

基于参数方差调节萤火虫算法的三维路径规划^①

白永珍

(哈尔滨工程大学 信息与通讯工程学院, 哈尔滨 150001)

摘要: 为了提高萤火虫算法大范围搜索时的速度和精度, 提出了一种参数方差调节萤火虫算法. 首先分析基本萤火虫算法, 在此基础上提出了参数方差调节萤火虫算法的核心思想: 计算种群亮度的方差评估种群的敛散性, 根据进程调节参数, 进而达到改进萤火虫算法的目的, 并给出了算法的实现步骤和流程; 然后在四个优化测试函数中将参数方差调节萤火虫算法与基本萤火虫算法、遗传算法、粒子群算法进行比较和分析, 发现参数方差调节萤火虫算法在测试中能迅速的找到符合精度要求的解, 且成功率是 100%, 具有较好的稳定性, 较之其他算法优势明显; 最后通过构建计算能量消耗的目标函数在有实际背景和地理参数的自主式水下潜器三维路径规划的仿真实验中应用参数方差调节萤火虫算法, 在三维海底环境中规划出符合要求的路, 从而证明了参数方差调节萤火虫算法在三维路径规划中的实用性.

关键词: 改进萤火虫算法; 参数方差调节; 自主式水下潜器; 三维空间; 路径规划

Three-Dimensional Path Planning Based on Parameter Variance Adjustment Firefly Algorithm

BAI Yong-Zhen

(College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China)

Abstract: In order to improve the speed and accuracy of firefly algorithm, a parameter variance adjustment firefly algorithm is proposed. Firstly, based on the analysis of the firefly algorithm the core idea of parameter variance adjustment algorithm firefly is developed: calculate the population variance of luminance assessment of convergence and divergence of populations, and adjust parameters according to the processing, which aims to improve the firefly algorithm, and then presents the algorithm implementation steps and processes. Secondly, compare and analysis the parameter variance adjustment firefly algorithm with basic firefly algorithm, genetic algorithm and particle swarm optimization with four optimization test functions, and find that the parameter variance adjustment firefly algorithm can quickly find the solutions which meet the accuracy requirements of the test, while the success rate is 100%. Compared with other algorithms, it has a clear advantage and good stability. Finally, the parameter variance adjustment firefly algorithm is applied to a real backgrounds and geographical parameters of three-dimension path planning simulation experiments by constructing the objective function to calculate the energy and it meet the required path planning in the three-dimensional ocean environment, which shows the validity of the parameter variance adjustment firefly algorithm.

Key words: improved firefly algorithm; parameter variance adjustment; autonomous underwater vehicle; three-dimensional space; path planning

萤火虫算法(Firefly Algorithm, FA)是由 Xin-she Yang 于 2008 年提出的一种新的生物启发算法. 它来源于对萤火虫群体行为的简化和模拟, 是一种基于群

体搜索的随机优化算法. 该算法提出以后很多学者进行了研究和改进. Xin-She Yang 利用萤火虫算法优化了几种典型的多峰值以及带有奇点或者随机部分的测

^① 基金项目:国家自然科学基金(41176074)

收稿时间:2014-08-29;收到修改稿时间:2014-10-16

试函数,证明了该算法的有效性^[1,2],并且引入 Levy-Flight 构建了 Levy-Flight 萤火虫算法 (Levy-Flight Firefly Algorithm, LFA)证明了在搜寻全局最优值方面 LFA 更高效,成功率也更高^[3,4]. Szymon 等人对萤火虫算法参数的取值进行了研究,给出了萤火虫算法的比较合适的参数范围^[5]. Shadi 等人提出了萤火虫的定向运动 (Directed Movement)^[6], Leandro 等人把混沌序列引入萤火虫算法,用混沌序列调整萤火虫算法的参数和,并用该算法优化可靠性和冗余度分配的基准测试函数,得到了较好的最优解^[7]. Rajdeep 等人把局部搜索过程引入到萤火虫算法,并用其优化等谱弹簧质点系统的误差函数,仿真结果证明该算法能够找到误差函数的多个最优值^[8]关键词库的结合大大提高了信息抽取算法的准确性和通用性,基于 Web 信息抽取的混合交通出行方案生成与表示系统的成功实验也证明了本文提出的 Web 信息抽取算法的实用性.

