

一种改进的变步长自适应 GSO 算法

黄 凯, 周永权

(广西民族大学数学与计算机科学学院, 南宁 530006)

摘 要: 基本萤火虫群优化(GSO)算法在求解全局优化问题时, 存在收敛速度慢、求解精度不高等问题。为此, 提出一种变步长自适应 GSO 算法。该算法在一定程度上可以避免 GSO 算法过早陷入局部最优, 且步长随迭代次数的增加而自适应地调整, 从而使算法在后期获得精度更高的解。运用 6 个标准测试函数进行实验, 结果表明, 与 GSO 算法相比, 该算法的收敛速度及精度均有明显提高。

关键词: 全局优化; 局部最优; 萤火虫群优化算法; 自适应

Improved Variation Step Adaptive GSO Algorithm

HUANG Kai, ZHOU Yong-quan

(College of Mathematics and Computer Science, Guangxi University for Nationalities, Nanning 530006, China)

【Abstract】 An improved Variation Step Adaptive Glowworm Swarm Optimization(VSAGSO) algorithm is proposed to solve the problem of slow convergence and low precision and easy to fall into local optimization of the Glowworm Swarm Optimization(GSO) algorithm. It endows a big initial step to each glowworm. The step is decreased dynamically along with the increase of iteration so that the algorithm can get more precise solution in the end of the algorithm. Experimental results with six test function show convergence speed and precision is dramatically improved, which testifies that VSAGSO is a valid method to solve the global optimization problem.

【Key words】 global optimization; local optimum; Glowworm Swarm Optimization(GSO) algorithm; adaptive

DOI: 10.3969/j.issn.1000-3428.2012.04.060

1 概述

自然科学和工程技术等领域许多问题的求解都可归结为全局优化问题。关于全局优化问题, 迄今为止, 研究者提出很多种类优化算法, 包括基于梯度的算法和各种启发式算法, 如 DFP 变尺度算法、黄金分割法、共轭方向法、Powell 方向加速法和区间方法等^[1-3], 这些算法对初值选取依赖性高, 且对函数的性态要求较高。许多工程问题的目标函数性态差, 一般很难用传统算法来求解。近年来, 随着计算智能技术的发展, 各种新的仿生智能算法相继被提出, 如粒子群算法、模拟退火算法、遗传算法等。然而由于各种函数的形态与复杂性不同, 因此在优化问题上每种算法都有各自的优点与不足。

人工萤火虫群优化(Glowworm Swarm Optimization, GSO)算法是种新型群智能算法^[4-5], 该算法近年来在计算智能领域引起人们极大关注, 并逐渐成为研究热点。随着研究的深入, 该算法已经成功应用于传感器的噪声测试^[6]和模拟机器人^[7]等。该算法捕捉极值速度快、捕捉效率高, 具有较强的通用性等优点。但该算法也存在着易陷入局部最优、收敛速度慢、易出现震荡、求解精度不高等问题。本文针对基本 GSO 算法的不足, 提出一种变步长自适应 GSO(Variation Step Adaptive GSO, VSAGSO)算法。

2 基本萤火虫群优化算法

在基本萤火虫群优化算法中, 首先随机地在解空间中初始化一群萤火虫, 每一只萤火虫都携带有初始值相同的荧光素, 萤火虫的亮度是与其荧光素值呈正比例关系, 而荧光素值又与其运动过程中所处位置的适应度值联系在一起, 荧光素值越高, 其邻域范围内的萤火虫吸引力就越强。邻域范围称为决策域范围, 其大小由其邻域半径 r_d 决定。 r_d 的大小在

初始最大值 r_s 与 0 之间变动, r_s 称为可视范围。在萤火虫运动中, 每一只萤火虫以一定的概率向其邻域范围内的邻居萤火虫前进。萤火虫 j 要成为萤火虫 i 的邻居, 必须满足 j 在 i 的邻域决策范围之内且 j 的荧光素值要高于 i 。通过萤火虫不断运动, 较多的萤火虫最后会聚集在适应度值较高的萤火虫周围。

