

# 多策略融合的改进麻雀搜索算法及其 AGV 路径规划应用

乐明皓 李凌

(沈阳化工大学信息工程学院 沈阳 110142)

**摘要:**针对麻雀搜索算法(sparrow search algorithm,SSA)存在依赖初始种群分布,易于陷入局部最优解,以及迭代后期种群多样性减少等问题,提出一种多策略融合的改进麻雀搜索算法(improved sparrow search algorithm,ISSA)。首先,采用 Sobol 序列初始化种群,保证初始种群的多样性;其次,分别引入随机反向学习策略和螺旋觅食策略改进发现者位置更新公式和加入者位置更新公式,以增强算法的全局搜索能力和跳出局部最优解的能力;最后,引入柯西变异对可能陷入局部最优解的麻雀进行扰动。实验选取 9 个标准测试函数进行性能测试,实验结果表明,改进后的算法性能有较大提升。将 ISSA 应用于 AGV(automated guided vehicle)路径规划,在 3 种地图环境下分别可以达到最优值 13.135 6、28.834 5 和 44.364 9,寻优能力和稳定性较原算法有较大提升。

**关键词:**麻雀搜索算法;随机反向学习;螺旋觅食;测试函数;AGV 路径规划

中图分类号: TP301 文献标识码:A 国家标准学科分类代码: 510.1040

## Improved sparrow search algorithm with multi strategy fusion and application in AGV path planning

Le Minghao Li Ling

(College of Information Engineering, Shenyang University of Chemical Technology, Shenyang 110142, China)

**Abstract:** This paper proposes an improved sparrow search algorithm (ISSA) with multi strategy fusion to address the issues of dependence on initial population distribution, susceptibility to local optima, and reduced population diversity in the later stages of iteration in sparrow search algorithm (SSA). Firstly, the population is initialized using Sobol sequences to ensure the diversity of the initial population. Secondly, random reverse learning strategy and spiral foraging strategy are introduced to improve the discoverer position update formula and the joiner position update formula, respectively, to enhance the algorithm's global search ability and ability to jump out of local optimal solutions. Finally, introducing Cauchy variation to perturb sparrows that may fall into local optima. Nine standard test functions were selected for performance testing in the experiment, and the results showed that the improved algorithm had a significant improvement in performance. Applying ISSA to Automated Guided Vehicle (AGV) path planning can achieve optimal values of 13.135 6, 28.834 5, and 44.364 9 in three map environments, respectively. The optimization ability and stability of the algorithm are significantly improved compared to the original algorithm.

**Keywords:** sparrow search algorithm; random reverse learning; spiral foraging; test function; AGV path planning

## 0 引言

群体智能优化算法通常是根据生物的觅食,繁衍等行为或群体事物的特性所设定的。近年来,新型群体智能优化算法发展迅速,各种新型群体智能优化算法层出不穷,

如灰狼优化算法(grey wolf optimizer,GWO)<sup>[1]</sup>、鲸鱼优化算法(whale optimization algorithm, WOA)<sup>[2]</sup>、蛇优化算法(snake optimization, SO)<sup>[3]</sup>、蜣螂优化算法(dung beetle optimizer, DBO)<sup>[4-5]</sup>等。此类算法应用范围广泛,可被应用于 AGV(automated guided vehicle)路径规划<sup>[6]</sup>、故障诊

断<sup>[7]</sup>、图像处理<sup>[8]</sup>等领域。由于传统的算法多数存在易于陷入局部最优解,全局搜索能力不足等缺点,围绕传统算法进行改进已成为当今研究的热点。

麻雀搜索算法(sparrow search algorithm, SSA)是Xue等<sup>[9]</sup>于2020年提出的一种新型群体智能优化算法,其具有结构简单,实现容易,调节参数少等优点,但仍存在依赖初始种群分布,跳出局部最优解能力弱,迭代后期种群多样性减少,全局搜索能力减弱等问题。针对这些问题,许多学者对麻雀搜索算法进行了改进。李江华等<sup>[10]</sup>提出了一种混合多策略改进的麻雀搜索算法,引入精英反向学习策略初始化种群,在加入者位置中引入Circle映射参数与余弦因子,并采用自适应机制随机选择麻雀个体,对其使用Levy飞行进行扰动,改进后的算法在寻优速度和收敛速度等方面有较大的提升。回立川等<sup>[11]</sup>提出了一种多策略改进的麻雀搜索算法,引入领域重心反向学习策略初始化种群,引入Levy飞行改进发现者位置更新公式,引入自适应权重和Levy飞行改进加入者位置更新公式,改进后的算法在稳定性,收敛性和搜寻精度方面有更优的效果。Yang等<sup>[12]</sup>提出了一种基于混沌映射和t分布变异的自适应麻雀搜索算法,引入Sine混沌映射初始化种群,引入动态自适应权重策略改进发现者位置更新公式,最后引入t分布自适应进行扰动,改进后的算法可以获得更好的寻优能力,收敛精度和收敛速度。苏莹莹等<sup>[13]</sup>提出了一种自适应混合策略麻雀搜索算法,引入Tent混沌映射和精英种群策略初始化种群,引入周期性自适应收敛因子改进发现者位置更新公式,调整加入者和预警者位置更新方式,最后引入多项式变异扰动,改进后的算法具有更好的寻优能力。

本文提出一种多策略融合的改进麻雀搜索算法(ISA-  
SA),引入Sobol序列初始化种群,提升种群的多样性,使  
算法在迭代初期能进行广泛的搜索;引入随机反向学习策  
略改进发现者位置更新公式,提升算法的全局搜索能力;  
引入金枪鱼群优化算法螺旋觅食策略改进加入者位置更  
新公式,提升算法的局部搜索能力;最后引入柯西变异策  
略对可能陷入局部最优解的麻雀个体进行扰动,帮助其跳  
出局部最优解。将改进后的算法使用9个标准测试函数  
测试其性能,并应用于AGV路径规划。通过实验可以看出,  
改进后算法的寻优能力、收敛性和稳定性都有较大提升,  
验证了其在解决AGV路径规划问题时的优越性与可行性。

## 1 麻雀搜索算法原理

麻雀搜索算法<sup>[9]</sup>是受到自然界中麻雀的捕食与反哺食行为的启发而提出的一种群体智能优化算法。在麻雀种群中,根据每个麻雀的能量储备(即适应度值)高低,可将麻雀分为发现者和加入者两类。发现者一般占整个种群的10%~20%,且具有较高的能量储备,负责搜寻食物。剩余的麻雀则为加入者,加入者中能量储备较高的麻

雀会时刻监视发现者并围绕其觅食,能量储备较低的麻雀则会飞往别处觅食。一个种群中发现者和加入者的比例是固定的,但每个麻雀所属的类别是根据能量储备不断变化的。同时,麻雀具有非常高的警惕性,有一部分的麻雀会成为预警者,一般占整个种群的10%~20%,当其感知到危险时,就会迅速向安全区域移动。

麻雀搜索算法的数学模型如下:

