝠算法(ACEBA)稳定收敛的充分条件是

$$0 < f_1 + f_2 < 2, f_1 + f_2 < 2 - \frac{f_3}{2}.$$

证明 由式(18)的差分方程,对方程两边取数学期望后得到

$$\begin{aligned} E(x(t+1)) - (2 - f_1 - f_2 - f_3)E(x(t)) + \\ (1 - f_1 - f_2)E(x(t-1)) = Q, \end{aligned}$$

方程右边可以表示为

$$\begin{aligned} Q = E(f_3(w_1x_{av}^t + w_2x_g^t) + f_1(x_g^t - w_2x_g^{t+1}) + \\ f_2(x_b^t + x_b^{t-1})), \end{aligned}$$

则其特征方程为

$$\lambda^2 - (2 - f_1 - f_2 - f_3)\lambda + (1 - f_1 - f_2) = 0,$$

特征根可以表示为

$$\lambda_i = \frac{1}{2}(2 - f_1 - f_2 - f_3) \pm \frac{1}{2}[(2 - f_1 - f_2 - f_3)^2 - 4(2 - f_1 - f_2 - f_3)(1 - f_1 - f_2)]^{1/2}, i = 1, 2.$$

其通解为

$$E(x(t)) = c_1\lambda_1^t + c_2\lambda_2^t + K,$$

其中 c_1 和 c_2 为待定系数,由初值决定; K 为常数.

对于二阶线性定常离散系统而言,当 $n=2$ 时,判断其特征根的模是否在单位圆范围内,此时引理2对应的稳定充要条件可以简化为判断是否满足 $D(1) > 0, D(-1) > 0, |1 - f_1 - f_2| < 1$.由 f_1, f_2 和 f_3 的基本特性可知,均满足 $f_1 > 0, f_2 > 0$ 和 $f_3 > 0$,当满足上述条件后,经过推理可以得到 $0 < f_1 + f_2 < 2, f_1 + f_2 < 2 - f_3/2$. \square

4 仿真分析

针对本文提出的自适应协同进化蝙蝠算法(ACEBA),利用15个基准函数进行仿真分析.首先设置算法进化参数信息,种群规模 N 设置为100,最大循环迭代次数 T_{\max} 为100,脉冲频率范围 $f_{\max} = 2$ 和 $f_{\min} = 0$,音量的衰减系数 $\alpha = 0.9$,以及 $\mu = 5, \tau = 5, \sigma_1 = 0.0948, \sigma_2 = 0.001, \sigma_3 = 0.002$.

仿真所需计算机硬件配置为Intel(R) Core(TM)i3-3220处理器,内存2.00 GB, Window 7系统,仿真软件为Matlab R2009 a. 分别利用同类型算法和不同类型算法进行仿真对比,其中函数 F_9 最小值为-1,函数 F_{10} 最小值为-1.03,其余函数的最优值都为0.以下详细进行阐述.

$$F_1(x) = \sum_{i=1}^n x_i^2, n = 2, x_i \in [-5.12, 5.12];$$

$$F_2(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, \\ x_i \in [-2.048, 2.048];$$

$$F_3(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)], n = 30, \\ x_i \in [-5.12, 5.12];$$

$$F_4(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right), \\ n = 30, x_i \in [-600, 600];$$

$$F_5(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}} \sum_{i=1}^n x_i^2\right) - \\ \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e, \\ n = 30, x_i \in [-30, 30];$$

$$F_6(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2, x_i \in [-10, 10];$$

$$F_7(x) = 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}), n = 30,$$

$$x_i \in [-500, 500];$$

$$F_8(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} + 0.5,$$

$$x_i \in [-100, 100];$$

$$F_9(x) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2), x_i \in [-100, 100];$$

$$F_{10}(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_1^2)x_2^2, x_i \in [-3, 3];$$

$$F_{11}(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7, x_i \in [-100, 100];$$

$$F_{12}(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3, x_i \in [-100, 100];$$

$$F_{13}(x) = \sum_i^{20} ix_i^4 + N(0, 1), x_i \in [-1.28, 1.28];$$

$$F_{14}(x) = (x_1 - 1)^2 + \sum_{i=2}^{20} i(2x_i^2 - x_{i-1})^2, x_i \in [-10, 10];$$

$$F_{15}(x) = \sum_{i=1}^{19} \left(0.5 + \frac{\sin^2 \sqrt{100x_i^2 - x_{i+1}^2} - 0.5}{1 + 0.001(x_i^2 - 2x_ix_{i+1} + x_{i+1}^2)}\right), x_i \in [-100, 100].$$

