

路径规划问题的多策略改进樽海鞘群算法研究

赵宏伟, 董昌林, 丁兵如, 柴海龙, 潘志伟

引用本文

赵宏伟, 董昌林, 丁兵如, 柴海龙, 潘志伟. 路径规划问题的多策略改进樽海鞘群算法研究[J]. 计算机科学 2024, 51(6A): 230600083-9.

ZHAO Hongwei, DONG Changlin, DING Bingru, CHAI Hailong, PAN Zhiwei. [Study on Multi-strategy Improved Salp Swarm Algorithm for Path Planning Problem](#) [J]. Computer Science, 2024, 51(6A): 230600083-9.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于改进自适应蚁群算法的移动机器人路径规划](#)

Path Planning for Mobile Robots Based on Modified Adaptive Ant Colony Optimization Algorithm
计算机科学, 2024, 51(6A): 230500145-9. <https://doi.org/10.11896/jsjcx.230500145>

[航母航空保障作业中异质群体的动态路径规划算法](#)

Dynamic Path Planning Algorithm for Heterogeneous Groups in Aircraft Carrier Aviation Support Operations
计算机科学, 2024, 51(3): 226-234. <https://doi.org/10.11896/jsjcx.221200119>

[面向绿色节能的智能网联电动车调度方法](#)

Intelligent Networked Electric Vehicles Scheduling Method for Green Energy Saving
计算机科学, 2023, 50(12): 285-293. <https://doi.org/10.11896/jsjcx.230100099>

[基于车流密度的车载边缘计算任务动态卸载策略](#)

Dynamic Unloading Strategy of Vehicle Edge Computing Tasks Based on Traffic Density
计算机科学, 2023, 50(11A): 220900199-7. <https://doi.org/10.11896/jsjcx.220900199>

[基于多元约束Petri网的水利测绘无人机路径规划](#)

Path Planning of Hydrographic Mapping UAV Based on Multi-constraint Petri Net
计算机科学, 2023, 50(6A): 220700079-7. <https://doi.org/10.11896/jsjcx.220700079>

路径规划问题的多策略改进樽海鞘群算法研究

赵宏伟 董昌林 丁兵如 柴海龙 潘志伟

沈阳大学信息工程学院 沈阳 110044

(Zhw30@163.com)

摘要 针对移动机器人寻找最优路径问题,提出了一种融合无标度网络、自适应权重和黄金正弦算法变异策略的樽海鞘群算法 BAGSSA(Adaptive Salp Swarm Algorithm with Scale-free of BA Network and Golden Sine)。首先,生成一个无标度网络来映射跟随者的关系,增强算法全局寻优的能力,在追随者进化过程中集成自适应权重 ω ,以实现算法探索和开发的平衡;同时选用黄金正弦算法变异进一步提高解的精度。其次,对12个基准函数进行仿真求解,实验数据表明平均值、标准差、Wilcoxon检验和收敛曲线均优于基本樽海鞘群和其他群体智能算法,证明了所提算法具有较高的寻优精度和收敛速度。最后,将BAGSSA应用于移动机器人路径规划问题中,并在两种测试环境中进行仿真实验,仿真结果表明,改进樽海鞘群算法较其他算法所寻路径更优,并具有一定理论与实际应用价值。

关键词 樽海鞘群算法;无标度网络;自适应权重;黄金正弦算法;路径规划

中图分类号 TP181

Study on Multi-strategy Improved Salp Swarm Algorithm for Path Planning Problem

ZHAO Hongwei, DONG Changlin, DING Bingru, CHAI Hailong and PAN Zhiwei

School of Information Engineering, Shenyang University, Shenyang 110044, China

Abstract Aiming at the problem of finding the optimal path for mobile robots, a salp swarm algorithm BAGSSA (adaptive salp swarm algorithm with scale-free of BA network and golden sine algorithm) combining scale-free network, adaptive inertia weight and golden sine algorithm mutation strategy is proposed. First, a scale-free topology network is generated to map the relationship of followers, so as to enhance the global optimization ability of the algorithm; and the adaptive inertia weight is introduced in the followers to form a spontaneous adjustment to the overall distribution of the population and enhance the ability of local optimization. The variation of the golden sine algorithm is selected to further improve the accuracy of the solution. Secondly, through the simulation solution of 12 benchmark functions, experimental data show that the average value, standard deviation, Wilcoxon test and convergence curve are better than that of the standard SSA and other swarm intelligence algorithms. The proposed algorithm has higher optimization accuracy and convergence speed. Finally, BAGSSA is applied to the path planning problem of mobile robots, and simulation experiments are carried out in two test environments. Simulation results show that the improved salp swarm algorithm is better than other algorithms in finding the path, and has certain theoretical and practical application value.

Keywords Salp swarm algorithm, Scale-free network, Self-adaption weighty, Golden sine algorithm, Route planning

1 引言

近年来,移动机器人技术得到了国内外的普遍关注,被广泛应用于军事、农业、服务业、医疗等领域。路径规划是移动机器人研究中最关键、最核心的技术。路径规划问题旨在通过特定的优化策略和限制条件,在移动空间内找到一条连接起始和终点目标的最优或接近最优的路径。在这一背景下,如何设计一条平滑而且能够避开障碍物的最佳路径,是目前仍具有挑战性的议题。随着移动机器人中路径规划问题的深入研究和应用以及所需规划场景的不断变化,使得传统的A*算法^[1]、人工势场^[2]、神经网络^[3]、强化学习^[4]等多种不同技术特点的算法在规划时间和路径结果

上常常难以获得满意的实验结果。

为了解决上述问题,学者们通过充分的研究和实验,发现元启发式智能优化算法在求解机器人路径规划方面具有独特优势,多种智能优化算法已在该领域得到成功应用,例如蚁群算法^[5]、遗传算法^[6]、粒子群算法^[7]等。但是这些算法在该领域中也存在一些问题,如遗传算法对参数敏感、对环境适应性差、搜索空间效率低、易陷入局部最优等;粒子群算法会对初始参数敏感以及遇到约束问题时需要额外的处理策略;蚁群算法不适应动态环境以及随机性带来的算法不稳定等现象。因此,在此领域的研究者们不断追求的目标和方向是寻找一种能够更高效解决问题、更好适应不同环境、具备更强寻优能力和鲁棒性的元启发式智能优化算法,以不断推动路径规划

基金项目:国家自然科学基金(71672117);中国博士后科学基金(2019M651142)

This work was supported by the National Natural Science Foundation of China (71672117) and China Postdoctoral Science Foundation (2019M651142).