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

式(1)表示一个麻雀种群,  $n$  表示种群中麻雀数量,  $d$  表示待求解问题的维数。

全体麻雀的适应度值如下:

$$F = \begin{bmatrix} f([x_{1,1}, x_{1,2}, \dots, x_{1,d}]) \\ f([x_{2,1}, x_{2,2}, \dots, x_{2,d}]) \\ \vdots \\ f([x_{n,1}, x_{n,2}, \dots, x_{n,d}]) \end{bmatrix} \quad (2)$$

发现者的位置更新公式如下:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(-\frac{i}{\alpha \cdot iter_{\max}}\right), & R_2 < ST \\ X_{i,j}^t + Q \cdot L, & R_2 \geq ST \end{cases} \quad (3)$$

式中:  $t$  表示当前迭代次数;  $iter_{\max}$  表示最大迭代次数;  $\alpha$  和  $Q$  为随机数,  $\alpha$  在  $(0,1]$  范围内,  $Q$  为服从正态分布;  $L$  为一个  $1 \times d$  的全 1 矩阵;  $R_2$  在  $[0,1]$  范围内, 表示预警值;  $ST$  在  $[0.5,1]$  范围内, 表示安全值。

当  $R_2 < ST$  时, 周围环境较为安全, 发现者可以进行广泛的搜寻。当  $R_2 \geq ST$  时, 有一部分麻雀预感到了危险, 发现者随即迅速飞往其他区域觅食。

加入者的位置更新公式如下:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst} - X_{i,j}^t}{i^2}\right), & i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L, & \text{其他} \end{cases} \quad (4)$$

式中:  $X_{worst}$  表示在第  $t$  次迭代中, 在第  $j$  维处于最差位置的元素;  $X_p^{t+1}$  表示在第  $t+1$  次迭代中, 发现者在第  $j$  维处于最优位置的元素;  $A^+ = A^T (AA^T)^{-1}$ ,  $A$  为一个  $1 \times d$  的矩阵, 矩阵中的元素随机赋值 1 或 -1。

当  $i > \frac{n}{2}$  时, 表示第  $i$  个麻雀的能量储备很低, 需要飞往其他地方觅食。其他麻雀则围绕处于最优位置的麻雀觅食。

预警者的位置更新公式如下:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t|, & f_i > f_g \\ X_{i,j}^t + K \cdot \left( \frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \epsilon} \right), & f_i = f_g \end{cases} \quad (5)$$

式中:  $X_{best}^t$  表示在第  $t$  次迭代中, 在第  $j$  维处于最优位置

的元素;  $\beta$  和  $K$  为随机数,  $\beta$  服从标准正态分布, 用作控制步长,  $K$  在  $[-1, 1]$  范围, 控制麻雀移动的方向;  $f_i$  为第  $i$  个麻雀的适应度值;  $f_g$  为当前全局最优适应度值;  $f_w$  为当前全局最差适应度值;  $\epsilon$  为一个大于 0 的极小值, 以防止分母为 0。

当  $f_i > f_g$  时, 麻雀正处于种群边缘位置, 预感到危险时迅速靠近种群中心。当  $f_i = f_g$  时, 麻雀正处于种群中心, 预感到危险时迅速靠近其他麻雀。

## 2 改进麻雀搜索算法

### 2.1 Sobol 序列初始化麻雀种群

在群体智能优化算法中, 种群的初始位置分布对算法的寻优能力和收敛速度有着至关重要的作用, 初始位置分布均匀的种群能使算法较快的进入最优区间, 进而提升算法的搜索精度。基本麻雀搜索算法采用随机方式生成麻雀种群, 设置种群数量为 100, 上限和下限分别为 0 和 1, 在三维空间中通过随机方式生成的麻雀种群如图 1 所示。从图 1 可以看出, 由于随机生成的麻雀种群在空间分布无规律, 导致麻雀种群在空间中分布不均匀, 存在种群聚集的现象。这会限制麻雀种群的搜索范围, 使麻雀种群易于陷入局部最优解, 影响算法的收敛速度和寻优精度。为降低种群分布不均匀对算法性能造成的影响, 本文引入 Sobol 序列初始化麻雀种群。

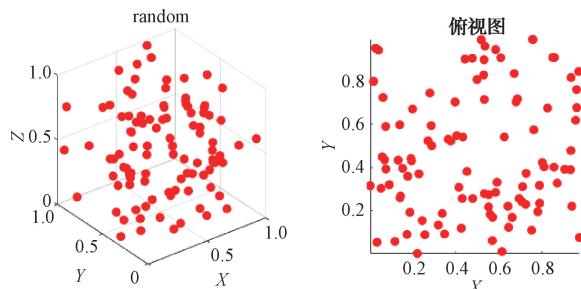


图 1 随机方式生成麻雀种群

Fig. 1 Random generation of sparrow population

Sobol 序列<sup>[14]</sup>是一种低差异序列, 序列中相邻点间的距离较小, 并且在多维空间中均匀分布。设置种群数量为 100, 上限和下限分别为 0 和 1, 在三维空间中通过 Sobol 序列生成的麻雀种群如图 2 所示。从图 2 可以看出, 麻雀种群在空间中分布较为均匀, 覆盖面积广泛。

通过 Sobol 序列生成麻雀种群公式如下:

$$x_i = lb + K_n \times (ub - lb) \quad (6)$$

式中:  $lb$  为搜索空间的下限;  $ub$  为搜索空间的上限;  $K_n$  为通过 Sobol 序列产生的数, 在  $[0, 1]$  范围内。

### 2.2 发现者位置更新方式改进

在基本麻雀搜索算法的发现者位置更新公式中, 当  $R_2 < ST$  时, 可以发现每一维的值都在减小, 随着迭代次数的增加, 种群极有可能向原点靠近。这会导致种群大量

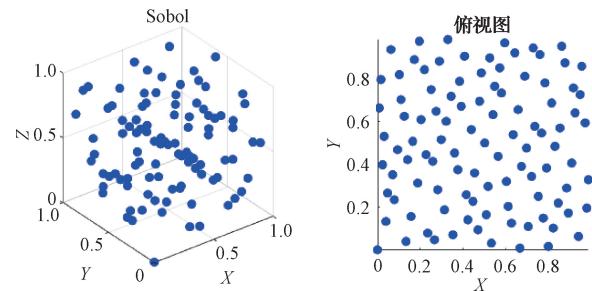


图 2 Sobol 序列生成麻雀种群

Fig. 2 Sobol sequence generated sparrow population

聚集在原点附近, 种群多样性减少, 不利于算法进行广泛的全局搜索。本文引入随机反向学习策略对发现者位置更新方式进行改进, 增强算法的全局搜索能力。

随机反向学习策略是 Long 等<sup>[15]</sup>在反向学习的基础上提出的, 该策略在反向解中引入随机因子, 扩大了反向解的分布范围, 增加了种群的多样性, 提升了算法全局搜索的能力, 其示意图如图 3 所示。随机反向学习公式如下:

