

融合改进 Sine 混沌映射的新型粒子群优化算法

刘磊¹, 姜博文¹, 周恒扬¹, 浦晨玮¹, 钱鹏飞^{1,2}, 刘波³

(1. 江苏大学机械工程学院, 212013, 江苏镇江; 2. 浙江大学流体动力基础件与机电系统全国重点实验室, 310027, 杭州; 3. 法赫德国王石油与矿业大学电气工程系, 31261, 沙特阿拉伯达兰)

摘要: 为了应对传统粒子群算法(PSO)存在的初始位置不均匀、易达到局部最优、搜索精度不高等问题,提出了一种基于改进 Sine 混沌映射的新型 PSO 算法。采用一种改进的 Sine 混沌映射技术代替传统的伪随机数方法生成初始粒子种群,以丰富种群的多样性。在原始基本位置更新公式的基础上增加两种新的位置更新机制,并分别引入一个高斯变异算子,以实现算法勘探性能和开发性能之间的动态平衡,以及在迭代过程中使粒子有效跳出局部最优。在由 7 个单峰函数、6 个多峰函数和 10 个固定维函数组成的基准测试函数和 3 个带约束经典工程优化设计问题上对所提出算法开展仿真实验,并与其他几种流行的 PSO 变体进行对比。仿真结果表明:与其他 PSO 变体相比,基于改进 Sine 混沌映射的新型 PSO 算法具有更快的收敛速度和更高的寻优精度,对于基准测试函数的寻优结果有 20 个排名第一,约为总测试函数的 87%;该算法在压力容器和工字梁设计优化中,综合性能排在第一位,应可用于解决一些实际工程优化问题。

关键词: 粒子群算法;混沌映射;高斯变异;基准函数;工程问题

中图分类号: TP301 **文献标志码:** A

DOI: 10.7652/xjtubxb202308018 **文章编号:** 0253-987X(2023)08-0182-12

A Novel Particle Swarm Optimization Algorithm Incorporating Improved Sine Chaotic Mapping

LIU Lei¹, JIANG Bowen¹, ZHOU Hengyang¹, PU Chenwei¹, QIAN Pengfei^{1,2}, LIU Bo³

(1. School of Mechanical Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China;

2. The State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China;

3. Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia)

Abstract: In order to address the problems of uneven initial positions, ease of reaching local optimum, and low search accuracy in traditional particle swarm algorithm (PSO), a novel PSO algorithm based on an improved Sine chaotic mapping is proposed. The improved Sine chaotic mapping technique is used instead of the traditional pseudo-random number method for generating the initial particle population to enrich the population diversity. Two new position update mechanisms are added to the original basic position update formula. A Gaussian mutation operator is introduced to achieve a dynamic balance between the exploration and exploitation performance of the algorithm, as well as to help particles effectively jump out of the local optima during the iteration.

收稿日期: 2023-03-30。 作者简介: 刘磊(1998—),男,博士生;钱鹏飞(通信作者),男,副教授,博士生导师。 基金项目: 国家自然科学基金资助项目(52075223);中国博士后科学基金资助项目(2021M691308);江苏省研究生科研与实践创新计划资助项目(KYCX23_3731)。

网络出版时间: 2023-05-15

网络出版地址: <https://kns.cnki.net/kcms/detail/61.1069.T.20230512.1619.002.html>

tion process. For three classical engineering optimization design problems with constraints, simulation experiments are performed for the proposed algorithm based on a benchmark test function consisting of seven single-peaked functions, six multi-peaked functions and ten fixed-dimensional functions. This algorithm is then compared with several other popular PSO variants. Simulation results show that the novel PSO algorithm based on improved Sine chaotic mapping has faster convergence speed and higher optimization-seeking accuracy than those of other PSO variants. For the benchmark test functions, it ranked first in 20 of them, accounting for about 87% of the total testing functions. The proposed algorithm ranked first in the overall performance of pressure vessel and I-beam design optimization, and can be used to solve some practical engineering optimization problems.

Keywords: PSO algorithm; chaotic mapping; Gaussian mutation; benchmark functions; engineering problems

粒子群(PSO)算法^[1]是一种群智能搜索优化算法,该算法于 20 世纪 90 年代被 Eberhart 等提出,由于其仅有个别参数需要调整,实现起来十分容易,得到了诸多行业内学者的关注^[2-4]。与众多元启发式算法^[5-7]一样,传统 PSO 算法由于不能有效地平衡全局搜索与局部搜索之间的关系、缺乏陷入局部最优的有效解决方案以及难以保证种群多样性等,导致在实际应用中存在收敛速度慢、搜索精度不高和易达到局部最优等问题^[8-9]。

为此,Shi 等添加了惯性权重因子以解决 PSO 算法通用性,形成了基本粒子群算法^[10]。为了改善 PSO 算法在解决高维复杂问题上的表现,Shi 等将惯性权重因子修改为从 0.9 到 0.4 线性递减的系数^[11]。后来,又将惯性权重更改为介于 0.5 到 1 之间的随机数,即 PSO-RIW 算法,实验结果表明该算法具有更好的鲁棒性^[12]。Ratnaweera 等引入了可变加速因子机制,仿真结果表明在优化单峰函数上优于时变惯性权重^[13]。Arumugam 等提出了 GLBEST-PSO 算法,该算法会在迭代过程中通过全局最优与个体最优的平均值来调整惯性权重,实验结果验证了该算法的有效性^[14]。Chegini 等借鉴混合 PSO 算法、正余弦算法和 Levy 飞行方法,提出了 PSOSCALF 算法,该算法增强了 PSO 算法的勘探能力,可以有效应对具有诱惑性的局部最优位置^[15]。总之,为了应对不同问题,采用不同策略改进粒子群算法已经成为了一种主流办法。

从目前来看,混沌映射已经成为实现对优化算法改进的一种常用有效手段。该方法基于混沌理论,通过混沌技术产生随机数代替常规伪随机数生成群体的初始位置,且在大量文献中已得到验证。范九伦等受分段 Tent 映射的启发,构建了一种新

的分段 Logistic 混沌映射^[16]。相较于分段 Tent 映射,它生成的序列避免了计算机由于精度不足而退化为周期序列的问题;同时相较于普通 Logistic 映射,随机分布更均匀,分叉速度更快,效率更好。陈志刚等为了改善 Logistic 序列在特定值出现不动点等吸引子问题,将出现空白区序列的 Logistic 映射参数 μ 设定 4,实验证明了改进的 Logistic 映射随机性更好,同时计算简单^[17]。刘志强等利用 Tent 混沌映射丰富了狼群算法^[5]的初始化,提出了 TG-WO,并通过实验验证了该算法在路径寻优上的优越性^[18]。刘金源等对 Sine 混沌映射进行了改进,以解决 Sine 映射分布不均匀的问题^[19]。

鉴于此,本文提出一种基于改进 Sine 混沌映射的新型粒子群优化算法,该算法的主要创新点如下。

(1)在位置更新公式中结合高斯变异思想,提出两种新型位置更新公式,旨在优化过程中动态平衡局部搜索与全局搜索之间的关系提高粒子运动的多样性,并在粒子陷入局部最优时能有效跳出。

