

# 基于改进引力搜索算法的桁架结构优化设计

李彦苍, 杨宗瑾

(河北工程大学, 河北 邯郸 056038)

**摘要:** 引力搜索算法是近几年提出的较有竞争力的群智能优化算法, 然而, 标准引力搜索算法存在后期收敛速度慢的缺点。为有效利用优化算法来解决结构优化的问题, 提出一种改进的引力搜索算法(improved gravitational search algorithm, IGSA)。通过引入 Logistic 映射, 使 GSA 初始种群遍历整个搜索空间, 提高算法找出最优解的可能性。通过引入粒子群算法(particle swarm optimization, PSO)的信息交互机制, 利用个体粒子历史最佳位置和种群历史最佳位置动态调整粒子的速度和位置, 使个体粒子更快地向适应度值更高的位置移动, 使算法搜索能力加强。对 6 个经典测试函数进行寻优, 结果表明改进后算法收敛速度快, 收敛精度高, 稳定性较好, 跳出局部最佳解的能力较强。用 IGSA 和 GSA 对 72 杆空间桁架进行尺寸优化, 与其他算法相比, 结果表明 IGSA 得到最优值的迭代次数明显减少, 得到的最优解明显优于通用算法。

**关键词:** 测试函数; 引力搜索算法; 自适应; 混沌映射; 桁架优化

中图分类号: TU323.4

文献标识码: A

文章编号: 1673-629X(2020)05-0049-07

doi:10.3969/j.issn.1673-629X.2020.05.010

## Optimization Design of Truss Structure Based on Improved Gravitational Search Algorithm

LI Yan-cang, YANG Zong-jin

(Hebei University of Engineering, Handan 056038, China)

**Abstract:** The gravitational search algorithm (GSA) is a competitive swarm intelligence optimization algorithm proposed in recent years. However, the standard gravitational search algorithm has the disadvantage of slow convergence in the later stage. In order to effectively solve the problem of structural optimization, an improved gravity search algorithm (IGSA) is proposed. By introducing Logistic mapping, the initial GSA population traverses the whole search space and improves the possibility of finding the optimal solution. Through the introduction of particle swarm optimization (PSO) information interaction mechanism, the particle speed and position can be dynamically adjusted by using the optimal position of individual particle history and the optimal position of population history, so that individual particles can move to the position with higher fitness value more quickly, and the search ability of the algorithm can be strengthened. Six classical test functions are optimized, and the results show that the improved algorithm has fast convergence speed, high convergence accuracy, high stability and strong ability to jump out of the local optimal solution. IGSA and GSA are used to optimize the size of the space truss of bar 72. Compared with other algorithms, the results show that the number of iterations of the optimal value obtained by IGSA is significantly reduced, and the optimal solution obtained is obviously superior to the general algorithm.

**Key words:** test function; gravitational search algorithm; adaption; chaos mapping; struss optimization

### 0 引言

对于桁架结构, 在给定结构形式、材料和形状的情况下, 优化各个杆件的尺寸, 使结构在满足约束条件的同时达到结构总质量最轻、设计合理性最佳、工程成本质量最轻, 即为结构优化<sup>[1-2]</sup>。结构优化设计变量多, 采用传统的计算方法解决这些问题面临着计算复杂度高、

计算时间长等问题, 对具有启发式特征的计算智能方法的研究越来越活跃。而过去十年里, 一系列智能算法正在兴起, 如遗传算法(genetic algorithm, GA)、粒子群优化算法(particle swarm optimization, PSO)、蚁群优化(ant colony optimization, ACO)、模拟退火(simulated annealing, SA)、和声搜索(harmony search, HS)和萤火虫

收稿日期: 2019-06-20

修回日期: 2019-10-22

网络出版时间: 2020-01-10

基金项目: 河北省高等学校科学技术研究重点项目(ZD2019114)

作者简介: 李彦苍(1974-), 男, 博士, 教授, 博导, 研究方向为计算机智能理论及其工程应用; 杨宗瑾(1994), 男, 研究生, 研究方向为计算机智能理论及其工程应用。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20200110.1118.014.html>