$$x_{rand} = lb + ub - rand \times x_i \quad (7)$$

式中:  $lb$  为搜索空间的下限;  $ub$  为搜索空间的上限;  $rand$  为  $[0, 1]$  区间内的随机数。

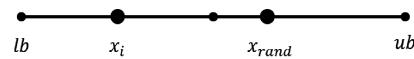


图 3 随机反向学习示意图

Fig. 3 Schematic diagram of random reverse learning

同时引入贪婪机制与式(3)上半部分进行比较, 选取适应度值较小的策略更新麻雀个体位置, 如式(8)所示。在防止原先已接近全局最优解的麻雀跳出该位置的同时, 扩大麻雀的搜索范围, 丰富麻雀种群的多样性。改进后的发现者位置更新公式如式(9)所示。

$$X_{better} = \begin{cases} X_{origin} = X_i^t \cdot \exp\left(-\frac{i}{\alpha \cdot iter_{max}}\right), \\ f(X_{origin}) \leq f(X_{reverse}) \\ X_{reverse} = lb + ub - rand \times X_i^t, \\ f(X_{origin}) > f(X_{reverse}) \end{cases} \quad (8)$$

$$X_i^{t+1} = \begin{cases} X_{better}, & R_2 < ST \\ X_i^t + Q \cdot L, & R_2 \geq ST \end{cases} \quad (9)$$

### 2.3 加入者位置更新方式改进

根据加入者位置更新公式可以看出, 适应度值较小的加入者会迅速靠近处于最优位置的麻雀个体, 聚集在最优个体附近, 易于陷入局部最优解。本文受到金枪鱼群优化算法中的螺旋觅食策略启发, 改进加入者位置更新方式, 从而增强加入者的局部搜索能力。

金枪鱼群优化算法(tuna swarm optimization, TSO)<sup>[16]</sup>是基于金枪鱼群的觅食策略提出的一种新型智能优化算

法,其中一种觅食策略是螺旋觅食策略。当金枪鱼群遇到猎物时,会形成紧密的螺旋追逐猎物。同时,每个金枪鱼个体之间还会相互交换信息,紧紧跟随其相邻个体,并随着最优个体进行位置移动。其位置更新公式如下:

$$\mathbf{X}_i^{t+1} = \begin{cases} \boldsymbol{\alpha}_1 \cdot (\mathbf{X}_{\text{best}}^t + \beta \cdot |\mathbf{X}_{\text{best}}^t - \mathbf{X}_i^t|) + \\ \boldsymbol{\alpha}_2 \cdot \mathbf{X}_i^t, & i = 1 \\ \boldsymbol{\alpha}_1 \cdot (\mathbf{X}_{\text{best}}^t + \beta \cdot |\mathbf{X}_{\text{best}}^t - \mathbf{X}_i^t|) + \\ \boldsymbol{\alpha}_2 \cdot \mathbf{X}_{i-1}^t, & i = 2, 3, \dots, n \end{cases} \quad (10)$$

$$\begin{cases} \boldsymbol{\alpha}_1 = a + (1-a) \times \frac{t}{t_{\max}} \\ \boldsymbol{\alpha}_2 = (1-a) - (1-a) \times \frac{t}{t_{\max}} \\ \beta = e^{bt} \times \cos(2\pi b) \\ l = e^{3\cos((t_{\max} + \frac{1}{t})^{-1}) \pi} \end{cases} \quad (11)$$

式中:  $\boldsymbol{\alpha}_1$  和  $\boldsymbol{\alpha}_2$  是控制个体向最优个体和前一个个体移动趋势的权重向量;  $a$  为在初期追随最佳个体和前一个个体的程度的常量;  $t$  为当前迭代次数;  $t_{\max}$  为最大迭代次数。

若最优个体未能搜寻到猎物,金枪鱼群则会随机选择目标点进行螺旋搜索,扩大种群的搜索范围。

改进后加入者位置更新公式如下:

$$\mathbf{X}_i^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}} - \mathbf{X}_i^t}{i^2}\right), & i > \frac{n}{2} \\ \boldsymbol{\alpha}_1 \cdot (\mathbf{X}_{\text{best}}^t + \beta \cdot |\mathbf{X}_{\text{best}}^t - \mathbf{X}_i^t|) + \\ \boldsymbol{\alpha}_2 \cdot \mathbf{X}_i^t, & i = m \\ \boldsymbol{\alpha}_1 \cdot (\mathbf{X}_{\text{best}}^t + \beta \cdot |\mathbf{X}_{\text{best}}^t - \mathbf{X}_i^t|) + \\ \boldsymbol{\alpha}_2 \cdot \mathbf{X}_{i-1}^t, & \text{其他} \end{cases} \quad (12)$$

式中:  $m$  表示适应度最好的发现者麻雀所在位置。

## 2.4 柯西变异扰动

在迭代过程中,一些麻雀个体可能已陷入局部最优解,从而造成前后两次迭代适应度值变化较小,将这类麻雀通过式(13)进行标定。

$$f(\mathbf{x}_i^{t+1}) - f(\mathbf{x}_i^t) < \frac{\sum_{i=1}^n f(\mathbf{x}_i^{t+1}) - f(\mathbf{x}_i^t)}{n} \quad (13)$$

式中:  $f(\mathbf{x}_i^{t+1})$  表示第  $i$  只麻雀在第  $t+1$  次迭代时的适应度值;  $f(\mathbf{x}_i^t)$  表示第  $i$  只麻雀前一次迭代时的适应度值;  $n$  表示麻雀个数。

当麻雀个体的适应度值变化程度小于麻雀种群平均适应度值变化程度时,该麻雀可能陷入了局部最优解。为帮助该类麻雀跳出局部最优解,引入柯西变异对其进行扰动。

柯西变异即为柯西分布,是一种连续型概率分布。如图 4 所示,柯西分布相比于高斯分布,峰值较小,衰减速度平缓,尾部衰减速度较慢,变异范围更加均匀,具有较好的扰动能力<sup>[17]</sup>。标准柯西变异表达式如式(14)所示。

$$\text{newx}_i^{t+1} = \mathbf{x}_i^{t+1} + \mathbf{x}_i^{t+1} \times \tan\left(\pi\left(u - \frac{1}{2}\right)\right) \quad (14)$$

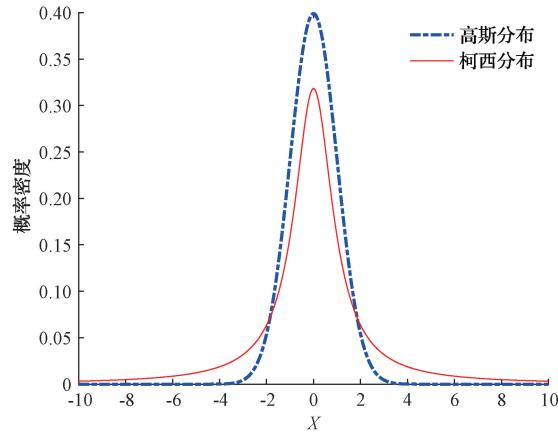


图 4 高斯分布和柯西分布概率曲线

Fig. 4 Probability curves of Gaussian distribution and Cauchy distribution

同样,为防止已接近全局最优位置的麻雀跳出该区域,采用贪婪机制判断柯西变异后的麻雀是否处于更优位置,如式(15)所示。

$$x_i^{t+1} = \begin{cases} \text{newx}_i^{t+1}, & f(\text{newx}_i^{t+1}) < f(\text{oldx}_i^{t+1}) \\ \text{oldx}_i^{t+1}, & f(\text{newx}_i^{t+1}) \geq f(\text{oldx}_i^{t+1}) \end{cases} \quad (15)$$