(2)引入改进 Sine 混沌映射生成 PSO 算法的初始种群,以解决传统 PSO 算法初始种群分布不均的问题,并提高种群的多样性。

1 标准粒子群优化算法

PSO 算法运行时, N 个粒子随机在给定的 n 维空间中生成速度矢量 $\mathbf{V}_i=(V_{i1}, V_{i2}, \dots, V_{in})$ 和位置矢量 $\mathbf{X}_i=(X_{i1}, X_{i2}, \dots, X_{in})$ 。之后,各粒子通过迭代更新位置与速度

$$\mathbf{V}_i(t+1)=\omega\mathbf{V}_i(t)+c_1r_1(\mathbf{X}_{pbi}(t)-\mathbf{X}_i(t))+c_2r_2(\mathbf{X}_{gb}(t)-\mathbf{X}_i(t)) \tag{1}$$

$$\mathbf{X}_i(t+1)=\mathbf{X}_i(t)+\mathbf{V}_i(t+1) \tag{2}$$

式中: $\mathbf{V}_i(t+1)$ 和 $\mathbf{X}_i(t+1)$ 分别为粒子 i 经 t 代更新后

的速度和位置; c_1 和 c_2 为加速系数; r_1 和 r_2 为介于 $[0, 1]$ 之间的随机数; w 为惯性权重; $\mathbf{X}_{pbi}(t)$ 为粒子 i 在 t 代中出现的最佳位置; $\mathbf{X}_{gb}(t)$ 为整个群体在在 t 代中出现的最佳位置。 $\mathbf{X}_{pbi}(t)$ 和 $\mathbf{X}_{gb}(t)$ 的更新公式为

$$\mathbf{X}_{pbi}(t+1)=\begin{cases}\mathbf{X}_i(t+1), f(\mathbf{X}_i(t+1))<f(\mathbf{X}_{pbi}(t)) \\ \mathbf{X}_{pbi}(t), f(\mathbf{X}_i(t+1))>f(\mathbf{X}_{pbi}(t))\end{cases}\quad (3)$$

$$\mathbf{X}_{gb}(t)=\min\{\mathbf{X}_{pb1}(t), \mathbf{X}_{pb2}(t), \cdots, \mathbf{X}_{pbN}(t)\}\quad (4)$$

此外,为防止粒子盲目搜索及运动幅度过大而错过最佳位置,还需对粒子速度与位置设定合适的搜索空间。设定粒子位置的上、下限分别为 $\mathbf{X}_{\min}$ 与 $\mathbf{X}_{\max}$,粒子在第 k 维的速度范围为 $[-\mathbf{V}_{k\max}, \mathbf{V}_{k\max}]$,一般采用该维位置变化范围的 20%作为 $\mathbf{V}_{k\max}$ 。

2 改进的 Sine 混沌映射

混沌映射的原理是首先利用混沌映射规则将优化变量映射到混沌变量的区间值空间,其次利用混沌变量特性进行寻优,最后将优化解线性转化为优化空间。常见的几种混沌映射公式如表 1 所示。

表 1 几种常见的混沌映射

Table 1 Introduction of several common chaotic mappings

映射类型	公式	搜索范围
Logistic	$x_{k+1}=\lambda x_k(1-x_k), \lambda \in (0, 4)$	$[0, 1]$
Cubic	$x_{k+1}=\rho x_k(1-x_k^2)$	$[0, 1]$
Chebyshev	$x_{k+1}=\cos(\arccos^{-1}(x_k)), a=0.5$	$[-1, 1]$
Tent	$x_{k+1}=\begin{cases}x_k/a, & x_k < a \\ (1-x_k)/(1-a), & x_k \geq a\end{cases}$	$[0, 1]$
Sine	$x_{k+1}=a\sin(\pi x_k), a \in [0, 1]$	$[0, 1]$
Sinusoidal	$x_{k+1}=ax_k^2\sin(\pi x_k), a \in [0, 4]$	$[0, 1]$

$$\mathbf{X}_i(t+1)=\begin{cases}\mathbf{X}_i(t)+\mathbf{V}_i(t+1), r_3>0.5\sin\left(\frac{t\pi}{2t_{\max}}\right) \\ \mathbf{X}_i(t)+u(\mathbf{X}_{gb}(t)-\mathbf{X}_i(t)), \lambda_{i\max}(t)>\alpha \\ (1+v)\mathbf{X}_i(t), \lambda_{i\max}(t)\leq\alpha\end{cases}\quad (6a)$$

式中: $t_{\max}$ 为总迭代次数; r_3 为 $[0, 1]$ 之间的随机数; α 为变异因子; v 和 u 是服从高斯分布的随机数; $\lambda_{i\max}(t)$ 是所有维度中 $\mathbf{X}_i(t)$ 和 $\mathbf{X}_{gb}(t)$ 之间的最大差值的权重,可以描述为

$$\begin{cases}\lambda_{i\max}(t)=\max\{\lambda_{i1}(t), \lambda_{i2}(t), \cdots, \lambda_{ik}(t), \cdots, \lambda_{in}(t)\} \\ \lambda_{ik}(t)=\frac{|x_{ik}(t)-x_{gbk}(t)|}{|x_{gbk}(t)|}\end{cases}\quad (7)$$

其中, $x_{ik}(t)$ 是第 t 次迭代时第 i 个粒子位置的第 k

由于不同的混沌映射具有不同的性质,故不同的混沌映射对优化算法的影响也很大。Logistic 映射具有较强的空间遍历性,但是会存在聚集区与空白区;Cubic 和 Chebyshev 映射具有初值敏感性,但是映射值在区间 $[0, 1]$ 内分布不均匀;Tent 映射具有良好的相关性,同时概率密度分布均匀,但是迭代后期容易衰退为周期序列;Sine 映射结构简单,效率高,但是同样存在概率密度分布不均的缺点;Sinusoidal 映射具有非线性反馈机制和初值敏感性,但存在空白区和不动点问题。

因此,鉴于普通混沌映射都存在着不足,往往都是选择改进的混沌映射作为群智能优化算法的初始化。其中改进最多两种便是 Logistic 映射和 Sine 映射,鉴于改进后的 Sine 混沌映射空间遍历性好,种群多样性丰富,相平面内分布均匀,同时结构简单、效率高,故本文选取改进 Sine 混沌映射作为改进粒子群算法的初始化方法。改进 Sine 混沌公式如下

$$\begin{cases}a_{i+1}=\sin(\kappa\pi a_i) \\ b_{i+1}=\sin(\kappa\pi b_i) \\ y_{i+1}=(a_{i+1}+b_{i+1})\%1\end{cases}\quad (5)$$

式中: a_i 和 b_i 的初始值取值范围为 $(0, 1)$;控制参数 k 为 1 200; y_{i+1} 为迭代混沌序列值;%为取余数运算。

3 提出的新型粒子群算法

为了在优化过程的不同阶段动态协调全局搜索和局部搜索,提出了新的粒子位置更新方程,以提高找到最优解的速度和精度,而不削弱其全局搜索能力。新的粒子位置更新方程为

$$\text{维值}, x_{gbk}(t) \text{ 是第 } t \text{ 次迭代时全局最佳位置的第 } k \text{ 维值}, \lambda_{ik}(t) \text{ 是 } x_{ik}(t) \text{ 与 } x_{gbk}(t) \text{ 之间的差值与 } x_{gbk}(t) \text{ 的比值。应注意,随机数 } u \text{ 和 } v \text{ 服从高斯分布}$$

$$\begin{cases}u\sim N(1, \sigma_u^2), \sigma_u=\sqrt{\mu\cos\left(\frac{(t-1)\pi}{2t_{\max}}\right)} \\ v\sim N(r_4, \sigma_v^2), \sigma_v=r_5\end{cases}\quad (8)$$

