

# 一种改进的麻雀搜索算法

刘睿<sup>1</sup>, 莫愿斌<sup>2,3\*</sup>

1. 广西民族大学 电子信息学院, 广西南宁 530006;
2. 广西民族大学 混杂计算与集成电路设计分析重点实验室, 广西南宁 530006;
3. 广西民族大学 人工智能学院, 广西南宁 530006)

**摘要:**麻雀搜索算法(SSA)作为一种新颖的群体智能优化算法,已被证明具有较好的寻优性能。但由于SSA在某些情况下迭代中后期搜索性减小,种群多样性降低,导致算法存在收敛速度慢、求解精度低、易陷入局部最优解等不足。针对SSA存在的缺陷,融合萤火虫算法(FA)迭代策略,提出了一种加入萤火虫搜索扰动的麻雀搜索优化算法(FSSA)。首先,在麻雀搜索后,利用萤火虫扰动策略对种群中所有个体进行位置更新,使得算法在解空间搜索更加充分,有效避免陷入局部最优,进而提升算法的收敛速度以及收敛精度。其次,通过6个基准测试函数对改进算法与粒子群优化算法(PSO)、鲸鱼优化算法(WOA)、原始的SSA算法进行对比,仿真结果表明该算法能够克服SSA易陷入局部最优的不足,在寻优精度、收敛速度以及鲁棒性等方面均获提升。最后,将FSSA应用于具有14座城市的旅行商问题(TSP)求解,仿真实验对比原始的SSA算法,该算法具有更好的结果,进一步验证了FSSA的寻优能力。

**关键词:**麻雀搜索算法;群体智能优化算法;萤火虫算法;旅行商问题;寻优能力

中图分类号:TP18

文献标识码:A

文章编号:1673-629X(2022)03-0021-06

doi:10.3969/j.issn.1673-629X.2022.03.004

## An Improved Sparrow Search Algorithm

LIU Rui<sup>1</sup>, MO Yuan-bin<sup>2,3\*</sup>

1. School of Electronic Information, Guangxi University for Nationalities, Nanning 530006, China;
2. Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning 530006, China;
3. Institute of Artificial Intelligence, Guangxi University for Nationalities, Nanning 530006, China)

**Abstract:** Sparrow search algorithm (SSA), as a novel swarm intelligence optimization algorithm, has been proved to be effective in searching. However, in some cases, the search ability and population diversity of SSA are reduced in the middle and late iterations, resulting in slow convergence speed, low accuracy and easy to fall into the local optimal solution. Aiming at the above defects of SSA, a sparrow search optimization algorithm with firefly search disturbance (FSSA) is proposed by fusing firefly algorithm (FA) iteration strategy. Firstly, after the sparrow search, the firefly disturbance strategy is used to update the position of all individuals in the population, which makes the algorithm search more fully in the solution space and effectively avoid falling into the local optimum, so as to improve the convergence speed and accuracy of the algorithm. Secondly, six benchmark functions are used to compare the improved algorithm with particle swarm optimization (PSO), whale optimization algorithm (WOA) and the original SSA algorithm. The simulation results show that the proposed algorithm can overcome the shortcoming that SSA is easy to fall into local optimization, and improve the optimization accuracy, convergence speed and robustness. Finally, FSSA is applied to solve the traveling salesman problem (TSP) with 14 cities. The simulation results show that the improved algorithm has better results than the original SSA algorithm, which further verifies the optimization ability of FSSA.

**Key words:** sparrow search algorithm; swarm intelligence algorithm; firefly algorithm; travelling salesman problem; optimization ability

收稿日期:2021-04-05

修回日期:2021-08-11

基金项目:国家自然科学基金项目(21466008);广西自然科学基金项目(2019GXNSFAA185017)

作者简介:刘睿(1996-),男,硕士研究生,CCF会员(E2693G),研究方向为智能信息控制;通讯作者:莫愿斌(1969-),男,博士,教授,研究方向为智能信息控制。