通信作者:董昌林(dong_clin@163.com)

方案的发展和进步。

由 Mirjalili 等提出了一种新型群智能算法,即樽海鞘群算法(Salp Swarm Algorithm, SSA)^[8],该算法结构简单、参数少、易于实现,并且具有较高认可度和影响力,已成为智能计算领域广受关注和研究的重要算法,并在诸多领域和问题的优化中得到成功应用,为解决路径规划问题提供了新的思路。考虑到 SSA 算法与其他群体智能算法的独特差异和潜在优势,相比传统方法,樽海鞘群算法突出了领导者与追随者的动态关系,追随者根据前一个个体位置更新自身,形成链式结构,这种机制维持了群体多样性,减少了陷入局部最优解的风险。此外,该算法的简洁性和对环境变化的适应性为解决实际问题提供了有力支持。在机器人规划等领域,樽海鞘群算法可能会对路径规划的平衡探索与利用提供新途径,有望在复杂任务中表现出色。因此,本文选择樽海鞘群算法,旨在深入探讨其在问题优化中的潜在价值和应用前景;最后总结全文。

2 相关工作

标准的樽海鞘群算法,在求解机器人路径规划问题时具有较强的优化能力,但是在求解一些高维复杂问题时依旧存在收敛速度慢、精度低、易陷入局部最优的缺点;诸多学者就针对以上缺点提出了该算法的各种改进策略。Sayed 等^[9]提出将混沌理论融入 SSA 算法中,即通过 10 种混沌映射来调节 SSA 算法的控制参数 c_2 ,使作为领导者的樽海鞘个体根据混沌映射更新最优位置,从而提高算法的结果精度和收敛速度。Yang 等^[10]提出采用多个独立的樽海鞘链扩展 SSA 算法,把多个樽海鞘群重组为一个新的种群,用于不同樽海鞘链之间的全局协作,从而增强算法的稳定性。Yu 等^[11]设置了两类领导者和两种跟随群体,其中执行正态过程搜索的领导者需要进行正态过程游走、交叉、选择等操作,兼顾了全局搜索和局部开发两种功能。Hegazy 等^[12]提出了能使收敛速度加快的固定惯性权重。Zhang 等^[13]利用混沌初始化和差分进化策略对 SSA 进行了优化,提升了 SSA 的性能。Nasri 等^[14]提出了一种基于 Levy 飞行轨迹的 SSA,提升了 SSA 的全局寻优能力,使得算法更可靠。Zhang 等^[15]提出了将 PSO 算法的随机惯性权重引入 SSA 算法的追随者位置更新公式中,增强和平衡 SSA 算法的勘探与开发能力,用 DE 算法的变异操作替代 SSA 算法的领导者位置更新操作,提高 SSA 算法的收敛速度和计算精度。Ren 等^[16]采用随机替换和双重自适应策略,提高了算法的收敛速度和算法的开发力。Ding 等^[17]将反向学习和黄金正弦算法融合,以提高算法的收敛速度。Zhang 等^[18]用莱维飞行策略的长短跳跃特点对领导者位置进行随机更新,以增强全局最优的搜索能力,其次增加对追随者位置的更新条件,让追随者不再盲目地跟随,从而加快收敛速度。

上述樽海鞘群算法的改进在追随者的更新中只与之前的个体相关,这种局部视角可能不利于其加速进化。此外,在领导者更新中虽然考虑了进化时间对优化性能的影响,但没有将樽海鞘群的空间分布作为影响因素。为了更好地求解机器人全局路径规划问题,提高樽海鞘群算法的寻优性能,扩展其应用范围,本文的主要改进是:

1)使用无标度网络来映射樽海鞘群的跟随者种群,尽可

能地增强樽海鞘种群的全局勘探能力,并达到算法对开发和探索的平衡。

2)在追随者的进化过程中集成自适应权重,自适应调节种群整体的分布,增强算法收敛速度和全局搜索能力。

3)在改进机制中采用黄金正弦算子变异策略,选择最差个体进行变异,可提高最差个体解的质量。

3 樽海鞘群算法

在樽海鞘群算法中,樽海鞘个体位置矢量 X 被用于在 k 维的空间中进行搜索,且 k 相当于决策变量的数量。樽海鞘种群初始化时是由 N 个维度为 k 的个体所构成的,因此以下的矩阵构成了樽海鞘群的种群。

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,k} \end{bmatrix} \quad (1)$$

在 SSA 算法中,追踪食物源的位置是樽海鞘种群中每一个个体的目标。基于此,领导者的更新公式如下:

$$x_j^1 = \begin{cases} F_j + c_1(c_2(ub_j - lb_j) + lb_j), & c_3 \geq 0.5 \\ F_j - c_1(c_2(ub_j - lb_j) + lb_j), & c_3 < 0.5 \end{cases} \quad (2)$$

其中, x_j^1 是领导者的在第 j 维上第一只樽海鞘的位置; ub_j 和 lb_j 分别是上下界; F_j 是食物源在第 j 维上的位置; c_2 和 c_3 都是在 $[0, 1]$ 区间内随机生成的两个数; c_1 是负责平衡探索和开发的收敛因子,计算式如下:

$$c_1 = 2e^{-\left(\frac{t}{T}\right)^2} \quad (3)$$

其中, t 表示目前的迭代次数; T 表示最大的迭代次数,迭代初期 c_1 的值越大,越有利于算法进行全局搜索,而在迭代后期 c_1 的值越小,越有利于算法的局部搜索。

根据牛顿运动定律,追随者的位置更新公式如下:

$$x_j^i = \frac{1}{2}at^2 + v_0t \quad (4)$$

其中, x_j^i 为第 i 个追随者在第 j 维上的位置, i 取值大于 2; t 为时间, v_0 为初速度; a 为加速度。由于每次追随者的位置更新只和它前一个位置的樽海鞘有关,因此速度为:

$$v_{\text{final}} = (x_j^{i-1} - x_j^i) / t \quad (5)$$

追随者的位置更新公式进而可以表示为:

$$x_j^i = \frac{1}{2}(x_j^{i-1} + x_j^i) \quad (6)$$

由于在实际的迭代过程中并不清楚食物源的位置,因此在迭代过程中会计算所有樽海鞘个体的适应度值,并将最优适应度值的樽海鞘位置作为当前的食物源位置。

4 改进樽海鞘群算法 BAGSSA

尽管基本 SSA 的结构很容易实现,但在探索和开发中仍有改进的空间,尤其是对樽海鞘群中的追随者。为此,本章拟在追随者位置更新规则中做出一些改进措施。

4.1 无标度网络策略

从追随者位置更新公式便可以看出,追随者的樽海鞘个体的位置是与前一个樽海鞘追随者个体有关且依赖于前一个体,因此基本的算法是具有盲目性的,而且考虑到樽海鞘群的空间分布因素,故加入无标度网络增强跟随者之间的联系。

复杂网络在图论中可以用边和节点表示:设图 $G = \langle V,$

E),其中 V 是整个图的顶点集, E 便是整个图的边集,与节点相关联的边数就是该节点的度。Barabasi 等^[19]于 1999 年通过分析大量的数据提出了无标度网络模型。该网络的度分布满足幂律分布,这种网络结构已经在现实的世界中得到证明,如互联网、大脑神经网络和生物网络。产生无标度网络的经典模型便是 BA 模型,其构建步骤共有 3 步。

