

基于佳点集和惯性权重的改进麻雀算法

孙夏丽, 李士心*, 刘清清, 王 坤

天津职业技术师范大学电子工程学院, 天津

收稿日期: 2021年8月26日; 录用日期: 2021年9月15日; 发布日期: 2021年9月28日

摘 要

针对麻雀算法(SSA)局部搜索能力差的问题, 提出一种改进的麻雀算法(GSSA)。首先, 采用佳点集的方法初始化麻雀个体, 增强种群多样性; 其次, 在发现者位置更新上采用对数惯性权重来协调局部搜索和全局搜索能力, 加快收敛速度; 同时在跟随者位置更新方式中引入 t 分布策略, 加强全局搜索能力; 最后, 对6个基准测试函数进行仿真实验表明, GSSA寻优精度与SSA算法相比可提高约51个数量级, 与同类改进算法相比精度可提高2个数量级, 且寻优速度加快。

关键词

麻雀搜索算法, 佳点集方法, 惯性对数递减, t 分布

Improved Sparrow Algorithm Based on Good Point Set and Inertia Weight

Xiali Sun, Shixin Li*, Qingqing Liu, Kun Wang

School of Electronic Engineering, Tianjin University of Technology and Education, Tianjin

Received: Aug. 26th, 2021; accepted: Sep. 15th, 2021; published: Sep. 28th, 2021

Abstract

In order to solve the problem of poor local search ability of sparrow algorithm, this paper proposes an improved sparrow algorithm (GSSA). Firstly, the best point set method is used to initialize individual position of sparrow, which lays the foundation for the diversity of global search; Secondly, log inertia weight is used to coordinate the ability of local search and global search in the location update of discoverer, and the convergence speed is accelerated; Then, the t distribution strategy is introduced into the follower position update mode to enhance the global search ability;

*通讯作者。

Finally, the simulation experiments on six benchmark functions show that the optimization accuracy of GSSA can be improved by about 51 orders of magnitude compared with SSA algorithm, 2 orders of magnitude compared with similar improved algorithms, and the optimization speed is accelerated.

Keywords

Sparrow Search Algorithm, Good Point Set Method, Inertial Troy Decreasing, t Distribution

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

仿生群智能算法是通过提取生物种群特征而凝练成的智能优化算法,如:粒子群算法(PSO) [1]、灰狼算法(GWO) [2]、鲸鱼算法(WOA) [3]等新型优化算法,其中,麻雀搜索算法(SSA) [4]是薛建凯受麻雀觅食和反捕食行为启发于2020年提出来的,这些算法简单适用性强,受诸多研究者的关注。

初始种群的多样性和遍历性对算法的准确度有一定的影响,针对初始化策略众多学者提出相应的改进方法。文献[5]和[6]分别采用改进 Tent 混沌序列和映射折叠次数无限的 Sin 混沌初始化麻雀种群,从而提高了初始解的质量;汤安迪[7]等通过对比立方混沌映射和 Logistic 映射,采用立方混沌映射初始化种群,增强种群多样性,扩大了麻雀搜索范围;石建平[8]等引用佳点集初始化种群并及时更新双策略协调进化的果蝇优化算法,使初始种群有较好的多样性,且加速了算法的收敛速度;龙文[9]等引入佳点集生成初始种群,为算法的全局搜索奠定基础,上述改进都是基于提高初始种群质量来优化算法性能。

迭代后期,种群多样性减弱,算法易陷入局部最优,为了协调全局和局部的搜索能力,使其跳出局部最优,找到全局最优,很多研究者在原有算法的基础上进行优化。王涛[10]将线性权重改为非线性权重,提高了勘探和开发能力;Saxena等[11]提出基于 β -混沌序列的自适应桥接机制来调节控制参数 a ,用于改进GWO算法,提高全局和局部的搜索能力;刘长安[12]等融合反向学习策略和自适应 t 分布变异,引入精英粒子,来扩大麻雀算法的搜索范围,增强算法后期的局部搜索能力;蔡艺君[13]等采用带随机扰动项的指数递减惯性权重更新跟随者的位置,平衡全局和局部搜索能力;王义[14]等引入自适应惯性权重因子平衡蜉蝣算法的搜索和开发能力。