## 0 引言

麻雀搜索算法 (sparrow search algorithm, SSA)<sup>[1]</sup> 是 2020 年由薛建凯和沈波提出的一种群体智能优化算法,该算法模拟了麻雀的觅食行为和反捕食行为,具有实现简单、调节参数少的优点,并且在文献[2-3]中与其他群体智能优化算法进行了对比,在收敛速度、搜索精度、稳定性方面有一定的优势,是一种很有潜力的群智能优化算法。如今,SSA 在支持向量机故障诊断<sup>[4]</sup>、动态路径规划<sup>[5]</sup>、工控入侵检测<sup>[6]</sup>等众多领域得到了广泛的应用。然而,SSA 与其他群体智能算法类似,在优化过程中也存在着易陷入局部最优、收敛精度不足等缺陷,需要进一步的研究和探索。众多学者针对 SSA 存在的不足进行改进,其中李敦桥<sup>[7]</sup>首先采用反向对立学习策略将麻雀种群进行初始化,再结合模拟退火算法产生的 Metropolis 准则对当前解与新解比较替换,提高了算法的寻优能力;吕鑫等人<sup>[8]</sup>受 BSA 算法的启发,对发现者和加入者位置更新公式做出了改进,保障了全局收敛,同时具有一定的种群多样性。上述改进策略均在一定程度上避免了 SSA 陷入局部最优,提升了算法的性能,但 SSA 在寻优精度、收敛速度以及稳定性等方面仍有待进一步改善。

在麻雀搜索算法的基础上,该文提出一种加入萤火虫扰动策略的改进算法。该改进策略主要是在麻雀搜索后,利用萤火虫搜索扰动对麻雀位置进行进一步的优化更新,以提高算法搜索性,增强解的多样性,从而避免算法陷入局部最优。该改进策略简洁有效,通过对 6 个基准测试函数进行仿真实验,对比其他算法,表明该算法较 SSA 寻优性能提升明显,同时将 FSSA 应用于常见 TSP 问题求解,获得了较好的结果,进一步验证了所提算法的寻优性能。

## 1 麻雀搜索算法

SSA 算法的设计灵感来源于鸟类的生物特性,依据麻雀的觅食行为建立的数学模型可归类为发现者-加入者模型,并加入了预警的机制,随机选取种群中的部分麻雀作为意识到危险的麻雀建立反捕食机制。每只麻雀是搜索空间中的一个粒子,可以代表问题的一个解。设定在种群中有发现者 (Producer) 和加入者 (Scrounger) 两种角色,发现者具备较高的适应度值,为加入者提供觅食即粒子搜索的方向和区域,引导种群进行搜索。麻雀个体在两种身份之中的变化处于动态平衡,且适应度值的变化决定了加入者是否跟随发现者。种群遭遇危险时,将触发反捕食行为,当预警值大于安全值时,加入者会追随发现者进行位置更新前往更安全区域觅食,在边缘的麻雀会向着群体中心更新自己的位置以规避危险,而位于种群中心的麻雀则

会进行随机游走。

麻雀集合如下所示:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}$$

其中,  $n$  表示麻雀的规模,  $d$  表示变量的维数。

麻雀的适应度值如下所示:

$$F_X = \begin{bmatrix} f([x_{1,1} & x_{1,2} & \cdots & x_{1,d}]) \\ f([x_{2,1} & x_{2,2} & \cdots & x_{2,d}]) \\ \vdots & \vdots & \vdots & \vdots \\ f([x_{n,1} & x_{n,2} & \cdots & x_{n,d}]) \end{bmatrix}$$

其中,  $f([x_{i,d}])$  表示个体适应度值。

发现者的位置更新公式为:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot \text{iter}_{\max}}\right) & R_2 < ST \\ X_{i,j}^t + Q \cdot L & R_2 \geq ST \end{cases} \quad (1)$$

其中,  $t$  为当前的迭代数,  $\text{iter}_{\max}$  为最大迭代数,  $\alpha$  为  $(0,1]$  之间均匀分布的随机数,  $R_2 \in [0,1]$ ,  $ST \in [0.5,1]$  分别表示预警值与安全值。  $Q$  为服从正态分布的随机数,  $L$  为  $1 \times d$  的矩阵,其中每个内部元素均为 1。当  $R_2 < ST$  时,麻雀种群的周围不存在危险或者捕食者,发现者可以广泛搜索食物,从而引导其他个体获得更高的适应度值。当  $R_2 \geq ST$  时,预警者发现危险靠近,立即释放危险信号,麻雀种群将做出反捕食行为,调整搜索策略,飞到安全的区域。