综上所述,改进后的算法流程如图 5 所示。

## 3 算法性能测试

### 3.1 实验设置

为了验证多策略融合的改进麻雀搜索算法具有良好的寻优能力和收敛能力,本文选取 9 个标准测试函数  $f_1 \sim f_9$  进行实验,如表 1 所示。其中,  $f_1 \sim f_4$  为单峰函数,  $f_5 \sim f_7$  为多峰函数,  $f_8 \sim f_9$  为固定维多峰函数,单峰函数常用于评估算法的全局搜索能力,多峰函数常用于评估算法跳出局部最优解的能力。并将多策略融合的改进麻雀搜索算法与 SSA<sup>[9]</sup>、TSO<sup>[16]</sup>、粒子群优化算法(particle swarm optimization, PSO)<sup>[18]</sup>、GWO<sup>[1]</sup>、WOA<sup>[2]</sup>进行对比实验,各算法参数设置如表 2 所示。为了确保实验的公平性,设置种群数量为 30,最大迭代次数为 500,每种算法独立运行 30 次。实验采用 MATLAB 2022a 平台,系统为 Windows10,处理器为 AMD Ryzen7 5800H with Radeon Graphics 3.20 GHz。运行结果如表 3 所示,最优值越接近测试函数最小值,平均值越接近最优值,标准差越小,则算法的寻优能力和稳定性更强。再单独运行每个测试函数,得出各个算法的收敛曲线,如图 6 所示。

对于测试函数  $f_1 \sim f_7$ ,由于通过 Sobol 序列生成的初始种群直接命中最优位置,为了展示改进算法的寻优能力,对处在最优位置的个体进行扰动。

### 3.2 实验结果分析

通过表 3 可以看出,在单峰函数中,ISSA 的各项指标均能达到最优,相较于 SSA 有明显的提升,结合图 6(a)~

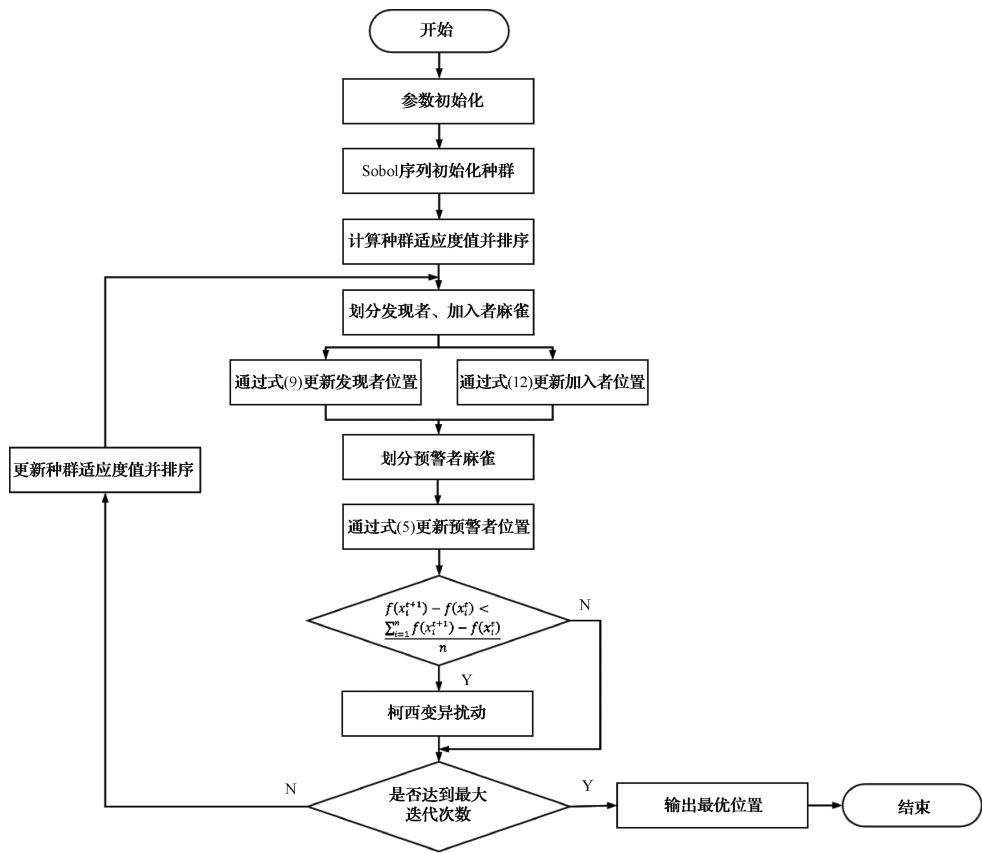


图 5 改进麻雀搜索算法流程

Fig. 5 Improved sparrow search algorithm flow chart

表 1 测试函数

Table 1 Test Functions

测试函数	维度	范围	最小值
$f_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	[-10, 10]	0
$f_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	30	[-100, 100]	0
$f_4(x) = \max_i \{  x_i , 1 \leq i \leq n \}$	30	[-100, 100]	0
$f_5(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5, 12.5, 12]	0
$f_6(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$	30	[-32, 32]	0
$f_7(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
$f_8(x) = \left[ \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$	2	[-65.536, 65.536]	0.9980
$f_9(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5364

表2 算法参数设置

Table 2 Algorithm parameter settings

算法	参数
ISSA	$PD = 0.2, SD = 0.1, ST = 0.8, a = 0.7$
SSA	$PD = 0.2, SD = 0.1, ST = 0.8$
TSO	$z = 0.05, a = 0.7$
PSO	$c_1 = c_2 = 2, w_{max} = 0.9, w_{min} = 0.2$
GWO	$a_{initial} = 2, a_{final} = 0$
WOA	$b = 1, a_{initial} = 2, a_{final} = 0$

(d)可以看出,ISSA 的收敛速度快,较其他算法有较大程度的提升。这表明 ISSA 具有较强的全局搜索能力。在

多峰函数中,对于函数  $f_5, f_6, f_7$ , ISSA、SSA 和 TSO 3 种算法在每次运行时都能达到测试函数的最小值,结合图 6(e)~(g)可以看出,ISSA 的收敛速度较 SSA 和 TSO 有大幅度的提升,在迭代初期就可以收敛到全局最优;对于函数  $f_8$ ,ISSA 的平均值和标准差虽然略差于 TSO,但是和其他算法相比仍有较大提升,结合图 6(h)可以看出,ISSA 的收敛速度也较快;对于函数  $f_9$ ,ISSA 各项指标均为最优,结合图 6(i)可以看出,ISSA 的收敛速度更快。这表明 ISSA 跳出局部最优解的能力较强。

