

DOI:10.15923/j.cnki.cn22-1382/t.2018.6.08

基于改进萤火虫算法的航路规划问题求解

姜 彬¹, 乔红宇², 施志刚¹

(1.江苏航运职业技术学院 机电工程系, 江苏 南通 226000;

2.江苏航运职业技术学院 轮机工程系, 江苏 南通 226000)

摘 要: 自适应调整萤火虫吸收系数和随机系数以改进随机步长。算法前期侧重全局寻优, 而算法后期局部寻优以提升萤火虫算法的快速性和准确性。用该算法对航路优化问题进行了仿真实验。

关键词: 萤火虫算法; 航路规划; 自适应; 随机步长

中图分类号: TP 391 **文献标志码:** A **文章编号:** 1674-1374(2018)06-0562-06

Based on an improved glowworm swarm algorithm path planning problem research

JIANG Bin¹, QIAO Hongyu², SHI Zhigang¹

(1.Department of Technical and Electrical Engineering, Jiangsu Vocation & Technical Shipping College, Nantong 226000, China;

2.Department of Marine Engineering, Jiangsu Vocation & Technical Shipping College, Nantong 226000, China)

Abstract: The firefly absorption coefficient and random coefficients are selected adaptively to improve the random step length. Global optimization first and then local optimization in the algorithm is to upgrade the rapidity and accuracy of the firefly algorithm. Simulation for the route optimization problem is carried.

Key words: firefly algorithm; route planning; adaptive; random step length.

0 引 言

航路规划在存在一些障碍物的环境中, 从一个初始位置 S 到达一个目标位置 G , 通过一些优化方法规划出从初始位置能够避开障碍顺利到达目标位置的路径。考虑航路规划问题本质上是基于抽象环境的模型搜索优化路径问题^[1]。根据对

环境的掌握情况, 航路规划可以划分为全局规划和局部规划。目前对航路规划问题的求解算法主要有人工势场法^[2-3]、可视图法^[4-5]和智能规划法^[6-7]。

文中在研究各种改进的萤火虫算法特点的基础上, 通过分析该问题的特性, 摒弃了以往算法的不足, 采用了自适应地选择吸收系数 γ 和随

收稿日期: 2018-05-12

基金项目: 江苏省教育厅基金资助项目(苏教科[2017]5号); 江苏省高校自然科学研究面上项目(17KJD413001)

作者简介: 姜 彬(1980—), 男, 汉族, 江苏如皋人, 江苏航运职业技术学院副教授, 硕士, 主要从事人工智能与智能系统方向研究,
E-mail: jiangb_28@ntsc.edu.cn.

机系数 α 来增加算法的全局寻优能力。针对传统方法收敛慢的不足,提出了随机步长的改进策略,在保证算法精确搜索的前提下,合理调整步长来加快算法的收敛性。最后以仿真实验对比几种算法的求解精度和速率,证明了文中提出改进的萤火虫算法的有效性。

1 航路规划问题建模

1.1 航路规划问题任务描述

航路规划是在一定的约束条件下设计和规划航路,从而寻找到从出发点到目标点的一个最优路径。例如无人机、无人地面探测车、无人水面舰艇等。它们需要在不同的场景下,规划出不同的航路,从而完成相应的任务,每次航行经过的一些航路点顺次连接起来的线段便是航路。航路规划示意如图1所示。

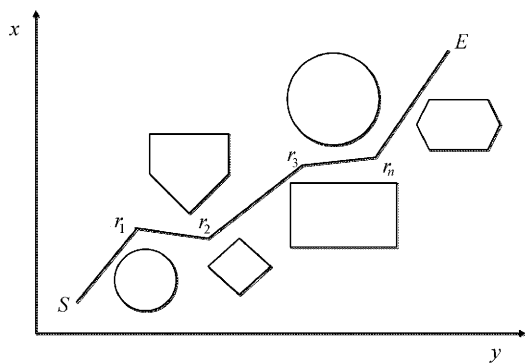


图1 航路规划示意图

若用 (r_1, r_2, \dots, r_n) 表示航路点序列,那么由该航路点构成的航路可以表示为 $P = (S, r_1, r_2, \dots, r_n, E)$ 。其中 S, E 分别表示出发点和目标点,而航路规划的目的就是确定出这中间的 N 个点的位置和顺序。