加入者的位置更新公式为:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{i^2}\right) & i > \frac{n}{2} \\ X_{\text{pbest}}^{t+1} + |X_{i,j}^t - X_{\text{pbest}}^{t+1}| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \quad (2)$$

其中,  $X_{\text{worst}}^t$ 、 $X_{\text{pbest}}^{t+1}$  分别为种群中在第  $t$  次迭代与第  $t+1$  次迭代中,麻雀个体在所处的全局最差位置与局部最优位置,  $A$  是一个矩阵内部元素为 1 或 -1 的多维矩阵,且  $A^+ = A^T (AA^T)^{-1}$ 。当  $i > n/2$  时,表明第  $i$  只麻雀处于最差的位置,根据算法设定规则,需要飞往其他的地方进行觅食从而获得更高的能量,其他情况则是第  $i$  只麻雀在当前发现者  $X_{\text{pbest}}^{t+1}$  的位置周围随机进行觅食。

麻雀种群中意识到危险的麻雀位置更新公式如下:

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{gbest}}^t + \beta \cdot |X_{i,j}^t - X_{\text{gbest}}^t| & f_i > f_g \\ X_{i,j}^t + k \cdot \left(\frac{|X_{i,j}^t - X_{\text{worst}}^t|}{(f_i - f_w) + \varepsilon}\right) & f_i = f_g \end{cases} \quad (3)$$

其中,  $X_{\text{gbest}}^t$  为当前麻雀种群的全局最优位置,  $\beta$  是步长控制参数, 服从均值为 0、方差为 1 的正态分布的随机数,  $\varepsilon$  为一个极小的常数, 用于避免式中分母出现 0 的情况。  $K \in [-1, 1]$  用来控制麻雀的运动方向。  $f_i$  为当前个体  $i$  的适应度值,  $f_g, f_w$  为当前麻雀种群的最优适应度值与最差适应度值。当  $f_i > f_g$  时, 表示该麻雀处于种群的边缘, 极易受到捕食者的攻击。当  $f_i = f_g$  时, 表明该麻雀位于种群的中间并且意识到了危险的存在, 该麻雀向其他个体进行靠拢, 保护自己避免受到捕食者的攻击。

## 2 加入萤火虫搜索扰动的麻雀搜索算法

### 2.1 萤火虫扰动策略

萤火虫算法 (firefly algorithm, FA)<sup>[9]</sup>, 是由剑桥学者 Yang 提出, 通过模拟萤火虫的发光行为实现位置优化。基本的麻雀搜索算法存在的缺陷如下: 算法在解空间搜寻目标时存在过早收敛, 即容易“早熟”致使最优解精度不足, 发现者在位置更新迭代后期容易直接跳跃至当前极值附近, 导致搜索范围不够从而困于局部最优, 同时寻优精度也会受到影响。为了改良以上所提到的不足之处, 主要在麻雀搜索后, 引入萤火虫算法的迭代策略, 将萤火虫扰动策略应用于算法之中, 借助萤火虫发光吸引的特性, 寻找邻域结构内位置较优的个体, 增强解的多样性, 通过增加随机扰动项, 扩展搜索区域, 提升算法的全局探索能力, 有利于引导算法跳出局部最优, 同时进一步更新麻雀位置, 使得种群整体搜索更加充分, 有利于提升收敛精度。萤火虫算法涉及的数学建模如下所示。

萤火虫的相对荧光亮度  $I$  为:

$$I = I_0 \cdot \exp(-\gamma r_{ij}) \quad (4)$$

其中,  $I_0$  为最大荧光亮度, 是距离为零处自身的荧光亮度, 且适应度值越优的个体具备的  $I_0$  值越大;  $\gamma$  为光强吸收系数, 体现萤火虫个体随着距离的增加、传播媒介的吸收影响下荧光的减弱效果;  $r_{ij}$  为各萤火虫之间的空间距离。

萤火虫的吸引度  $\beta$  为:

$$\beta = \beta_0 \cdot \exp(-\gamma r_{ij}^2) \quad (5)$$

其中,  $\beta_0$  为最大吸引度, 即个体光源处的吸引度。

萤火虫扰动位置更新公式如下:

$$x_i = x_i + \beta \cdot (x_j - x_i) + \alpha \cdot [\text{rand}(\cdot) - 1/2] \quad (6)$$