由上述分析可知,ISSA 在面对不同类型的测试函数时所表现出的良好寻优能力,具有较为优异的全局搜索能力,跳出局部最优解的能力强,稳定性和收敛性较好。

表3 测试函数运行结果

Table 3 Test function run results

测试函数	算法	最优值	平均值	标准差
$f_1$	ISSA	0	0	0
	SSA	$9.7876 \times 10^{-267}$	$1.4418 \times 10^{-65}$	$7.8973 \times 10^{-65}$
	TSO	$7.2059 \times 10^{-263}$	$1.6847 \times 10^{-228}$	0
	PSO	$6.5708 \times 10^{-6}$	$3.4699 \times 10^{-4}$	$9.0604 \times 10^{-4}$
	GWO	$4.6107 \times 10^{-29}$	$1.2687 \times 10^{-27}$	$1.8096 \times 10^{-27}$
	WOA	$3.6248 \times 10^{-86}$	$9.3569 \times 10^{-73}$	$2.9857 \times 10^{-72}$
$f_2$	ISSA	0	$1.7706 \times 10^{-234}$	0
	SSA	0	$1.8678 \times 10^{-43}$	$1.0219 \times 10^{-42}$
	TSO	$1.8398 \times 10^{-129}$	$1.7622 \times 10^{-116}$	$8.6235 \times 10^{-116}$
	PSO	$2.4030 \times 10^{-3}$	$3.6413 \times 10^{-2}$	$6.0615 \times 10^{-2}$
	GWO	$1.2067 \times 10^{-17}$	$1.1883 \times 10^{-16}$	$1.4970 \times 10^{-16}$
	WOA	$2.2129 \times 10^{-58}$	$4.6257 \times 10^{-51}$	$2.3374 \times 10^{-50}$
$f_3$	ISSA	0	0	0
	SSA	0	$9.7396 \times 10^{-51}$	$5.3346 \times 10^{-50}$
	TSO	$4.9083 \times 10^{-245}$	$4.9114 \times 10^{-213}$	0
	PSO	28.7596	83.7718	35.5284
	GWO	$1.2068 \times 10^{-8}$	$9.4175 \times 10^{-6}$	$3.6263 \times 10^{-5}$
	WOA	$1.5343 \times 10^4$	$4.3260 \times 10^4$	$1.4413 \times 10^4$
$f_4$	ISSA	0	$6.4213 \times 10^{-229}$	0
	SSA	0	$6.6825 \times 10^{-40}$	$3.3564 \times 10^{-39}$
	TSO	$8.4566 \times 10^{-129}$	$1.0161 \times 10^{-113}$	$3.8639 \times 10^{-113}$
	PSO	$6.7340 \times 10^{-1}$	1.0571	$2.1083 \times 10^{-1}$
	GWO	$1.7505 \times 10^{-7}$	$1.1586 \times 10^{-6}$	$1.3638 \times 10^{-6}$
	WOA	$2.3316 \times 10^{-1}$	44.8826	26.6064
$f_5$	ISSA	0	0	0
	SSA	0	0	0
	TSO	0	0	0
	PSO	28.8738	56.7235	14.8475

续表

测试函数	算法	最优值	平均值	标准差
$f_6$	GWO	0	2.811 4	4.643 4
	WOA	0	$3.7896 \times 10^{-15}$	$2.0756 \times 10^{-14}$
	ISSA	0	0	0
	SSA	0	0	0
	TSO	0	0	0
	PSO	$2.6857 \times 10^{-3}$	$2.1871 \times 10^{-1}$	$4.6166 \times 10^{-1}$
	GWO	$7.4607 \times 10^{-14}$	$1.0315 \times 10^{-13}$	$1.6598 \times 10^{-14}$
	WOA	0	$3.5527 \times 10^{-15}$	$2.4685 \times 10^{-15}$
	ISSA	0	0	0
$f_7$	SSA	0	0	0
	TSO	0	0	0
	PSO	$2.1445 \times 10^{-6}$	$1.0603 \times 10^{-2}$	$1.0199 \times 10^{-2}$
	GWO	0	$4.6071 \times 10^{-3}$	$6.9780 \times 10^{-3}$
	WOA	0	$5.4634 \times 10^{-3}$	$2.9924 \times 10^{-2}$
	ISSA	$9.9800 \times 10^{-1}$	1.064 1	$3.6225 \times 10^{-1}$
	SSA	$9.9800 \times 10^{-1}$	8.136 3	5.554 4
	TSO	$9.9800 \times 10^{-1}$	$9.9800 \times 10^{-1}$	$3.5709 \times 10^{-16}$
	PSO	$9.9800 \times 10^{-1}$	3.005 5	2.216 3
$f_8$	GWO	$9.9800 \times 10^{-1}$	3.818 6	3.052 7
	WOA	$9.9800 \times 10^{-1}$	3.713 8	3.542 3
	ISSA	-10.536 4	-10.536 4	$2.4240 \times 10^{-15}$
	SSA	-10.536 4	-9.995 6	1.650 1
	TSO	-10.536 4	-10.536 4	$3.2821 \times 10^{-15}$
	PSO	-10.536 4	-9.521 4	2.341 5
	GWO	-10.536 2	-10.264 4	1.481 3
	WOA	-10.536 1	-7.917 9	2.881 0

### 3.3 Wilcoxon 秩和检验

为了更进一步展示 ISSA 的性能,采用 Wilcoxon 秩和检验判断 ISSA 与 SSA、TSO、PSO、GWO、WOA 算法之间是否存在显著差异,以展现出 ISSA 性能的提升程度。Wilcoxon 秩和检验是一种非参数统计方法,设置显著性水平为 5%,并计算  $p$  值。当  $p < 0.05$  时,表示两种算法之间存在显著性差异。当  $p > 0.05$  时,表示两种算法性能相当,不存在显著性差异。NAN 表示两种算法运行结果相同,无法进行显著性分析。

设置相同的实验条件,Wilcoxon 秩和检验结果如表 4 所示。对于函数  $f_5$ 、 $f_6$ 、 $f_7$ ,ISSA 无法与 SSA 和 TSO 进行显著性分析,是因为 3 种算法每次迭代都可以搜寻到函数最小值。除此之外,ISSA 在大部分情况下和其他算法相比存在显著性差异。这表明 ISSA 的性能较原算法有较大提升,并且相较于其他算法也展现出了更为优异的性能,进一步验证了 ISSA 的优越性。

### 4 AGV 路径规划应用

#### 4.1 地图建模

本文使用栅格法建立地图模型,栅格法具有容易实现,可拓展性强等特点,常用于构建路径规划地图。构建  $N \times N$  大小的地图,并均匀分割成  $N$  行  $N$  列大小相等的正方形栅格,白色栅格表示空白区域,路径规划时可以从该区域经过。黑色栅格表示障碍物,路径规划时需要避开该区域。本文分别构建了  $10 \times 10$ 、 $20 \times 20$  和  $30 \times 30$  大小的栅格地图,如图 7~9 所示,用于研究多策略融合的改进麻雀搜索算法在简单和复杂环境下的寻优能力。