式中: r_4 为 $[-1.5, 1.5]$ 之间的随机数; r_5 为 $[0, 1]$ 之间的随机数; μ 是由自适应系数,公式为

$$\mu = \begin{cases} \frac{20}{3}\lambda_{\max}^2(t) - 7\lambda_{\max}(t) + \frac{7}{3}, \\ \alpha < \lambda_{\max}(t) \leq 0.5 \\ 0.3, \lambda_{\max}(t) > 0.5 \end{cases} \quad (9)$$

根据式(8)、(9)可以看出,服从高斯分布的随机数 u 的浓度不仅受迭代次数的影响,还受参数 $\lambda_{\max}(t)$ 的影响。

对于所提出的新的位置更新式(6),改进的部分等式(6b)的启用概率随着迭代次数非线性增加,并且等式(6b)中服从高斯分布的随机数 u 可以在迭代过程中自适应地调整聚散程度。这样的设计可以使算法在前半部分具有强大的全局搜索能力,并在后半部分增强其局部搜索能力。

从式(6)可以看出,在算法的早期迭代中,新的位置更新等式(6b)的启用概率略小,这导致更好的全局搜索能力。此外,即使算法在早期阶段使用了新的粒子更新公式,这也不会削弱其探索能力,而是增强了粒子运动的多样性。这是由于该算法在实现的早期阶段迭代次数较少,并且粒子与后期阶段相比更加分散,这使得服从高斯分布的随机数 u 更加分散,从而增强了粒子的探测能力。从式(6)中还可以看出,在随后的迭代中,新的位置更新等式(6b)的启用概率增加,从而增强算法的局部搜索能力。这是由于在算法的多次迭代之后,粒子更靠近,并且随机数 u 的分布趋于集中。因此,如果此时使用此公式来更新粒子的位置,则会导致粒子在全局最优位置附近快速聚集,而不受速度的影响,这增强了在全局最佳位置附近的搜索能力。为此,该设计不仅提高了粒子的运动多样性,更好地平衡了探索和开发,还提高了算法的收敛性。

当要更新的粒子使用新的粒子位置更新方程(6b)并且粒子非常接近全局最佳位置($\lambda_{\max}(t) \leq \alpha$)时,会导致粒子很少或没有移动。这种粒子称为惰性粒子。对于上述变化系数 α 的选择,有两个考虑因素。一个是 α 是 $[0, 0.1]$ 之间的随机数,另一个则是 $\alpha=0$ 。前者称为 NIPSO 算法,后者称为 NIPSO-0 算法。对于前者,当粒子的潜力不足时, NIPSO 算法可能会跳出局部最优,而对于后者, NIPSO-0 算法可能仅在粒子的所有潜力都被利用后才跳出。此外,所设计的新颖位置更新公式为处理陷入局部最优提供了一种新的解决方案。该解决方案将服从时变高斯分布的随机数 v 引入到新的位置更新方程(6b)中。通过这种方式,粒子被重新激活以保持它们在搜索空间中的活动,从而有效地防止其进入惰性状态。

综上所述可知,所提出的 NIPSO 和 NIPSO-0

算法不仅可以提高粒子的运动多样性,动态协调探索与开发的关系,而且可以有效激活粒子,防止其进入惰性状态。

考虑到惯性权重和加速度常数系数对寻找全局最优的影响,在文献[14]后,加入了个体和全局最优适应度的影响,使惯性权重自适应调整,而加速度系数则直接采用了文献[13]的表达方式。因此,惯性系数 w_i 和加速度系数 c_1 、 c_2 可以分别表示为

$$w_i = \begin{cases} \left| \frac{w_{\max} f(\mathbf{X}_{gb}(t))}{f(\mathbf{X}_{pbi}(t))} - \frac{(w_{\max} - w_{\min})t}{t_{\max}} \right|, \\ f(\mathbf{X}_{pbi}(t)) > 0 \\ w_{\max} - \frac{(w_{\max} - w_{\min})t}{t_{\max}}, \\ f(\mathbf{X}_{pbi}(t)) = 0 \\ \left| \frac{w_{\max} f(\mathbf{X}_{pbi}(t))}{f(\mathbf{X}_{gb}(t))} - \frac{(w_{\max} - w_{\min})t}{t_{\max}} \right|, \\ f(\mathbf{X}_{pbi}(t)) < 0 \end{cases} \quad (10)$$

$$\begin{cases} c_1 = 2.5 - \frac{2t}{t_{\max}} \\ c_2 = 0.5 + \frac{2t}{t_{\max}} \end{cases} \quad (11)$$

式中: $w_{\min}$ 为惯性权重的最小值; $w_{\max}$ 为惯性权重的最大值; f_{pbi} 为粒子的个体最优适应性; f_{gb} 为粒子的全局最优适应性。

4 实验结果与讨论

为了更好地评估所提出的算法的能力,使用 23 个具有不同特征的基准函数进行测试^[20]。这些函数包括 7 个单峰基准函数($F_1 \sim F_7$)、6 个多峰基准函数($F_8 \sim F_{13}$)和 10 个多峰低维基准函数($F_{14} \sim F_{23}$)。将提出的算法与几个典型的改进 PSO 算法进行了比较,包括 PSO-LDIW^[11]、PSO-RIW^[12]、PSO-TVAC^[13] 和 GLBEST-PSO^[14]。

由于元启发式算法的随机性较强,仅运行一次对每个基准函数的仿真测试结果是不可靠的。因此,为了确保每种算法的仿真测试结果具有说服力,更加公平地比较每种算法的性能,将粒子群人口规模和迭代次数分别设置为 50 和 5 000,每个基准函数的测试次数为 30。为了清楚地观察和比较对基准函数的测试结果,以相应结果的平均值 R_m 、标准差 R_{std} 和最佳值 R_b 作为评价指标,并列于下文的表中。将 R_m 用作性能指标,后文表中每个基准函数的最佳均值用下划线标出。

