

基于蚁群算法优化的布谷鸟搜索算法

张烈平,何佳洁,于滢琳,杨振宇,骆颖雄

(桂林理工大学 机械与控制工程学院,广西 桂林 541004)

摘 要: 针对标准布谷鸟搜索算法采用 Levy 飞行机制生成新的鸟巢,使得每次更新的鸟巢位置的随机性较大的问题,提出一种蚁群算法优化的布谷鸟搜索算法. 首先,提出的算法将待更新的鸟巢位置作为蚁群优化算法的一组初始解在极小的范围内进行搜索寻优. 之后,将此次蚁群优化算法搜索所得的解作为新的候选解. 当蚁群优化算法搜索寻优的候选解优于 Levy 飞行产生的候选解时,替换掉 Levy 飞行的候选解. 最后,再进行布谷鸟搜索算法择优算子,根据遗弃概率替换新的鸟巢位置,实现更新后的鸟巢位置更加趋向于最优解. 通过六个典型的测试函数将提出的算法与标准布谷鸟算法进行了寻优性能比较. 实验结果表明,提出的算法能够提升布谷鸟搜索算法候选解的质量,提高算法的收敛速度和收敛精度.

关键词: Levy 飞行;布谷鸟搜索算法;蚁群优化算法;鸟巢位置更新策略

中图分类号: TP301.6

文献标识码: A

文章编号: 1000—7180(2018)12—0021—06

A Cuckoo Search Algorithm Based on Ant Colony Algorithm Optimization

ZHANG Lie-ping, HE Jia-jie, YU Yan-lin, YANG Zhen-yu, LUO Ying-xiong

(College of Mechanical and Control Engineering, Guilin University of Technology, Guilin 541004, China)

Abstract: Aiming at the problem that standard cuckoo search algorithm using Levy flight mechanism to generate a new bird nest which makes the location of updated bird nest more randomly, a new cuckoo search algorithm based on ant colony algorithm optimization was proposed in this paper. Firstly, the location of the nest to be updated was used as a group of initial solutions of ant colony optimization algorithm to search and optimize in a minimal range in the proposed algorithm. After that, the solution optimized by the ant colony optimization algorithm was considered as a new candidate solution. And if the candidate solution optimized by the ant colony optimization algorithm was better than the candidate solution generated by Levy flight, the solution generated by Levy flight should be replaced with the candidate solution optimized by the ant colony optimization algorithm. At last, the optimal operator of the cuckoo search algorithm was used to replace the new nest location according to the abandoned probability, which could make the updated nest location more inclined to the optimal solution. The searching performance of proposed algorithm was compared with the standard cuckoo algorithm through six typical testing functions. Testing experimental results show that the proposed algorithm can improve the quality of the candidate solution of the cuckoo search algorithm and improve the convergence speed and the accuracy of the algorithm.

Key words: Levy flight, cuckoo search algorithm, ant colony algorithm, update strategy of bird nest location

1 引言

布谷鸟搜索 (Cuckoo Search, CS) 算法是由剑

桥大学 Xin She Yang 教授和 S. Deb 教授于 2009 年提出的一种新兴启发算法^[1]. 它通过模拟布谷鸟育雏行为和信天翁、果蝇等动物的 Levy 飞行模

收稿日期: 2018—02—20; 修回日期: 2018—03—26

基金项目: 国家自然科学基金(61741303); 广西自然科学基金 (2017GXNSFAA198161); 广西空间信息与测绘重点实验室基金(15-140-07-23, 16-380-25-23)