#### 4.2 仿真分析

将多策略融合的麻雀搜索算法与基本麻雀搜索算法进行对比实验,以验证多策略融合的麻雀搜索算法在进行路径规划时较基本麻雀搜索算法性能有较大提升。两种算法参数设置同表 2,种群数量为 30,迭代 200 次,分别在

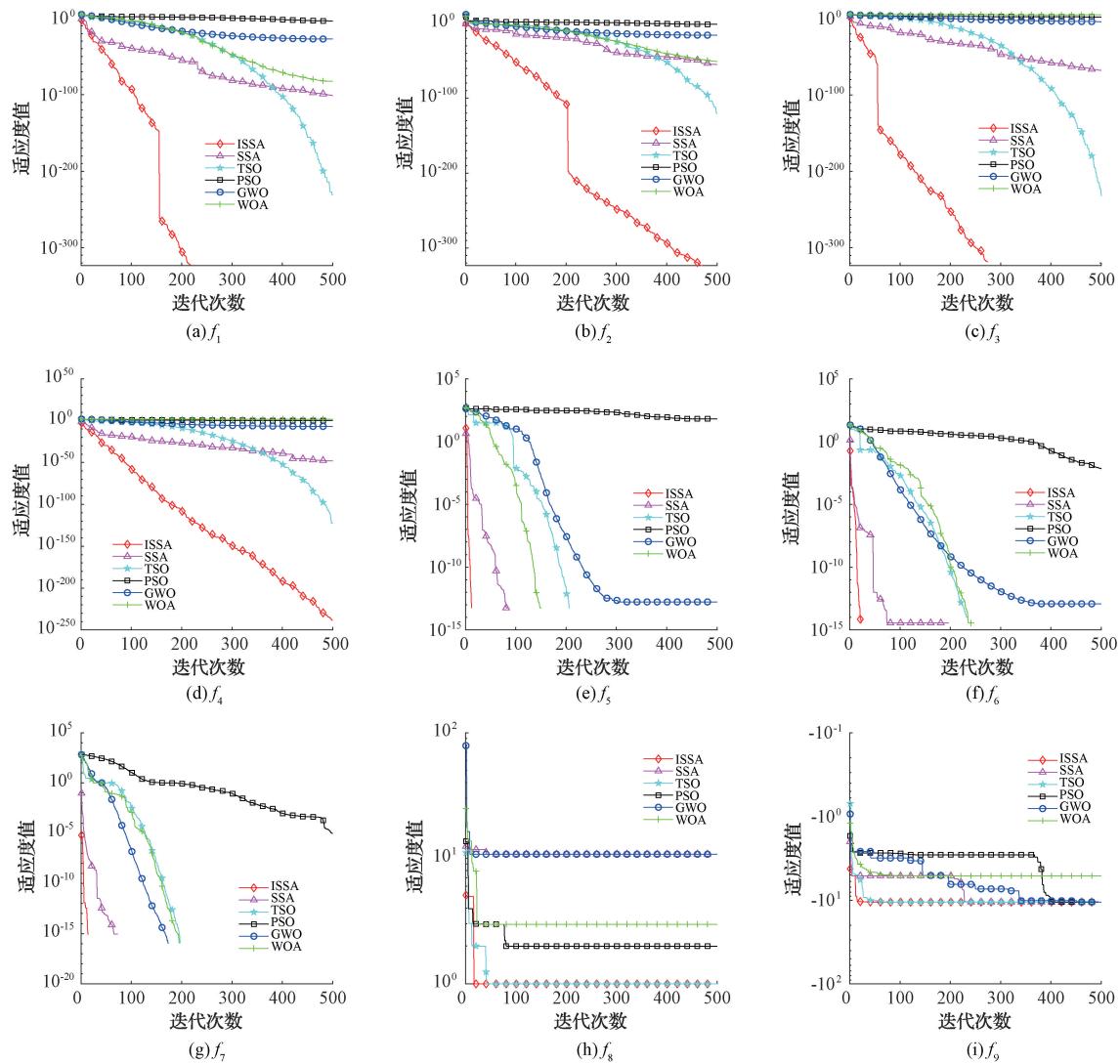


图 6 不同测试函数收敛曲线  
Fig. 6 Convergence curve of different test functions

表 4 Wilcoxon 秩和检验  
Table 4 Wilcoxon rank sum test

测试函数	SSA	TSO	PSO	GWO	WOA
$f_1$	$1.21 \times 10^{-12}$				
$f_2$	$5.57 \times 10^{-10}$	$3.02 \times 10^{-11}$	$3.02 \times 10^{-11}$	$3.02 \times 10^{-11}$	$3.02 \times 10^{-11}$
$f_3$	$4.57 \times 10^{-12}$	$1.21 \times 10^{-12}$	$1.21 \times 10^{-12}$	$1.21 \times 10^{-12}$	$1.21 \times 10^{-12}$
$f_4$	$4.30 \times 10^{-8}$	$3.02 \times 10^{-11}$	$3.02 \times 10^{-11}$	$3.02 \times 10^{-11}$	$3.02 \times 10^{-11}$
$f_5$	NAN	NAN	$1.21 \times 10^{-12}$	$4.47 \times 10^{-12}$	$3.34 \times 10^{-1}$
$f_6$	NAN	NAN	$1.21 \times 10^{-12}$	$1.16 \times 10^{-12}$	$3.63 \times 10^{-9}$
$f_7$	NAN	NAN	$1.21 \times 10^{-12}$	$3.13 \times 10^{-4}$	$3.34 \times 10^{-1}$
$f_8$	$5.02 \times 10^{-10}$	$6.03 \times 10^{-6}$	$3.28 \times 10^{-6}$	$1.19 \times 10^{-11}$	$3.05 \times 10^{-11}$
$f_9$	$1.16 \times 10^{-8}$	$3.51 \times 10^{-2}$	$6.90 \times 10^{-1}$	$9.31 \times 10^{-12}$	$9.31 \times 10^{-12}$