在 GSO 中, 每一次迭代都由 2 个阶段组成: 第 1 阶段是荧光素更新阶段; 第 2 阶段是萤火虫的运动阶段。

(1) 荧光素更新阶段

在此阶段中, 每一只萤火虫都按式(1)对荧光素进行更新。

$$l_i(t) = (1 - \rho)l_i(t-1) + \gamma J(x_i(t)) \quad (1)$$

其中, $l_i(t)$ 为第 t 代第 i 个萤火虫的荧光素值; $\rho \in (0, 1)$ 为控制荧光素值的参数; γ 为评价函数值的参数; $J(x_i(t))$ 为适应度函数值。

(2) 萤火虫运动阶段

在此阶段当中, 萤火虫 i 以一定的概率选择邻域范围内的萤火虫 j 并朝其运动, 概率选择如式(2)所示。萤火虫 i 下一时刻的位置由式(3)确定, 在运动阶段的最后进行决策域范围的更新, 用式(4)进行更新。

路径概率选择公式:

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (2)$$

基金项目: 广西自然科学基金资助项目(0991086)

作者简介: 黄 凯(1985—), 男, 硕士, 主研方向: 智能计算; 周永权, 教授、博士

收稿日期: 2011-08-26 **E-mail:** yongquanzhou@126.com

位置更新公式:

$$x_i(t+1) = x_i(t) + s * \begin{pmatrix} x_j(t) - x_i(t) \\ \|x_j(t) - x_i(t)\| \end{pmatrix} \quad (3)$$

其中, $x_i(t) \in R^m$ 表示 i 萤火虫在 m 维实数空间的位置; $\|\cdot\|$ 表示标准欧基里德距离; $s(s>0)$ 表示移动步长。

决策域范围更新公式:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_i - |N_i(t)|)\}\} \quad (4)$$

其中, β 是一个比例常数; n_i 是控制邻域范围内邻居萤火虫个数的参数; $|N_i(t)|$ 是 i 萤火虫邻域内邻居萤火虫个数。

3 本文算法

3.1 算法原理及变步长策略分析

GSO 算法在求解全局优化问题, 具有求解速度快、效率高、通用性强等优点。但由于在萤火虫的运动中, 每一只萤火虫采用固定步长进行移动, 如果固定步长取值较大, 则可能导致萤火虫跳离最优解, 出现收敛速度慢、后期易出现震荡现象等问题。如果固定步长取值较小, 可能导致算法过早的陷入局部最优解, 同样使得得不到较高精度的解。因此, 本文采用式(5)^[8]使萤火虫的移动步长随着迭代次数的增加而自适应改变, 在算法的运行初期保持一个最大值, 避免算法过早的陷入局部最优, 随着迭代次数的增加, 步长自适应的减小, 最后保持一个最小值, 从而有使算法在后期快速收敛, 得到更精确的解。

$$\begin{cases} step = step \times a + step_{min} \\ a = \exp(-30 \times (\frac{t}{T_{max}})^p) \end{cases} \quad (5)$$

其中, t 为当前迭代次数; T_{max} 为最大的迭代次数; $step_{min}$ 为步长的最小阈值, 本文取值为 0.002; p 为大于 1 的整数, 本文 p 取值为 10, 可据具体情况在取值范围[1,30]之间选取, 图 1 给出了 p 分别为 3、5 和 10 时, 系数 a 的变化曲线^[8], 这里, 定义 t/T_{max} 为 k 。

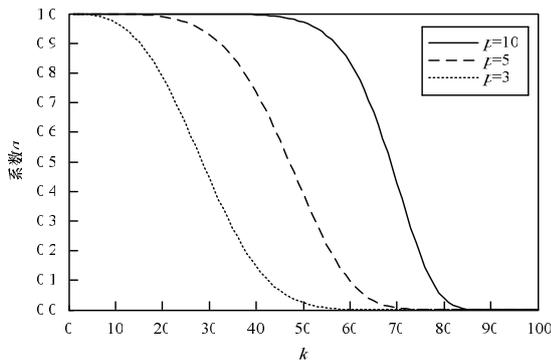


图 1 系数 a 的变化曲线

3.2 算法步骤

本文算法步骤如下:

Step1 初始化 $\rho, \gamma, \beta, s, l_0, m, n, step_{min}, p$ 等参数, 在问题的解空间中随机地初始化萤火虫群, 并初始化最大的迭代次数 T_{max} 。