式,用以设计求解实际复杂优化问题.由于CS算法的模型简单、可调参数少、收敛速度快的特性,近年来被广泛应用于人工智能、工业调度、商业优化、神经网络等各个发展领域^[2].在标准CS算法的迭代过程中,布谷鸟的寻优探索过程和新的鸟巢的位置更新公式都采用的 Levy 随机飞行策略来实现的. Levy 飞行的随机跳跃性能能够提升算法的全局搜索能力.同时将 Levy 飞行应用在迁移算子的位置更新策略中来获取新的鸟巢位置能够跳出局部最优解,防止算法搜索过程过于早熟.同时,对鸟巢位置的随机更新也能增强算法的种群多样性.但另一方面,由于全局随机更新位置使得随着算法迭代过程的进行更新位置并不能朝着最优解方向步进,这样会导致算法的寻优过程收敛速度变得十分的缓慢,特别是在搜索到最优解位置附近时候会导致搜索效率非常低,有可能导致种群在最优解附近来回震荡、跳跃而无法收敛,甚至可能使得种群出现迷失现象. Luus R 等学者指出,一种算法无法在搜索的范围和搜索的结果之间同时达到最好的结果,一般将局部搜索算法混合进去可以有效的提升算法的性能^[3].基于这些因素,很多学者对CS算法提出了各种优化改进.例如: Amirhossein 等人用粒子群优化算法(Particle Swarm Optimization, PSO)较强的局部搜索能力集合布CS算法的全局搜索性能提出一种 PSO 和 CS 的混合算法^[4];江浩等提出的在寻优过程中引入单独的局部搜索策略后再进入下一次迭代^[5]. 聂慧等提出引入精英个体的概念并且对精英个体进行的局部搜索策略及对首领的寿命衰老机制^[6]. 李娜等人提出了一种基于 PSO 的 CS 算法,使得鸟窝的更新能像 PSO 算法中的粒子一样总是追寻当前最优与全局最优^[7].

蚁群优化(Ant Colony Optimization, ACO)算法是由意大利学者 M. Dorigo 等人在观察自然界中蚂蚁的觅食行为而提出的一种群体智能优化算法^[8]. 蚁群算法有着许多的优点,如较强的鲁棒性、良好的局部寻优性能和全局搜索性能、并行性计算、便于和其他优化方法相结合^[9]. 因此,以及针对标准CS算法的不足,本文提出一种基于蚁群算法优化的CS算法(Ant Clony Optmial Cuckoo Search, ACOCS). 算法采用 ACO 算法对CS算法的位置更新策略进行优化,使得候选解随着迭代朝着最优解方向进化,以解决CS算法的随机位置更新可能导致的算法搜索后期收敛速度慢、收敛精度不高等

问题.

2 标准布谷鸟搜索算法

标准CS算法是受到布谷鸟育雏行为的启发,通过引入鸟类的 Levy 飞行机制来模拟自然界中布谷鸟寻找鸟巢、产卵的过程,属于全局智能优化算法. Yang 和 Deb 提出该算法设定的以下三种理想状态^[10]:

(1)布谷鸟在每一代的繁衍过程中只下一枚鸟蛋,并且随机选择一个鸟窝来孵化.

(2)在找到更好的鸟窝之前,将当前最佳的鸟巢作为下一代的宿主进入下一次繁衍.

(3)每一代可供选择的总的鸟巢数量是一定的,布谷鸟下的鸟蛋被宿主发现的概率 P_a . 如果鸟蛋被宿主察觉,宿主将会放弃该鸟蛋,或者放弃整个巢穴,并寻找合适的位置建立新的鸟巢.

其中 P_a 表示宿主卵不是自己的孩子的概率, $P_a \in [0, 1]$, 大部分的研究者将 P_a 设定为 0.25^[11].

标准CS算法通过产生候选种群、择优选择和随机迁移三个基本过程实现布谷鸟需找鸟巢的仿真过程. 在上述三个理想状态的基础上,算法中布谷鸟飞行寻找线路和位置通过式(1)产生候选种群:

$$X_i^{t+1} = X_i^t + \alpha \oplus L(\beta) \quad (1)$$

式中, X_i^{t+1} , X_i^t 分别指第 i ($i = 1, 2, \dots, n$, n 为鸟巢数量) 个鸟巢位置在第 $t+1$ 代、第 t 代中位置向量 ($X_i^t = x_{i,1}^t, x_{i,2}^t, \dots, x_{i,j}^t, x_{i,d}^t$; d 为每个鸟巢的维数; j 代表任意第 j 维, $j = 1, 2, \dots, d$; t 为算法代数, $t = 1, 2, \dots, tc$, tc 为算法收敛时的代数); α 指步长因子,用于控制随机搜索的范围,是大于零的常数,对不同的情况,可取不同的值; $\oplus$ 指点对点相乘; $L(\beta)$ 是 Levy 飞行的随机寻优线路; $\alpha \oplus L(\beta)$ 是 Levy 飞行步长,表示从第 i 代的所有鸟窝(可行解)分别以 Levy 随机分布的方式搜寻到第 $i+1$ 代鸟窝(可行解)飞行的距离.

Levy 飞行的随机寻优线路与时间 t 的关系服从如式(2)所示的 Levy 概率分布,式中 λ 为幂次系数, μ 为正态分布的随机数. 从式(2)可以看出,CS 算法的寻优路径由频繁的短跳跃与偶然出现的长跳跃组成,这种寻优方式可以使算法拥有更大的搜索空间,更容易跳出局部最优.

$$\text{Levy}(\beta) \sim \mu = t^\beta, \quad (1 < \beta \leq 3) \quad (2)$$

为实现CS算法,采用文献[1]的方法来模拟 Levy(β) 飞行跳跃路径,即步长,如式(3)所示:

$$\text{Levy}(\beta) = \mu / |\nu|^{\frac{1}{\beta}} \quad (4)$$

式中, μ, v 都是正态分布的随机数; β 为参数偏度,一般取 $\beta = 1.5^{[12]}$.

在实际的优化问题中,鸟巢的位置 X_i 代表所有变量(d 维)的有效取值空间,而鸟巢的适用度值代表变量取不同的值所与对应的目标函数值.在进化过程中,通过式(1)对鸟巢位置更新后,用随机数 $r \in [0, 1]$ 与 P_a 对比,若 $r > P_a$ 则对 X_i^{t+1} 进行随机改变,反之不变.最后保留测试值较好的一组鸟窝位置 Y_i^{t+1} 此时仍记为 X_i^{t+1} .

在 CS 算法随机产生候选算子过程中,解的更新即生成新的鸟巢的策略同样是使用的 Levy 飞行机制.这样的随机游走策略中由于每次跳跃的方向和步长等不稳定因素,使得每次更新的鸟巢位置(解)的随机性大,这样算法在后期无法达到快速有效的收敛.解决的基本办法就是综合全局快速搜索和局部仔细搜索的优点,以实现高效的全局搜索和准确的局部搜索的平衡.

3 蚁群算法优化的布谷鸟搜索算法

3.1 优化思路及算法流程

提出的算法采用 ACO 算法对 CS 算法的位置更新策略进行优化,通过 ACO 算法的局部搜索能力提升 Levy 飞行产生的候选解的质量,使得候选解随着迭代朝着最优解方向进化.在 CS 算法每次迭代寻到最优解时候,即在 Levy 飞行获得待更新的鸟巢位置的时候,将待更新的鸟巢位置作为一组初始解引入 ACO 搜索算法中作为 ACO 算法的初始解,在极小的范围内进行一次 ACO 算法搜索,将此次搜索所得的解作为新的候选解.根据适应度与 Levy 飞行产生的候选鸟巢进行对比,再次执行择优算子.当 ACO 搜索的候选解优于 Levy 飞行产生的候选解时,替换掉 Levy 飞行的候选解,再进行 CS 算法择优算子和根据遗弃概率 P_a 替换新的鸟巢位置.通过这样选取候选解,使得解的质量有所提升,使更新后的鸟巢位置更加趋向最优解靠近,从而提升算法的搜索性能,从而提升 CS 算法的收敛速度和收敛精度.

