计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190400036

分段加权布谷鸟算法及其应用

臧 睿 刘笑笑
东北林业大学数学系 哈尔滨 150040
(1452680854@qq.com)

摘 要 为解决布谷鸟局部搜索与全局搜索的协调问题,提高后期收敛速度,对算法搜索进行分段处理,通过引入一种动态自适应步长控制量以及相应的分段加权位置更新公式,提出一类改进的布谷鸟算法。选取12个经典约束优化问题和部分结构优化设计问题对改进算法进行验证。研究结果表明,相对于其他算法,该算法对以上大部分问题具有较好的运算结果。

关键词:布谷鸟算法;动态自适应步长控制量;结构优化设计;罚函数法
中图法分类号 O22

Segment Weighted Cuckoo Algorithm and Its Application

ZANG Rui and LIU Xiao-xiao
Department of Mathematics, Northeast Forestry University, Harbin 150040, China

Abstract In order to solve the coordination problem between cuckoo local search and global search, improve the convergence speed in the later stage, and search the segmentation process of the algorithm, an improved cuckoo algorithm is proposed by introducing a dynamic adaptive step control variable and the corresponding segment weighted position update formula. The improved algorithm is verified by selecting 12 classical constrained optimization problems and some structural optimization design problems. The research results show that, compared with other algorithms, this algorithm is more efficient for most of the above problems.

Keywords Cuckoo algorithm, Dynamic adaptive step control variable, Structural optimization design, Penalty function method

1 引言

2009年,Yang等[1-4]首先提出了布谷鸟搜索(Cuckoo Search,CS)算法。自从CS算法被提出以来,许多学者针对求解精度和收敛速度以及目标选择等问题对该算法进行了改进。也有学者将CS算法与遗传算法相结合,在某种程度上弥补了该算法的不足。2010年,Yang和Suash在CS算法的生物学原理和数学模型的建立上进行了进一步的研究和探索,并运用基准测试函数验证了CS算法,将其应用于工程学优化问题,得到了较好的优化效果。2012年,Wang等[5]为完善布谷鸟搜索算法的收敛性理论,建立了CS算法的Markov链模型,分析该Markov链的有限齐次性;在此基础上通过分析鸟窝位置的群体状态转移过程,指出随机序列将进入最优状态集;同时证明了CS算法满足随机搜索算法全局收敛的两个条件。2015年,Yang等[6]针对多目标优化问题提出了一种改进的CS算法。CS算法具有简单高效、计算时间短、求解多样性更强、容易调试等优点,但是仍然存在很多问题,比如求解精度不高,在求解全局最优值以及局部最优值时候容易陷入过稠等问题。为提升算法性能,本文对基本CS算法进行了进一步的探讨和分析,引入了动态自适应步长控制量并改进了位置更新公式。针对约束优化问题,结合罚函数法给出相应的约束优化算法。研究结果表明,该算法对若干经典的非线性规划问题有较好的效果。

2 布谷鸟算法及其改进

2.1 布谷鸟算法

在自然界中,布谷鸟寻找产卵的鸟窝位置是随机或者类随机的,为了模拟这种行为模式,Yang提出了一个数学模型,可以用以下3点理想化条件简单概括。

1)每只布谷鸟随机选择鸟窝产蛋,并且每次只产一个蛋。

2)最好的鸟窝被保留到下一代。

3)可以利用的巢主鸟窝数量 n 是固定的,外来鸟蛋被巢主鸟发现的概率是 p_a,其中 $p_a\in[0,1]$。在这种情况下,巢主鸟或者将布谷鸟的鸟蛋丢弃,或者干脆抛弃这个鸟窝,在一个新的位置建立一个全新的鸟窝。

基于上述3条规则,布谷鸟位置更新和寻窝路径如下:

$$x_i^{t+1}=x_i^t+\alpha\oplus L(\lambda),i=1,2,\cdots,n \tag{1}$$

其中,x_i^t 为第 i 个鸟窝在第 t 代的鸟窝位置;α 为步长比例因子,一般取为0.1;$\oplus$为点乘;$L(\lambda)$为随机飞行步长,服从列维分布;n 为鸟巢数量。

布谷鸟搜索算法的基本流程描述如算法1所示。

算法1 布谷鸟搜索算法

目标函数:$f(x),x=(x_1,x_2,\cdots,x_d)^T$;初始化一个具有n个鸟窝的种群 $x_i(i=1,2,\cdots,n)$;
while(t<最大迭代次数)
通过列维飞行随机地得到一个鸟窝i
评估它的质量/适应度 F_i