Step2 对所有萤火虫按式(1)更新荧光素值。

Step3 进入运动阶段后, 求每只萤火虫的邻域邻居萤火虫集合。

Step4 用轮盘赌方法选择移动方向的萤火虫 $j(j \in N_i(t))$, 并用式(3)进行位置的更新。

Step5 用式(5)对萤火虫步长进行更新。

Step6 用式(4)对决策域半径进行更新。

Step7 完成一次迭代, 判断是否满足结束条件, 记录结果, 退出迭代, 否则转 Step2 进入下一次迭代。

4 实验分析

4.1 实验测试函数

为验证 VSAGSO 算法的有效性, 本文采用 6 个标准测试函数对算法进行测试, 如表 1 所示, 其中, f_1 是低维的函数, $f_2 \sim f_6$ 是文献[8]给出的高维函数, 测试函数如下:

$$f_1(x) = g(x)h(x)$$

$$g(x) = 1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)$$

$$h(x) = 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_2^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$$

$$f_2(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j^2)$$

$$f_3 = \sum x_i^2$$

$$f_4 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$$

$$f_5 = \sum_{i=1}^{n-1} (100(x_{i+1}^2 - x_i)^2 + (1 - x_i)^2)$$

$$f_6 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

表 1 标准测试函数

函数	维数	搜索范围	最优值
f_1	2	[-2,2]	3
f_2	10	[-100,100]	0
f_3	10	[-10,10]	0
f_4	10	[-100,100]	0
f_5	10	[-600,600]	0
f_6	10	[-100,100]	0

实验参数设置如表 2 所示。实验所用萤火虫为 50 只, 迭代次数为 1000, 其他可变参数数据不同函数的特征设置如表 3 所示。

表 2 VSAGSO 及 GSO 算法参数取值

ρ	γ	β	n_i	l_0
0.4	0.6	0.08	5	5

表 3 VSAGSO 及 GSO 算法可变参数取值

函数	步长 s	可视范围 r_s
f_1	0.03	2
f_2	3.00	5
f_3	3.00	5
f_4	3.00	5
f_5	3.00	5
f_6	3.00	5

4.2 测试平台

实验采用 Matlab2008a 编写的仿真程序, 在 Windows XP 操作系统, Intel® Core 2 Duo CPU E4500 2.20 GHz, 1 GB 内存的 PC 机。

4.3 实验仿真结果分析

对选用 6 个标准测试函数, 进行 20 次独立实验, 分别取

其最优值、最差值和均值进行对比, 结果如表 4、图 2~图 7 所示。

表 4 标准测试函数实验结果对比

函数	算法	最优值	最差值	平均值
f_1	VSA	3.000 010 894 649 046	8.710 436 984 603 946	3.024 935 757 505 407
	GSO	3.006 277 715 219 376	8.710 436 984 603 946	4.360 643 859 453 527
f_2	VSA	0.467 989 353 221 600	2.609 586 749 993 165	1.402 482 912 571 272
	GSO	0.848 152 461 233 635	2.609 586 749 993 165	1.823 879 213 460 648
f_3	VSA	0.102 772 866 798 469	0.503 380 333 296 108	0.279 276 134 385 142
	GSO	0.198 229 041 100 304	0.587 917 674 220 058	0.417 601 279 241 011
f_4	VSA	0.013 541 921 792 442	0.062 406 344 425 471	0.029 208 257 729 205
	GSO	0.025 784 188 039 498	0.062 406 344 425 471	0.044 011 844 171 525
f_5	VSA	13.151 089 967 619 935	30.093 457 750 597 565	20.363 006 052 399 960
	GSO	15.687 328 979 928 465	30.093 457 750 597 565	22.558 256 574 103 492
f_6	VSA	10.569 177 164 610 904	26.864 433 857 580 877	18.292 188 771 258 854
	GSO	15.933 350 619 526 319	26.864 433 857 580 877	21.085 294 969 111 416

