

# 基于标准萤火虫算法的改进与仿真应用

臧 睿 李辉辉

(东北林业大学理学院 哈尔滨 150040)

**摘 要** 通过对一种智能优化算法——萤火虫算法的研究,在标准萤火虫算法中引入一种新型的自适应惯性权重来提高算法的收敛速度,并提出用虚拟萤火虫来加强萤火虫之间的相互协作和信息共享,进而改进了萤火虫的位置更新公式。针对算法中萤火虫位置的越界问题和边界早熟问题,引入一种对称边界变异,提高了改进后的算法的寻优率。对 6 个标准测试函数的实验结果表明:改进后的萤火虫算法的有效性、收敛速度得到了明显的提高。最后对两个经典工程优化问题进行了计算,运用改进后的算法所得的结果优于其它算法所得结果,也验证了萤火虫算法在改进后的适用性。

**关键词** 萤火虫算法,自适应惯性权重,相互协作,信息共享,边界变异,工程优化

中图分类号 TP301 文献标识码 A

## Improvement and Simulation Application Based on Standard Firefly Algorithm

ZANG Rui LI Hui-hui

(College of Science, Northeast Forestry University, Harbin 150040, China)

**Abstract** Through studying a kind of intelligent optimization algorithm namely firefly algorithm, the standard updating formula of firefly algorithm is improved by introducing the new adaptive inertia weight to increase the convergence speed of the algorithm and the virtual firefly is used to enhance the cooperation and the exchange of information between the fireflies. For the firefly algorithm in cross-border issues and early boundary problem, a symmetric boundary mutation is introduced, so as to improve the optimization rate of algorithm. The experiment results of six standard test functions show that the effectiveness and the convergence speed of the improved firefly algorithm are improved. In the end, the algorithm is applied to two classical engineering optimization problems, the superiority of the improved firefly algorithm is confirmed by the experiment results, and the applicability of the improved firefly algorithm is verified.

**Keywords** Firefly algorithm, Adaptive inertia weight, Mutual cooperation, Exchange of information, Boundary mutation, Engineering optimization

## 1 引言

2008 年剑桥学者 Xin-She Yang 提出了萤火虫算法(Firefly Algorithm, FA)<sup>[1]</sup>,它受启发于自然界中萤火虫的运动,是一种高级启发式算法。大量的实验测试结果表明萤火虫算法对于求解很多类型函数有着显著效果<sup>[2]</sup>。萤火虫算法由于简单易操作,需要设置的参数少且具有收敛速度快、效率高的特点,已在压力管道设计、工程等优化问题<sup>[3]</sup>、机器人、潜水器的路径规划<sup>[4,5]</sup>等领域得到了广泛的应用。Xin-She Yang 等<sup>[6,7]</sup>将标准萤火虫算法分别应用于工程优化中的管柱设计问题和三杆平面桁架结构问题上,从实验结果的对比中表明了萤火虫算法在工程优化问题中的优越性。

自萤火虫算法提出以来,国内外有很多学者对该算法进行了改进分析。如莫愿斌等人将单纯形法运用在标准萤火虫算法中从而提高算法的成功率<sup>[8]</sup>;冯艳红等提出一种基于混沌理论的动态种群萤火虫算法,减少了萤火虫的无效运动从

而提高了算法精度<sup>[9]</sup>;吴东周等改进了萤火虫算法中的最大吸引力并应用在 T-S 模糊识别中<sup>[10]</sup>。通过改进使得萤火虫算法在无约束优化、约束优化、多目标规划等诸多问题上更加适用。而这些改进算法未考虑到在迭代后期相对吸引力偏大而产生的振荡,在萤火虫的位置更新过程中也忽略了萤火虫之间的信息交流,而且在萤火虫的位置超出可行域时将边界值赋予萤火虫容易使算法陷入局部优化。

针对以上问题,本文介绍了一类改进的萤火虫算法,在标准萤火虫算法中引入了一种新颖的自适应性惯性权重、相互协作、信息共享和边界变异,并加以仿真实验与应用。

## 2 标准萤火虫算法

萤火虫算法是受自然界中的萤火虫会通过发光来吸引其它萤火虫与其进行沟通、配偶等特点所启发,是一种仿生智能优化算法,仿生原理如下<sup>[11]</sup>:

(1)自然界里的萤火虫总是会向着比较亮的的萤火虫移

本文受中央高校基本科研业务费专项资金(DL09BB40)资助。

臧 睿(1977—),男,博士,副教授,主要研究方向为最优化理论与应用,E-mail:17710573213@163.com;李辉辉(1990—),男,硕士,主要研究方向为最优化理论与应用。

动,这种移动跟萤火虫的雌雄没有关系,而最亮的萤火虫会随机移动。

(2)移动与两只萤火虫之间的相互吸引力有关系,吸引力的大小跟萤火虫自身的亮度成正比关系,而其跟萤火虫之间的距离成反比关系。

(3)在优化问题中,目标函数值的大小可以看成萤火虫的亮度,萤火虫的位置看作目标函数的解,亮度越高函数值越优,亮度最高的萤火虫的位置即是所求目标函数的最优解。

标准萤火虫算法的基本流程如下<sup>[12]</sup>。

(1)设置萤火虫的数目为  $n$ ,算法的最大迭代次数为  $MaxGeneration$ ,根据目标函数的维数随机产生  $n$  个萤火虫,由这些萤火虫的位置根据目标函数计算出  $n$  个亮度;

(2)比较上一步  $n$  个萤火虫的亮度进行排序,设置参数:常数  $\alpha$ 、最大吸引力  $\beta_0$ 、光吸收系数,按照式(1)给出位置更新公式将所有的萤火虫向最亮的萤火虫  $j$  移动,并对最亮的萤火虫进行随机扰动。

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \beta(r_{ij})(\vec{x}_j(t) - \vec{x}_i(t)) + \alpha \vec{\epsilon}_j \quad (1)$$

式(1)表示萤火虫  $j$  的亮度优于萤火虫  $i$  的亮度,萤火虫  $i$  会向萤火虫  $j$  移动,其中  $t$  表示算法迭代的次数; $\vec{x}_i, \vec{x}_j$  表示萤火虫  $i$  和萤火虫  $j$  在  $d$  维空间中所处的位置; $\alpha$  为 0 到 1 之间的常数; $\vec{\epsilon}_j$  是一个随机向量,它满足均匀分布、高斯分布或者其他分布。式(1)中  $\beta(r_{ij})$  表示萤火虫  $i$  与萤火虫  $j$  之间的相互吸引力:

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \quad (2)$$

式中  $r_{ij}$  表示萤火虫  $i$  与萤火虫  $j$  之间的空间距离:

$$r_{ij} = \|\vec{x}_i - \vec{x}_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (3)$$

式(1)中,等式的左边表示萤火虫  $i$  更新后的位置,等式右边的第 1 项表示迭代前  $\vec{x}_i$  的位置,第 2 项表示由于萤火虫  $j$  的吸引,萤火虫  $i$  的变化量,第 3 项是随机扰动项。 $\beta_0$  表示萤火虫的最大吸引力; $\gamma$  表示光吸收系数,反映了吸引力的变化并影响了算法的收敛速度和优化效果,一般情况下,  $\gamma \in [0.01, 100]$ 。

根据更新后的萤火虫的位置重新计算所对应的亮度,进行一次迭代,当迭代次数达到  $MaxGeneration$  时则停止迭代并输出最优值;否则,转到上一步继续更新萤火虫的位置。

### 3 改进萤火虫算法

针对标准萤火虫算法,本文从以下 3 个方面进行改进。

#### 3.1 算法的改进-1

从标准的萤火虫算法中可以看出:在迭代后期,相对吸引力变大影响了算法的寻优率。现可在萤火虫位置更新公式中引入惯性权重,此时算法记为 WFA。

$$\vec{x}_i(t+1) = w(t) \cdot \vec{x}_i(t) + \beta(r_{ij})(\vec{x}_j(t) - \vec{x}_i(t)) + \alpha \vec{\epsilon}_j \quad (4)$$