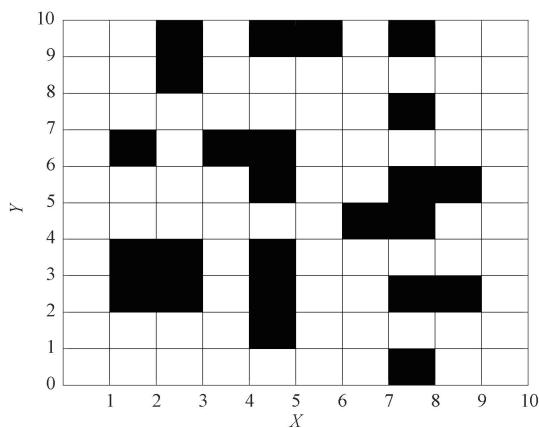


图 7 10×10 棚格地图

Fig. 7 10×10 grid map

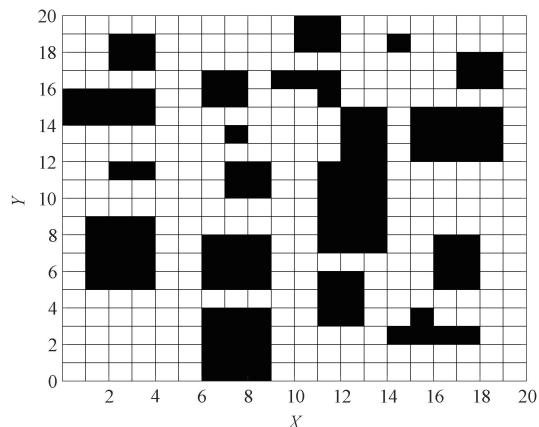


图 8 20×20 棚格地图

Fig. 8 20×20 grid map

图 7~9 的棚格地图环境下运行,运行结果如表 5 所示。

从表 5 可以看出,在 10×10 大小棚格地图环境中,ISSA 和 SSA 的最优值相同,但 ISSA 的标准差和平均值都优于 SSA。在 20×20 和 30×30 大小棚格地图环境中,

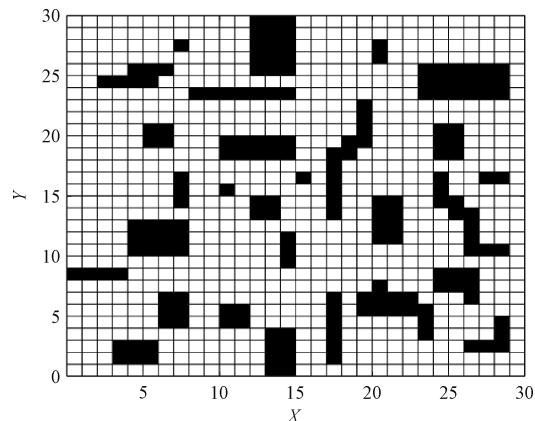


图 9 30×30 棚格地图

Fig. 9 30×30 grid map

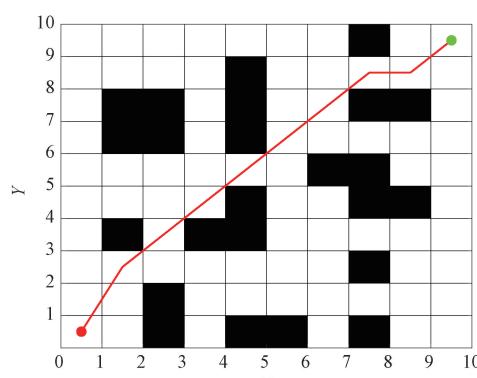
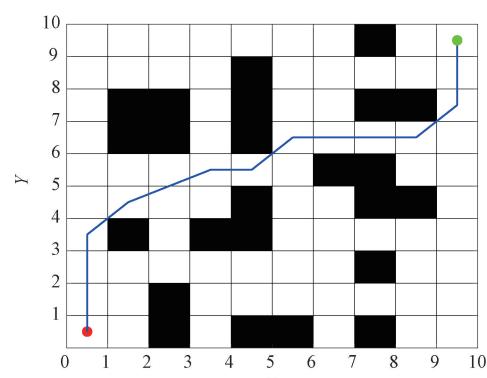
表 5 棚格地图运行结果

Table 5 Grid map run results

地图大小	算法	最优值	标准差	平均值
10×10	ISSA	13.135 6	0	13.135 6
	SSA	13.135 6	0.715 3	14.825
20×20	ISSA	28.834 5	0.319 8	29.110 9
	SSA	30.225 3	1.455 9	33.474 8
30×30	ISSA	44.364 9	0.668 7	45.916 2
	SSA	48.953 6	3.937 8	56.433 4

ISSA 各项指标都优于 SSA,且都有较为明显的提升。

ISSA 和 SSA 在 3 个棚格地图环境下路径曲线如图 10 所示。在 10×10 大小棚格地图环境中,对比图 10 (a) 和 (b) 可以看出,SSA 的曲线较为曲折,而 ISSA 的曲线较为平滑。在 20×20 大小棚格地图环境中,对比图 10 (c) 和 (d) 同样可以看出,ISSA 的曲线相较于 SSA 的曲线较为平滑,弯折较少。在 30×30 大小棚格地图环境中,对比图 10(e) 和 (f) 可以看出,ISSA 和 SSA 的路径有较大差异,这表明 SSA 可能陷入了局部最优解,并且跳出局部最优解的能力不佳。

(a) 10×10 棚格地图ISSA路径  
(a) 10 × 10 grid map ISSA path(b) 10×10 棚格地图SSA路径  
(b) 10 × 10 grid map SSA path