为了保证融合后 ACOCS 算法的搜索性能,防止 ACO 算法搜索过程过长,同时又能有效的提升 Levy 飞行候选解的质量,本文提出公式(4)作为 ACO 算法在可行解范围内的搜索半径.同时将 CS 算法本次寻优的解集合作为 ACO 算法的初始解在以初始解为中心 d_i 为半径的搜索区域内进行寻优过程.

$$d_i = \frac{\|n_{\text{worst}} - n_{\text{best}}\|}{2} \quad (5)$$

式中, d_i 指的是第 i 代时 ACO 算法的搜索半径; n_{worst} 指的是当前最差鸟窝位置; n_{best} 为当前最佳鸟窝位置.

根据提出的改进思路,算法具体实现流程可以描述为如下:

步骤 1:设置参数和种群初始化.设置种群规模 popnum,步长因子 α ,发现概率 P_a ,最大迭代次数 T_{max} ,然后根据问题的属性初始化种群;

步骤 2:判断是否达到结束条件.如果达到结束条件,算法终止且输出当前最优解,如果未达到跳转步骤 3;

步骤 3:产生候选种群操作.按照公式(1)产生子代,若子代超出问题的约束范围则需采用适当的方法处理不可行解;

步骤 4:执行择优选择算子.针对每个个体,将候选解和当前解的适应度进行对比,保存适应度高的进入下一次迭代;

步骤 5:执行随机迁移算子.从步骤 4 更新后的候选种群中选择部分适应度较差的个体按照概率 P_a 进行变异;

步骤 6:对于步骤 5 中适应度差的鸟窝利用 Levy 飞行随机获得候选鸟窝;

步骤 7:根据公式(4)计算 ACO 算法的搜索半径 d_i ,利用 ACO 算法在候选鸟巢位置走遍搜索半径 d_i 的区域内进行局部搜索.并对候选鸟巢依据适应度进行择优,选取适应度优秀的鸟巢作为新的候选鸟巢;

步骤 8:再次执行择优选择算子.对于执行了变异操作的个体,比较其变异前和变异后的适应度,若变异后适应度更高则将变异后的个体纳入种群中,否则保留变异前的个体,跳转步骤 2.

3.2 仿真环境和参数设置

本文的仿真测试环境如下:Window 10 操作系统,Inter Core CPU T6400 2.10 GHz;2.00 GB 内存,MATLAB 2014b 仿真软件.测试算法为标准 CS 算法,以及本文提出的 ACOCS 算法,两种算法的初始鸟群数取 $N=25$,初始设定抛弃概率取 $P_a=0.25$,步长因子取 $\alpha=1$,算法结束条件为寻优次数 $T_{\text{max}}=3\ 000$ 或者搜索的最优解精度到达误差限制.采用如表 1 所示的 6 个常见测试函数进行测试,6 个常见测试函数的基本特征如表 2 所示.

表 1 6 个常见的测试函数

函数名	表达式
Ackley	$f_1(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{j=1}^n x_j^2}\right) - \exp\left[\frac{1}{n} \sum_{j=1}^n \cos(2\pi x_j)\right] + 20 + e$
Griewank	$f_2(x) = \frac{1}{4000} \sum_{i=0}^{n-1} x_i^2 - \sum_{i=0}^{n-1} x_i^2 \cos\left(\frac{x_i}{\sqrt{i+1}}\right) + 1$
Rastrigrin	$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$
Rosenbrock	$f_4(x) = \sum_{i=1}^n (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$
Schwefel	$f_5(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2$
Sphere	$f_6(x) = \sum_{i=1}^n X_i^2$

表 2 用于测试的 6 个函数基本特征

测试函数	搜索域 (i=1,2,...,n)	全局最小值	最优适应值
Ackley	$-15 \leq x_i \leq 30$	$x^* = (0, \dots, 0)$	$f(x^*) = 0$
Griewank	$-600 \leq x_i \leq 600$	$x^* = (0, \dots, 0)$	$f(x^*) = 0$
Rastrigrin	$-10 \leq x_i \leq 10$	$x^* = (1, \dots, 1)$	$f(x^*) = 0$
Rosenbrock	$-5.12 \leq x_i \leq 5.12$	$x^* = (0, \dots, 0)$	$f(x^*) = 0$
Schwefel	$-5.12 \leq x_i \leq 5.12$	$x^* = (0, \dots, 0)$	$f(x^*) = 0$
Sphere	$-10 \leq x_i \leq 10$	$x^* = (0, \dots, 0)$	$f(x^*) = 0$