1)增长的全连通网络:定义一个规模为 m_0 个节点的网络开始,每一次加入一个新的节点,并且与已有的节点连接,类似于现实中的网络不断增长。

2)优先权规则:新的节点连接到已有的 m 个节点时优先与节点度高的节点连接。对于新加入的节点与已存在的第 i 个节点相连接的概率为:

$$p(k_i)=\frac{k_i}{\sum_n k_n} \quad (7)$$

其中, n 为当前网络节点的个数, k_i 为第 i 个节点的度。

3)重复以上步骤,直到达到目标点数和边数的要求之后网络便构建完成。

本文首先构建出一个无标度网络结构来映射跟随者的关系,接下来通过 BA 模型生成与跟随者数量相同的网络,在 t 时间步长内,将第 i 个跟随者的适应度值 $fitness_i$ 与所构成网络中的节点的度 k 相关联,表示为 $fitness_i \propto k$ 。其中跟随者可在网络中随机选择邻居 ρ 进行位置更新,更新公式如下:

$$x_j^i = \frac{1}{2}(x_j^i + x_j^\rho), \rho \in \text{Neighbor}(i) \quad (8)$$

其中,Neighbor(i)表示第 i 个跟随者所有邻居的集合。最终在构建的追随者无标度网络中,每一个跟随者至少有 m 个邻居,因此邻居多的追随者会有更多不确定的更新机会,最终导致樽海鞘群的探索和开发更加平衡和多样化。

4.2 自适应权重策略

为了对整个樽海鞘群进行动态调整,考虑集成自适应权重策略。该权重的 ω 如下:

$$\omega = \left(1 - \frac{t}{T_{\max}}\right) e^{-c} \quad (9)$$

$$c = \sum_{i=1}^N \sqrt{\sum_{j=1}^{\dim} (x_j^i - \bar{x}^j)^2} \times \frac{1}{N \times D} \quad (10)$$

其中,两个 x 分别表示樽海鞘群中第 j 维的第 i 个樽海鞘个体的适应度值和整个樽海鞘群的平均适应度值。其中搜索空间的最长对角线的距离为 $D = \sqrt{\sum_{j=1}^{\dim} (ub_j - lb_j)^2}$,其中 $\dim$ 指搜索空间的最大维数。考虑到优化整个樽海鞘群算法的性能,将此处的自适应权重策略与无标度网络策略结合得出一个最终的追随者位置更新公式:

$$x_j^i = \frac{1}{2}(\omega \times x_j^i + r_1 \times x_j^\rho + r_2 \times F_j), \rho \in \text{Neighbor}(i) \quad (11)$$

其中, r_1 和 r_2 是 0,1 闭区间的随机数。从式(11)可以看出,追随者位置更新的方向是由 3 个因素所影响的,分别是食物源、邻居和自身,其中食物源和邻居对追随者的位置更新影响更大。通过无标度网络和自适应权重的改进之后,在 T 和 $T+1$ 个时间步长中,每一个樽海鞘个体的位置都不是固定的。由于改进机制中解的值与网络节点的度成正比,因此樽海鞘的追随者个体选择更好的解作为邻居的概率就会越大。于是个体的搜索便更加向着最优解靠近,与基本的 SSA 算法中只考虑相邻个体的追随者规则相比,改进后的机制更加有利于追随者的开发和探索。

4.3 黄金正弦算子变异策略

黄金正弦算法是由 Tanyildiz 等^[20]于 2017 年最新提出的新型智能算法,由于受到正弦函数和单位圆的启发,黄金正弦算法对整个单位圆的搜索便类似于整个搜索空间内的寻优过程,同时取黄金分割数以便搜索可以产生较好结果的区域并且缩小搜索的空间,加快了算法的收敛速度。黄金正弦算子的具体计算式如下:

$$X_i(t+1) = X_i(t) | \sin R_1 | + R_2 \sin(R_1) | a X_i - b X_i(t) | \quad (12)$$

其中, R_1 是 0 至 2π 的随机数, r_2 是 0 至 π 之间的随机数, a 和 b 是采用黄金分割法之后产生的系数,计算式如下:

$$\begin{aligned} a &= -\pi(1-\tau) + \pi\tau \\ b &= \pi(1-\tau) - \pi\tau \end{aligned} \quad (13)$$

其中, τ 是黄金分割数。本文充分利用黄金正弦算子的优势,在循环多次仍未收敛的樽海鞘追随者中选取一定比例的最差个体进行变异,打开更优的搜索空间,向着更优解靠拢,具体计算式如下:

$$\begin{aligned} X_j^i(t+1) &= X_j^i(t) | \sin R_1 | + R_2 \sin(R_1) | a F_j - b X_i(t) | \\ X_j^i(t) &\in fitness_{\text{worst}} \end{aligned} \quad (14)$$

在进行黄金正弦变异之后,选取变异后个体适应度值小于原种群所对应的适应度值的个体进行位置更新。

BAGSSA 算法的伪代码如算法 1 所示。

算法 1 BAGSSA 算法

输入:种群大小 N,最大迭代次数 $T_{\max}$,无标度网络参数 m, m_0 以及上下界 ub, lb

输出:Return F

1. 初始化领导者和跟随者的位置,通过 BA 模型构建一个 $N/2$ 节点的无标度网络
2. While $t < T_{\max}$ do
3. 计算每一个樽海鞘个体的适应度值并排序
4. 关联跟随者 M 之间的拓扑关系
5. 记录最优个体位置 FoodPosition
6. 根据式(9)计算出自适应权重
7. for each $i \leq N$ do
8. if $i \leq \frac{N}{2}$
9. 根据式(2)更新领导者位置
10. else
11. 根据式(11)更新追随者位置
12. End if
13. End for
14. 处理上下边界值
15. if F 超过 c 轮的循环还没有明显改善 do
16. 选择占群体一定比例的最差个体,根据式(14)进行黄金正弦算子变异
17. end if
18. end while

5 仿真实验和算法分析

为了验证本文提出的 BAGSSA 算法的性能的优越性和鲁棒性,本文将进行两组实验,首先对本文的改进策略进行分析,再对其他的改进 SSA 算法进行对比分析。

5.1 测试函数

表 1 列出了 12 个基准测试函数,维度为 10~200 维,最优值都是 0,为了提高每次仿真实验的公平性,两组实验都采

用表 1 所列的测试函数。其中 F1—F4 是单峰函数,主要用来检测算法的局部搜索能力和收敛速度,F5—F12 为多峰函数,主要用来检测算法的跳出局部最优能力和全局优化能力。