4.1 开发能力评估

6 种优化算法 (NIPSO、NIPSO-0、PSO-LDIW、PSO-TVAC、PSO-RIW、GLBEST-PSO) 对单模态

基准函数($F_1 \sim F_7$)的测试结果如表 2 所示。

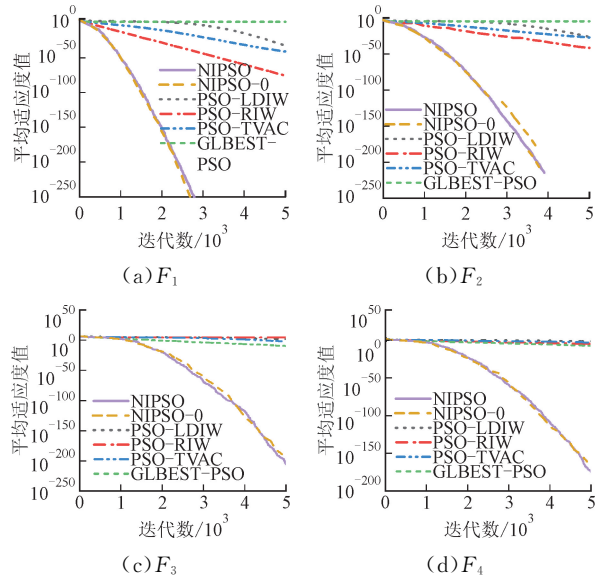
表 2 单峰函数算法的结果

Table 2 Results of algorithms on unimodal functions

函数	指标	数值					
		NIPSO	NIPSO-0	PSO-LDIW	PSO-TVAC	PSO-RIW	GLBEST-PSO
F_1	R_m	<u>0</u>	<u>0</u>	7.07×10^{-34}	6.08×10^{-42}	1.77×10^{-76}	4.96×10^1
	R_{std}	0	0	2.63×10^{-33}	1.01×10^{-41}	7.21×10^{-76}	1.01×10^2
	R_b	0	0	1.62×10^{-37}	1.32×10^{-43}	7.92×10^{-81}	3.63×10^{-70}
F_2	R_m	<u>0</u>	<u>0</u>	4.67×10^{-23}	5.04×10^{-24}	4.94×10^{-39}	2.32
	R_{std}	0	0	9.47×10^{-23}	4.41×10^{-24}	2.58×10^{-38}	1.76
	R_b	0	0	3.54×10^{-25}	4.99×10^{-25}	2.57×10^{-46}	1.37×10^{-1}
F_3	R_m	<u>8.74×10^{-207}</u>	1.03×10^{-195}	3.49×10^{-1}	3.84×10^{-5}	1.67×10^2	2.27×10^{-12}
	R_{std}	4.79×10^{-206}	5.65×10^{-195}	3.04×10^{-1}	1.03×10^{-4}	9.13×10^2	8.41×10^{-12}
	R_b	7.67×10^{-268}	5.79×10^{-253}	8.91×10^{-2}	8.30×10^{-7}	2.01×10^{-8}	1.07×10^{-15}
F_4	R_m	<u>7.82×10^{-176}</u>	1.07×10^{-167}	1.66×10^{-1}	3.12×10^{-3}	3.71×10^{-5}	1.02×10^{-7}
	R_{std}	4.28×10^{-175}	5.84×10^{-167}	8.47×10^{-2}	2.32×10^{-3}	5.65×10^{-5}	1.37×10^{-7}
	R_b	6.29×10^{-208}	4.77×10^{-209}	6.58×10^{-2}	4.29×10^{-4}	1.94×10^{-6}	4.46×10^{-9}
F_5	R_m	8.83	<u>1.73</u>	3.62×10^1	2.46×10^1	2.02×10^1	3.44×10^2
	R_{std}	1.84	1.34	2.83×10^1	2.74×10^1	2.01×10^1	9.31×10^2
	R_b	1.90	7.65×10^{-3}	3.10×10^{-1}	1.02×10^{-1}	3.10	1.25×10^1
F_6	R_m	1.08×10^{-9}	1.40×10^{-9}	2.47×10^{-32}	5.94×10^{-29}	<u>2.36×10^{-33}</u>	6.74×10^1
	R_{std}	1.17×10^{-9}	1.94×10^{-9}	4.08×10^{-32}	3.14×10^{-29}	4.63×10^{-33}	1.07×10^2
	R_b	7.81×10^{-11}	3.90×10^{-11}	0	1.17×10^{-29}	0	3.08×10^{-33}
F_7	R_m	<u>2.80×10^{-4}</u>	8.15×10^{-3}	4.23×10^{-3}	3.53×10^{-2}	2.01×10^{-3}	1.27×10^{-2}
	R_{std}	2.88×10^{-4}	1.12×10^{-3}	1.37×10^{-3}	1.65×10^{-2}	6.12×10^{-4}	4.30×10^{-3}
	R_b	4.86×10^{-6}	2.33×10^{-3}	1.08×10^{-3}	9.21×10^{-3}	8.72×10^{-4}	4.50×10^{-3}

从表 2 可以看出,对于函数 F_1 和 F_2 , NIPSO 算法和 NIPSO-0 算法都找到了最优解,并列第一。对于函数 F_3 、 F_4 和 F_7 , NIPSO 在整体优化结果的均值上排在了第一,体现了 NIPSO 优秀的算法性能。对于函数 F_5 , NIPSO-0 的均值和全局最优解均排在了第一。虽然在对函数 F_6 的测试中, PSO-LDIW、PSO-TVAC 和 PSO-RIW 的平均值和全局最优解都超过了 NIPSO 和 NIPSO-0,但从整体性能来看, 7 个单峰测试函数中 NIPSO 或 NIPSO-0 均值排第一的有 6 个,表现出良好的性能。

图 1 为 6 种粒子群优化算法在单峰测试函数上的收敛情况。可以清楚地看出, NIPSO 算法与 NIPSO-0 算法在函数 $F_1 \sim F_4$ 上有着几乎一致的收敛率,而且相比较其他粒子群算法,收敛速度更快。



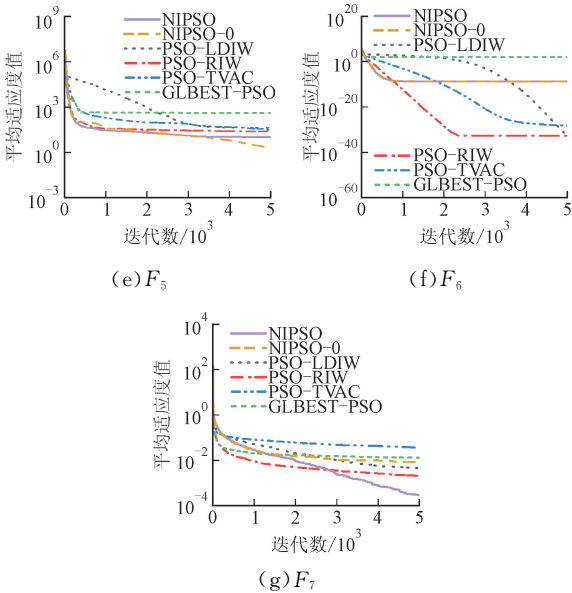


图 1 单峰函数算法的收敛曲线

Fig. 1 Convergence curves of algorithms on unimodal functions

从上述的分析中可以得出结论:本文所提出的 NIPSO 和 NIPSO-0 算法在与其他粒子群算法的比

较中表现出优秀的开发能力,不仅收敛速度快,而且求解到的最优值也更好,并且同时兼顾了算法的效率和精度。NIPSO 和 NIPSO-0 相比,NIPSO-0 在寻找全局最优时略胜一筹,除 F_1 和 F_2 两个函数都寻到最优外, $F_4 \sim F_6$ 3 个测试函数都是 NIPSO-0 的全局最优好于 NIPSO。但是,在历史最优适应度值平均值这一数值比较中,NIPSO 算法有 4 个测试函数的结果好于 NIPSO-0 算法,稳定性更好,并且除 F_5 之外全局最优值解也是与 NIPSO 是同一数量级的。故首先从算法性能的稳定性的出发,再考虑全局最优解的情况下,在处理单峰函数上 NIPSO 是首选。

4.2 勘探能力评估

多峰函数如 $F_8 \sim F_{23}$ 有多个局部最小值,这些函数可以用来测试优化算法跳出局部最小值并搜索全局最优的能力。表 3 和图 2 为 6 种优化算法对多峰高维函数 $F_8 \sim F_{13}$ 的测试结果。可以看出,与其他算法相比,所提出的 NIPSO 在上述多峰高维函数上可以实现更高的寻优精度。综合可知,NIPSO 在寻找多峰高维函数最优解的过程中具有更好的探索能力。