其中,  $x_i$  和  $x_j$  为麻雀  $i$  和  $j$  的空间位置,  $\alpha$  为步长控制参数,  $\text{rand}(\cdot) \in [0, 1]$  为服从均匀分布的随机因子。

### 2.2 FSSA 算法步骤

FSSA 算法流程如下:

Step1: 初始化种群, 设置种群大小  $N$ , 最大迭代次

数  $\text{Iteration}$ , 发现者比例, 意识到危险的麻雀比例、安全阈值等参数。

Step2: 计算当前麻雀种群个体的适应度值并进行排序, 找出当前最优值以及最差值。

Step3: 按比例选取适应度值较优的麻雀作为发现者, 根据公式(1)更新发现者位置。

Step4: 种群中剩下的作为加入者, 根据公式(2)更新加入者位置。

Step5: 按比例在种群中随机选取部分个体作为意识到危险的麻雀, 根据公式(3)更新意识到危险的麻雀位置, 并计算新的适应度值, 如果比当前最优值好就进行更新操作。

Step6: 引入萤火虫算法, 此时种群中搜索粒子等价于萤火虫个体; 依据上一步迭代后的位置, 计算适应度函数值作为各萤火虫最大荧光亮度  $I_0$ , 并根据公式(4)、(5)计算萤火虫的荧光亮度  $I$  与吸引度  $\beta$ , 决定种群的搜索方向。

Step7: 利用萤火虫扰动公式(6)对种群进行位置更新, 并对处于最优位置的萤火虫进行随机扰动。

Step8: 计算适应度值, 保留最优个体位置。

Step9: 检验是否满足停止条件, 若满足则算法结束输出最优结果, 否则转向 Step2。

## 3 基准函数测试

为了验证 FSSA 的寻优性能, 通过选取文献[10-12]中 6 个不同类型的基准测试函数, 将 FSSA 与粒子群优化算法 (particle swarm optimization, PSO)<sup>[13]</sup>、鲸鱼优化算法 (whale optimization algorithm, WOA)<sup>[14]</sup>、SSA 算法进行仿真对比实验。

所有计算过程重复进行 30 次, 结果取最好值、最差值、平均值以及标准差作为算法性能的评判标准, 实验的集成开发环境为 Matlab(R2018b), 操作系统为 win10, 64 位。设置四种算法迭代次数为 1 000, 种群数量为 100, 表 1 为基准测试函数相关信息, 表 2 记录了四种算法的实验结果。为了直观地体现算法的寻优效果, 图 1 给出了基准函数的搜索空间以及对应四种算法的收敛曲线。

由表 2 对单峰测试函数、多峰测试函数以及低维多峰测试函数的实验结果可反映出四种算法关于局部开发能力、全局探索能力以及跳出局部最优的能力的强弱; 从单峰测试函数  $f_1 \sim f_3$  的测试结果来看, FSSA 表现出了良好的收敛速度以及寻优精度, 可以准确寻优到理想值, 优于其他算法, 相比之下改进算法有明显提升。对于多峰测试函数  $f_4$  和  $f_5$ , FSSA 能够有效地逃脱局部最优并找到全局最优解, 对比其他算法, 寻优结果数量级有较大提升, 在取得较好的平均值同时标准

差值较小,说明 FSSA 具有良好的鲁棒性,进一步说明了改进算法的有效性。对于低维多峰测试函数  $f_6$ ,虽然 FSSA 较其他算法寻优效果仅有小幅提升,但是

FSSA 具备的寻优精度最高,通过对平均值与标准差的对比,FSSA 的寻优稳定性明显优于其余算法。

表 1 基准测试函数

Benchmark functions	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^n x_i^2$	[-100, 100]	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	[-10, 10]	0
$f_3(x) = \max_i \{  x_i , 1 \leq i \leq n \}$	[-100, 100]	0
$f_4(x) = \sum_{i=1}^n -x_i \sin \sqrt{ x_i }$	[-500, 500]	418.982 n
$f_5(x) = \frac{\pi}{n} \{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	[-50, 50]	0
$f_6(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	[-65, 65]	1