算法(firefly algorithm, FA)<sup>[3-8]</sup>。这些方法不需要传统的数学假设,因此与传统方法相比,增加了全局最优的定位概率。元启发式优化方法的随机性质可以允许在比传统优化方法中更大部分的搜索空间中进行探索。国内外学者也将智能算法应用到桁架结构优化中,如混沌增强碰撞体算法(enhanced chaotic colliding bodies algorithms, ECBO)<sup>[9]</sup>、改进二元蝙蝠柔性抽样算法(improved binary bat flexible sampling algorithm, IBBA)<sup>[10]</sup>、模拟植物生长算法等均对桁架进行结构优化<sup>[11]</sup>,有效推动了桁架结构优化的发展。但在算法收敛速度、稳定性方面还有研究的空间。

Rashedi 等人根据引力定律与运动定律,提出新的元启发式方法,即引力搜索算法(gravitational search algorithm, GSA)<sup>[12]</sup>。之后国内外学者对 GSA 进行了研究和改进,如文献[13]引入人工蜂群搜索机制,帮助算法跳出局部最优值;文献[14]通过动态调整引力常数,平衡全局搜索能力;文献[15]提出了一种神经和模糊重力搜索算法,加快了算法的收敛速度,改善了全局搜索能力。GSA 已被应用于无人机航路规划、电力控制、医学等领域<sup>[16-20]</sup>。虽然 GSA 具有高精度和稳定性的特点,但它具有易陷入局部最优、搜索精度低和收敛速度低的缺点。当种群中找到其当前的最佳位置时,其他群体将迅速收集到它。因此,GSA 陷入局部最优。

在该研究中,提出了改进引力搜索算法,用于桁架结构尺寸优化。提出的 IGSA 旨在消除 GSA 的缺点,并提高其对 GSA 的局部搜索能力。为了达到这个目的,通过引入混沌映射使 GSA 初始种群遍历整个搜索空间,提高算法找出最优解的可能性。通过引入粒子群算法中的粒子运动方程,提高粒子间的信息交流和记忆能力,以帮助算法逃出局部最优解,找到全局最优解。文中采用了 6 个典型的测试函数以验证 IGSA 的性能,优化结果表明,改进后的引力搜索算法收敛速度更快,收敛精度更高,更有可能找到全局最优解。文中对 72 杆空间桁架进行尺寸优化,用 IGSA 优化后的尺寸比 GSA 和其他算法得出的结果要小,说明 IGSA 是一种具有高全局搜索能力的强大优化算法。

## 1 基本引力搜索算法

引力搜索算法是从万有引力公式和牛顿第二定律中得到灵感,由 Rashedi 等人于 2009 年提出<sup>[12]</sup>。引力搜索算法(GSA)是一种新的群体优化智能算法。每一个粒子都有 4 个属性:位置,惯性质量,被动和主动引力质量。粒子的位置代表问题的潜在解,引力和惯性质量则用于适应度函数。

引力搜索算法中,首先对粒子的位置与速度执行初始化,之后通过迭代,更新粒子的位置与速度。如:第  $t$

次迭代中的第  $i$  个粒子与位置向量  $X_i(t) = \{x_i^1, \dots, x_i^d, \dots, x_i^D\}$  和一个速度矢量  $V_i(t) = \{v_i^1, \dots, v_i^d, \dots, v_i^D\}$ ,其中  $D$  是解空间的维度。

算法中每个粒子的被动和主动引力质量和惯性质量相等,均为  $M_i, i = 1, 2, \dots, N, N$  为种群规模。 $M_i$  的定义如下:

$$m_i(t) = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \quad (1)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (2)$$

其中,  $\text{fit}_i(t)$  为粒子  $i$  在第  $t$  次迭代时的适应度函数值,  $\text{best}(t)$  与  $\text{worst}(t)$  为第  $t$  次迭代时所有粒子中最优与最差适应度函数值。

在算法中,定义在第  $d$  维上,粒子  $j$  对粒子  $i$  的作用力大小  $F_{ij}^d$  为:

$$F_{ij}^d(t) = G(t) \frac{M_j(t) \times M_i(t)}{R_{ij}(t) + \varepsilon} \times (x_j^d(t) - x_i^d(t)) \quad (3)$$

其中,  $R_{ij}(t)$  为粒子  $i$  与粒子  $j$  之间的 Euclidean 距离,  $\varepsilon$  为极小的常数,防止分母为 0。 $G(t)$  为第  $t$  次迭代时引力常量取值,  $G$  是时间的递减函数。必须指出的是,  $G(t)$  对于确定 GSA 的性能很重要,定义如下:

$$G(t) = G_0 \exp(-\beta \frac{t}{t_{\max}}) \quad (4)$$

其中,  $G_0$  和  $\beta$  分别是  $G$  的初始值和常数,  $t_{\max}$  是最大迭代次数。

在第  $d$  维空间中,粒子  $i$  收到的总作用力来自于其他所有粒子的施加的力矢量和:

$$F_i^d(t) = \sum_{j \in \text{kbest}, j \neq i}^N \text{rand}_j \times F_{ij}^d(t) \quad (5)$$