本文实验中所有算法的最大迭代次数为 1 000 次,种群个数为 30,为了保持算法结果的公平性,本次实验中的测试均独立运行 50 次。

表 1 基准函数
Table 1 Basis functions

函数	函数表达式	最优值	特征
F_1	$f_1(x)=\sum_{i=1}^D x_i ^{i+1}$	0	单峰
F_2	$f_2(x)=\sum_{i=1}^D(10^6)^{\frac{i-1}{D-1}}x_i^2$	0	单峰
F_3	$f_3(x)=x_1^2+10^6\sum_{i=2}^Dx_i^2$	0	单峰
F_4	$f_4(x)=\sum_{i=1}^Dx_i^2+(\sum_{i=1}^D0.5x_i^2)^2+(\sum_{i=1}^D0.5x_i^2)^4$	0	单峰
F_5	$f_5(x)=10^6x_1^2+\sum_{i=2}^Dx_i^2$	0	多峰
F_6	$f_6(x)=\sum_{i=1}^D\frac{x_i^2}{4000}-\prod_{i=1}^D\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	0	多峰
F_7	$f_7(x)=\sum_{i=1}^{D-1}(100(x_i^2-x_{i+1})^2+(x_i-1)^2)$	0	多峰
F_8	$f_8(x)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^Dx_i^2})-\exp\left(\frac{1}{D}\sum_{i=1}^D\cos(2\pi x_i)\right)+20+e$	0	多峰
F_9	$f_9(x)=\left[\frac{1}{D-1}\sum_{i=1}^{D-1}(\sqrt{s_i}(\sin(50.0s_i^{0.2})+1))\right]^2,s_i=\sqrt{x_i^2+x_{i+1}^2}$	0	多峰
F_{10}	$f_{10}(x)=\sum_{i=1}^D(x_i^2-10\cos(2\pi x_i))+10$	0	多峰
F_{11}	$f_{11}(x)=g(x_1,x_2)+g(x_2,x_3)+\cdots+g(x_D,x_1)g(x,y)=0.5+\frac{\sin^2(\sqrt{x^2+y^2})}{(1+0.001(x^2+y^2))^2}$	0	多峰
F_{12}	$f_{12}(x)=\sum_{i=1}^D[y_i^2-10\cos(2\pi y_i)+10]y_i=\begin{cases}x_i,& x_i <\frac{1}{2}\\\frac{round(2x_i)}{2},& x_i \geq\frac{1}{2}\end{cases}$	0	多峰

本文实验环境如下: Intel(R) Core(TM) i7-6700HQ, 8.00 GB 内存, 算法基于 Matlab R2021a 用 M 语言编写。生产无标度网络的参数设置为 $m=m_0=4$ 。算法性能采用平均值 (mean) 和标准差 (std) 来反映算法的收敛精度和稳定性, 为了判断本文算法是否具有显著的差异性, 引入 Wilcoxon 秩和检验进行差异性检验, 所获得的实验结果在 5% 的显着性水平下进行统计检验。零假设指对比算法之间没有显著的差异, 如果 p 大于 0.05, 则说明两种算法的性能相差不大, 如果 p 小于 0.05, 则认为是拒绝零假设^[21]的有力验证, 两种对比算法的性能有着显著差异。Mean 值越小, 算法更优, 当 Mean 值相同时, 就对算法的标准差进行比较, 标准差越小, 算法的性

能越优。进行 Wilcoxon 秩和检验时无法与自己进行比较, 因此将值标注为 NaN。

5.2 不同策略的对比实验结果和分析

为验证本文的改进策略对算法带来的性能提升, 将融合单一策略之后的改进算法与标准算法 SSA 和 BAGSSA 算法进行数据对比。首先设定融合无标度网络策略的改进 SSA 算法为 BSSA; 融合自适应权重的改进 SSA 算法为 ASSA; 融合黄金正弦算子变异策略的改进算法为 GSSA。表 2 列出了各个改进策略与标准 SSA 算法分别对 12 个测试函数运行 50 次之后所得结果的均值、标准差以及秩和检验的结果。

表 2 单一策略仿真实验结果
Table 2 Simulation results of single strategy

函数	结果	BAGSSA	SSA	BSSA	ASSA	GSSA
F_1	Mean	0	1.940×10^4	4.054×10^{-112}	4.207×10^{-15}	1.110×10^{-27}
	Std	0	3.690×10^4	4.678×10^{-112}	5.065×10^{-15}	8.600×10^{-28}
	p-value	NaN	1.210×10^{-12}	1.210×10^{-12}	1.210×10^{-12}	1.210×10^{-12}
F_2	Mean	0	1.750×10^{-10}	8.645×10^{-106}	5.345×10^{-15}	3.070×10^{-35}
	Std	0	3.940×10^{-10}	3.704×10^{-105}	1.084×10^{-17}	5.930×10^{-35}
	p-value	NaN	1.210×10^{-12}	1.210×10^{-12}	1.210×10^{-12}	1.210×10^{-12}
F_3	Mean	2.990×10^{-114}	1.310×10^9	2.549×10^{-102}	4.425×10^{-14}	1.580×10^{-13}
	Std	7.540×10^{-115}	4.280×10^8	5.674×10^{-103}	2.508×10^{-16}	4.650×10^{-14}
	p-value	NaN	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}
F_4	Mean	4.012×10^{-115}	7.520×10^2	6.909×10^{-108}	1.000×10^{-14}	1.020×10^{-8}
	Std	4.345×10^{-115}	3.520×10^2	4.375×10^{-109}	3.540×10^{-16}	1.730×10^{-9}
	p-value	NaN	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}
F_5	Mean	9.671×10^{-139}	7.020×10^3	9.008×10^{-110}	5.212×10^{-14}	1.910×10^{-28}
	Std	4.009×10^{-138}	2.070×10^3	7.031×10^{-105}	6.893×10^{-16}	1.130×10^{-28}
	p-value	NaN	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}
F_6	Mean	0	1.780	0	5.211×10^{-1}	1.700×10^2
	Std	0	5.260×10^{-1}	0	3.785×10^{-1}	2.109×10^1
	p-value	NaN	1.210×10^{-12}	NaN	1.210×10^{-12}	1.210×10^{-12}

(续表)						
函数	结果	BAGSSA	SSA	BSSA	ASSA	GSSA
F_7	Mean	4.354	1.610×10^4	5.645	3.956	0
	Std	4.077×10^{-1}	7.530×10^4	1.312×10^{-1}	1.341×10^{-2}	0
	p-value	1.210×10^{-12}	5.070×10^{-11}	3.020×10^{-11}	3.020×10^{-11}	NaN
F_8	Mean	7.770×10^{-16}	5.450	3.212	3.244×10^{-1}	7.640×10^{-15}
	Std	0	6.660	0	6.414×10^{-13}	3.440×10^{-15}
	p-value	NaN	1.210×10^{-12}	NaN	1.210×10^{-12}	1.210×10^{-12}
F_9	Mean	4.106×10^{-54}	1.300	3.312×10^{-48}	5.454×10^{-4}	0
	Std	4.037×10^{-26}	5.820×10^{-1}	8.231×10^{-48}	3.544×10^{-4}	0
	p-value	NaN	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	NaN
F_{10}	Mean	0	1.220×10^4	0	2.246×10^{-14}	5.020×10^{-11}
	Std	0	1.800×10^3	0	3.345×10^{-15}	1.062×10^{-11}
	p-value	NaN	1.210×10^{-12}	NaN	1.210×10^{-12}	1.210×10^{-12}
F_{11}	Mean	0	3.520×10	0	3.701×10	3.750×10^{-15}
	Std	0	1.700	0	6.940×10	1.210×10^{-15}
	p-value	NaN	1.210×10^{-12}	NaN	1.210×10^{-12}	1.210×10^{-12}
F_{12}	Mean	0	4.080×10^4	3.312×10^{-20}	6.045×10^{-15}	8.524×10^{-15}
	Std	0	3.110×10^3	6.456×10^{-14}	5.141×10^{-15}	6.458×10^{-13}
	p-value	NaN	1.210×10^{-12}	NaN	1.210×10^{-12}	1.210×10^{-12}