基金项目:中央高校基本科研业务费专项资金(DL09BB40)
This work was supported by the Fundamental Research Found of the Central Universities (DLO9BB40).
通信作者:刘笑笑(728629126@qq.com)

从 n 个鸟窝中随机选择一个鸟窝 j
if($F_i<F_j$)
then
用新的解替换 j
end
抛弃比较差的鸟窝的 p_a(精度),并通过列维飞行建立新的鸟窝;
保留最优解
排列所有解并找出当前最优解
end while

2.2 改进的布谷鸟算法

在标准的 CS 算法中,α 为定值,α 比较小时不利于全局搜索,比较大时不利于局部搜索。对此,Yang 等[6]在求解多目标优化问题时提出一种 α 动态自适应变化方法。本文采用了类似的 α 动态自适应调整方法,定义如下:

$$\alpha^{t+1}=\alpha^t\cdot 10^{K(r-T)} \tag{2}$$

其中,K 为变化率,r 为新解中改变的比值,T 为阈值,K 的取值为 10,变化率 T 的取值则需要根据 Rechenberg[7]提出的参数自动调整的 1/5 原则,本文中取值 0.2。r 的计算方法为新解中相比原个体的最优解变小的个体总数除以种群大小。本文种群大小取值 25。当 $r<T$ 时,$K(r-T)$ 的值为负数,$10^{K(r-T)}$ 的值小于 1,α 减小,当 $r>T$ 时,$K(r-T)$ 的值为正数,$10^{K(r-T)}$ 的值大于 1,α 增大。

在标准的 CS 算法中,布谷鸟的飞行路线是随机的,在后期会导致收敛速度慢的问题。Cai 等[8]在改进蝙蝠算法时使用了三角翻转策略更新位置,扩大了搜索范围。如图 1 所示,向量$\overrightarrow{x_i^t x_{\text{worst}}^t}$转到向量$\overrightarrow{x_i^t x_{\text{mid}}^t}$时,点$\vec{x}_{\text{worst}}^t$转到点$\vec{x}_{\text{mid}}^t$,这个过程提高了局部搜索性能,这是因为 $f(\vec{x}_{\text{mid}}^t)<f(\vec{x}_{\text{worst}}^t)$。但将该策略应用于全局搜索时,效果较差。基于此,本文提出一类位置更新公式,当迭代次数较少时,应用局部搜索策略加快收敛速度,迭代次数较多时,应用全局搜索策略,以扩大搜索范围。

$$x_i^{t+1}=\begin{cases}x_i^t+\dfrac{1-\beta}{\beta}(x_{\text{mid}}^t-x_{\text{worst}}^t)+\alpha^t\cdot 10^{(K(r-T))}\oplus L(\lambda), & t\leqslant t_0\\ x_i^t+(x_{worst}^t-x_i^t)\cdot\beta+\alpha^t\cdot 10^{(K(r-T))}\oplus L(\lambda), & t>t_0\end{cases} \tag{3}$$

其中,$i=1,2,\cdots,n$,t 为迭代次数,t_0 为取定的正整数,β 为[0,1]之间的随机数。

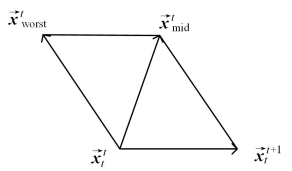

图 1 PNCS 算法位置更新图示

Fig. 1 PNCS algorithm position update diagram

3 改进的布谷鸟算法与罚函数法的结合

3.1 罚函数法

罚函数法是一种用来求解约束问题的间接解法。它的基本思想是利用问题中的约束函数做出适当的罚函数,从而将一个有约束的优化问题转化为一系列无约束的优化问题进行求解[9]。

对于约束优化问题:

$$\begin{aligned}&\min F(x)\\ &\text{s.t. } g_j(x)\leqslant 0, j=1,2,\cdots,p\\ &\quad h_j(x)=0, j=p+1,\cdots,m, l_i\leqslant x_i\leqslant u_i, i=1,2,\cdots,n\end{aligned} \tag{4}$$