但是基本萤火虫算法存在不足,特别是当算法针对大搜索范围寻优的时候,其精度比小搜索范围的寻优精度低、收敛速度慢,因此在大搜索范围的优化问题中,基本萤火虫算法的搜索能力还远远不够^[9-12],为此本文提出了一种参数方差调节萤火虫算法.算法的核心思想是通过计算种群亮度的方差评估种群的敛散性,进而定义优化进程;根据进程调节参数,从而达到根据优化需求灵活调节参数的目的,最终能够克服基本萤火虫算法在大范围搜索中寻优精度低、收敛速度慢的问题.

1 参数方差调节萤火虫算法

1.1 基本萤火虫算法

自然界中萤火虫利用特有的闪光信号来吸引异性,借此完成求偶交配及繁殖的使命,少数萤火虫借助闪光信号捕食其他种类的萤火虫或者警告其他捕食者以保护自己.萤火虫算法就是模拟自然界中萤火虫的发光行为,通过随机生成一群萤火虫,萤火虫之间靠亮度进行不同程度的吸引,并在吸引的过程中随机的移动,从而更新萤火虫的位置.

假设待优化问题的解空间为 d 维,则把萤火虫 i 表示为 d 维空间中一个 d 维的解向量 $\mathbf{x}_i(x_{i1}, x_{i2}, \dots, x_{id})$,即萤火虫 i 在 d 维搜索空间中的位置是 \mathbf{x}_i .定义萤火虫的绝对亮度是由目标函数直接决定的,则绝对亮度的大小直接表示了萤火虫所代表

的潜在解的目标函数值的优劣,绝对亮度更大的萤火虫所代表的潜在解更好.定义萤火虫之间的吸引规则为:绝对亮度大的萤火虫会吸引绝对亮度小的萤火虫向它移动,即萤火虫会向代表更好的潜在解的萤火虫移动.基于这种规则,每只萤火虫都会向整个解空间内绝对亮度比它大的萤火虫移动,按照位置更新公式更新自己的位置,即得到一个新的解.绝对亮度最大的萤火虫则随机移动.

萤火虫算法开始时在解空间内初始化一组随机解 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n, n$ 为萤火虫的个数.算法的每次迭代完成所有萤火虫位置的更新,从而得到一组新的解.在这种迭代过程中,随着萤火虫位置的更新,它们的绝对亮度也通过目标函数的作用随之更新.经过一段时间的迭代,萤火虫会聚集在绝对亮度较大的萤火虫周围,即得到目标函数的最优解.

1.2 基本萤火虫算法描述

基本萤火虫算法流程如下:

步骤 1: 初始化算法基本参数.萤火虫的数目 m , 最大吸引度 β_0 , 光吸收系数 γ , 步长因子 α , 最大迭代次数 $maxT$.

步骤 2: 随机初始化萤火虫位置,计算萤火虫目标函数值并作为最大荧光亮度 I_i .

步骤 3: 根据 $I_{ij}(r_{ij}) = I_i e^{-\gamma r_{ij}^2}$ 计算萤火虫 i 对萤火虫 j 的相对亮度 $I_{ij}(r_{ij})$, 得萤火虫 i 对萤火虫 j 的吸引力 $\beta_{ij}(r_{ij})$ 为 $\beta_{ij}(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2}$.

步骤 4: 确定萤火虫之间的距离,萤火虫 i 到萤火虫 j 的距离 r_{ij} 可以定义为笛卡尔距离,即:

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

步骤 5: 由(2)式计算萤火虫 i 被萤火虫 j 吸引并向其移动的位置更新:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \beta_{ij}(r_{ij})(\mathbf{x}_j(t) - \mathbf{x}_i(t)) + \alpha \mathbf{\epsilon}_i \quad (2)$$

步骤 6: 当满足终止条件的时候结束,否则返回到步骤 2 继续进行.