采用这种方式的航路表达有很多优点。例如可以通过增加航路节点的数量来提高航路的精度,并且在设计航路的过程中可以将一个较大的整体问题划分为几个较小的区域路径规划子问题,相对于直接求解大规模问题,小规模子问题更易求解。

1.2 航路规划模型设计

航路优劣的评价依据便是设计航路的各种指标。实际上航路规划目标函数的设计是要综合考虑航路所需满足的各种指标,从而制定出合适的目标函数,给搜索算法提供正确的寻优方向。目

标函数的设计首先要保证将航路设计的各种指标进行一个全面综合的考虑,然后尽可能地减少目标函数的计算量。文中的航路规划指标主要考虑航路长度和威胁对航路所产生的影响。

$$J_1 = L_S + \sum_{i=1}^{n-1} \Delta L_i + L_E$$

$$\Delta L_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad (1)$$

式中: (x_i, y_i, z_i) ——第 i 个航路点的坐标;

$(x_{i-1}, y_{i-1}, z_{i-1})$ ——第 $i-1$ 个航路点的坐标, $1 \leq i < n$;

L_S ——从出发点开始到达第一个航路规划点的距离;

L_E ——最后一个航路规划点和终点的距离;

ΔL_i ——相邻的两个航路规划点之间的距离。

同时,还要考虑在航路规划过程中,无人机或者无人车等设备要尽可能地避开一些障碍或者危险。因此,给出如下目标函数:

$$J = \omega_1 J_1 + \omega_2 J_2 \quad (2)$$

$$J_2 = \begin{cases} \frac{K_j}{R_j^4}, & \sqrt{x^2 + y^2 + z^2} \in R_j \\ 0, & \sqrt{x^2 + y^2 + z^2} \notin R_j \end{cases} \quad (3)$$

式(2)表示综合的目标函数,由两部分构成。这两部分的重要程度通过加权的方式构成。其中 J_1 由式(1)给出, J 表示整个航路的路程, J_2 则表示航路中对无人机产生的各种威胁。若有 j 个威胁目标,那么当航路进入到距离该威胁目标的威胁半径 R_j 之内时,我们认为该威胁目标会对无人机产生威胁,那么这样的航路就是不安全的, K_j 为威胁矫正因子。

2 改进的萤火虫算法

2.1 基本萤火虫算法分析

萤火虫可以通过对光线的感知来获得其它个体发出的信息来完成相应的一些行为。对光线的感知主要有两个因素:其一是光强度 I 与距离光源的距离 r 的平方成反比;其二是光在空气中传播会逐渐减弱,一般情况下萤火虫发出的光是能够在几百米外被感知到。

在萤火虫算法的构建过程中,需要设定三个理想化的约束条件^[13]:

约束 1: 萤火虫不存在性别的差异, 它们都会飞向吸引力大、发光亮度较大的萤火虫。

约束 2: 萤火虫吸引力的大小和它的亮度成正比。亮度随着个体间距离的增加而减小, 该规则与空气吸收光的实际情况是相似的。若某一个萤火虫没有任何比它更具有吸引力的其它的萤火虫存在, 那么该虫将会随机自由飞行。