4.1 同类型算法对比分析

将本文提出的ACEBA,利用函数 $F_1 \sim F_6$,分别与基本蝙蝠算法(BA)、文献[13]提出的多种群协同蝙蝠算法(BatRM-s)、文献[16]提出的定向蝙蝠算法(NBA)、文献[19]提出的二进制蝙蝠算法(BBA)进行仿真对比分析,将5种算法分别独立运行20次,并统计平均最优值(Best)、全局平均值(Mean)和标准差(Std),同时为了更充分分析本文所提出算法的优化效果,将ACEBA与对比算法进行单侧 t 检验,构造的检验统计量为

$$t = \frac{M_1 - M_2}{\sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}.$$

其中: m_1 和 M_2 分别表示本文所提出算法和待对比算法的平均值; n_1 和 n_2 表示样本数量,在文中可利用算法独立运行次数表示; S_1 和 S_2 表示本文所提出算法和待对比算法的方差.由于函数的 $F_1 \sim F_6$ 最优值均为0,给定右侧检验假设为 $H_0 : \mu_1 \geq \mu_2, H_1 :$

$\mu_1 < \mu_2, \mu_1$ 和 μ_2 分别是本文算法和对比算法的理论最优均值.给定显著性水平0.01,则拒绝域为 $W = \{t > t_{0.01}(n_1 + n_2 - 2)\}$,经查表后 $t_{0.01}(38) = 2.428$,统计结果如表1所示.

表1 同类型算法仿真对比结果

函数	算法	Best	Mean	Std	$t_{0.01}$
F_1	BA	1.35×10^{-3}	6.55×10^{-3}	3.44×10^{-3}	4.90
	BatRM-s	5.32×10^{-4}	2.13×10^{-3}	9.40×10^{-4}	7.18
	NBA	3.25×10^{-4}	5.09×10^{-3}	3.77×10^{-4}	9.36
	BBA	6.18×10^{-4}	8.64×10^{-3}	8.25×10^{-3}	4.55
	ACEBA	7.69×10^{-5}	4.78×10^{-4}	2.37×10^{-4}	—
F_2	BA	6.24×10^{-3}	8.13×10^{-2}	2.94×10^{-2}	5.27
	BatRM-s	7.24×10^{-4}	5.51×10^{-3}	6.28×10^{-3}	8.92
	NBA	4.57×10^{-4}	7.24×10^{-3}	6.76×10^{-3}	11.15
	BBA	8.17×10^{-3}	5.23×10^{-2}	4.56×10^{-2}	22.68
	ACEBA	3.01×10^{-4}	9.14×10^{-4}	6.13×10^{-3}	—
F_3	BA	7.12×10^{-4}	5.71×10^{-3}	5.69×10^{-3}	4.19
	BatRM-s	3.78×10^{-5}	2.89×10^{-4}	5.41×10^{-4}	6.83
	NBA	8.77×10^{-5}	4.56×10^{-4}	2.34×10^{-4}	9.72
	BBA	5.12×10^{-4}	9.15×10^{-3}	2.60×10^{-3}	4.39
	ACEBA	3.17×10^{-5}	2.76×10^{-4}	8.55×10^{-5}	—
F_4	BA	5.98×10^{-4}	3.50×10^{-3}	7.17×10^{-3}	3.78
	BatRM-s	6.16×10^{-4}	1.75×10^{-3}	3.98×10^{-3}	16.42
	NBA	3.67×10^{-4}	2.40×10^{-3}	4.23×10^{-3}	33.17
	BBA	7.27×10^{-4}	4.33×10^{-3}	8.55×10^{-3}	8.60
	ACEBA	1.49×10^{-4}	6.51×10^{-4}	2.67×10^{-4}	—
F_5	BA	8.43×10^{-4}	3.42×10^{-3}	6.61×10^{-3}	13.78
	BatRM-s	6.19×10^{-5}	7.61×10^{-4}	6.12×10^{-4}	7.03
	NBA	2.37×10^{-4}	1.16×10^{-3}	3.05×10^{-3}	9.15
	BBA	4.28×10^{-4}	3.81×10^{-3}	7.19×10^{-3}	3.43
	ACEBA	5.44×10^{-5}	7.10×10^{-4}	6.52×10^{-4}	—
F_6	BA	7.39×10^{-4}	6.87×10^{-3}	5.22×10^{-3}	2.88
	BatRM-s	4.19×10^{-4}	3.49×10^{-3}	5.48×10^{-3}	6.39
	NBA	3.63×10^{-4}	4.55×10^{-3}	3.98×10^{-3}	17.72
	BBA	9.42×10^{-4}	7.27×10^{-3}	6.73×10^{-3}	5.43
	ACEBA	2.59×10^{-4}	3.34×10^{-3}	8.29×10^{-4}	—