3.3 实验仿真及结果分析

以维数为 5 和 15 寻优结果的最小值、最大值、平均值以及方差作为寻优过程的性能评判指标,对 CS、ACOCs 算法性能进行比较. 仿真结果如表 1 所示,表中各性能指标值是算法单独运行 30 次的平均值. 图 1~图 6 是二种算法在维数为 5 的时候的六个测试函数相应的收敛曲线.

表 3 二种算法寻优结果对比

测试函数	维数	算法	最小值	最大值	平均值	迭代次数	成功率/%
Ackley	5	CS	7.245 1e-8	1.000 2e-6	9.776 8e-7	492	100
		ACOCs	3.201 8e-9	9.121 3e-7	9.193 6e-8	261	100
	15	CS	2.242 5e-8	9.178 4e-7	5.190 8e-7	850	98
		ACOCs	8.351 2e-13	0.313 1e-6	9.507 5e-9	371	100
Griewank	5	CS	9.087 3e-19	9.031 6e-16	6.040 7e-17	564	100
		ACOCs	0	0	0	265	100
	15	CS	-8.065 3e-17	5.033 3e-13	3.019 1e-16	1 205	100
		ACOCs	0	0	0	547	100
Rastrigrin	5	CS	-2.110 9e-22	3.225 0e-16	9.204 8e-19	624	100
		ACOCs	-1.156 3e-24	8.161 6e-19	3.323 1e-23	215	100
	15	CS	1.170 4e-15	2.690 7e-14	8.007 1e-15	1 149	99
		ACOCs	8.690 7e-18	4.279 3e-12	1.487 3e-16	578	100
Rosenbrock	5	CS	4.095 0e-19	3.923 2e-16	2.562 0e-18	751	100
		ACOCs	0	0	0	397	100
	15	CS	3.424 6e-12	9.728 6e-9	1.271 2e-10	1 615	99
		ACOCs	0	0	0	802	100
Schwefel	5	CS	4.013 9e-18	9.028 6e-16	1.603 3e-17	672.2 30	100
		ACOCs	2.005 5e-20	7.011 7e-17	1.609 2e-19	433.021 2	100
	15	CS	-5.063 8e-15	9.061 4e-12	3.137 0e-14	1 574.210	100
		ACOCs	7.833 8e-18	8.187 8e-13	0.853 9e-14	1 124.301	100
Sphere	5	CS	6.338 3e-33	9.941 6e-29	3.611 9e-31	1 057.123	100
		ACOCs	1.086 0e-37	7.269 0e-31	1.520 9e-35	626.390	100
	15	CS	6.253 4e-14	9.157 4e-13	2.240 7e-13	2 328.900	99
		ACOCs	2.021 7e-16	7.003 5e-14	0.179 1e-14	1 681.621	100