步骤 7: 输出结果,程序结束.整个算法的流程图,如图 1.

1.3 参数方差调节萤火虫算法的提出

首先,萤火虫算法是一种并行的随机搜索算法,合理安排安排算法的参数才能使得萤火虫找到理想的寻优结果.这类参数指的是光吸收系数 γ 和随机常 α .

通过实验分析得出, 在一定的范围内, 光吸收系数 γ 越小, 算法的收敛速度越快, 其对应的寻优精度越差. 同样, 在一定范围下, α 越小, 寻优精度越差, 而过大的 α 会导致算法不稳定. α 越大, 要想达到相同的收敛速度, 与其匹配的 γ 也应该随之增大.

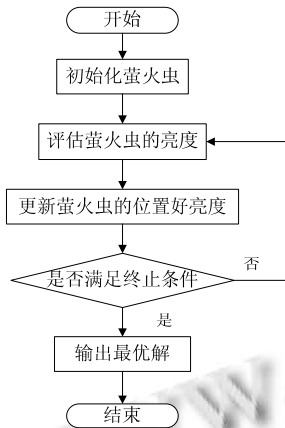


图 1 基本萤火虫算法流程图

其次需要注意的是, 对于优化进程, 即优化初期、后期的判定, 很多文献使用迭代次数, 对于收敛不反弹的算法, 迭代次数可以较为准确的衡量, 但是对于随机搜索的算法, 反弹的情况常有发生, 使用迭代次数就不能准确反映反弹的情况. 但如果能够检测种群的敛散情况, 把种群发散定义为优化初期, 种群收敛定义为优化后期, 让参数随着种群的敛散性变化, 则可以避免算法反弹造成的进程衡量错误的情况. 对于种群敛散性的测量, 认定每个萤火虫的亮度可以唯一确定它的解, 则可以通过求萤火虫种群亮度的方差来反映种群的敛散情况, 从而定义优化进程. 根据进程调节参数, 达到根据优化需求灵活调节参数的目的.

综上, 这种通过计算种群亮度的方差评估种群的敛散性, 根据进程调节参数, 进而达到根据优化需求灵活调节参数的目的的改进的萤火虫算法被命名为参数方差调节萤火虫算法(parameter variance adjustment firefly algorithm, PVFA).

1.4 参数方差调节萤火虫算法描述

参数方差调节萤火虫算法流程描述如下:

步骤 1: 初始化萤火虫算法参数, 包括萤火虫的数目 m , 最大吸引度 β_0 , 光吸收系数 γ , 步长因子 α , 最大迭代次数 $maxT$.

步骤 2: 根据公式(1)确定萤火虫之间的距离, 萤火虫 i 到萤火虫 j 的距离 r_{ij} 可以定义为笛卡尔距离.

步骤 3: 计算归一化算子公式为

$$I = \begin{cases} \max(|I_i - I_{Avg}|), \max(|I_j - I_{Avg}|) > 1 \\ 1, \max(|I_i - I_{Avg}|) \leq 1 \end{cases} \quad (3)$$

其中萤火虫 i 的亮度为 I_i , 整个萤火虫种群的平均亮度为 I_{Avg} .

步骤 4: 计算方差, 公式为

$$\sigma_i^2 = \frac{1}{m} \sum_{i=1}^m \left(\frac{I_i - I_{Avg}}{I} \right)^2 \quad (4)$$

步骤 5: 计算 α 和 γ , 其公式如下

$$\alpha_i = \alpha_b + \exp(-k\sigma_i^2)(\alpha_e - \alpha_b)s \quad (5)$$

$$\gamma_i = \gamma_b + \exp(-k\sigma_i^2)(\gamma_e - \gamma_b) \quad (6)$$

其中下标 b 为起始, e 为终止, k 为调节参数. σ_i^2 反映的是萤火虫种群中萤火虫的收敛程度, σ_i^2 越小, 说明萤火虫种群越集中; σ_i^2 越大, 说明萤火虫种群分布越分散. 种群越分散, 参数越接近起始值, 种群越集中, 参数越接近终止值.