由表1的统计结果可以看出:本文提出的ACEBA在大部分性能指标上都能取得最优或者较优值,显示出ACEBA具有良好的进化性能;同时也发现,BBA编码具有局限性,寻优性能相比基本蝙蝠算法提升不明显. BatRM-s由于在协同进化框架上作了比较多的改进,采用多个种群协同进化,并设置了主种群和辅助种群的环形进化方式,从而有效地提高了种群个体的交流方式,在勘探和开采之间实现了有机协调,因而进化性能有所提高. NBA也采用了有利于全局勘探的随机操作和自适应性参数操作,从而提高

了蝙蝠算法的开采能力,而本文提出的ACEBA不仅在进化框架上进行自适应调整,确保全局勘探的正确性,同时也注重内部参数的自适应调整,提高开采品质,确保在整体上提高算法性能。

另外,本文还给出了统计检验结果,由表1可以看出,在显著性水平为0.01下,检验统计量 t 的计算结果均大于2.428,也就是说均落在拒绝域范围内,因此拒绝假设 $H_0: \mu_1 \geq \mu_2$,接受假设 $H_1: \mu_1 < \mu_2$,说明本文提出的算法平均值优于其他对比算法。

为了进一步对所提出的ACEBA进行对比分析,给出 F_3 函数的仿真对比图,如图3所示。

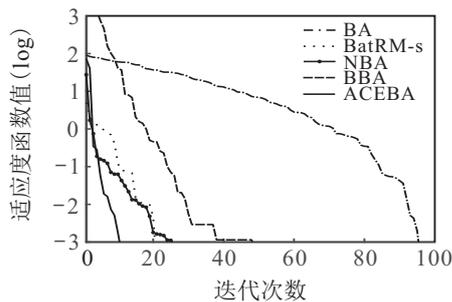


图3 同类算法对比仿真图

由图3可以看到:本文所提出的ACEBA兼顾了全局自适应框架结构和参数自适应,因而收敛效果比较平稳,NBA与BatRM-s也采用了相应的全局搜索和局部开采操作,都能较为顺利地寻找到较优解;同时也注意到,基本BA与BBA,搜索过程中方向性不强,往往需有较多的迭代次数才能找到最优解,因此需要针对问题的特性选择适合的优化方法。

4.2 不同类型算法对比分析

为了更充分地进行对比分析,将本文提出的ACEBA与混合遗传算法(CGA)^[20]、自调节粒子群算法(SRPSO)^[2]、增强型蚁群算法(RACO)^[22]、快速人工蜂群算法(FABC)^[23]进行对比,利用函数 $F_7 \sim F_{12}$,将5种算法分别独立运行20次,并统计平均最优值(Best)、全局平均值(Mean)和标准差(Std),统计结果如表2所示。

由表2的统计结果可看出:本文提出的ACEBA与几种改进智能进化算法相比也依然占有优势,在大部分的基准函数上都能取得较优的收敛结果,算法整体性能依然优越。由对比结果可以看出,CGA由于引入了细菌觅食算法中的趋药性,在多次进化过程中也能找到最优解,从 F_9 和 F_{10} 函数的统计结果可以清晰地看到这一点,但整体性能还是与本文算法有一定差距。FABC对跟随蜂实施多次开采操作,侧重于寻优速度上的提高,提高了算法每次执行的效率,各项寻优指标均劣于ACEBA。SRPSO在最优粒子中采用了