其中,$f(x)$是目标函数,$g_j(x)$和 $h_j(x)$是不等式约束函数和等式约束函数,l_i 和 u_i 是决策变量 x_i 的上下界。

对于问题(4),定义目标函数:

$$\Phi(x,r^{(k)})=f(x)+r^{(k)}\{\sum_{u=1}^{p}G[g_u(x)]+\sum_{u=p+1}^{m}H[h_u(x)]\} \tag{5}$$

其中,$G[g_u(x)]$为不等式约束惩罚项,$H[h_u(x)]$为等式约束惩罚项,$r^{(k)}$为罚因子。

3.2 含罚函数项的布谷鸟算法

本文提出适应于一般约束优化的含罚函数项的布谷鸟算法(PNCS)。具体步骤如下:

步骤 1 初始化参数。

步骤 2 随机产生 n 个鸟窝,按照目标函数测试最优值,并保留最优值。

步骤 3 利用位置更新公式进行更新位置,按 $\Phi(x,r^{(k)})$ 对新位置进行测试,并保留最优位置到下一代。

步骤 4 随机产生一个均匀的数 $r\in(0,1)$,并与鸟蛋被发现的概率 p_a 进行比较。如果 $r>p_a$,则利用列维飞行公式对位置 x_i^{t+1} 进行随机改变,反之 x_i^{t+1} 不变。再对新的鸟窝位置进行测试,保留最优的位置到下一代。

步骤 5 判断最优位置是否满足最优解条件,如果满足则停机输出结果,否则转到步骤 6。

步骤 6 迭代次数超过预设最大迭代次数时停机,否则返回步骤 2。

当惩罚因子较小时,输出的最优解往往不满足约束条件。本文采用 Bao[10]提出的罚因子选取方法,将惩罚因子取为 $r^{(k)}=\dfrac{1}{M}$,$M=M\times\gamma$,$\gamma\in(0,1)$决定 M 的增长速度。

基本布谷鸟算法的步骤如算法 2 所示。

算法 2 基本布谷鸟算法

输入:阈值,变化率,鸟窝数量等数据

```
1. for j=1:25
      if fobj(new_nest(j,:))<fmin
        an=an+1
      end
    end
  r=a_n/25
  b=b*^(k*(r-T))
2. if(N_iter<15000)
    s=s+(m-x)*((1-lambda)/lambda)+stepsize.*randn(size(s))
  else
  s=s+(x-s)*lambda+stepsize.*randn(size(s))
```

过程 1 为式(2)的算法代码,过程 2 为式(3)的算法代码。

4 仿真实验与数值分析

为验证 PNCS 算法的有效性,选取 7 个标准测试函数[11]和 5 个结构优化设计问题[12]进行实验,并与其他算法[13-18]进行比较。由于等式约束集内部一般为空集,迭代点极易偏离约束集,以致在惩罚因子极大的情况下仍不能得到可行解。本文针对以下问题中的等式约束,采用降维的方式对迭代点加以限制,解决了以上问题。

PNCS算法的参数设置为 $N=25$，$P_a=0.25$，$K=10$，$T=0.2$。每个问题均独立运行30次。

4.1　标准测试函数

Problem1：

$$\min f(x)=(x_1-10)^3+(x_2-20)^3$$

$$\text{s.t. } g_1(x)=-(x_1-5)^2-(x_2-5)^2+100\leqslant 0$$

$$g_2(x)=(x_1-6)^2+(x_2-5)^2+82.81\leqslant 0$$

$$13\leqslant x_1\leqslant 100,0\leqslant x_2\leqslant 100$$

从表1中可以看出，对于Problem1，与HM[15]，SAPF[14]和CRGA[13]相比，PNCS算法有更好的结果，其得出的最优值、平均值和最差值都要比其他两种算法更优，数据的波动更小。图2给出了PNCS算法对Problem1的寻优迭代曲线。

表1　4种算法对Problem1的统计结果比较

Table 1　Statistical results comparison of 4 algorithms with Problem1

算法	最差值	平均值	最优值	标准差
HM	−5473.9	−6342.6	−6952.1	0
SAPF	−6934.304	−6953.061	−6961.046	5.876
CRGA	−6077.123	−6740.288	−6956.251	2.70×10^2
PNCS	−6961.812	−6961.813	−6961.813	0.000259