步骤 6: 更新参数方差调节萤火虫算法

$$X(i) = X(i) + k \cdot \beta_{ij}(X(j) - X(i)) + 2\alpha \cdot (rand - 0.5) \quad (7)$$

$$k = Light(j) / Light(i) \quad (8)$$

上述更新公式在第二部分加入了系数 k , k 的意义是当两只萤火虫进行比较, 待更新萤火虫亮度较小, 则计算亮度较大萤火虫 j 与较小萤火虫 i 的亮度商. 系数 k 在亮度有很大差异的情况下可以提高萤火虫吸引力, 有利于快速找到较优解.

步骤 7: 当满足终止条件的时候结束, 否则返回到步骤 2 继续进行.

步骤 8: 输出结果, 程序结束.

整个算法的流程图, 如图 2.

2 算法仿真分析

本测试内容是在四个优化测试函数中对算法进行不同定义域范围的测试^[13,14], 该实验的目的是求出这 4 个测试函数的全局最小值. 测试函数的数学表达式和寻优搜索范围见表 1, 它们的全局最小值都是 0. 测试中在大范围搜索的条件下, 把参数方差调节萤火虫

算法与基本萤火虫算法、遗传算法(Genetic Algorithm, GA)、粒子群算法(Particle Swarm Optimization, PSO)比较,验证参数方差调节萤火虫算法的有效性.待测试优化函数以及三维效果图如图 3-图 5.

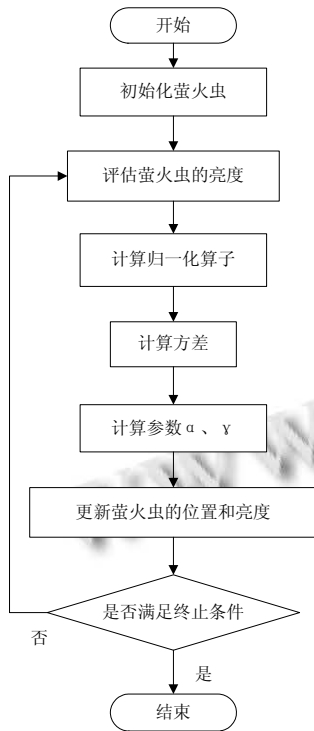


图 2 参数方差调节萤火虫算法流程图

四种算法的参数设置分别为:

1)萤火虫算法: 最大吸引力 $\beta_0 = 1.0$, 光吸收系数

$\gamma = 1.0$ $\alpha = 1.0, \varepsilon = rand - 1/2$, 其中 $rand$ 是在 $[0,1]$ 上服从均匀分布的随机数.

2)参数方差调节萤火虫算法: 最大吸引力 $\beta_0 = 1.0$, 光吸收 $\gamma_b = 1.0, \gamma_e = 0.8, \alpha_e = 0.01, \alpha_b = \sqrt{(\max_range - \min_range)^2 \cdot d}$ 调节参数 $k = 5, \varepsilon = rand - 1/2$, 其中 $rand$ 是在 $[0,1]$ 上服从均匀分布的随机数.

3)遗传算法: 采用二进制编码, 编码的位数设定为 20; 轮盘赌选择方法, 单点交叉方式; 交叉概率为 $p_c = 0.6$, 变异概率为 $p_m = 1/n$. 其中, n 为待定目标函数的维数.

4)粒子群算法: 惯性权重和加速常数采用推荐的参数, 即 $\omega = 0.729, c_1 = c_2 = 1.494$; 最大速度采用为搜索范围的上限, 即 $V_{max} = X_{max}$.