由表 2 中的 Mean 值可知,除了 F_8 测试函数 ASSA 的寻优精度略小于标准 SSA,其余的改进策略 50 次的寻优精度相比 SSA 都有不同程度的提升,其中无标度网络改进算法 BSSA 在寻优精度上的提升是最大的,其次为自适应权重策略改进算法 ASSA。以函数 F_5 为例,BAGSSA 算法相比其他两种改进策略的算法在寻优精度上提升了 29 和 125 个数量级。这说明融合各个改进策略之后的 BAGSSA 在收敛精度上是更优于单一的改进策略;分析表中的标准差可知,除了个别测试函数之外,BSSA,ASSA,GSSA 和 BAGSSA 算法测试 50 次所得的结果均小于 SSA,测试函数 F_6 可以看出 BSSA 和 ASSA 有效增强了算法跳出局部最优解的能力,由此也证明了无标度网络策略以及自适应权重策略增强了本文算法的稳定性,同时也证明了本文改进算法的合理性。

为了方便观察收敛情况,本文对收敛适应度值取以 10 为

底的对数,图 1 给出了部分基准测试时函数的收敛曲线。由图 1 可知,在函数 $F_1, F_2, F_3, F_5, F_{10}, F_{12}$ 的收敛曲线图上可以看出,BAGSSA 在不同的测试函数上具有连续收敛的能力,收敛速度明显快于其他方法,这得益于 BAGSSA 更好的开发能力。改进策略所达到的收敛精度都是优于标准 SSA 算法的,且由于 BSSA 和 ASSA 具有良好的跳出局部最优解的能力,使得收敛精度得到更进一步的提高,由此说明了自适应惯性权重改进策略、无标度网络策略以及黄金正弦算子变异策略均有效加快了收敛速度;算法整体的收敛速度和精度都得到了有效的提升。

可以发现,BSSA 或 ASSA 在一些测试函数中表现良好,但获得最佳值的 BAGSSA 的数量最多。与单一策略相比,无论是采用无标度拓扑结构的跟随者还是采用自适应权重策略,这种探索与开发相结合的 BAGSSA 算法在优化性能方面都具有更强的优势。

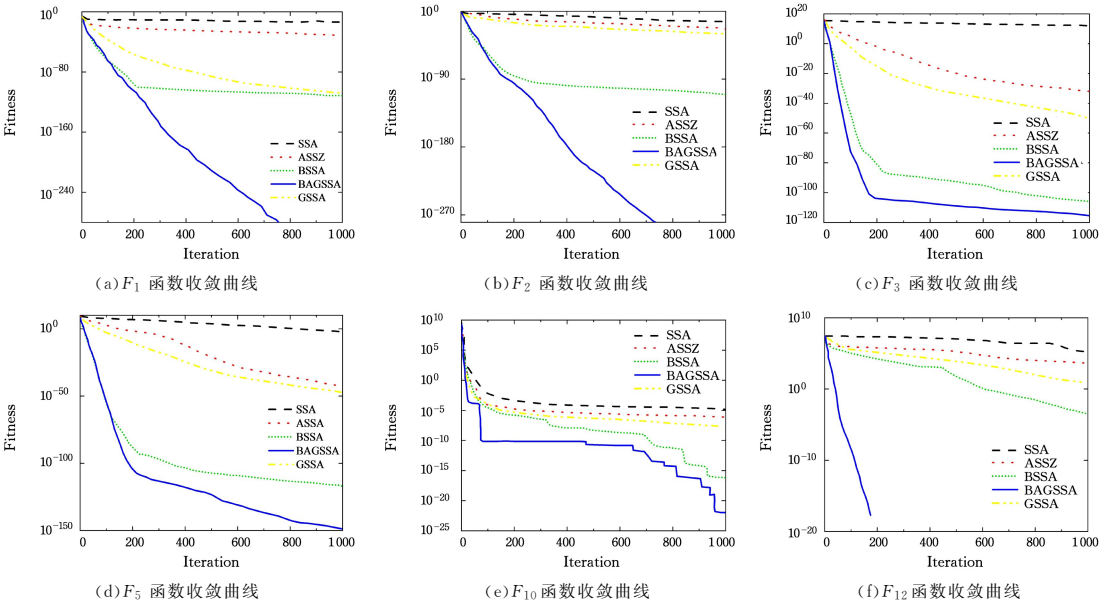


图 1 不同测试函数下单一策略的收敛曲线

Fig. 1 Convergence curves of single strategies with iffereent test functions

5.3 BAGSSA 与其他算法的对比实验

为验证本文改进算法的优越性,选取疯狂自适应樽海鞘群算法 CASSA^[22]、多子群的共生非均匀高斯变异樽海鞘群

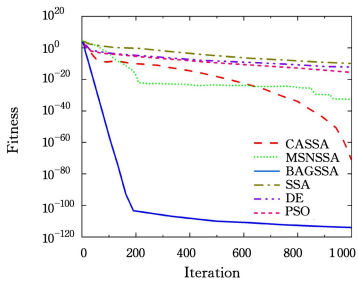
算法 MSNSSA^[23]、PSO、差分进化算法 DE^[24]、SSA 的性能进行比较。各种算法的参数设置均参考原文。表 3 列出了各个对比算法运行 50 次之后结果的均值和标准差以及 Wilcoxon