约束 3: 一个萤火虫的亮度或者吸引力由所需求解的目标函数来确定。

基本的萤火虫算法的核心思想是每个萤火虫会被其它的绝对亮度大于自身的萤火虫吸引。每个萤火虫以此来移动自身的位置。首先把萤火虫 i 的绝对亮度 I_i 和我们求解的优化问题目标函数联系起来。如果萤火虫 i 比萤火虫 j 的绝对亮度大, 则萤火虫 i 吸引萤火虫 j 向它移动。萤火虫 i 对萤火虫 j 的相对亮度可以决定它们之间的互相引力大小。定义了萤火虫 i 对萤火虫 j 的相对亮度, 如下式:

$$I_{ij}(r_{ij}) = I_i e^{-\gamma r_{ij}^2} \quad (4)$$

式中: I_i —— 萤火虫 i 的绝对亮度;

γ —— 光吸收系数, 可设为常数;

r_{ij} —— 萤火虫 i 到 j 的距离。

现在假定萤火虫 i 对萤火虫 j 的吸引力是与两个萤火虫之间的相对亮度成正比例, 那么根据萤火虫 i 相对亮度公式, 可得萤火虫 i 对萤火虫 j 的吸引力为:

$$\beta_{ij}(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \quad (5)$$

式中: β_0 —— 最大吸引力, 根据经验给出取值 $\beta_0 = 1$;

γ —— 空气对光的吸收率, 代表了吸引力随着距离变化的情况, 对于很多的问题可以取 $\gamma \in [0.01, 100]$;

r_{ij} —— 萤火虫 i 对萤火虫 j 的距离, 可以定义为笛卡尔距离, 即

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (6)$$

萤火虫 j 被萤火虫 i 吸引并向其移动, 文中算法的位置更新公式为:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha l \left(\text{rand}() - \frac{1}{2} \right) \quad (7)$$

式中: x_i, x_j —— 萤火虫 i, j 所处的空间位置;

$\beta_{ij}(r_{ij})$ —— 萤火虫 i 对萤火虫 j 的吸引力;

α —— 常数, 一般取 $\alpha \in [0, 1]$;

ϵ_i —— 高斯分布、均匀分布或者其它分布得到的随机向量;

l —— 随机运动步长, 是一个常数。

由上式分析可知, 更新的位置取决于吸引力大小和带有特定系数的随机数。

2.2 改进的萤火虫算法

2.2.1 基本萤火虫算法缺陷分析

如前所述, 萤火虫算法的寻优是依靠萤火虫之间的相互作用实现的, 为了能够在较短的时间内找到最优解, 一般要求优化算法在搜索早期具有较强的全局寻优能力。分析式(7)可知, 在搜索初期, 每个萤火虫个体位于整个解的空间上, 当搜索范围较大的时候, 两个萤火虫之间的距离会比较大, 导致 $\beta_{ij}(r_{ij})$ 很小, 所以位置更新的第二项 $\beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i)$ 是非常小的, 导致了吸引力对萤火虫移动更新的影响很小。另外式中第三项也比较小, 萤火虫自己的局部搜索范围也很小。这就造成算法无法准确找到最优解所在区域。到了算法搜索后期, 萤火虫已经比较接近最优解, 此时情况和之前的恰恰相反, 萤火虫之间的距离会非常小, 导致 $\beta_{ij}(r_{ij})$ 很大, 同时由于随机项的作用, 使得获得的解依然有较大的变化。

由于每次每只萤火虫的位置更新要受到比它更优的萤火虫的影响, 而这些更优的萤火虫不一定是合理的解。这种错误的搜索方向可以依靠整个种群中其它较为优秀的萤火虫来修正, 但这一过程显然是影响了算法的收敛速率。

2.2.2 参数自适应调整策略的改进

由萤火虫的运动公式可以看出, 决定萤火虫下一步位置的因素有 3 个, 分别是当前的萤火虫和比较优秀的萤火虫的距离、光吸收系数 γ 、随机系数 α 。由于萤火虫的位置是随机分布的, 所以第一个因素具有很大的随机性。因此我们考虑对后两个参数进行改进, 依据算法的搜索过程设计自适应的参数调整策略来修改参数, 从而控制算法的搜索性能。