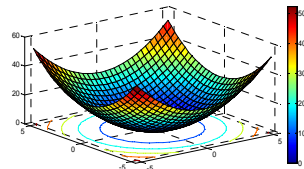


图 3 Sphere 函数三维图

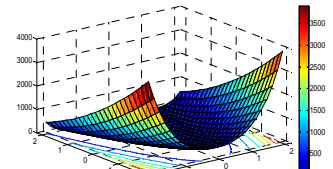


图 4 Rosenbrock 函数三维图

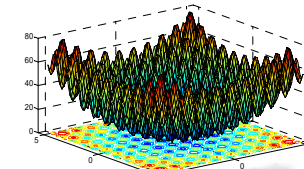


图 5 Rastrigin 函数三维图

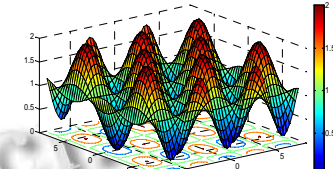


图 6 Griewank 函数三维图

表 1 测试函数表

函数名	数学表达式	搜索范围
Sphere	$f_1(\vec{x}) = \sum_{i=1}^n x_i^2$	$[-5.12, 5.12]^n$
Rosenbrock	$f_2(\vec{x}) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-2.048, 2.048]^n$
Rastrigin	$f_3(\vec{x}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$
Griewank	$f_4(\vec{x}) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-8, 8]^n$

2.1 固定迭代次数优化测试

为了公平地比较算法的优化性能, 本测试在大范

围搜索范围内设置四种算法群体规模为 , 算法进行 2000 次迭代以后终止, 各个测试函数的维数设定为 30

维. 分别用萤火虫算法、参数方差调节萤火虫算法、粒子群算法和遗传算法对以上 4 个测试函数进行寻优, 独立进

行 30 次实验, 统计 30 次实验最终搜索到的函数全局最小值的最优值、最差值、平均值和标准差. 结果见表 2.

表 2 固定迭代次数优化测试表

测试函数及其搜索范围	算法	最优值	最差值	平均值	标准差
Sphere	FA	5.93E+001	4.60E+004	2.21E+004	1.24E+004
	PVAFA	2.32E-016	1.32E-010	1.51E-011	3.05E-011
	PSO	3.79E-046	1.34E-037	6.11E-039	2.54E-038
	GA	4.26E-004	1.33E-002	2.30E-003	2.90E-003
Rosenbrock	FA	5.16E+004	7.75E+009	2.61E+009	2.18E+009
	PVAFA	7.97E-013	1.40E-008	2.01E-009	3.35E-009
	PSO	5.82E-004	6.77E+001	5.18E+000	1.24E+001
	GA	1.16E+000	1.75E+002	7.31E+001	5.01E+001
Rastrigin	FA	6.08E+003	3.94E+004	2.54E+004	9.22E+003
	PVAFA	0.00E+000	3.04E-008	3.45E-009	6.06E-009
	PSO	10.01E-001	1.41E+001	7.96E+000	3.68E+000
	GA	1.90E+001	1.36E+002	6.06E+001	1.41E+002
Griewank	FA	4.26E+002	6.76E+002	5.81E+002	6.60E+001
	PVAFA	4.82E-011	2.67E-007	2.73E-008	5.74E-008
	PSO	0.00E+000	9.35E-002	2.22E-002	2.59E-002
	GA	5.43E-004	3.48E-001	9.43E-002	7.19E-002

从表 2 中可以看到: 对于 sphere 函数, 粒子群算法拥有最优结果: 最优值为 3.79E-046, 标准差为 2.54E-038, 较之其他的算法在此函数的测试中效果最好. 而对于其他的三个优化函数, 参数方差调节萤火虫算法与其他三种算法相比都得到了最好的优化结果, 且标准差较小, 说明算法较稳定. 而基本萤火虫算法在在大范围的测试中, 寻优能力较差, 一直找不到最优解. 由此得出结论: 参数方差调节萤火虫算法相比基本萤火虫算法搜索能力更强, 具有较高的寻优精度.