秩和检验。图 2 给出了各个测试函数被算法处理 50 次后 得到的收敛精度较高的一次收敛曲线。

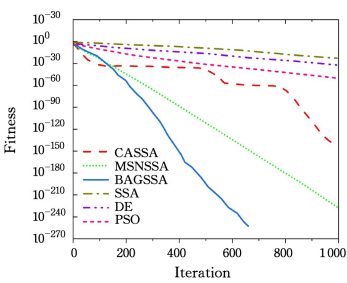
表 3 改进算法对比仿真实验结果

Table 3 Simulation experiment result comparison of improved algorithms

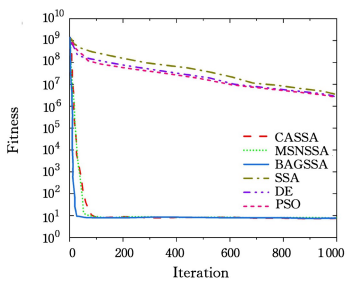
函数	结果	SSA	CASSA	PSO	MSNSSA	DE	BAGSSA
F_1	Mean	6.013×10^{-9}	1.620×10^{-70}	8.939	4.600×10^{-30}	2.629×10^4	0
	Std	1.317×10^{-9}	2.015×10^{-69}	2.881	2.790×10^{-29}	2.292×10^3	0
	p	1.320×10^{-11}	1.320×10^{-11}	1.320×10^{-11}	1.320×10^{-11}	1.967×10^{-16}	NaN
F_2	Mean	1.302×10^{-1}	2.650×10^{-64}	2.577	8.236×10^{-30}	2.910×10^3	0
	Std	2.732×10^{-1}	1.020×10^{-65}	2.569	5.032×10^{-29}	4.487×10^3	0
	p	1.320×10^{-11}	1.320×10^{-11}	1.320×10^{-11}	1.320×10^{-11}	1.320×10^{-11}	NaN
F_3	Mean	6.199×10^{-9}	7.056×10^{-66}	9.695	1.454×10^{-17}	2.741×10^4	3.365×10^{-114}
	Std	1.344×10^{-9}	2.170×10^{-66}	2.976	2.323×10^{-16}	2.410×10^3	4.367×10^{-115}
	p	7.545×10^{-1}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	NaN
F_4	Mean	1.830×10^{-2}	3.465×10^{-71}	3.280×10^{-2}	6.156×10^{-26}	1.975×10	9.364×10^{-115}
	Std	6.607×10^{-3}	3.565×10^{-78}	1.096×10^{-2}	8.156×10^{-24}	4.434	8.957×10^{-115}
	p	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	NaN
F_5	Mean	8.554×10^{-1}	1.563×10^{-64}	1.870	2.566×10^{-22}	2.112×10^6	4.636×10^{-139}
	Std	8.929×10^{-1}	8.415×10^{-60}	1.203	5.415×10^{-22}	7.586×10^5	9.364×10^{-138}
	p	6.627×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	4.010×10^{-11}	NaN
F_6	Mean	9.723×10^{-3}	0	1.084	0	2.375×10^2	0
	Std	1.136×10^{-2}	0	2.449×10^{-2}	0	2.071×10	0
	p	1.967×10^{-16}	NaN	1.967×10^{-16}	NaN	1.967×10^{-16}	NaN
F_7	Mean	2.448×10^{-3}	7.126	1.703	5.152	2.021×10^8	4.398
	Std	5.117×10^{-3}	3.145×10^{-1}	7.555×10^{-1}	3.207×10^{-1}	4.610×10^7	2.333×10^{-1}
	p	4.884×10^{-3}	2.021×10^{-11}	2.021×10^{-11}	NaN	2.021×10^{-11}	1.236×10^{-11}
F_8	Mean	1.129×10	8.268×10^{-16}	6.327	6.256×10^{-1}	5.186×10	8.777×10^{-16}
	Std	6.673	0	1.932	5.156	6.176	0
	p	1.320×10^{-11}	NaN	1.320×10^{-11}	1.320×10^{-11}	1.320×10^{-11}	NaN
F_9	Mean	2.000×10	5.326×10^{-54}	5.326×10^{-54}	6.229×10^{-7}	5.326×10^{-54}	5.326×10^{-54}
	Std	3.858×10^{-5}	3.365×10^{-35}	1.722×10^{-4}	6.325×10^{-8}	9.893×10^{-2}	7.964×10^{-26}
	p	4.010×10^{-11}	NaN	NaN	4.010×10^{-11}	NaN	NaN
F_{10}	Mean	5.912×10^5	0	2.708×10^6	0	1.006×10^6	0
	Std	6.938×10^5	0	3.264×10^6	0	3.403×10^5	0
	p	1.967×10^{-16}	NaN	1.967×10^{-16}	NaN	1.967×10^{-16}	NaN
F_{11}	Mean	3.138×10^2	6.325	9.158×10^2	3.701×10	5.625×10^2	0
	Std	3.168×10^2	6.364	4.152×10^2	7.369×10	1.087×10^2	0
	p	2.804×10^{-16}	4.236×10^{-4}	2.804×10^{-16}	2.804×10^{-16}	2.804×10^{-16}	NaN
F_{12}	Mean	1.988	0	3.502	0	1.033×10	0
	Std	9.593×10^{-1}	0	1.571	0	3.369×10^{-1}	0
	p	8.738×10^{-3}	NaN	1.138×10^{-13}	NaN	1.138×10^{-13}	NaN



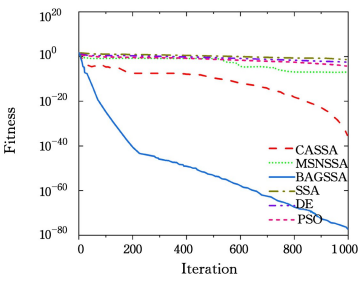
(a) F_3 函数收敛曲线



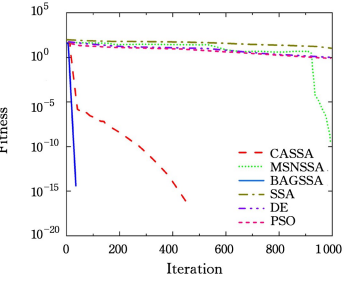
(b) F_4 函数收敛曲线



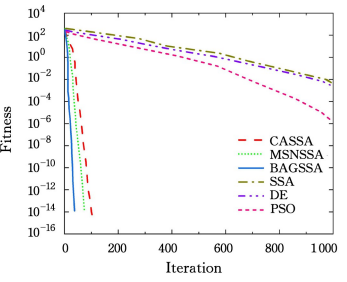
(c) F_7 函数收敛曲线



(d) F_9 函数收敛曲线



(e) F_{11} 函数收敛曲线



(f) F_{12} 函数收敛曲线

图 2 不同测试函数下对比算法的收敛曲线

Fig. 2 Convergence curve comparison with different test functions

由表 3 可得,当测试函数为 $F_1, F_2, F_3, F_4, F_5, F_9$ 时, BAGSSA 的实验结果中均值都小于其他对比算法,从而可以得出 BAGSSA 具有连续收敛的能力,收敛速度明显快于其他方法,这得益于 BAGSSA 具有更好的开发能力。当测试函数为 F_7, F_8, F_{11} 时,算法结果与其他对比算法的精度是相似的,值略优于其他对比算法。当测试函数为 F_6, F_{10}, F_{12} 时, BAGSSA 算法的均值与 CASSA 和 MSNSSA 是一致的。对单峰和多峰测试函数的结果表明, BAGSSA 在所有函数都具有竞争性,标准差证明本文算法的优越性是比较稳定的。根据 Wilcoxon 符号秩检验的统计,可以发现 BAGSSA 算法与其他算法有显著的差异。

由图 2 的收敛曲线可知,当测试函数为 F_3, F_4, F_{11} 时, BAGSSA 的适应度值一旦达到某一个量级,收敛速度明显优于其他算法,说明 BAGSSA 跳出局部极值的能力也是更优的。从图 2 所示的 F_{12} 收敛曲线可以得出,改进算法 CASSA 和 MSNSSA 虽然精度均可以达到最优值,但是本文算法的迭代次数却更少。BAGSSA 能获得较好收敛能力的原因是,它在樽海鞘群中的跟随者具有无标度的拓扑相关性,最优个体被赋予度最大的节点位置。与基本的 SSA 相比,这种 BAGSSA 的主要群体相当于在其位置更新中增加了一个更加全面的指导。

众所周知,优化算法在多峰测试函数中非常容易陷入局部最优解。而 BAGSSA 可以借助自适应权重策略来平衡整个群体的拥挤程度。当群体在决策空间中拥挤时,该机制将引导跟随者通过式(11)向相应的邻居扩散,从而使 BAGSSA 获得跳出局部最优解的机会。

5.4 时间复杂度分析

根据 BAGSSA 算法所提出的改进的伪代码可以看出, BAGSSA 算法的计算复杂度主要体现在初始化、排序、位置更新、自适应权重、拓扑关系的搜索和黄金正弦算子变异机制等方面。