其中,  $w(t)$  表示惯性权重。惯性权重分为固定权重和时变权重,在粒子群算法中常用的时变权重有线性递减权重<sup>[13]</sup>和自适应性权重<sup>[14]</sup>。本文采用了一种新型的自适应性权重,公式如下:

$$w(t) = e^{-\left| \frac{f(\vec{x}_{best}(t))}{f(\vec{x}_{best}(t)) - E(f)} \right|} \quad (5)$$

其中,  $f(\vec{x}_{best}(t))$  为第  $t$  次迭代的最优值,  $E(f) = \frac{f(\vec{x}_1(t)) + f(\vec{x}_2(t)) + \dots + f(\vec{x}_n(t))}{n}$  为第  $t$  次迭代的所有函数值的平均值。

惯性权重值的大小影响了萤火虫移动的速度,从式(5)可以看出,每一次迭代的惯性权重值由上一次迭代后产生的所有函数值所决定,在迭代后期,萤火虫的位置与最优解接近时  $w(t)$  变小,萤火虫的移动速度也将变小,从而避免了振荡现象,增强了算法搜索能力,这样即可加快收敛速度,进而提高算法的寻优率。

图 1 示出以 Sphere 函数和 Ackley 函数为例的迭代曲线,从图中可以看出 WFA 算法的收敛速度要优于 FA 算法。

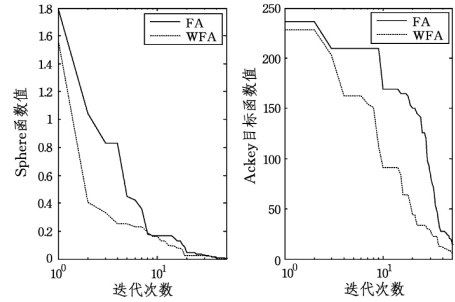


图 1 算法迭代曲线图对比

#### 3.2 算法的改进-2

从标准萤火虫的算法中得知,萤火虫的每次更新都会向着当前最亮的萤火虫移动,而这种移动是由于萤火虫之间的相对吸引力产生的。由文献<sup>[15]</sup>可知,在粒子群算法中每一个粒子自身带有一定的信息,在算法的每次更新中都会产生相互协作和信息共享。可以把这种思想运用在标准萤火虫算法中,每个萤火虫的位置和亮度可以看作其自身的信息,从而加快算法的收敛速度。在萤火虫每次的位置更新中除了要考虑萤火虫与当前最亮萤火虫之间的相互吸引力之外,还要考虑它与周围其他萤火虫之间的吸引力,进而实现萤火虫之间的相互协作和信息共享,所以在萤火虫的位置更新公式中进行了改进:在每一次迭代中计算出每只萤火虫的亮度,假设最优萤火虫的位置为  $\vec{x}_{best}$ ,萤火虫  $j$  的亮度比萤火虫  $i$  的亮度的大,它们的位置分别为  $\vec{x}_j$  和  $\vec{x}_i$ ,根据这 3 个萤火虫的位置关系,由式(6)计算出虚拟萤火虫(Virtual firefly)  $j_0^*$  的位置。

$$\vec{x}_{j_0^*} = \vec{x}_{best} + \vec{x}_j - \vec{x}_i \quad (6)$$

令  $\vec{x}_{j_0} = k \cdot (\vec{x}_{j_0^*} - \vec{x}_i) + \vec{x}_i$ , 则新的位置更新公式如下:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \beta(r_{ij_0})(\vec{x}_{j_0}(t) - \vec{x}_i(t)) + \alpha \vec{\epsilon}_j \quad (7)$$