本测试在大范围对算法指定优化精度目标, 当算法的寻优结果达到精度即停止. 独立运行 30 次, 记录计算 30 次的平均迭代次数. 测试规定每种算法最大迭代次数 10000 次, 若迭代 10000 次仍找不到指定精度对应的解, 则视为失败. 其中测试函数的指定精度 Sphere 为 0.01, Rosenbrock 为 100, Rastrigin 为 100, Griewank 为 0.1, 算法其他参数设置同固定迭代次数实验.

2.2 固定精度优化测试

从表 3 中可以看到: 不管针对哪种测试函数, 参数方差调节萤火虫算法都能迅速的找到符合精度要求的解, 且成功率是 100%, 具有较好的稳定性, 较之其他算法优势明显. 而基本萤火虫算法在测试过程中,

寻优速度慢, 在 Sphere 函数测试中成功率没有达到 100% 只有 95.77%, 而在 Griewank 函数测试中甚至找不到能达到指定精度的解. 粒子群算法虽然在 Sphere 函数和 Rosenbrock 函数的测试中表现良好, 但是在 Rastrigin 函数测试中, 成功率也仅仅只有 33.33%, 在 Griewank 函数测试中只是达到 96.67% 而没有达到和参数方差调节萤火虫算法一样的 100%. 通过以上分析可以得出结论, 在大范围搜索固定精度的条件下, 参数方差调节萤火虫算法相比基本萤火虫算法收敛速度更快, 寻优精度更高, 具有更强的搜索能力.

表 3 算法平均迭代次数和成功率

测试函数	算法	平均迭代次数	成功率(%)
Sphere	FA	4900	95.77
	PVAFA	256	100
	PSO	374	100
	GA	1975	100
Rosenbrock	FA	5488	100
	PVAFA	40	100
	PSO	877	100
	GA	2778	93.55
Rastrigin	FA	9060	100
	PVAFA	35	100
	PSO	300	33.33
	GA	1689	100
Griewank	FA	—	0
	PVAFA	312	100
	PSO	344	96.67
	GA	2155	86.77

3 实验验证

首先对参数方差调节萤火虫算法(PVAFA)与粒子群算法(PSO)算法进行路径规划进行对比,测试三维路径规划的可行性以及有效性.

在真实地形进行测试,航路起始点为(115.2°E, 18.870°N),深度为 370 米,终止点为(112.85°E, 19.210°N),深度为 500 米:对两种算法种群为 10,维数分别是 5、10、20、30、40、50 等五种情况下进行路径规划实验,算法能够找到规避障碍物的目标航路视为搜索成功,算法最大迭代数设为 400. 停止方法为采集每 20 次的路径长度,如果这 20 次之内长度方差小于一个定值则视为路径规划完成. 针对每一种情况,参数方差调节萤火虫和粒子群算法分别运行 50 次,统计每种算法的成功率,最优情况下的航路长度及平均耗时. 统计结果见表 4.

表 4 PVAFA 算法与 PSO 算法路径规划对比

维数	算法	成功率(%)	最优航路长度(米)	平均耗时(秒)
5	PSO	49	2.02E+03	11.3
	PVAFA	15	1.72E+03	12.62
10	PSO	30	2.33E+03	14.77
	PVAFA	5	1.89E+03	14.41
20	PSO	68	2.20E+03	19.31
	PVAFA	89	1.81E+03	48.15
30	PSO	100	1.85E+03	29.63
	PVAFA	100	1.72E+03	57.48
40	PSO	100	1.77E+03	76.73
	PVAFA	100	1.71E+03	78.47
50	PSO	100	1.69E+03	177.97
	PVAFA	100	1.69E+03	128.56

从表中可以发现粒子群随着维数增加路径长度反而减小,这是因为维数多少在一定程度上决定了初始化节点位置的多样性,维数多,位置的多样性好,粒子群算法就能凭借初始化的优势找到最优值. 当粒子群算法维数较少,算法很快在地表找到“最优位置”,并陷入局部最优,此时的路径已经成型较好,局部优化算子只能对其进行局部平滑,整个路径不会有太大变化,而参数方差调节萤火虫算法沟通比粒子群充分,一旦有更好的解立刻更新所有种群,并且经改进后搜索能力更强,因此可以很快获得更优的解,不会过度依赖初始化,因此没有出现明显的长度随维数增加而递减的现象.