其中,  $\text{kbest}$  定义为粒子质量按降序从最佳值到第  $k$  个粒子的集合,  $k$  在开始时初始化为  $k_0$  并随时间减小。这里,  $k_0$  设定为  $N$ ,并且线性减小到单位 1。 $\text{rand}_j$  为  $[0, 1]$  中均匀分布的随机数。

根据牛顿第二定律,粒子的加速度如下:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \quad (6)$$

粒子的速度和位置如下式:

$$v_i^d(t+1) = \text{rand}_i \times v_i^d(t) + a_i^d(t) \quad (7)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (8)$$

其中,  $\text{rand}_i$  是区间  $[0, 1]$  中均匀分布的随机数。

## 2 算法的改进

### 2.1 混沌初始化策略

混沌理论是指对混沌动力系统的研究。混沌系统是对其初始条件高度敏感的非线性动力系统。在初始

条件导致系统的结果变化很大。混沌具有遍历性和均匀性的特点,利用混沌的良好特征对 GSA 初始化,可以使 GSA 初始种群遍历整个搜索空间,提高算法找出最优解的可能性。文中采用 Logistic 映射进行种群初始化,其方程如下:

$$f(k+1) = af(k)(1-f(k)) \quad (k=0,1,2,\dots, \\ 0 \leq f(k) \leq 1) \quad (9)$$

当变量  $a=4$  时,该模式进入完全混沌状态。

混沌算法的步骤如下:

(1) 将  $n$  个初始值分别赋予式(9),可以得到  $n$  个不同的混沌值,令当前的最优适应值的粒子为  $x^*$ ,  $g(x^*)$  为当前最佳适应值。

(2) 利用混沌变量进行搜索。

$$x_j(k) = x_j^* + p_j t_j(k) - q_j \quad (10)$$

其中,  $p_j$  和  $q_j$  为常数且大于 0。通过对其值的调整,可以把混沌变量的取值范围转化为其对应的优化变量的取值范围。

根据下式计算性能指标:

$$g(k) = g(x(k)), x(k) = (x_1(k), x_2(k), \dots, x_n(k)) \quad (11)$$

(3) 如果  $g(k) < g(x^*)$ , 则  $x^* = x(k)$ , 循环达到指定的次数结束,否则,转到(2)进行循环。

利用混沌算法产生初始解的过程如下:

step1: 随机产生  $N$  个初始解,并将其映射到 Logistic 方程的定义域内。

step2: 用 Logistic 方程进行迭代产生混沌序列。

step3: 将产生的混沌序列映射到原来的定义域内。

step4: 将两类初始解进行排序,将适应度较优的解作为种群的初始解。

## 2.2 PSO-GSA 混合优化算法

基本的引力搜索算法 GSA 是依靠粒子间的相互作用力来传递信息和更新位置,然而,缺少对历史最优解和群体最优解的保存和学习功能,可能会丢失一些更好的解,不利于算法的收敛速度。

粒子群算法 PSO 是 Kennedy 和 Eberhart 提出的一种进化算法<sup>[21]</sup>。粒子群算法的灵感来源于鸟群的社会行为。它使用许多粒子在搜索空间中飞行,以找到最佳解。同时,粒子都会看到它们路径中的最佳粒子。粒子会考虑它们自己的最佳解,以及到目前为止发现的最佳解。

粒子群算法中的每一个粒子都要考虑当前位置、当前速度、距离到最优和距离到最优的距离,才能对粒子群的位置进行修正。

文中将粒子群算法中对个体历史最优解和群体最优解的记忆和学习的思想用于引力搜索算法,将

式(7)改进为:

$$v_i^d(t+1) = r_{i,1} v_i^d(t) + a_i^d(t) + c_1 r_{i,2} (pbest_i^d(t) - x_i^d(t)) + c_2 r_{i,3} (gbest^d(t) - x_i^d(t)) \quad (12)$$

其中,  $pbest_i^d$  是第  $d$  维中第  $i$  个粒子的历史最优解的位置,而  $gbest^d$  是第  $d$  维中所有粒子中最优解的位置。 $c_1, c_2$  为  $[0,1]$  的常数。 $r_{i,1}, r_{i,2}, r_{i,3}$  分别是区间  $[0,1]$  中的均匀随机数。