通过实验得到:当  $k = \frac{1}{4}$  时,算法得到的效果比较满意。

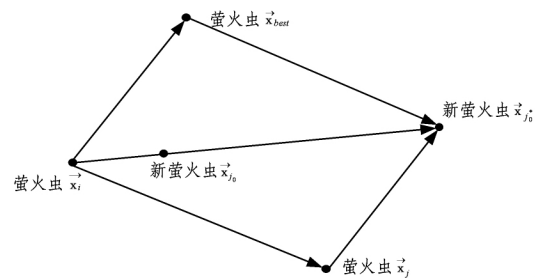


图 2 萤火虫位置更新示意图

结合了算法改进-1 和算法改进-2 的算法记为 VWFA。萤火虫的位置更新公式改为:

$$\vec{x}_i(t+1) = w(t) \cdot \vec{x}_i(t) + \beta(r_{ij_0})(\vec{x}_{j_0}(t) - \vec{x}_i(t)) + \alpha \vec{\epsilon}_j \quad (8)$$

此时,在萤火虫的移动过程中,萤火虫不再向当前最亮的萤火虫移动,而是结合当前所有萤火虫的信息向虚拟萤火虫

移动。相互协作和信息共享使得萤火虫种群具有多样性,从而提高了算法的搜索能力。

### 3.3 算法的改进-3

在萤火虫算法中,当萤火虫的位置超出可行域时,则将边界值赋予萤火虫,策略如下:

```

if  $x_{i,k} > x_{\max}$ 
     $x_{i,k} = x_{\max}$ 
end if
else if  $x_{i,k} < x_{\min}$ 
     $x_{i,k} = x_{\min}$ 
end if

```

此时则很容易使算法陷入局部最优,同时可能会在边界处出现早熟现象导致算法的寻优率降低。而针对这种问题,参照文献[16],本文引入一种对称边界变异。具体策略如下:

```

if  $x_{i,k} > x_{\max}$ 
     $x_{i,k} = 2x_{\max} - x_{i,k}$ 
end if
else if  $x_{i,k} < x_{\min}$ 
     $x_{i,k} = 2x_{\min} - x_{i,k}$ 
end if

```

这样不仅可以把萤火虫的位置保持在可行域内,而且提高了萤火虫位置的多样性,亦可跳出算法在边界值上局部最优,从而使得算法的收敛速度和寻优率得到改善。结合边界变异的萤火虫算法记为 BFA。

综上所述,将3种改进结合在标准萤火虫算法中形成一类改进算法 BVWFA。

## 4 算法仿真测试

### 4.1 测试函数

布谷鸟算法 CA 中  $N=25, Pa=0.25$ ; 粒子群算法 PSO 中种群数目  $n=400$ ; 差分算法 DE 中种群规模  $Np=50$ ; 标准萤火虫算法 FA、本文改进算法 BVWFA 的参数设置均为:  $\alpha=0.5, \beta_0=0.2, \gamma=1, n=50$ , 每个函数均独立运行 30 次。

例 1 Sphere 函数:

$$\min f_1(x) = \sum_{i=1}^d x_i^2, \text{搜索范围: } [-100, 100]^d, d=30.$$

例 2 Ackley 函数:

$$\min f_2(x) = -20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i)} + 20 + e, \text{搜索范围: } [-32, 32]^d, d=30.$$

例 3 Rastrigin 函数:

$$\min f_3 = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10], \text{搜索范围: } [-5.12, 5.12]^d, d=30.$$

例 4 Schaffer 函数:

$$\min f_4 = 0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}, \text{搜索范围: } [-10, 10]^d, d=2.$$

例 5 Griewank 函数:

$$\min f_5 = \frac{1}{400}\sum_{i=1}^N x_i^2 - 5\prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \text{搜索范围: } [-100, 100]^d, d=30.$$

例 6 Zakharov 函数:

$$\min f_6 = \sum_{i=1}^d x_i^2 + \left(\sum_{i=1}^d 0.5ix_i\right)^2 + \left(\sum_{i=1}^d 0.5ix_i\right)^4, \text{搜索范围: } [-5, 10]^d, d=30.$$