为了有效地避免这些缺陷,提出一种基于佳点集和惯性权重的改进麻雀算法,主要在以下三方面进行改进。首先,采用佳点集的方式生成初始种群,保证种群个体的多样性;其次,采用对数惯性权重递减策略对发现者的位置进行更新,以平衡全局和局部搜索能力;最后,采用 t 分布策略对跟随者的位置进行更新,避免算法陷入局部最优,提高全局搜索能力。仿真实验表明,改进的麻雀算法在寻优性能上较原来的算法有一定的提高。

2. 基本麻雀搜索算法

SSA算法[4]与其他算法相比,具有收敛速度快,稳定性好等优点,但易陷入局部最优解。麻雀觅食由发现者、跟随者和预警者组成的。发现者为麻雀种群提供觅食区域和方向,跟随者来获得食物,当周围有捕食者时,群体中一个或多个预警者会发出声音,整个群体会飞到其他安全区域进行觅食。

设 n 只麻雀组成的群体及对应适应度函数分别可表示为: $X = [x_1, x_2, \dots, x_n]^T$ 和 $F = [f(x_1), f(x_2), \dots, f(x_n)]^T$ 。在迭代的过程中, 发现者的位置更新如下:

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

其中, i 为迭代次数; $j=1, 2, \dots, d$ 。 T_{\max} 为最大迭代次数; X_{ij} 为第 i 只麻雀在第 j 维中的位置; $\alpha \in (0, 1]$ 为随机数; $R_2 (R_2 \in [0, 1])$ 和 $ST (ST \in [0.5, 1])$ 分别为预警值和安全值; Q 是服从正态分布的随机数; L 为 $1 \times d$ 的全 1 矩阵。当 $R_2 < ST$ 时, 种群处于安全环境, 发现者可广泛搜索; 当 $R_2 \geq ST$ 时, 侦查者发现捕食者并释放危险信号, 种群立即向安全区域靠拢。

跟随者的位置更新为:

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

式中, X_p 、 X_{worst} 分别为当前全局最优和最差的位置。 A 为 $1 \times d$ 的全 1 或 -1 矩阵, 且 $A^+ = A^T (AA^T)^{-1}$ 。当 $i > n/2$ 时, 适应度值较低的第 i 个加入者为得到较高的能量, 飞往其他地方。

预警麻雀占种群的 10%~20%, 位置更新为:

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

其中, X_{best} 为当前全局最优位置。 β 为步长控制参数, 是服从标准正态分布的随机数。 $K \in [-1, 1]$ 是一个随机数, f_i 、 f_g 和 f_w 分别为当前麻雀个体、全局最佳和最差的适应度值。 ε 为最小的常数, 可避免分母为零。当 $f_i = f_g$ 时, 表示麻雀处于种群中间, 当危险来临时, 及时靠近其他麻雀来调整搜索策略。

3. 改进麻雀搜索算法

3.1. 佳点集初始化麻雀种群

初始种群均匀分布可确保种群的多样性和遍历性, 提高算法的搜索性, 在 SSA 中采用随机初始化种群个体, 难以保证种群的多样性。佳点集是一种分布均匀、有效的选点方式, 利用佳点集的均匀性初始化种群, 可以提高种群的多样性。目前, 佳集点初始化种群的方式已经应用到很多智能算法[15] [16]中, 并取得了有效的结果, 故采用佳点集进行种群初始化。

3.2. 惯性权重对数递减策略

对群体智能算法而言, 平衡全局搜索能力和局部搜索能力对算法性能影响较大。SSA 在找到最优解后迅速向最优解靠拢, 很难协调全局与局部之间的性能, 故易陷入局部最优, 为此, 引入对数惯性权重 ω 来平衡全局和局部的搜索能力, 在搜索前期, 收敛速度快, 具有较高的全局搜索能力可以快速找到全局最优; 在搜索后期, 收敛速度慢, 具有局部搜索能力, 可以提高收敛速度, 故采用惯性权重对数递减策略[17], 即

$$\omega = (\omega_{\max} - \omega_{\min}) (1 - \alpha \times \log_{\text{iter}_{\max}} k) \quad (4)$$