式(12)可能会影响到算法的探索阶段,因为该式是一个恒定的速度更新公式,为了防止新的更新速度法降低勘探能力,将  $c_1, c_2$  改为动态的变量。自适应地减少  $c_1$ , 增加  $c_2$ , 使得当算法到达开发阶段时,粒子趋向于向最优解快速移动。由于进化算法中勘探阶段和开发阶段没有明显的界限,因此这种自适应方法适应于这两个阶段。此外,这种自适应方法强调在第一个迭代中进行探索,在最后一个迭代中进行开发。

$c_1$  较高时,算法偏向于标准 GSA, 而较高的  $c_2$  体现 PSO 在执行搜索过程中粒子之间的传递信息的能力。动态的  $c_1, c_2$  使得算法可以像标准 GSA 探索搜索空间,并像 PSO 一样保存和利用在探索阶段搜索到的个体最优解和种群最优解。

## 2.3 改进算法的流程

文中所提出的改进引力搜索算法的流程如下:

step1: 利用混沌初始化粒子的位置  $X$ , 设定种群规模  $N$ , 维度  $\dim$  和最大迭代次数。

step2: 由目标函数计算每个粒子的适应度值。并保存、更新  $pbest$  和  $gbest$ 。

step3: 根据式(1)、式(2)、式(7)、式(8)分别计算每个粒子的惯性质量  $M(t)$ , 引力  $F_i^d(t)$  和加速度  $a_i^d$ 。

step4: 根据式(12)更新粒子的速度,根据式(8)更新粒子的位置。

step5: 保存、更新  $pbest$  和  $gbest$ 。

step6: 判断算法是否满足结束条件,若不满足,则返回 step2, 满足条件时,结束算法,输出最佳值。

图 1 为改进引力搜索算法的流程。

## 3 试验验证

为验证文中改进算法的有效性,将改进后的引力搜索算法 IGSA 与标准引力搜索算法 GSA 进行对比试验。

### 3.1 参数设置

种群规模  $N=50$ ; 最大迭代次数  $\max \text{it}=1000$ ;  $\beta=20$ ;  $G_0=100$ ; 种群维度  $\dim=30$ ;  $c_1 = (-2t^3/T^3) + 2$ ;  $c_2 = 2t^3/T^3$ , 其中  $t$  为当前迭代次数,  $T$  为最大迭代次数, 曲线如图 2 所示。

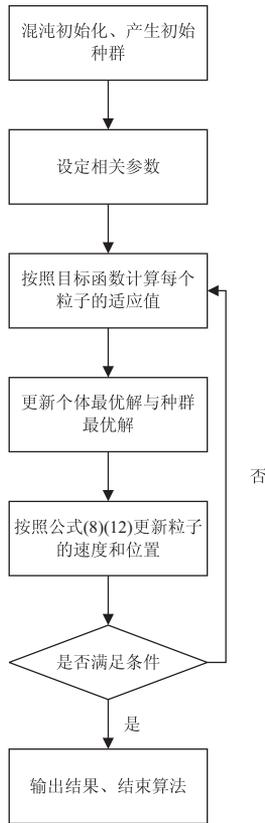


图 1 改进引力搜索算法的流程

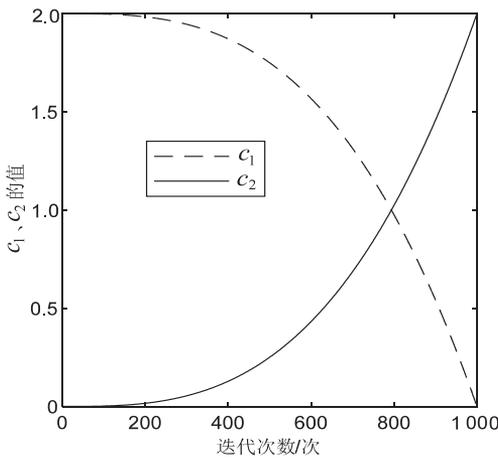
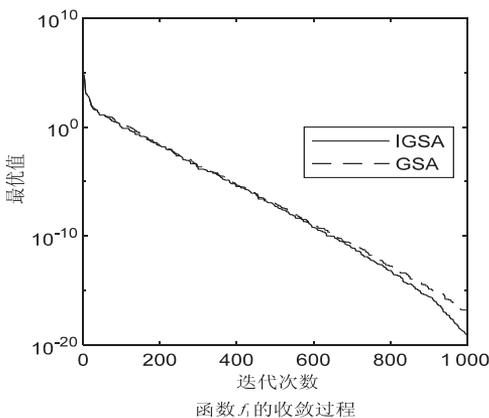