表 1 算法对测试函数的结果对比

算法	最优值	平均值	最差值	标准差	
f <sub>1</sub>	CA	525.6627	911.0378	1442.8	237.8369
	PSO	2.2044E-5	9.0107E-4	4.5326E-3	0.0012
	DE	31199	41176.0833	44865	3777.5418
	FA	4.1957E-4	5.7744E-4	7.0917E-4	8.8602E-5
	BVMFA	1.6458E-5	2.3784E-5	2.9036E-5	2.8576E-6
f <sub>2</sub>	CA	8.5156	10.6129	13.4597	1.1354
	PSO	12.3168	13.9434	15.7346	1.8761
	DE	19.0617	19.8481	20.1458	0.2421
	FA	0.0730	0.0980	0.1286	0.0294
	BVMFA	0.0022	0.0027	0.0032	2.1596E-4
f <sub>3</sub>	CA	1.24E+2	1.56E+2	1.98E+2	19.3447
	PSO	0.8891	0.5337	0.5208	0.0777
	DE	314.3773	349.1585	372.3536	16.7204
	FA	316.9155	332.5569	356.4998	10.1794
	BVMFA	1.0222E-4	1.2220E-4	1.4441E-4	1.2398E-5
f <sub>4</sub>	CA	4.1727E-05	0.0050	0.0097	0.0044
	PSO	0	0	0	0
	DE	2.0302E-6	0.0043	0.0097	0.0042
	FA	5.4348E-9	0.0062	0.0097	0.0021
	BVMFA	5.8485E-11	0.0029	0.0097	0.0044
f <sub>5</sub>	CA	1.1245	1.2392	1.4452	0.0712
	PSO	1.2338E-4	0.0134	0.0350	0.0113
	DE	7.7031	11.5154	13.4295	1.1546
	FA	5.4278E-5	0.0121	0.0222	0.0073
	BVMFA	7.3202E-6	1.1129E-5	1.3990E-5	1.2802E-6
f <sub>6</sub>	CA	360.6619	585.1390	1163.6775	169.7728
	PSO	2.2226E-4	0.21855	1.6637	0.4988
	DE	324.9570	600.6483	2588.9877	399.3133
	FA	34.5311	78.6587	128.7317	25.8198
	BVMFA	1.2801E-6	5.1108E-6	7.7224E-6	1.2088E-6