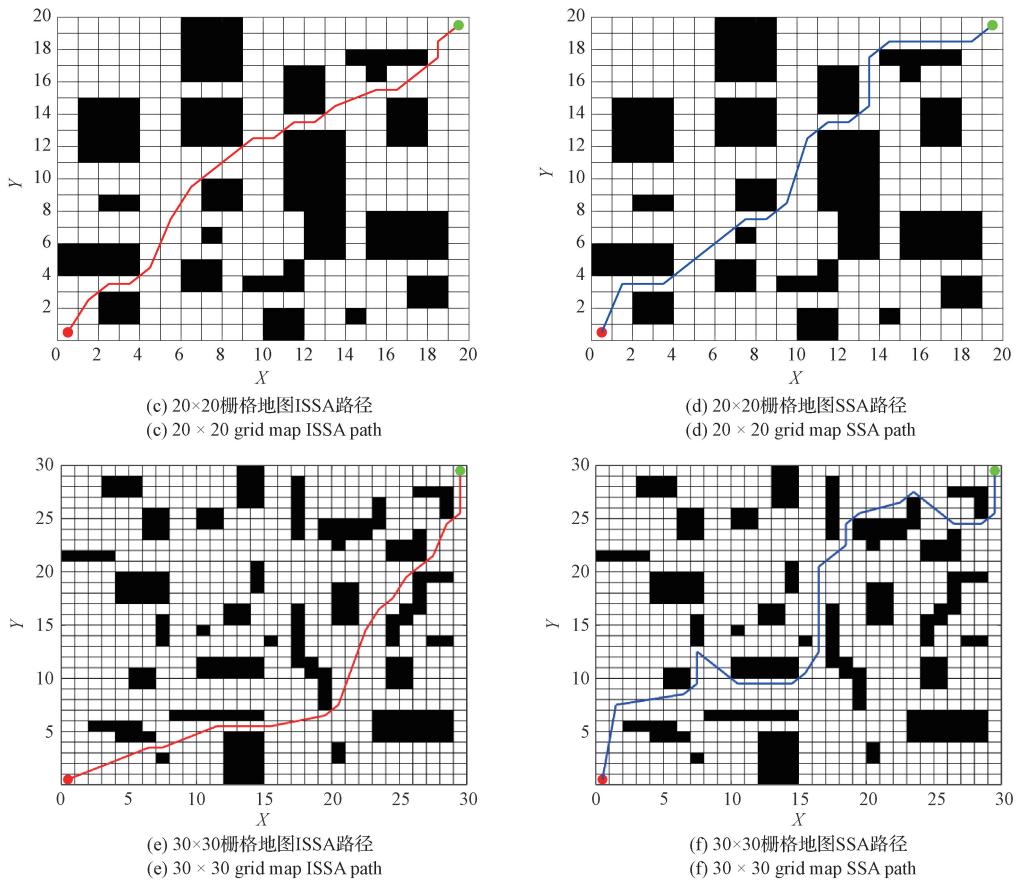


图 10 路径规划图

Fig. 10 Path planning diagram

针对如上分析可以看出,ISSA 的寻优能力,跳出局部最优解的能力和稳定性较 SSA 有较大提升,在简单和复杂环境下的性能都较为优越。

## 5 结 论

本文针对麻雀搜索算法依赖初始种群分布,易于陷入局部最优解,以及迭代后期种群多样性减少等问题,提出了一种多策略融合的改进麻雀搜索算法。采用 Sobol 序列初始化种群,引入随机反向学习策略改进发现者位置更新公式,引入螺旋觅食策略改进加入者位置更新公式,引入柯西变异对可能陷入局部最优解的麻雀进行扰动。为验证改进后的算法具有较为优异的性能,实验选取 9 个标准测试函数进行性能测试,实验结果表明,改进后的算法性能较原算法有较大提升,且和其他算法相比也展现出了更为优异的性能。最后,将改进后的算法应用于 AGV 路径规划,实验结果表明,ISSA 相较于 SSA,其寻优能力,跳出局部最优解的能力和稳定性更为优异。下一步的研究可以将 ISSA 应用到其他领域,如故障诊断,图像分割等。

## 参 考 文 献

[1] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf

optimizer [J]. Advances in Engineering Software, 2014, 69(3): 46-61.

- [2] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [3] HASHIM F, HUSSIEN A. Snake optimizer: A novel meta-heuristic optimization algorithm [J]. Knowledge-Based Systems, 2022, 242: 108-320.
- [4] XUE J, SHEN B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization [J]. The Journal of Supercomputing, 2022, 79(7): 7305-7336.
- [5] 万怡华, 张雪梅. 混合多策略改进蜣螂算法的避障路径规划 [J]. 电子测量技术, 2024, 47(2): 69-78.
- [6] WAN Y H, ZHANG X M. Obstacle avoidance path planning for hybrid multi strategy improved dung beetle algorithm [J]. Electronic Measurement Technology, 2024, 47(2): 69-78.
- [7] 肖金壮, 余雪乐, 周刚, 等. 一种面向室内 AGV 路径规划的改进蚁群算 [J]. 仪器仪表学报, 2022, 43(3): 277-285.
- [8] XIAO J ZH, YU X L, ZHOU G, et al. An improved

ant colony algorithm for indoor AGV path planning[J]. Chinese Journal of Scientific Instrument, 2022, 43(3): 277-285.

[7] 肖晓晖. 基于改进果蝇优化算法优化 SVM 的模拟电路故障诊断[J]. 电子测量与仪器学报, 2019, 33(5): 57-64.

XIAO X H. Simulation circuit fault diagnosis based on improved drosophila optimization algorithm and SVM optimization[J]. Journal of Electronic Measurement and Instrumentation, 2019, 33(5): 57-64.

[8] 李鹏, 陈守静, 杨山山, 等. 基于 Logistic 映射的果蝇算法优化 Otsu 图像分割方法[J]. 国外电子测量技术, 2022, 41(7): 9-17.

LI P, CHEN SH J, YANG SH SH, et al. Optimization of Otsu image segmentation method based on Logistic mapping drosophila algorithm[J]. Foreign Electronic Measurement Technology, 2022, 41(7): 9-17.

[9] XUE J, SHEN B. A novel swarm intelligence optimization approach: Sparrow search algorithm [J]. Systems Science Control Engineering, 2020, 8(1): 22-34.

[10] 李江华, 王鹏晖, 李伟. 一种混合多策略改进的麻雀搜索算法[J]. 计算机工程与科学, 2024, 46(2): 303-315.

LI J H, WANG P H, LI W. A hybrid multi strategy improved sparrow search algorithm [J]. Computer Engineering and Science, 2024, 46(2): 303-315.

[11] 回立川, 李瑶, 李欢欢, 等. 多策略改进的麻雀搜索算法[J]. 辽宁工程技术大学学报(自然科学版), 2023, 42(6): 722-732.

HUI L CH, LI Y, LI H H, et al. Multi strategy improved sparrow search algorithm[J]. Journal of Liaoning University of Engineering and Technology (Natural Science Edition), 2023, 42(6): 722-732.

[12] YANG X X, LIU J, LIU Y, et al. A novel adaptive sparrow search algorithm based on chaotic mapping and t-distribution mutation [J]. Applied Sciences, 2021, 11(23): 11192.

[13] 苏莹莹, 王升旭. 自适应混合策略麻雀搜索算法[J]. 计算机工程与应用, 2023, 59(9): 75-85.

SU Y Y, WANG X SH. Adaptive hybrid strategy sparrow search algorithm[J]. Computer Engineering and Applications, 2023, 59(9): 75-85.

[14] THARWAT A, SCHENCK W. Population initialization techniques for evolutionary algorithms for single-objective constrained optimization problems: deterministic vs. stochastic techniques[J]. Swarm and Evolutionary Computation, 2021, 67(12): 100952-100971.

[15] LONG W, JIAO J, LIANG X, et al. A random opposition-based learning grey wolf optimizer[J]. IEEE Access, 2019, 7: 113810-113825.

[16] XIE L, HAN T, ZHOU H, et al. Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization[J]. Computational Intelligence and Neuroscience, 2021, 2021: 1-22.

[17] 卢梦蝶, 鲁海燕, 侯新宇, 等. 融合柯西变异的鸟群与算术混合优化算法[J]. 计算机工程与应用, 2023, 59(14): 62-75.

LU M D, LU H Y, HOU X Y, et al. A hybrid optimization algorithm of bird swarm and arithmetic based on cauchy mutation[J]. Computer Engineering and Applications, 2023, 59(14): 62-75.

[18] 胡文建, 杨阳, 刘保安, 等. 基于粒子群算法的 5G 大规模 MIMO 通信系统效率评估[J]. 国外电子测量技术, 2022, 41(2): 46-52.

HU W J, YANG Y, LIU B AN, et al. Efficiency evaluation of 5G large-scale MIMO communication systems based on particle swarm optimization algorithm [J]. Foreign Electronic Measurement Technology, 2022, 41(2): 46-52.

## 作者简介

乐明皓, 本科, 主要研究方向为智能优化算法、智能控制方法。

E-mail: leminghao1234@163.com

李凌(通信作者), 博士, 副教授, 硕士生导师, 主要研究方向为复杂系统建模、优化及智能控制方法。

E-mail: liling@syuct.edu.cn