首先介绍吸收系数 γ 的调整策略。吸收系数控制着光强度的变化, 在决定算法的收敛速度和行为方面起到非常重要的作用。从理论上来分析, 当 $\gamma \rightarrow 0, \beta = \beta_0$, 那么此时的吸收系数作为一个常数出现, 也就是说萤火虫发出的光不会随着距离增大而出现衰减。当 $\gamma \rightarrow \infty, \beta(r) = \delta(r)$, 即一个狄拉克函数, 它表示萤火虫的吸引力接近于

0, 萤火虫近似的做随机搜索。要使得萤火虫算法在前期能够较快地确定最优解的位置, 后期具备较强的局部搜索能力。我们设计了如下的自适应调节公式:

$$\gamma_i = \gamma_b + e^{-k\sigma_i^2} (\gamma_e - \gamma_b) \quad (8)$$

式中: γ_b —— 初始值;

γ_e —— 最终值, $\gamma_e > \gamma_b$;

k —— 调节参数, $k > 0$ 。

吸收系数随着萤火虫种群位置方程的变化曲线如图 2 所示。

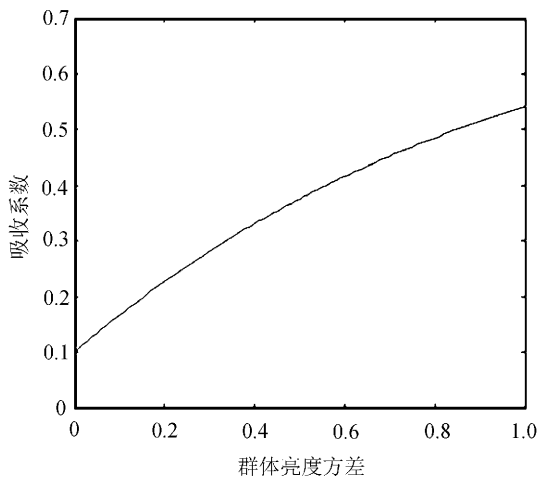


图 2 吸收系数变化关系

然后我们对随机系数参数 α 进行调整。随机系数很大程度上会影响萤火虫的随机运动。要使萤火虫的随机搜索能力增强, 则增大 α , 此时算法有着较强的全局搜索能力; 要使萤火虫的随机搜索能力下降, 则减小 α , 此时算法有着较强的局部搜索能力。前面分析了常见的优化算法搜索策略, 搜索初期要进行大范围的搜索, 以便确定全局最优的位置, 搜索的末期由于距离最优点很近了, 此时需要加强局部搜索能力提高算法的收敛速率。通过以上分析大致可以给出 α 的定性选择策略, 即前期较大, 后期要减小 α 。我们设计了自适应的 α 的改变策略如下:

$$\alpha_i = \alpha_b - \frac{1}{e^{k\sigma_i^2}} (\alpha_e - \alpha_b) \quad (9)$$

式中: α_b —— 初始值;

α_e —— 最终值, $\alpha_e < \alpha_b$;

k —— 调节参数, $k > 0$ 。

随机系数的变化趋势如图 3 所示。

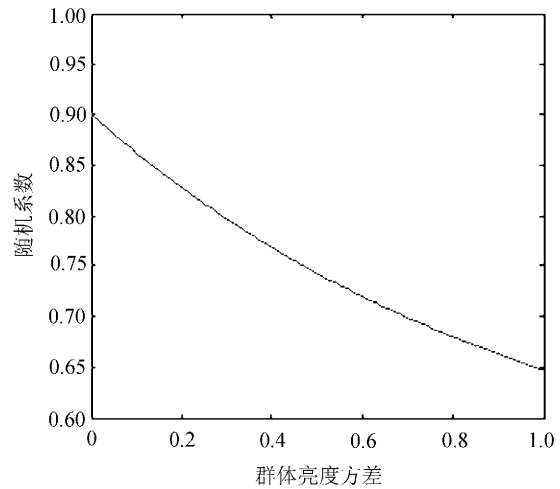


图 3 随机系数变化关系

分析图 3 可知, 随机系数在萤火虫位置方差较大的时候, 近似于线性的由大变小, 最后当种群位置方差较小的时候, 其值保持在最小值附近。这样的变化过程综合考虑了全局搜索和局部搜索能力的平衡, 提高了算法的寻优性能。