算法的初始化阶段依赖于随机函数,它的时间复杂度为:

$$T(N)_1 = O(N \times \text{dim})$$

排序时采用了快速排序算法,时间复杂度为:

$$T(n)_2 = O(N^2)$$

所提出的自适应权重的时间复杂度为:

$$T(N)_3 = O(N \times \text{dim})$$

搜索拓扑关系需要 $O(1)$ 的运算,然而重复以上过程的次数就是算法所需要迭代的次数,均为 $T_{\max}$,因此整个位置更新过程所需要的时间复杂度为:

$$T(N)_4 = O(T_{\max} \times N \times \text{dim})$$

采用黄金正弦变异算子机制之后最坏情况下的时间复杂度为:

$$T(N)_5 = O\left(\frac{T_{\max}}{\tau} \times N \times \tau\right)$$

将以上数据进行整理之后, BAGSSA 算法的时间复杂度的上界为:

$$T(N) = O(T_{\max} \times (N^2 + N \times \text{dim}))$$

6 BAGSSA 在路径规划中的应用

随着智能化时代的到来,路径规划问题也作为近年来的热门问题之一,已经被许多的学者研究学习,本文将改进后的 BAGSSA 算法应用于路径规划问题。

6.1 环境建模

本节主要研究在静态的环境下,移动机器人如何寻找最优路径问题,地图环境建模对机器人路径规划及其重要。栅格法^[25]具有简单高效的实用性,因此选择栅格法建立仿真环境。该方法就是将二维平面划分成长度和宽度相同的网格,每个网格的边长为 1,在编码时用 1 代表障碍物,就是环境建模中所看到的黑色区域;0 代表无障碍方格,是其中的白色区域。模拟机器人路径规划的两种栅格环境如图 3(a)、图 3(b)所示。

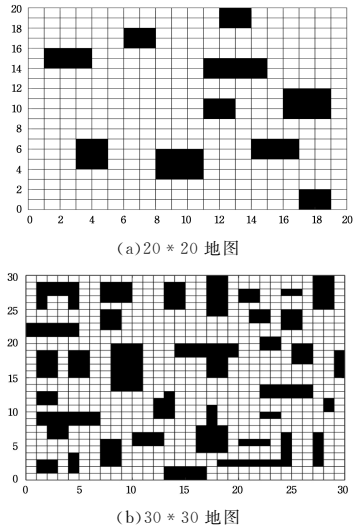


图 3 环境模型

Fig. 3 Model of environment

6.2 构建目标函数

在进行路径规划时,需要考虑路径长度最短这一因素,基于这个因素来构建本次实验的目标函数,其表达式为:

$$f(L) = \sum_{i=1}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{15}$$

其中, $f(L)$ 是当前的路径长度, (x_i, y_i) 是当前的路径坐标。

6.3 实验设置

本节对所提算法进行模拟仿真,并与第 5.3 节所涉及的基本樽海鞘群算法 SSA, CASSA, MSNSSA, PSO, DE 进行对比。所有算法的初始种群规模均为 30, 迭代次数为 100 次, 将 6 种算法在两种实验环境下独立运行 30 次并记录最优路径、平均路径以及平均寻优耗时, 比较结果如表 4 所列, 算法路径规划如图 4 所示。

表 4 不同环境下 6 种算法的实验数据对比

Table 4 Comparison of experimental data of 6 algorithms in different environments

测试环境	指标	SSA	CASSA	MSNSSA	PSO	DE	BAGSSA
环境 1	最优路径长度	27.6788	28.0234	27.6785	28.1256	27.6788	26.0532
	平均路径长度	29.3652	29.9653	29.5426	29.6788	30.9854	26.3653
	时间/s	1.9562	1.9125	1.8451	1.8256	1.8640	1.0680
环境 2	最优路径长度	45.3651	46.3251	46.3310	44.6464	45.3198	43.3214
	平均路径长度	50.2545	49.0214	51.3232	49.5244	50.3652	44.3254
	时间/s	3.0696	2.8547	2.9241	3.1552	3.6521	2.0474