其中， α 为对数调整因子， ω_{\max} 、 ω_{\min} 分别为最大和最小权重， k 为迭代次数。

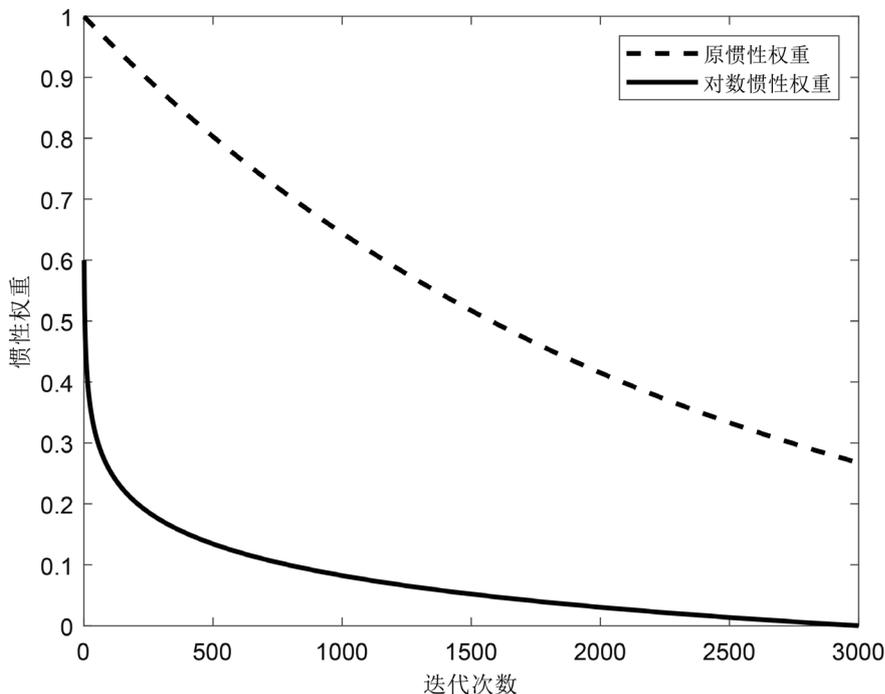


Figure 1. Comparison of law curves of original inertia weight and logarithmic inertia weight
图 1. 原惯性权重与对数惯性权重规律曲线对比图

图 1 为原惯性权重与对数惯性权重的规律曲线对比图。在迭代的过程中，随着对数惯性权重的不断变化，全局搜索与局部搜索得到平衡。发现者改进后的位置更新方式如下：

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \omega, & R_2 < ST \\ X_{i,j}^t + Q \cdot L, & R_2 \geq ST \end{cases} \quad (5)$$

其中， ω 为惯性权重。

3.3. 自适应 t 分布改进策略

t 分布(学生分布)的曲线形态与参数自由度有关。自由度 n_t 越大，曲线形态表现越高耸，曲线越接近于标准正态分布曲线；当自由度 n_t 无限大时， t 分布为高斯分布，即 $t(n_t \rightarrow \infty) \rightarrow N(0,1)$ ，具有较强的局部搜索能力；自由度 n_t 越小，曲线形态表现越低平；当自由度 $n_t = 1$ 时， t 分布为柯西分布，即 $t(n_t = 1) \rightarrow C(0,1)$ ，具有较强的全局搜索能力。

对麻雀位置利用自适应 t 分布[18]的变异策略进行更新如下所示：

$$x_i^{t+1} = x_i^t + x_i^t * t(iter) \quad (6)$$

其中， x_i^{t+1} 为变异后的麻雀位置； x_i^t 为第 i 个麻雀的位置； $t(iter)$ 为以算法的迭代次数为参数自由度的 t 分布。

当发现者迭代一定次数后，跟随者变为发现者，继续种群提供觅食区域和方向，为避免局部最优，在跟随者更新中引入 t 分布策略，提高全局搜索能力。改进后的跟随者的位置更新方式如下：