表 2 仿真实验结果

Benchmark functions	Algo	Opt	Worst	Mean	Std
$f_1$	PSO	1.864	16.473	3.686	1.728
	WOA	1.191 e-207	2.028 e-188	1.987 e-195	0
	SSA	0	0	0	0
	FSSA	0	0	0	0
$f_2$	PSO	0.171	11.741	4.229	3.782
	WOA	5.027 e-119	7.258 e-108	1.933 e-113	1.483 e-113
	SSA	0	2.737 e-258	5.641 e-259	0
	FSSA	0	0	0	0
$f_3$	PSO	0.575	2.240	1.092	0.716
	WOA	5.809 e-04	75.257	16.235	20.587
	SSA	0	0	0	0
	FSSA	0	0	0	0
$f_4$	PSO	-10 075.505	-8 421.981	-8 712.382	684.277
	WOA	-12 569.219	-8 569.405	-11 569.417	675.091
	SSA	-9 872.621	-7 271.246	-9 139.652	614.377
	FSSA	-27 482.582	-21 324.028	-24 783.661	658.072
$f_5$	PSO	1.889 e-01	1.511	0.245	0.268
	WOA	1.913 e-05	1.885 e-04	5.268 e-05	3.377 e-05
	SSA	2.403 e-13	5.343 e-09	4.042 e-10	2.137 e-10
	FSSA	2.234 e-18	4.732 e-15	5.441 e-16	4.275 e-16
$f_6$	PSO	0.998	3.507	1.442	0.557
	WOA	0.998	2.982	1.764	1.362
	SSA	0.998	12.670	1.779	2.737
	FSSA	0.998	0.998	0.998	0