表2 不同类型算法仿真对比结果

函数	算法	Best	Mean	Std
F_7	CGA	2.87×10^{-3}	1.79×10^{-2}	8.26×10^{-3}
	SRPSO	7.09×10^{-4}	3.87×10^{-3}	3.41×10^{-4}
	RACO	4.72×10^{-4}	9.64×10^{-4}	3.60×10^{-4}
	FABC	4.33×10^{-4}	1.29×10^{-3}	3.42×10^{-4}
	ACEBA	4.16×10^{-4}	7.03×10^{-4}	2.87×10^{-4}
F_8	CGA	9.57×10^{-4}	4.26×10^{-3}	2.09×10^{-3}
	SRPSO	3.17×10^{-5}	3.47×10^{-4}	1.95×10^{-4}
	RACO	7.21×10^{-5}	1.65×10^{-4}	3.18×10^{-4}
	FABC	9.18×10^{-5}	5.79×10^{-4}	4.32×10^{-4}
	ACEBA	2.55×10^{-5}	7.33×10^{-4}	2.54×10^{-4}
F_9	CGA	-1	-0.86	8.73×10^{-2}
	SRPSO	-1	-0.92	6.17×10^{-2}
	RACO	-1	-0.93	5.59×10^{-2}
	FABC	-1	-0.91	6.78×10^{-2}
	ACEBA	-1	-0.93	4.19×10^{-2}
F_{10}	CGA	-1.03	-0.88	2.71×10^{-2}
	SRPSO	-1.03	-0.95	1.46×10^{-3}
	RACO	-1.03	-0.99	2.97×10^{-3}
	FABC	-1.03	-0.97	3.28×10^{-3}
	ACEBA	-1.03	-0.99	9.71×10^{-4}
F_{11}	CGA	4.93×10^{-4}	3.25×10^{-3}	1.24×10^{-3}
	SRPSO	3.24×10^{-5}	9.75×10^{-5}	6.31×10^{-4}
	RACO	2.69×10^{-5}	1.38×10^{-4}	4.81×10^{-4}
	FABC	3.77×10^{-5}	2.06×10^{-4}	1.74×10^{-4}
	ACEBA	1.90×10^{-5}	8.99×10^{-5}	1.56×10^{-4}
F_{12}	CGA	8.39×10^{-4}	8.51×10^{-3}	1.48×10^{-4}
	SRPSO	6.06×10^{-5}	4.56×10^{-4}	7.02×10^{-5}
	RACO	7.24×10^{-5}	7.49×10^{-4}	5.52×10^{-4}
	FABC	1.97×10^{-4}	9.03×10^{-4}	7.16×10^{-4}
	ACEBA	5.73×10^{-5}	4.50×10^{-4}	4.27×10^{-5}

自调节惯性权重,同时也在其他粒子中采用了搜索方向自我感知,较好地协调了勘探和开采操作。RACO则考虑了蚂蚁以前的遍历路径,并以此指导蚂蚁后续的路径选择和信息素更新,因而上述两种算法与本文算法寻优效果性能指标接近,甚至在函数 F_8 的平均值和标准差上还占有一定优势。从整体寻优效果上对比,本文算法仍然能取得最佳的寻优效果,主要是因为ACEBA不只是简单采用自我感知和调节,而是引入了自适应的框架结构和自适应参数,同时利用优良解指导进化方向,因而与SRPSO和RACO相比,提高了算法进化结构和进化方式的灵活性,有效地增强了算法进化能力。

进一步对所提出的ACEBA进行对比分析,给出 F_{11} 函数的迭代收敛图,如图4所示。

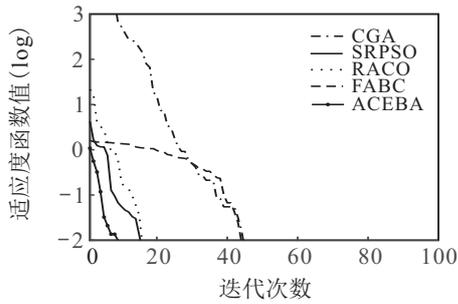


图4 不同类型算法对比仿真图

由图4的收敛迭代结果可以看出:CGA虽然引入了细菌觅食算法中的趋药特性,在一定程度上提高了算法进化程度,但受制于进化框架结构,寻优能力仍然有待提高;FABC侧重于挖掘跟随蜂的进化能力,在提高算法效率的同时,也在一定程度上增强了算法进化能力;SRPSO和RACO则具有较强的寻优能力,与CGA和FABC相比,能够快速向最优解趋近;同时也看到本文提出的ACEBA利用较少的迭代次数能迅速向最优解收敛,显示出本文算法设计时充分考虑了勘探和开采能力,表现出强大的进化能力。