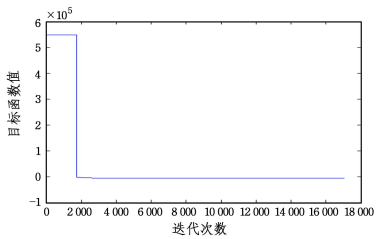

图2　Problem1的寻优迭代曲线

Fig. 2　Optimization iteration curve of Problem1

Problem2：

$$\min f(x)=-(\sqrt{n})^n\cdot\prod_{i=1}^{n}x_i$$

$$\text{s.t. } h(x)=\sum_{i=1}^{n}x_i^2=1$$

$$0\leqslant x_i\leqslant 1;i=1,\cdots,n$$

表2　4种算法对Problem2的统计结果比较

Table 2　Statistical results comparison of 4 algorithms with Problem2

算法	最差值	平均值	最优值	标准差
HM	−0.9978	−0.9989	−0.9997	N. A
PSO	−1.0042690	−1.0048795	−1.0049865	1.0
MBA	−0.996539	−0.999147	−0.999813	5.44×10^{-4}
PNCS	−0.209853674	−1.130849777	−1.902454314	0.609079146

从表2中可以看出，PNCS算法求解Problem2的最差值逊色于其他算法，但PNCS算法的平均值和最优值都要比HM[15]，PSO[20]，MBA[20]算法优。图3给出了Problem2的寻优迭代曲线。

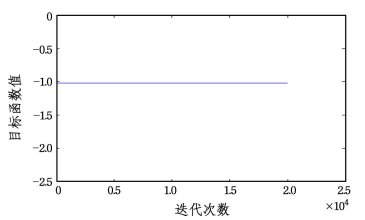

图3　Problem2的寻优迭代曲线

Fig. 3　Optimization iteration curve of Problem2

4.2　结构优化设计

(1)减速装置设计问题

在减速装置设计问题中，减速器的重量受齿轮齿的弯曲应力、表面应力、轴的横向偏转和轴的应力的约束，如图3所示。变量 x_1 到 x_7 分别表示面宽、齿模、小齿轮中的齿数、轴承之间的第一轴的长度、轴承之间的第二轴的长度以及第一轴和第二轴的直径。该问题被描述为：

$$\min f(x)=0.7854x_1x_2^2(3.3333x_3^2+14.9334x_3-43.0934)-1.508x_1(x_6^2+x_7^2)+7.4777(x_6^3+x_7^3)+0.7854(x_4x_6^2+x_5x_7^2)$$

$$\text{s.t. } g_1(x)=\frac{27}{x_1x_2^2x_3}-1\leqslant 0$$

$$g_2(x)=\frac{397.5}{x_1x_2^2x_3^2}-1\leqslant 0$$

$$g_3(x)=\frac{1.93x_4^3}{x_2x_6^4x_3}-1\leqslant 0$$

$$g_4(x)=\frac{1.93x_5^3}{x_2x_7^4x_3}-1\leqslant 0$$

$$g_5(x)=\frac{\left[(745(x_4/x_2x_3))^2+16.9\times10^6\right]^{1/2}}{110x_6^3}-1\leqslant 0$$

$$g_6(x)=\frac{\left[(745(x_5/x_2x_3))^2+157.5\times10^6\right]^{1/2}}{85x_7^3}-1\leqslant 0$$

$$g_7(x)=\frac{x_2x_3}{40}-1\leqslant 0$$

$$g_8(x)=\frac{5x_2}{x_1}-1\leqslant 0$$

$$g_9(x)=\frac{x_1}{12x_2}-1\leqslant 0$$

$$g_{10}(x)=\frac{1.5x_6+1.9}{x_4}-1\leqslant 0$$

$$g_{11}(x)=\frac{1.1x_7+1.9}{x_5}-1\leqslant 0$$

$$2.6\leqslant x_1\leqslant 3.6,0.7\leqslant x_2\leqslant 0.8,17\leqslant x_3\leqslant 28,$$

$$7.3\leqslant x_4\leqslant 8.3,7.3\leqslant x_5\leqslant 8.3,2.9\leqslant x_6\leqslant 3.9,$$

$$5.0\leqslant x_2\leqslant 5.5$$

从表3中可以看出，PNCS算法对减速装置设计问题的结果优于DEDS[17]，DELC[16]，HEAA[18]和MBA[20]算法，不管是最差值平均值还是最优值，都要优于其他算法的求解。图4给出了减速装置的寻优迭代曲线。

表3　5种算法减速装置设计问题的统计结果比较

Table 3　Statistical results comparison of 5 algorithms against design problem of deceleration device