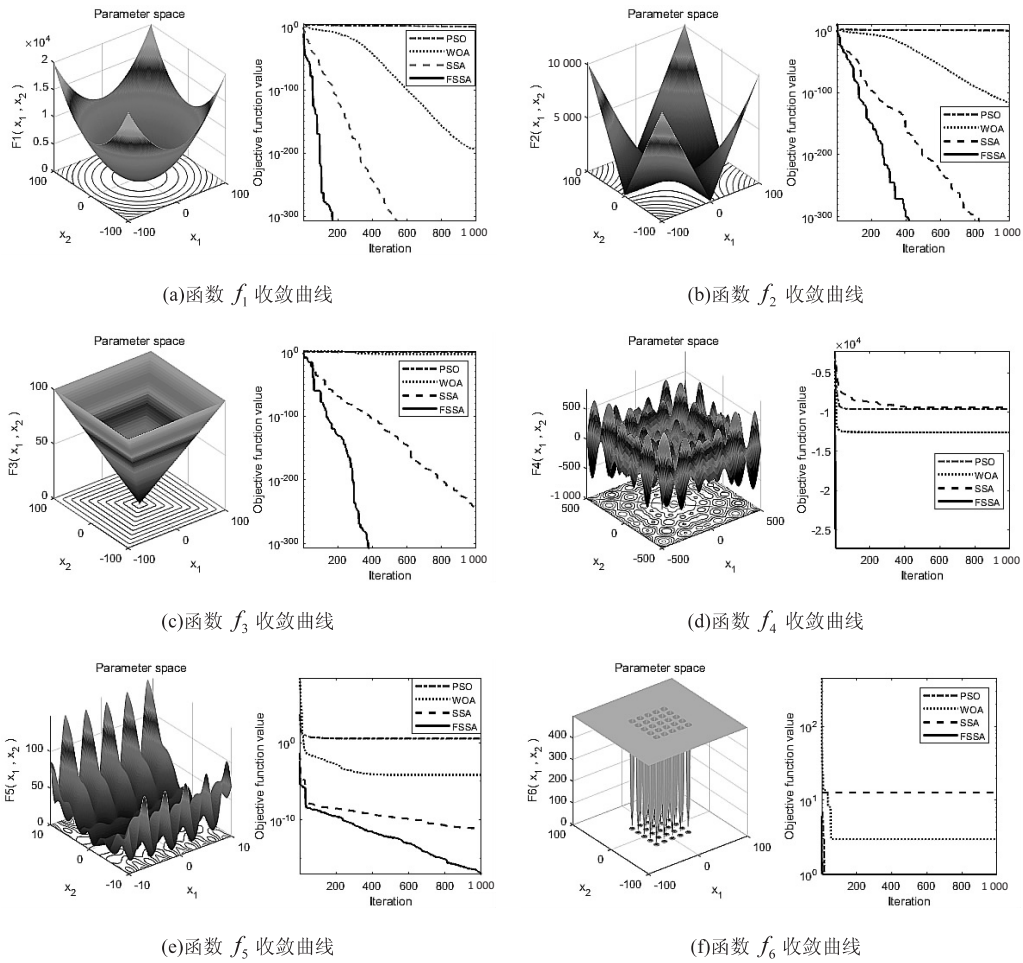


图1 搜索空间及优化曲线

## 4 FSSA 在 TSP 问题中的应用

### 4.1 TSP 问题描述

旅行商问题(traveling salesman problem, TSP), 又译为旅行推销员问题等, 最早由 Dantzig 等人于 1959 年提出<sup>[15]</sup>, 是组合优化中一道著名的 NP 难问题, 对其的求解也一直是学术界关注的热点。

TSP 问题可描述为: 假设一位旅行商需要途经若干城市以推销其货物, 从其中一座城市出发, 遍历每座城市一次最后回到出发点, 且各城市之间位置已知, 则如何选取、规划路线才能使得旅行商总行程最短。

设各座城市为  $V = (v_1, v_2, \dots, v_n)$ , 对城市的访问

顺序为  $T = (t_1, t_2, \dots, t_n)$ , 每座城市之间的距离为  $d_{(v_i, v_j)}$ , 则 TSP 问题的目标函数为:

$$F = \min \sum_{i=1}^{n-1} d_{(v_i, v_{i+1})} + d_{(v_n, v_1)} \quad (7)$$

### 4.2 仿真实验

该文采用具有 14 座城市的 TSP 问题进行测试, 各城市的初始坐标见表 3, 设置种群数量为 100, 最大迭代次数为 1 000, 分别对 SSA 与 FSSA 进行 20 次独立测试, 表 4 记录了两种算法求解 TSP 问题的最好值、最差值和平均值。图 2 为各城市初始位置分布情况与 FSSA 求解 TSP 的最优解, 图 3 为 FSSA、SSA 收敛曲线对比。

表3 14 座城市的坐标

城市	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x	16.47	16.47	20.09	22.39	25.23	22.00	20.47	17.20	16.30	14.05	16.53	21.52	19.41	20.09
y	96.10	94.44	92.54	93.37	97.24	96.05	97.02	96.29	97.38	98.12	97.38	95.59	97.31	94.55