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}} - X_{i,j}^t}{t^2}\right), & i > \frac{n}{2} \\ X_p^{t+1} + X_p^{t+1} * t(\text{iter}), & \text{otherwise} \end{cases} \quad (7)$$

其中, X_p^{t+1} 为当前发现者占据的最佳位置。

3.4. 算法流程

针对以上分析的 SSA 算法缺点, GSSA 算法采用佳点集进行种群初始化, 且在发现者中引入对数惯性权重, 在跟随者中引入 t 分布策略, 具体实现步骤如下:

step1: 参数初始化。包括最大迭代次数、种群规模、发现者数量 PDn、预警者数量 SD、初始值上下界;

step2: 种群初始化。根据 3.1 节中的佳点集初始化麻雀种群;

step3: 计算每只麻雀的适应度值, 并得到麻雀的最佳适应度和最差适应度值的个体位置;

step4: 选取适应度值前 PDn 只麻雀为发现者, 并根据式(5)进行位置更新;

step5: 剩余的个体作为跟随者, 采用式(7)对其进行位置更新;

step6: 从种群中选取 SD 只作为警戒者, 采用式(3)的方式进行位置更新;

step7: 完成本次迭代后, 重新计算个体的适应度值, 并记录全局最优解和其适应度值, 及最差位置及其适应度值;

step8: 判断算法是否满足最大迭代次数要求, 若没有, 则返回步骤 3, 继续进行迭代计算; 否则, 返回循环结束, 输出最优结果。

4. 仿真实验与结果分析

4.1. 实验环境与参数设置

为验证 GSSA 算法的性能, 选取粒子群算法(PSO)、灰狼算法(GWO)、基本麻雀算法(SSA)及文献[19]所提出的正余弦和 Levy 飞行麻雀算法(ISSA)在基准函数上进行测试对比, 通用条件以文献[19]为准, 表 1 为算法参数设置, 表 2 为基准测试函数参数。

Table 1. Parameter setting table

表 1. 参数设置表

参数设置			
PSO	$c_1 = c_2 = 1, \omega = 0.3, V_{\max} = 2, V_{\min} = -2$	ISSA	PD = 20%, SD = 10%
GWO	α 从 2 递减至 0	GSSA	PD = 20%, SD = 10%
SSA	PD = 20%, SD = 10%		

Table 2. Benchmark function

表 2. 基准测试函数

函数名	函数(维度)	取值范围	最优值	函数名	函数(维度)	取值范围	最优值
Sphere	f_1 (30)	[-100, 100]	0	Rosenbrock	f_4 (10)	[-100, 100]	0
Schwefel2.22	f_2 (50)	[-10, 10]	0	Rastrigin	f_5 (30)	[-5.12, 5.12]	0
Schwefel1.2	f_3 (100)	[-100, 100]	0	Schwefel2.1	f_6 (30)	[-100, 100]	0

4.2. 算法性能结果对比

本文采用 Inter(R) Core(TM) i7-4700HQ CPU @ 2.40 GHz, 内存 8.00 GB, Window7 系统和 matlab R2019b 系统对算法进行仿真实验, 为了增强实验数据的说服力, 各个算法在每个基准函数上独立运行 30 次, 实验结果见表 3。