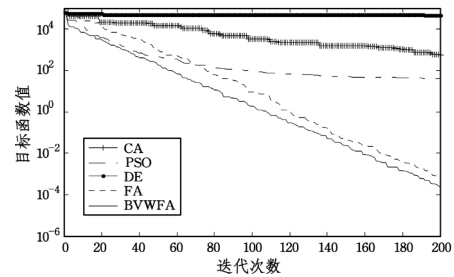


图 3 算法对例 1 的收敛曲线对比

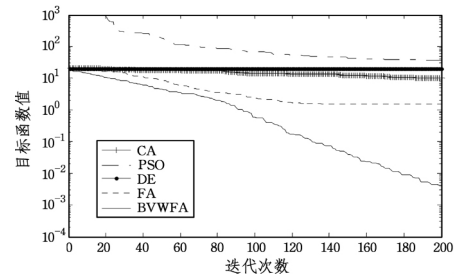


图 4 算法对例 2 的收敛曲线对比

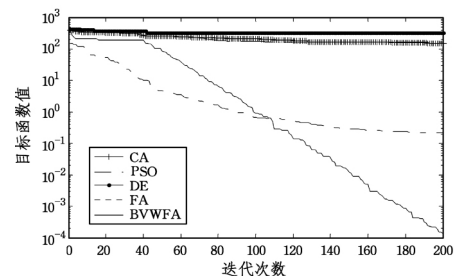


图 5 算法对例 3 的收敛曲线对比

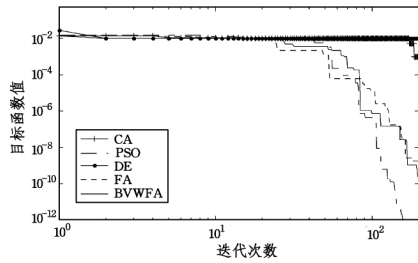


图6 算法对例4的收敛曲线对比

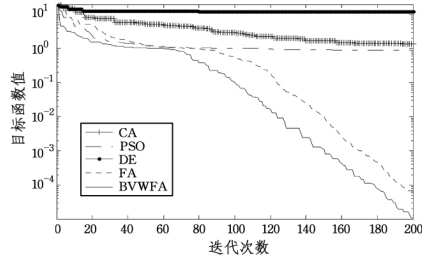


图7 算法对例5的收敛曲线对比

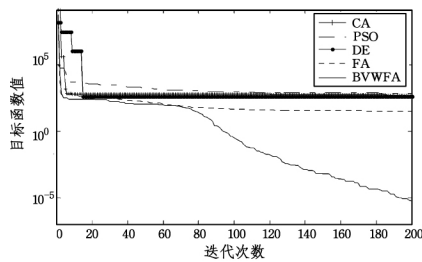


图8 算法对例6的收敛曲线对比

上述6个标准测试函数中既有单峰值问题和多峰值问题,又有高维问题和低维问题,通过仿真实验可知: $f_4$ 是一个二维函数,PSO的运算结果要优于其他结果,而在其余的4种算法中,BVWFA明显优于CA,DE,FA。在 $f_1, f_2, f_3, f_5, f_6$ 5个函数的结果中可以看出,BVWFA算法的最优值、平均值、最差值都要优于其他5种算法,在标准差方面高出其他算法几个数量级。从算法的收敛曲线图中能看出BVWFA的收敛速度也优于其他算法。根据以上6个测试函数仿真结果的比较,说明了本文的BVWFA算法的可操作性和有效性,标准萤火虫算法在经过本文的改进后,能够缩小误差,并能快速地收敛到最优解。

#### 4.2 工程应用

##### 例7 管柱设计问题<sup>[6]</sup>:

图9呈现的是设计管柱负载 $P=2500\text{N}$ 时最小成本的例子,管柱材料应力度为 $\sigma_y$ ,弹性模量为 $E$ ,管柱的高度为 $L$ 。把管柱的成本作为目标函数,函数中的未知量有管柱的直径 $d$ 和管柱的厚度 $t$ ,其中 $d$ 的取值范围为 $2\sim 14\text{cm}$ , $t$ 的取值范围为 $0.2\sim 0.8\text{cm}$ 。

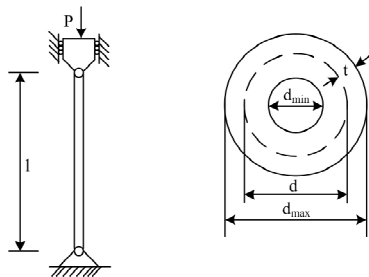


图9 管柱设计问题

管柱设计问题可简化为:

$$\begin{aligned} \min f_7 &= 9.8dt + 2d \\ \text{s. t. } &\begin{cases} g_1 = \frac{8PL^2}{\pi^3 E dt(d^2 + t^2)} - 1 \leq 0 \\ g_2 = \frac{P}{\pi dt \sigma_y} - 1 \leq 0 \end{cases} \end{aligned}$$

其中, $2\text{cm} \leq d \leq 14\text{cm}$ , $0.2\text{cm} \leq t \leq 0.8\text{cm}$ , $P = 2500\text{N}$ , $\sigma_y = 500\text{N/cm}^2$ , $E = 0.85 \times 10^6\text{N/cm}^2$ , $L = 250\text{cm}$ 。

算法中参数设置均为: $\alpha = 0.5, \beta_0 = 0.2, \gamma = 1, n = 50, \text{MaxGeneration} = 200$ 。两种算法对例7的结果对比如表2和图10所示。

表2 两种算法对例7的结果对比

	最优值	平均值	最差值	最优解
FA	26.53217	26.53504	26.53972	(5.45139, 0.29196)
BVWFA	26.49951	26.49957	26.49966	(5.45116, 0.29197)