表 3 多峰高维函数的算法结果

Table 3 Results of algorithms on multimodal high-dimensional functions

函数	指标	数值					
		NIPSO	NIPSO-0	PSO-LDIW	PSO-TVAC	PSO-RIW	GLBEST-PSO
F_8	R_m	-1.01×10^4	-1.02×10^4	-7.53×10^3	-7.75×10^3	-7.02×10^3	-4.43×10^3
	R_{std}	4.43×10^2	4.08×10^2	5.37×10^2	6.02×10^2	6.76×10^2	1.79×10^3
	R_b	-1.08×10^4	-1.10×10^4	-8.60×10^3	-9.09×10^3	-8.20×10^3	-8.64×10^3
F_9	R_m	0	1.54×10^1	3.86×10^1	2.21×10^1	4.44×10^1	3.64×10^1
	R_{std}	0	1.25×10^1	1.15×10^1	8.54	1.45×10^1	1.24×10^1
	R_b	0	0	2.49×10^1	9.95	2.49×10^1	8.29
F_{10}	R_m	4.44×10^{-16}	2.58×10^{-15}	1.00×10^{-14}	5.56×10^{-14}	2.81×10^{-1}	4.96×10^{-1}
	R_{std}	0	2.00×10^{-15}	3.38×10^{-15}	8.97×10^{-14}	5.93×10^{-1}	1.04
	R_b	4.44×10^{-16}	4.44×10^{-16}	7.55×10^{-15}	2.18×10^{-14}	4.00×10^{-15}	7.55×10^{-15}
F_{11}	R_m	0	9.34×10^{-3}	2.19×10^{-2}	9.92×10^{-3}	1.06×10^{-2}	1.42
	R_{std}	0	2.22×10^{-2}	2.75×10^{-2}	1.30×10^{-2}	1.29×10^{-2}	2.08
	R_b	0	0	0	0	0	0
F_{12}	R_m	4.93×10^{-8}	6.19×10^{-7}	2.29×10^{-32}	2.35×10^{-30}	6.57×10^{-2}	3.26×10^1
	R_{std}	1.34×10^{-7}	1.95×10^{-6}	3.68×10^{-32}	1.33×10^{-30}	1.60×10^{-1}	8.11
	R_b	2.87×10^{-12}	1.42×10^{-11}	1.57×10^{-32}	7.72×10^{-31}	1.57×10^{-32}	2.00×10^1
F_{13}	R_m	2.36×10^{-9}	1.35×10^{-32}	1.10×10^{-3}	3.37×10^{-29}	1.83×10^{-3}	8.52×10^3
	R_{std}	4.17×10^{-9}	0	3.35×10^{-3}	1.32×10^{-29}	4.16×10^{-3}	4.71×10^3
	R_b	4.10×10^{-12}	1.35×10^{-32}	1.35×10^{-32}	1.29×10^{-29}	1.35×10^{-32}	1.24×10^3

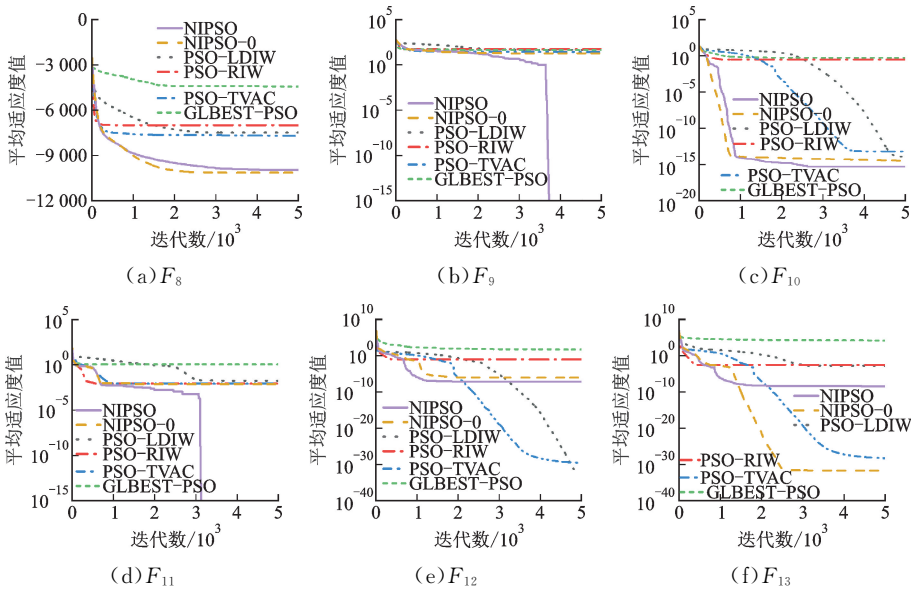


图 2 多峰高维函数算法的收敛曲线

Fig. 2 Convergence curves of algorithms on multimodal high-dimensional functions

表 4 和图 3 为 6 种优化算法对多峰低维函数 ($F_{14} \sim F_{23}$) 的测试结果。对于函数 F_{14} 、 F_{17} 和 F_{18} , 涉及的所有优化算法都能成功找到全局最小值。对于函数 F_{15} 、 F_{21} 、 F_{22} 和 F_{23} , NIPSO 排名第一。对于函数 F_{16} , 除 GLBEST-PSO 外, 其他算法都能成功找到最优解。对于函数 F_{20} , 各算法

的勘探能力都相差无几, 其中 PSO-TVAC 排在第一。对于函数 F_{21} , NIPSO 算法排在第一位。由此可以得出结论, NIPSO 在寻找多峰低维函数的最优解方面有更好的表现。因此可以得出结论, NIPSO 算法在所有上述算法中的探索能力是有竞争力的。

表 4 多峰低维函数的算法结果

Table 4 Results of algorithms on multimodal low-dimensional functions

函数	指标	数值					
		NIPSO	NIPSO-0	PSO-LDIW	PSO-TVAC	PSO-RIW	GLBEST-PSO
F_{14}	R_m	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}
	R_{std}	0	0	0	0	0	0
	R_b	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}
F_{15}	R_m	3.42×10^{-4}	4.05×10^{-4}	1.11×10^{-3}	3.45×10^{-4}	3.69×10^{-4}	3.74×10^{-3}
	R_{std}	1.68×10^{-4}	2.92×10^{-4}	3.65×10^{-3}	1.86×10^{-4}	2.32×10^{-4}	7.57×10^{-3}
	R_b	3.07×10^{-4}	3.07×10^{-4}	3.07×10^{-4}	3.07×10^{-4}	3.07×10^{-4}	3.07×10^{-4}
F_{16}	R_m	-1.03	-1.03	-1.03	-1.03	-1.03	-1.02
	R_{std}	0	0	0	0	0	1.49×10^{-2}
	R_b	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03
F_{17}	R_m	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
	R_{std}	0	0	0	0	0	0
	R_b	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
F_{18}	R_m	3.00	3.00	3.00	3.00	3.00	3.00
	R_{std}	0	0	0	0	0	0
	R_b	3.00	3.00	3.00	3.00	3.00	3.00

续表

函数	指标	数值					
		NIPSO	NIPSO-0	PSO-LDIW	PSO-TVAC	PSO-RIW	GLBEST-PSO
F_{19}	R_m	<u>-3.86</u>	<u>-3.86</u>	<u>-3.86</u>	<u>-3.86</u>	-3.84	<u>-3.86</u>
	R_{std}	0	0	0	0	0	0
	R_b	-3.86	-3.86	-3.86	-3.86	-3.86	-3.86
F_{20}	R_m	-3.27	-3.28	-3.25	<u>-3.29</u>	-3.26	-3.27
	R_{std}	5.92×10^{-2}	5.83×10^{-2}	5.99×10^{-2}	5.54×10^{-2}	6.03×10^{-2}	6.03×10^{-2}
	R_b	-3.32	-3.32	-3.32	-3.32	-3.32	-3.32
F_{21}	R_m	<u>-9.47</u>	-8.98	-6.72	-8.72	-6.88	-3.64
	R_{std}	1.76	2.44	3.38	2.45	3.23	1.48
	R_b	-1.02×10^1	-1.02×10^1	-1.02×10^1	-1.02×10^1	-1.02×10^1	-7.77
F_{22}	R_m	<u>-1.04×10^1</u>	-9.18	-7.74	-9.70	-7.76	-4.56
	R_{std}	2.22×10^{-7}	2.53	3.55	1.82	3.59	1.71
	R_b	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-8.11
F_{23}	R_m	<u>-1.00×10^1</u>	-9.39	-7.93	-9.61	-6.45	-4.75
	R_{std}	1.65	2.62	3.54	2.44	3.93	1.58
	R_b	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-8.28