为了更充分地进行对比分析,将本文所提出的ACEBA,利用函数 $F_{13} \sim F_{15}$ 与文献[24]中的已有研究成果进行对比,对比结果如表3所示。

表3 本文算法与已有研究成果对比分析

函数	算法	Best	Mean
F_{13}	CS	7.36×10^{-1}	2.73
	FFA	1.27×10^{-2}	6.21×10^{-2}
	DE	6.62	8.81
	PBIL	13.8	25.9
	LFPSO	6.78×10^{-3}	4.42×10^{-2}
	KH	1.37×10^{-2}	8.63×10^{-2}
	IKH	5.19×10^{-6}	3.96×10^{-4}
	ACEBA	4.87×10^{-6}	3.49×10^{-4}
F_{14}	CS	6.35×10^3	3.50×10^4
	FFA	2.22	17
	DE	2.49×10^3	7.86×10^3
	PBIL	8.23×10^4	2.91×10^5
	LFPSO	1.18	7.28
	KH	22.5	3.79×10^2
	IKH	6.67×10^{-1}	6.74×10^{-1}
	ACEBA	6.89×10^{-4}	4.27×10^{-3}
F_{15}	CS	2.7	4.25
	FFA	5.99	7.27
	DE	3.5	4.96
	PBIL	6.47	7.49
	LFPSO	4.62	9.23×10^{-1}
	KH	2.23×10^{-1}	1.18×10^2
	IKH	0	1.7
	ACEBA	3.51×10^{-3}	5.49×10^{-2}

由表3的对比结果可以看出,本文提出的ACEBA相比早期传统的低阶进化算法(如DE、PBIL等算法),具有极大的优势.低阶算法如果没有引入有利于全局勘探和局部开采的操作方法,则寻优性能极差,很难寻找到全局最优解.传统进化算法(如CS、KH等算法),如果没有考虑协同进化或者自适应结构,则容易陷入局部极值,而IKH采用了局部开采方法和精英策略,极大提高了算法的全局寻优能力.本文提出的ACEBA考虑了进化框架结构的自适应变化,因此寻优能力得到了极大的增强,与相关文献已有研究成果对比可以看出,在大部分函数性能指标上,都能取得最优的收敛效果。

5 结论

蝙蝠算法作为一种新兴的仿生智能进化算法,已经引起越来越多的关注.本文针对其进化过程的基本特性和相关特点,提出了一些改进策略,采用自适应的进化框架使参数能够自适应调整,从而更新了传统蝙蝠算法的进化过程,使得蝙蝠算法的进化过程更具智能性和组织性.经过仿真验证对比,并与同类型算法和不同类型智能进化算法对比分析,充分看到了本文算法的特性和优缺点,也反映了进化计算领域当前很难逾越的鸿沟——No Free Lunch定理,即很难做到进化算法的所有指标都达到最优.因此在提高算法的收敛精度和进化效率的同时,也必须注重降低算法的系统开销,使得算法能够具备普适性,这也是以后需要进一步研究的方向。

参考文献(References)

- [1] Yang X S. Nature-inspired metaheuristic algorithms[M]. Bristol: Luniver Press, 2008.
- [2] Yang X S. A new metaheuristic bat-inspired algorithm[C]. Studies of Nature-inspired Cooperative Strategies for Optimization (NISCO 2010). Berlin: Springer, 2010: 65-74.
- [3] Yang X S, Gandomi A H. Bat algorithm: A novel approach for global engineering optimization[J]. Engineering Computation, 2012, 29(5): 464-483.
- [4] Wang G G, Chu H C, Seyedali M. Three-dimensional path planning for UCAV using an improved bat algorithm[J]. Aerospace Science and Technology, 2016, 49(2): 231-238.
- [5] Osaba E, Yang X S, Diaz F, et al. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems[J]. Engineering Applications of Artificial Intelligence, 2016, 48(2): 59-71.
- [6] Selim Y, Ecir U K. A new modification approach on bat