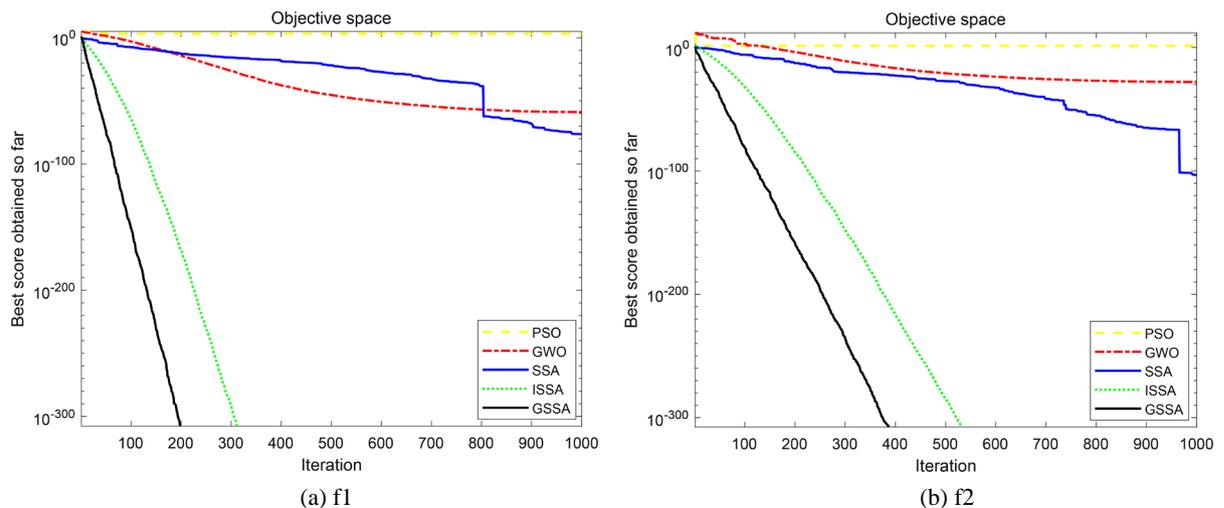
Table 3. Comparison of benchmark function optimization results
表 3. 基准测试函数优化结果比较

统计量	算法	f_1	f_2	f_3	f_4	f_5	f_6
平均值	PSO	1.63E+03	2.83E+01	1.19E+05	1.63E+05	1.79E+02	1.52E+01
	GWO	2.86E-58	1.25E-23	6.84E+00	2.70E+01	7.39E-01	1.02E-14
	SSA	4.11E-45	2.71E-51	3.31E-16	2.07E-04	0.00E+00	1.82E-11
	ISSA	0.00E+00	0.00E+00	0.00E+00	5.61E-05	0.00E+00	0.00E+00
	GSSA	0.00E+00	0.00E+00	0.00E+00	2.77E-06	0.00E+00	0.00E+00
标准差	PSO	5.59E+2	4.67E+00	5.60E+04	7.56E+04	2.33E+01	3.14E+00
	GWO	1.05E-57	4.23E-23	1.69E+01	0.82E+00	2.87E+00	1.04E-14
	SSA	2.20E-44	1.46E-50	1.13E-15	3.86E-04	0.00E+00	5.53E-11
	ISSA	0.00E+00	0.00E+00	0.00E+00	1.81E-04	0.00E+00	0.00E+00
	GSSA	0.00E+00	0.00E+00	0.00E+00	6.60E-06	0.00E+00	0.00E+00

由表 3 可知, GSSA 求解的各基准测试函数的均值比其他三种算法更接近全局最优, 精确度高; GSSA 的标准差更小, 稳定性高。GSSA 在原算法的基础上最多提升了 50 个数量级, 提升效果较大, 在函数 f_4 上, 比 ISSA 在均值和标准差上提升了 2 个数量级, 说明 GSSA 的寻优效果更好。