图 2 系数曲线



### 3.2 测试函数

通过表 1 中的 6 个标准测试函数对改进的引力搜索算法和标准引力搜索算法进行测试,并对结果进行分析。其中  $f_1$  和  $f_2$  为单峰值函数,算法对其收敛的速度是检验算法性能好坏的重要指标。 $f_3 \sim f_6$  为复杂的多峰值函数,存在较多的局部最优解,用于检验算法的全局搜索能力。

### 3.3 仿真与分析

分别用改进引力搜索算法 IGSA 和标准引力搜索算法 GSA 对 6 个测试函数进行求解,得出相应的最优值曲线,如图 3 ~ 图 5 所示。由图可知,改进后的算法比标准算法收敛速度更快,收敛精度更高,全局搜索能力更强。

表 1 标准测试函数

函数名	测试函数	范围
单峰值测试函数	$f_1 = \sum_{i=1}^d x_i^2$	$[-100, 100]^d$
单峰值测试函数	$f_2 = \sum_{i=1}^d  x_i  + \prod_{i=1}^d  x_i $	$[-10, 10]^d$
多峰高纬度测试函数	$f_3 = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$
多峰高纬度测试函数	$f_4 = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n \mu(x_i, 5, 100, 4)$	$[-50, 50]$
固定维度多峰测试函数	$f_5 = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	$[-65, 65]$
固定维度多峰测试函数	$f_6 = - \sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0, 16]$

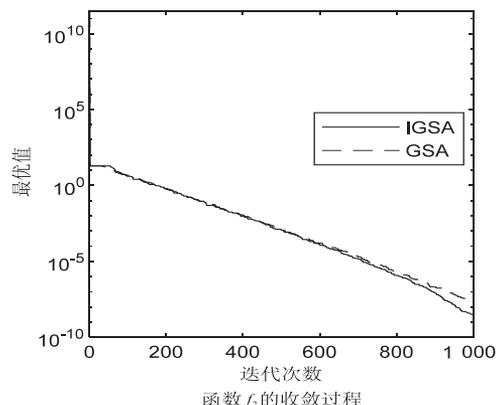


图 3 单峰值测试函数对比

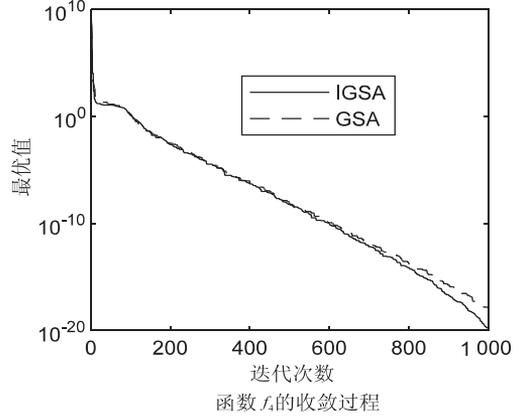
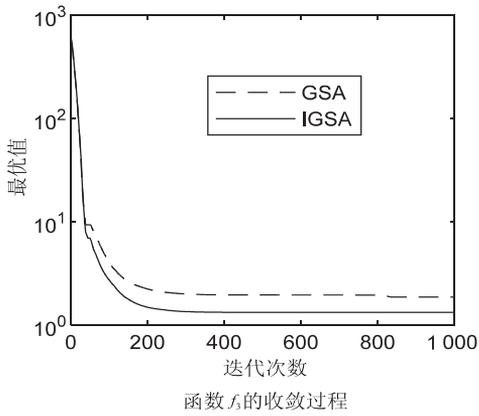