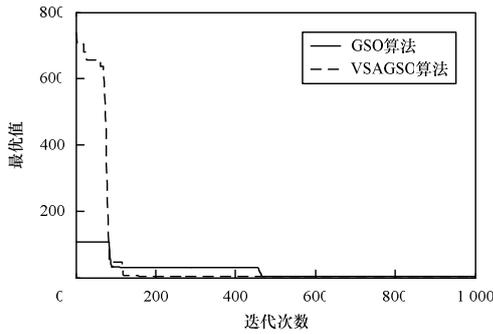


图 2 f_1 函数收敛曲线对比

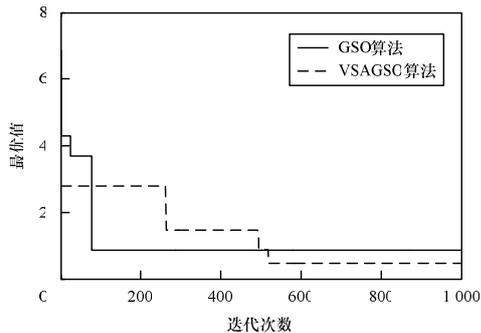


图 3 f_2 函数收敛曲线对比

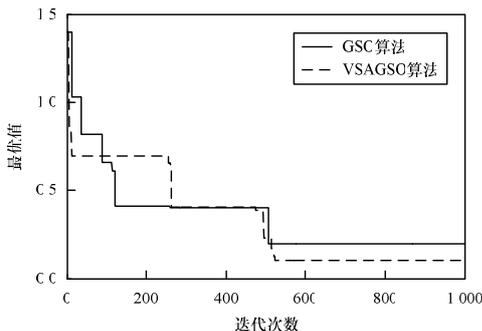


图 4 f_3 函数收敛曲线对比

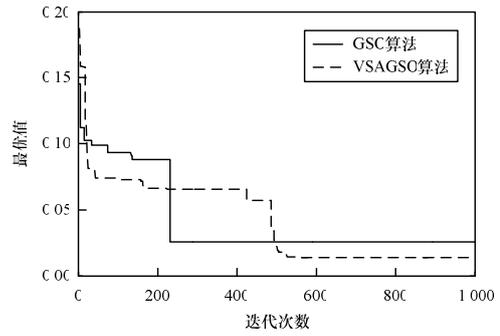


图 5 f_4 函数收敛曲线对比

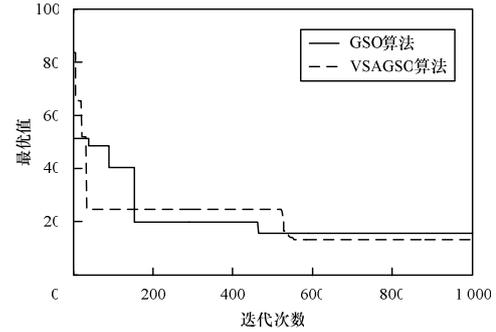


图 6 f_5 函数收敛曲线对比

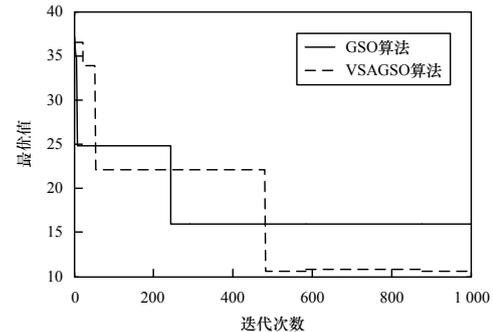


图 7 f_6 函数收敛曲线对比

从图 2~图 7 可以看出, VSAGSO 的收敛曲线在收敛后期都在 GSO 曲线的下方, 由此可见, VSAGSO 算法的收敛速度和求解精度要比 GSO 算法高, 且在算法后期随着求解精度的提高而有效地避免震荡现象产生。

5 结束语

本文提出一种自适应变步长人工萤火虫群优化算法, 该算法通过步长随着迭代次数的增加而自适应地调整, 提高萤火虫算法寻优的全局搜索能力, 有效提高算法效率。实验结果表明, 改进算法收敛速度快、计算精度高, 但在求高维函数优化问题时表现一般, 因此, 如何提高算法在求高维全局优化问题时的效率和精度是今后的研究方向。

参考文献

- [1] 袁亚湘, 孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 2003.
- [2] 陈开周. 最优化计算方法[M]. 西安: 西安电子科技大学出版社, 1990.
- [3] Csended M. Numerical Experiences with a New Generalized Subinterval Selection Criterion for Interval Global Optimization[J]. Reliable Computing, 2003, 9(2): 109-125.

(下转第 193 页)