- algorithm for solving optimization problems[J]. *Applied Soft Computing*, 2015, 28(3): 259-275.
- [7] Gao M L, Shen J, Yin L J, et al. A novel visual tracking method using bat algorithm[J]. *Neurocomputing*, 2016, 177(12): 612-619.
- [8] Najmeh S J, Salwani A, Abdul R H. Optimization of neural network model using modified bat-inspired algorithm[J]. *Applied Soft Computing*, 2015, 37(12): 71-86.
- [9] Mohammad H K, Taher N. A new intelligent online fuzzy tuning approach for multi-area load frequency control: Self adaptive modified bat algorithm[J]. *Electrical Power and Energy Systems*, 2015, 71(10): 254-261.
- [10] Chandrasekhar Y, Sydulu M, Sailaja K M. A multi-objective shuffled bat algorithm for optimal placement and sizing of multi distributed generations with different load models[J]. *Electrical Power and Energy Systems*, 2016, 79(7): 120-131.
- [11] Sakkayaphop P. A hybrid bat algorithm with natural-inspired algorithms for continuous optimization problem[J]. *Artif Life Robotics*, 2016, 21(1): 112-119.
- [12] Luo J, Liu L H, Wu X Y. A double-subpopulation variant of the bat algorithm[J]. *Applied Mathematics and Computation*, 2015, 263(7): 361-377.
- [13] Najmeh S J, Salwani A, Abdul R H. Multi-population cooperative bat algorithm-based optimization of artificial neural network model[J]. *Information Sciences*, 2015, 294(2): 628-644.
- [14] 辛斌, 陈杰, 窦丽华, 等. 群搜索优化中基于分布熵的多样性控制[J]. *模式识别与人工智能*, 2009, 22(3): 374-380.
(Xin B, Chen J, Dou L H, et al. Diversity control based on distribution entropy in population-based search and optimization[J]. *Pattern Recognition and Artificial Intelligence*, 2009, 22(3): 374-380.)
- [15] Ursem R K. Diversity guided evolutionary algorithms[C]. *Proc of the 7th Int Conf on Parallel Problem Solving from Nature*. Granada, 2002: 462-471.
- [16] Asma C, Rabia K, Mohamed B. New directional bat algorithm for continuous optimization problems[J]. *Expert Systems with Applications*, 2017, 69(3): 159-175.
- [17] Meng X B, Gao X Z, Liu Y, et al. A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization[J]. *Expert Systems with Applications*, 2015, 42(10): 6350-6364.
- [18] 金欣磊, 马龙华, 吴铁军, 等. 基于随机过程的PSO收敛性分析[J]. *自动化学报*, 2007, 33(12): 1263-1268.
(Jin X L, Ma L H, Wu T J, et al. Convergence analysis of the particle swarm optimization based on stochastic processes[J]. *Acta Automatica Sinica*, 2007, 33(12): 1263-1268.)
- [19] Seyedali M, Syed M M, Yang X S. Binary bat algorithm[J]. *Neural Computing & Application*, 2014, 25(5): 663-681.
- [20] Kedar N D, Rajashree M. Chemo-inspired genetic algorithm for function optimization[J]. *Applied Mathematics and Computation*, 2013, 220(5): 394-404.
- [21] Tanweer M R, Suresh S, Sundararajan N. Self-regulating particle swarm optimization algorithm[J]. *Information Sciences*, 2015(294): 182-202.
- [22] Rana F, Alireza M, Richard J, et al. Enriched ant colony optimization and its application in feature selection[J]. *Neurocomputing*, 2014, 142(10): 354-371.
- [23] Phuc N H, Chang W A. Fast artificial bee colony and its application to stereo correspondence[J]. *Expert Systems with Applications*, 2016, 45(1): 460-470.
- [24] Jensi R, Wiselin G. An improved krill herd algorithm with global exploration capability for solving numerical function optimization problems and its application to data clustering[J]. *Applied Soft Computing*, 2016, 46(4): 230-245.

作者简介

刘振(1983—), 男, 讲师, 博士, 从事智能推理与决策优化技术等研究, E-mail: hylz1008@126.com;

鲁华杰(1984—), 男, 讲师, 硕士, 从事智能火控技术的研究, E-mail: LUhuajie_yantai@163.com;

刘文彪(1980—), 男, 讲师, 博士, 从事指控技术及其在作战中的应用等研究, E-mail: lwb80314@sohu.com.

(责任编辑: 孙艺红)