2.2.3 随机步长的改进

萤火虫算法一直有着收敛速度较慢的不足。萤火虫算法中若增大随机步长, 则萤火虫自助的搜索能力会增强, 这样萤火虫可以在更广阔的空间上去全局搜索。当萤火虫之间的距离较近的时候, 我们要适当地控制萤火虫的自主搜索能力, 以便加快萤火虫算法的收敛性。

根据以上的定性分析, 随着萤火虫之间的距离来调节随机步长, 给出下面的公式。

$$x_i = x_i + \beta_0 e^{-\gamma_{ij}^2} (x_j - x_i) + 2ar_{ij} \left(\text{rand}() - \frac{1}{2} \right) \quad (10)$$

式中: r_{ij} —— 第 i, j 个萤火虫的间距。

从式(10)分析可知, 萤火虫的随机步长随着 r_{ij} 变化。两个萤火虫距离较远的时候, 萤火虫被吸引力指导的可能性较小, 萤火虫会在一定范围内随机搜索。当萤火虫距离较近的时候, 吸引力占据主导地位, 萤火虫的随机搜索行为减小。从而加速整个改进萤火虫算法的收敛速率。

2.2.4 改进萤火虫算法描述

根据上述改进策略, 结合基本萤火虫算法, 给出了改进萤火虫算法的定义。

首先定义算法的相关参数, 并在一定范围内随机生成萤火虫的初始种群。然后根据目标函数计算萤火虫的亮度, 依据亮度对萤火虫进行排序。

判断算法是否满足收敛条件,若满足,则输出最优解;不满足,则继续进行下一步操作。计算萤火虫的亮度方差,依据亮度方差给出改进萤火虫算法的吸收系数和随机系数。然后再更新萤火虫的位置信息。最后再根据自主飞行策略,确定萤火虫是否进行自主飞行,然后跳转。

3 仿真实验(改进的萤火虫算法性能测试实验)

在经典的 Benchmark 上对文中所提出的改进的萤火虫算法进行测试,仿真实验选取的目标函数见表 1。

表 1 测试函数

函数名	表达式
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$
Rosenbrock	$f_2(x) = \sum_{i=1}^n [100(x_i^2 - x_{i+1}) + (x_i - 1)^2]$
Rastrigin	$f_3(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$
Schwefel	$f_4(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i$

三种算法计算结果对比见表 2。

表 2 三种算法计算结果对比

函数名	GA	FFC	MFCC
Sphere	2×10^{-1}	5.88	1.88×10^{-11}
Rosenbrock	2.34×10^1	6.21×10^1	9.75×10^{-2}
Rastrigin	4.11×10^{-1}	3.29	3.52×10^{-6}
Schwefel	5.67×10^1	8.22×10^{-1}	1.96×10^{-3}

将文中所提的萤火虫算法和基本萤火虫算法(GA 以及 PSO 算法)在航路规划问题的求解结果进行比较。搜索范围定为 $[-100, 100]^n$ 。改进的萤火虫算法的参数设置为: $\beta_0 = 1, \gamma_e = 1, \gamma_b = 0.8, \alpha_e = 0.1, \alpha_b = 1, k = 5$ 。