图4 多峰值函数对比

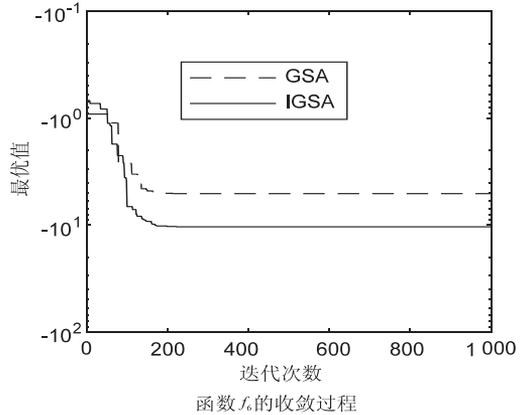
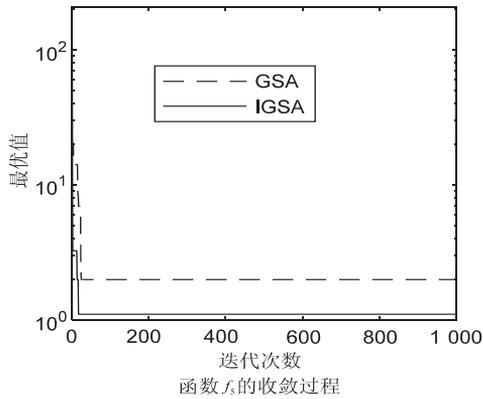


图5 固定维度多峰值函数对比

为进一步验证 IGSA 的优化性能,分别把 2 种算法运行 30 次,并在表 2 中显示出相应的最小值、平均值、方差。

出:对于  $f_1$  和  $f_2$  函数,IGSA 在最小值、平均值和方差方面均小于 GSA。从图 3 中可以得出,在保证得到较好的全局最优解的情况下,IGSA 的收敛速度要比 GSA 更快,并且精度更高。

(1)单峰值函数。对于  $f_1$ 、 $f_2$  来说,其收敛速度的快慢是检验算法性能的重要特点。在表 2 中可以看

表2 IGSA 与 GSA 算法结果对比

函数	算法	最小值	平均值	方差
$f_1$	GSA	7.552 1e-17	1.012 50e-16	4.62e-35
	IGSA	4.653 0e-18	4.741 5e-18	7.58e-38
$f_2$	GSA	5.140 4e-08	5.855 8e-08	1.08e-17
	IGSA	6.214 51e-09	6.314 1e-09	1.70e-18
$f_3$	GSA	13.119	13.119 9	53.87
	IGSA	4.109 8	4.109 8	17.99
$f_4$	GSA	0.098 7	0.109 9	2.91e-6
	IGSA	1.858 1e-19	1.947 4e-19	9.78e-8
$f_5$	GSA	2.844 6	2.982 1	12.68
	IGSA	1.992 0	1.992 0	6.49
$f_6$	GSA	-5.087 7	-5.087 7	12.53
	IGSA	-10.402 9	-10.402 9	4.74

(2)多峰值函数。多峰值函数存在较多的局部最优解,寻找全局最优解的难度较大。对于函数  $f_3 \sim f_6$

而言,算法寻优的难点在于搜索到全局最优解。从图 4 到图 5 可以看出:IGSA 搜索到最优值均小于 GSA。

可以从表 2 的结果中看到:在  $f_3 \sim f_6$  中,IGSA 搜索到的最小值、平均值和方差都小于 GSA。

### 4 桁架分析

选取了 1 个典型的桁架优化算例,验证了 IGSA 在求解约束桁架形状和尺寸优化问题上的有效性与可靠性。并将 IGSA 的优化结果与 GSA 和其他优化技术的优化结果进行了比较。提出的 IGSA 由 MATLAB 软件实现,版本为 9.3.0.713579 (R2017b),操作系统:Microsoft Windows 10 专业版 Version 10.0 (Build 17134)。算法最大迭代次数为 500。算法计算 10 次,取其最佳值作为计算最终结果值。

72 杆空间桁架结构,杆件分为 16 组,节点 1~6 沿  $x, y$  的最大位移限值是 6.35 mm,最大的许可应力为  $[-172.375, 172.375]$  MPa,密度  $\rho = 2\ 678\text{ kg/m}^3$ ,弹性模量  $E = 68\ 950\text{ MPa}$ ,长和宽均为 3.048 m,每节的高度为 1.524 m,荷载工况及作用位置与文献[22]相同。算法的控制参数设置为:最大迭代次数 500,搜索