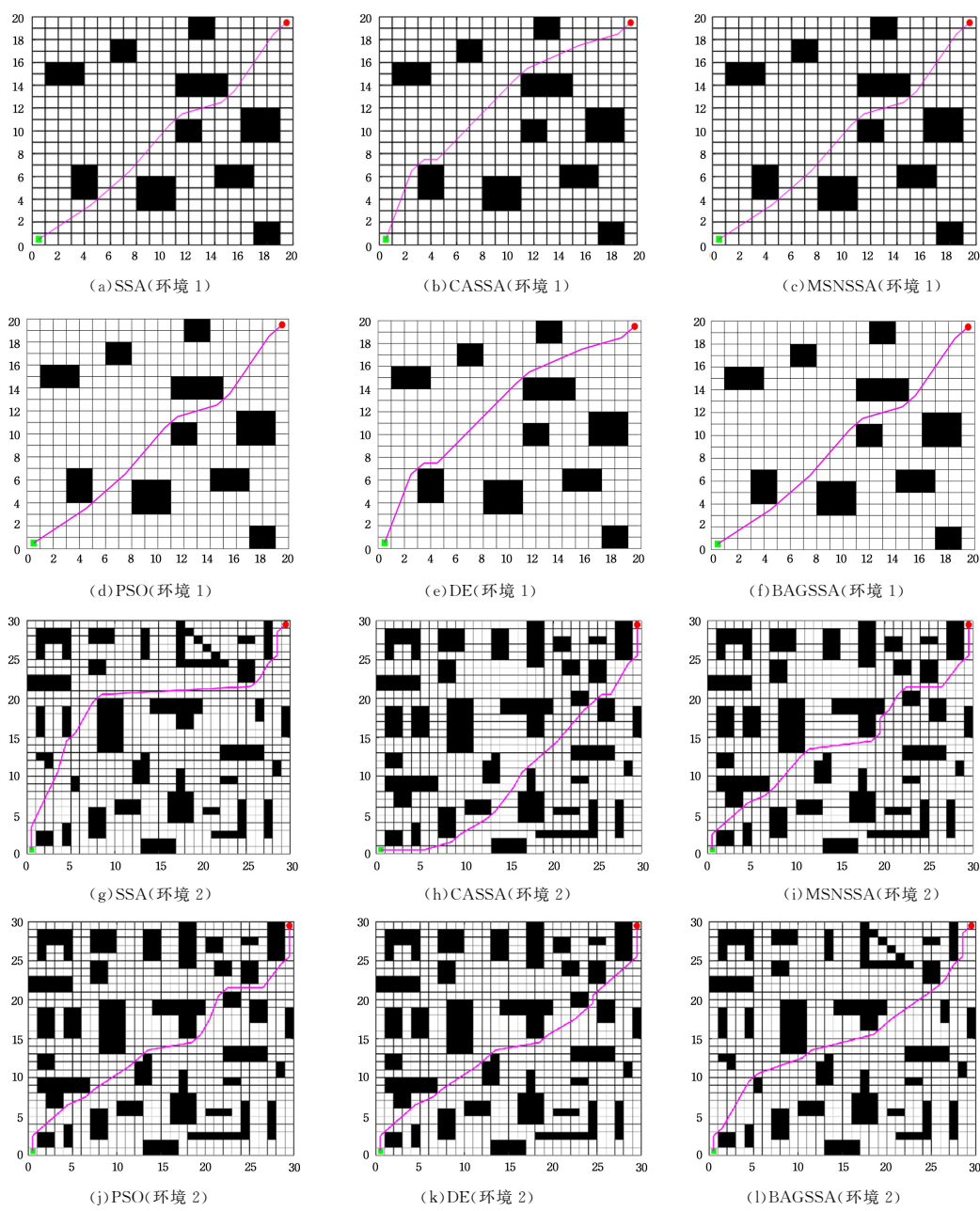


图 4 不同算法的规划路径对比

Fig. 4 Comparison of planning paths of different algorithms

由表 4 可知,本文算法在障碍物较为简单的环境 1 中最优路径长度较其他 5 种算法平均同比下降了 5% 左右,平均路径长度则是同比下降了 10% 左右,平均耗时也优于其他 5 种算法;在障碍物较为复杂的环境 2 中,本文算法的最优路径长度较其他算法缩短了 6% 左右,平均路径长度缩短了 13% 左右,平均耗时同样优于其他算法。以上分析可以说明 BAGSSA 算法在障碍物更多的环境中,具有更好的寻优能力。从图 4 中可以看出,BAGSSA 算法的无标度网络在前期可以合理规划轨迹,融合自适应权重和黄金正弦变异策略在后期能够帮助算法调整路径,更进一步缩短了规划路径。

结束语 本文提出了一种改进的樽海鞘群算法的跟随者机制,以加强算法的探索 and 开发能力。为解决传统 SSA 算法收敛精度低、速度慢、难以跳出局部最优等问题,本文改进了一种融合无标度网络的跟随者机制,由此提高了全局的收敛速度。同时,加入了自适应权重策略去平衡算法迭代过程中的局部搜索能力,并对最差个体进行黄金正弦算子变异。

通过对 12 个基准函数进行求解,对改进策略进行了有效性分析,且使用 Wilcoxon 秩和检验对算法显著性水平进行验证。通过实验充分论证了本文改进的 BAGSSA 算法的有效性,同样 BAGSSA 在解决路径规划工程问题也得到了令人满意的实验结果,符合本文算法改进的初衷,下一步计划将改进算法应用在不同的工程实践问题中。

参考文献

[1] WANG H W, MA Y, XIE Y, et al. Path Planning for Mobile Robots Based on Smooth A* Algorithm[J]. Journal of Tongji University(Natural Science Edition), 2010, 38(11): 1647-1655.

[2] YU Z Z, YAN J H, ZHAO J, et al. Improved artificial potential field method for mobile robot path planning [J]. Journal of Harbin Institute of Technology, 2011, 43(1): 50-55.

[3] LI W S, SUN J, CHEN W. Real-time obstacle avoidance algorithm for robots based on BP neural network [J]. Journal of In-

- strumentation, 2019, 40(11):204-211.
- [4] ZHANG F H, LI N, YUAN R P, et al. Robot path planning algorithm based on reinforcement learning [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2018, 46(12):65-70.
 - [5] ZHANG Q, CHEN B K, LIU X Y, et al. Optimal Path Planning for Mobile Robots Based on Improved Potential Field Ant Colony Algorithm [J]. Journal of Agricultural Machinery, 2019, 50(5):23-32, 42.
 - [6] WEI T G, LONG C N. Mobile robot path planning based on improved genetic algorithm [J]. Journal of Beijing University of Aeronautics and Astronautics, 2020, 46(4):703-711.
 - [7] KENNEDY J, EBERHART R. Particle swarm optimization [C] // Proceedings of ICNN '95-International Conference on Neural Networks. IEEE, 1995, 4:1942-1948.
 - [8] MIRJALILI S, GANDOMI A H, MIRJALILI Z, et al. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems [J]. Advances in engineering software, 2017, 114:163-191.
 - [9] SAYED G I, KHORIBA G, HAGGAG H. A novel chaotic salp swarm algorithm for global optimization and feature selection [J]. Applied Intelligence, 2018, 48:3462-3481.
 - [10] YANG B, ZHONG L, ZHANG X, et al. Novel bio-inspired memetic salp swarm algorithm and application to MPPT for PV systems considering partial shading condition [J]. Journal of cleaner production, 2019, 215:1203-1222.
 - [11] YU J S, WU L. TWO Types of Leaders Salp Swarm Algorithm [J]. Computer Science, 2021, 48:254-260.
 - [12] HEGAZY A E, MAKHLOUF M A, EL-TAWEL G S. Improved salp swarm algorithm for feature selection [J]. Journal of King Saud University-Computer and Information Sciences, 2020, 32(3):335-344.
 - [13] ZHANG H, LIU T, YE X, et al. Differential evolution-assisted salp swarm algorithm with chaotic structure for real-world problems [J]. Engineering with Computers, 2023, 39(3):1735-1769.
 - [14] NASRI D, MOKEDDEM D, BOUROUBAB, et al. A novel levy flight trajectory-based salp swarm algorithm for photovoltaic parameters estimation [J]. Journal of Information and Optimization Sciences, 2021, 42(8):1841-1867.
 - [15] ZHANG Z Q, LU X F, SUI L S, et al. A Salp Swarm Algorithm Integrating Random Inertial Weights and Differential Mutation Operations [J]. Computer Science, 2020, 47(8):297-301.
 - [16] REN H, LI J, CHEN H, et al. Stability of salp swarm algorithm with random replacement and double adaptive weighting [J]. Applied Mathematical Modelling, 2021, 95:503-523.
 - [17] DING M F, WU K Q, XIAO P. Golden Sine Salp Group Algorithm Based on Multi-Strategy Fusion [J/OL]. Journal of Nanjing University of Information Technology (Natural Science Edition): 1-13. [2023-03-20]. <http://kns.cnki.net/kcms/detail/32.1801.N.20230313.1346.002.html>.
 - [18] ZHANG Y, QIN L X. Improved Salp Swarm Algorithm Based on Levy Flight Strategy [J]. Computer Science, 2020, 47(7):154-160.
 - [19] BARABÁSI A L, ALBERT R. Emergence of scaling in random networks [J]. Science, 1999, 286(5439):509-512.
 - [20] TANYILDIZI E, DEMIR G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm [J]. Advances in Electrical & Computer Engineering, 2017, 17(2):71-78.
 - [21] DERRAC J, GARCÍA S, MOLINAD, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms [J]. Swarm and Evolutionary Computation, 2011, 1(1):3-18.
 - [22] ZHANG D M, CHEN Z Y, XIN Z Y, et al. Crazy Adaptive Salp Swarm Algorithm [J]. Control and Decision, 2020, 35(9):2112-2120.
 - [23] CHEN Z Y, ZHANG D M, XIN Z Y. Multi-subgroup symbiotic non-uniform Gaussian variant salp group algorithm [J]. Acta Automatica Sinica, 2022, 48(5):1307-1317.
 - [24] STORN R, PRICE K. Differential evolution-a simple and efficient Heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11(4):341.
 - [25] MAURER C R, QI R, RAGHAVAN V. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(2):265-270.



ZHAO Hongwei, born in 1976, doctor, professor, is a member of CCF (No. 32583S). His main research interests include big data and artificial intelligence, complex modeling and analysis, edge computing.



DONG Changlin, born in 1999, is a graduate student and a member of CCF (No. H1948G). His main research interests include complex modeling and analysis.