表中维数少时,成功率不高,实验发现失败的原因大多在于节点的连线穿过障碍物,由于实验中节点中间避障点只设了 10 个,在路径维数较少的情况下,中间避障点仍然不够多. 因此,在路径规划选取节点数时,选择 30 维以上较好,如果综合考虑计算时间,则最终节点数应该选定为 30.

最后对采用参数方差调节萤火虫算法的水下自主机器人三维路径规划采用真实地形数据并且结合海流海况进行测试,测试参数方差调节萤火虫算法三维路径规划的可行性以及有效性,设置起点为 112°E, 17.160°N, 深度 150 米,终点为 111°E, 16.450°N, 深度 300 米,种群数量 10,维数为 30,迭代次数 200 次.

实验 1,以能量最省为目标进行规划,记录能量消耗和路径长度于表 5,实验效果图如图 7. 实验 2,算法设置不变,在同时存在海流和地形的环境中进行以路径长度最短为目标的三维路径规划,记录能量消耗和路径长度于表 5,实验效果图为图 8. 实验 3,算法设置不变,在同时存在海流和地形的环境中进行以综合考虑能量消耗和路径长度为目标的三维路径规划,记录能量消耗和路径长度于表 5,实验效果为图 10.

表 5 不同指标路径规划统计表

	能量消耗值	路径长度
实验 1	2.5806	1.83E+003
实验 2	396.56	1.28E+003
实验 3	4.5766	1.50E+003

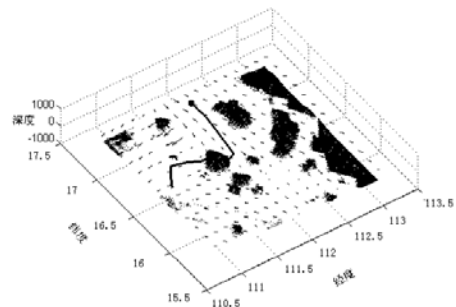


图 7 实验 1 效果图

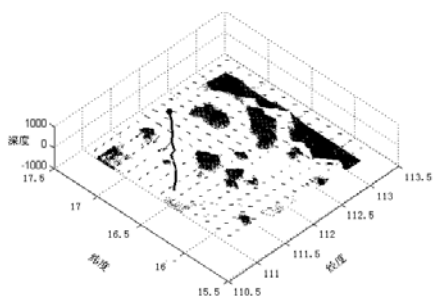


图 8 实验 2 效果图

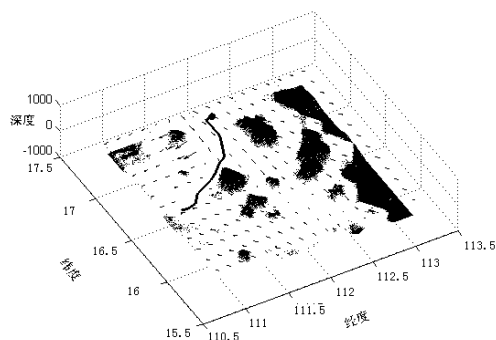


图 9 实验 3 效果图

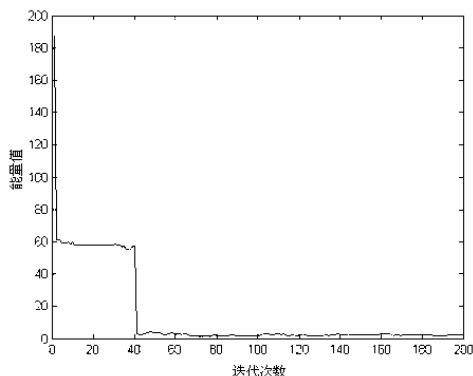


图 10 能量最省路径规划收敛曲线

由实验结果可以看出, 算法基本达到了需要的效果. 图 10 显示了路径的能量变化曲线, 可以看到在 10 代以内算法能量迅速下降, 40 代时能量再次下降, 使路径找到能量消耗更小、局部更加平滑的路径, 40 代之后, 能量值只是小幅变化. 因此可证明, 应用参数方差调节萤火虫算法可以在三维海底环境中规划出符合要求的路径.