两种算法在不同维数求解结果对比见表 3。

通过实验验证了萤火虫算法在一般优化问题中的性能,将改进的萤火虫算法和基本的萤火虫算法进行对比实验,来测试该算法的特性。算法种群规模都设置为 10,维数分别取 5、10、20、30、40、50 等 6 种情况进行航路规划实验。若算法能找到规避障碍物的航路,则表明搜索成功。针对每一种情况,萤火虫算法和粒子群算法的停止条件是连续迭代 400 次。分别独立运行算法 60 次,统计每种算法的成功率、航路长度和计算时间。出发点与目标点深度均为 300 m。出发点的经纬度为 (111.10E, 17.07N), 目标点的经纬度为 (112.50E, 16.69N)。

表 3 两种算法在不同维数求解结果对比

维数	算法	成功次数	成功率/%	最优航路长度/m	平均耗时/s
5	粒子群算法	13	28	1 688.87	50.32
	萤火虫算法	2	4	1 688.87	57.17
10	粒子群算法	13	28	1 688.87	56.70
	萤火虫算法	23	48	1 688.87	70.69
20	粒子群算法	31	60	1 688.87	75.54
	萤火虫算法	44	92	1 688.87	101.85
30	粒子群算法	50	100	1 688.87	90.06
	萤火虫算法	50	100	1 688.87	134.34
40	粒子群算法	50	100	1 688.96	107.54
	萤火虫算法	50	100	1 688.87	163.28
50	粒子群算法	50	100	1 689.00	121.98
	萤火虫算法	50	100	1 688.87	196.49

从以上实验数据可知,针对该航路规划问题,改进的萤火虫算法均能成功搜索到从起点到终点的航路,且航路能够成功规避障碍物,从而证明了算法的有效性。除了维数为5的情况外,利用萤火虫算法进行航路规划的成功率均较高,航路长度较短,说明该算法具有较好的性能。但是,在维数较小的时候,两种算法的成功率均较低,这是由于算法的维数对于规划航路的航路点数,较少的维数会导致算法在没有航路点的区域失去避障能力,造成规划航路穿过了障碍物。从平均的计算时间可以看出,萤火虫算法耗时较粒子群算法长,这主要是由于萤火虫算法在执行过程中,每一个萤火虫均要根据比它亮的萤火虫调节自身位置。随着维数的增加,两种算法在规划的速率都会越来越慢,随之而来的计算时间也会大大增加。

4 结 语

在对各种萤火虫算法的研究基础上提出了一种新的萤火虫算法。给出了吸收系数、随机系数和随机步长对萤火虫算法求解结果的不同影响。通过自适应调整吸收系数和随机系数来平衡算法的全局寻优能力和局部寻优能力,然后依靠改进的随机步长选择方法,分别在优化过程的前期和

后期采用不同的随机步长策略来调整萤火虫算法的收敛性。实验结果表明,该算法比基本萤火虫在优化问题的求解上有着更好的效果。证明了该算法在航路规划问题求解中的有效性和准确性。

参考文献:

- [1] 孔姝睿,刘淑芬.基于改进粒子群算法的反舰导弹航路规划研究[J].测控技术,2017,36(11):66-69.
- [2] 申舟,谢文俊,赵晓林.基于人工势场的无人机战场威胁建模研究[J].计算机仿真,2014,2(15):60-64.
- [3] 孙明君,史建国.多无人机航路重规划方法[J].计算机应用,2014,5(29):1480-1482.
- [4] 阎昊,樊兴,夏学知.图结构与Dijkstra算法在无人机航迹规划中的应用[J].火力与指挥控制,2013,4(2):155-157,160.
- [5] 代君,任淑红,王晓璐.小型无人机姿态航向参考系统信息融合算法[J].长春工业大学学报,2016,37(1):52-55.
- [6] Marco A, Contreras-Cruz, Victor Ayala-Ramirez, et al. Mobile robot path planning using artificial bee colony and evolutionary programming[J]. Applied Soft Computing, 2015, 12(30):319-328.
- [7] Hongwei Mo, Lifang Xu. Research of biogeography particle swarm optimization for robot path planning [J]. Neurocomputing, 2015, 12(148):91-99.