空间维数 16,种群总数为 75。

在相同约束条件下,与其他文献中的结果进行比较,优化结果见表 3,图 6 给出了 IGSA 求解 72 杆桁架结构设计问题的典型收敛曲线。

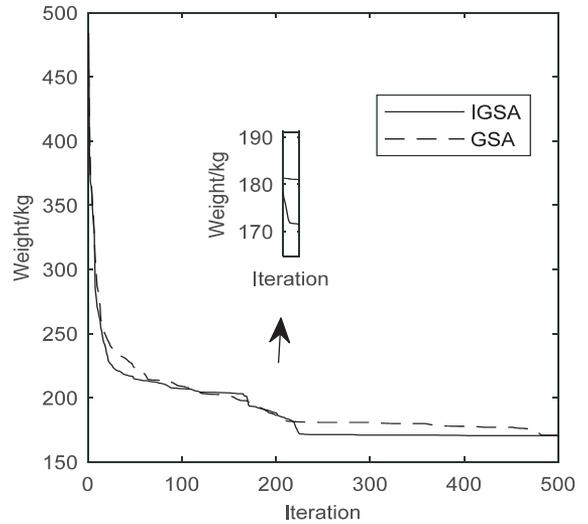


图 6 72 杆空间桁架结构的收敛曲线

表 3 72 杆空间桁架优化结果的比较

组号	杆件号	文献[22]	文献[23]	文献[24]	文献[25]	GSA	IGSA
A1	1~4	101.355	102.258	100.000	103.871	101.89	100.968
A2	5~12	345.548	382.967	345.161	359.354	342.981	350.257
A3	13~16	264.258	220.258	309.677	243.225	271.92	263.290
A4	17~18	367.290	391.999	335.483	326.451	361.63	369.870
A5	19~22	326.903	170.516	296.774	394.193	326.05	343.032
A6	23~30	335.483	353.548	341.935	343.225	340.12	331.225
A7	31~34	64.516	64.516	77.419	64.516	64.516	64.516
A8	35~36	64.516	97.355	106.451	64.516	64.516	64.516
A9	37~40	825.869	713.999	745.160	803.869	811.16	820.063
A10	41~48	332.128	373.741	377.419	338.064	326.04	332.322
A11	49~52	64.516	64.516	64.516	64.516	64.516	64.516
A12	53~54	64.516	64.516	64.516	64.516	64.516	64.516
A13	55~58	1 224.062	1 340.901 1	1 132.256 1	1 172.90	1 235.29	1 213.352
A14	59~66	332.774	324.774	325.806	338.516	337.48	331.741
A15	67~70	64.516	64.516	67.741	64.516	64.516	64.516
A16	71~72	64.516	64.516	100.000	64.516	64.516	64.516
重量		172.21	176.28	174.98	172.91	172.48	172.19

注:重量栏的单位为 kg,其余项单位为  $\text{mm}^2$ 。

从表 3 可以看出,IGSA 得到的 72 杆空间桁架优化后的重量为 172.19 kg,与 GSA 和文献[22-25]中得到的结果相比较,重量得到不同程度上的减小。从图 6 可以看出,改进后的引力搜索算法在第 220 次迭代时收敛到最优解,而标准引力搜索算法在第 480 次时收敛到相应的最优解,因此可以看出改进引力搜索算法搜寻最优解的能力优于标准引力搜索算法,且更

快收敛到最优解。

### 5 结束语

针对标准引力搜索算法收敛速度慢,易陷入局部最优值的问题,引入混沌映射,使得初始化时,粒子均匀地分布在搜索空间,使得算法更快地找到最优解,避免陷入局部最优解。引入粒子群算法中的粒子