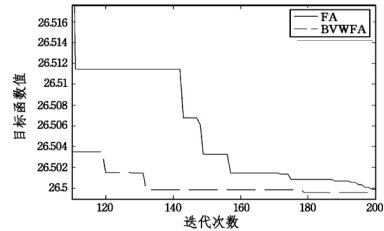


图10 两种算法对例7的收敛曲线对比

文献[6]中给出了Hsu和Liu等人,Rao,Yang对于管柱设计问题用3种不同算法所得到的最优解,与本文改进后萤火虫算法的运算结果进行对比,如表3所列。

表3 管柱设计问题最佳方案的结果对比

	Hsu and Liu	Rao	Yang	BVWFA
d	5.4507	5.44	5.45139	5.45116
t	0.292	0.293	0.29196	0.29197
$g_1$	$-7.8 \times 10^{-5}$	-0.8579	-0.0241	-1.618E+5
$g_2$	0.1317	0.0026	-0.1095	-1.765E+5
最优解	25.5316	26.5323	26.53217	26.49951

可以看到改进的萤火虫算法得到的是符合约束条件下的最好结果。

##### 例8 三杆平面桁架结构问题<sup>[7]</sup>:

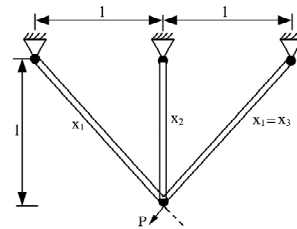


图11 三杆平面桁架结构问题

在三杆桁架结构中,3个杆的长度均是 $l$ ,截面积分别为 $x_1, x_2$ 和 $x_3$ ,其中 $x_1 = x_3$ 。结点处受的力为 $P$ ,杆的应力度为 $\sigma$ 。以3个杆满足一定条件时的最小体积为目标函数。

三杆平面桁架结构问题可简化为:

$$\begin{aligned} \min f_8 &= (2\sqrt{2}x_1 + x_2) \times l \\ \text{s. t. } &\begin{cases} g_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \\ g_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \\ g_3 = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0 \end{cases} \end{aligned}$$

(下转第132页)

[27] Nefian A. A Hidden Markov Model-Based Approach for Face Detection and Recognition[D]. Georgia:Georgia Institute of Technology,1999

[28] Bobulski J. 2DHMM-Based Face Recognition Method[J]. Image Processing and Communications Challenges,2016(1):11-18

[29] Arya K V, Anukriti. Face recognition under partial occlusion using HMM and Face Edge Length Model[R]. The 9th International Conference on Industrial and Information Systems(ICIS),2014

[30] 李忠健,杨宜民,梅平. 基于改进的隐马尔科夫模型的人脸识别方法[J]. 工业控制计算机,2014(11):29-31

[31] William H,William T,Brian P. Numerical recipes in C;the art of

scientific computing [M]. New York: Cambridge University Press,1986

[32] Rowley H A,Baluja S,Kanade T. Neural Network-Based Face Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,1998(1):23-38

[33] Raptis M, Kirovski D, Hoppe H. Real-time classification of dance gestures from skeleton animation[J]. Eurographics/ACM Siggraph Symposium on Computer Animation,2011(2):147-156

[34] Oluwatoyin P. Popoola, Kejun Wang. Video-Based Abnormal Human Behavior Recognition—A Review[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),2012(6):865-878

(上接第 116 页)

其中,  $0 \leq x_1 \leq 1 \text{cm}^2$ ,  $0 \leq x_2 \leq 1 \text{cm}^2$ ;  $l = 100 \text{cm}$ ,  $P = 2 \text{KN}$ ,  $\sigma = 2 \text{KN/cm}^2$ 。

算法中参数设置均为:  $\alpha = 0.5$ ,  $\beta_0 = 0.2$ ,  $\gamma = 1$ ,  $n = 50$ ,  $\text{MaxGeneration} = 200$ 。两种算法对例 8 的结果对比如表 4、图 12 所示。

表 4 两种算法对例 8 的结果对比

	最优值	平均值	最差值	最优解
FA	263.89625	263.90614	264.01298	(0.78863, 0.40838)
BVWFA	263.89585	263.89740	263.90247	(0.78870, 0.40817)