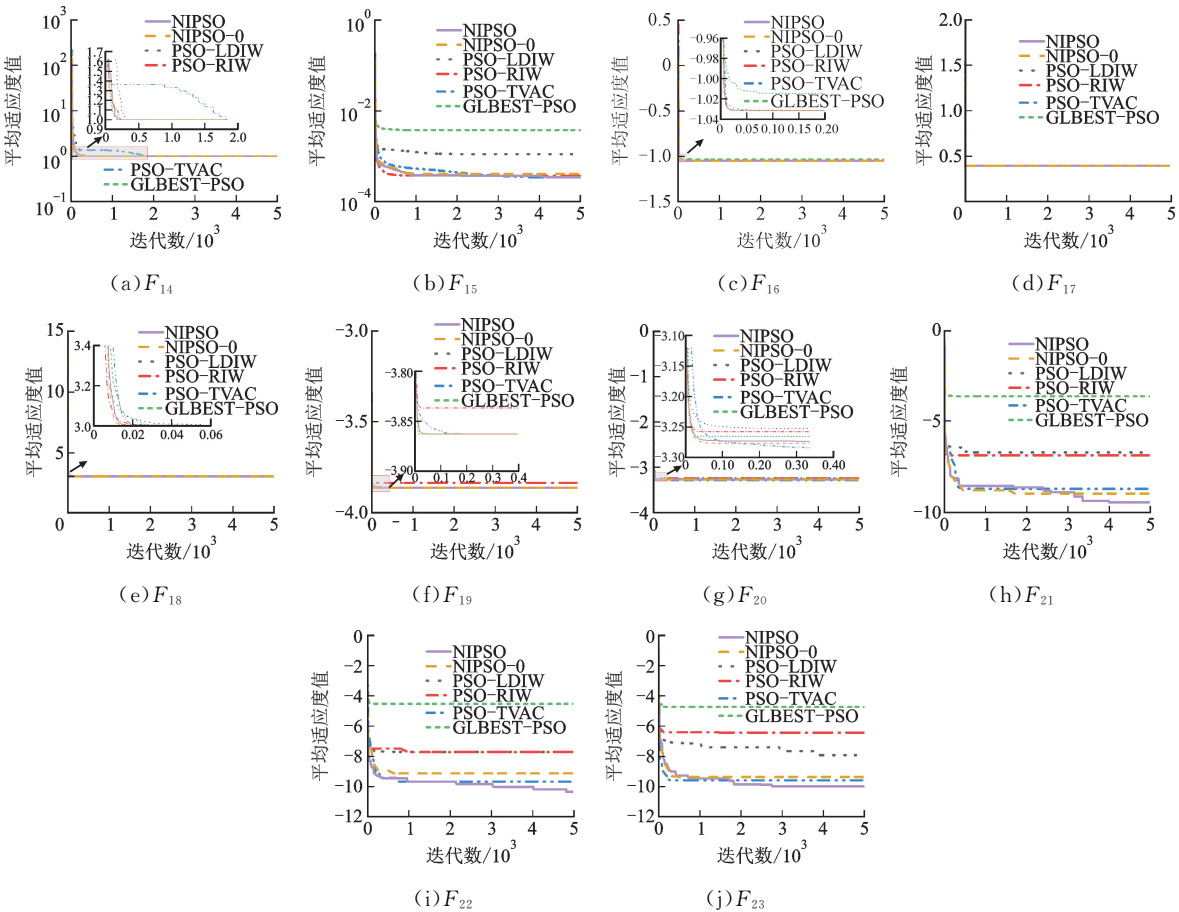


图 3 多峰低维函数算法的收敛曲线

Fig. 3 Convergence curves of algorithms on multimodal low-dimensional functions

此外,结合对 23 个测试函数的标准差,与其他算法相比可知,NIPSO 在函数 $F_1 \sim F_4$ 、 $F_6 \sim F_7$ 、 $F_9 \sim F_{19}$ 和 $F_{21} \sim F_{23}$ 的测试中排第一,在函数 F_5 和 F_8 中排第二。可以看出,NIPSO 算法具有良好的稳定性。综上所述可知,在综合性能方面,NIPSO 算法是首选。

4.3 经典工程问题的比较

除上述 23 个基准测试函数的评估方法外,还有一种常见的用于评估元启发式算法的方法,那就是应用改进算法来解决实际的工程问题。相比于基准函数,大多数经典工程问题的最优值是未知的,同时实际的问题中还有许多的制约因素。因此,通过约束问题来评估所提出的改进算法是必要的。在本节中,主要测试了 3 个受约束的工程问题:焊接梁、压力容器、工字钢设计问题。

4.3.1 焊接梁设计问题

焊接梁工程问题的适应度函数是焊接梁的施工成本。焊接梁设计示意如图 4 所示。

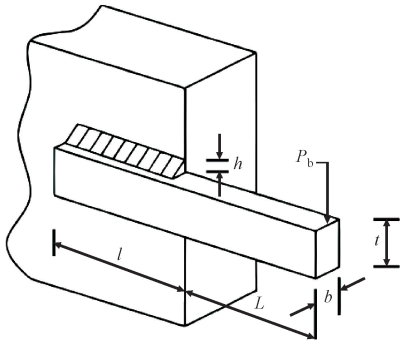


图 4 焊接梁设计示意

Fig. 4 Schematic of welded beam design

该问题包括 4 个约束条件,并且涉及到 4 个参数,焊接梁具体设计问题的公式如下

变量: $\mathbf{X} = [x_1, x_2, x_3, x_4] = [h, L, T, b]$

目标函数:

$$f(\mathbf{X}) = 1.104\,71x_1^2x_2 + 0.048\,11x_3x_4(14 + x_2)$$
(12)

约束条件:

$$\begin{cases} g_1(\mathbf{X}) = \tau(\mathbf{X}) - 13\,600 \leq 0 \\ g_2(\mathbf{X}) = \sigma(\mathbf{X}) - 30\,000 \leq 0 \\ g_3(\mathbf{X}) = \delta(\mathbf{X}) - 0.25 \leq 0 \\ g_4(\mathbf{X}) = x_1 - x_4 \leq 0 \\ g_5(\mathbf{X}) = 6\,000 - P_c(\mathbf{X}) \leq 0 \\ g_6(\mathbf{X}) = 0.125 - x_1 \leq 0 \\ g_7(\mathbf{X}) = 1.104\,7x_1^2 + \\ \quad 0.048\,11x_3x_4(14 + x_2) - 5 \leq 0 \end{cases}$$
(13)

变量区间: $0.1 \leq x_1, x_4 \leq 2; 0.1 \leq x_2, x_3 \leq 10$

式中 $\tau(\mathbf{X})$ 、 $\sigma(\mathbf{X})$ 、 $\delta(\mathbf{X})$ 、 $P_c(\mathbf{X})$ 参见文献[14]。

将本文提出的算法与其他几个经典粒子群算法比较,选取粒子数为 50,迭代数为 2 000,每个算法运行 30 次。焊接梁优化结果如表 5 所示。其中,所列 x_1 、 x_2 、 x_3 、 x_4 为最优解时的值。结果表明,NIPSO 的平均值排在了第 3,均方差排在了第二,整体性能仅次于 PSO-LDIW。