表4 两种算法的测试结果

算法	最好值	最差值	平均值
SSA	32.05	37.23	34.62
FSSA	30.98	31.05	30.98

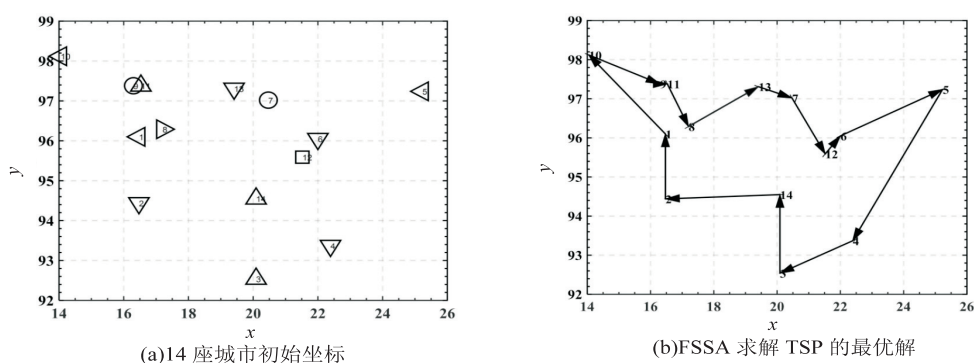


图 2 TSP 问题仿真实验

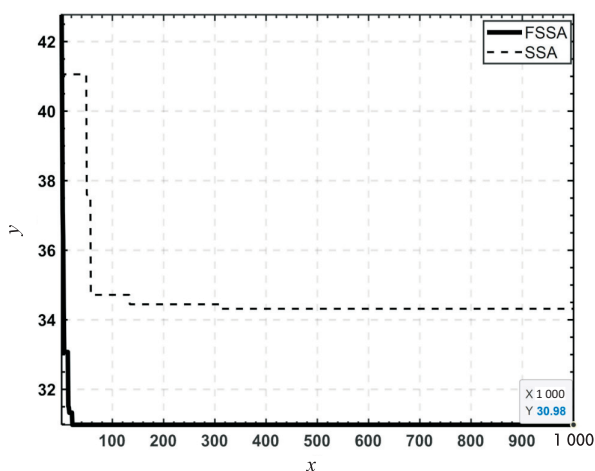


图 3 两种算法求解 TSP 的收敛曲线

从表 4 的实验结果可以看出, FSSA 算法在求解 TSP 问题时可以在最大迭代次数内寻找到最优解, 且随着实验次数的增多平均值仍近似等于最优值, 体现了 FSSA 良好的鲁棒性; 通过图 3 可看出, SSA 的迭代收敛次数近似于 400 次, 而 FSSA 在迭代少于 50 次时即完成收敛产生最优路径长度 30.98, 说明该文所提算法具有较快的收敛速度与较高的收敛精度; 对比最终求解结果, FSSA 较 SSA 在获得的最优解上提升了 3.34%, 最差解上提升了 16.60%, 平均值上提升了 10.51%, 进一步说明了该算法的有效性。

## 5 结束语

该文提出了一种加入萤火虫搜索扰动的改进麻雀搜索算法, 通过在麻雀搜索后利用萤火虫扰动策略对麻雀位置进一步的优化更新, 提高了算法的搜索性, 丰富了解的多样性, 同时采用 6 种不同的基准测试函数验证了 FSSA 的寻优性能, 与 PSO、WOA 和 SSA 相比, FSSA 具有更好的收敛速度与收敛精度。

最后通过将 FSSA 应用于具有 14 座城市的 TSP 路径规划, 取得了较好的结果, 进一步验证了 FSSA 的寻优能力。

## 参考文献:

- [1] 薛建凯. 一种新型的群智能优化技术的研究与应用[D]. 上海: 东华大学, 2020.
- [2] XUE J, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, 8(1): 22-34.
- [3] 李雅丽, 王淑琴, 陈倩茹, 等. 若干新型群智能优化算法的对比研究[J]. *计算机工程与应用*, 2020, 56(22): 1-12.
- [4] 马晨佩, 李明辉, 巩强令, 等. 基于麻雀搜索算法优化支持向量机的滚动轴承故障诊断[J]. *科学技术与工程*, 2021, 21(10): 4025-4029.
- [5] 赵普凡, 王伟. 基于动态路径规划的车载自组织网络贪婪路由协议[J]. *国外电子测量技术*, 2020, 39(11): 46-51.
- [6] 杜 晔, 王子萌, 黎妹红. 基于优化核极限学习机的工控入侵检测方法[J]. *信息安全*, 2021, 21(2): 1-9.
- [7] 李敦桥. 一种混合麻雀搜索算法[J]. *电脑知识与技术*, 2021, 17(5): 232-234.
- [8] 吕 鑫, 慕晓冬, 张 钧. 基于改进麻雀搜索算法的多阈值图像分割[J]. *系统工程与电子技术*, 2021, 43(2): 318-327.
- [9] YANG X S. Firefly algorithm[J]. *Nature-Inspired Meta-Heuristic Algorithms*, 2008(20): 79-90.
- [10] YAO X, LIU Y. Evolutionary programming made faster[J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(2): 82-102.
- [11] DIGALAKIS J G, MARGARITIS K G. On benchmarking functions for genetic algorithms[J]. *International Journal of Computer Mathematics*, 2001, 77(4): 481-506.
- [12] YANG X S. Firefly algorithm, stochastic test functions and design optimisation[J]. *International Journal of Bio Inspired Computation*, 2010, 2(2): 78-84.
- [13] KENNEDY J. Particle swarm optimization[C]//*Proceedings of 1995 IEEE int. conf. neural networks*. Perth, Australia: IEEE, 2011: 1942-1948.
- [14] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [15] FLOOD M M. The traveling-salesman problem[J]. *Operations Research*, 1956, 4(1): 61-75.