个体的记忆性和信息交互机制,并动态调整学习因子,以加快算法的运算效率,更快找到较优值,使得算法避免陷入局部最优值,最终找到全局最优值。

通过6个测试函数,表明改进后的算法能够较好地找到全局最优解,寻优能力更强。通过对桁架进行结构优化,说明算法适用于空间桁架结构杆件截面尺寸优化设计。

算法对代理的速度公式和后期收敛速度做了改进,在引力常数方面还有待进一步研究。

#### 参考文献:

- [1] BANDYOPADHYAY D, SAHA S, SOHAIL T. Identification of parameter of truss structure by limited static strain measurement [J]. *Recent Advances in Structural Engineering*, 2019, 1: 375-388.
- [2] SHAKYA A, NANAKORN P, PETPRAKOB W. A ground-structure-based representation with an element-removal algorithm for truss topology optimization [J]. *Structural & Multidisciplinary Optimization*, 2018, 58(2): 657-675.
- [3] 张 鸽, 张 弘, 李宗亮. 混合遗传算法在塔吊装载问题中的应用 [J]. *计算机与数字工程*, 2018, 46(10): 1966-1969.
- [4] 陈 迪, 李 宁. 基于粒子群算法的医学图像分类算法研究 [J]. *电子设计工程*, 2019, 27(2): 189-193.
- [5] 许向阳, 吴泽华. 基于改进蚁群算法的消防路线规划系统 [J]. *通信技术*, 2018, 51(6): 1358-1364.
- [6] NORMASARI N M E, YU V F, BACHTIYAR C, et al. A simulated annealing heuristic for the capacitated green vehicle routing problem [J]. *Mathematical Problems in Engineering*, 2019(2): 1-18.
- [7] CHOUDHARY A, BAGHEL A S, SANGWAN O P. Efficient parameter estimation of software reliability growth models using harmony search [J]. *IET Software*, 2018, 11(6): 286-291.
- [8] ELKHECHAFI M, BENMAMOUN Z, HACHIMI H, et al. Firefly algorithm for supply chain optimization [J]. *Lobachevskii Journal of Mathematics*, 2018, 39(3): 355-367.
- [9] KAVEH A, DADRAS A, MONTAZERAN A H. Chaotic enhanced colliding bodies algorithms for size optimization of truss structures [J]. *Acta Mechanica*, 2018, 229(7): 2883-2907.
- [10] HAMZEKOLAEI N S, MIRI M, RASHKI M, et al. An improved binary bat flexible sampling algorithm for reliability-based design optimization of truss structures with discrete-continuous variables [J]. *Engineering Computations*, 2018, 35(2): 641-671.
- [11] 石开荣, 阮智健, 姜正荣, 等. 模拟植物生长算法的改进策略及桁架结构优化研究 [J]. *建筑结构学报*, 2018, 39(1): 120-128.
- [12] RASHEDI E, NEZAMABDI - POUR H, SARYAZDI S. GSA: a gravitational search algorithm [J]. *Information Sciences*, 2009, 179(13): 2232-2248.
- [13] 丁知平. 融合混沌反学习与蜂群搜索算子的引力搜索算法 [J]. *计算机系统应用*, 2018, 27(4): 196-201.
- [14] 李 静, 高兴宝. 基于动态引力常数和种群递减的引力搜索算法 [J]. *系统科学与数学*, 2018, 38(1): 78-85.
- [15] PELUSI D, MASCELLA R, TALLINI L, et al. Neural network and fuzzy system for the tuning of gravitational search algorithm parameters [J]. *Expert Systems with Applications*, 2018, 102(15): 234-244.
- [16] KHATIBINIA M, SALAJEGHEH E, SALAJEGHEH J, et al. Reliability-based design optimization of reinforced concrete structures including soil - structure interaction using a discrete gravitational search algorithm and a proposed metamodel [J]. *Engineering Optimization*, 2013, 45(10): 1147-1165.
- [17] 王 宇, 陈海涛, 李海川. 基于引力搜索算法的植保无人机三维路径规划方法 [J]. *农业机械学报*, 2018, 49(2): 28-33.
- [18] PEROVIĆ B D, TASIĆ D S, KLIMENTA D O, et al. Optimising the thermal environment and the ampacity of underground power cables using the gravitational search algorithm [J]. *IET Generation Transmission & Distribution*, 2018, 12(2): 423-430.
- [19] CHAWLA A, BHALJA B R, PANIGRAHI B K, et al. Gravitational search based algorithm for optimal coordination of directional overcurrent relays using user defined characteristic [J]. *Electric Power Components & Systems*, 2018, 46(2): 1-13.
- [20] 冯 飞, 刘培学, 李 丽, 等. FCM融合改进的GSA算法在医学图像分割中的研究 [J]. *计算机科学*, 2018, 45(S1): 252-254.
- [21] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// *Proceedings of IEEE international conference on neural networks*. Washington DC: IEEE, 1995: 1942-1948.
- [22] ZHOU M, ROZVANY G I N. DCOC: an optimality criteria method for large systems Part II: algorithm [J]. *Structural and Multidisciplinary Optimization*, 1993, 6(4): 250-262.
- [23] SCHMIT L A, FARSHI B. Some approximation concepts for structural synthesis [J]. *AIAA Journal*, 1974, 12(5): 692-699.
- [24] ERBATUR F, HASANTEBI O, TNTUNCN I, et al. Optimal design of planar and space structures with genetic algorithms [J]. *Computers & Structures*, 2000, 75(2): 209-224.
- [25] VENKAYYA V B. Design of optimum structures [J]. *Computers & Structures*, 1971, 1(1-2): 265-309.