算法	最差值	平均值	最优值	标准差
DEDS	2994.471066	2994.471066	2994.471066	3.6×10^{12}
DELC	2994.471066	2994.471066	2994.471066	1.9×10^{12}
HEAA	2994.752311	2994.613368	2994.499107	7.0×10^{-2}
MBA	2999.652444	2996.769019	2994.482453	1.56
PNCS	2994.480045	2979.983043	2922.117611	28.589667 46

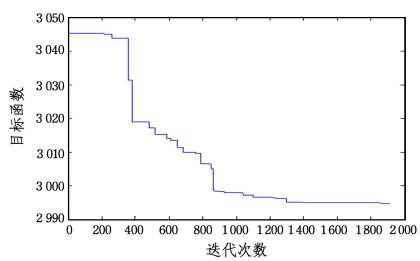

图4　减速装置的寻优迭代曲线

Fig. 4　Optimization iteration curve of deceleration device

(2)齿轮设计问题

齿轮传动设计的目的是使齿轮传动比的成本降到最低，如图5所示。问题的决策变量分别是 n_A，n_B，n_D 和 n_F，它们分别表示为 x_1，x_2，x_3 和 x_4。该问题被描述如下：

$$\min f(x)=((1/6.931)-(x_3x_2/x_1x_4))^2$$

$$\text{s.t. } 12\leqslant x_i\leqslant 60; i=1,2,3,4$$

表4 3种算法对齿轮设计问题的最佳解决方案的结果比较

Table 4 Results comparison of optimal solutions of three algorithms against gear design problem

变量	ABC	MBA	PNCS
x_1	49	43	54.749 667 29
x_2	16	16	12.000 001 54
x_3	19	19	39.496 220 07
x_4	43	49	60
$f(x)$	$2.700\,857\times10^{-12}$	$2.700\,857\times10^{-12}$	$5.104\,45\times10^{-28}$

从表4中可以看出，PNCS算法对齿轮设计问题的结果远优于ABC[19]和MBA[20]算法的求解值。图5给出了齿轮设计问题的寻优迭代曲线。

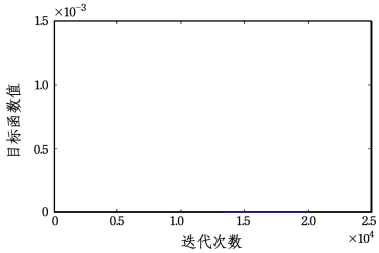

图5 齿轮设计的寻优迭代曲线

Fig. 5 Optimization iteration curve of gear design

下面给出在其他标准测试函数以及结构优化问题上，PNCS算法与其他算法的最优值结果比较。

Problem 3：

$$\min f(x)=(x_1-2)^2+(x_2-1)^2$$

$$\text{s.t. } h(x)=x_1-2x_2+1=0$$

$$g(x)=-(x_1^2/4)-x_2^2+1\geqslant 0$$

$$-10\leqslant x_i\leqslant 10; i=1,2$$

Problem 4：

$$\min f(x)=x_1^2+(x_2-1)^2$$

$$\text{s.t. } h(x)=x_2-x_1^2=0$$

$$-1\leqslant x_i\leqslant 1; i=1,2$$

Problem 5：

$$\min f(x)=(x_1^2+x_2-11)^2+(x_1+x_2^2-7)^2$$

$$\text{s.t. } g_1(x)=4.84-(x_1-0.05)^2-(x_2-2.5)^2\geqslant 0$$

$$g_2(x)=x_1^2+(x_2-2.5)^2-4.84\geqslant 0$$

$$0\leqslant x_i\leqslant 6; i=1,2$$

Problem 6：

$$\max f(x)=\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$$

$$\text{s.t. } g_1(x)=x_1^2-x_2+1\leqslant 0$$

$$g_2(x)=1-x_1+(x_2-4)^2\leqslant 0,$$

$$0\leqslant x_i\leqslant 10; i=1,2$$

Problem7：

$$\min f(x)=(x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6+7x_6^2+x_7^4+4x_6x_7-10x_6-8x_7$$