从表 3 中二种算法的最小值、最大值、平均值以及方差来看,不管维数是 5 还是 15,ACOCs 算法的

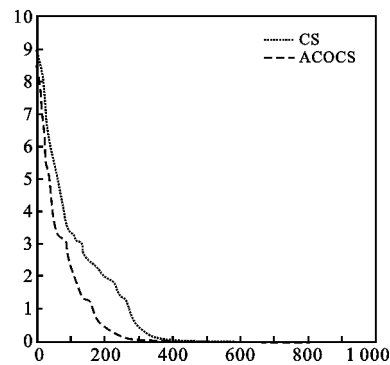


图 1 Ackley 函数的收敛性曲线

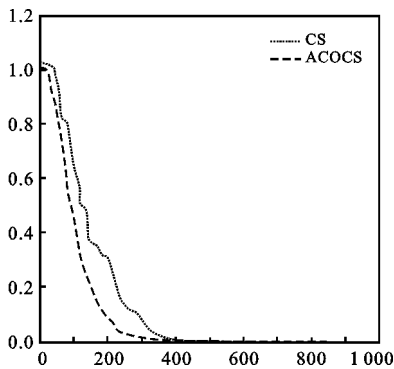


图 2 Griewank 函数的收敛性曲线

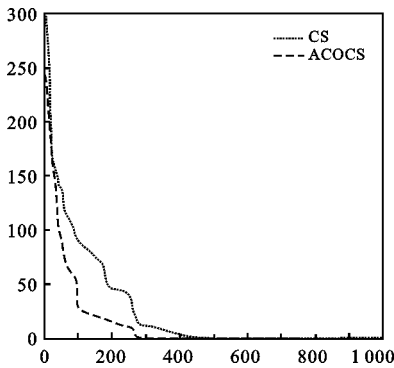


图 3 Rastrigrin 函数的收敛性曲线

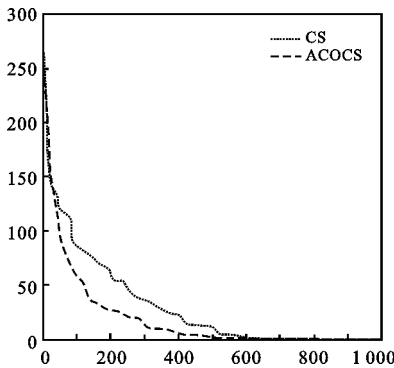


图 4 Rosenbrock 函数的收敛性曲线

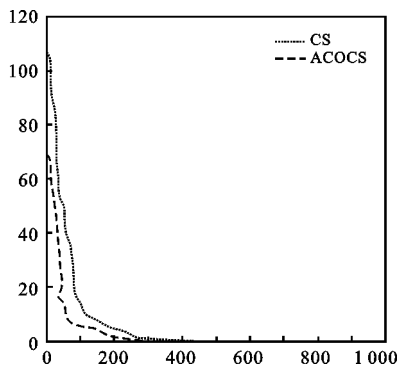


图 5 Schwefel 函数的收敛性曲线

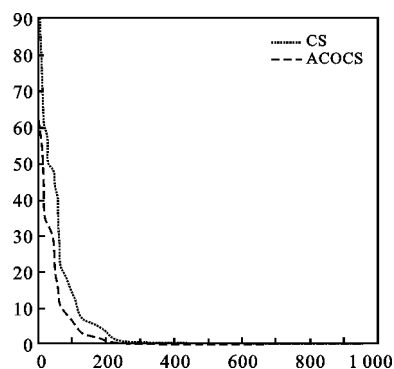


图 6 Sphere 函数的收敛性曲线

所有寻优数据均要由于 CS 算法的寻优数据. 同时, 还可以看出, 在维数 $N=5$ 时, CS 算法、ACOCs 算法两者在次数结束条件下都能够完成收敛, 但 ACOCS 算法的收敛速度比 CS 算法的快, 最优解的质量也比 CS 算法好. 而在维数 $N=15$ 时, CS 算法除了在 Griewank 和 Schwefel 测试函数的时候能够完全收敛之外, 针对其余 4 个测试函数都无法达到完全的收敛, 而 ACOCS 算法每个函数都能完全收敛. 因此, ACOCS 算法相对于 CS 算法能够更快的找到最优解且解的质量也更好, 尤其是对于复杂函数, ACOCS 的搜索性能明显优于 CS 算法.

4 结束语

针对 CS 算法位置更新策略依赖随机性产生的候选解的质量优劣的问题, 给出一种基于 ACO 优化的 CS 算法. 提出的算法在 CS 算法产生候选解替换进化之前用局部搜索能力强的 ACO 算法对新的候选解进行再次寻优, 使得候选解朝着最优解的方向前进. 通过 6 个测试函数的仿真实验验证了该算法产生的候选解质量高于 CS 算法, 可以提升 CS 算法的收敛速度和收敛精度.