4.3. 算法收敛曲线分析

为了更直观的体现 GSSA 算法的收敛性能, 图 2(a)~(f)展示了 6 种算法的迭代收敛曲线。横坐标为迭代次数, 纵坐标为适应度 Log 的值。



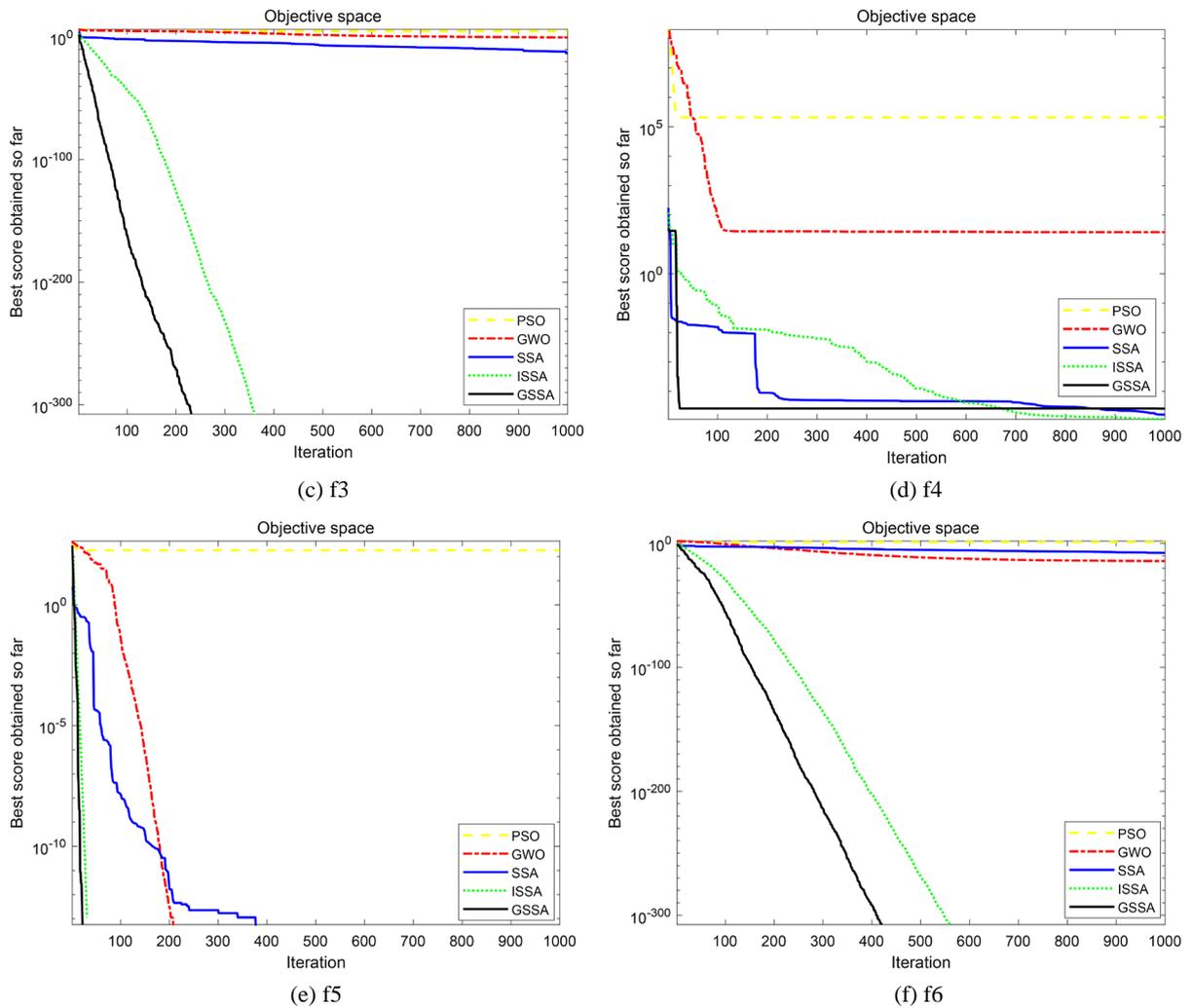


Figure 2.6 benchmark function convergence curves
图 2.6 6 个基准测试函数收敛曲线

由以上的收敛曲线可知，GSSA 相较于其他算法都具有更佳的收敛性，GSSA 斜率小，收敛速度快，收敛精度高；其中图 2(a)~(c)、图 2(e)、图 2(f)收敛曲线平滑，拐点较少；在图 2(d)上虽然有拐点，但求解精度高。

5. 结论

本文提出一种基于优化初始解，更新发现者和跟随者的个体位置进行改进的麻雀算法。利用佳点集的均匀性和遍历性来改善初始种群的质量；引入对数惯性权重对发现者的位置更新更新以平衡全局搜索和局部开发能力；利用自适应 t 分布的全局搜索性来更新跟随者个体位置。通过在 6 个基准测试函数上进行仿真实验，结果表明，GSSA 比 PSO、GWO、SSA、ISSA 在求解速度和精确度上有更强的优化性能。