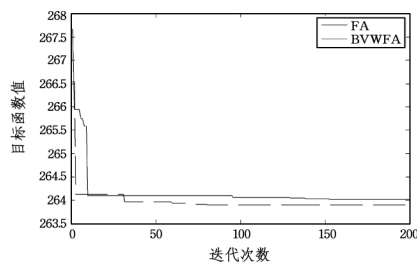


图 12 两种算法对例 8 的收敛曲线对比

文献[7]中给出了 Park 等人、Ray 和 Saini 等人、Yang 对于管柱设计问题用 3 种不同算法所得到的最优解,与本文改进后萤火虫算法 BVWFA 的运算结果进行对比,如表 5 所列。

表 5 三杆平面桁架结构问题最佳方案的结果对比

	Park	Ray and Saini	Yang	BVWFA
$x_1$	0.78879	0.795	0.78863	0.78870
$x_2$	0.40794	0.395	0.40838	0.40817
$g_1$	0	-0.00169	0	0
$g_2$	-0.26778	-0.26124	-0.26802	-1.4642
$g_3$	-0.73223	-0.74045	-0.73198	-0.5358
最优值	263.8965	264.300	263.8962	263.8958

从算法运算的结果对比中看出改进萤火虫算法要优于其它算法。

结束语 本文提出了一种基于标准萤火虫算法的改进算法,在标准算法中引入自适应性惯性权重提高了算法的寻优率,并引入相互协作和信息共享的思想改进算法中的迭代公式,从而提高了算法的收敛精度,使算法在降低迭代次数的条件下能够快速地收敛到最优解,对称边界变异的引入也增加了萤火虫算法中种群的多样性。通过 6 个不同类型的函数的

测试验证了改进后萤火虫算法 BVWFA 的可行性,最后通过对文中的两个工程实例问题的计算证实了 BVWFA 的有效性和实用性。

## 参考文献

[1] Yang Xin-she. Nature-Inspired Metaheuristic Algorithms[M]. Luniver Press,2010

[2] Yang X S,贺兴时. 群体智能和智能优化算法[J]. 纺织高校基础科学学报,2013,26(3):287-296

[3] 杨艳,周永权,罗林,等. 人工萤火虫群优化算法求解约束优化问题[J]. 小型微型计算机系统,2014,35(1):185-188

[4] 刘鹏,刘弘,郑向伟,等. 基于改进萤火虫算法的动态自动聚集路径规划方法[J]. 计算机应用研究,2011,28(11):4146-4149

[5] 董静. 萤火虫算法研究及其在水下潜器路径规划中的应用[D]. 哈尔滨:哈尔滨工程大学,2013

[6] Gandomi A H, Yang Xin-she, Alavi A H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems[J]. Engineering with Computers,2013,29(1):17-35

[7] Yang Xin-she, Gandomi A H. Bat algorithm: a novel approach for global engineering optimization[J]. Engineering Computations,2012,29(5):464-483

[8] 莫愿斌,马彦迪. 单纯形法的改进萤火虫算法及其在非线性方程组求解中的应用[J]. 智能系统学报,2014,9(6):747-755

[9] 冯艳红,刘建芹,贺毅朝. 基于混沌理论的动态种群萤火虫算法[J]. 计算机应用,2013,33(3):796-799

[10] 吴东周,丁学明. 基于改进萤火虫算法的 T-S 模型辨识[J]. 计算机仿真,2013,30(3):327-330

[11] 赵玉新, Yang X S, 刘利强. 新兴元启发式优化方法[M]. 北京:科学出版社,2013

[12] 刘长平,叶春明. 一种新颖的仿生群智能优化算法:萤火虫算法[J]. 计算机应用研究,2011,28(9):3295-3297

[13] 罗辞勇,陈民铀,韩力. 适应性粒子群寻优算法[J]. 控制与决策,2009,24(6):859-863

[14] 陈寿文. 基于质心和自适应指数惯性权重改进的粒子群算法[J]. 计算机工程与应用,2015,3(3):675-679

[15] 杨朋尊. 粒子群算法的拓扑结构分析与研究[D]. 太原:太原理工大学,2010

[16] 宋莉,邓长寿,曹良林. 粒子群优化算法的边界变异策略比较研究[J]. 计算机工程,2015,41(3):191-197