参考文献:

[1] Yang X S, Deb S. Cuckoo search via levy flights[C]// Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC'09). India, Coimbatore, 2009: 210-214.

[2] Mohammad Shehab, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar. A survey on applications and variants of the cuckoo search algorithm[J]. Applied Soft Computing, 2017(61): 1041-1059.

[3] Luus R, Jaakola T H I. Optimization by direct search and systematic reduction in the size of search region [J]. Aiche Journal, 1973, 19(4):760-766.

[4] Ghodrati A, Lotfi S. A hybrid CS/PSO algorithm for global optimization[C]// Proceedings of the 4th Asian conference on Intelligent Information and Database Systems -Volume Part III. Berlin, Springer-Verlag, 2012:89-98.

[5] 江浩, 阮奇, 念美玲, 等. 改进的布谷鸟搜索算法及其在多效蒸发优化中的应用[J]. 计算机与应用化学, 2014, 31(11):1363-1368.

[6] 聂慧, 刘波, 韦向远, 等. 求解资源受限项目调度问题的改进布谷鸟搜索算法[J]. 桂林理工大学学报, 2013, 33(33):529-536.

[7] 李娜, 贺兴时. 基于粒子群算法的布谷鸟搜索算法[J]. 纺织高校基础科学学报, 2014, 27(3):374-379.

[8] He Y, Chen D, Zhao W. Ensemble classifier system

based on ant colony algorithm and its application in chemical pattern classification [J]. Chemometrics & Intelligent Laboratory Systems, 2006, 80(1):39-49.

[9] Söbester A, Forrester A I J. Ant colony optimization [J]. Alphascript Publishing, 2010, 28(3):1155-1173.

[10] Yang X S, Deb S. Cuckoo Search: recent advances and applications[J]. Neural Computing and Applications, 2014, 24(1):169-174.

[11] Naik M, Nath M R, Wunnava A, et al. A new adaptive cuckoo search algorithm[C]//2015 IEEE 2nd International Conference on Recent Trends in Information Systems. Kolkata, India, 2015:1-5.

[12] Vazquez R A. Training spiking neural models using cuckoo search algorithm[J]. Evolutionary Computation, 2011, 30(1):679-686.

作者简介:

张烈平 男, (1971-), 博士, 教授. 研究方向为无线传感器网络、系统优化与调度. E-mail: 25761108@qq. com.

何佳洁 女, (1992-), 硕士研究生. 研究方向为检测技术与自动化装置.

于滢琳 女, (1993-), 硕士研究生. 研究方向为检测技术及自动化装置.

杨振宇 男, (1993-), 硕士研究生. 研究方向为控制工程.

骆颖雄 男, (1990-), 硕士. 研究方向为计算机应用.

(上接第 20 页)

[7] 安天实验室. 修改硬盘固件的木马: 探索方程式组织的攻击组件[EB/OL]. (2015-03-13). [2017-12-01]. <https://www.secpulse.com/archives/5326.html>.

[8] Kaspersky Lab. Equation: the death star of malware galaxy [EB/OL]. (2015-02-16). [2017-12-01]. <https://securelist.com/equation-the-death-star-of-malware-galaxy/68750/>.

[9] ENC Lab. The open SSD project[EB/OL]. (2017-06-05). [2017-12-01]. http://www.openssd-project.org/wiki/The_OpenSSD_Project

作者简介:

刘政林 男, (1968-), 博士, 教授. 研究方向为微电子学与固体电子学.

陈小强(通讯作者) 男, (1993-), 硕士研究生. 研究方向为微电子学与固体电子学. E-mail: chenxiaoqiangmail@qq. com.

刘文超 男, (1983-), 博士研究生. 研究方向为微电子学与固体电子学.

郑怡恺 女, (1994-), 硕士研究生. 研究方向为微电子学与固体电子学.

邹雪城 男, (1964-), 博士, 教授. 研究方向为微电子学与固体电子学.