基金项目

天津市科技特派员项目(19JCTPJC41500)。

参考文献

- [1] Kennedy, J. and Eberhart, R.C. (1995) Particle Swarm Optimization. *IEEE International Conference on Neural Networks*, Volume 4, 1942-1948.
- [2] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [3] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [4] Xue, J. and Shen, B. (2020) A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Systems & Control Engineering: An Open Access Journal*, **8**, 22-34. <https://doi.org/10.1080/21642583.2019.1708830>
- [5] 吕鑫, 慕晓冬, 张钧, 等. 混沌麻雀搜索优化算法[J/OL]. 北京航空航天大学学报, 2021, 47(8): 1-10. <https://doi.org/10.13700/j.bh.1001-5965.2020.0298>, 2021-06-16.
- [6] 毛清华, 张强. 融合柯西变异和反向学习的改进麻雀算法[J]. 计算机科学与探索, 2021, 15(6): 1155-1164.
- [7] 汤安迪, 韩统, 徐登武, 等. 基于混沌麻雀搜索算法的无人机航迹规划方法[J/OL]. 计算机应用, 1-11. <http://kns.cnki.net/kcms/detail/51.1307.TP.20201124.1519.002.html>, 2021-06-16.
- [8] 石建平, 刘国平, 李培生, 等. 双策略协同进化果蝇优化算法及其应用[J/OL]. 计算机集成制造系统, 1-16. <http://kns.cnki.net/kcms/detail/11.5946.TP.20201125.1815.004.html>, 2021-06-16.
- [9] 龙文, 赵东泉, 徐松金. 求解约束优化问题的改进灰狼优化算法[J]. 计算机应用, 2015, 35(9): 2590-2595.
- [10] 王涛. 非线性权重和柯西变异的蝗虫算法[J]. 微电子学与计算机, 2020, 37(5): 82-86.
- [11] Saxena, A., Kumar, R. and Das, S. (2019) β -Chaotic Map Enabled Grey Wolf Optimizer. *Applied Soft Computing*, **75**, 84-105. <https://doi.org/10.1016/j.asoc.2018.10.044>
- [12] 柳长安, 冯雪菱, 孙长浩, 赵丽娟. 基于改进麻雀算法的最大二维熵分割方法[J/OL]. 激光技术, 1-15. <http://kns.cnki.net/kcms/detail/51.1125.tn.20210423.1018.004.html>, 2021-06-16.
- [13] 蔡艺君, 贺兴时, 杨新社. 基于指数惯性权重和自适应变异的樽海鞘算法[J/OL]. 纺织高校基础科学学报, 2021, 34(2): 108-116.
- [14] 王义, 张达敏, 张琳娜, 等. 基于黄金正弦与自适应融合的蜉蝣优化算法[J/OL]. 计算机应用研究, 1-7. <https://doi.org/10.19734/j.issn.1001-3695.2021.03.0042>, 2021-06-16.
- [15] 龙文, 伍铁斌. 协调探索和开发能力的改进灰狼优化算法[J]. 控制与决策, 2017, 32(10): 1749-1757.
- [16] 陈义雄, 梁昔明, 黄亚飞. 基于佳点集构造的改进量子粒子群优化算法[J]. 中南大学学报(自然科学版), 2013, 44(4): 1409-1414.
- [17] 戴文智, 杨新乐. 基于惯性权重对数递减的粒子群优化算法[J]. 计算机工程与应用, 2015, 51(17): 14-19+52.
- [18] 于建芳, 刘升, 韩斐斐, 肖子雅. 基于柯西变异的蚁狮优化算法[J]. 微电子学与计算机, 2019, 36(6): 45-49+54.
- [19] 毛清华, 张强, 毛承成, 等. 混合正弦余弦算法和 Lévy 飞行的麻雀算法[J/OL]. 山西大学学报(自然科学版): 1-6. <https://doi.org/10.13451/j.sxu.ns.2020135>, 2021-09-23.