表 5 焊接梁优化结果

Table 5 Optimization results of welded beam

指标	数值					
	NIPSO	NIPSO-0	PSO-LDIW	PSO-TVAC	PSO-RIW	GLBEST-PSO
最优解时的 x_1	0.202 176 348	0.206 255 720	0.205 729 640	0.205 729 640	0.249 476 496	0.205 729 640
最优解时的 x_2	3.541 303 954	3.463 857 219	3.470 488 666	3.470 488 666	3.164 449 042	3.470 488 666
最优解时的 x_3	9.057 535 495	9.027 687 635	9.036 623 910	9.036 623 910	7.767 779 883	9.036 623 910
最优解时的 x_4	0.205 630 509	0.207 432 435	0.205 729 640	0.205 729 640	0.278 429 637	0.205 729 640
目标函数最优值	1.725 663 669	1.736 149 430	1.724 852 309	1.724 852 309	2.003 557 464	1.724 852 309
目标函数平均值	1.759 370 331	1.787 255 971	1.724 853 192	1.746 512 758	2.581 131 980	35.026 587 450
目标函数均方差	0.044 582 233	0.056 761 658	0.000 003 837	0.046 562 673	0.446 561 517	182.254 384 200
目标函数最差值	1.881 898 788	1.992 858 908	1.724 873 025	1.909 054 135	3.873 4420 580 1 000.000 000 000	

4.3.2 压力容器设计问题

压力容器设计问题的主要目的是将压力容器的

建造成本降至最低。施工总成本包括焊接费、成型加工费和材料费,并且受到压力容器几何尺寸的影响

响。压力容器的几何示意如图 5 所示。

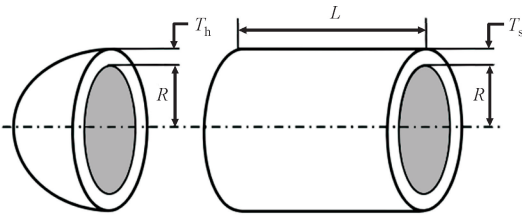


图 5 压力容器设计示意图

Fig. 5 Schematic of pressure vessel design

为了获得最低成本就必须计算出压力容器的最佳设计尺寸^[21],该问题的数学模型如下

变量: $\mathbf{X}=[x_1, x_2, x_3, x_4]=[T_s, T_h, R, L]$

目标函数:

$$f(\mathbf{X})=0.622\,4x_1x_3x_4+1.778\,1x_2x_3^2+3.166\,1x_1^2x_4+19.84x_1^2x_3$$

(14)

表 6 压力容器优化结果

Table 6 Optimization results of pressure vessel

指标	数值					
	NIPSO	NIPSO-0	PSO-LDIW	PSO-TVAC	PSO-RIW	GLBEST-PSO
最优解时的 x_1	0.812 5	0.812 5	0.812 5	0.812 5	0.812 5	0.812 5
最优解时的 x_2	0.437 5	0.437 5	0.437 5	0.437 5	0.437 5	0.437 5
最优解时的 x_3	42.098 445 60	42.098 445 60	42.098 445 60	42.098 445 60	42.071 814 44	42.098 445 60
最优解时的 x_4	176.636 595 8	176.636 595 8	176.636 595 8	176.636 595 8	176.966 901 0	176.636 595 8
目标函数最优值	6 059.714 335	6 059.714 335	6 059.714 335	6 059.714 335	6 062.960 814	6 059.714 335
目标函数平均值	6 152.816 051	6 399.876 304	6 279.668 706	6 413.091 309	6 500.484 093	6 471.718 241
目标函数均方差	112.499 744 1	326.521 206 5	155.996 341 2	281.346 957 3	312.187 844 3	406.675 181 5
目标函数最差值	6 410.458 597	7 332.841 508	6 410.086 760	7 332.841 508	7 318.708 742	7 544.492 518

4.3.3 工字梁设计问题

工字梁设计问题的主要目的是使工字梁的垂直偏转最小化,所以该问题的目标函数是梁的垂直偏转^[22]。工字梁设计示意如图 6 所示,该问题所涉及的设计参数为 长度 b 、高度 h 以及厚度 t_f 和 t_w 。

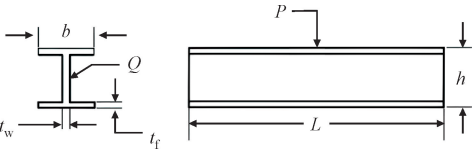


图 6 工字梁设计示意

Fig. 6 Schematic of I-beam design

工字梁问题的数学公式和约束条件如下

变量: $\mathbf{X}=[x_1, x_2, x_3, x_4]=[b, h, t_w, t_f]$ 。

约束条件:

$$\begin{cases} g_1(\mathbf{X})=-x_1+0.019\,3x_3\leqslant 0 \\ g_2(\mathbf{X})=-x_2+0.019\,3x_3\leqslant 0 \\ g_3(\mathbf{X})=-\pi x_3^2x_4-\frac{4}{3}\pi x_3^3+1\,296\,000\leqslant 0 \\ g_4(\mathbf{X})=x_4-240\leqslant 0 \end{cases}$$

(15)

变量区间: $1\times 0.062\,5\leqslant x_1, x_2\leqslant 99\times 0.062\,5$;
 $10\leqslant x_3, x_4\leqslant 200$ 。

压力容器优化结果如表 6 所示。可以看出,除 PSO-RIW 之外,所有算法都找到了同一最优解。但是,比较运行 30 次的平均值、均方差和最差值, NIPSO都排在了第一,故在该问题的求解上 NIPSO 的性能最好。

目标函数:

$$f(\mathbf{X})=\frac{5\,000}{[x_3(x_2-2x_4)^3]/12+x_1x_4^3/6+2x_1x_4[(x_2-x_4)/2]^2}$$

(16)

约束条件:

$$\begin{cases} g_1(\mathbf{X})=2x_1x_3+x_3(x_2-2x_4)\leqslant 0 \\ g_2(\mathbf{X})=\\ \frac{18x_2\times 10^4}{x_3(x_2-2x_4)^3+2x_1x_3(4x_4^2+3x_2(x_2-2x_4))}+\\ \frac{15x_1\times 10^3}{(x_2-2x_4)x_3^2+2x_3x_1^3}-6\leqslant 0 \end{cases}$$

(17)

变量区间: $10\leqslant x_1\leqslant 50$; $10\leqslant x_2\leqslant 80$; $0.9\leqslant x_3$;
 $x_4\leqslant 5$ 。

工字梁优化结果如表 7 所示。可以看出,NIPSO 和 PSO-LDIW、PSO-RIW 都找到了最优解,但是比较平均值和均方差两项数据,NIPSO 均排在第一,故 NIPSO 的稳定性更好,综合性能排第一。

表 7 工字梁优化结果
Table 7 Optimization results of I-beam

指标	数值					
	NIPSO	NIPSO-0	PSO-LDIW	PSO-TVAC	PSO-RIW	GLBEST-PSO
最优解时的 x_1	50.000 000 00	50.000 000 00	50.000 000 00	50.000 000 00	49.089 183 62	50.000 000 00
最优解时的 x_2	80	80	80	80	80	80
最优解时的 x_3	0.900 000 000	0.900 000 000	0.900 000 000	0.900 000 000	0.900 000 248	0.900 000 000
最优解时的 x_4	2.321 792 261	2.321 791 750	2.321 792 261	2.321 792 261	2.365 675 907	2.321 789 479
目标函数最优值	0.013 074 119	0.013 074 121	0.013 074 119	0.013 074 119	0.013 087 280	0.013 074 132
目标函数平均值	0.015 971 648	0.021 768 410	0.024 664 236	0.027 561 766	0.021 908 020	0.082 614 837
目标函数均方差	0.015 870 422	0.026 522 994	0.030 054 267	0.032 949 170	0.026 475 759	0.035 364 734
目标函数最差值	0.1	0.1	0.1	0.1	0.1	0.1