$$\text{s.t. } g_1(x)=127-2x_1^2-3x_2^4-x_3-4x_4^2-5x_5\geqslant 0$$

$$g_2(x)=282-7x_1-3x_2-10x_3^2-x_4+x_5\geqslant 0$$

$$g_3(x)=196-23x_1-x_2^2-6x_6^2-8x_7\geqslant 0$$

$$g_4(x)=-4x_1^2-x_2^2-3x_1x_2-2x_3^2-5x_6+11x_7\geqslant 0$$

$$-10\leqslant x_i\leqslant 10; i=1,2,3,4,5,6,7$$

压力容器设计问题：

$$\min f(x)=0.6224x_1x_3x_4+1.7781x_2x_3^2+3.1661x_1^2x_4+19.84x_1^2x_3$$

$$\text{s.t. } g_1(x)=-x_1+0.0193x_3\leqslant 0$$

$$g_2(x)=-x_2+0.00954\leqslant 0$$

$$g_3(x)=-\pi x_3^2x_4-(4/3)\pi x_3^3+129\,600\leqslant 0$$

$$g_4(x)=x_4-240\leqslant 0$$

$$0\leqslant x_i\leqslant 100; i=1,2$$

$$10\leqslant x_i\leqslant 200; i=3,4$$

压缩弹簧设计最优化问题：

$$\min f(x)=(x_3+2)x_2x_1^2$$

$$\text{s.t. } g_1(x)=1-(x_2^3x_3/71.785x_1^4)\leqslant 0$$

$$g_2(x)=(4x_2^2-x_1x_2/125\,66(x_2x_1^3-x_1^4))+(1/5\,108x_1^2)-1\leqslant 0$$

$$g_3(x)=1-(140.45x_1/x_2^2x_3)\leqslant 0$$

$$g_4(x)=(x_2+x_1)/1.5-1\leqslant 0$$

$$0.05\leqslant x_1\leqslant 2.00$$

$$0.25\leqslant x_2\leqslant 1.30$$

$$2.00\leqslant x_3\leqslant 15.00$$

焊接梁设计问题：

$$\min f(x)=1.104\,71x_1^2x_2+0.048\,11x_3x_4(14+x_2)$$

$$\text{s.t. } g_1(x)=\tau(x)-\tau_{\max}\leqslant 0$$

$$g_2(x)=\sigma(x)-\sigma_{\max}\leqslant 0$$

$$g_3(x)=x_1-x_4\leqslant 0$$

$$g_4(x)=0.104\,71x_1^2+0.048\,11x_3x_4(14+x_2)-5\leqslant 0$$

$$g_5(x)=0.125-x_1\leqslant 0$$

$$g_6(x)=\delta(x)-\delta_{\max}\leqslant 0$$

$$g_7(x)=P-P_c(x)\leqslant 0$$

$$0.1\leqslant x_i\leqslant 2; i=1,4$$

$$0.1\leqslant x_i\leqslant 10; i=2,3$$

由于篇幅所限，表5和表6中只列出部分代表性算法的结果进行比较。

表5 其他标准测试函数的最佳解决方案的结果比较

Table 5 Results comparison of optimal solutions of other standard test functions

算法	Problem3	Problem4	Problem5	Problem6	Problem7
HS	1.377 0	—	13.590 845	—	680.641 357 4
MBA	1.393 464 9	0.750 000	13.590 842	—	680.632 202
HM	—	—	—	−0.095 825 0	—
PNCS	1.393 464 981	0.750 000	13.590 841	0.999 999 9	680.630 215 5

表 6　结构优化问题的最佳解决方案的结果比较

Table 6　Results comparison of optimal solutions for structural optimization problems

算法	压力容器设计问题	压缩弹簧设计问题	焊接梁设计问题
CPSO	6061.0777	0.0126747	1.728024
MBA	5889.3216	—	1.724853
PNCS	2684.089819	0.0126652	1.724852543

综合以上结果可以看出，与其他算法相比，PNCS 算法对 Problem1、Problem2、Problem6、压力容器设计问题、减速装置设计问题以及齿轮设计问题有较好的结果。对 Problem4、Problem5、Problem7、弹簧设计问题以及焊接梁设计问题，PNCS 算法与其他算法的结果相似。但对 Problem3，PNCS 算法的结果略逊于 HS 算法。

结束语　结合罚函数思想，本文提出了一种罚函数与改进的布谷鸟算法结合的算法，改进了位置更新公式，扩大了搜索范围；并且通过引入动态自适应步长控制量，扩展了算法的全局勘探寻解能力，加快了算法的后期收敛速度。基于 7 个标准测试函数与 5 个结构设计优化问题，对比了已有若干算法的测试结果，得到了较高精度的最优解，结果表明该算法具有较好的有效性与可行性。本文在定义动态自适应步长控制量中仍将 K 选为常数，针对复杂的高维非线性优化问题，在迭代初始阶段应具备更大范围的搜索能力，因此可将 K 推广为迭代次数 t 的函数 $K(t)$。其中，$K(t)$ 满足单调递减并且 $\lim K(t)=K_0$，K_0 为常数。例如，$K(t)=\frac{1}{t}+K_0$。有关的改进和验证将在后续的工作中开展。

参 考 文 献

[1] YANG X S, DEB S. Cuckoo search via Lévy flights [C]// World Congress on Nature & Biologically Inspired Computing. IEEE, 2009: 210-214.