4 结语

针对基本萤火虫算法在大范围寻优问题上的搜索能力还远远不够这个问题, 提出了一种参数方差调节

萤火虫算法, 提高了算法在大范围搜索时候的寻优能力和收敛速度. 通过与基本萤火虫算法和其他较为成熟的算法的比较结果可以看出, 参数方差调节萤火虫算法的寻优精度和收敛速度有显著的提高, 且算法稳定, 具有大范围搜索优化的能力. 通过将参数方差调节萤火虫算法应用于自主式水下潜器三维路径规划实验仿真研究, 证明了该算法的实际有效性, 为进一步研究萤火虫算法的改进以及自主式水下潜器的三维路径规划提供了一种新的思路.

本文的局限性在于, 只是在实验的角度论证的光吸收系数对算法的寻优结果和收敛速度的影响的基础上对基本萤火虫算法进行了改进, 相关内容还需要理论研究和证明. 另外, 在进行三维路径规划仿真实验的时候只考虑了自主式水下潜器在海水静态情况下的路径规划, 事实上, 海流的流向、流速会随时间变化, 应考虑时变海流的影响, 构造相应的目标函数并提高算法的运算效率.

参考文献

- 1 Yang XS. Firefly algorithms for multimodal optimization. Stochastic algorithms: foundations and applications. Berlin Heidelberg, Springer. 2009. 169–178.
- 2 Yang XS. Firefly algorithm, stochastic test functions and design optimization. International Journal of Bio-Inspired Computation, 2010, 2(2): 78–84.
- 3 Yang XS. Firefly algorithm, levy flights and global optimization. Research and Development in Intelligent Systems XXVI. London, Springe. 2010. 209–218.
- 4 Yang XS, Deb S. Eagle strategy using lévy walk and firefly algorithms for stochastic optimization. Nature Inspired Cooperative Strategies for Optimization. Berlin Heidelberg, Springer. 2010. 101–111.
- 5 Lukasik S, Żak S. Firefly algorithm for continuous constrained optimization tasks. Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems. Berlin Heidelberg, Springer. 2009. 97–106.
- 6 Farahani S, Abshouri A, Nasiri B, et al. An improved firefly algorithm with directed movement. Proc. of 4th IEEE International Conference on Computer Science and Information Technology, Chengdu. IEEE. 2011. 248–251.

- 7 Coelho LDS, Bernert DLA. A chaotic firefly algorithm applied to reliability redundancy optimization. 2011 IEEE Congress on Evolutionary Computation. New Orleans, IEEE. 2011. 517–521.
- 8 Dutta R, Ganguli R, Mani V. Exploring isospectral spring–mass systems with firefly algorithm. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, 2011, 467(2135): 3222–3240.
- 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.计算机应用,2013,33(3):796–799,805.
- 10 亢少将.萤火虫优化算法的研究与改进[硕士学位论文].广州:广州工业大学,2013.
- 11 黄正新,周永权.自适应步长萤火虫群多模态函数优化算法.计算机科学,2011,31(7):1084–1087.
- 12 李咏梅,周永权,韦军.用于函数优化的层次结构萤火虫群算法.应用科学学报,2012,30(4):391–396.
- 13 王迎菊.混合型人工萤火虫群优化算法及应用研究[硕士学位论文].南宁:广西民族大学,2012.
- 14 刘开丽.粒子群优化算法的研究与应用[硕士学位论文].合肥:安徽大学,2013.