5 结 论

本文提出了一种基于改进 Sine 混沌映射机制的新型 PSO 算法。新的位置更新公式利用两个高斯变异算子来平衡算法的勘探和开发性能,并提供算法有效跳出局部最优的可能,使得算法在寻优能力和收敛精度都得到了显著提高。通过与 PSO-LDIW、PSO-TVAC、PSO-RIW 和 GLBEST-PSO 算法在大量基准测试函数和 3 个具有约束的工程优化问题上进行了对比仿真实验,发现新提出的算法具有寻优快、精度高、后期依旧能够跳出局部最优的能力。与 NIPSO-0 相比,NIPSO 的寻优更加稳定,故 NIPSO 更优秀。

在今后的研究中,可以通过混合多种元启发式算法或融入新的数学思想以增加种群多样性,平衡全局与局部搜索,跳出局部最优,从而实现更高的寻优精度和更快的收敛速度。此外,改进后的优化算法未来应可被应用在函数优化、路径规划、目标跟踪、控制器参数优化等方面^[23-25]。

参考文献:

[1] KENNEDY J, EBERHART R. Particle swarm optimization [C]//Proceedings of ICNN'95. Piscataway, NJ, USA: IEEE, 1995: 1942-1948.

[2] 王保民, 齐湛江, 闫瑞翔, 等. 基于随机权重粒子群算法的 SCARA 机器人动力学参数辨识 [J]. 西安交通大学学报, 2021, 55(9): 20-27.

WANG Baomin, QI Zhanjiang, YAN Ruixiang, et al. Parameter identification of SCARA robot based on ran-

dom weight particle swarm optimization [J]. Journal of Xi'an Jiaotong University, 2021, 55(9): 20-27.

[3] 高嘉乐, 邢清华, 李龙跃, 等. 采用投影螺旋搜索的改进粒子群算法 [J]. 西安交通大学学报, 2018, 52(6): 48-54.

GAO Jiale, XING Qinghua, LI Longyue, et al. An improved particle swarm optimization algorithm with projective spiral searches [J]. Journal of Xi'an Jiaotong University, 2018, 52(6): 48-54.

[4] 郭森, 秦贵和, 张晋东, 等. 多目标车辆路径问题的粒子群优化算法研究 [J]. 西安交通大学学报, 2016, 50(9): 97-104.

GUO Sen, QIN Guihe, ZHANG Jindong, et al. A novel particle swarm optimization for multi-objective vehicle routing problem [J]. Journal of Xi'an Jiaotong University, 2016, 50(9): 97-104.

[5] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69: 46-61.

[6] DHIMAN G, KUMAR V. Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems [J]. Knowledge-Based Systems, 2019, 165: 169-196.

[7] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.

[8] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.

- [9] SHAMI T M, EL-SALEH A A, ALSWAITTI M, et al. Particle swarm optimization: a comprehensive survey [J]. IEEE Access, 2022, 10: 10031-10061.
- [10] SHI Y, EBERHART R. A modified particle swarm optimizer [C]//Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. Piscataway, NJ, USA: IEEE, 1998: 69-73.
- [11] EBERHART R C, SHI Y. Comparing inertia weights and constriction factors in particle swarm optimization [C]//Proceedings of the 2000 Congress on Evolutionary Computation. Piscataway, NJ, USA: IEEE, 2000: 84-88.
- [12] EBERHART R C, SHI Yuhui. Tracking and optimizing dynamic systems with particle swarms [C]//Proceedings of the 2001 Congress on Evolutionary Computation. Piscataway, NJ, USA: IEEE, 2001: 94-100.
- [13] RATNAWEERA A, HALGAMUGE S K, WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240-255.
- [14] ARUMUGAM M S, RAO M V C. On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems [J]. Applied soft Computing, 2008, 8(1): 324-336.
- [15] CHEGINI S N, BAGHERI A, NAJAFI F. PSOSCALF: a new hybrid PSO based on Sine Cosine algorithm and levy flight for solving optimization problems [J]. Applied soft Computing, 2018, 73: 697-726.
- [16] 范九伦, 张雪峰. 分段 Logistic 混沌映射及其性能分析 [J]. 电子学报, 2009, 37(4): 720-725.
FAN Jiulun, ZHANG Xuefeng. Piecewise Logistic chaotic map and its performance analysis [J]. Acta Electronica Sinica, 2009, 37(4): 720-725.
- [17] 陈志刚, 梁涤青, 邓小鸿. Logistic 混沌映射性能分析与改进 [C]//2015 中国计算机网络安全年会论文集. 北京: 中国电子学会, 2015: 360-367.
- [18] 刘志强, 何丽, 袁亮, 等. 采用改进灰狼算法的机器人路径规划 [J]. 西安交通大学学报, 2022, 56(10): 49-60.
LIU Zhiqiang, HE Li, YUAN Liang, et al. Path planning of mobile robot based on TGWO algorithm [J]. Journal of Xi'an Jiaotong University, 2022, 56(10): 49-60.
- [19] 刘金源, 葛继科, 唐籍涛. 一种基于改进型 Sine 映射的快速混沌图像加密算法 [J]. 重庆科技学院学报(自然科学版), 2020, 22(5): 75-80, 90.
LIU Jinyuan, GE Jike, TANG Jitao. A fast chaotic image encryption algorithm based on improved sine map [J]. Journal of Chongqing University of Science and Technology(Natural Sciences Edition), 2020, 22(5): 75-80, 90.
- [20] NENAVATH H, KUMAR JATOTH D R, DAS D S. A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking [J]. Swarm and Evolutionary Computation, 2018, 43: 1-30.
- [21] GANDOMI A H, YANG Xinshe, ALAVI A H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems [J]. Engineering with Computers, 2013, 29(1): 17-35.
- [22] WANG G G. Adaptive response surface method using inherited Latin hypercube design points [J]. Journal of Mechanical Design, 2003, 125(2): 210-220.
- [23] QIAN Pengfei, LUO Hui, LIU Lei, et al. A hybrid Gaussian mutation PSO with search space reduction and its application to intelligent selection of piston seal grooves for homemade pneumatic cylinders [J]. Engineering Applications of Artificial Intelligence, 2023, 122: 106156.
- [24] 钱鹏飞, 罗辉, 单位银, 等. 新型双作用气浮气缸优化设计及其工况分析 [J]. 西安交通大学学报, 2022, 56(3): 12-21.
QIAN Pengfei, LUO Hui, SHAN Weiyin, et al. Optimal design and working condition analysis of a novel double-acting air-floating pneumatic cylinder [J]. Journal of Xi'an Jiaotong University, 2022, 56(3): 12-21.
- [25] 钱鹏飞, 浦晨玮, 刘磊, 等. 新型高频纵振减摩气缸的滑模运动轨迹跟踪控制 [J]. 西安交通大学学报, 2022, 56(10): 22-30.
QIAN Pengfei, PU Chenwei, LIU Lei, et al. Sliding mode motion trajectory tracking control of a novel high-frequency longitudinal vibration friction-reducing pneumatic cylinder [J]. Journal of Xi'an Jiaotong University, 2022, 56(10): 22-30.

(编辑 陶晴)