[2] YANG X S, DEB S. Engineering optimisation by cuckoo search [J]. International Journal of Mathematical Modelling and Numerical Optimisation, 2010, 1(4): 330-343.

[3] TUBA M, SUBOTIC M, STANAREVIC N. Modified cuckoo search algorithm for unconstrained optimization problems [C]// Proceedings of the 5th European Conference on European Computing Conference. World Scientific and Engineering Academy and Society, 2011: 263-268.

[4] YANG X S, DEB S. Multiobjective cuckoo search for design optimization [J]. Computer & Operations Research, 2013, 40(6): 1616-1624.

[5] WANG F, HE X S, WANG Y, et al. Markov Model and Convergence Analysis Based on Cuckoo Search Algorithm[J]. Computer Engineering, 2012, 38(11): 180-182.

[6] YANG H H, XIE P M, ZHANG X F, et al. Improved cuckoo search algorithm for multi-objective optimization problems[J]. Journal of Zhejiang University (Engineering Science), 2015, 49(8): 1600-1608.

[7] BACK T. Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms [M]. Oxford: Oxford University Press, 1996: 161-164.

[8] CAI X, WANG H, CUI Z, et al. Bat algorithm with triangle-flipping strategy for numerical optimization[J]. International Journal of Machine Learning and Cybernetics, 2018, 9(2): 199-215.

[9] ZHIGAO L, XIANG W, JU L, et al. Application of Genetic Algorithm and Penalty Function Method in Roll-Forming Process Parameters Optimization [J]. China Mechanical Engineering, 2009, 20(14): 1704-1707.

[10] BAO J H. Solution to multiple-choice knapsack problem using penalty function method implemented by genetic algorithm[J]. Computer Engineering & Design, 2008, 29(17): 4518-4520.

[11] SADOLLAH A, BAHREININEJAD A, ESKANDAR H, et al. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems[J]. Applied Soft Computing Journal, 2013, 13(5): 2592-2612.

[12] CAGNINA L C, ESQUIVEL S C, COELLO C A C. Solving engineering optimization problem with the simple cinstrained particle swarm optimizea [J]. Informatica (Slovenia), 2008, 32(3): 319-326.

[13] ADILAMIRJANOV. The dynamics of a changing range genetic algorithm under stabilizing selection[J]. International Journal of Modern Physics C, 2009, 20(7): 1063-1079.

[14] TESSEMA B, YEN G G. A Self Adaptive Penalty Function Based Algorithm for Constrained Optimization[C]// IEEE Congress on Evolutionary Computation. 2006.

[15] KOZIEL S, MICHALEWICZ Z. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization [M] // Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. 1999: 19-44.

[16] LING W, LI L P. An effective differential evolution with level comparison for constrained engineering design[J]. Structural & Multidisciplinary Optimization, 2010, 41(6): 947-963.

[17] MIN Z, LUO W J, WANG X F. Differential evolution with dynamic stochastic selection for constrained optimization[J]. Information Sciences, 2008, 178(15): 3043-3074.

[18] WANG Y, CAI Z, ZHOU Y, et al. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique[J]. Structural and Multidisciplinary Optimization, 2009, 37(4): 395-413.

[19] KARABOGA D, BASTURK B. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems[C]// Foundations of Fuzzy Logic & Soft Computing, International Fuzzy Systems Association World Congress. IFSA, Cancun, Mexico, 2007.

[20] SADOLLAH A, BAHREININEJAD A, ESKANDAR H, et al. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems[J]. Applied Soft Computing Journal, 2013, 13(5): 2592-2612.

ZANG Rui, born in 1977, associate professor. His main research interests include optimization theory and algorithm.

LIU Xiao-xiao, born in 1994, master. Her main research interests include optimization theory and algorithm.